African vultures optimization algorithm based Choquet fuzzy integral for global optimization and engineering design problems

Maha Nssibi, Ghaith Manita, Francis Faux, Ouajdi Korbaa, Elyes Lamine

African vultures optimization algorithm based Choquet fuzzy integral for global optimization and engineering design problems

Maha Nssibi · Ghaith Manita · Francis Faux · Ouajdi Korbaa · Elyes Lamine

Abstract

Addressing complex optimization problems demands innovative solutions capable of navigating the interdependencies among variables, a reality often oversimplified by traditional metaheuristics. To address this challenge, this paper presents an enhanced African Vultures Optimization Algorithm, termed *ci*-AVOA, that incorporates the Choquet Integral, a powerful operator adept at considering criteria significance and interconnectedness in optimization scenarios. Unlike its predecessor, the *ci*-AVOA treats optimization problems in their true complexity by recognizing and accounting for the relationships between variables. The performance of *ci*-AVOA is evaluated on ten CEC2020 benchmark functions and four engineering design problems, pitted against other renowned optimization algorithms and the original AVOA. Across low and high dimensional benchmark functions, *ci*-AVOA consistently outperforms its counterparts, underpinning its superiority. This superior performance is further validated using non-parametric statistical tests, solidifying *ci*-AVOA as an effective and robust tool for tackling complex optimization problems. In essence, this study provides a significant contribution by augmenting a well-known metaheuristic with the Choquet Integral to devise a superior algorithm, *ci*-AVOA. This innovation extends the problem-solving capabilities of metaheuristics, promising more accurate and robust solutions for complex, real-world optimization problems.

Abbreviations

μ	Fuzzy mesure
$ci - AVOA$	Choquet Integral African Vultures Optimization Algorithm
$AOAVOA$	Improved Hybrid Acquila Optimizer and African Vultures Optimization Algorithm
AOS	Atomic Orbital Search
AVG	Average results
$AVOA$	African Vultures Optimization Algorithm
$Best$	Best Results
$BestVulture_1$	First best vulture
$BestVulture_2$	Second best vulture

*CEC′*2020	CEC'2020 Competition Benchmark
Cos	Function of cosine
F	Starvation rate
GA	Genetic algorithm
GWO	Grey Wolf Optimizer
h	A random number between [− 2,2]
IAVOA	Improved African Vultures Optimization Algorithm
iter	Number of iterations
*L*1	Probability parameter to select the first best vulture
L2	Probability parameter to select the second best vulture
lb	The lower bound of search spaces
max_{iter}	Maximum number of iterations
Mean	Average results
N	Number of vultures
P_1	A random number between [0,1]
P_2	A random number between [0,1]
P_3	A random number between [0,1]
P_i	Vulture position vector
PSO	Particle Swarm Optimization
R_i	One best vulture selected
$rand_1$	A random number between [0,1]
$rand_2$	A random number between [0,1]
$rand_3$	A random number between [0,1]
$rand_4$	A random number between [0,1]
$rand_5$	A random number between [0,1]
$rand_6$	A random number between [0,1]
$rand_{p1}$	A random number between [0,1]
$rand_{p2}$	A random number between [0,1]
$rand_{p3}$	A random number between [0,1]
Sin	Function of sine
SSA	Salp Swarm Optimization
STD	Standard Deviation
TAVOA	Enhanced African Vultures Optimization Algorithm with tent map and time varying mechanism
ub	The Upper bound of search spaces
V_i	Fitness value of best vultures
w	A parameter that determines the probability of entering the exploration and exploitation phases
z	A random number between [− 1,1]

1 Introduction

Optimization problems can be defined as the process of choosing the optimal solution from a set of possible ones. These problems are of great interest in many applications where it is necessary to maximize profit, production, and efficiency in many scientific fields, including engineering issues, operations research, economics, etc. (Du and Pardalos 1998).

Deterministic techniques and stochastic methods can be used to categorize existing optimization problems (Pardalos et al. 2000).

Deterministic methods (or gradient based methods) such as Conjugate Gradient and the Newton–Raphson method (Costilla-Enriquez et al. 2020), require knowledge of the gradient of the fitness function to direct the search effectively. These algorithms guarantee finding the optimal solutions but are only helpful in solving small dimension problems. Indeed when faced with more complex problems gradient-based methods suffer from three serious limitations: a slow rate of convergence, a strong dependence on the starting point and finally the risk of falling into an optimal local solution. Solving large-scale problems using exact algorithms requires a lot of time and resources, which is practically impossible with deterministic approaches.

This has been the main reason for the development of metaheuristic algorithms that take inspiration from observations of natural phenomena (nature, swarms, and physical processes). Indeed metaheuristics can address NP hard problems (multi-modal, non-convex, discontinuous and non-differentiable) including those with many local optima. The main advantages of metaheuristics are an acceptable time of convergence, low consumption and the absence of reliance on gradient information (Greco et al. 2021).

Consequently, metaheuristics algorithms are essential for various engineering applications, including engineering design (Carbas et al. 2021), feature selection (Manita and Korbaa 2020; Nssibi et al. 2021), electrical engineering (Razmjooy et al. 2021), parameter extraction, image segmentation, and path planning (Ramadas and Abraham 2019; Ouertani et al. 2022). Metaheuristic algorithms fall into a number of areas, including those based on evolution (Opara and Arabas 2019), trajectory (Dekkers and Aarts 1991), swarms (Mafarja et al. 2020), bio-inspiration (Faris et al. 2018), physics/chemistry (Abualigah et al. 2022), humans (Askari et al. 2020), and plants (Yang 2012).

Metaheuristic algorithms work in a particular way using two main steps: exploration and exploitation. The exploration phase tries to find a solution in the unknown spaces of the problem. The exploitation phase searches more precisely in the area found in the exploration phase in order to decrease the algorithm's level of randomness and to improve the level of accuracy (Khajehzadeh et al. 2011). When the exploration phase of the algorithm is prioritized, it converges more rapidly enabling the algorithm to search the solution space more randomly. Hence it yields solution sets with greater diversity of elements. In the exploitation phase, the objective is to find solutions in a more localized area in order to increase the quality and accuracy of the solution sets. However there are some difficulties. Exploitation capacity will suffer if the exploration capacity is increased, and vice versa. These effects are reciprocal. So it appears difficult to find an acceptable balance between these two stages.

In this paper we are interested in the African Vultures Optimization Algorithm (Abdollahzadeh et al. 2021) i.e., a swarm-based metaheuristic that mimics the foraging and navigation behaviors of African vultures in nature. While AVOA is known for its ability to find global optima by combining exploration and exploitation, it often encounters difficulties in the latter phase. When solutions are closely clustered or when a problem demands a delicate balance between multiple objectives, this may lead to less-than-optimal solutions.

This study introduces an advanced metaheuristic global optimization algorithm, *ci*-AVOA, aiming to improve upon the existing African Vultures Optimization Algorithm. The newly proposed *ci*-AVOA addresses the shortcomings of AVOA by leveraging the Choquet Integral operator (Choquet 1954). This method is specifically designed to consider the significance and interdependence of optimization criteria, offering a more comprehensive approach to solve complex optimization problems where criteria are interconnected.

The integration of the Choquet Integral enables the *ci*-AVOA to deliver more robust performance when dealing with multi-criteria optimization problems, particularly those where the criteria are not entirely independent and to effectively model real-world scenarios, significantly enhancing its ability to deliver more accurate and robust solutions.

The structure of the remainder of this study is organized as follows: Sect. 2 outlines related works concerning the combination of metaheuristics with uncertainty approaches. Section 3 offers a concise description of the foundational African Vulture Optimization Algorithm (AVOA), the Choquet integral, and the newly proposed *ci*-AVOA algorithm. Section 4 presents the results from two separate experiments, the CEC2020 test suite benchmark functions, and provides a comprehensive analysis of the performance of various algorithms. Section 5 illustrates the practical application of the *ci*-AVOA algorithm to solve engineering problems. In Sect. 6, the strengths and limitations of the proposed approach are discussed, alongside potential future research directions. Finally, Sect. 7 summarizes the main findings and conclusions of this study.

2 Related work

As mentionned in the introduction, metaheuristic algorithms can identify optimal/near-optimal solutions by mimicking natural behaviours but may suffer from some limitations. Although in some cases they can be very efficient and converge rapidly to the optimal solution, in other cases there might be a very long execution time or a non-satisfactory solution might be obtained (Chopard et al. 2018).

A solution to overcome these drawbacks is to improve the exploration or exploitation strategy. In the literature, many recent papers address this issue. For example the Moth flame optimization (MFO) algorithm based on the principle of navigation technique of moth toward moon suffers from an inability to make a good trade-off between global and local search. Hence, a modified dynamic opposite learning (DOL) strategy is proposed in Sahoo et al. (2023) and an exploration operator, namely Weibull flight distribution in addition with a Fibonacci search process-based technique is presented in Sahoo et al. (2023). The Whale optimization algorithm (WOA) based on the hunting behavior of humpback whales has a considerable convergence speed but suffers from diversity in the solution due to the low exploration of search space. In Chakraborty et al. (2022) the random solution selection process is altered to increase exploration and the whale's cooperative hunting strategy is incorporated to balance the exploration and exploitation phase of WOA. A variant of Butterfly Optimization Algorithm employs a self-adaptive parameter setting and a local search strategy embedded with Levy flight search to make a better trade-off between exploration and exploitation (Sharma et al. 2022). In the same way, a modified backtracking search algorithm (BSA) framework based on the quasi reflection-based initialization, quantum Gaussian mutations, adaptive parameter execution, and quasi-reflection-based jumping limitation is presented in Nama et al. (2022).

Another solution to increase robustness will be the hybridization of metaheuristics with algorithmic components originating from other techniques. Numerous examples of hybridization can be found in literature. A hybrid sine cosine butterfly optimization algorithm (m-SCBOA), in which a modified butterfly optimization algorithm is combined with sine cosine algorithm to achieve superior exploratory and exploitative search capabilities is proposed in Sharma et al. (2022). Based on the Slime Mould algorithm, four areas of hybridization, progress, changes, and optimization are discussed in Gharehchopogh et al. (2023).

In order to resolve the travelling Salesman Problem considered as NP-hard, a Farmland Fertility Algorithm and a hyperheuristic technique based on the Modified Choice Function (MCF) has been presented in Gharehchopogh et al. (2022) to discover the shortest Hamilton route that visits each city precisely once and then returns to the starting point. A hybrid approach based on a binary version of the Farmland Fertility Algorithm achieves feature selection in the classification of Intrusion Detection Systems (IDSs). Results tested on two valid IDSs datasets, namely NSL- KDD and UNSW-NB15 shows that the method performs better than the classic machine Learning classifiers (Naseri and Gharehchopogh 2022).

Different usages of the Quantum Computing concept, i.e., a new field of research that includes elements from mathematics, physics, and computing in metaheuristics has been reviewed in Gharehchopogh (2023). For community detection in complex networks, the Cuckoo Search Optimization algorithm disadvantaged by problems of premature convergence, is improved using a Genetic Algorithm in order to increase performances in the exploration and the exploitation phases (Shishavan and Gharehchopogh 2022). Concerning the sparrow search algorithm, a literature review on variants, improvement, hybridization, and optimization is detailed in Gharehchopogh et al. (2023). In the context of social behaviour simulation, the particle swarm optimization (PSO) algorithm that has a high risk to fall into a poor local optimum, is hybridized with the backtracking search optimization algorithm to resolve the original PSO algorithm's problems (Zaman and Gharehchopogh 2022). Experimental results outperforms several state-of-the-art PSO variants on almost all of the benchmark problems (Zaman and Gharehchopogh 2022). A hybrid method based on the fruit fly algorithm and the ant lion optimizer algorithm has acceptable performance on simulation results on the dataset KDD Cup99, NSL- KDD, and UNSW-NB15 in the context of the intrusion detection system (Samadi Bonab et al. 2020). In Mohammadzadeh and Gharehchopogh (2021) the authors introduce an approach to solve the problem of lower convergence rate and lower population diversity in metaheuristics using a multi-agent systems (MASs) where several metaheuristic algorithms are considered as separate competitive and cooperating agents. The hybrid model tested on 32 complex benchmark functions and shows effectiveness and powerfulness for solving high-dimensional optimization problems. Finally the Salp Swarm Optimization (SSO) and the African Vulture Optimization Algorithm (AVOA) have been hybridized and evaluated on 52 benchmarks with excellent performance in solving global optimization problems (Gharehchopogh 2022).

A complementary approach to increasing the robustness of metaheuristics is to consider uncertainty, due to imprecise observation of natural phenomena, in the modelling process (2020). In fact, ignoring uncertainty is one strategy, but such an approach could produce a sub-optimal solution. Various models can be chosen in function of the nature of the uncertainty such as probabilistic modelling (Bianchi et al. 2002; Wang and Zhao 2022) or fuzzy sets (Alkan and Kahraman 2020). For example, a combination of the Fuzzy c-means clustering algorithm and Chemical Reaction Optimization metaheuristic is used in Nayak et al. (2017) to obtain optimal cluster centers. In a similar vein, Zhang and Ma (2020) a hybridation of fuzzy clustering and PSO metaheuristic is used to balance exploration and exploitation in Zhang and Ma (2020) or to improve the accuracy of Intrusion Detection Systems (Mishra and Naik 2019). In Arriola et al. (2022) the authors carried out a study that aimed to analyze the progress of current research related to energy and sustainability and utilized the fuzzy optimization approach of 96 retrieved publications. The article shows that fuzzy optimization enables significant results on technological, environmental, and economical factors. Still in the context of energy consumption, a hybrid metaheuristic (particle filter and particle swarm optimization) (Pozna et al. 2022) or a gentic algorithm (Wang et al.

2023) enables to the optimal tuning of fuzzy-controllers. In order to design an optimal taxing strategy of carbon emissions, fuzzy random matrix generators have been used in tandem with the cuckoo search technique in Ganesan and Elamvazuthi (2022). In the domain of multi-attribute group decision-making, an algorithm based on complex intuitionistic fuzzy values and using the particle swarm optimization algorithm is presented in Rani and Garg (2022).

Moreover, different fuzzy measures and fuzzy integrals have been successfully implemented to solve and optimize a variety of problems. The non-additive Choquet Integral is particularly interesting as it can simulate interactions between criteria, such as redundancy or synergy, in addition to the relative relevance of each criterion (Choquet 1954). The literature concerning hybridizing metaheuristic algorithms and Choquet Integral is sparse and recent. For instance, in the study of Boissiere et al. (2007), authors used Choquet Integral as the aggregated objective function to guide the Tabu Search metaheuristic. The proposed approach was applied to a multi-criteria public transport network design problem. The results obtained on a classical benchmark network are quite attractive compared to those from over-tested techniques. The Choquet integral was used as an aggregation operator to allow for trade-offs between the criteria such as travel time, cost, and CO_2 with the Tabu Search metaheuristic in a system called DyCOS for optimizing dynamic carpooling (Cheikh-Graiet et al. 2020). In a study by Khalouli et al. (2008), a multi-criteria hybrid flow shop scheduling problem was addressed by combining a multi-criteria Ant Colony Optimization (ACO) algorithm with the Choquet Integral method. The proposed method produced multiple schedules based on different criteria and performed well compared to other constructive heuristics in terms of quality and computation time. Similarly, in Branke et al. (2016), the Choquet Integral was used as a preferred model in interactive evolutionary multi-objective optimization. The authors introduced an interactive multi-objective evolutionary algorithm to determine the most preferred part of the Pareto-optimal set and compared it with other methods on benchmark problems in various dimensions.

The *ci*-AVOA algorithm presented in this article is a continuation of previous work considering fuzzy measures such as the Choquet integral to consider the significance and interdependence of optimization criteria. This point is the main motivation of the proposed study, where the use of the non-additive Choquet integral along with a metaheuristic optimization algorithm is introduced.

3 Proposed approach *ci*-AVOA

In this section, a description of African Vultures Optimization Algorithm (AVOA), and Choquet Integral method as well as the proposed approach combining AVOA and Choquet Integral are provided.

3.1 Conventional AVOA

The African Vultures Optimization Algorithm (AVOA), as described in Abdollahzadeh et al. (2021), imitates the behavior of African vultures, including crawling, foraging, and competing for food. It utilizes two exploration strategies to generate diverse solutions and four exploitation strategies to improve potential solutions. Additionally, transition strategies are implemented to balance the exploration and exploitation phases.

According to the authors, this new metaheuristic proved efficient in solving large-scale optimization problems.

3.1.1 Identifying the best vulture in a group

To determine the best vulture in any group, the process begins by creating an initial population, and then calculating the fitness values of all the solutions. The best vulture of the first group is determined by selecting the solution with the highest fitness value, while the best vulture of the second group is determined by selecting the solution with the second highest fitness value. The rest of remaining vultures move towards the best solutions using Eqs. 1 and 2. The fitness values of all vultures are recalculated within each iteration.

$$R_i = \begin{cases} BestVulture_1, if\ P_i = L_1 \\ BestVulture_2, if\ P_i = L_2 \end{cases}, \tag{1}$$

$$P_i = \frac{V_i}{\sum_{i=1}^{n} V_i} \tag{2}$$

where R_i represents the position vector of one of the two best vultures selected in the current iteration, $BestVulture_1$ refers to the position vector of the best vulture in the first group in the current iteration, and $BestVulture_2$ refers to the position vector of the best vulture in the second group in the current iteration. The L_1 and L_2 parameters take values between 0 and 1, with their sum being 1. P_i is obtained according to the roulette wheel strategy as shown in Eq. 2. V_i represents the fitness value of the first and second group of vultures, and n represents the total number of first and second group of vultures.

3.1.2 Starvation rate of vultures

When vultures are full, they have high energy and travel longer distances in search of food. When they're hungry, they don't have enough energy to fly long and forage alongside the stronger vulture. Eqs. 3 and 4 are used for mathematical modeling of this behavior.

$$F = \left(2 \times rand_1 + 1\right) \times z \times \left(1 - \frac{iter_i}{maxiter}\right) + t \tag{3}$$

$$t = h \times \left(\sin^w\left(\frac{\pi}{2} \times \frac{iter_i}{maxiter}\right) + \cos\left(\frac{\pi}{2} \times \frac{iter_i}{maxiter}\right) - 1\right) \tag{4}$$

where *iter* represents the valid number of iterations and *maxiter* the maximum number of iterations. The parameters $rand_1$, z and h take random values in the ranges $[0, 1], [-1, 1]$and$[-2.2]$, respectively. The parameter w is a fixed number, and increasing the value of w increases the probability of entering the exploration phase late in the optimization process. If the value of $|F|$ is greater than or equal to 1, the algorithm enters the

exploration phase and the vultures search for food at random distances from one of the best solutions.

3.1.3 Exploration

In order to benefit from different exploration strategies, the $rand_{P1}$ and P_1 parameters, which take random values in the [0, 1] range, are compared. If P_1 is greater than or equal to $rand_{P1}$, Eqs. 5 and 6 are used.

$$P(i+1) = R(i) - D(i) \times F \tag{5}$$

$$D(i) = |2 \times rand \times R(i) - P(i)| \tag{6}$$

If P_1 is less than $rand_{P1}$, Eq. 7 is used. $P(i+1)$ represents the position vector of the vulture in the next iteration, F represents the saturation rate of the vulture in the current iteration.

$$P(i+1) = R(i) - F + rand_2 \times ((ub - lb) \times rand_3 + lb) \tag{7}$$

If the value of the variable $|F|$ is less than 1 then, the exploitation stage is passed and the vultures search for food in the neighborhood of one of the best solutions. The lower and upper limits for the variable values are ub and lb. The parameters $rand$, $rand_2$ and $rand_3$ take random values between 0 and 1.

3.1.4 Exploitation

In order to benefit from different exploitation strategies, the parameters P_2, P_3, $rand_{p2}$, $rand_{p3}$ which take random values in the range of [0,1], are compared.

- Exploitation stage 1: If the value of the variable $|F|$ is between 0.5 and 1 then, and if P_2 is equal to or greater than $rand_{p2}$, Eqs. 8 and 9 are used.

$$P(i+1) = D(i) \times (F + rand_4) - d(t) \tag{8}$$

$$d(t) = R(i) - P(i) \tag{9}$$

  If P_2 is less than $rand_{p2}$, Eqs. 10 and 11 are used. Thus, vultures swirling flights and sieges over the food source are modeled. Parameters $rand_4$, $rand_5$ and $rand_6$ take random values between 0 and 1.

$$P(i+1) = R(i) - (S1 + S2) \tag{10}$$

$$\begin{aligned} S1 &= R(i) \times \left(\frac{rand_5 \times P(i)}{2\pi} \right) \times \cos(P(i)) \\ S2 &= R(i) \times \left(\frac{rand_6 \times P(i)}{2\pi} \right) \times \sin(P(i)) \end{aligned} \tag{11}$$

- Exploitation stage 2: $|F|$ if its value is less than 0.5 and P_3 is equal to or greater than $rand_{p3}$, Eqs. 12 and 13 are used.

$$P(i+1) = \frac{A1 + A2}{2} \tag{12}$$

$$A1 = BestVulture_1(i) - \frac{Bestvulture_1(i) \times P(i)}{Bestvulture_1(i) \times P(i)^2} \times F$$
$$A2 = BestVulture_2(i) - \frac{Bestvulture_2(i) \times P(i)}{Bestvulture_2(i) \times P(i)^2} \times F \quad (13)$$

If P_3 is less than $rand_{p3}$, Eqs. 14 and 15 are used. Thus, the accumulation and aggressive bickering of vultures around the food source is modeled.

$$P(i+1) = R(i) - |d(t)| \times F \times Levy(d) \quad (14)$$

Levy(d) denotes the Lévy flight function used to increase the performance of the AVOA. The mathematical expression of Lévy flight is as follows:

$$Levy(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\lambda}}}, \sigma = \left(\frac{\Lambda(1+\lambda) \times \sin\left(\frac{\pi\lambda}{2}\right)}{\Lambda\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \right)^{\frac{1}{\lambda}} \quad (15)$$

λ is a constant number, d represents the dimensions of the problem, the parameters u and v take random values in the range of 0 to 1.

The flowchart of AVOA metaheuristic is illustrated in Fig. 1.

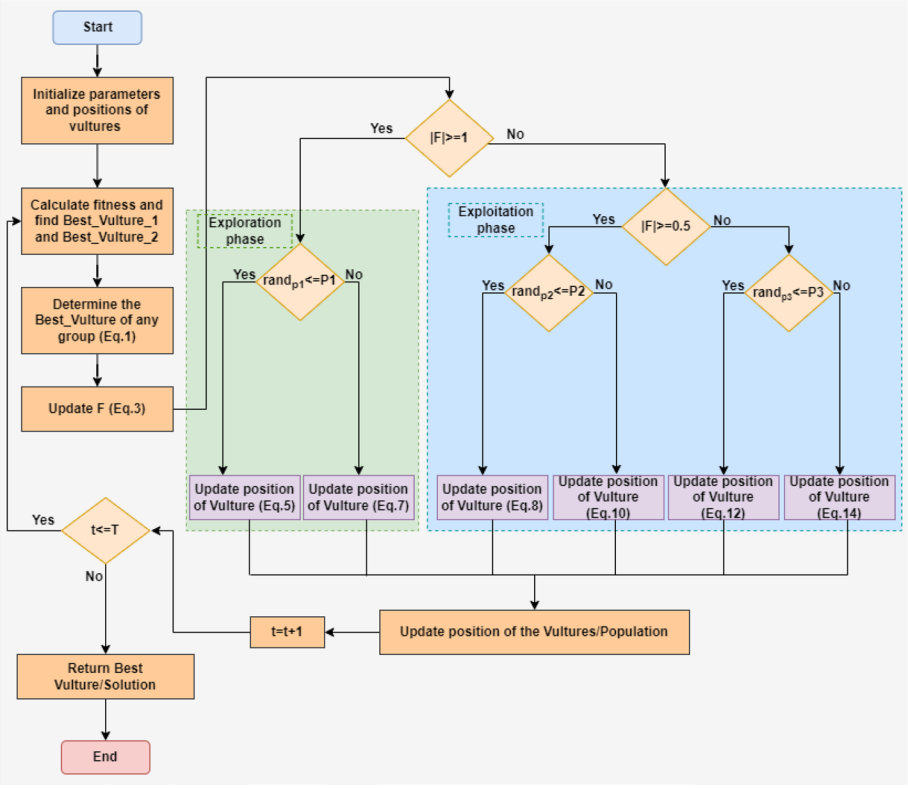


Fig. 1 Flowchart of AVOA

3.2 Choquet integral

The additivity axiom of classical measure theory is effective in describing various types of measurements in perfect, error-free situations, but it doesn't fully capture measurements in real-life scenarios. Since the behavior of vultures in the wild is unpredictable, measurements in this context are inherently non-additive. To account for this, fuzzy measures use less stringent axioms of monotonicity with respect to set inclusion, rather than the additivity axiom used in classical measures.

Definition 1 (Fuzzy measure) A fuzzy measure or non-additive measure μ on a set $X = (x_1, x_2, \ldots, x_n)$ oh n criteria is a set function $\mu : 2^X \to [0, 1]$ satisfying the following three properties:

(1) $\mu(\emptyset) = 0$
(2) $\mu(X) = 1$ (normality)
(3) $\forall B, C \subset X$, if $B \subseteq C$ then $\mu(B) \leq \mu(C)$ (monotonicity)

Fuzzy measures are utilized to indicate the level of significance of subsets of criteria, and their usefulness in decision-making is due to their capacity to handle the interrelationships between criteria (Grabisch 1997). Since the component-measures of a fuzzy measure do not allow for the direct calculation of the measure of the union of two distinct subsets, Sugeno (1974) proposed the λ-fuzzy measure which has supplementary characteristics.

Definition 2 (λ-measure) For $\lambda > -1$, the fuzzy measure μ_λ is a λ-fuzzy measure if it satisfies:

$$\mu_\lambda(A \cup B) = \mu_\lambda(A) + \mu_\lambda(B) + \lambda\mu_\lambda(A)\mu_\lambda(B) \tag{16}$$

for every pairs and disjoint subsets A and B of X

The value of the parameter λ can be found from the equation $\mu_\lambda(X) = 1$ which is equivalent to solving:

$$\lambda + 1 = \prod_{i=1}^{n}(1 + \lambda\mu_\lambda(x_i)). \tag{17}$$

Note that a normalized fuzzy measure has $2^n - 2$ parameters whereas a λ-measures is one class of fuzzy measures with $n - 1$ parameters.

The λ-measure is a parametric fuzzy measure that is usually used with a fuzzy integral such as the Choquet Integral defined such that:

Definition 3 (Choquet Integral) The Choquet Integral of $G = (g_1, g_2, ..., g_n) \in \mathbb{R}_+^n$ according to the λ-fuzzy measure μ_λ is defined by:

$$C_\mu(G) = \sum_{i=1}^{n} \left(G_{\tau(i)} - G_{\tau(i-1)}\right)\mu_\lambda(\{\tau(i), \ldots, \tau(n)\}) \tag{18}$$

where τ is a permutation on X such that $G_{\tau(1)} \leq G_{\tau(2)} \leq \ldots \leq G_{\tau(n-1)} \leq G_{\tau(n)}$ and $G_{\tau(0)=0}$.

In this regard, the Choquet Integral computes the score of each attribute with respect to a λ-fuzzy measure μ_λ.

3.3 *ci*-AVOA

The problem we raise in this study is how to exploit the best solutions already found. Therefore, we propose a dynamic aggregation function that takes into account the objective functions of each solution and aggregates the fitness measures of the existing solutions into a global measure based on the λ-fuzzy measure and Choquet integral.

Referring to the AVOA algorithm, the transition between exploration and exploitation depends on the hunger rate of vultures F. Therefore, four hunting strategies are used to achieve various position updating of vultures. This allows the algorithm to effectively exploit the solution information in the search space to reach the global optimum. Nonetheless, in the four hunting strategies, the aggregation approach models a static aggregation method that does not consider the uncertainty in communication between vultures. In fact, the movement of all vultures toward the food source relies on a simple arithmetic combination as modelled by Eqs. 5 and 7 in the exploration phase and Eqs. 10 and 12 in the exploitation phase.

Given the above analysis, we incorporate the Choquet fuzzy integral in the exploitation and exploration phases of AVOA to pattern an effective communication between vultures/solutions.

In order to utilize the Choquet fuzzy integral, it is necessary to determine the fuzzy measure values of each solution's fitness. These values represent the strength of individual solutions and their combinations. Once all the fuzzy measure values and corresponding fitness values are identified, the Choquet fuzzy integral method can be applied to compute the overall solution.

Let $Z()$ denote the fuzzy measure of a set of solutions $S = \{S_1, S_2, S_3, \ldots, S_n\}$ and $V = \{v_1, v_2, v_3, \ldots, v_n\}$ represent the fitness values of individual solutions in S. For any subset $C_i = \{S_1, S_2, S_3, \ldots, S_i\}$ of S where $1 \le i \le n$, the aggregated value can be computed using the Choquet fuzzy integral with the help of the following equation, assuming that $v_1 \ge v_2 \ge v_3 \ge \cdots \ge v_n$.

$$\begin{aligned}\text{Choquet}_Z(v_1, v_2, v_3, \ldots, v_n) = v_n Z(C_n) + (v_{n-1} - v_n) Z(C_{n-1}) + \cdots \\ + (v_1 - v_2) Z(C_1)\end{aligned} \tag{19}$$

To compute the fuzzy membership values of solution combinations, the first step is to calculate the value of λ referring to Eq. 17 and based on the following equation:

$$1 + \lambda = \prod_{i=1}^{n} \left(Z(\{S_i\}) \lambda + 1 \right) \tag{20}$$

After finding the roots of the characteristic equation, the value of λ can be determined. With this value, the fuzzy membership value of any solution combination can be calculated using the following equation repeatedly.

$$Z(\{S_p, S_q\}) = Z(\{S_p\}) + Z(\{S_q\}) + \lambda Z(\{S_p\}) Z(\{S_q\}) \tag{21}$$

where $1 \le p, q \le n$.

We develop seven variants of AVOA with Choquet fuzzy integral in order to cover all phases and determine the best aggregation.

1. The first variant modifies the exploitation phase first stage (*ci*-AVOA-1). Hence, the Eq. (10) could be written as follows:

$$P(i+1) = \text{Choquet}_{Z}\ (S1, S2). \tag{22}$$

2. The second variant modifies the exploitation phase second stage (*ci*-AVOA-2). Then, the Eq. (12) is reformulated as follows:

$$P(i+1) = \text{Choquet}_{Z}\ (A1, A2). \tag{23}$$

3. The third we incorporate the modification on the two exploitation stages at the same time (*ci*-AVOA-3)
4. The fourth variant we modify the exploration phase (*ci*-AVOA-4). Therefore, Eqs. (5) and (7) are mathematically reformulated as follows:

$$P(i+1) = \text{Choquet}_{Z}\ (B1, B2), \tag{24}$$

where B1 and B2 are calculated as follows:

$$\begin{aligned} B1 &= R(i) - D(i) \times F \\ B2 &= R(i) - F + rand_2 \times ((ub - lb) \times rand_3 + lb) \end{aligned} \tag{25}$$

5. The fifth we assemble the modification of the exploitation phase first stage with the exploration phase (*ci*-AVOA-5).
6. The sixth variant models the exploitation phase second stage with the exploration phase including Choquet Integral (*ci*-AVOA-6)
7. The seventh combines the third variant with the fourth variant (*ci*-AVOA-7).

The study's principal motivation rests upon advancing the existing African Vultures Optimization Algorithm (AVOA), which, while effective in various optimization scenarios, exhibits certain limitations that this work strives to address.

One notable limitation of the AVOA is its inherent challenge in efficiently handling the optimization problems where criteria are interconnected. The AVOA, like many traditional optimization algorithms, is prone to treating each criterion in an optimization problem as isolated, thereby possibly overlooking their interdependencies. Such overlook can lead to suboptimal solutions, particularly in complex multi-objective optimization problems where the criteria's interplay significantly impacts the quality of the solutions.

To overcome these limitations, the proposed *ci*-AVOA leverages the Choquet Integral operator. The Choquet Integral is a potent mathematical tool, specifically designed to aggregate criteria, considering both their importance and mutual dependencies. It provides a sophisticated means of handling multiple interconnected criteria, thereby enhancing the problem-solving capacity of the optimization algorithm.

The integration of the Choquet Integral within the AVOA framework allows *ci*-AVOA to manage the interconnected criteria more efficiently, significantly boosting its ability to tackle complex multi-criteria optimization problems. This integration is anticipated to provide more accurate and robust solutions where the criteria are not entirely independent.

Therefore, the primary justification for the integration of the Choquet Integral is its ability to account for the interdependencies among criteria, a feature that significantly enhances the capability of the algorithm to deal with real-world scenarios. As a result, the *ci*-AVOA algorithm offers a more comprehensive approach to complex optimization problems, making it a valuable tool in the field of global optimization.

4 Experimental results and discussion

In this section, we present a performance and accuracy study of the proposed approach, *ci*-AVOA. This study involves a benchmark of 10 functions from the CEC2020 test suite. The first series of experiments is conducted on the seven developed variants of AVOA. The second series involves comparing the best version of the proposed approach, *ci*-AVOA, with state-of-the-art metaheuristics including: Genetic Algorithm (GA) (Mirjalili 2019), Particle Swarm Optimization (PSO) (Venter and Sobieszczanski-Sobieski 2003), Grey Wolf Optimizer (GWO) (Mirjalili et al. 2014), Salp Swarm Algorithm (SSA) (Mirjalili et al. 2017), Atomic Orbital Search (AOS) (Azizi 2021), African Vultures Optimization (AVOA) (Abdollahzadeh et al. 2021), Improved African vulture optimization algorithm (IAVOA) (Liu et al. 2022), Improved hybrid Aquila Optimizer and African Vultures Optimization Algorithm (AOAVOA) (Xiao et al. 2022), and enhanced African Vultures Optimization Algorithm with tent map and time varying mechanism (TAVOA) (Fan et al. 2021).

4.1 Test functions

The proposed algorithm *ci*-AVOA has been assessed through a comprehensive performance evaluation using 10 benchmark functions from the Congress On Evolutionary Computation 2020 (CEC2020). These benchmark functions serve as challenging objective functions that provide a robust test for optimization algorithms. They encompass a variety of

Table 1 CEC2020 test functions

	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Function	1	Shifted and Rotated Bent Cigar Function (CEC2017 F1)	100
Basic Functions	2	Shifted and Rotated Schwefel's Function (CEC2014 F11)	1100
	3	Shifted and Rotated Lunacek bi-Rastrigin Function (CEC2017 F7)	700
	4	Expanded Rosenbrock's plus Griewangk's Function (CEC2017 F19)	1900
Hybrid Functions	5	Hybrid Function 1 ($N = 3$) (CEC2014 F17)	1700
	6	Hybrid Function 2 ($N = 4$) (CEC2017 F16)	1600
	7	Hybrid Function 3 ($N = 5$) (CEC2014 F21)	2100
Composition Functions	8	Composition Function 1 ($N = 3$) (CEC2017 F22)	2200
	9	Composition Function 2 ($N = 4$) (CEC2017 F24)	2400
	10	Composition Function 3 ($N = 5$) (CEC2017 F25)	2500

Search range: $[-100, 100]^D$

types: unimodal, basic, hybrid, and composition functions, which are denoted as $f_1 - f_{10}$. Each function has a known global optimum symbolized as $Fi*$.

The evaluation of these benchmark functions occurs within a defined search space of $[-100, 100]^D$, where D symbolizes the dimensions, which are either 10 or 20 in this study. To effectively solve these benchmark functions, a strategic balance between exploration (searching the entire problem space) and exploitation (refining the solutions already found) is mandatory. Further details concerning the benchmark functions are provided in Table 1.

As an initial step, we compare the performance of the seven proposed variants of *ci*-AVOA with the original AVOA. This comparative analysis, presented in Table 4, employs a range of statistical fitness metrics including the best solution found (Best), the average of all solutions found (Mean), and the standard deviation (Std). The best solution reflects the optimal output the algorithm was able to achieve in its search, essentially providing a direct measure of the algorithm's efficacy in pinpointing the most optimal solution to a given problem. The mean value, on the other hand, gives a general overview of the algorithm's performance, taking into account all the solutions it finds during its search, thus serving as a measure of the algorithm's consistent performance. The standard deviation, meanwhile, is an index of the variation in the solutions that the algorithm generates. A lower standard deviation indicates the algorithm's capability to generate solutions that are consistently close to the mean, thereby testifying to the algorithm's stability and reliable performance across different runs. These metrics are mathematically defined as follows:

$$\text{Best Fitness (Best)} = \max_{i=1}^{\text{Max}_{iter}} BS_i$$

$$\text{Average Fitness (Mean)} = \frac{1}{\text{Max}_{iter}} \sum_{i=1}^{Max_{iter}} BS_i$$

$$\text{Standard Deviation (STD)} = \sqrt{\frac{\sum_{i=1}^{M} \left(BS_i - \mu\right)^2}{Max_{iter}}},$$

where BS is the best score obtained for each iteration.

The second phase of experiments features a thorough review of qualitative and quantitative measures, including the use of the Wilcoxon rank sum test. This test is a non-parametric statistical method employed to ascertain whether the outcomes of the proposed *ci*-AVOA method differ significantly from those of other comparative algorithms (Neuhäuser 2011). In essence, the Wilcoxon rank-sum test assigns ranks to all observed values as if they were members of a single group, subsequently aggregating the ranks of each separate group. Under the null hypothesis, the assumption is made that both samples originate from the same population, and any observed discrepancies between the two rank sums arise purely from sampling error. The p value produced by this statistical test serves as an indicator of the degree of significance. Specifically, a p value of less than 0.05 signifies that there exists a significant difference at the 5% level.

In addition to the Wilcoxon rank test, the study also incorporates the Friedman rank test, the Friedman aligned rank test and the Quade test (Derrac et al. 2011). These methods are specifically designed to calculate the mean rank of three or more matched groups, providing a broader statistical perspective for assessing the performance of several algorithms. Combined, these rank-based tests provide a robust and comprehensive statistical framework for performance comparison.

Table 2 Six scenarios conducted to evaluate the sensitivity of the conventional AVOA to its parameters

Parameters	p1	p2	p3	Alpha	Gamma	Beta
Scenario 1	[0.1,1] Step = 0.1	0.4	0.6	0.8	0.2	2.5
Scenario 2	0.6	[0.1,1] Step = 0.1	0.6	0.8	0.2	2.5
Scenario 3	0.6	0.4	[0.1,1] Step = 0.1	0.8	0.2	2.5
Scenario 4	0.6	0.4	0.6	[0.1,1] Step = 0.1	0.2	2.5
Scenario 5	0.6	0.4	0.6	0.8	[0.1,1] Step = 0.1	2.5
Scenario 6	0.6	0.4	0.6	0.8	0.2	[0.5,5] Step = 0.5

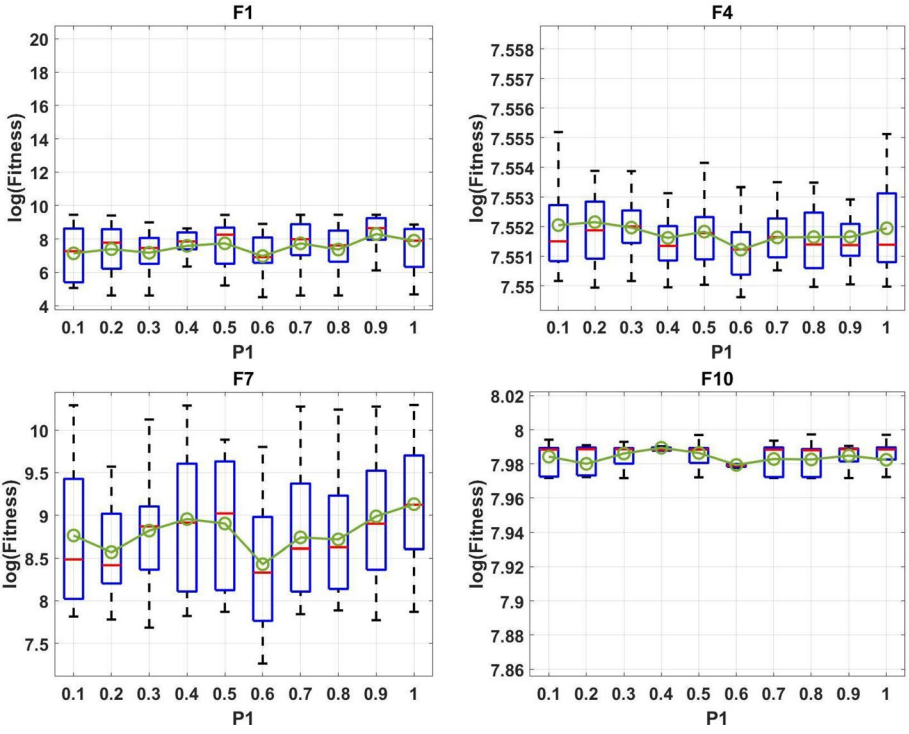


Fig. 2 Tuning of parameter p1 according to the first scenario (Sen1)

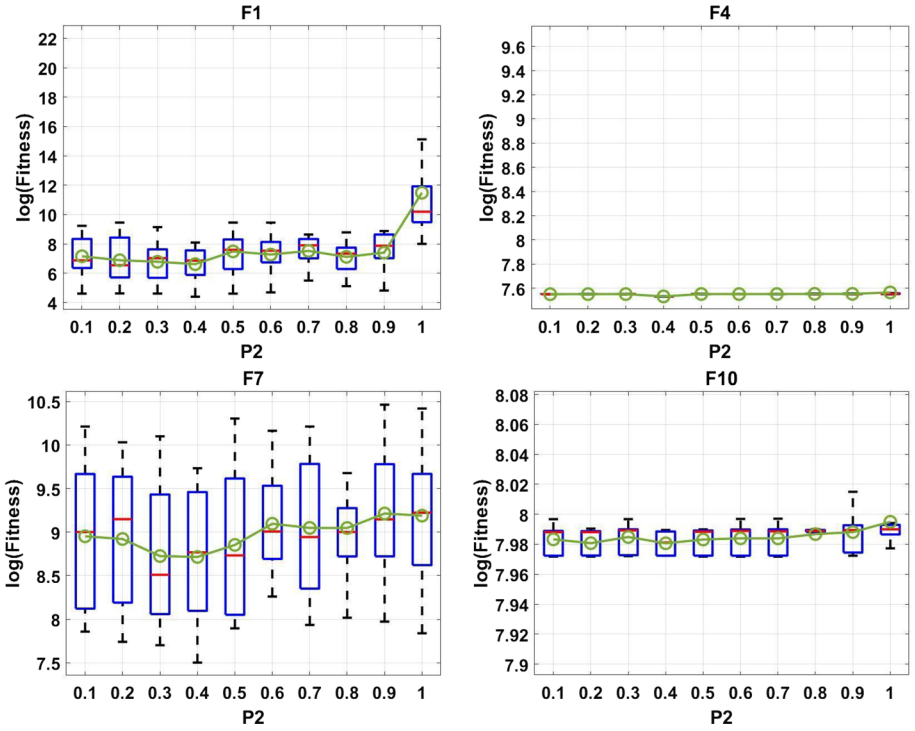


Fig. 3 Tuning of parameter p2 according to the second scenario (Sen2)

An essential part of our experimental evaluation involves the use of convergence graphs. These plots provide a clear visualization of how the performance of the *ci*-AVOA method and its competitors evolves over the course of the optimization process. They offer insight into the speed and stability of the algorithms as they approach the optimal solution.

Additionally, history and trajectory charts provide a comprehensive understanding of the dynamic behavior of the search agents in the algorithm. The history charts demonstrate the areas of the search space visited by the vultures, while the trajectory charts capture the changes in the positions of the vultures during the optimization process. These graphical representations are crucial for assessing the exploration and exploitation behaviors of the search agents, and in turn, the effectiveness of the optimization strategy.

Furthermore, boxplots and radar plots are employed to articulate the comparative performance of the algorithms. These graphical tools deliver an intuitive and comprehensible view of the distribution of the algorithm performances, as well as their convergence behaviors.

4.2 Parameter settings

The tests were conducted using MATLAB 2020a on a Windows 10 operating system with an i7-1.80 GHz processor and 16 GB of RAM. The comparison between the algorithms was performed over 30 independent runs, where the population size was fixed at 30 for

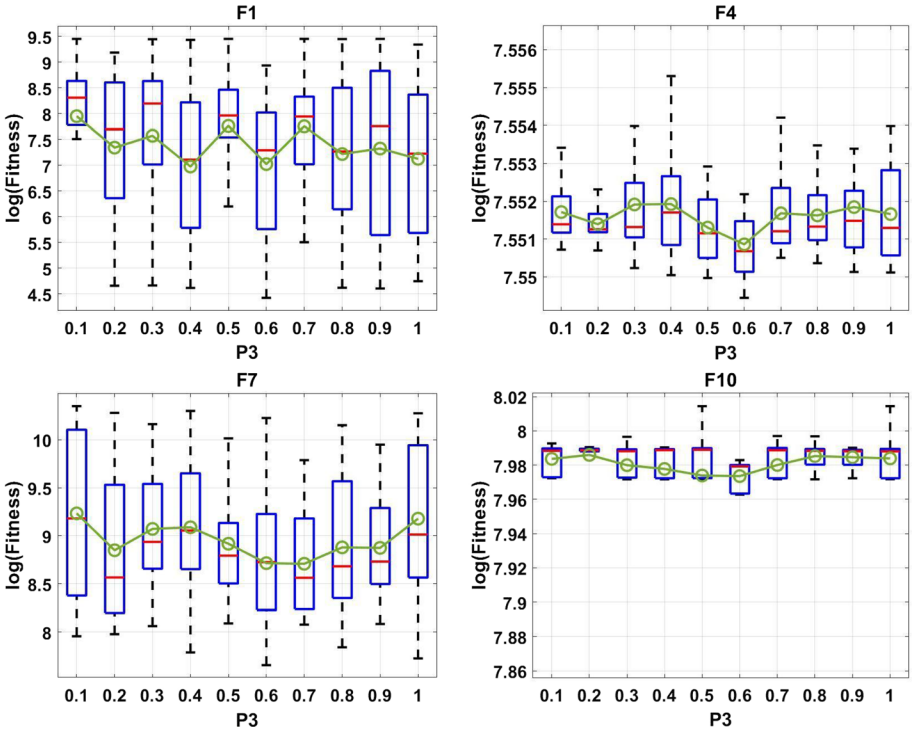


Fig. 4 Tuning of parameter p3 according to the third scenario (Sen3)

problems with dimensions 10 and 20, and the maximum number of iterations was set to 1000.

Tuning parameters plays a vital role in the realm of metaheuristic algorithms, as it exerts a significant influence on their performance and effectiveness in tackling optimization problems. The act of adjusting these parameters can yield substantial benefits, such as enhancing convergence rates, expediting the convergence process, and ultimately bolstering the overall performance of the algorithms. As a result, we have decided to put the conventional AVOA through a series of six distinct scenarios, as summarized in Table 2. By isolating and testing each parameter individually, we aim to gain a clearer understanding of its impact on the algorithm's performance and behavior. This comprehensive approach will enable us to fine-tune the AVOA and optimize its performance for various optimization tasks.

Based on the analysis of Figs. 2, 3, 4, 5, 6, 7, we have reached the following conclusions regarding the best parameters for the AVOA algorithm:

- The optimal value for the parameter α is 0.8,
- The optimal value for the parameter γ is 2.5,
- The optimal value for the parameter β is 0.2,
- The best value for the parameter $p1$ is 0.6,
- The best value for the parameter $p2$ is 0.4,
- and the best value for the parameter $p3$ is 0.6.

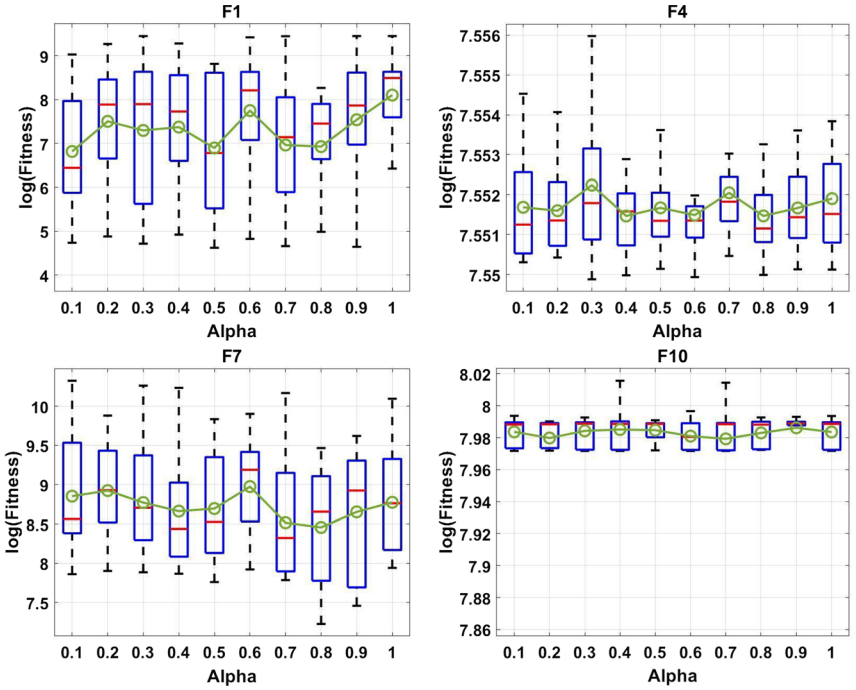


Fig. 5 Tuning of parameter α according to the fourth scenario (Sen4)

These findings provide valuable guidance for effectively configuring the AVOA algorithm to achieve superior performance in solving diverse optimization problems.

After this tuning part, Table 3 summarizes the parameters of all algorithms used in this experimental study. For the AVOA algorithm and its variants, its parameters were tuned as described earlier in this paper. However, for the other algorithms considered in the study, their parameters were adopted directly from their original publications. By using the parameters reported in the literature, we ensure a fair and consistent comparison with the previously established results.

4.3 Results for 10 and 20-dimensional problems for the 7 variants of *ci*-AVOA

Tests were conducted on the seven proposed variants of the AVOA algorithm, using the CEC2020 benchmark test function with dimensions of 10 and 20, and 30 independent runs. These results are summarized in Tables 4 and 5. We evaluated the proposed approach's performance using various significant statistical measures, including the best, mean, and standard deviation (STD) of the fitness values. In these tables, the best results have been emphasized in bold for clarity. The statistical outcomes indicate that the sixth version of the AVOA (*ci*-AVOA-6), in which we incorporated the Choquet fuzzy integral into the second stage of the exploitation phase and the exploration phase of AVOA, yielded the most promising results for 9 out of the 10 functions assessed.

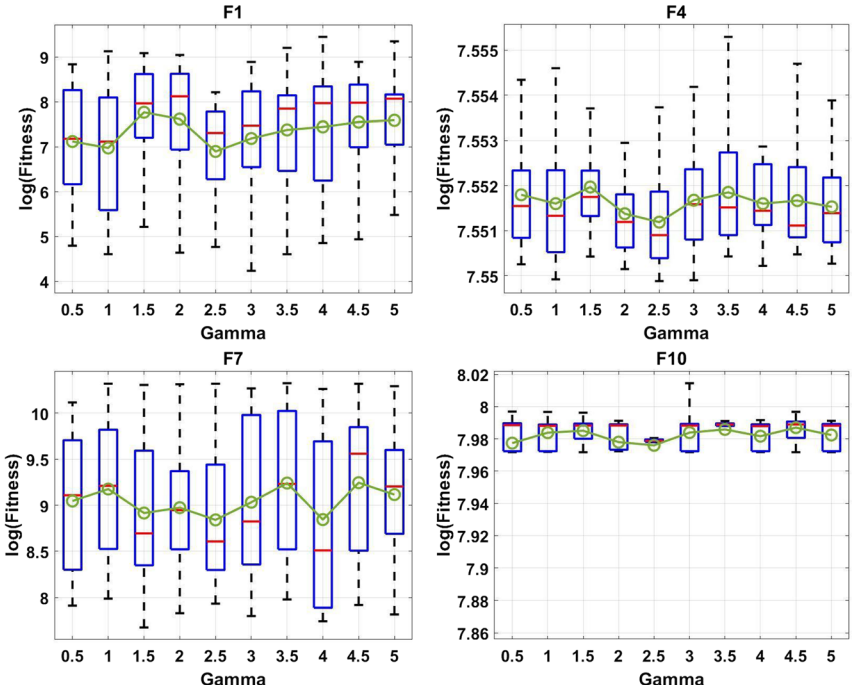


Fig. 6 Tuning of parameter γ according to the fifth scenario (Sen5)

Table 3 Algorithms parameter settings

Algorithm	Parameter	Value
AOS	α (depth weight)	50
	β (multiplier weight)	0.2
	Vmax (maximum velocity)	6
GWO	a	2
GA	Crossover rate	0.9
	Mutation rate	0.1
PSO	c1 (Cognitive factor)	2
	c2 (Social factor)	2
	Vmax (Maximum velocity)	6
	Wmax (Maximum bound on inertia weight)	0.9
	Wmin (Minimum bound on inertia weight)	0.4
SSA	c1 (depth weight)	50
	c2 (multiplier weight)	Random
	c3	Random
AVOA	L_1	0.8
	L_2	0.2
	k	2.5
	P_1	0.6
	P_2	0.4
	P_3	0.6
All of them	Search agents (particles, atoms, vultures...)	30
	Maximum iterations	1000

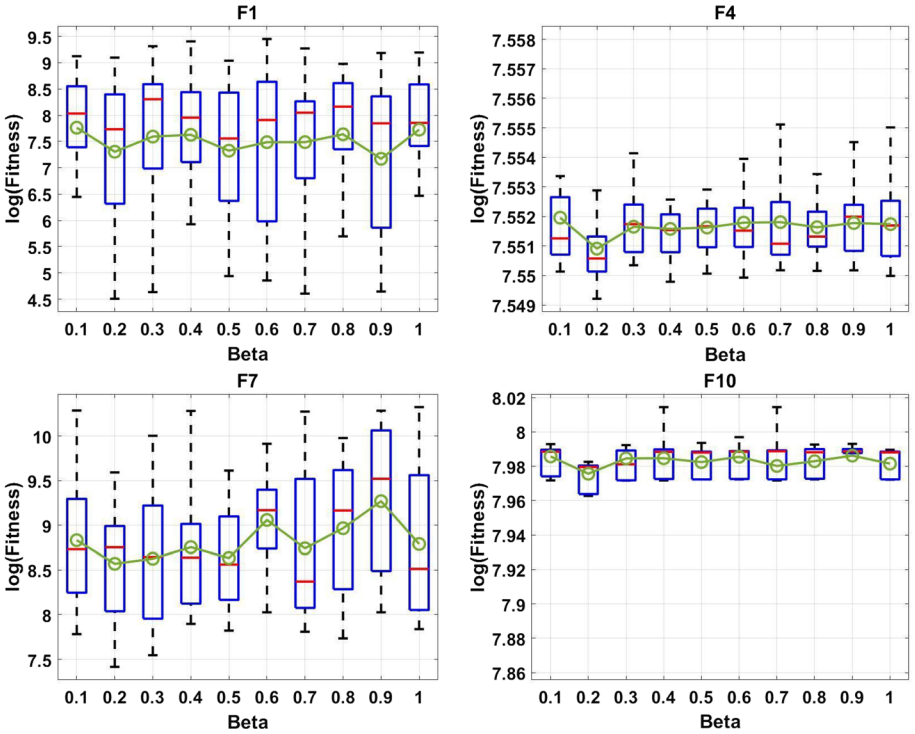


Fig. 7 Tuning of parameter β according to the sixth scenario (Sen6)

Although these initial results are promising, further investigations are necessary to conclusively establish the effectiveness of this version. Accordingly, we have visualized the behaviors of the search agents of AVOA in Figs. 8, 9, 12, and those of *ci*-AVOA-6 in Figs. 10, 11, 13. The search agents' behaviors were evaluated based on four criteria: search history, convergence behavior, the average fitness of the population, and the trajectory of the first vulture. We have analyzed these behaviors in 2D for the ten CEC2020 test suite functions in 20-dimensional space.

The search history chart displays the positions visited by the vultures, while the convergence behavior graph illustrates the changes in the performance of the leading vulture over the course of the optimization process. The population fitness average graph reflects the variation in the population's average fitness throughout the optimization, and the trajectory chart of the first vulture depicts how the vulture's position changes over time.

Upon examining the search history outlined in the second column of Figs. 10, 11, 13, it is apparent that the *ci*-AVOA-6 algorithm tends to prioritize exploration over exploitation, a significant deviation from the behavior of the traditional AVOA algorithm. Additionally, the trajectory chart of the first vulture provides insights into the vultures' search behavior. This chart, presented in the third column of Figs. 10, 11, 13, shows that the first vulture undergoes rapid changes in position early in the optimization process, followed by a gradual slowing of movements as the optimization proceeds.

The fourth column of Figs. 10, 11, 13 depicts the best vulture's average fitness level, which generally shows a rapid decline, indicating a shift from exploration to exploitation

as the optimization process unfolds. Moreover, the *ci*-AVOA-6 method, evidenced by the average fitness of the population, suggests that the optimization process is slowed by the vultures' fuzzy movements and by identifying increasingly accurate fitness value solutions. This behavior indicates that the *ci*-AVOA-6 algorithm focuses more on exploring promising areas during the optimization process. The evidence in these figures confirms the superior performance of the proposed *ci*-AVOA-6 version compared to the traditional AVOA.

Based on these results, we chose to use the *ci*-AVOA-6 variant for subsequent experiments. It demonstrated superior performance across all tested functions, indicating its ability to efficiently solve unimodal functions (*f1*), basic functions (*f2–f4*), hybrid functions (*f5–f7*), and composition functions (*f8–f10*). The pseudocode for *ci*-AVOA-6 is provided in Algorithm 1.

Algorithm 1 Pseudocode of *ci*-AVOA

Require: The population size N and maximum number of iterations T
Ensure: The location of Vulture and its fitness value
  Initialize the random population P_i $(i = 1, 2, ..., N)$
  while (stopping condition is not met) **do**
    Calculate the fitness values of Vulture
    Set PBestVulture1 as the location of Vulture (First best location Best Vulture Category 1)
    Set PBestVulture2 as the location of Vulture (Second best location Best Vulture Category 2)
    for each Vulture (P_i) **do**
      Select $R(i)$ using Eq. (1)
      Update the F using Eq. (3)
      if $(|F| \geq 1)$ **then**
        if $(P_1 \geq \text{rand}P_1)$ **then**
          Update the location Vulture using Eq. (5)
        else
          Update the location Vulture using Eq. (7)
        end if
      else
        if $(|F| \geq 0.5)$ **then**
          if $(P_2 \geq \text{rand}P_2)$ **then**
            Update the location Vulture using Eq. (23): $P(i+1) = \text{Choquet}_Z(A1, A2)$
          else
            Calculate $B1$ and $B2$ using Eq. (25)
            Update the location Vulture using Eq. (24): $P(i+1) = \text{Choquet}_Z(B1, B2)$
          end if
        else
          if $(P_3 \geq \text{rand}P_3)$ **then**
            Update the location Vulture using Eq. (12)
          else
            Update the location Vulture using Eq. (14)
          end if
        end if
      end if
    end for
  end while
  return PBestVulture1.

4.4 Results for 10-dimensional CEC2020 problems in comparison to other metaheuristics

The results obtained from the proposed *ci*-AVOA algorithm, as displayed in Table 6, are evaluated against other algorithms, using significant statistical measures like the best, mean, and standard deviation (STD) of fitness values.

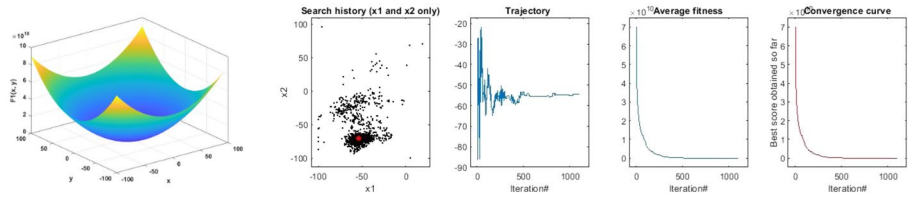
Table 4 Comparison of the fitness values over 30 experiments obtained by the proposed *ci*-AVOA variants and the original AVOA over CEC2020 test suite with Dimension 10

	AVOA	*ci*-AVOA-1	*ci*-AVOA-2	*ci*-AVOA-3	*ci*-AVOA-4	*ci*-AVOA-5	*ci*-AVOA-6	*ci*-AVOA-7
f-1								
Best	101.37	100.08	102.72	196.33	109.12	154.88	**100.00**	191.41
Mean	3339.64	4529.76	3836.07	3276.87	3568.45	3389.46	**100.01**	3835.20
Std	3687.91	3713.54	3209.76	2738.80	3755.94	2864.63	**0.02**	3140.41
f-2								
Best	1491.10	1351.95	1369.03	1403.06	1367.38	1416.17	1470.82	**1239.67**
Mean	1925.31	2045.02	1950.82	1924.26	1979.30	1891.44	1971.03	**1878.38**
Std	223.33	377.89	297.43	246.62	331.90	206.20	**178.93**	310.05
f-3								
Best	733.11	736.37	730.82	743.76	736.22	729.52	**714.26**	735.38
Mean	769.12	774.05	764.14	776.39	775.62	772.51	**725.51**	773.21
Std	17.55	20.01	19.14	16.75	21.41	17.16	**8.56**	18.41
f-4								
Best	1901.04	1901.36	1901.96	1901.21	1901.23	1900.84	**1900.27**	1901.00
Mean	1903.74	1903.89	1904.70	1903.86	1904.38	1904.67	**1901.16**	1904.14
Std	1.96	1.47	2.00	1.81	2.46	2.06	**0.52**	2.25
f-5								
Best	2250.68	2785.18	1868.56	2330.76	2852.70	2691.33	**1833.59**	2488.03
Mean	11580.09	12384.69	14687.08	18223.82	15797.90	18110.84	**2215.04**	16098.44
Std	12629.84	9298.45	14578.38	15700.37	12205.38	15240.04	**311.13**	22418.33
f-6								
Best	1601.42	1602.06	1615.00	1601.18	1601.70	1601.99	**1600.91**	1601.06
Mean	1754.23	1809.07	1811.61	1773.83	1790.03	1749.81	**1719.73**	1764.85
Std	99.23	110.10	91.67	113.41	104.51	**84.61**	96.75	99.26
f-7								
Best	2837.36	2773.64	2724.19	3247.13	2630.20	2620.27	**2100.64**	2373.20
Mean	10778.52	7859.74	9944.47	10181.82	10654.17	10431.76	**2257.35**	8485.34
Std	5917.40	7665.36	6962.98	7288.85	7709.92	8114.99	**138.71**	6322.37
f-8								
Best	2234.17	2251.17	2250.60	2230.99	2252.09	2251.33	**2223.03**	2253.84
Mean	2304.66	2307.19	2305.27	2303.16	2306.41	2304.17	**2295.99**	2305.11
Std	13.86	11.97	11.01	21.51	11.83	14.04	21.54	**10.36**
f-9								
Best	**2500.00**	**2500.00**	**2500.00**	**2500.00**	**2500.00**	**2500.00**	2733.02	**2500.00**
Mean	2754.52	2763.58	2742.35	2734.98	2758.80	**2734.69**	2747.64	2741.37
Std	79.75	73.96	111.78	108.84	74.09	109.20	**8.29**	112.18
f-10								
Best	2600.24	2897.93	2897.74	**2600.16**	2897.82	2600.25	2897.89	2897.94
Mean	2931.38	2936.99	2926.73	**2920.92**	2931.28	2925.29	2935.20	2928.81
Std	67.59	23.64	24.89	67.01	24.42	64.94	**22.25**	24.00

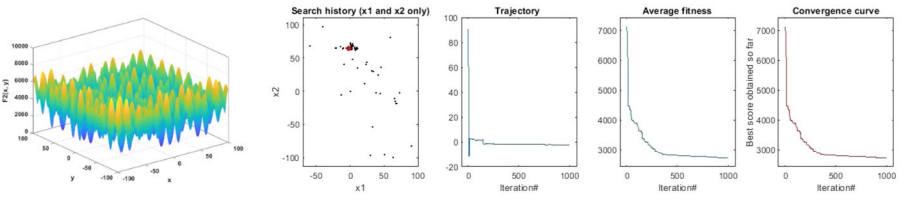
Table 5 Comparison of the fitness values over 30 experiments obtained by the proposed *ci*-AVOA variants and the original AVOA over CEC2020 test suite with Dimension 20

	AVOA	*ci*-AVOA-1	*ci*-AVOA-2	*ci*-AVOA-3	*ci*-AVOA-4	*ci*-AVOA-5	*ci*-AVOA-6	*ci*-AVOA-7
f-1								
Best	123.29	103.01	120.98	164.51	101.91	106.44	**100.00**	100.19
Mean	3729.91	2529.18	3079.70	4270.83	2642.43	3038.87	**100.00**	2798.42
Std	3857.38	3171.45	3186.58	4522.55	2679.89	3399.22	**0.00**	3799.22
f-2								
Best	1951.75	2051.87	1869.40	1812.73	1258.05	1838.46	**1106.89**	1810.82
Mean	2782.98	2975.02	2652.92	2782.26	2663.04	2566.39	**1910.04**	2691.82
Std	391.53	455.04	460.87	400.33	540.93	400.09	**258.04**	484.77
f-3								
Best	804.70	780.64	797.94	809.19	776.08	802.48	**716.33**	792.52
Mean	859.32	848.10	846.82	861.94	839.47	853.71	**723.57**	850.75
Std	34.94	33.76	39.70	29.94	37.16	29.94	**6.62**	33.00
f-4								
Best	1905.42	1902.89	1905.73	1903.77	1905.10	1904.75	**1900.48**	1904.12
Mean	1913.03	1911.70	1916.50	1914.52	1914.57	1913.73	**1901.22**	1914.29
Std	6.33	4.28	7.44	5.92	6.04	6.61	**0.39**	6.75
f-5								
Best	59270.18	39091.35	16802.46	33986.72	67100.26	8298.17	**1986.36**	20263.52
Mean	533411.37	500811.56	453015.20	391568.00	474994.52	387068.65	**2229.85**	328137.22
Std	439528.68	346301.15	369259.76	374417.93	446879.10	339240.86	**147.19**	290836.65
f-6								
Best	1636.59	1790.33	1628.29	1724.03	1622.68	1727.84	**1600.48**	1649.92
Mean	1951.88	2004.08	2009.68	1991.20	1904.55	2022.28	**1720.61**	1962.00
Std	197.14	162.73	233.84	186.94	198.51	180.29	**111.06**	162.75
f-7								
Best	4537.77	11151.97	38124.97	22898.15	31517.72	29189.27	**2101.28**	24853.42
Mean	306315.33	159713.14	266807.73	154806.34	230030.46	280306.72	**2352.70**	204113.36
Std	223578.92	162104.77	245438.69	100411.37	179384.81	279469.90	**183.62**	133272.96
f-8								
Best	2300.00	2300.00	2300.00	2300.00	2300.00	2300.00	**2239.27**	2300.00
Mean	2737.23	2887.04	2874.07	2999.94	2572.31	3141.61	**2299.09**	3088.98
Std	1136.68	1194.95	1324.56	1296.67	840.43	1435.05	**14.18**	1341.36
f-9								
Best	2865.15	2890.70	2888.51	2870.72	2550.00	2550.00	**2500.00**	2550.00
Mean	2968.17	2960.73	2998.37	2975.22	**2550.00**	**2550.00**	2724.12	**2550.00**
Std	57.01	49.87	69.60	58.81	**0.00**	**0.00**	77.13	**0.00**
f-10								
Best	2912.61	2914.11	2912.09	2913.94	**2700.00**	**2700.00**	2897.81	**2700.00**
Mean	2985.46	2975.47	2981.26	2968.31	**2700.00**	**2700.00**	2928.58	**2700.00**
Std	32.48	34.64	33.02	32.83	**0.00**	**0.00**	25.02	**0.00**

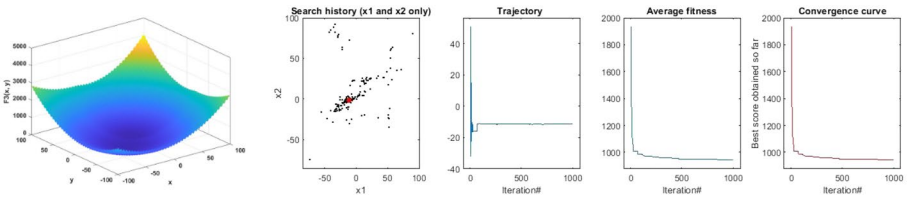
Notably, for the unimodal function $f1$, the proposed AVOA variant recorded the best values. It's worth highlighting that unimodal functions consist of a single global optimum and lack local optima, thus serving as an effective platform for analyzing the exploitation



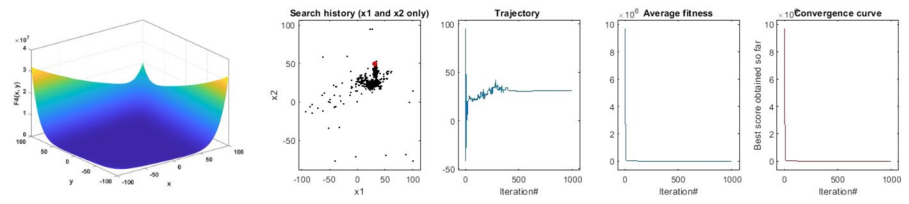
(a) F1



(b) F2



(c) F3

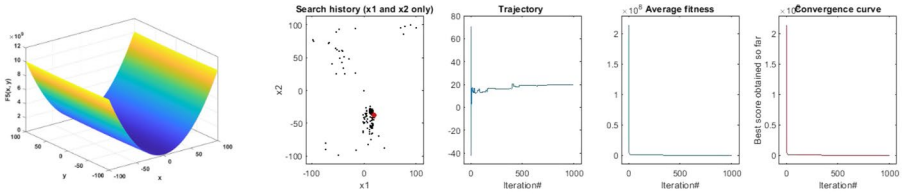


(d) F4

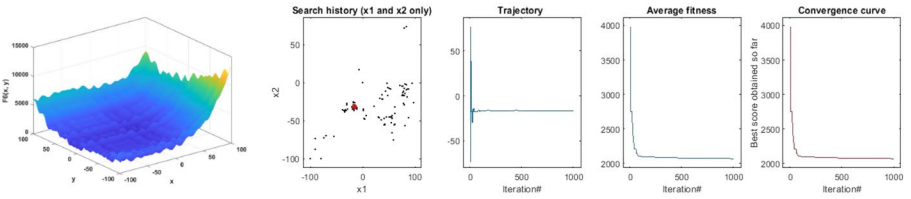
Fig. 8 Qualitative results of AVOA for 10-dimensional $f_1 - f_4$ CEC2020 test functions

capabilities of the various algorithms. The data suggests that our proposed approach notably enhances the exploitation of the search space.

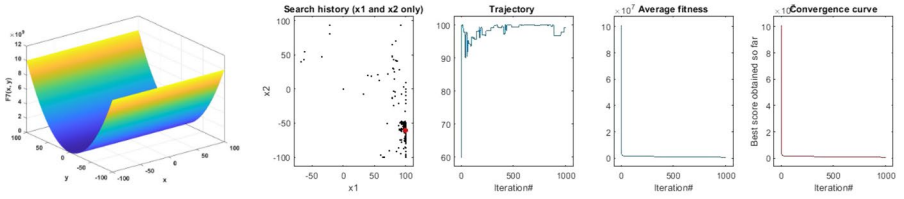
On the other hand, the basic functions $f2 - f4$ witnessed the superior performance of two algorithms, namely, PSO and GWO, while AOAVOA reported the smallest standard deviation. Contrasting with unimodal and basic functions, hybrid functions encompass numerous local optima, thereby presenting a challenge in evading these misleading optima. This aspect facilitates the exploration of an algorithm's exploratory abilities. In function *f*5, the *ci*-AVOA didn't attain the best performance, with the best fitness value achieved by IAVOA. For function *f*6, the PSO obtained the best fitness value, while the smallest mean was achieved by SSA, with the AOAVOA securing the smallest standard deviation. Function *f*7 saw *ci*-AVOA report the closest value to the best achieved by PSO and also secure the lowest mean. For function *f*8, GA achieved the best fitness value, but *ci*-AVOA obtained the smallest mean. The results indicate that *ci*-AVOA demonstrates superior



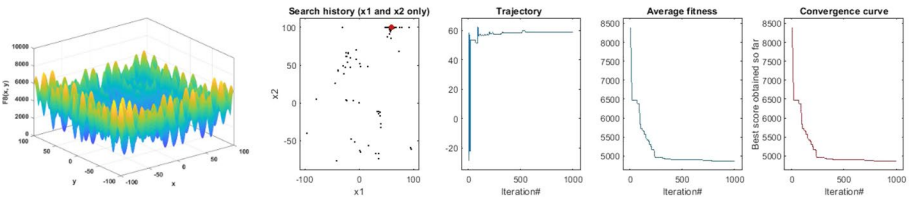
(a) F5



(b) F6



(c) F7



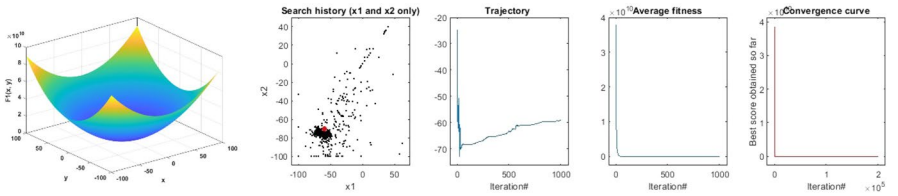
(d) F8

Fig. 9 Qualitative results of AVOA for 10-dimensional $f_5 - f_8$ CEC2020 test functions

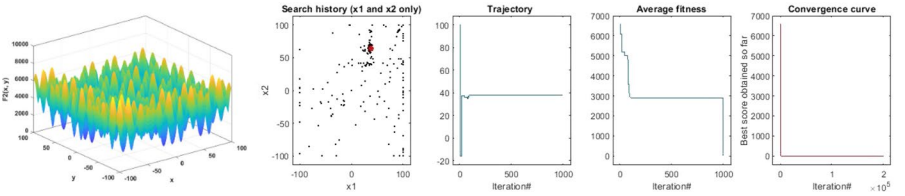
potential in navigating past the local optima. For the composition functions, $f9 - f10$, all comparative algorithms rendered competitive results.

Overall, the *ci*-AVOA algorithm proved to be highly effective in delivering satisfactory solution quality for low-dimensional problems.

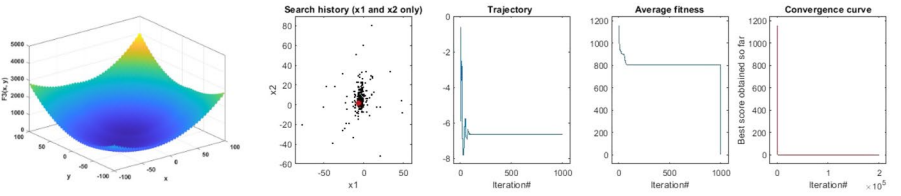
Another critical aspect in evaluating an optimization algorithm is its rate of convergence, that is, the speed of reaching the global optimum. In this context, the convergence curve can be instrumental in comparing different algorithms' performance. This curve, characterized by two key features, the ability to attain the global solution and the speed at which it does so, essentially plots the evolution of fitness values against the number of iterations. Such graphical representations serve as valuable tools for understanding the algorithm's rate of convergence and its stability, both of which are fundamental indicators of an optimization algorithm's performance. Examining Figs. 14, 15, and 16, we can see that the *ci*-AVOA not only converges to an optimal solution but does so more quickly and



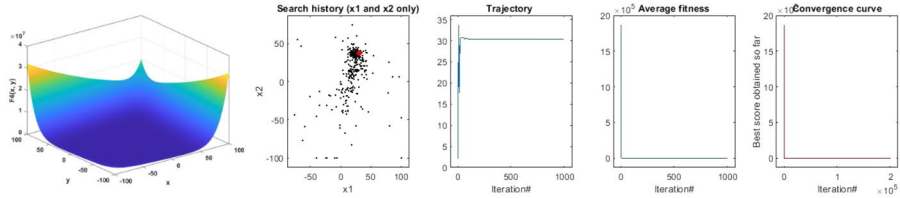
(a) F1



(b) F2



(c) F3



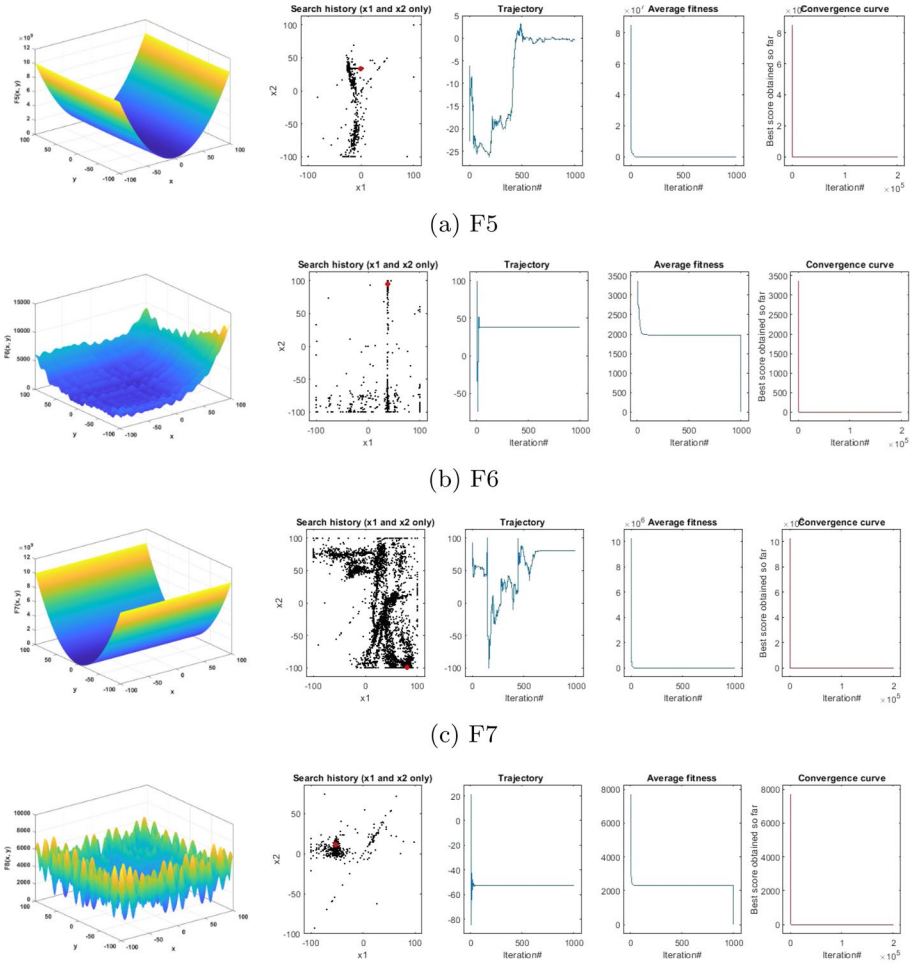
(d) F4

Fig. 10 Qualitative results of *ci*-AVOA-6 for 20-dimensional $f_1 - f_4$ CEC2020 test functions

consistently across all tested functions. The speed at which *ci*-AVOA arrives at the optimal solution and reaching a stable point in fewer iterations is a significant achievement as it indicates high efficiency. It means that the *ci*-AVOA algorithm effectively navigates the search space and zeros in on the global optimum without unnecessary exploration. This efficiency is a desirable trait in real-world applications, where computational resources and time may be limited.

The fact that *ci*-AVOA reaches a stable point for all functions indicates its robustness. The algorithm isn't easily trapped in local optima, and it can adapt and find the global optimal solution regardless of the specific function's characteristics.

Also, the use of a semi-log scale and the shift by a gap of 10^{-3} ensure that the nuances in convergence behavior aren't lost due to the presence of zero values. This adjustment allows us to discern differences in performance among the various algorithms more clearly.



(a) F5

(b) F6

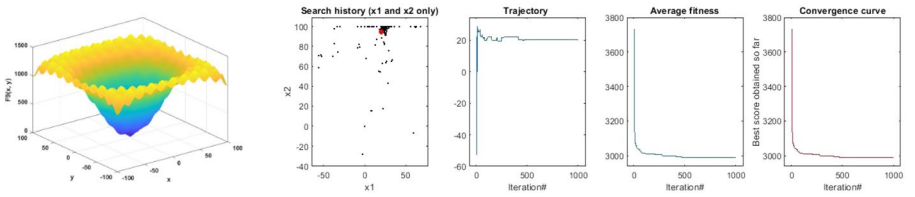
(c) F7

(d) F8

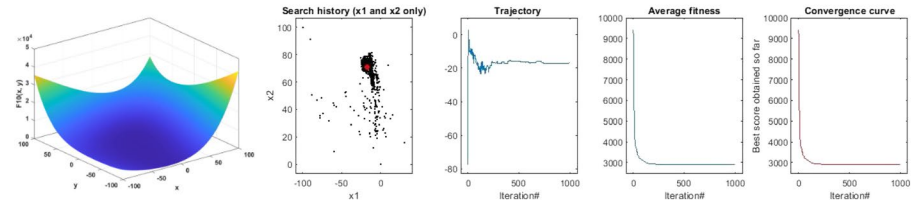
Fig. 11 Qualitative results of *ci*-AVOA-6 for 20-dimensional $f_5 - f_8$ CEC2020 test functions

In summary, the convergence graphs indicate that the proposed *ci*-AVOA algorithm combines efficiency, rapid convergence, and robustness. It's able to find high-quality solutions quickly and consistently across a range of functions, making it a highly competitive optimization algorithm.

The study also employed non-parametric tests, specifically the Wilcoxon and Friedman tests. The results of the Wilcoxon test, as presented in Table 7, indicate significant differences between the proposed *ci*-AVOA algorithm and the other algorithms in most functions, demonstrating its significant improvement over similar algorithms. The Table 7 presenting the results of the Wilcoxon rank-sum test, compares the performance of the proposed *ci*-AVOA algorithm with six other optimization algorithms (GA, PSO, GWO, SSA, AOS, AVOA, TAVOA, AOAVOA, and IAVOA) on the CEC2020 benchmark functions for dimension 10.

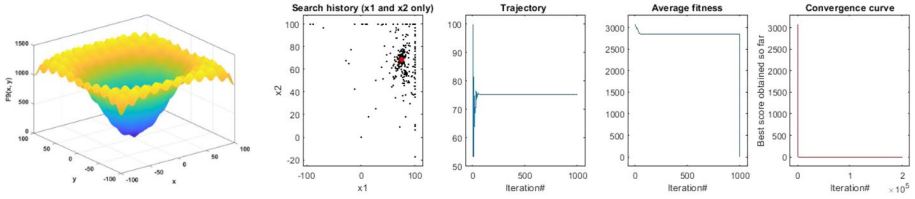


(a) F9

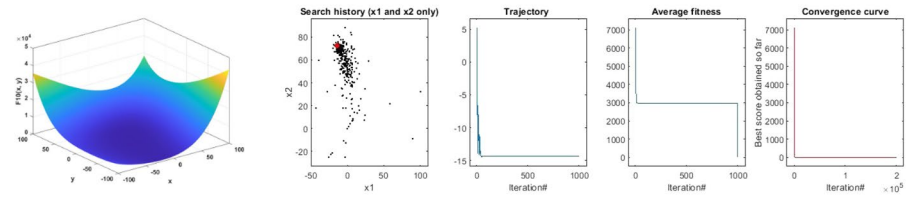


(b) F10

Fig. 12 Qualitative results of AVOA for 10-dimensional $f_9 - f_{10}$ CEC2020 test functions



(a) F9



(b) F10

Fig. 13 Qualitative results of *ci*-AVOA-6 for 20-dimensional $f_9 - f_{10}$ CEC2020 test functions

The test's p value is shown, along with an indication of whether *ci*-AVOA won (performed better), tied (performed equivalently), or lost (performed worse) on each function. The sum at the end of the table shows the total count of wins (+), ties (=), and losses (-) against each algorithm.

– Function F1: *ci*-AVOA performs significantly better than all other algorithms, with extremely low p-values (below 0.05, indicating statistical significance).

Table 6 A comparison of the fitness values over 30 experiments obtained by the proposed *ci*-AVOA and other competitor algorithms over CEC2020 test suite with $Dim = 10$

	GA	PSO	GWO	SSA	AOS	AVOA	TAVOA	AOAVOA	IAVOA	***ci*-AVOA**
f-1										
Best	331.5378	110.5245	6728348	1798.071	9244.348	101.375	391.994	4931.277	1370.372	**100.000**
Mean	2016.231	1442.529	83026425	7136.577	31883.48	3339.643	2396.298	29791.259	13032.307	**100.008**
Std	2060.154	1817.466	1.45E+08	4638.735	27945.88	3687.913	1660.130	24005.349	12956.782	**0.020**
f-2										
Best	1248.49	1225.783	**1190.37**	1235.857	1356.198	1491.098	1344.879	1273.222	1513.992	1470.825
Mean	1589.506	1693.594	**1566.478**	1682.9	1817.902	1925.313	1698.437	1639.792	1841.497	1971.031
Std	191.7413	251.7613	217.3531	230.0391	299.9854	223.333	166.171	**158.428**	164.956	178.930
f-3										
Best	715.5837	**705.6038**	727.8197	721.7992	725.6865	733.111	717.451	724.492	725.776	714.264
Mean	723.7203	**722.3006**	735.1568	728.0872	764.9572	769.121	724.873	732.572	748.167	725.514
Std	5.055313	6.612482	4.793494	7.930264	21.65182	17.547	**3.371**	3.974	11.494	8.561
f-4										
Best	1900.567	1900.387	1902.345	1900.471	1900.994	1901.038	1900.575	1901.872	1900.998	**1900.266**
Mean	1903.477	**1901.137**	1903.869	1901.191	1902.127	1903.736	1901.509	1902.533	1902.145	1901.156
Std	3.794376	**0.439147**	2.168611	0.443674	0.969459	1.960	0.930	0.781	0.772	0.525
f-5										
Best	15412.37	1980.981	4027.054	1970.606	2132.097	2250.676	2469.442	1918.162	**1746.693**	1833.591
Mean	637929.3	4508.969	6273.437	3966.499	5540.571	11580.093	32738.619	1982.831	**1798.250**	2215.036
Std	538654	2705.946	2585.33	1355.852	2663.602	12629.842	34202.555	57.336	**34.999**	311.126
f-6										
Best	1600.491	**1600.194**	1605.027	1601.998	1601.328	1601.422	1674.696	1607.346	1611.055	1600.914
Mean	1754.116	1781.162	1723.308	**1658.328**	1745.551	1754.226	1780.280	1696.143	1738.329	1719.728
Std	77.95607	72.76216	76.41773	64.76705	90.67132	99.226	56.013	**47.149**	53.342	96.754
f-7										
Best	2707.29	**2100.353**	2731.572	2237.149	2525.831	2837.359	2223.990	2541.392	2289.520	2100.639
Mean	232794.2	2308.818	7852.783	3006.741	5349.594	10778.520	121314.495	5205.759	3880.579	**2257.350**
Std	394192	147.3863	4630.574	498.5173	3299.848	5917.398	119446.619	1934.723	1552.004	**138.712**

Table 6 (continued)

	GA	PSO	GWO	SSA	AOS	AVOA	TAVOA	AOAVOA	IAVOA	*ci*-AVOA
f-8										
Best	**2217.888**	2300.883	2311.649	2219.18	2266.028	2234.167	2274.686	2276.196	2269.268	2223.027
Mean	2299.743	2388.488	2375.111	2300.091	2300.862	2304.663	2316.939	2328.211	2300.782	**2295.988**
Std	19.51042	272.9918	260.2297	19.11365	8.252148	13.862	85.306	110.008	**7.930**	21.541
f-9										
Best	2500.012	**2500**	2709.419	2500.121	2500.22	2500.000	2623.245	2743.370	2500.081	2733.019
Mean	2741.237	2725.978	2753.524	2730.994	**2634.049**	2754.520	2722.996	2748.712	2677.926	2747.638
Std	58.48164	77.77796	11.93798	54.68895	137.5095	79.746	53.773	**3.968**	75.268	8.295
f-10										
Best	2897.895	2897.743	2899.498	2897.76	2897.946	**2600.242**	2897.867	2901.961	2773.951	2897.888
Mean	2939.377	2919.311	2934.483	**2917.281**	2934.021	2931.378	2931.737	2927.483	2923.615	2935.199
Std	20.98848	23.58632	16.38775	23.52814	31.47473	67.594	**16.345**	18.232	33.673	22.250

Table 7 Wilcoxon rank sum test between the proposed *ci*-AVOA and other competitors algorithms on CEC2020 test suit functions with *Dim* = 10

PSO		GWO		SSA		AOS		AVOA		TAVOA		AOAVOA		IAVOA	
p	Win	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win
1.201E-05	+	6.796E-08	+	6.796E-08	+	6.796E-08	+	6.796E-08	+	0.42896	=	3.02E-11	+	4.80E-07	+
0.00597	−	4.540E-06	−	6.796E-08	−	0.10751	=	0.36484	=	0.00030	−	6.28E-06	−	0.38710	=
0.91242	=	0.00037	+	6.796E-08	+	5.227E-07	+	1.431E-07	+	4.50E-11	+	2.87E-10	+	3.52E-07	+
0.34421	=	6.796E-08	+	1.431E-07	+	0.00037	+	3.939E-07	+	2.60E-08	+	2.38E-07	+	2.03E-07	+
0.00055	+	6.796E-08	+	6.796E-08	+	7.948E-07	+	3.416E-07	+	0.23985	=	3.02E-11	−	3.02E-11	−
0.98245	=	0.57922	=	0.00010	+	0.25030	=	0.06787	=	0.05555	=	0.01327	−	0.97052	=
0.91242	=	6.796E-08	+	6.796E-08	+	6.796E-08	+	6.796E-08	+	0.00168	+	0.01076	+	1.09E-05	+
0.44136	=	6.796E-08	+	0.32346	=	0.28530	=	4.680E-05	+	0.00014	+	0.07245	=	0.00062	+
0.70841	−	0.01143	−	6.796E-08	−	0.56085	=	9.278E-05	−	0.00056	−	0.00015	+	0.00019	−
0.01006	−	0.94608	=	0.52499	=	0.73527	=	0.02564	−	0.07245	=	0.03644	−	0.03644	−
2+/5=/3−		**6+/2=/2−**		**6+/2=/2−**		**5+/5=/0−**		**6+/2=/2−**		**4+/4=/2−**		**5+/1=/4−**		**5+/2=/3−**	

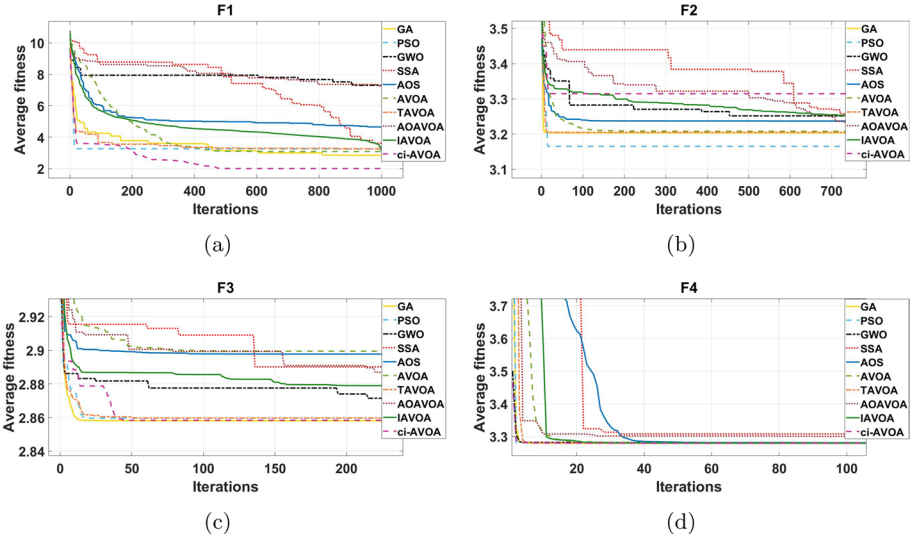


Fig. 14 Convergence curves of algorithms on the $f_1 - f_4$ 10-dimensional functions with respect to iterations

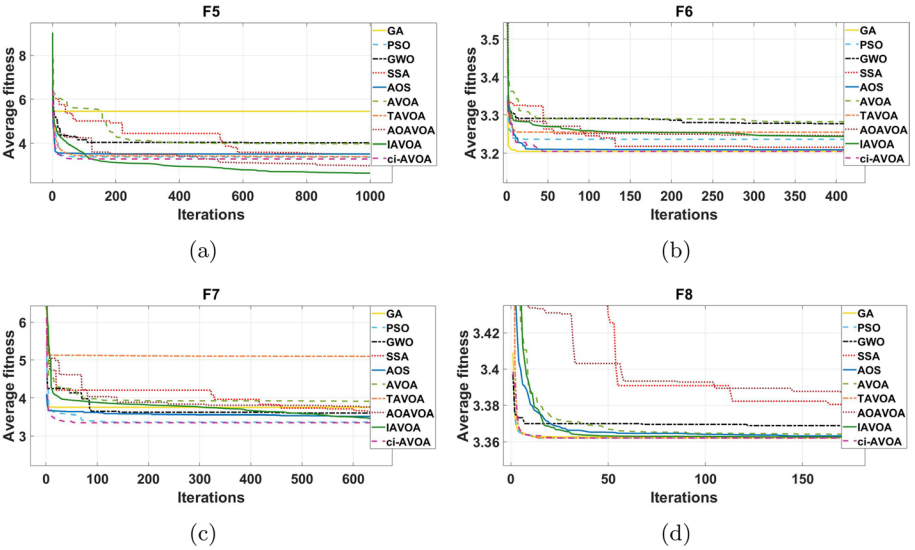
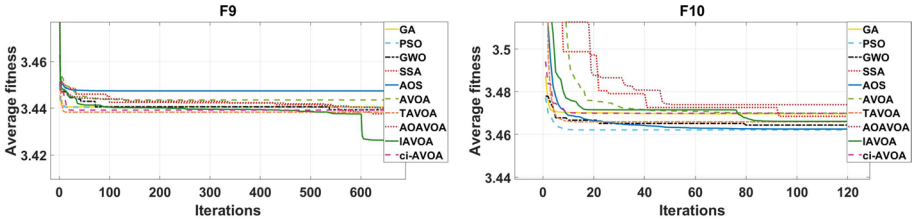


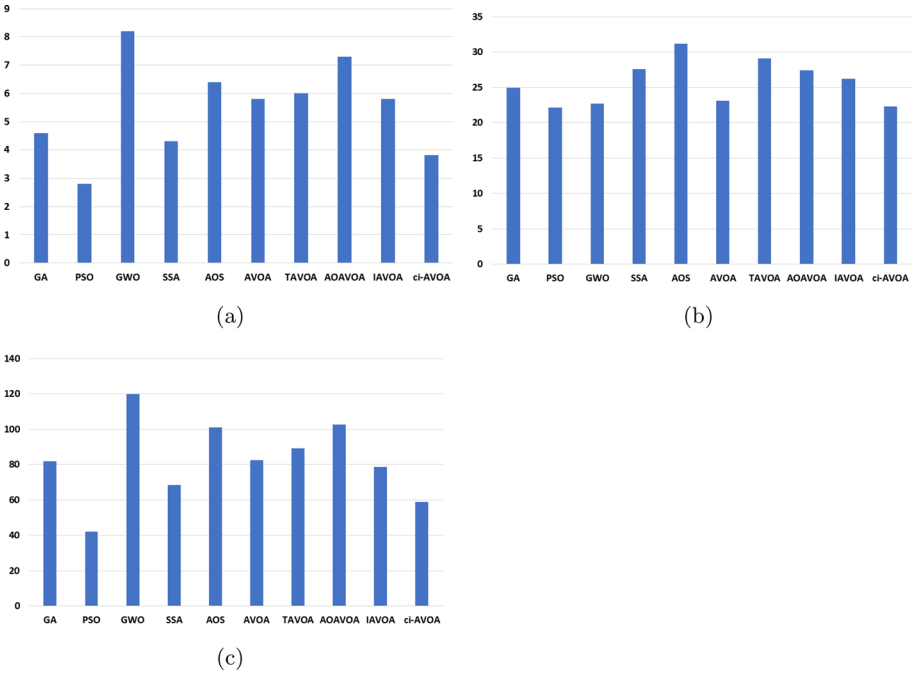
Fig. 15 Convergence curves of algorithms on the $f_5 - f_8$ 10-dimensional functions with respect to iterations

- Function F2: *ci*-AVOA performs significantly worse than GA, PSO, and GWO, as indicated by the negative signs and low p-values. It ties with AOS and AVOA, as the p-values are greater than 0.05, indicating no statistical difference in performance.
- Function F3: *ci*-AVOA performs significantly better than GWO, SSA, AOS, and AVOA, but it ties with GA and PSO, as indicated by the p-values greater than 0.05.
- Functions F4, F5, and F7: *ci*-AVOA significantly outperforms all the other algorithms.



(a) (b)

Fig. 16 Convergence curves of algorithms on the $f_9 - f_{10}$ 10-dimensional functions with respect to iterations



(a) (b)

(c)

Fig. 17 Average ranking of the algorithms by Friedman test (**a**), Friedman aligned test (**b**) and Quade test (**c**) for 10-dimensional problems

- Function F6: *ci*-AVOA ties with most of the algorithms, except SSA, where it performs better.
- Function F8: *ci*-AVOA outperforms GWO and AVOA but ties with the rest.
- Function F9: *ci*-AVOA ties with GA, AOS, performs worse than PSO, GWO, SSA, and AVOA.
- Function F10: *ci*-AVOA performs better than GA, worse than PSO and AVOA, but ties with the rest.

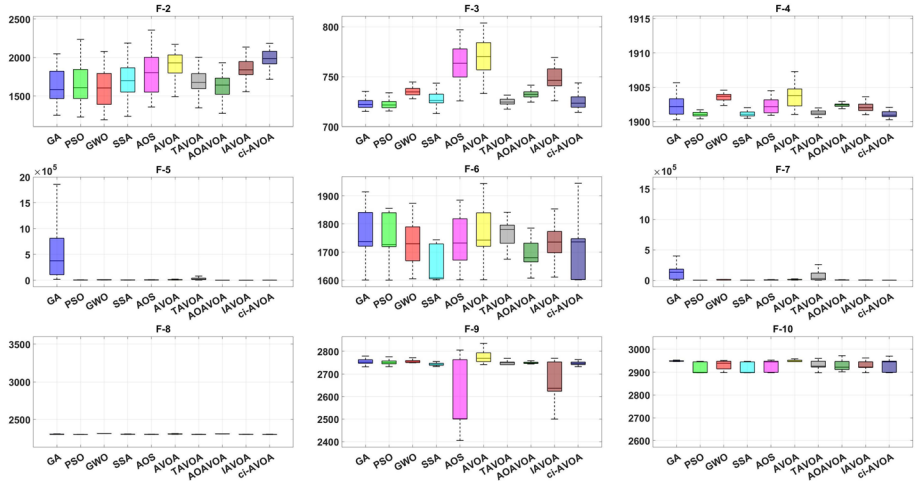


Fig. 18 The boxplot curves of the proposed *ci*-AVOA and the competitor algorithms obtained over basic,hybrid and composition functions from CEC2020 benchmark with $Dim = 10$

Overall, the *ci*-AVOA demonstrates superior performance, either outperforming or tying with the other algorithms on a majority of the functions. Notably, *ci*-AVOA's exceptional performance against GA-where it records victories in 5 out of 10 functions, ties in 4, and loses only in one-highlights its effectiveness and robustness. Even in instances where *ci*-AVOA does not take the lead, it often ties with the competing algorithms, suggesting comparable performance. Despite a few exceptions, such as its performance on Function F2, where it didn't fare as well, the *ci*-AVOA algorithm generally provides strong competition and proves to be a reliable and efficient option for tackling diverse optimization tasks.

This analysis indicates that *ci*-AVOA generally demonstrates superior or equivalent performance to the other algorithms on the tested functions, suggesting its effectiveness and robustness in handling optimization tasks of varying complexity.

The Friedman rank sum test is a non-parametric statistical test utilized for comparing more than two groups that are related. This test is applied to explore the mean rank differences across the various algorithmic approaches to understand their relative performances. In addition to the standard Friedman rank sum test, we employed two advanced variants of the test, specifically, the Friedman aligned ranks test and the Quade test. Both of these methods provide more robust, reliable statistical comparisons that account for specific data characteristics. Each test works by assigning rank values to each algorithm based on its performance. The lower the rank value, the better the algorithm's performance (Derrac et al. 2011).

The results depicted in Fig. 17a, b, and c show the PSO algorithm achieving the first rank, demonstrating its superior performance over the others. However, the *ci*-AVOA algorithm closely follows the PSO, obtaining the second rank and thus indicating its high effectiveness. While the PSO algorithm tops the rank, the relatively low rank of *ci*-AVOA indicates its commendable performance. It is notable that the proposed *ci*-AVOA approach generates results that are competitive with the leading PSO algorithm, especially considering the various complexities of low-dimensional optimization problems. In conclusion, these comprehensive test results provide evidence of the acceptability

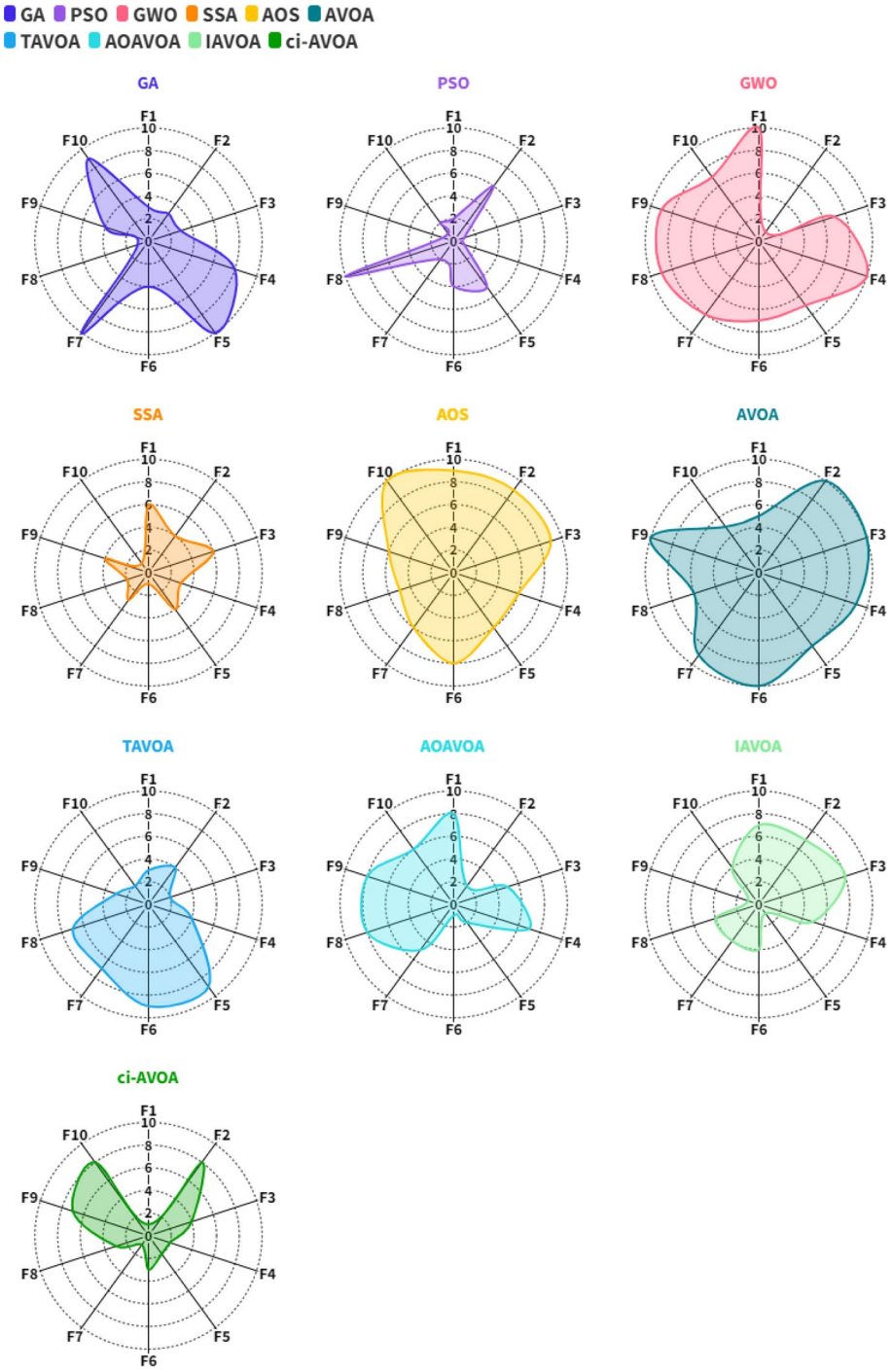


Fig. 19 The radarplot curves of the proposed *ci*-AVOA and the competitor algorithms obtained over CEC2020 benchmark with $Dim = 10$

and robustness of the proposed *ci*-AVOA method in handling low-dimensional problems, making it a strong candidate for application in diverse optimization tasks.

The utilization of Box-plot charts facilitates a comprehensive comparison of algorithmic performance results. These charts are instrumental in differentiating the potential of metaheuristic algorithms through their visual depiction of data symmetry and dispersion. A box plot communicates six essential statistical parameters, namely: the maximum and minimum values, median (or second quartile), first quartile (lower quartile), third quartile (upper quartile), and potential outliers. These parameters serve to provide a succinct summary of the data distribution and spread, allowing us to make broad assessments about the performance characteristics of each algorithm.

The Box-plot representations in Fig. 18 detail the distribution of the best and mean fitness values yielded by each algorithm for the basic, hybrid, and composition functions of the CEC2020 test suite. Observing these plots, it's noticeable that the *ci*-AVOA algorithm box plots exhibit the lowest values in almost all the tested functions, when compared to the other algorithms under consideration.

The median value of the *ci*-AVOA's results, which is the midpoint of the data set, along with its first and third quartiles, which respectively mark the midpoints of the lower and upper halves of the data, are distinctly lower than those of the other tested algorithms. This observation indicates that the Choquet fuzzy integral method, as incorporated in the proposed *ci*-AVOA algorithm, considerably bolsters the diversity within the population and enhances the algorithm's abilities in both exploration and exploitation.

Moreover, this finding is evidence of the robustness and stability of the proposed *ci*-AVOA algorithm. It demonstrates its effectiveness in dealing with a variety of optimization functions, as the lower quartile, median, and upper quartile all fall in the lower range of the performance metrics. This condensed spread of results indicates a high degree of consistency in the performance of the *ci*-AVOA algorithm across different optimization tasks.

On the other hand, the AOS and GA algorithms displayed less satisfactory performance. This is made evident through the larger spread of their box-plots, as indicated by the larger

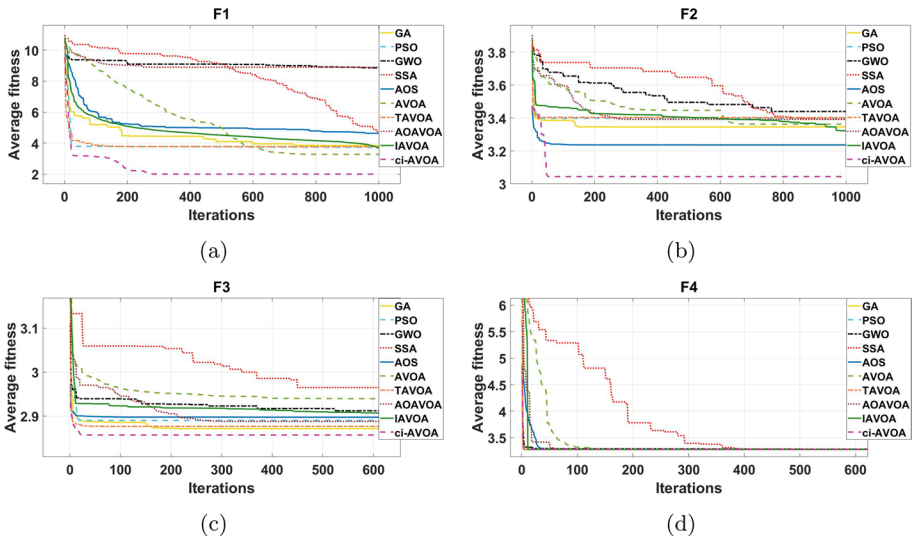


Fig. 20 Convergence curves of algorithms on the $f_1 - f_4$ 20-dimensional functions with respect to iterations

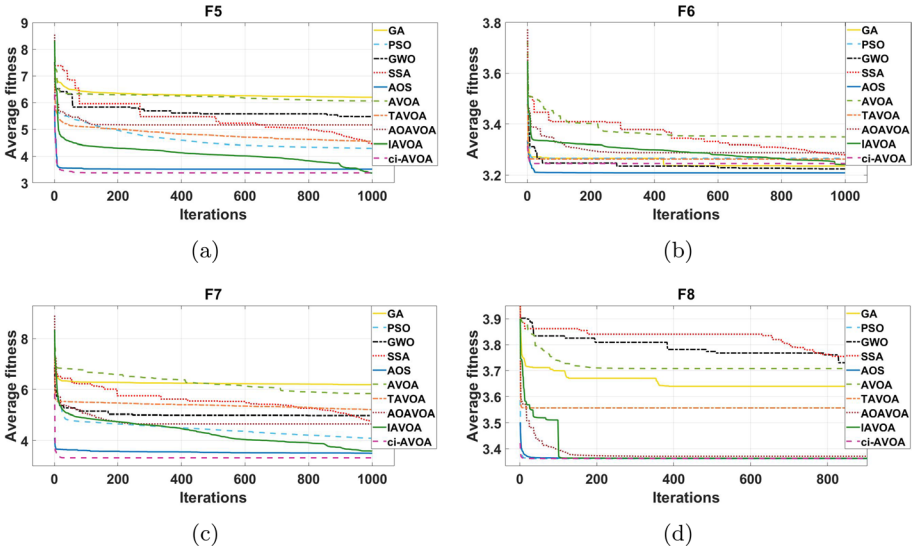


Fig. 21 Convergence curves of algorithms on the $f_5 - f_8$ 20-dimensional functions with respect to iterations

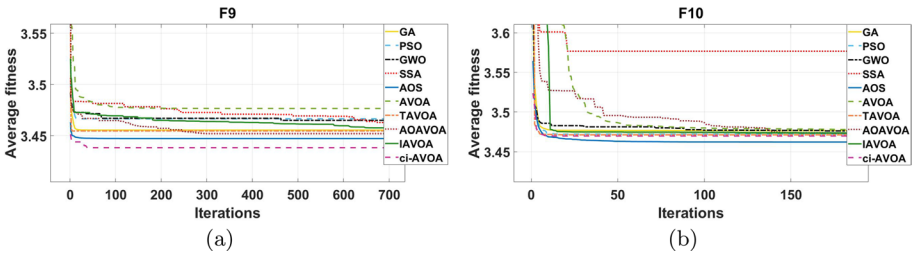


Fig. 22 Convergence curves of algorithms on the $f_9 - f_{10}$ 20-dimensional functions with respect to iterations

distance between their quartiles, denoting a larger variance in their performance results. This relatively wide distribution suggests inconsistency in their performance, making them less reliable when compared to *ci*-AVOA for handling diverse optimization problems.

The study utilizes an array of radar plots as an insightful means of visually comparing the performance of various algorithms. A distinguishing characteristic of radar plots is their utilization of shadow area to represent performance. The optimal method is, hence, the one that generates the smallest shadow area, corresponding to better performance across the multiple metrics represented on the radar plot axes.

The radar plots in this study use the average fitness distance from the known global optimum as a measure. This particular metric offers an insightful quantification of how close an algorithm's results are to the best possible outcome. By monitoring the proximity of the algorithm's fitness values to the global optimum, we gain a clear understanding of the algorithm's accuracy in converging towards the ideal solution.

The average fitness distance, the standard deviation (SD) of the average error is employed as an additional measure to assess the reliability of each algorithm. The standard deviation provides an estimate of the variability or dispersion of the average error values.

Lower standard deviation values indicate that the results are closely clustered around the mean, implying greater consistency and reliability in the performance of the algorithm.

Taking into account both the average fitness distance and standard deviation metrics, we can interpret the results depicted in Fig. 19. The radar plot allows for a multifaceted comparison of the algorithms' performance, considering both their accuracy in approaching the global optimum and their reliability in producing consistent results. From these combined assessments, it becomes evident that the proposed *ci*-AVOA outperforms the other algorithms. The *ci*-AVOA method yields a smaller shadow area in the radar plot, indicating closer proximity to the global optimum (lower average fitness distance) and a more concentrated cluster of results (lower standard deviation). This dual advantage highlights the superior accuracy and reliability of the *ci*-AVOA, making it a more trustworthy and effective choice for tackling optimization problems compared to the other algorithms under consideration.

4.5 Results for 20-dimensional CEC2020 problems in comparison to other metaheuristics

The performances of GA, PSO, GWO, SSA, AOS, AVOA, TAVOA, AOAVOA, IAVOA and *ci*-AVOA for 20-dimensional problems are given in Table 17. The obtained results from 30 independent runs on the different statistical measures of best, mean and standard deviation indicate that *ci*-AVOA succeeded in reaching the global optimum. For all the test functions unimodal, basic, hybrid and composition *ci*-AVOA outperformed the comparative algorithms. Unimodal and basic functions give insight into the exploitation of algorithms. Hybrid and composition functions present many local optimums where

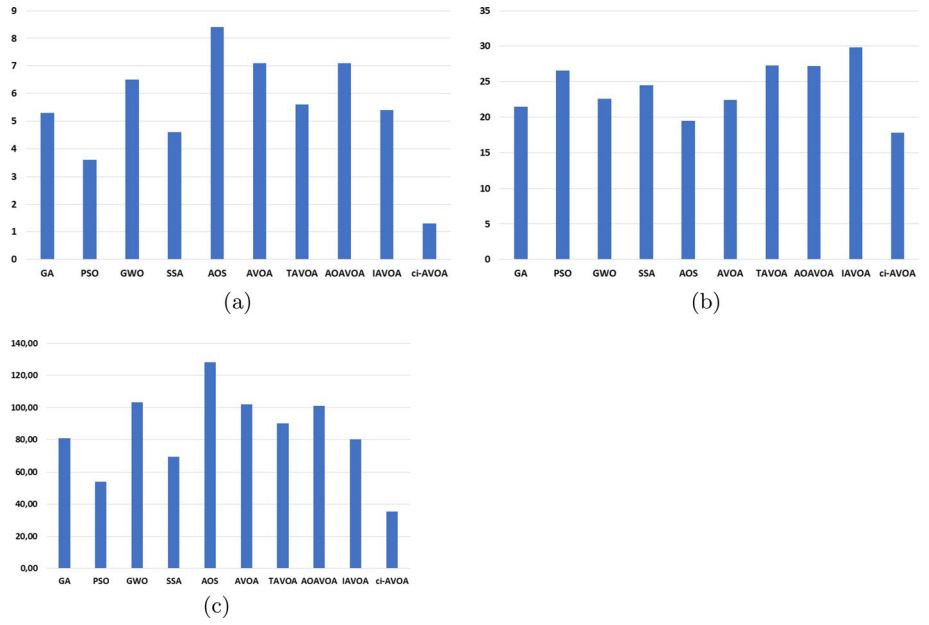


Fig. 23 Average ranking of the algorithms by Friedman test (**a**), Friedman aligned test (**b**) and Quade test (**c**) for 20-dimensional problems

the exploration of algorithms can be investigated. The obtained performance of *ci*-AVOA indicates that the proposed approach remarkably enhances the exploitation and exploration capabilities of the AVOA algorithm. We can conclude that the Choquet fuzzy integral improves the diversification and exploitation of solutions by considering how the variables interact with each other. Therefore, the proposed approach achieved good results regarding solution quality for high-dimensional problems.

In evaluating the performance of metaheuristic algorithms, a vital aspect to consider is the balance between exploration and exploitation. The *ci*-AVOA algorithm presents an interesting case in this regard, as evidenced by the convergence curves illustrated in Figs. 20, 21, and 22. These curves represent the algorithm's performance in 20-dimensional space for all CEC2020 test functions.

When the *ci*-AVOA convergence curve is adjacent with those of other algorithms such as GA, PSO, GWO, SSA, AOS, and AVOA, its unique properties become more discernible. One of the distinguishing features of the *ci*-AVOA curve is its relative smoothness. This indicates a gradual decrease in error over the optimization process. The smoother curve suggests that the *ci*-AVOA maintains a steady progression towards the global optimum without sudden variation.

This smooth progression is indicative of a well-balanced interaction between the algorithm's exploration and exploitation operators. In the context of optimization algorithms, exploration refers to the ability to probe diverse regions of the search space, while exploitation involves focusing the search around promising areas to refine solutions. A well-balanced algorithm effectively combines these two aspects, enabling it to identify the most promising regions of the search space (exploration) and then thoroughly examine these regions for the optimal solution (exploitation).

The *ci*-AVOA, with its well-balanced exploration and exploitation, demonstrates superior performance across a range of test function categories, including unimodal, basic, and hybrid. The algorithm not only attains higher accuracy but also excels in locating the global optimum more effectively compared to its counterparts. This performance manifests

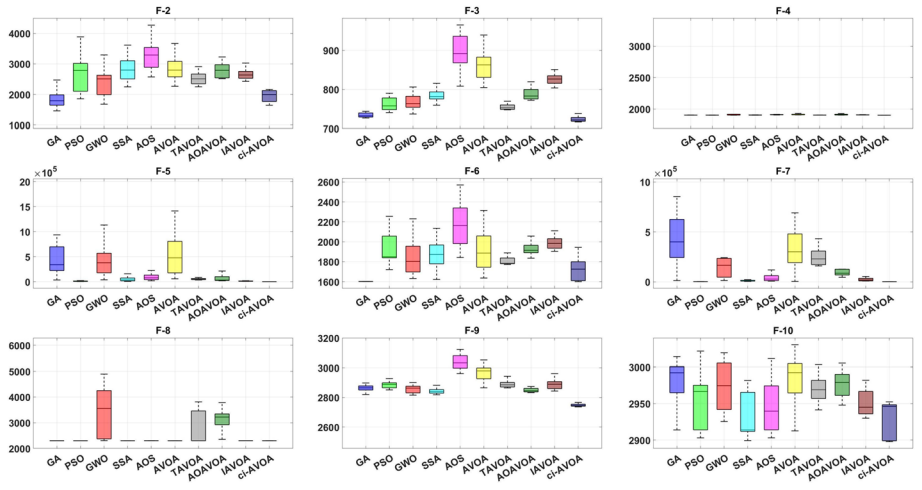


Fig. 24 The boxplot curves of the proposed *ci*-AVOA and the competitor algorithms obtained over basic,hybrid and composition functions from CEC2020 benchmark with $Dim = 20$

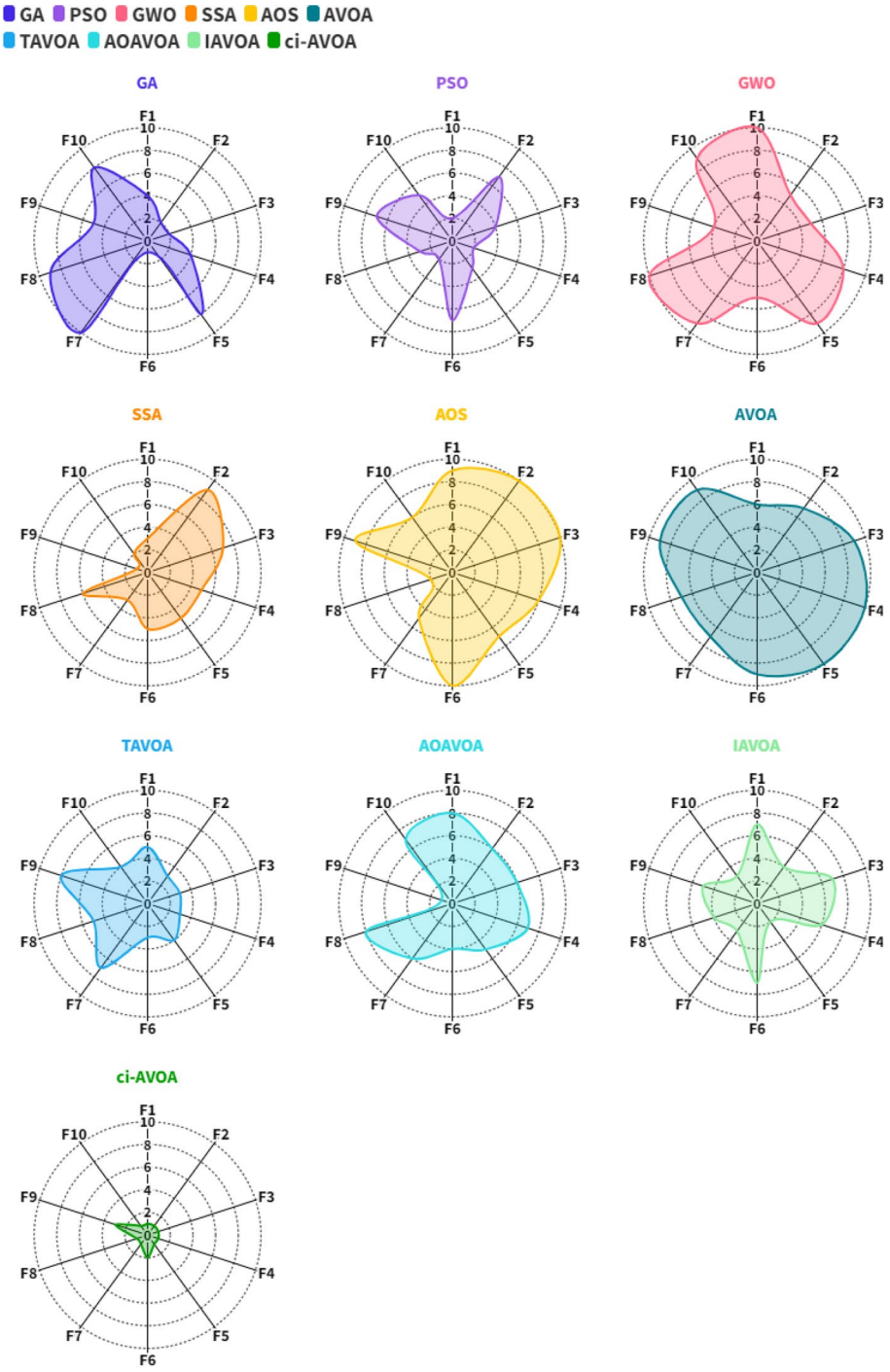


Fig. 25 The radarplot curves of the proposed *ci*-AVOA and the competitor algorithms obtained over CEC2020 benchmark with $Dim = 20$

in various ways, from achieving faster convergence to minimizing error rates, further highlighting the effectiveness and efficiency of the *ci*-AVOA algorithm.

Table 18 presents the results of a Wilcoxon rank sum test comparing the proposed *ci*-AVOA algorithm with other competing algorithms on the CEC2020 test suite functions, with a dimensionality of 20.

In this table, each function is tested against different algorithms (GA, PSO, GWO, SSA, AOS, AVOA, TAVOA, AOAVOA, and IAVOA), resulting in two primary outputs: p value (p) and win indication. The p value quantifies the probability that the difference observed is due to random chance, with a lower value indicating a stronger statistical evidence of an actual difference. In this table, smaller p values signify that the *ci*-AVOA algorithm performed significantly better than the compared algorithm.

For example, in the F1 test, *ci*-AVOA consistently outperformed all other algorithms, as indicated by the "+" symbol and very low p-values. In F2, the *ci*-AVOA performed similarly to GA ($p = 0.20843$, "="), but outperformed all other algorithms, as shown by the low p-values and "+" symbols. The test for function F8 presents a mixed outcome, with *ci*-AVOA performing similarly to GA and PSO, but outperforming GWO and AOS. In summary, *ci*-AVOA outperforms or is equal to the competing algorithms in the majority of the tests. The "Sum" row shows a tally of the results: *ci*-AVOA outperforms GA and PSO in 8 tests, draws in 2, and loses in none. It beats GWO and AOS in all 10 tests, and it wins over SSA and AVOA in 9 tests and draws in 1. This implies the *ci*-AVOA is a highly competitive algorithm on the CEC2020 test suite with a dimension of 20.

The Friedman statistical test and its more sophisticated versions, the Friedman aligned rank test and the Quade test, are deployed to evaluate the performance of various algorithms by considering the means of the objective function values. These non-parametric statistical procedures are particularly useful for comparing multiple paired groups to uncover significant differences. The strength of these tests lies in their ability to cope with non-normally distributed data, and they're frequently employed in the analysis of algorithmic performance in computational studies.

In our study, we use these statistical methods to compare *ci*-AVOA with other optimization approaches. Figure 23a, b, and c illustrate the results of the Friedman, Friedman aligned rank, and Quade tests, respectively.

The positive outcomes of these tests, as depicted in the mentioned figures, provide substantial support for the superior global optimization capability and stability of the *ci*-AVOA algorithm. These tests yield rankings of the different algorithms based on their performance, with lower ranks indicating superior performance. In our case, the *ci*-AVOA algorithm consistently achieves lower ranks, demonstrating its superior performance compared to the other algorithms tested. These tests, coupled with the other experimental results presented, confirms the case for the *ci*-AVOA's efficacy. Compared to the traditional AVOA and other well-established optimization approaches, *ci*-AVOA demonstrates consistently strong performance across multiple test scenarios and under various performance measures. Consequently, these experiments reaffirm the robustness, reliability, and superior performance of the proposed *ci*-AVOA algorithm in tackling complex optimization problems.

Box plots for basic, hybrid, and composition benchmark functions are presented in Fig. 24. These box plots provide a visual representation of the spread of the performance metrics of different optimization algorithms.

Upon careful examination, it becomes evident that the proposed *ci*-AVOA method consistently produces the smallest values in terms of overall minimum, maximum, and median metrics across almost all the test functions, with the sole exception being the

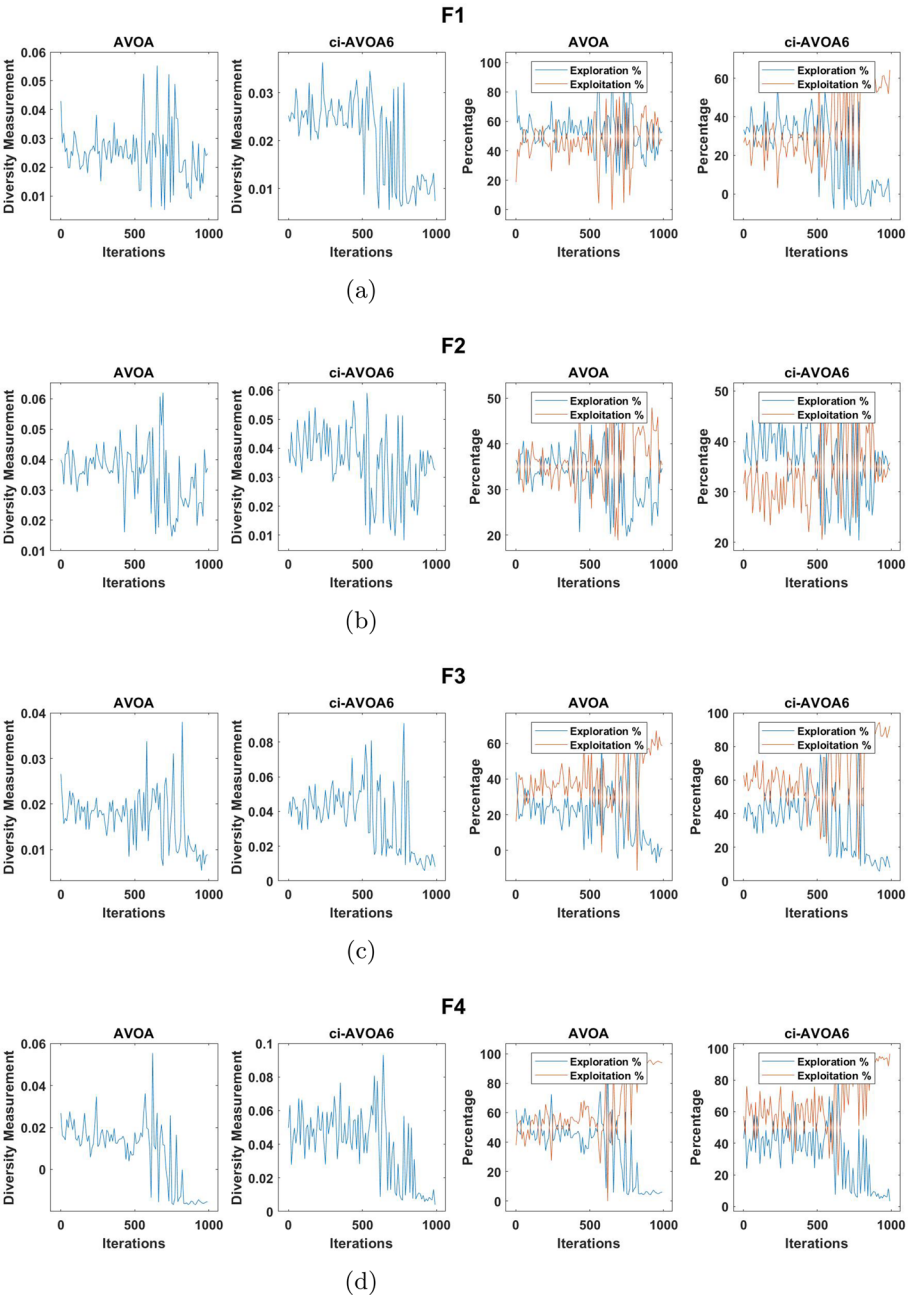


Fig. 26 Diversity analysis of AVOA for $f_1(a) - f_4(d)$ CEC2020 test functions

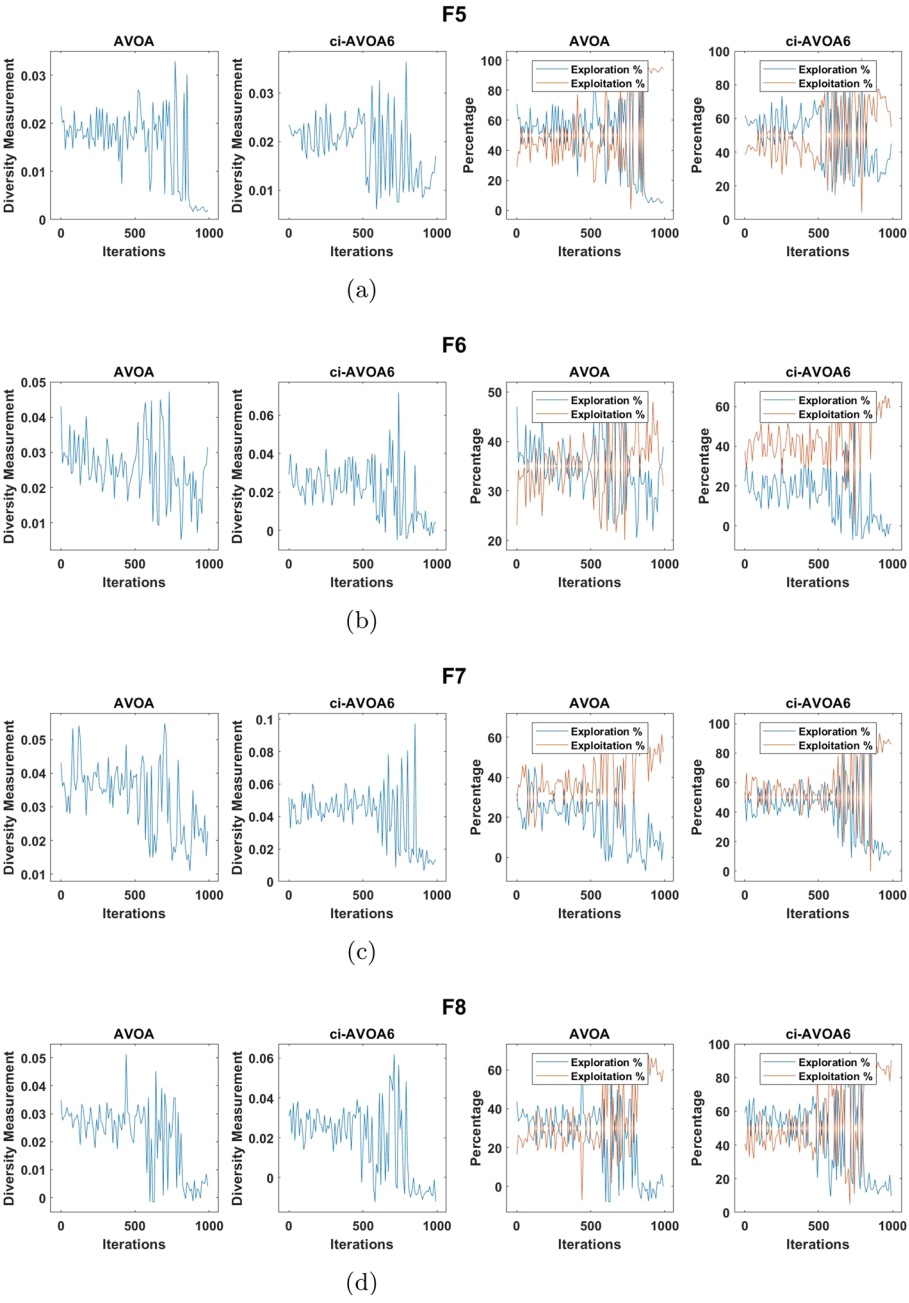


Fig. 27 Diversity analysis of AVOA for $f_5(a) - f_8(d)$ CEC2020 test functions

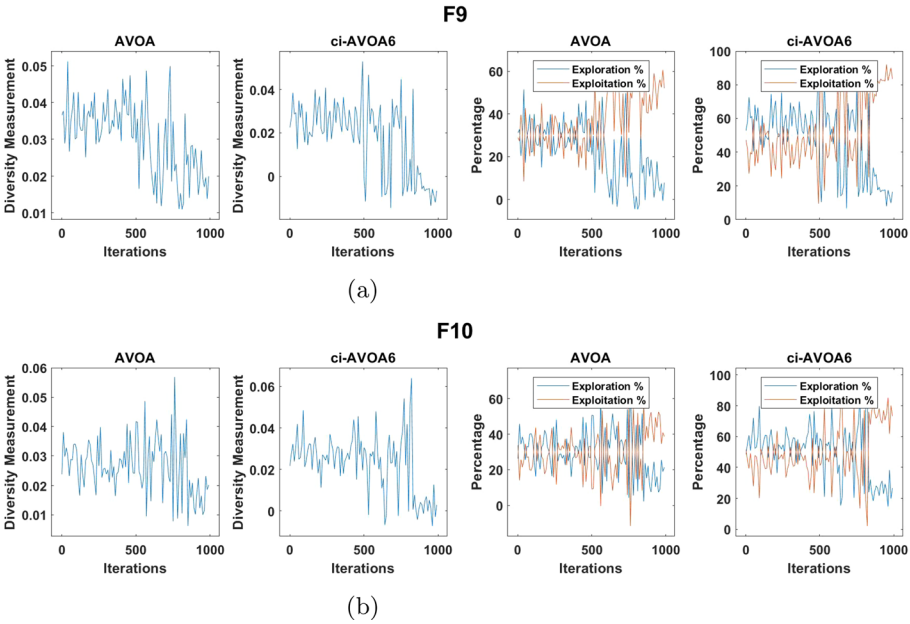


Fig. 28 Diversity analysis of AVOA for $f_9 - (a) f_{10}(b)$ CEC2020 test functions

function $f - 10$. These metrics highlight the high efficacy of the *ci*-AVOA method in finding the optimal solution, even in high-dimensional search spaces.

In addition to these positional measures, the dispersion and variance, which are measures of the spread of data, are also examined for each algorithm. Notably, *ci*-AVOA demonstrates the least spread amongst the algorithms compared, indicating its robustness and stability. A small spread suggests that the algorithm's performance is reliable, with little variation across different runs, making it an appealing choice for solving complex optimization problems.

Further reinforcing the superiority of the *ci*-AVOA method is the radar plot behavior analysis performed on the CEC2020 functions, as shown in Fig. 25. In a radar plot, each spoke represents a different measure of performance, and a smaller shadow area suggests superior performance across all measures. Here, the *ci*-AVOA method stands out, producing the smallest shadow area compared to the other algorithms.

This clearly underscores the strength of the *ci*-AVOA method, demonstrating its superior performance across multiple measures in high-dimensional search spaces. As such, the combined analysis of box plots and radar plots provides compelling evidence supporting the robustness, stability, and effectiveness of the *ci*-AVOA approach in tackling complex optimization problems.

4.6 Diversity analysis

The concept of maintaining solution diversity is widely acknowledged as a crucial factor in developing new metaheuristics. Working with a diverse range of solutions can enhance our understanding of the computational process, resulting in a more granular and thorough

analysis of the problem at hand (Osuna-Enciso et al. 2022). It's important to note, however, that relying purely on the study of convergence graphs and statistical measures, such as mean, best, worst, and standard deviation of solutions obtained across multiple iterations—may not offer a complete view of the metaheuristic algorithm's search behaviour. While these indicators provide valuable insights, they may not fully capture the effectiveness of these algorithms' performance in certain contexts. This concern is particularly salient for swarm-based metaheuristic algorithms, where understanding the behaviours of individual swarm members, and the swarm as a whole, is crucial for accurate performance assessment (Morales-Castañeda et al. 2020). Motivated by this need for a more comprehensive evaluation approach, our research employs a dimension-wise diversity measurement method, as recommended by previous studies (Hussain et al. 2019). This measurement strategy provides a more holistic and nuanced understanding of the algorithm's performance by quantifying the exploration and exploitation phases in a multi-dimensional context. The adopted technique allows us to monitor how the diversity of solutions evolves over time and helps us understand how effectively the algorithm explores the search space and exploits the promising regions. This analytical approach provides us with a more detailed and in-depth insight into the performance of the algorithm, thereby facilitating a more reliable and comprehensive evaluation. By understanding the exploration and exploitation phases quantitatively, we can better gauge the algorithm's effectiveness and its potential applications in tackling complex optimization problems. Consequently, the dimension-wise diversity measurement technique serves as an invaluable tool in our study, enhancing our ability to assess the proposed *ci*-AVOA algorithm's performance thoroughly and accurately. The specific formulas for quantifying these phases are presented in the following equations:

$$\mathrm{Div}_j = \frac{1}{n\,\mathrm{Pop}} \sum_{i=1}^{n\,\mathrm{Pop}} \mathrm{median}\left(\mathrm{Prey}^j\right) - \mathrm{Prey}_i^j$$

$$\mathrm{Div} = \frac{1}{\mathrm{Dim}} \sum_{j=1}^{\mathrm{Dim}} \mathrm{Div}_j$$

where median is the median of dimension j in the Prey matrix.

$$\mathrm{Exploration} = \frac{\mathrm{Div}}{\mathrm{Div}} \times 100$$

$$\mathrm{Exploitation} = \frac{|\ \mathrm{Div} - \mathrm{Div}_{\max}\ |}{\mathrm{Div}} \times 100$$

When examining Figs. 26, 27, and 28, a notable feature of the African Vultures Optimization Algorithm based on the Choquet Integral (*ci*-AVOA) becomes apparent—a distinctive ability to oscillate between two distinct phases of diversity. In an impressive display, the *ci*-AVOA sustains high average diversity throughout the entire range of test functions included in the CEC2020 benchmark set

This propensity of *ci*-AVOA to maintain high levels of diversity is a critical strength of this metaheuristic algorithm. This characteristic avoids premature convergence to local optima and ensures an exhaustive exploration of the solution space, increasing the probability of locating the global optimum.

The alternation between the two phases signifies a dynamic balance maintained by *ci*-AVOA between exploration and exploitation throughout the optimization process, a balance that is vital to the success of any metaheuristic algorithm. The exploration phase,

Table 8 Time execution comparison of different algorithms

Algorithm	GA		PSO		GWO		SSA		AOS		AVOA		TAVOA		AOAVOA		IAVOA		*ci*-AVOA	
Function	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20	D=10	D=20
f-1	30.72	227.61	19.38	94.27	11.99	56.93	12.66	60.65	25.31	126.31	27.2	44.20	24.65	117.58	68.02	410.85	50.10	321.88	309.53	537.50
f-2	29.07	157.87	19.21	90.38	11.66	62.67	12.83	58.94	27.38	121.13	15.9	50.50	24.49	121.61	68.11	341.67	48.28	248.25	302.23	400.73
f-3	30.23	148.16	17.93	84.56	12.38	60.06	12.76	58.84	25.70	122.28	13.2	44.04	25.14	118.90	68.31	330.51	48.15	232.72	429.25	371.48
f-4	28.47	154.45	17.24	92.05	10.70	55.46	10.91	56.54	24.83	127.72	20.6	51.61	25.62	112.00	64.00	337.63	45.71	246.51	465.06	387.75
f-5	28.52	177.48	16.81	93.94	10.72	59.62	11.01	61.37	23.79	120.47	11.71	52.37	21.73	120.98	63.02	357.56	45.33	271.42	528.06	400.30
f-6	30.97	148.36	19.07	235.93	12.15	70.01	12.67	73.52	25.51	126.76	18.41	51.83	24.82	143.53	68.63	168.03	50.04	384.29	419.76	1254.67
f-7	30.96	156.13	19.12	103.18	11.23	88.16	11.19	75.15	25.84	368.43	17.90	64.35	22.42	163.31	68.02	612.72	50.08	259.31	384.15	623.90
f-8	29.13	158.74	17.02	104.853	11.13	67.64	11.56	73.41	24.51	151.22	21.68	72.17	32.69	141.05	64.77	377.60	46.15	207.27	371.88	426.87
f-9	29.32	163.40	17.17	100.86	11.33	72.12	11.85	77.25	24.62	135.60	12.07	57.15	23.18	149.37	65.28	371.12	46.49	264.26	410.76	534.38
f-10	28.86	170.13	17.02	101.98	11.14	64.10	11.54	66.82	24.40	150.49	12.00	58.75	22.67	130.93	64.39	384.72	45.88	272.11	406.45	1192.94

Table 9 Algorithm complexity comparison

Algorithm	Complexity
GA	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
PSO	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
GWO	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
SSA	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
AOS	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
AVOA	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$
TAVOA	–
AOAVOA	–
IAVOA	–
ci-AVOA	$\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D \times 2^{2D})$

hallmarked by substantial diversity, empowers the *ci*-AVOA to scan a broad region of the solution space. Conversely, the exploitation phase concentrates the search towards promising areas within the solution space, directing efforts towards finding high-quality solutions.

The demonstrated capacity of *ci*-AVOA to preserve high diversity levels while effectively oscillating between these phases suggests a degree of adaptability and flexibility. This adaptability enables *ci*-AVOA to adjust its search strategy based on the current stage of the optimization process and the characteristics of the solution space. For instance, it may lean towards exploration in the early stages or when the search seems to be trapped in a suboptimal region, shifting its focus to exploitation when promising solutions have been identified.

Thus, the ability of *ci*-AVOA to maintain diversity and balance between exploration and exploitation plays a significant role in its robust performance on the CEC2020 benchmark functions. This adaptability makes the *ci*-AVOA well-suited for tackling a wide array of optimization problems.

Subsequent analysis confirms that the *ci*-AVOA inherently supports diversity. This is elucidated as follows: initially, the algorithm deploys multiple agents (vultures) into different groups, spreading them across the solution space. This dispersion of starting points encourages a broad exploration of the solution space from the onset. As the algorithm progresses, the positions of the vultures are iteratively updated, influenced by the best-performing vulture of their group and the collective average position, ensuring continuous diversity. This approach allows the algorithm to balance exploration and exploitation effectively. Moreover, by continuously validating the vultures' positions within the defined solution boundaries, the algorithm prevents convergence towards invalid solutions, encouraging exploration across all feasible regions within the problem domain. These inherent mechanisms within the *ci*-AVOA promote overall diversity, supporting thorough exploration and avoiding premature convergence to inferior solutions.

4.7 Computational complexity

The time complexity analysis using Big-$\mathcal{O}$ notation is covered in this subsection. Its goal is to clarify how the suggested method can improve performance. The population size (noted as N), the number of dimensions (indicated as D), and the maximum number of iterations (stated as T) all have a significant bearing on how well the proposed method performs.

In line with the original research paper (Abdollahzadeh et al. 2021), the computation complexity of the AVOA algorithm is contingent on three critical procedures: the initialization phase $\mathcal{O}(N)$, the evaluation of fitness $\mathcal{O}(N \times T)$, and the vultures' update process $\mathcal{O}(N \times T \times D)$). Therefore, authors claimed that the computational complexity is equal to $\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D)$.

Nonetheless, in the proposed approach, we revamp the update mechanism by incorporating a novel averaging technique known as the Choquet integral. Hence, the complexity of the vultures' update process will be altered because the Choquet integral requires more computational resources compared to the original averaging technique. The most challenging task in this new averaging technique is the calculation of the λ-measure, which requires a computational complexity of $\mathcal{O}(2^{2D})$. It is evident to note that this procedure is significantly computationally demanding. Therefore, the computational of the proposed approach *ci*-AVOA equates to: $\mathcal{O}(N \times T) + \mathcal{O}(N \times T \times D \times 2^{2D})$.

Table 9 present a comprehensive comparison of various algorithms based on their time execution and complexity. The purpose of this comparison is to analyze the efficiency and performance of these algorithms in solving specific computational problems.

From a cursory look at the Table 8, it's clear that the execution times of the *ci*-AVOA algorithm are generally higher compared to the other algorithms, especially in the D=20 dimension. The execution times range from approximately 302.23 (for function f-2 with D=10) to a high of 1254.67 (for function f-6 with D=20). The data implies that *ci*-AVOA may be a more complex and computationally heavy algorithm compared to the others. However, a critical aspect to consider is the quality of the solutions obtained from these execution times. The *ci*-AVOA algorithm, while slower, provides more optimal and robust solutions to the optimization problems.

Table 9 provides a comparison of the computational complexity of the various algorithms of this study. The complexity expression suggests that *ci*-AVOA has two main components: one that scales linearly with both the number of items and the number of iterations (i.e., $\mathcal{O}(N \times T)$), and another that increases exponentially with the dimensionality (i.e., $\mathcal{O}(N \times T \times D \times 2^{2D})$).

This suggests that *ci*-AVOA's computational complexity can grow rapidly as the dimensionality of the problem (D) increases, due to the 2^{2D} term. This could be one of the reasons why *ci*-AVOA had longer execution times in the earlier Table 8, especially for larger values of D.

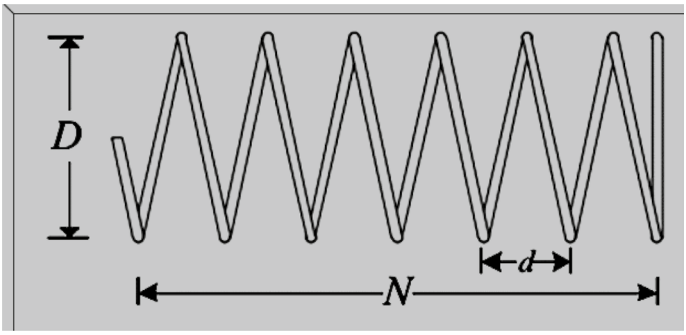


Fig. 29 The schematic presentation of the tension/compression design problem

Table 10 Comparison of results for tension/compression design problem

Algorithm	Best	Mean	Worst	X1	X2	X3
GA	0.01270775	0.0142465	0.0178335	0.05323457	0.39505032	9.35085622
GWO	0.01266527	0.01330382	0.01454197	0.05164673	0.35570014	11.3488749
PSO	0.01268448	0.01274235	0.01277366	0.05192345	0.36205162	10.9949711
SSA	0.01268617	0.01272374	0.01279144	0.05099097	0.34013677	12.3446656
AOS	0.01266711	0.0130054	0.01562856	0.05158892	0.35431069	11.4332309
AVOA	0.01267314	0.01279046	0.01309318	0.05103479	0.34118068	12.2615592
ci-AVOA	**0.01266523**	**0.01266523**	**0.01266523**	0.05168906	0.35671775	11.2889653

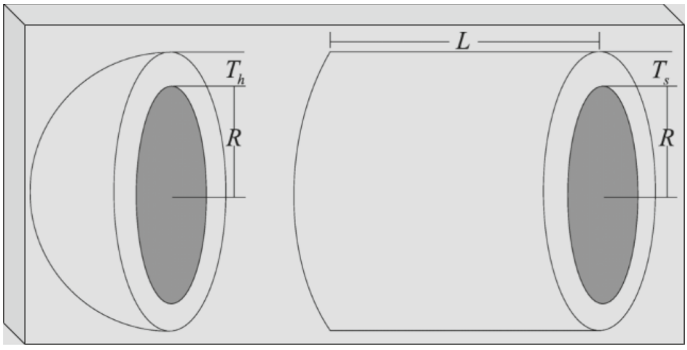


Fig. 30 The schematic presentation of the pressure vessel design problem

In summary, although the computational complexity of *ci*-AVOA indicates it might struggle with high-dimensional optimization problems due to its exponential scaling, it remains a powerful algorithm capable of solving complex problems, as demonstrated by previous tests.

In terms of optimization metaheuristics, the quality of a solution depends on both the performance of the algorithm and the efficiency of the algorithm.

When you're dealing with high-dimensional optimization problems, efficiency becomes particularly important. If an algorithm, such as *ci*-AVOA, has higher computational complexity, but consistently provides better or more accurate solutions, it could still be considered powerful and useful for certain complex problems.

Therefore, the trade-off between computational complexity and the quality of solutions is a common theme in optimization. While the *ci*-AVOA algorithm may have higher computational complexity, its ability to solve complex problems indicates it's still a valuable tool in the optimization metaheuristics.

4.8 Discussion and analysis

The superior performance of the proposed *ci*-AVOA algorithm can be primarily attributed to its innovative hybridization of the Choquet fuzzy integral and the AVOA. This merge exploits the strengths of both techniques, enabling dynamic potential movement search for each solution during the optimization process. In contrast, the traditional AVOA

Table 11 Comparison of results for pressure vessel design problem

Algorithm	Best	Mean	Worst	X1	X2	X3	X4
GA	5990.34064	6536.90565	7320.83108	0.8352327	0.41284216	43.2748592	162.584925
GWO	5950.78927	6412.74126	7179.14848	0.8147187	0.40271588	42.2134043	175.217262
PSO	5909.38713	6057.11384	7374.93904	0.77923078	0.38999352	40.3251009	200
SSA	5925.1993	6394.75126	7224.71161	0.78989938	0.39036597	40.9160713	192.686672
AOS	5988.33285	6598.28914	7323.72698	0.81601984	0.41418318	42.2764263	174.449778
AVOA	5885.36694	6160.28651	7260.51134	0.77818863	0.38465904	40.3206543	199.985585
ci-AVOA	**5885.33277**	**5885.33277**	**5885.33277**	0.77816864	0.38464916	40.3196187	200

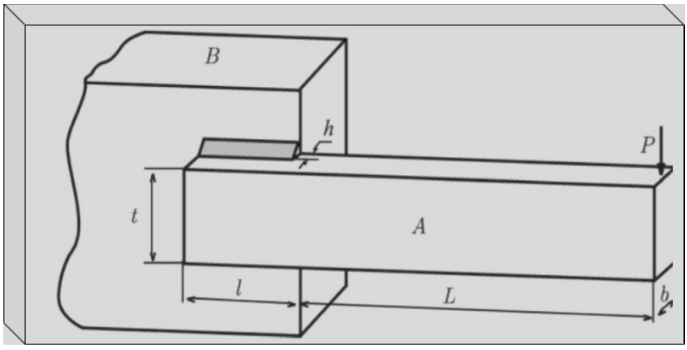


Fig. 31 The schematic presentation of the welded beam design problem

relies solely on a random operator to update the population position throughout the search process.

The rationale behind this fusion is the belief that the concept of fuzzy aggregation, when used in tuning the AVOA control parameters, provides a more effective optimization strategy than using the traditional AVOA approach. The application of the Choquet fuzzy integral in both the exploration and exploitation phases of the AVOA ensures a balanced interplay between the two, enhancing the algorithm's efficacy.

Early in the optimization process, this unique approach emphasizes the exploration ability, which promotes population diversity and broadens the search space. Later in the process, it assists the algorithm in exploiting the favorable solutions that have been uncovered, thereby accelerating the speed of convergence.

Several notable advantages accompany the incorporation of the Choquet fuzzy integral in the AVOA. Firstly, it enhances population diversity and mitigates the issue of local optima, which can be a prevalent problem in optimization algorithms. Secondly, its implementation is straightforward, making it accessible to various users. Lastly, and most distinctively, it does not necessitate any parameter tuning, setting it apart from other methods that often require complex and time-consuming parameter adjustment.

The *ci*-AVOA algorithm's robustness and improved accuracy make it a vital tool in fields where precision is of paramount importance. It has the potential to be applied in various domains, such as supply chain management, where it can optimize routing to save

resources; energy systems for efficient grid management; healthcare, where it can aid in effective resource allocation; and machine learning, particularly for hyperparameter tuning.

Furthermore, the *ci*-AVOA algorithm's unique approach advances our understanding of swarm intelligence. It highlights how the incorporation of fuzzy logic principles into swarm intelligence algorithms can significantly enhance their performance. This innovative strategy sets a benchmark for future research in the field of global optimization and opens up new avenues for further exploration and development. This development could stimulate additional research and innovation, pushing the boundaries of what is currently achievable in the field of optimization.

5 *ci*-AVOA for engineering applications

Ensuring the robustness of optimization algorithms can be achieved by addressing engineering design problems, as the outcomes of these problems serve as a crucial benchmark for optimizing other comparable problems. Thus, the behaviour of the proposed *ci*-AVOA is tested on four classical engineering design problems that are considered optimization problems with most of the metaheuristic algorithms. The four engineering problems presented in this section are (i) tension/compression spring, (ii) pressure vessel; (iii) welded beam and (iv) gear train. The optimization algorithms were executed for 30 independent with 500 iterations.

5.1 Tension/compression spring design problem

The purpose of the tension/compression spring design (TCSD) problem is to find the ideal number of active coils (N), wire diameter (d), and mean coil diameter (D) of spring, as shown in Fig. 29, which will result in the least amount of spring weight. The mathematical formulation of the tension/compression design problem is given below.

Minimize $f(X) = \left(x_3 + 2\right)x_2x_1^2$

Subject to

Table 12 Comparison of results for welded beam design problem

Algorithm	Best	Mean	Worst	X1	X2	X3	X4
GA	1.78631579	2.51426605	3.60742656	0.17321513	4.34420024	9.02784068	0.20613015
GWO	**1.72485231**	1.78108601	2.1075185	0.20572964	3.47048867	9.03662391	0.20572964
PSO	1.73488278	1.75345314	1.76593505	0.20535434	3.49930325	9.01875188	0.20702005
SSA	1.72513587	1.72691836	1.73705089	0.20570272	3.47163804	9.03786915	0.20572361
AOS	1.72730524	1.81504944	2.00130025	0.20430742	3.50146843	9.03977477	0.20572221
AVOA	1.72485256	1.72502571	1.72687098	0.2057295	3.47049148	9.03662461	0.20572964
***ci*-AVOA**	**1.72485231**	**1.72485231**	**1.72485231**	0.20572964	3.47048867	9.03662391	0.20572964

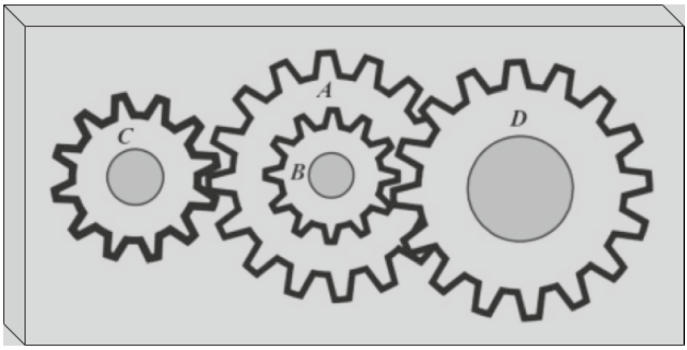


Fig. 32 The schematic presentation of the gear train design problem

Table 13 Comparison of results for gear train design problem

Algorithm	Best	Mean	Worst	X1	X2	X3	X4
GA	**0**	1.53E-15	4.56E-14	51.5944255	24.6223807	16.5452638	54.7263968
GWO	**0**	**0**	**0**	36.129187	18.0363861	15.7222344	54.4003257
PSO	6.99E-22	3.58E-13	2.67E-12	45.9088373	14.7172905	17.7753382	39.495293
SSA	7.63E-19	1.02E-15	6.30E-15	41.2387818	21.9110894	12	44.1911481
AOS	4.46E-18	1.90E-14	8.85E-14	41.3674391	14.1494424	12	28.4483997
AVOA	**0**	**0**	**0**	35.2164701	13.0446141	13.8588131	35.580115
ci-AVOA	**0**	**0**	**0**	47.6613282	21.7918623	15.5468775	49.2682662

$$g_1(X) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \le 0, \quad g_2(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_3(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0, \quad g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variables range: $0.05 \le x_1 \le 2.00, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15.0$

Table 10 displays a comparison of the design variables and results obtained for the tension/compression spring design problem using the *ci*-AVOA algorithm and other optimization algorithms. The best, average, and worst optimal values are highlighted in bold font. Despite the close and competitive results, the *ci*-AVOA algorithm outperforms the other algorithms in solving the tension/compression spring design problem.

5.2 Pressure vessel design problem

The goal of optimizing the pressure vessel design problem depicted in Fig. 30 is to minimize the expenses related to material, welding, and shaping. The four variables to be optimized in this problem are the thickness (T_s), the inner radius (R), the head thickness (T_h), and the length of the cylindrical section of the vessel (L).

Minimize $f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

Table 14 15 real-world constrained optimization problems (Kumar et al. 2020)

Prob	Code	Name	D	g	h	f*
RW1	RC15	Weight Minimization of a Speed Reducer	7	11	0	2.9944244658E+03
RW2	RC16	Optimal Design of Industrial refrigeration System	14	15	0	3.2213000814E-02
RW3	RC20	Three-bar truss design problem	2	3	0	2.6389584338E+02
RW4	RC21	Multiple disk clutch brake design problem	5	7	0	2.3524245790E-01
RW5	RC22	Planetary gear train design optimization problem	9	10	1	5.2576870748E-01
RW6	RC23	Step-cone pulley problem	5	8	3	1.6069868725E+01
RW7	RC24	Robot gripper problem	7	7	0	2.5287918415E+00
RW8	RC25	Hydro-static thrust bearing design problem	4	7	0	1.6161197651E+03
RW9	RC26	Four-stage gear box problem	22	86	0	3.5359231973E+01
RW10	RC27	10-bar truss design	10	3	0	5.2445076066E+02
RW11	RC28	Rolling element bearing	10	9	0	1.4614135715E+04
RW12	RC29	Gas Transmission Compressor Design (GTCD)	4	1	0	2.9648954173E+06
RW13	RC30	Tension/compression spring design (case 2)	3	8	0	2.6138840583E+00
RW14	RC32	Himmelblau's Function	5	6	0	-3.0665538672E+04
RW15	RC33	Topology Optimization	30	30	0	2.6393464970E+00

Subject to

$$g_1(X) = -x_1 + 0.019x_3 \le 0, \quad g_2(X) = -x_2 + 0.00954x_3 \le 0,$$
$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, g_4(X) = x_4 - 240 \le 0,$$

Variables range: $0 \le x_1 \le 99, 0 \le x_2 \le 99, 10 \le x_3 \le 200, 10 \le x_4 \le 200$

Table 11 presented in this study compares the performance of the proposed *ci*-AVOA algorithm with other optimization algorithms in solving the pressure vessel design problem. The results are shown in Table 11, with the best values highlighted in bold font. The comparison shows that the proposed *ci*-AVOA algorithm performs better than all the other algorithms in terms of all statistical indicators. Moreover, each data for the proposed *ci*-AVOA algorithm is the best among all the other optimization algorithms for solving the pressure vessel design problem.

5.3 Welded beam design problem

The objective of the welded beam design problem is to minimize fabrication costs, subject to certain constraints. The problem involves four design variables: weld thickness (h), length of the attached section of the bar (l), height of the bar (t), and thickness of the bar (b). The problem structure is illustrated in Fig. 31, and the mathematical model associated with the problem is defined as follows.

Minimize $f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$

Subject to

$$g_1(X) = \tau(X) - \tau_{\max} \le 0$$
$$g_2(X) = \sigma(X) - \sigma_{\max} \le 0$$
$$g_3 = x_1 - x_4 \le 0$$
$$g_4(X) = 0.1047x_1^2 + 0.04811x_3x_4\left(14.0 + x_2\right) - 5.0 \le 0$$
$$g_5(X) = 0.125 - x_1 \le 0$$
$$g_6(X) = \delta(X) - \delta_{\max} \le 0$$
$$g_7(X) = P - P_{\mathrm{c}}(X) \le 0$$
$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{X_2}{2\mathbb{R}} + (\tau'')^2}$$
$$\tau' = \frac{p}{\sqrt{2x_1x_2}} \quad \tau'' = \frac{MR}{\mathsf{T}}$$
$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_1^2}{12} + \left(\frac{x_1 + x_1}{2}\right)^2\right]\right\}$$
$$\sigma(X) = \frac{6\mathrm{PL}_2}{x_4x_3^2}, \quad \delta(X) = \frac{4\mathrm{PL}^3}{\mathrm{E}_4x_3^3}$$
$$P = 6000\mathrm{lb},\ L = 14\ \mathrm{in}$$
$$E = 30 \times 10^6\mathrm{psi} \quad G = 12 \times 10^6\mathrm{psi}$$
$$\tau_{\max} = 13.6 \times 10^3\mathrm{psi}, \quad \sigma_{\max} = 30 \times 10^3\mathrm{psi}, \quad \delta_{\max} = 0.25$$
Variables range
$$0.1 \le x_{1,4} \le 2$$
$$0.1 \le x_{23} \le 10$$

Table 12 presents the statistical and comparison results of various optimization algorithms and the proposed *ci*-AVOA approach for the welded beam design problem. The results show that the *ci*-AVOA approach performed better than the other algorithms, and its results were comparable to those of the GWO algorithm.

5.4 Gear train design problem

The main objective of this engineering design challenge is to optimize the maximum errors in the gear ratio of the gear train that is commonly used in automobiles. The problem involves four integer design variables that depend on the number of teeth in the gears, and three discrete design variables, which are the first gear module ($m1$), the second gear module ($m2$), and the number of gear teeth (P). Furthermore, there are ten inequality constraints and one equality constraint in this problem. The structure of the problem is illustrated in Fig. 32, and its mathematical model is defined accordingly.

Minimize $f(X) = \left(\frac{1}{6.931} - \frac{x_1x_2}{x_3x_4}\right)^2$

Table 15 Statistical results of 15 real-world constrained optimization problems

Problem	COLSHADE	EnMODE	sCMAgES	SSAS	AVOA	*ci*-AVOA
RW1						
Best	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**
Avg	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**
Worst	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**	**2994.42447**
Std	9.82E-13	**4.64E-13**	5.17E-12	2.92E-09	1.12E-12	**4.64E-13**
RW2						
Best	**0.032213**	**0.032213**	0.03696494	0.03221359	0.03714448	**0.032213**
Avg	**0.032213**	**0.032213**	0.03840363	0.03221395	4757.91991	0.03221307
Worst	**0.032213**	**0.032213**	0.04150552	0.03221555	33826.8354	0.03221338
Std	**1.26E-16**	2.02E-10	0.0011229	4.32E-07	10712.1372	1.09E-07
RW3						
Best	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**
Avg	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**
Worst	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**	**263.895843**
Std	**0**	**0**	8.22E-11	1.66E-12	1.16E-14	**0**
RW4						
Best	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**
Avg	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**
Worst	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**	**0.23524246**
Std	**0**	1.13E-16	1.13E-16	4.89E-10	1.09E-16	1.13E-16
RW5						
Best	0.53705882	0.5268955	0.53	0.57292431	0.54984911	**0.52576871**
Avg	0.64206488	**0.5283951**	0.53314762	1.10654775	0.56664177	0.53206514
Worst	2.83117647	**0.5308577**	0.54384615	2.67909021	0.61201238	0.53705882
Std	0.45694122	0.00129033	0.00398768	0.47461922	0.01543045	0.00405916
RW6						
Best	**16.0698687**	**16.0698687**	16.1990361	**16.0698687**	**16.0698687**	**16.0698687**
Avg	**16.0698687**	**16.0698687**	16.3776146	**16.0698687**	**16.1535559**	**16.0698687**
Worst	**16.0698687**	**16.0698687**	16.7039405	**16.0698687**	**17.1159581**	**16.0698687**
Std	1.79E-14	3.33E-14	0.15520876	2.08E-09	0.28964902	**1.13E-14**
RW7						
Best	**2.54378557**	**2.54378557**	2.91492326	2.54378558	2.57474504	2.54378558
Avg	2.58033738	**2.54378557**	3.09322711	2.54380069	2.67109132	2.54378922
Worst	3.44229386	**2.54378557**	3.49041474	2.54416316	3.17768375	2.54380133
Std	0.17960025	**3.86E-12**	0.1511585	7.55E-05	0.19044695	4.47E-06
RW8						
Best	**1616.11977**	**1616.11977**	3089.46436	1616.11983	**1616.11977**	**1616.11977**
Avg	1634.73016	**1616.11977**	3323.92226	1616.1447	**1616.11977**	**1616.11977**
Worst	1753.53866	1616.11978	3580.6948	1616.47613	**1616.11977**	**1616.11977**
Std	34.815161	2.06E-06	137.822241	0.07302926	6.59E-09	**1.20E-12**
RW9						
Best	42.8049656	37.3488483	64.7161639	38.6528218	112.464526	**7.27671526**
Avg	45.2694825	38.3322953	95.2787368	40.8252695	173.943721	**25.3568829**
Worst	48.6693014	40.2824795	209.803404	45.5060439	384.001456	**39.5752718**

Table 15 (continued)

Problem	COLSHADE	EnMODE	sCMAgES	SSAS	AVOA	*ci*-AVOA
Std	1.68312303	**0.80578114**	35.9972536	2.14956722	67.7992945	8.45449481
RW10						
Best	**524.450761**	**524.450761**	524.608739	524.468308	530.871439	524.459106
Avg	**524.450761**	**524.450761**	524.801769	524.474592	531.949586	524.616274
Worst	**524.450761**	**524.450761**	525.132758	524.490874	534.303135	524.764671
Std	**1.76E-10**	3.97E-08	0.15763762	0.00676661	1.03843925	0.10481801
RW11						
Best	16958.2023	**14614.1357**	**14614.1357**	**14614.1357**	16958.2023	16958.2023
Avg	16958.2023	**14614.1357**	**14614.1357**	**14614.1357**	16958.2023	16958.2023
Worst	16958.2023	**14614.1357**	**14614.1357**	**14614.1357**	16958.2023	16958.2023
Std	**3.71E-12**	9.28E-12	8.97E-12	1.56E-09	3.44E-11	**3.71E-12**
RW12						
Best	**2964895.42**	**2964895.42**	2964897.19	**2964895.42**	**2964895.42**	**2964895.42**
Avg	**2964895.42**	**2964895.42**	2964929.69	**2964895.42**	**2964895.42**	**2964895.42**
Worst	**2964895.42**	**2964895.42**	2965096.83	**2964895.42**	**2964895.42**	**2964895.42**
Std	1.43E-09	1.43E-09	52.1944522	2.12E-09	8.68E-07	**1.37E-09**
RW13						
Best	**2.65855917**	**2.65855917**	4.12411617	**2.65855917**	2.90322382	**2.65855917**
Avg	2.66183393	2.82498149	6.86230628	**2.65855917**	2.93373078	**2.65855917**
Worst	2.69949371	3.63593367	26.2009589	**2.65855917**	3.20334904	**2.65855917**
Std	0.01133426	0.36467252	5.90019502	2.09E-11	0.06854164	**4.53E-16**
RW14						
Best	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**
Avg	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**
Worst	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**	**− 30665.539**
Std	6.85E-12	3.71E-12	1.47E-10	1.64E-09	5.77E-11	**3.71E-12**
RW15						
Best	**2.6393465**	**2.6393465**	**2.6393465**	2.63934651	**2.6393465**	**2.6393465**
Avg	**2.6393465**	**2.6393465**	**2.6393465**	2.63934652	**2.6393465**	**2.6393465**
Worst	**2.6393465**	**2.6393465**	**2.6393465**	2.63934652	**2.6393465**	**2.6393465**
Std	1.25E-15	**7.80E-16**	1.06E-14	1.85E-09	9.06E-16	9.06E-16

Variables range

$$12 \leq x_1, x_2, x_3, x_4 \leq 60$$

Table 13 shows the comparison results of the gear train design problem solved using various optimization methods including the proposed *ci*-AVOA. The optimal values are highlighted in bold. The results demonstrate that all algorithms have comparable best, worst, and average fitness values, with similar performance to GA, GWO, and the original AVOA.

The proposed *ci*-AVOA algorithm achieves the best set of design parameters for the tension/compression, pressure vessel, welded beam, and gear train problems in a small

Table 16 Friedman ranks for the 15 real-world constrained optimization problems

	COLSHADE	EnMODE	sCMAgES	SSAS	AVOA	***ci***-AVOA
RW1	**1**	2	5	2	6	2
RW2	**1**	2	5	6	4	3
RW3	**1**	**1**	5	**1**	6	**1**
RW4	**1**	**1**	**1**	**1**	6	**1**
RW5	4	**1**	2	5	6	3
RW6	3	**1**	6	4	5	1
RW7	**1**	2	6	5	3	4
RW8	3	**1**	6	4	5	**1**
RW9	4	2	5	6	3	**1**
RW10	1	2	5	6	3	4
RW11	4	**1**	1	6	3	4
RW12	**1**	**1**	6	5	4	**1**
RW13	3	**1**	6	5	4	**1**
RW14	**1**	**1**	5	4	6	**1**
RW15	**1**	**1**	5	3	6	3
Average rank	2	**1.333**	4.6	4.2	4.667	2.067
Rank	2	**1**	5	4	6	3

number of function evaluations and exhibits superior statistical performance. As a result, the proposed *ci*-AVOA is highly effective in optimizing related problems and offers a promising solution for engineering design problems.

5.5 CEC2020 real-world constrained optimization problems

This section of our research includes tests performed on 15 real-world optimization problems to further explore the performance of the *ci*-AVOA algorithm. We appose the *ci*-AVOA's efficiency with four of the top-performing algorithms from the "CEC2020 Competition on Real-World Single Objective Constrained Optimization" (Kumar et al. 2020). These comprise of the SASS algorithm (Kumar et al. 2020), sCMAgES (Kumar et al. 2020), EnMODE (Sallam et al. 2020), and COLSHADE (Gurrola-Ramos et al. 2020), with the SASS algorithm emerging as the competition's winner.

Table 14 provides a concise overview of the key attributes of the problems and delineates the maximum quota of function evaluations assigned to each problem. A comprehensive review of these problems can be found in Kumar et al. (2020). Both inequality and equality constraints are typical in real-world optimization problems, hence managing these constraints adeptly is vital for efficient optimization. Several constraint handling strategies have been proposed in literature, including Deb's rule (Deb 2000), stochastic ranking (Runarsson and Yao 2000), global competitive ranking (Sarker et al. 2002), adaptive penalty method (Barbosa and Lemonge 2002), multiple ranking (Ho and Shimizu 2007), and self-adaptive penalty strategy (Tessema and Yen 2006).

It is imperative to note that the selection of a constraint management approach is beyond the scope of this study. That being said, the *ci*-AVOA algorithm exclusively

employs the self-adaptive penalty mechanism to deal with constraints. This approach has effectively optimized the *ci*-AVOA algorithm by successfully managing the constraints. Additionally, the four comparison algorithms, namely SASS (Kumar et al. 2020), sCMAgES (Kumar et al. 2020), EnMODE (Sallam et al. 2020), and COLSHADE (Gurrola-Ramos et al. 2020), retain their parameters at the initial values as defined in their respective literature.

Table 15 presents the statistical results of 15 real-world constrained optimization problems. For each problem, the results obtained by six different algorithms: COLSHADE, EnMODE, sCMAgES, SSAS, AVOA, and *ci*-AVOA are compared.

For the first problem, RW1, all six algorithms performed identically in terms of the best, average, and worst results. The standard deviation was smallest for the EnMODE and *ci*-AVOA algorithms. In the second problem, RW2, the best results were achieved by the COLSHADE, EnMODE, and *ci*-AVOA algorithms, but the AVOA algorithm performed significantly worse in terms of average and worst results.

The third problem, RW3, saw all algorithms achieving identical results again, with zero standard deviation except for sCMAgES and SSAS. In RW4, all algorithms achieved identical best, average, and worst results. For problem RW5, *ci*-AVOA had the best result, but the EnMODE algorithm had the best average result.

In the sixth problem, RW6, the best, average, and worst results were identical for COLSHADE, EnMODE, SSAS, and *ci*-AVOA algorithms. In RW7, EnMODE and *ci*-AVOA performed better in terms of best results, with EnMODE also having the smallest standard deviation. For problem RW8, COLSHADE, EnMODE, AVOA, and *ci*-AVOA achieved the best results, with *ci*-AVOA also having the smallest standard deviation.

The ninth problem, RW9, saw the *ci*-AVOA algorithm outperform the others significantly in terms of the best result. For RW10, COLSHADE and EnMODE had identical and best results, with COLSHADE also having the smallest standard deviation.

In problem RW11, COLSHADE had the best and smallest standard deviation results. RW12 saw the best result from the COLSHADE and AVOA algorithms. In RW13, the best result came from the *ci*-AVOA algorithm, with COLSHADE having the smallest standard deviation.

In problem RW14, the best result was achieved by the *ci*-AVOA algorithm, with the smallest standard deviation coming from COLSHADE. Finally, in RW15, the best results were achieved by the COLSHADE and *ci*-AVOA algorithms, with the smallest standard deviation again coming from COLSHADE.

The comprehensive analysis of the optimization problem solutions shows varied performance across the six different algorithms-COLSHADE, EnMODE, sCMAgES, SSAS, AVOA, and *ci*-AVOA. All algorithms had their own strengths and weaknesses, with no single algorithm consistently outperforming the others across all 15 problems. This demonstrates the complex nature of real-world constrained optimization problems, where the optimal algorithm may depend on the specific problem's characteristics.

One particularly noteworthy observation is the performance of the *ci*-AVOA algorithm. Despite not being a dedicated constraint programming algorithm, *ci*-AVOA consistently produced competitive, and in some cases superior, results. This is a remarkable accomplishment as it suggests that the algorithm's design and operation principles are robust and adaptive enough to handle complex, real-world constrained optimization problems effectively.

The performance of *ci*-AVOA was especially striking in problems RW5, RW9, RW13, and RW14, where it achieved the best results. Even in scenarios where *ci*-AVOA did not

achieve the top spot, it was usually amongst the top performers, showing a consistently high level of performance across various problem types.

In addition, *ci*-AVOA was often amongst the algorithms with the smallest standard deviation, indicating that its performance is not only strong on average, but also reliably consistent. This consistency is crucial in real-world applications where predictability of results is often as important as the quality of the best solution.

In conclusion, despite not being a dedicated constraint programming algorithm, *ci*-AVOA has demonstrated its effectiveness in solving a wide variety of constrained optimization problems. This highlights the potential of more generalized optimization algorithms to compete with, and sometimes even outperform, their specialized counterparts. This could pave the way for future research focusing on the development of similarly robust, adaptable algorithms that are capable of tackling a wide array of problem types (Table 16).

6 Pros and Cons of the *ci*-AVOA algorithm

In this section, the strengths and limitations of the novel *ci*-AVOA algorithm are discussed. The *ci*-AVOA algorithm is motivated by the unique foraging behaviors of African vultures and is designed to find optimal solutions by harmoniously balancing exploration and exploitation. The distinctive hierarchical structure, flight patterns, and foraging mechanisms of African vultures are translated into mathematical operations, which ensure an appropriate balance between exploration and exploitation. This balance fosters globally optimal solutions for most of the CEC benchmark functions in dimensions 10, and 20. In addition, *ci*-AVOA has proven to be effective in tackling a wide range of constrained optimization challenges despite not being a specialised constraint programming algorithm. This demonstrates the potential for more generalist optimization algorithms to compete with and occasionally even beat their more specialized equivalents. This may provide the possibility for further investigation into the development of robust and adaptable algorithms that can handle a variety of problem categories.

Nevertheless, despite its merits, the *ci*-AVOA has some limitations. The algorithm may require a relatively longer time to yield efficient solutions, which presents an opportunity for further enhancements to increase its computational efficiency. The introduction of mechanisms to expedite the run time could be beneficial in future research. Some potential strategies include the implementation of chaotic maps and opposition-based learning mechanisms during the initial population generation.

Moreover, while the Choquet Integral enables *ci*-AVOA to effectively address problems with interdependent criteria, the task of determining the fuzzy measures might be challenging in practical applications. It could become computationally intensive, particularly when dealing with a high number of criteria. Therefore, developing efficient strategies for fuzzy measures estimation can be another area for future research.

Hence, while the *ci*-AVOA algorithm demonstrates notable strengths, especially in addressing the interdependence of optimization criteria, future work should consider the aforementioned potential improvements for enhancing its efficiency and usability in real-world applications.

7 Conclusion and future work

In this paper, we have introduced an enhanced variant of the African Vultures Optimization Algorithm (AVOA), named *ci*-AVOA, which strategically incorporates the Choquet fuzzy integral to significantly boost its search and exploitation capacities. Retaining the straightforward and implementable nature of the original AVOA, the *ci*-AVOA marks a significant stride in optimization algorithm research.

One of the pivotal results of this study lies in the exceptional performance of the *ci*-AVOA when evaluated using ten diverse functions from the CEC2020 test suite and applied to four practical engineering design problems. Not only did the *ci*-AVOA demonstrate notable superiority over the original AVOA across various benchmark functions, but it also showcased its capability to adeptly handle an extensive range of problem structures, thus offering more encompassing solutions.

Moreover, the comparative analysis carried out between the *ci*-AVOA and several renowned metaheuristics, including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Salp Swarm Algorithm (SSA), Atomic Orbital Search (AOS), the original AVOA and the improved version of AVOA, Improved African vulture optimization algorithm (IAVOA), Improved hybrid Aquila Optimizer and African Vultures Optimization Algorithm (AOAVOA) and enhanced African Vultures Optimization Algorithm with tent map and time varying mechanism (TAVOA), underscored the competitive edge of the *ci*-AVOA. The algorithm, in many instances, matched or even exceeded the performance of these well-established competitors, particularly in terms of solution accuracy, convergence speed, and robustness across both low and high-dimensional problem domains.

The non-parametric statistical tests further solidified these findings, confirming the superior performance of *ci*-AVOA. It showcased impressive flexibility, resilience, and optimization capabilities, establishing itself as a formidable contender in the field of metaheuristic optimization algorithms.

The primary contributions of this study are twofold. First, we presented the innovative *ci*-AVOA, marking a significant evolution of the original AVOA by successfully incorporating the Choquet fuzzy integral. This integration addresses the multi-criteria nature of many real-world optimization problems, thus enhancing the algorithm's performance significantly.

Second, through a series of comprehensive evaluations and comparisons, we demonstrated the remarkable competitiveness of the *ci*-AVOA in relation to a variety of well-known metaheuristic algorithms. These findings reinforce the relevance and potential of *ci*-AVOA for a wide array of optimization challenges.

As we look ahead, we anticipate extending this research to develop a multi-objective version of *ci*-AVOA, which can address even more complex optimization problems, and to its potential application in other challenging real-world scenarios. Our study thus paves the way for future exploration into more refined and versatile optimization algorithms.

Appendix

See Tables 17, 18.

Table 17 A comparison of the fitness values over 30 experiments obtained by the proposed *ci*-AVOA and other competitor algorithms over CEC2020 test suite with *Dim* = 20

	GA	PSO	GWO	SSA	AOS	AVOA	TAVOA	AOAVOA	IAVOA	*ci*-AVOA
f-1										
Best	122.0545	101.6207	1313.065	100.2056	11883.08	123.2901	139.5242	973.72	4780.506	**100**
Mean	2725.173	1060.533	8.5E+08	2702.644	31353.43	3729.906	2735.379	3.14E+08	13423.32	**100**
Std	2854.728	1269.642	6.87E+08	2666.315	14452.45	3857.378	2095.827	4.04E+08	7384.499	**2.82E-11**
f-2										
Best	1458.894	1854.206	1677.705	2245.902	2572.409	1951.75	1822.484	1911.675	1848.196	**1106.892**
Mean	**1846.93**	2662.126	2417.23	2816.4	3257.75	2782.98	2379.755	2614.698	2535.188	1910.038
Std	278.9054	580.816	439.6799	382.4232	435.6716	391.5335	280.5284	305.1958	261.7808	**258.0391**
f-3										
Best	726.5017	740.4536	736.8086	759.3941	808.2911	804.7039	736.8282	756.7489	756.6253	**716.3309**
Mean	733.7206	762.1418	768.5173	787.5454	896.8916	859.3248	750.9013	780.358	813.449	**723.5698**
Std	**6.000588**	15.50477	19.58282	20.69479	44.09268	34.94083	8.688589	16.23791	22.10361	6.621017
f-4										
Best	1901.934	1901.308	1901.74	1901.781	1904.399	1905.421	1901.452	1902.46	1903.212	**1900.482**
Mean	1903.018	1902.687	1996.012	1904.137	1910.096	1913.034	1902.93	1909.005	1905.665	**1901.22**
Std	0.883093	0.908458	301.4812	1.694562	2.979822	6.332662	1.005398	8.500119	2.141262	**0.386174**
f-5										
Best	35737.03	4621.679	38953.68	7035.183	19442.17	59270.18	10416.91	4661.617	2339.058	**1986.359**
Mean	441932.7	10598.26	459237.3	56909.29	91261.12	533411.4	51328.17	52025.7	9673.837	**2229.852**
Std	294348.4	5732.321	426512	56939.86	57538.91	439528.7	40444.28	68849.06	4982.974	**147.1855**
f-6										
Best	1601.025	1721.416	1632.198	1621.908	1842.574	1636.592	1661.562	1698.142	1731.355	**1600.482**
Mean	**1613.783**	1923.513	1839.053	1894.057	2170.297	1951.884	1798.204	1879.055	1938.769	1720.606
Std	**36.51793**	161.5097	165.0366	162.0955	216.9646	197.137	102.6565	86.38547	104.0658	111.0564
f-7										
Best	12814.12	2464.79	13595.72	4681.346	7432.089	4537.768	10645.3	8970.191	3664.358	**2101.281**
Mean	430170.7	3083.232	198052.2	12427.47	40314.03	306315.3	186716.1	102136.9	17192.78	**2352.704**
Std	248770.8	722.0497	230521.8	6066.045	33411.2	223578.9	120460.9	126527.9	14652.65	**183.6239**

Table 17 (continued)

	GA	PSO	GWO	SSA	AOS	AVOA	TAVOA	AOAVOA	IAVOA	*ci*-AVOA
f-8										
Best	2300.012	2300	2305.77	2300	2301.544	2300	2300.004	2304.748	2301.527	**2239.269**
Mean	2817.661	2689.955	3404.113	2746.269	2303.514	2737.227	2717.799	2896.297	2367.405	**2299.092**
Std	1119.843	954.7467	987.3686	1093.154	**1.246227**	1136.681	729.8931	541.6398	245.7626	14.17817
f-9										
Best	2820.131	2500	2815.579	2818.254	2960.852	2865.154	2831.77	2824.484	2740.363	**2500**
Mean	2865.253	2870.558	2857.23	2842.887	3038.987	2968.167	2876.204	2842.861	2859.653	**2724.115**
Std	22.21779	92.12783	27.84366	18.57209	48.18128	57.00639	26.77678	**14.72359**	55.20307	77.12744
f-10										
Best	2913.958	2902.998	2925.456	2899.354	2903.039	2912.608	2909.191	2924.32	2905.88	**2897.81**
Mean	2979.842	2953.33	2973.571	2935.301	2948.278	2985.457	2956.725	2962.221	2939.542	**2928.578**
Std	29.5533	36.51722	31.98264	29.1534	39.49382	32.47525	**22.34077**	24.20094	22.51194	25.01682

Table 18 Wilcoxon rank sum test between the proposed *ci*-AVOA and other competitors algorithms on CEC2020 test suit functions with *Dim* = 20

***ci*-AVOA vs**	GA		PSO		GWO		SSA		AOS		AVOA		TAVOA		AOAVOA		IAVOA	
Functions	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win	p	Win
F1	6.80E-08	+	1.20E-05	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	0.39527	=	2.43E-05	+	7.12E-09	+
F2	0.20843	=	0.00166	+	0.00056	+	6.79E-08	+	6.79E-08	+	6.79E-08	+	0.09334	=	0.51060	+	0.98231	=
F3	0.00010	+	1.20E-05	+	7.90E-08	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	3.02E-11	+	7.39E-11	+	1.39E-06	+
F4	6.80E-08	+	6.85E-05	+	7.90E-08	+	7.90E-08	+	6.80E-08	+	6.80E-08	+	4.50E-11	+	0.00095	+	2.60E-08	+
F5	6.80E-08	+	1.20E-05	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	3.65E-08	+	7.69E-08	+	1.78E-10	+
F6	0.00051	+	0.01462	+	0.01436	+	0.00025	+	2.22E-07	+	0.00027	+	2.32E-06	+	0.00177	+	0.09049	+
F7	6.80E-08	+	4.71E-05	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	6.80E-08	+	0.37108	=	0.00059	+	3.47E-10	+
F8	0.54277	=	0.12902	=	7.90E-08	+	0.75573	+	0.00092	+	0.27329	=	0.97052	=	0.00195	+	0.00403	+
F9	6.79E-08	+	1.20E-05	+	6.79E-08	+	6.79E-08	+	6.79E-08	+	6.79E-08	+	1.21E-12	+	1.21E-12	+	1.21E-12	+
F10	5.16E-06	+	0.08222	=	0.00062	+	0.10749	=	0.03151	+	1.81E-05	+	1.21E-12	+	1.21E-12	+	1.21E-12	+
Sum	**8+/2=/0−**		**8+/2=/0−**		**10+/0=/0−**		**9+/1=/0−**		**10+/0=/0−**		**9+/1=/0−**		**6+/4=/0−**		**10+/0=/0−**		**9+/1=/0−**	

Author contributions Conception and design of the study: MN, GM, and FF were responsible for the conception and design of the study. Data collection, analysis, and manuscript preparation: MN and GM acquired the data for the study, while MN, GM, and FF analyzed and interpreted the data. MN, GM, and FF drafted the manuscript, and FF and OK revised the manuscript critically for important intellectual content. Approval of the final manuscript: MN, GM, OK, FF, and EL all approved the final version of the manuscript to be published. In summary, MN, GM, and FF played a crucial role in designing the study, collecting and analyzing data, and drafting the manuscript. FF and OK provided critical input during the revision process, while all authors approved the final version of the manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

References

Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158:107408

Abualigah L, Elaziz MA, Sumari P, Khasawneh AM, Alshinwan M, Mirjalili S, Shehab M, Abuaddous HY, Gandomi AH (2022) Black hole algorithm: a comprehensive survey. Appl Intell. https://doi.org/10.1007/s10489-021-02980-5

Alkan N, Kahraman C (2020) Fuzzy metaheuristics: a state-of-the-art review. In: International Conference on Intelligent and Fuzzy Systems, pp. 1447–1455. Springer

Arriola ER, Ubando AT, Chen W-H (2022) A bibliometric review on the application of fuzzy optimization to sustainable energy technologies. Int J Energy Res 46(1):6–27

Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. Knowl-Based Syst 195:105709

Azizi M (2021) Atomic orbital search: a novel metaheuristic algorithm. Appl Math Model 93:657–683

Barbosa HJ, Lemonge AC (2002) An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, pp. 287–294

Bianchi L, Gambardella LM, Dorigo M (2002) Solving the homogeneous probabilistic traveling salesman problem by the aco metaheuristic. International workshop on ant algorithms. Springer, pp 176–187

Boissiere J, Martin F, Teypaz N, Mauris G, Cung V (2007) Using Choquet-integral for guiding Tabu search in multi-criteria public transport network design. IFAC Proc Vol 40(18):617–622

Branke J, Corrente S, Greco S, Słowiński R, Zielniewicz P (2016) Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. Eur J Oper Res 250(3):884–901

Carbas S, Toktas A, Ustun D (eds) (2021) Nature-inspired metaheuristic algorithms for engineering optimization applications. Springer tracts in nature-inspired computing. Springer, Singapore

Chakraborty S, Sharma S, Saha AK, Saha A (2022) A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. Artif Intelli Rev. https://doi.org/10.1007/s10462-021-10114-z

Cheikh-Graiet SB, Dotoli M, Hammadi S (2020) A Tabu search based metaheuristic for dynamic carpooling optimization. Comput Ind Eng 140:106217

Chopard B, Tomassini M, Chopard B, Tomassini M (2018) Performance and limitations of metaheuristics. An Introduction to Metaheuristics for Optimization, 191–203

Choquet G (1954) Theory of capacities. Annales de L'institut Fourier, pp 131–295

Costilla-Enriquez N, Weng Y, Zhang B (2020) Combining Newton-Raphson and stochastic gradient descent for power flow analysis. IEEE Trans Power Syst 36(1):514–517

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):311–338

Dekkers A, Aarts E (1991) Global optimization and simulated annealing. Math Program 50(1):367–393

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Du D, Pardalos PM (1998) Handbook of combinatorial optimization, vol 4. Springer, New York

Fan J, Li Y, Wang T (2021) An improved african vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism. PLoS ONE 16(11):0260725

Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl 30(2):413–435
Ganesan T, Elamvazuthi I (2022) Bilevel optimization of taxing strategies for carbon emissions using fuzzy random matrix generators. Smart Cities and Machine Learning in Urban Health, IGI Global. 210–234
Gharehchopogh FS (2022) An improved Harris hawks optimization algorithm with multi-strategy for community detection in social network. J Bionic Eng. https://doi.org/10.1007/s42235-022-00303-z
Gharehchopogh FS (2023) Quantum-inspired metaheuristic algorithms: comprehensive survey and classification. Artif Intell Rev 56(6):5479–5543
Gharehchopogh FS, Abdollahzadeh B, Arasteh B (2022) An improved farmland fertility algorithm with hyper-heuristic approach for solving travelling salesman problem. Tech Science Press
Gharehchopogh FS, Ucan A, Ibrikci T, Arasteh B, Isik G (2023) Slime mould algorithm: a comprehensive survey of its variants and applications. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-023-09883-3
Gharehchopogh FS, Namazi M, Ebrahimi L, Abdollahzadeh B (2023) Advances in sparrow search algorithm: a comprehensive survey. Arch Comput Methods Eng 30(1):427–455
Grabisch M (1997) K-order additive discrete fuzzy measures and their representation. Fuzzy Sets Syst 92(2):167–189
Greco S, Pavone MF, Talbi E-G, Vigo D (eds) (2021) Metaheuristics for combinatorial optimization. Advances in intelligent systems and computing, vol 1332. Springer
Gurrola-Ramos J, Hernàndez-Aguirre A, Dalmau-Cedeño O (2020) Colshade for real-world single-objective constrained optimization problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE
Ho PY, Shimizu K (2007) Evolutionary constrained optimization using an addition of ranking method and a percentage-based tolerance value adjustment scheme. Inf Sci 177(14):2985–3004
Hussain K, Salleh MNM, Cheng S, Shi Y (2019) On the exploration and exploitation in popular swarm-based metaheuristic algorithms. Neural Comput Appl 31:7665–7683
Khajehzadeh M, Taha MR, El-Shafie A, Eslami M (2011) A survey on meta-heuristic global optimization algorithms. Res J Appl Sci Eng Technol 3(6):569–578
Khalouli S, Ghedjati F, Hamzaoui A (2008) Hybrid approach using ant colony optimization and fuzzy logic to solve multi-criteria hybrid flow shop scheduling problem. In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, pp. 44–50
Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S (2020) A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol Comput 56:100693
Kumar A, Das S, Zelinka I (2020) A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 11–12
Kumar A, Das S, Zelinka I (2020) A self-adaptive spherical search algorithm for real-world constrained optimization problems. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 13–14
Liu R, Wang T, Zhou J, Hao X, Xu Y, Qiu J (2022) Improved African vulture optimization algorithm based on quasi-oppositional differential evolution operator. IEEE Access 10:95197–95218
Mafarja M, Heidari AA, Faris H, Mirjalili S, Aljarah I (2020) Dragonfly algorithm: theory, literature review, and application in feature selection. Nature-inspired optimizers, 47–67
Manita G, Korbaa O (2020) Binary political optimizer for feature selection using gene expression data. Comput Intell Neurosci. https://doi.org/10.1155/2020/8896570
Mirjalili S (2019) Genetic algorithm. Evolutionary algorithms and neural networks. Springer, pp 43–55
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
Mishra D, Naik B (2019) Detecting intrusive behaviors using swarm-based fuzzy clustering approach. In: Nayak J, Abraham A, Krishna BM, Chandra Sekhar GT, Das AK (eds) Soft computing in data analytics. Advances in intelligent systems and computing. Springer, pp 837–846
Mohammadzadeh H, Gharehchopogh FS (2021) A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection. Int J Commun Syst 34(3):4670
Morales-Castañeda B, Zaldivar D, Cuevas E, Fausto F, Rodríguez A (2020) A better balance in metaheuristic algorithms: Does it exist? Swarm Evol Comput 54:100671
Nama S, Sharma S, Saha AK, Gandomi AH (2022) A quantum mutation-based backtracking search algorithm. Artif Intell Rev. https://doi.org/10.1007/s10462-021-10078-0

Naseri TS, Gharehchopogh FS (2022) A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. J Netw Syst Manag 30(3):40
Nayak J, Naik B, Behera HS, Abraham A (2017) Hybrid chemical reaction based metaheuristic with fuzzy c-means algorithm for optimal cluster analysis. Expert Syst Appl 79:282–295
Neuhäuser M (2011). In: Lovric M (ed) Wilcoxon-Mann-Whitney Test. Springer, Heidelberg, pp 1656–1658
Nssibi M, Manita G, Korbaa O (2021) Binary Giza pyramids construction for feature selection. Procedia Comput Sci 192:676–687
Opara KR, Arabas J (2019) Differential evolution: a survey of theoretical analyses. Swarm Evol Comput 44:546–558
Osuna-Enciso V, Cuevas E, Castañeda BM (2022) A diversity metric for population-based metaheuristic algorithms. Inf Sci 586:192–208
Ouertani MW, Manita G, Korbaa O (2022) Improved antlion algorithm for electric vehicle charging station placement. In: 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 265–271. IEEE
Pardalos PM, Romeijn HE, Tuy H (2000) Recent developments and trends in global optimization. J Comput Appl Math 124(1–2):209–228
Hughes M, Goerigk M, Dokka T (2020) Particle swarm metaheuristics for robust optimisation with implementation uncertainty, Elsevier. Comput Operat Res 122:104998
Pozna C, Precup R-E, Horváth E, Petriu EM (2022) Hybrid particle filter-particle swarm optimization algorithm and application to fuzzy controlled servo systems. IEEE Trans Fuzzy Syst 30(10):4286–4297
Ramadas M, Abraham A (2019) Metaheuristics for data clustering and image segmentation. Intelligent systems reference library. Springer
Rani D, Garg H (2022) Multiple attributes group decision-making based on trigonometric operators, particle swarm optimization and complex intuitionistic fuzzy values. Artif Intell Rev 56(2):1787–1831
Razmjooy N, Ashourian M, Foroozandeh Z (eds.) (2021) Metaheuristics and Optimization in Computer and Electrical Engineering. Lecture Notes in Electrical Engineering, vol. 696. Springer, Cham
Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294
Sahoo SK, Saha AK, Nama S, Masdari M (2023) An improved moth flame optimization algorithm based on modified dynamic opposite learning strategy. Artif Intell Rev 56(4):2811–2869
Sahoo SK, Sharma S, Saha AK (2023) A novel variant of moth flame optimizer for higher dimensional optimization problems. J Bionic Eng. https://doi.org/10.1007/s42235-023-00357-7
Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ (2020) Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE
Samadi Bonab M, Ghaffari A, Soleimanian Gharehchopogh F, Alemi P (2020) A wrapper-based feature selection for improving performance of intrusion detection systems. Int J Commun Syst 33(12):4434
Sarker R, Mohammadian M, Yao X, Runarsson T, Yao X (2002) Constrained evolutionary optimization: The penalty function approach. Evolutionary Optimization, 87–113
Sharma S, Chakraborty S, Saha AK, Nama S, Sahoo SK (2022) mlboa: a modified butterfly optimization algorithm with LaGrange interpolation for global optimization. J Bionic Eng 19(4):1161–1176
Sharma S, Saha AK, Roy S, Mirjalili S, Nama S (2022) A mixed sine cosine butterfly optimization algorithm for global optimization and its application. Clust Comput 25(6):4573–4600
Shishavan ST, Gharehchopogh FS (2022) An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks. Multimed Tools Appl 81(18):25205–25231
Sugeno M (1974) Theory of fuzzy integrals and its applications. Doct. Thesis, Tokyo Institute of technology
Tessema B, Yen GG (2006) A self adaptive penalty function based algorithm for constrained optimization. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 246–253. IEEE
Venter G, Sobieszczanski-Sobieski J (2003) Particle swarm optimization. AIAA J 41(8):1583–1589
Wang S, Zhao Q (2022) Probabilistic Tabu search algorithm for container liner shipping problem with speed optimisation. Int J Prod Res 60(12):3651–3668
Wang Y, Zhang Y, Zhang C, Zhou J, Hu D, Yi F, Fan Z, Zeng T (2023) Genetic algorithm-based fuzzy optimization of energy management strategy for fuel cell vehicles considering driving cycles recognition. Energy 263:126112
Xiao Y, Guo Y, Cui H, Wang Y, Li J, Zhang Y (2022) Ihaoavoa: an improved hybrid Aquila optimizer and African vultures optimization algorithm for global optimization problems. Math Biosci Eng 19(11):10963–11017
Yang X-S (2012) Flower pollination algorithm for global optimization. In: International Conference on Unconventional Computing and Natural Computation, pp. 240–249 . Springer

Zaman HRR, Gharehchopogh FS (2022) An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. Eng Comput 38(Suppl 4):2797–2831
Zhang J, Ma Z (2020) Hybrid fuzzy clustering method based on fcm and enhanced logarithmical pso (elpso). Comput Intell Neurosci. https://doi.org/10.1155/2020/1386839

Authors and Affiliations

Maha Nssibi[1,2] · Ghaith Manita[1,3] · Francis Faux[4] · Ouajdi Korbaa[5] · Elyes Lamine[6]

✉ Maha Nssibi
maha.nssibi@ensi-uma.tn

Ghaith Manita
gaith.manita@esen.tn

Francis Faux
francis.faux@univ-jfc.fr

Ouajdi Korbaa
Ouajdi.Korbaa@centraliens-lille.org

Elyes Lamine
elyes.lamine@mines-albi.fr

1 Laboratory MARS, LR17ES05, ISITCom, University of Sousse, Sousse, Tunisia

2 ENSI, University of Manouba, Manouba, Tunisia

3 ESEN, University of Manouba, Manouba, Tunisia

4 IRIT, CNRS, University of Toulouse, Toulouse, France

5 ISITCom, University of Sousse, Sousse, Tunisia

6 Industrial Engineering Center, IMT Mines Albi, 81000 Albi, France