Nelder-Mead Enhanced Gazelle Optimizer for Solving Complex Optimization Problems

Beytullah Yağız [1], Şeyma Nur Atar [2], Erdal Eker [3], Serdar Ekinci [4], Davut Izci [5,*]
[1] Department of Statistics, Van Yuzuncu Yil University, Van, Turkey
[2,3] Vocational School of Social Sciences, Muş Alparslan University, Muş, Turkey
[4] Department of Computer Engineering, Bitlis Eren University, Bitlis, Turkey
[5] Department of Electrical and Electronics Engineering, Bursa Uludag University, Bursa, Turkey
Email: [1] beytmat@gmail.com, [2] seymatar.04@gmail.com, [3] erdal.eker@alparslan.edu.tr, [4] ekinciser@yahoo.com, [5] davutizci@gmail.com
*Corresponding Author

***Abstract*—This paper presents the improved gazelle optimization algorithm, which is a new approach in the field of metaheuristic optimization algorithms inspired by nature. By hybridizing the classical gazelle optimization algorithm with the Nelder-Mead simplex method, the improved gazelle optimization algorithm was developed. The proposed IGOA algorithm aims to combine GOA's global search capability with NM's local healing power to provide a more balanced and effective optimization of optimization problems. The performance of the algorithm was evaluated by 30 independent runs on the CEC2017 benchmark functions. The statistical results obtained from the analyses of the mean, standard deviation, best and worst values and Wilcoxon signed ranks test show that IGOA exhibits a superior or competitive performance compared to other current optimization algorithms. Furthermore, the boxplot and convergence curves revealed that IGOA exhibited stable convergence behavior and had a low tendency to get stuck at local optimums. Big-O analysis, on the other hand, confirmed that the algorithm can scale efficiently even in high-dimensional problems. The results prove that the IGOA algorithm is a highly competitive, effective and generalizable tool in solving complex optimization problems.**

Keywords*—Optimization, Metaheuristic Algorithm, Gazelle Optimization Algorithm, Nelder-Mead Simplex Method, Statistical Analysis

I. INTRODUCTION

Stochastic methodologies inspired by natural processes have garnered significant attention in recent times. These optimization techniques typically emulate the social or individual behaviors of animal populations or other natural phenomena. Initially, a random set of candidate solutions is generated based on machine learning principles, and algorithms are developed employing metaheuristic strategies that facilitate multifaceted searches [1].

Metaheuristic algorithms exhibit a convergent approach towards optimal solutions, offering effective means to address nondeterministic polynomial problems, which are notably challenging to resolve. In the context of problem optimization, metaheuristic methods achieve convergence to the global solution by navigating the search space through two distinct phases: exploration and exploitation. The exploration phase enhances the diversity of solutions, whereas the exploitation phase concentrates on a specific area to yield optimal local solutions. A limitation of this approach is the algorithm's potential inability to escape local optima. To mitigate this limitation, metaheuristic algorithms employ various solution methods and can effectively utilize different strategies due to their flexible structure [2]. Swarm-based algorithms endeavor to balance exploitation and exploration, a critical aspect for effectively investigating promising regions within the search space and approaching the global optimal solution [3].

The study of metaheuristic algorithms has garnered significant attention in recent scholarly literature, and this paper explores several notable examples. The genetic algorithm (GA) is an adaptation and machine learning methodology rooted in genetics and natural selection. By combining structured yet randomized information exchange between artificial chromosomes with the Darwinian principle of 'survival of the fittest,' it demonstrates a robust search intuition [4]. The particle swarm optimization algorithm (PSO) is positioned between genetic algorithms and evolutionary programming, utilizing stochastic processes. PSO enhances potential solutions by 'flying' them through multidimensional space, facilitating a balanced exploration between known optimal regions and unexplored areas [5]. The ant colony optimization algorithm (ACO) draws inspiration from ant foraging behavior, where ants mark favorable pathways by releasing pheromones for other colony members. This mechanism is employed by ACO to address optimization problems [6]. The artificial bee colony algorithm (ABC) is an optimization algorithm modeled on honeybee swarm behavior. The artificial bee colony consists of three groups: working bees, scout bees, and exploratory bees. The bee waiting in the dance area to select food sources is termed the scout bee, the bee revisiting previous food sources is the working bee, and the bee searching randomly is the exploratory bee. Half of the colony comprises working bees, while the other half consists of follower bees. Each food source is attended by one working bee, corresponding to the number of food sources. When a food source is depleted, the dependent bee becomes an exploratory bee [7]. The grey wolf optimization algorithm (GWO) is inspired by the behavior of gray wolves, mimicking their leadership hierarchy and hunting mechanisms. Four types of grey wolves simulate the hierarchy: alpha, beta, delta, and omega. The algorithm implements the three fundamental steps of hunting: searching for prey, encircling prey, and capturing prey by attacking [8]. Research has also been conducted on Harris hawks, focusing on their pack intelligence. The Harris hawks optimization algorithm (HHO) is based on their cooperative hunting behavior and "surprise attack" technique, where multiple hawks attack from different directions to surprise prey. Harris hawks exhibit various pursuit patterns depending on prey escape behavior and scenario dynamics [9]. In animal clusters, the primary motive is hunger. The hunger games search optimization algorithm (HGS) is inspired by this motive. HGS is an optimization method based on animals' hunger-driven behaviors, utilizing hunger as an adaptive weight at each step, thereby rendering the search process dynamic. HGS, with its simple structure and high performance, yields more efficient results than existing optimization methods [10].

While metaheuristic algorithms offer advantages, they present challenges that must be addressed. A primary limitation is their slow convergence rate, which often results in local optima entrapment, particularly in complex, multi-modal problems. This tendency can lead to unpredictable outcomes. The efficacy of these algorithms depends on precise tuning of parameters, such as crossover, mutation, cooling schedule, and initial temperature. These parameters significantly influence algorithm performance and convergence. Numerous parameter settings can complicate implementation and burden users. Furthermore, some methods work only in continuous search spaces, requiring complex coding schemes, while local search capabilities may remain inadequate. The use of equations and complex operators can increase computational cost, resulting in longer execution times [11]. Therefore, meticulous parameter configuration is crucial in applying metaheuristic algorithms, along with developing strategies for specific problem structures. The no free lunch (NFL) theorem becomes relevant, indicating that metaheuristic algorithms cannot universally produce optimal solutions and should be integrated with various local or global algorithms. The NFL theorem states that no universally superior algorithm exists in optimization and machine learning. It emphasizes selecting algorithms pragmatically, choosing the most suitable one based on problem characteristics. The theorem stresses the importance of experimenting with diverse algorithms across different problems and establishes a theoretical boundary on algorithm performance. The mathematical expression of the NFL theorem is presented in (1) [12].

For any two α_1 and α_2 algorithms, selected from a set of f functions and a set of α algorithms as shown in (1). In (1), the expression per is the optimal value that the algorithm finds for the specified function f, $\langle . \rangle_f$ means that all functions are averaged. In this context, algorithms have been developed using many hybrid methods.

$$\langle per(succes|\alpha_1, f)\rangle_f = \langle per(succes|\alpha_2, f)\rangle_f \tag{1}$$

For example, opposed-based learning (OBL) is a technique designed to enhance the efficiency of metaheuristic optimization algorithms. OBL addresses candidate solutions generated through stochastic iteration, as well as opposing solutions situated in opposite regions, which are frequently closer to the global optimum than random solutions. The foundation of OBL lies in concepts such as predictions and counter-predictions, weights and contrasting weights, and actions and counteractions. The potential to extend existing learning algorithms has been substantiated by examples [13]. An OBL-based manta ray foraging optimization (MRFO) algorithm was developed to optimize well-known test functions, with statistical results demonstrating superior performance compared to similar algorithms [14]. The pattern search (PS) method seeks to identify optimal solutions for nonlinear and unconstrained problems without requiring derivative knowledge. It integrates optimization techniques that provide global convergence theory without necessitating sufficient reduction conditions. PS constructs a "mesh" around a specified point in the solution space, which is updated as the process advances, achieving global convergence with relaxed step acceptance conditions [15]. In a study, the improved hunger games search (Imp-HGS) algorithm was developed to address deficiencies in the classical hunger games search (HGS) algorithm. Imp-HGS enhances search efficiency by integrating pattern search and elite opposition-based learning. When tested on various functions and engineering problems, it achieved lower error rates compared to existing methods, demonstrating effectiveness for complex optimization problems [16]. Another strategy is the random walks (RW) approach. One study proposes random walk dandelion optimization (RW-DO), hybridized with a random walk-based local search strategy, to overcome the tendency of the dandelion optimization algorithm (DO) to become trapped at local optima. The random walk strategy disrupts stable progression by interfering with the subsequent iteration, thereby preventing premature convergence to local optima. This facilitates the algorithm's escape from local optima and increases the likelihood of reaching superior global solutions. During the exploration phase, the deviation effect diminishes in subsequent iterations, resulting in more stable outcomes. The primary advantage of RW-DO is its ability to overcome the classical DO algorithm's challenges in reaching the global optimum, achieving a better balance between exploration and exploitation. The study evaluated RW-DO's performance in various classical engineering problems, with results indicating that the algorithm excelled in both exploration and exploitation phases, producing solutions closer to the global optimum [17].

The Nelder-Mead simplex method (NM) is a prominent direct search technique designed to identify the minimum value of multivariate nonlinear functions by utilizing only function values, without requiring derivative information. This method involves calculating test points and their corresponding function values, followed by the creation of a new simplex. Iterations proceed until a reduction in function values is achieved. The Nelder-Mead algorithm is distinguished by its ability to generate a new simplex with merely one or two function evaluations per iteration, rendering it more cost-effective than other direct search methods [18]. In a particular study, the arithmetic optimization algorithm was enhanced through the integration of opposition-based learning and the Nelder-Mead simplex search method, resulting in the development of the ObAOANM algorithm. Opposition-based learning augmented the algorithm's discovery capability, while the Nelder-Mead method enhanced its exploitation capacity. The algorithm was assessed using test functions with known unimodal and multimodal, demonstrating superior performance compared to the original arithmetic optimization algorithm. When applied to the optimal design of the proportional-integral-differential (PID) controller in automobile cruise system, the ObAOANM algorithm exhibited greater success than existing methods in both statistical and dynamic analyses. Time-domain analyses further confirmed the method's superior performance [19]. Given that the development of the gazelle optimization algorithm (GOA) is facilitated by the NM strategy, this strategy will be elaborated upon in subsequent sections.

GOA is a highly effective metaheuristic approach for problem optimization [20]. This algorithm draws inspiration from the behaviors of gazelles as they evade predators. Gazelles possess distinctive characteristics that aid in their escape, with predators achieving only a 34% success rate in capturing them. Within the GOA framework, gazelles demonstrate survival and foraging behaviors to identify optimal solutions within the solution space. In the absence of predators, gazelles engage in random grazing over extensive areas, analogous to the algorithm's exploration phase, which involves investigating diverse solution regions. Conversely, when predators are present, gazelles seek the nearest refuge, mirroring the exploitation phase where solutions converge towards the optimal outcome. These phases are iteratively executed until the termination criteria are satisfied. GOA serves as an effective global stochastic optimizer, characterized by its straightforward yet potent search capabilities. Its dual-phase structure—unrestricted grazing and seeking shelter when threatened—facilitates both the explore of novel solutions and the refinement of promising solutions. Nevertheless, GOA exhibits certain limitations. In some unimodal functions, it may prematurely converge and deviate from the global optimum, becoming ensnared in local minima. The algorithm's efficacy is also contingent upon appropriate parameter settings, such as population size and iteration count. Overall, GOA is a high-performance, stable optimization algorithm that adeptly balances exploration and exploitation, although it may occasionally converge prematurely [20]. Subsequent sections will evaluate hybrid studies involving GOA, accompanied by a comprehensive literature review.

The primary objective of this study is to conduct a comprehensive evaluation of the features of the improved gazelle optimization algorithm (IGOA). This algorithm represents an enhancement of the gazelle optimization algorithm (GOA) through the integration of the Nelder-Mead simplex method (NM) strategy. This paper aims to elucidate the performance of IGOA in comparison with other contemporary optimization algorithms documented in the literature. To achieve this, the efficacy of IGOA across various problem types was assessed using a range of statistical analyses and technical methodologies.

During the evaluation of the IGOA algorithm, a series of independent experiments were conducted using the CEC 2017 benchmark functions to objectively assess the algorithm's performance. The CEC 2017 benchmark function set has been selected due to its comprehensive array of functions. The diversity within this set facilitates a multifaceted evaluation of the algorithm, thereby effectively demonstrating the robustness and stability of the proposed IGOA hybrid algorithm. The results were analyzed employing fundamental statistical measures, including mean, best, worst, and standard deviation. Additionally, significance analyses, such as the Wilcoxon signed-rank test, were utilized to compare the algorithm with other contemporary methods. The algorithm's stability and solution diversity were illustrated through boxplot graphs, while its speed and stability in reaching the optimum were assessed using convergence curves. The scalability and computational efficiency of the algorithm were demonstrated through Big-O analysis. The findings indicate that the IGOA algorithm yields more stable, reliable, and high-performance solutions compared to both the classical GOA and other prevalent optimization algorithms. These results underscore that IGOA provides a competitive and generalizable approach to optimization problems.

The remainder of this paper is structured as follows. Section 2 introduces the theoretical foundations of metaheuristic optimization algorithms, with a particular focus on the classical GOA and its mathematical formulation. Section 3 presents the proposed GOA (IGOA), developed by hybridizing GOA with the NM simplex method, and details its underlying mechanism and pseudocode. Section 4 describes the experimental design, including benchmark functions, performance indicators, and statistical methods employed for evaluation. Section 5 discusses the obtained results through comparative analyses against other well-known optimizers, supported by boxplots, convergence curves, and complexity assessments. Finally, Section 6 concludes the paper by summarizing the main contributions, highlighting the advantages and limitations of the proposed IGOA, and outlining potential directions for future research.

A. Previous Works on GOA

In research on GOA, a hybrid SCA-GOA (HSCAGOA) has been introduced. This algorithm integrates GOA exploitation strategy to address challenges. Inspired by gazelle behavior, GOA enhances local search capabilities, improving exploration and exploitation balance. The incorporation of Brownian motion and Lévy flight mechanisms augments its reconnaissance capabilities [21]. While GOA effectively resolves various optimization problems, it can become trapped in local optima during complex scenarios. This study proposes a hybrid gazelle optimization algorithm and differential evolution (HGOADE), combining GOA with differential evolution (DE). HGOADE generates potential solutions using GOA, refines them through DE's mutation and crossover steps, and further optimizes through GOA's phases. This integration enhances search performance by combining DE's extensive search with GOA's dynamic capabilities [22]. An efficient optimization method is crucial for solving complex problems. To enhance GOA's performance, an improved GOA (IGOA) incorporating orthogonal learning and Rosenbrock's direct rotation strategy is proposed [23]. The hybrid GOA-DE algorithm integrates DE's processes into GOA's iteration to improve solution optimization. Testing on engineering design problems shows the hybrid algorithm achieves better results than traditional methods [24]. Similar studies exist in literature [25]. GOA variants with adaptive strategy, Lévy flight, roulette wheel selection, and random walk have been developed to enhance its effectiveness across optimization scenarios [26].

GOA is recognized for its application in real-world problems. GOA has been employed in fractional-order proportional–integral–derivative (FOPID) controllers for micro-DC motors, showing superior performance and precision compared to other optimization techniques [27]. The adaptive chaotic dynamic gazelle optimization algorithm (ACD-GOA) has been developed for feature selection problems, incorporating strategies to enhance search capability and convergence speed. A dynamic oppositional learning strategy mitigates premature convergence, while adaptive inertial weight and sigmoid function improve search efficiency. Population diversity is maintained through elite and information exchange strategies to avoid local optima [28]. The fractional-order nonlinear autoregressive exogenous (F-NARX) model represents real-time processes, with parameter estimation being critical. GOA has been applied to estimate F-NARX system parameters, incorporating a Grünwald–Letnikov derivative into the classical nonlinear autoregressive model. A mean-squared error-based conformity function evaluated GOA's performance, focusing on convergence, accuracy, complexity, and robustness. GOA's flexibility has been tested in estimating complex parameters in the electrically stimulated muscle model (ESMM) for paralyzed muscle rehabilitation [29]. Cervical cancer remains prevalent among women, with Pap smear tests crucial for early diagnosis. This study presents computer aided cervical cancer diagnosis utilizing the gazelle optimizer algorithm with deep learning (CACCD-GOADL), combining deep learning and GOA to detect cervical cancer in Pap smear images. The model uses enhanced MobileNetv3 for feature extraction, with GOA optimizing hyperparameters. The stacked overlearning machine method distinguishes between cancerous and healthy cells, with successful results on the Herlev dataset [30]. The GOA algorithm is highly regarded for its robust global search capabilities, yet it encounters challenges such as premature convergence and a tendency to become trapped in local minimum during local searches. These issues are exacerbated by the absence of enhancement parameters or operators to address weaknesses in local search areas, as well as a high sensitivity to hyperparameters like swarm size and iteration count. These challenges have led to the development of the IGOA algorithm [20], [21],[26]. Although hybrid studies based on GOA [21]-[26] have made some progress in addressing these problems, the Nelder–Mead (NM) simplex method effectively preserves the global exploration strength of the GOA. It enhances the exploitative power of NM by utilizing its derivative-free local optimization capability around the best individual every 50 iterations. Consequently, this method reduces the likelihood of early convergence, improves convergence stability, and supports scalability in high-dimensional spaces. The statistical analyses in the article underscore the superiority and competitiveness of the proposed IGOA compared to other algorithms.

II. Algorithms

A. Mathematical Model of GOA

A swarm should be established randomly by defining its lower and upper boundaries. The initial phase commences with the selection of candidate populations of gazelles (X), as illustrated in (2).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & x_{i,j} & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (2)$$

In (3), the swarm shows its random formation with certain measures. Here $x_{i,j}$, indicates the position of each candidate.

$$x_{i,j} = rand \times (UB_j - LB_j) + LB_j \quad (3)$$

In this context, $rand$ refers to randomly selected numbers, while UB_j and LB_j denote the upper and lower bounds of the problem, respectively. In the majority of iterations, the best solution identified thus far is considered the optimal solution. During the matrix formation process, the gazelle with the highest potential is selected. If, throughout all iterations, the gazelle with the highest potential supersedes the previously best-performing gazelle, the optimal solution, referred to as $Elite$ is updated accordingly.

$$Elite = \begin{bmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,d-1} & x'_{1,d} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,d-1} & x'_{2,d} \\ \vdots & \vdots & x'_{i,j} & \vdots & \vdots \\ x'_{n,1} & x'_{n,2} & \cdots & x'_{n,d-1} & x'_{n,d} \end{bmatrix} \quad (4)$$

In (4), $x'_{i,j}$ denotes the upper gazelle vector utilized in the construction of the $Elite$ matrix. In this context, the concept of hunting is applied, with the hunter serving as the search agent. This approach is predicated on the observation that the hunters pursue the gazelles as they flee to a safe zone. Consequently, the hunters who pursue the escaping gazelles will proceed to the search area. The $Elite$ is updated in each iteration.

$$f_B(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right) \quad (5)$$

The Lévy flight strategy, referred to in (5), produces random marches. Here, μ is the arithmetic mean, σ is the standard deviation, and x is the best position, and $1 < \alpha \leq 2$. Lévy states the stable process in terms of integrals.

$$L(x_j) \approx |x_j|^{1-\alpha} \quad (6)$$

$$f_L(x;\alpha,\gamma) = \frac{1}{\pi}\int_0^\infty exp(-\gamma q^\alpha)\cos(qx)\,\delta q \quad (7)$$

In this part, the scale properties of the motion α are controlled by the dispersion index. At (8), the scale unit is denoted by y. In the Levy strategy, the α parameter takes values in the range [0.3, 1.99].

$$Lévy(\alpha) = 0.05 \times \frac{x}{|y|^{\frac{1}{\alpha}}} \quad (8)$$

$\sigma_y = 1$ ve $\alpha = 1.5$, and α, x, and y expressed in equation (9) to equation (11).

$$x = Normal(0, \sigma_x^2) \tag{9}$$

$$y = Normal(0, \sigma_y^2) \tag{10}$$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\frac{\pi\alpha}{2})}{\Gamma\left(\frac{(1+\alpha)}{2}\right)\alpha 2^{\frac{(\alpha-1)}{2}}} \right]^{\frac{1}{\alpha}} \tag{11}$$

The GOA algorithm emulates the behavior of gazelles as they evade predators and seek refuge. It comprises two stages, with the exploitation phase simulating the grazing behavior of gazelles.

$$\overrightarrow{gazelle_{\iota+1}} = \overrightarrow{gazelle_\iota} + \ s *. \vec{R} *. \overrightarrow{R_B} *. (\overrightarrow{Elite_\iota} - \overrightarrow{R_B} *. \overrightarrow{gazelle_\iota}) \tag{12}$$

In (12), $\overrightarrow{gazelle\iota + 1}$ is the solution of the next iteration, $\overrightarrow{gazelle\iota}$ is the solution at the current iteration, s denotes the grazing speed of the gazelles, $\overrightarrow{R_B}$ is a vector containing random numbers representing the Brownian motion. R is a vector of uniform random numbers in $[0,1]$. The reflexes of gazelles against the attack form the reconnaissance phase. It $R, Lévy$ denotes a random number vector based on its distributions expressed in (13) and (14). CF is a control parameter that limits the predator in the equation.

$$\overrightarrow{gazelle_{\iota+1}} = \overrightarrow{gazelle_\iota} + S.\mu.\vec{R} *. \overrightarrow{R_L} *. (\overrightarrow{Elite_\iota} - \overrightarrow{R_B} *. \overrightarrow{gazelle_\iota}) \tag{13}$$

$$\overrightarrow{gazelle_{\iota+1}} = \overrightarrow{gazelle_\iota} + S *. \mu. CF *. \overrightarrow{R_B} *. (\overrightarrow{Elite_\iota} - \overrightarrow{R_L} *. \overrightarrow{gazelle_\iota}) \tag{14}$$

Where $CF = (1 - \frac{iter}{Max_{iter}})^{(2\frac{iter}{Max_{iter}})}$. The general formula of the GOA algorithm is given in (15).

$$\overrightarrow{gazelle_{\iota+1}} = \begin{cases} \overrightarrow{gazelle_\iota} + CF[\overrightarrow{LB} + \vec{R} *. (\overrightarrow{UB} - \overrightarrow{LB}) *. \vec{U} & if\ r \le PSRs \\ \overrightarrow{gazelle_\iota} + [PSRs(1-r) + r](\overrightarrow{gazelle_{r1}} - \overrightarrow{gazelle_{r2}}) & else \end{cases} \tag{15}$$

B. Nelder Mead Method

One of the most used methods to strengthen the solution of optimization problems and to optimize different problems is the Nelder-Mead simplex search method (shown in Fig. 1) [18]. This method initially outlines a simplex in d dimensions, comprising $d+1$ vertices $x_1, x_2, \ldots x_d, x_{d+1}$, and then progressively creates a series of simplices to approximate an optimal point for an objective function $(f(x))$. In each iteration, the vertices of the simplex are rearranged to establish a relationship of $f(x_1) \le f(x_2) \le \cdots \le f(x_{d+1})$, where the best and worst vertices are denoted by x_1 and x_{d+1},respectively. To substitute the worst vertex, four scalar coefficients—reflection, ρ, expansion, γ, contraction, β, and shrinkage—are employed. Additionally, a centroid point $\bar{x}$ is determined given in (16).

$$\bar{x} = \frac{1}{d}\sum_{i=1}^{d} x_i \tag{16}$$

Following the definitions mentioned earlier, the connection of $f(x_1) \le f(x_2) \ldots \le f(x_{d+1})$ is assessed by examining $f(x)$ and organizing the simplex as part of the algorithm's initial step. Once this is completed, the reflection point, x_r, is determined in the following manner, and the reflection phase is illustrated in (17).

$$x_r = \bar{x} + \rho(\bar{x} - x_{d+1}) \tag{17}$$

The next step involves assessing $f(x_r)$, where x_{d+1} is substituted with x_r for $f(x_1) \le f(x_r) < f(x_d)$. The expansion point, x_e, calculated and $f(x_e)$ is assessed in the next step for $f(x_r) < f(x_1)$ using (18).

$$x_e = \bar{x} + \gamma(x_r - \bar{x}) \tag{18}$$

For the condition of $f(x_e) < f(x_r)$, x_{d+1} has been changed x_e, otherwise, x_r replaces x_{d+1}. Then, for $f(x_d) \le f(x_r) < f(x_{d+1})$, the outsite contraction point, x_c, is calculated and $f(x_c)$ is evaluated as in (19).

$$x_c = \bar{x} + \beta(x_r - \bar{x}) \tag{19}$$

The second step substitutes x_{d+1} with x_{oc} for $f(x_{oc}) \le f(x_r)$. In other cases, a shrinkage step is executed. An internal contraction point, x'_i, is determined in the following manner, for $f(x_{d+1}) \le f(x_r)$, and $f(x'_i)$ is assessed.

$$x'_i = \bar{x} + \beta(x_{d+1} - \bar{x}) \tag{20}$$

The step described above substitutes x_{d+1} with x'_i for $f(x'_i) < f(x_{d+1})$. If this is not the case, a shrinkage step is executed. This shrinkage step, which is the final step in the NM method, generates new points by contracting them according to the following definition.

$$v_i = x_1 + \delta(x_i - x_1), i = 2, 3, \ldots, d+1 \tag{21}$$

C. Proposed Hybrid IGOA Algorithm

The default IGOA algorithm represents a hybrid optimization approach, integrating GOA with the NM method. As is detailed in the pseudocode given in Algorithm I, the algorithm executes the classical GOA, updating agent positions. By emulating gazelles evading predators and seeking refuge, the GOA identifies optimal individuals within the population and generates new solutions through these selections. A key feature of IGOA is its periodic activation of the Nelder-Mead method. The algorithm employs the NM method every 50 iterations to establish a simplex around the best solution, facilitating local enhancements.

Algorithm I. Pseudocode of IGOA

Input: Population size, Maximum iterations, Bounds [lb, ub], Problem dimension, Objective function
Output: Optimal solution, Optimal fitness, Convergence history
Initialization
- *Set parameters (PSRs=0.34, S=0.88, s=random)*
- *Initialize optimal solution and fitness*
- *Generate gazelle population within bounds*

Main Process (while iterations < max_iterations)
- *Evaluate Population*
  - *Calculate fitness for gazelles*
  - *Update best gazelle if superior*
- *Update Position*
  - *Create elite matrix from best gazelle*
  - *Calculate control factor: CF*
  - *Generate Levy and Brownian vectors*
- *Move Gazelles*
  - *For each gazelle:*
    - *If random > 0.5: Exploit*
    - *Else: Explore (global search with Levy flights)*
- *Apply Enhancement*
  - *Re-evaluate gazelles*
  - *Update optimal solution if improved*
  - *Nelder-Mead Enhancement (every 50 iterations)*
    - *Apply local optimization*
    - *Refine solution*
  - *Switch Positions*
    - *Update random positions for diversity*
- *Return optimal solution, fitness, convergence curve*

End

The NM simplex method was assessed over 10, 20, and 30 iterations within the algorithm, demonstrating improvements; however, the most pronounced results were observed after 50 iterations. This was particularly evident in the optimization of hyperparameters during these trials. The NM method is a robust optimization technique that operates without derivatives, aiming to identify the optimal value of the objective function. This approach combines the global search capability of GOA with the local search capability of NM. The hybrid structure enables efficient exploration of the solution space while refining promising solutions. The integration of Nelder-Mead enhances algorithm performance without substantial computational burden. The IGOA algorithm thus combines GOA's exploration capability with NM's local refinement for efficient optimization. Table 1 lists the CEC2017 benchmark functions considered in this study and Table 2 provides the statistical results obtained from those functions. Analyzing the performance metrics in Table 2 reveals that the IGOA algorithm consistently ranks first across all functions. This suggests that the IGOA algorithm is highly responsive, effective, and suitable for generalization. Additionally, the rank value of 2 for the GOA algorithm indicates that the hybridized main algorithm is also a strong performer.

The Wilcoxon signed-ranks test serves as a non-parametric alternative to the paired t-test. It evaluates the differences in performance between two classifiers by ranking both positive and negative differences. Unlike the t-test, the Wilcoxon test is more cautious, as it only requires qualitative comparability of differences. Statistically, it is considered safer because it does not rely on the assumption of normal distribution and is less influenced by outliers compared to the t-test. Although the t-test is more powerful when its assumptions hold true, the Wilcoxon test can be more effective when those assumptions are not met. A typical nonparametric method involves counting how frequently an algorithm performs better, worse, or the same as others, either in pairwise comparisons or by tallying the datasets where it surpasses all others. According to the null hypothesis of equivalent algorithms, each should win in N/2 out of N datasets, with ties evenly distributed between classifiers [31].

The findings in this section are derived from the mean, standard deviation, best/worst values, and Wilcoxon signed-rank test conducted over 30 independent runs on 30 functions from CEC 2017. The IGOA exhibits superior average performance and reduced variance across all functions, particularly excelling in functions F1, F4–F10, and F12–F30 when compared to the GOA and other existing algorithms. The Wilcoxon test results largely favor IGOA, except in cases where it shares identical values with functions F3 and F25 and shows a relative disadvantage against GOA in F21 and the MFO in F22. Box plots indicate that IGOA's medians surpass those of its competitors, suggesting that its solutions are stable and less affected by outliers. Even with extreme values noted in F22, F23, F26, and F28, IGOA's central tendency measurements remain optimal. Convergence curves illustrate that IGOA achieves rapid recovery in the initial phases, consistently progressing towards the optimum without becoming trapped in local minima. Furthermore, the Big-O analysis confirms the method's scalability in high dimensions, as the number of transactions increases approximately linearly with the problem size.

Upon examination of Table 3, it is evident that the algorithm demonstrates lower performance compared to GOA in the F11 function and MFO in the F22 function. However, it exhibits equivalent performance to GOA in the F3 and F25 functions. Notably, the algorithm surpasses other algorithms in a total of 83 pairwise comparisons. These findings provide empirical evidence that the IGOA algorithm outperforms other algorithms, establishing it as a highly competitive and effective algorithm. A boxplot serves as a tool to reveal the general patterns hidden within a dataset and is especially useful for showcasing the characteristics of a large dataset. Boxplots employ components like the median, interquartile range (IQR), and whiskers to represent the central tendency and distribution of the data [32].

When Fig. 2 is examined, it is seen that the IGOA algorithm produces the closest values in functions 22, 23, 26 and 28, but also has extreme values. In these functions, compared to other algorithms, the best image still belongs to the IGOA algorithm. The boxplot produced by the IGOA algorithm in all functions shows that the values of this algorithm are optimally close to each other, stable and effective.

Convergence curves show a system's performance on a task relative to resources used. In some cases, a resource budget is preset, while in others, the goal is achieving results with minimal resources. These budgets include examples reviewed or time allocated in an environment. The performance metric indicates model efficacy. Learning curves are crucial for machine learning decisions: Data acquisition helps determine needed data points for desired performance, enabling future performance predictions. For a specific learner, focus may be on reducing training time or mitigating overfitting [33].

Upon examination of Fig. 3, it can be observed that the IGOA algorithm consistently avoids becoming trapped in local optima, demonstrates steady convergence, and is not susceptible to premature convergence across all functions. This outcome is evident from the convergence curves, which illustrate the algorithm's ability to achieve optimal results efficiently by utilizing the maximum number of iterations.

Big-O analysis (given in Fig. 4) serves as a crucial theoretical framework for evaluating an algorithm's performance with large data sets and its resource efficiency. It assesses the algorithm's behavior in addressing the problem, independent of the programming language employed. Specifically, it provides an upper bound on the runtime and memory usage that an algorithm will require for the most challenging problems it aims to optimize. This analysis is particularly essential in the development of large-scale systems and performance-critical applications [34].

$$Big-O = \max\_iter * N * D \tag{22}$$

where $\max\_iter$ is maximum number of iterations, N stands for the swarm size, D expresses the size of the problem. The Big-O analysis was conducted using the IGOA algorithm on five distinct functions selected from the CEC 2017 set. These functions include unimodal (F1, F5), multimodal (F7, F10), and composite (F20) functions. The x-axis represents the increase in the number of variables of the function, extending up to 1000 variables at specified intervals, while the y-axis denotes the fundamental operations of the algorithm during this process. The observation that all functions exhibit a nearly overlapping and linear increase indicates that as the problem size expands, the number of operations increases proportionally, signifying that they all scale at an equivalent rate. The linear complexity graph suggests that even at substantial sizes, these functions can be scaled efficiently. This analysis demonstrates that the scalability of the selected algorithms and functions is high.

When assessing the tables and figures related to the periodic IGOA Nelder–Mead local search strategy, these components are generated through the GOA convergence phase of the initial local search. This approach aims to tackle the difficulties of balancing exploration and exploitation, thereby offering protection. Some functions, which show relatively weak or comparable results, indicate that the method provides absolute stability and general superiority across the problem class. However, a significant statistical advantage is observed in favor of the total IGOA, as evidenced by the majority of tests. As a result, IGOA, as evaluated in CEC 2017, delivers competitive or superior performance across all function variations. It maintains computational efficiency while ensuring convergence stability on a linear scale, thus presenting a viable solution for large-dimensional problems.

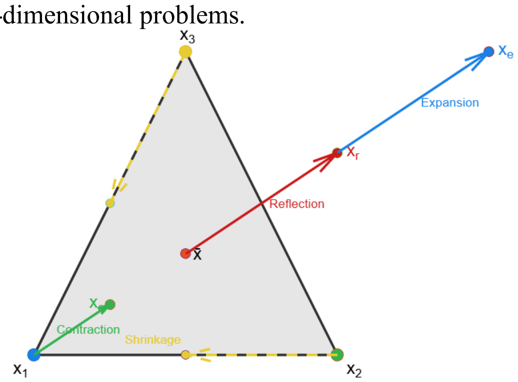


Fig. 1. Nelder-mead process

Table 1. CEC 2017 benchmark function set

Function	Name of the function	Class	Optimum
F1	Shifted and Rotated Bent Cigar Function	Unimodal	100
F3	Shifted and Rotated Zakharov Function	Unimodal	300
F4	Shifted and Rotated Rosenbrock's Function	Multimodal	400
F5	Shifted and Rotated Rastrigin's Function	Multimodal	500
F6	Shifted and Rotated Expanded Schaffer's F6 Function	Multimodal	600
F7	Shifted and Rotated Lunacek Bi-Rastrigin Function	Multimodal	700
F8	Shifted and Rotated Noncontinuous Rastrigin's Function	Multimodal	800
F9	Shifted and Rotated Levy Function	Multimodal	900
F10	Shifted and Rotated Schwefel's Function	Multimodal	1000
F11	Hybrit Function 1 (N = 3)	Hybrid	1100
F12	Hybrit Function 2 (N = 3)	Hybrid	1200
F13	Hybrit Function 3 (N = 3)	Hybrid	1300
F14	Hybrit Function 4 (N = 4)	Hybrid	1400
F15	Hybrit Function5 (N = 4)	Hybrid	1500
F16	Hybrit Function6 (N = 4)	Hybrid	1600
F17	Hybrit Function 7 (N = 5)	Hybrid	1700
F18	Hybrit Function 8 (N = 5)	Hybrid	1800
F19	Hybrit Function 9 (N = 5)	Hybrid	1900
F20	Hybrit Function 10 (N = 6)	Hybrid	2000
F21	Composition Function 1 (N = 3)	Composition	2100
F22	Composition Function 2 (N = 3)	Composition	2200
F23	Composition Function 3 (N = 4)	Composition	2300
F24	Composition Function 4 (N = 4)	Composition	2400
F25	Composition Function 5 (N = 5)	Composition	2500
F26	Composition Function 6 (N = 5)	Composition	2600
F27	Composition Function 7 (N = 6)	Composition	2700
F28	Composition Function 8 (N = 6)	Composition	2800
F29	Composition Function 9 (N = 3)	Composition	2900
F30	Composition Function 10 (N = 3)	Composition	3000

Table 2. Results of algorithm analysis via CEC 2017

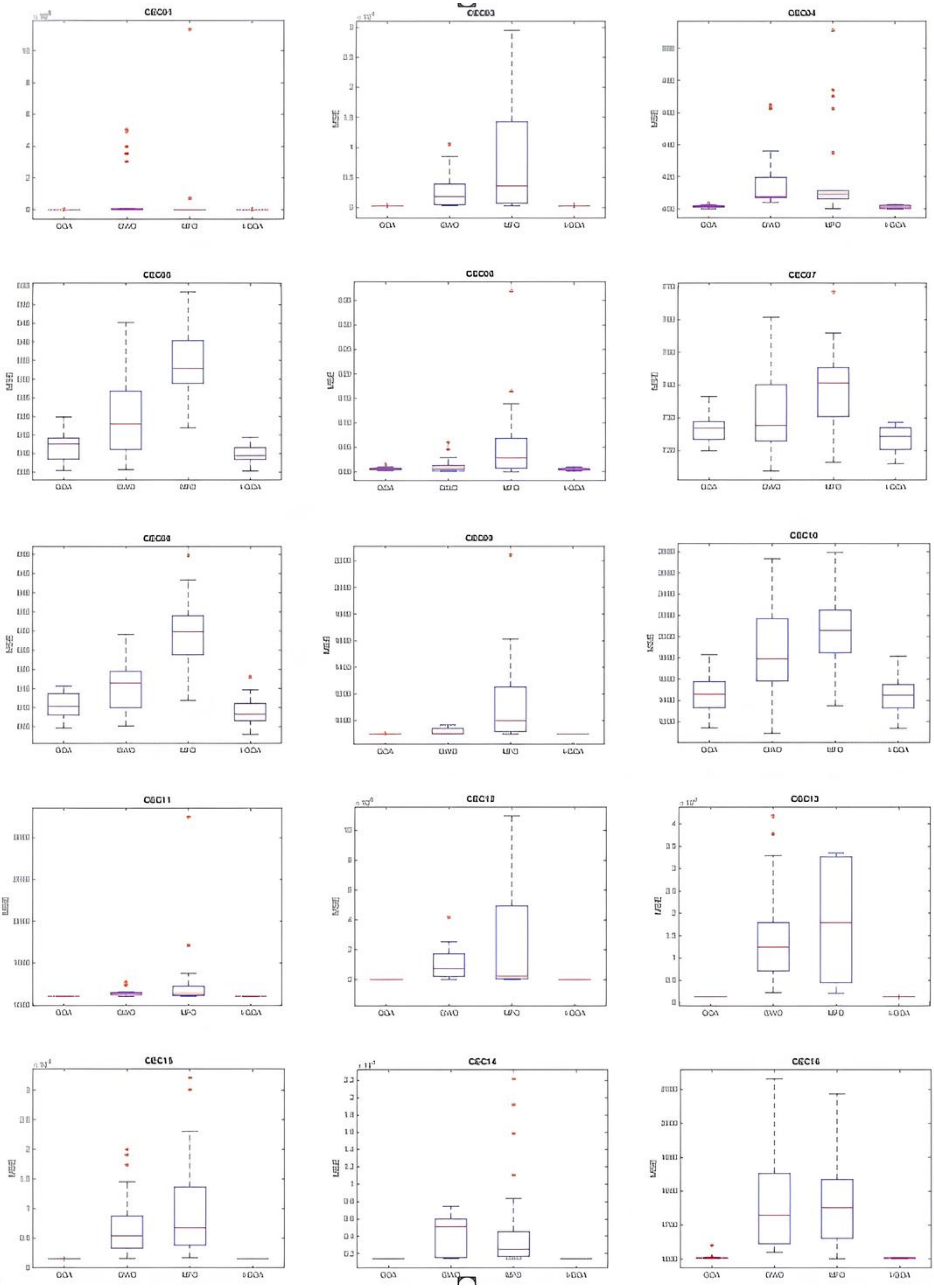
Function 2017	Metric	GOA	IGOA	GWO	MFO
F1	Mean	2.3642E+03	1.3926E+03	8.1322E+07	4.2512E+07
	Std	1.7953E+03	9.0671E+2	1.6666E+08	2.0711E+08
	Best	1.0073E+03	3.6968E+2	2.4790E+04	1.0119E+02
	Worst	8.1981E+03	4.0581E+03	5.0591E+08	1.1351E+09
	Rank	2	1	4	3
F3	Mean	3.0003E+02	3.0003E+02	2.6204E+03	8.0185E+03
	Std	1.8123E-02	2.2868E-02	2.5172E+03	8.9271E+03
	Best	3.0001E+02	3.0000E+02	3.5069E+02	3.0132E+02
	Worst	3.0010E+02	3.0009E+02	1.0567E+04	2.9518E+04
	Rank	2	1	2	3
F4	Mean	4.0232E+02	4.0142E+02	4.1677E+02	4.1987E+02
	Std	5.9441E-01	9.1641E-01	1.8035E+01	2.6345E+01
	Best	4.0166E+02	4.0007E+02	4.0407E+02	4.0038E+02
	Worst	4.0374E+02	4.0281E+02	4.6505E+02	5.1144E+02
	Rank	2	1	3	4
F5	Mean	5.1189E+02	5.0991E+02	5.1995E+02	5.3429E+02
	Std	3.8528E+00	2.3559E+00	1.0366E+01	1.0454E+01
	Best	5.0549E+02	5.0539E+02	5.0571E+02	5.1691E+02
	Worst	5.1985E+02	5.1432E+02	5.4514E+02	5.5342E+02
	Rank	2	1	3	4
F6	Mean	6.0064E+02	6.0050E+02	6.0117E+02	6.0530E+02
	Std	1.5902E-01	2.0558E-01	1.3607E+00	7.4495E+00
	Best	6.0046E+02	6.0011E+02	6.0014E+02	6.0001E+02
	Worst	6.0113E+02	6.0103E+02	6.0598E+02	6.3695E+02
	Rank	2	1	3	4
F7	Mean	7.2622E+02	7.2380E+02	7.3132E+02	7.3844E+02
	Std	3.7083E+00	3.6049E+00	1.1262E+01	1.0888E+01
	Best	7.1991E+02	7.1602E+02	7.1380E+02	7.1644E+02
	Worst	7.3652E+02	7.2858E+02	7.6069E+02	7.6847E+02
	Rank	2	1	3	4
F8	Mean	8.1203E+02	8.0903E+02	8.1574E+02	8.2907E+02
	Std	1.9070E+00	3.6932E+00	6.4654E+00	8.4852E+00
	Best	8.0928E+02	8.0303E+02	8.0524E+02	8.1194E+02
	Worst	8.1563E+02	8.1797E+02	8.2907E+02	8.4975E+02
	Rank	2	1	3	4
F9	Mean	9.0012E+02	9.0008E+02	9.1739E+02	1.1392E+03
	Std	5.8202E-02	3.7147E-02	2.5285E+01	3.0975E+02
	Best	9.0005E+02	9.0001E+02	9.0006E+02	9.0000E+02
	Worst	9.0030E+02	9.0016E+02	9.6901E+02	2.2426E+03
	Rank	2	1	3	4
F10	Mean	1.5710E+03	1.4599E+03	1.8347E+03	2.0349E+03
	Std	1.2499E+02	1.6280E+02	3.8551E+02	3.2150E+02
	Best	1.4280E+03	1.1368E+03	1.0888E+03	1.3507E+03
	Worst	1.8163E+03	1.8163E+03	2.7348E+03	2.7915E+03
	Rank	2	1	3	4
F11	Mean	1.1057E+03	1.1047E+03	1.1444E+03	1.2568E+03
	Std	1.0605E+00	1.6212E+00	4.0162E+01	3.9590E+02
	Best	1.1044E+03	1.1008E+03	1.1000E+03	1.1026E+03
	Worst	1.1089E+03	1.1068E+03	1.2710E+03	3.2470E+03
	Rank	2	1	3	4
F12	Mean	1.6199E+03	1.4873E+03	1.0179E+06	2.0951E+06
	Std	7.1517E+01	7.6236E+01	9.9908E+05	3.0314E+06
	Best	1.5174E+03	1.2906E+03	1.2053E+04	7.3601E+03
	Worst	1.7640E+03	1.5962E+03	4.1644E+06	1.0955E+07
	Rank	2	1	3	4
F13	Mean	1.3190E+03	1.3161E+03	1.4386E+04	1.8920E+04
	Std	2.3302E+00	4.1192E+00	9.9641E+03	1.2438E+04
	Best	1.3160E+03	1.3081E+03	2.2456E+03	2.1063E+03
	Worst	1.3262E+03	1.3264E+03	4.1759E+04	3.3531E+04
	Rank	2	1	3	4
F14	Mean	1.4214E+03	1.4183E+03	4.1208E+03	4.7405E+03
	Std	1.9471E+00	5.3277E+00	2.1731E+03	5.3845E+03
	Best	1.4178E+03	1.4077E+03	1.4714E+03	1.4424E+03
	Worst	1.4253E+03	1.4265E+03	7.4376E+03	2.2185E+04
	Rank	2	1	3	4
F15	Mean	1.5047E+03	1.5033E+03	6.9340E+03	9.5666E+03

	Std	9.0979E-01	8.6924E-01	5.1520E+03	8.3334E+03
	Best	1.5034E+03	1.5018E+03	1.5643E+03	1.6501E+03
	Worst	1.5066E+03	1.5046E+03	1.9968E+04	3.2080E+04
	Rank	2	1	3	4
F16	Mean	1.6067E+03	1.6035E+03	1.7542E+03	1.7766E+03
	Std	3.2806E+00	8.6777E-01	1.3796E+02	1.2971E+02
	Best	1.6042E+03	1.6019E+03	1.6202E+03	1.6012E+03
	Worst	1.6175E+03	1.6049E+03	2.1311E+03	2.0865E+03
	Rank	2	1	3	4
F17	Mean	1.7281E+03	1.7221E+03	1.7622E+03	1.7820E+03
	Std	2.8286E+00	5.8293E+00	2.7250E+01	5.3377E+01
	Best	1.7249E+03	1.7069E+03	1.7204E+03	1.7259E+03
	Worst	1.7382E+03	1.7268E+03	1.8558E+03	1.9798E+03
	Rank	2	1	3	4
F18	Mean	1.8198E+03	1.8146E+03	2.8002E+04	2.7926E+04
	Std	2.3768E+00	3.5315E+00	1.4182E+04	1.5330E+04
	Best	1.8159E+03	1.8075E+03	7.4517E+03	4.1697E+03
	Worst	1.8234E+03	1.8204E+03	5.4658E+04	5.5330E+04
	Rank	2	1	4	3
F19	Mean	1.9042E+03	1.9037E+03	1.8498E+04	2.3220E+04
	Std	4.5311E-01	8.2534E-01	4.4828E+04	2.7722E+04
	Best	1.9034E+03	1.9017E+03	1.9346E+03	2.0335E+03
	Worst	1.9052E+03	1.9053E+03	2.5282E+05	1.1120E+05
	Rank	2	1	3	4
F20	Mean	2.0243E+03	2.0202E+03	2.1158E+03	2.0929E+03
	Std	2.1788E+00	5.2057E+00	6.8920E+01	6.7477E+01
	Best	2.0212E+03	2.0055E+03	2.0272E+03	2.0050E+03
	Worst	2.0308E+03	2.0252E+03	2.2814E+03	2.2812E+03
	Rank	2	1	4	3
F21	Mean	2.2379E+03	2.2188E+03	2.3075E+03	2.3082E+03
	Std	5.4044E+01	4.2131E+01	3.5884E+01	5.7963E+01
	Best	2.2002E+03	2.2000E+03	2.2008E+03	2.2000E+03
	Worst	2.3189E+03	2.3137E+03	2.3394E+03	2.3736E+03
	Rank	2	1	3	4
F22	Mean	2.3043E+03	2.2875E+03	2.3840E+03	2.3088E+03
	Std	7.0074E-01	3.4632E+01	2.2783E+02	4.0236E+01
	Best	2.3033E+03	2.2007E+03	2.3016E+03	2.2260E+03
	Worst	2.3059E+03	2.3050E+03	3.1653E+03	2.4642E+03
	Rank	2	1	4	3
F23	Mean	2.6169E+03	2.6102E+03	2.6247E+03	2.6296E+03
	Std	2.6386E+00	1.8878E+01	1.1410E+01	1.1118E+01
	Best	2.6137E+03	2.5126E+03	2.6085E+03	2.6126E+03
	Worst	2.6223E+03	2.6207E+03	2.6587E+03	2.6567E+03
	Rank	2	1	4	3
F24	Mean	2.6135E+03	2.5354E+03	2.7478E+03	2.7633E+03
	Std	1.1601E+02	6.3530E+01	1.3255E+01	9.5009E+00
	Best	2.4268E+03	2.4337E+03	2.7286E+03	2.7458E+03
	Worst	2.7565E+03	2.7354E+03	2.7770E+03	2.7844E+03
	Rank	2	1	3	4
F25	Mean	2.9010E+03	2.8979E+03	2.9377E+03	2.9487E+03
	Std	1.1663E+01	2.9778E-01	1.4873E+01	3.4602E+01
	Best	2.8978E+03	2.8977E+03	2.9003E+03	2.8977E+03
	Worst	2.9439E+03	2.8989E+03	2.9510E+03	3.0379E+03
	Rank	2	1	3	4
F26	Mean	2.8849E+03	2.8338E+03	3.2079E+03	3.0817E+03
	Std	3.1593E+01	1.0768E+02	4.0798E+02	2.5410E+02
	Best	2.8141E+03	2.6047E+03	2.6090E+03	2.8000E+03
	Worst	2.9006E+03	2.9004E+03	3.9318E+03	3.9994E+03
	Rank	2	1	4	3
F27	Mean	3.0902E+03	3.0900E+03	3.1041E+03	3.0960E+03
	Std	3.1897E-01	4.9542E-01	2.3830E+01	4.7660E+00
	Best	3.0897E+03	3.0891E+03	3.0905E+03	3.0903E+03
	Worst	3.0909E+03	3.0910E+03	3.1975E+03	3.1128E+03
	Rank	2	1	4	3
F28	Mean	3.1354E+03	3.1031E+03	3.4027E+03	3.3187E+03
	Std	9.1220E+01	7.6552E+01	6.3401E+01	9.1203E+01
	Best	3.1014E+03	2.8301E+03	3.2169E+03	3.1703E+03
	Worst	3.4118E+03	3.4118E+03	3.5027E+03	3.4121E+03

	Rank	2	1	4	3
F29	Mean	3.1601E+03	3.1512E+03	3.2030E+03	3.2249E+03
	Std	1.0301E+01	7.4703E+00	4.8362E+01	5.1750E+01
	Best	3.1434E+03	3.1386E+03	3.1412E+03	3.1418E+03
	Worst	3.1875E+03	3.1636E+03	3.3063E+03	3.3588E+03
	Rank	2	1	3	4
F30	Mean	3.9447E+03	3.7001E+03	9.0919E+05	7.5989E+05
	Std	6.0351E+02	3.5507E+02	1.0872E+06	8.6294E+05
	Best	3.4469E+03	3.4565E+03	4.0432E+03	8.1657E+03
	Worst	5.8737E+03	5.2439E+03	4.4340E+06	3.3916E+06
	Rank	2	1	4	3

Table 3. Comparison of IGOA with other algorithms via "Wilcoxon signed rank test"

Function 2017	Metric	GOA	GWO	MFO
F1	p-value	2.8486E-02	1.7344E-06	2.5967E-05
	W/T/L	W	W	W
F3	p-value	8.1302E-01	1.7344E-06	1.7344E-06
	W/T/L	T	W	W
F4	p-value	7.1570E-04	1.7344E-06	2.6033E-06
	W/T/L	W	W	W
F5	p-value	0.0132E+00	6.3198E-05	1.7344E-06
	W/T/L	W	W	W
F6	p-value	1.5658E-02	0.0316E+00	2.3704E-05
	W/T/L	W	W	W
F7	p-value	0.0196E+00	0.0093E+00	6.3391E-06
	W/T/L	W	W	W
F8	p-value	2.2551E-03	2.6134E-04	1.7344E-06
	W/T/L	W	W	W
F9	p-value	4.9498E-02	2.3534E-06	1.9209E-06
	W/T/L	W	W	W
F10	p-value	7.5746E-03	4.4493E-05	2.8786E-06
	W/T/L	W	W	W
F11	p-value	8.7297E-03	2.1266E-06	2.8786E-06
	W/T/L	W	W	W
F12	p-value	2.3534E-06	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F13	p-value	6.8359E-03	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F14	p-value	6.0350E-03	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F15	p-value	1.0246E-05	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F16	p-value	1.7344E-06	1.7344E-06	1.9209E-06
	W/T/L	W	W	W
F17	p-value	2.8434E-05	1.7344E-06	1.9209E-06
	W/T/L	W	W	W
F18	p-value	1.7988E-05	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F19	p-value	2.4308E-02	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F20	p-value	3.8811E-04	1.7344E-06	3.5152E-06
	W/T/L	W	W	W
F21	p-value	5.1931E-02	1.9209E-06	5.7517E-06
	W/T/L	L	W	W
F22	p-value	8.7297E-03	1.4936E-05	0.0571E+00
	W/T/L	W	W	L
F23	p-value	2.4147E-03	4.4493E-05	3.8822E-06
	W/T/L	W	W	W
F24	p-value	0.0036E+00	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F25	p-value	0.1915E+00	1.7344E-06	2.6033E-06
	W/T/L	T	W	W
F26	p-value	0.0068E+00	7.6909E-06	3.1817E-06
	W/T/L	W	W	W
F27	p-value	3.3269E-02	1.7344E-06	1.7344E-06
	W/T/L	W	W	W
F28	p-value	9.2710E-03	1.7344E-06	1.9209E-06
	W/T/L	W	W	W
F29	p-value	6.1564E-04	3.5152E-06	5.7517E-06
	W/T/L	W	W	W
F30	p-value	0.0449E+00	1.7344E-06	1.7344E-06
	W/T/L	W	W	W



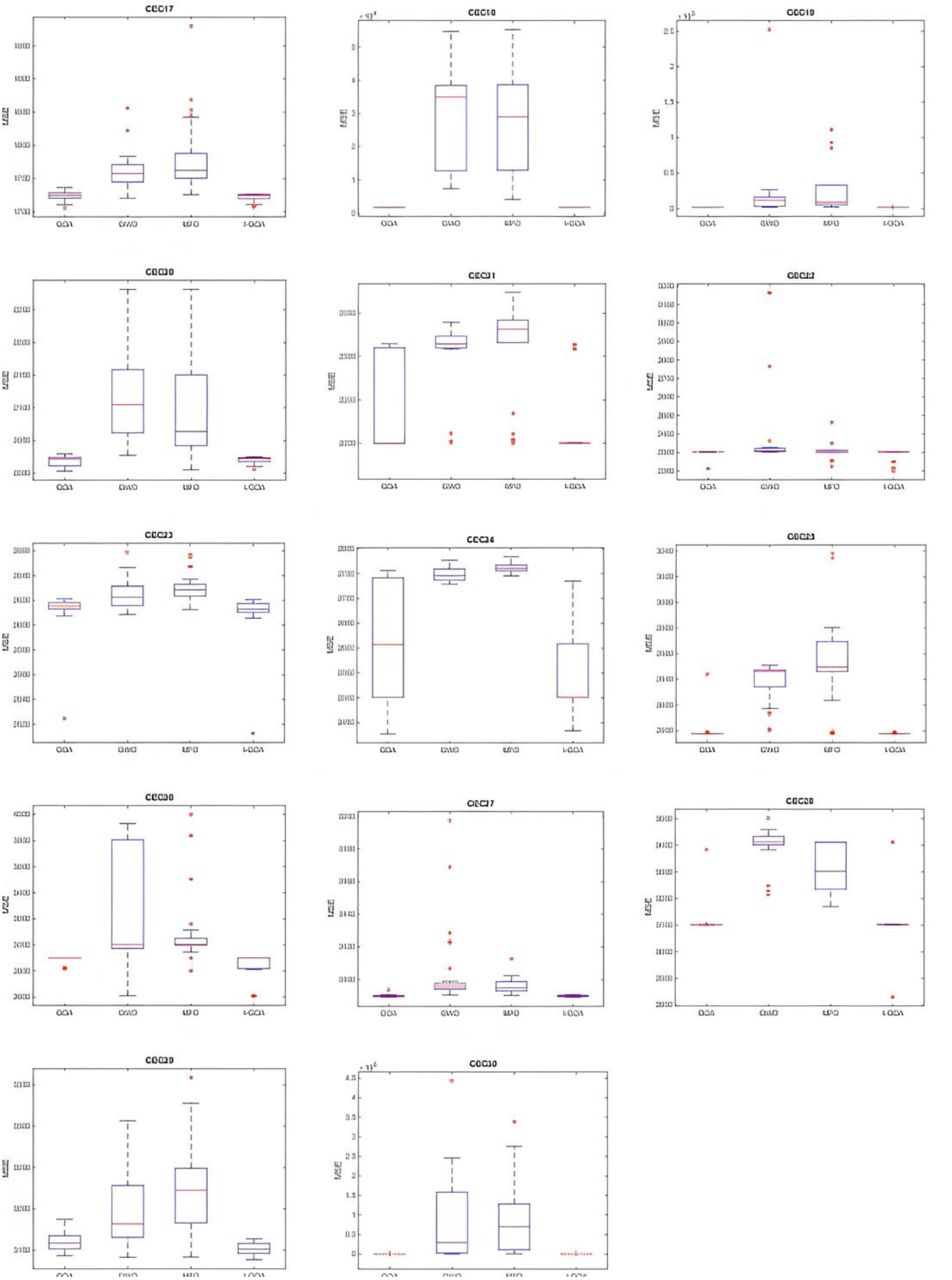
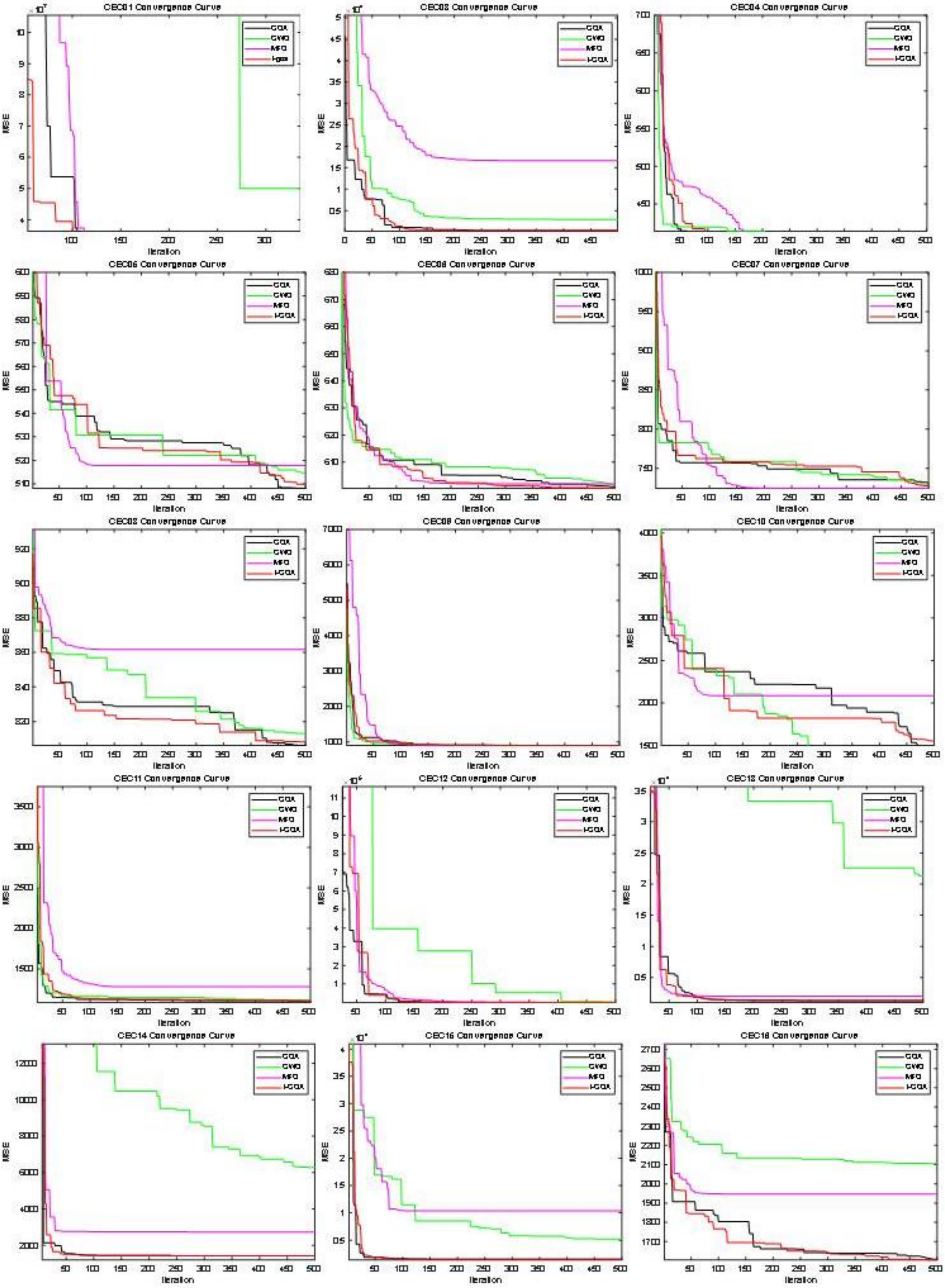


Fig. 2. Representation of CEC 2017 results of algorithms with boxplot figures



CEC01 Convergence Curve
CEC02 Convergence Curve
CEC04 Convergence Curve
CEC05 Convergence Curve
CEC06 Convergence Curve
CEC07 Convergence Curve
CEC08 Convergence Curve
CEC09 Convergence Curve
CEC10 Convergence Curve
CEC11 Convergence Curve
CEC12 Convergence Curve
CEC13 Convergence Curve
CEC14 Convergence Curve
CEC16 Convergence Curve
CEC18 Convergence Curve
GOA
GWO
MFO
I-GOA
MSE
Iteration

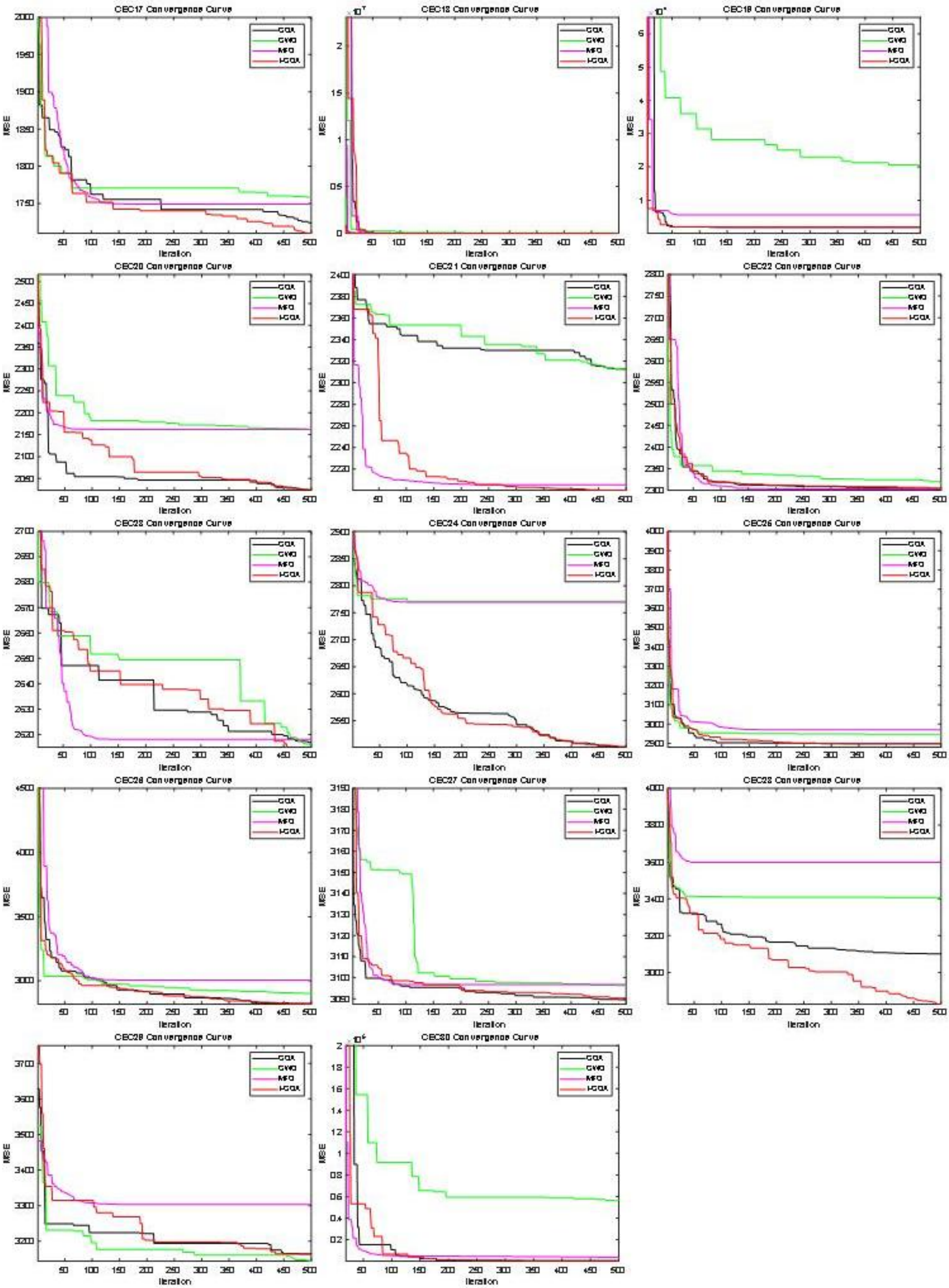


Fig. 3. Representation of CEC 2017 results of algorithms with converge curve

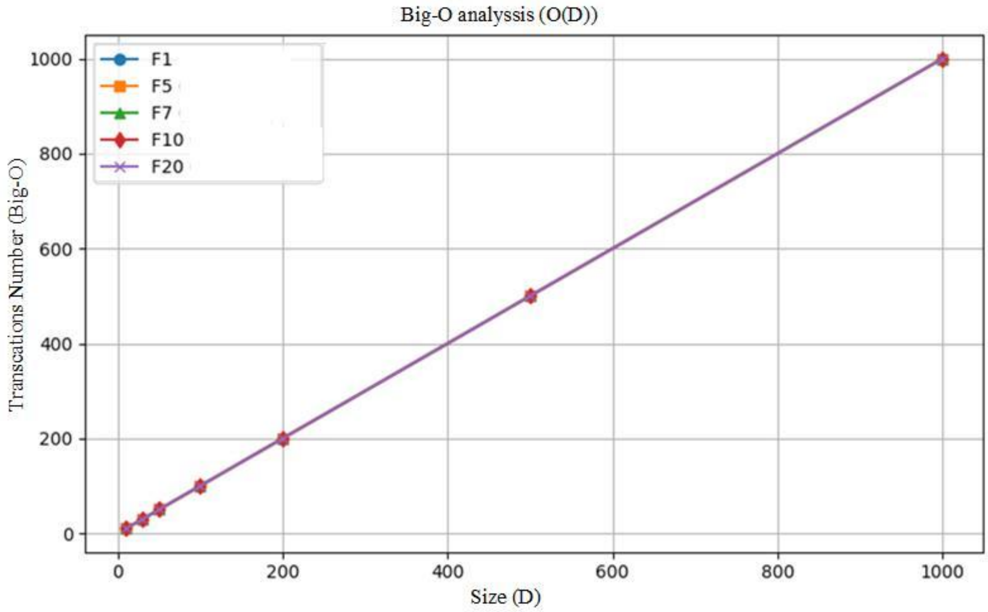


Fig. 4. Big-O analyze

III. Conclusion

The IGOA algorithm presented constitutes an innovative optimization strategy integrating global and local search capabilities by hybridizing classical GOA with the NM method. Its scientific contribution to nature-inspired metaheuristic algorithms is evident in achieving a robust equilibrium between exploration and exploitation while yielding superior outcomes. The amalgamation of NM's local optimization prowess with GOA's global search capability has enabled the algorithm to excel in complex functions with multiple peaks and real-world engineering challenges. The statistical performance of IGOA was rigorously evaluated through 30 independent runs on CEC 2017 benchmark functions. Findings indicate that in terms of mean, best, worst, and standard deviation values, IGOA consistently ranks first across most functions compared to classical GOA and other contemporary algorithms. Comparisons using the Wilcoxon signed-rank test demonstrated that IGOA outperformed competitors in most of 87 pairwise comparisons. Boxplot analyses reveal that solutions generated by IGOA are stable, consistent, and proximate to optimal values, while convergence curves show the algorithm steadily reaches the optimum without premature convergence. Big-O analyses indicate that IGOA possesses linear complexity scalability and can efficiently handle large-scale problems. The algorithm's strengths include effective global and local optimizations due to its hybrid structure, relative insensitivity to parameter settings, and high generalizable success across problem types. Its production of statistically significant and consistent results enhances reliability in practical applications. Despite all its strengths, the IGOA algorithm also has its limitations. Unimodal functions may occasionally lead to early convergence and deviation from the global optimum. The algorithm's performance depends on appropriate selection of key parameters such as population size and iterations; incorrect settings can result in performance loss. The hybrid nature's computational overhead can increase runtime for very large-scale problems. Furthermore, success requires careful tuning of additional parameters, such as Nelder-Mead's application frequency and simplex size.

Future research endeavors may consider hybridizing the IGOA with various chaotic maps, adaptive parameter mechanisms, or alternative methods. In the context of dynamic or multi-objective optimization problems, evaluating the algorithm's performance and its integration into real-time applications constitutes a significant area of investigation. Additionally, in fields such as deep learning, big data analytics, and industrial process optimization, employing IGOA for hyperparameter optimization or feature selection could enhance its practical utility. The IGOA algorithm makes a substantial contribution to metaheuristic optimization, demonstrating high competitiveness and generalizability both theoretically and practically. With its hybrid structure, superior performance, and extensive application potential, IGOA emerges as an innovative optimizer for future optimization studies. The IGOA algorithm showcases remarkable optimization capabilities across various emerging sectors. In the realms of artificial intelligence and machine learning, it enhances fine-tuning through global search and Nelder-Mead local optimization, especially for high-dimensional problems like network hyperparameter searches. It provides adaptive solutions for traffic signal timing and route planning in smart cities. Moreover, it optimizes efficiency-cost trade-offs in energy management and microgrid distribution. The algorithm reliably converges in healthcare applications, including drug discovery and medical imaging. Additionally, it offers flexible optimization for scheduling, robotic planning, and quality control in Industry 4.0 manufacturing.

References

[1] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and L. Abualigah, "Binary aquila optimizer for selecting effective features from medical data: a covid-19 case study," *Mathematics*, vol. 10, no. 11, p. 1929, 2022, https://doi.org/10.3390/math10111929.

[2] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, "Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Advances*

in Engineering Software, vol. 174, p. 103282, 2022, https://doi.org/10.1016/j.advengsoft.2022.103282.

[3] S. Ekinci and D. Izci, "Enhancing IIR system identification: Harnessing the synergy of gazelle optimization and simulated annealing algorithms," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, p. 100225, 2023, https://doi.org/10.1016/j.prime.2023.100225.

[4] D. E. Goldberg, "The genetic algorithm approach: why, how, and what next?," In *Adaptive and learning systems: Theory and applications*, pp. 247-253, 1986, https://doi.org/10.1007/978-1-4757-1895-9_17.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942-1948 vol.4, 1995, https://doi.org/10.1109/ICNN.1995.488968.

[6] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," in *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006, https://doi.org/10.1109/MCI.2006.329691.

[7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, pp. 459-471, 2007, https://doi.org/10.1007/s10898-007-9149-x.

[8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46-61, 2014, https://doi.org/10.1016/j.advengsoft.2013.12.007.

[9] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849-872, 2019, https://doi.org/10.1016/j.future.2019.02.028.

[10] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, p. 114864, 2021, https://doi.org/10.1016/j.eswa.2021.114864.

[11] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information sciences*, vol. 237, pp. 82-117, 2013, https://doi.org/10.1016/j.ins.2013.02.041.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997, https://doi.org/10.1109/4235.585893.

[13] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, pp. 695-701, 2005, https://doi.org/10.1109/CIMCA.2005.1631345.

[14] D. Izci, S. Ekinci, E. Eker and M. Kayri, "Improved Manta Ray Foraging Optimization Using Opposition-based Learning for Optimization Problems," *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-6, 2020, https://doi.org/10.1109/HORA49412.2020.9152925.

[15] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on optimization*, vol. 7, no. 1, pp. 1-25, 1997, https://doi.org/10.1137/S1052623493250780.

[16] S. Ekinci, D. Izci, E. Eker, L. Abualigah, C. L. Thanh, and S. Khatir, "Hunger games pattern search with elite opposite-based solution for solving complex engineering design problems," *Evolving Systems*, vol. 15, no. 3, pp. 939-964, 2024, https://doi.org/10.1007/s12530-023-09526-9.

[17] E. Eker, "Development of Random Walks Strategy-Based Dandelion Optimizer and Its Application to Engineering Design Problems," in *IEEE Access*, vol. 13, pp. 56547-56575, 2025, https://doi.org/10.1109/ACCESS.2025.3554505.

[18] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112-147, 1998, https://doi.org/10.1137/S1052623496303470.

[19] D. İzci, S. Ekinci, M. Kayri, and E. Eker, "A Novel Enhanced Metaheuristic Algorithm for Automobile Cruise Control System," *Electrica*, vol. 21, no. 3, 2021, https://doi.org/10.5152/electrica.2021.21016.

[20] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer," *Neural Computing and Applications*, vol. 35, no. 5, pp. 4099-4131, 2023, https://doi.org/10.1007/s00521-022-07854-6.

[21] G. Singh *et al*., "Enhanced Global Optimization Using a Novel Hybrid Sine Cosine-Gazelle Algorithm with Brownian Motion and Lévy Flight Mechanisms," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, p. 108, 2025, https://doi.org/10.1007/s44196-025-00823-6.

[22] S. Biswas *et al*., "Integrating Differential Evolution into Gazelle Optimization for advanced global optimization and engineering applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 434, p. 117588, 2025, https://doi.org/10.1016/j.cma.2024.117588.

[23] L. Abualigah, A. Diabat, and R. A. Zitar, "Orthogonal learning Rosenbrock's direct rotation with the gazelle optimization algorithm for global optimization," *Mathematics*, vol. 10, no. 23, p. 4509, 2022, https://doi.org/10.3390/math10234509.

[24] G. Singh, S. Biswas, B. Maiti and U. K. Bera, "A Novel Hybrid Gazelle Optimization Algorithm with Differential Evolution for Solving Engineering Design Problem," *2024 IEEE Silchar Subsection Conference (SILCON 2024)*, pp. 1-6, 2024, https://doi.org/10.1109/SILCON63976.2024.10910408.

[25] M. Abdel-Salam, H. Askr, and A. E. Hassanien, "Adaptive chaotic dynamic learning-based gazelle optimization algorithm for feature selection problems," *Expert Systems with Applications*, vol. 256, p. 124882, 2024, https://doi.org/10.1016/j.eswa.2024.124882.

[26] R. Mahajan, H. Sharma, K. Arora, G. P. Joshi, and W. Cho, "Comparative analysis of the gazelle Optimizer and its variants," *Heliyon*, vol. 10, no. 17, 2024, https://doi.org/10.1016/j.heliyon.2024.e36425.

[27] D. Izci and S. Ekinci, "Fractional order controller design via gazelle optimizer for efficient speed regulation of micromotors," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 6, p. 100295, 2023, https://doi.org/10.1016/j.prime.2023.100295.

[28] M. Abdel-Salam, H. Askr, and A. E. Hassanien, "Adaptive chaotic dynamic learning-based gazelle optimization algorithm for feature selection problems," *Expert Systems with Applications*, vol. 256, p. 124882, 2024, https://doi.org/10.1016/j.eswa.2024.124882.

[29] T. A. Khan, *et al*., "A gazelle optimization expedition for key term separated fractional nonlinear systems with application to electrically stimulated muscle modeling," *Chaos, Solitons & Fractals*, vol. 185, p. 115111, 2024, https://doi.org/10.1016/j.chaos.2024.115111.

[30] M. K. Nour, I. Issaoui, A. Edris, A. Mahmud, M. Assiri and S. S. Ibrahim, "Computer Aided Cervical Cancer Diagnosis Using Gazelle Optimization Algorithm With Deep Learning Model," in *IEEE Access*, vol. 12, pp. 13046-13054, 2024, https://doi.org/10.1109/ACCESS.2024.3351883.

[31] R. Mallipeddi and P. N. Suganthan, "Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems," *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2010, https://doi.org/10.1109/CEC.2010.5586330.

[32] E. Alpaydm, "Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms," in *Neural Computation*, vol. 11, no. 8, pp. 1885-1892, 1999, https://doi.org/10.1162/089976699300016007.

[33] A. D. K. Tareen, M. S. A. Nadeem, K. J. Kearfott, K. Abbas, M. A. Khawaja, and M. Rafique, "Descriptive analysis and earthquake prediction using boxplot interpretation of soil radon time series data," *Applied radiation and isotopes*, vol. 154, p. 108861, 2019, https://doi.org/10.1016/j.apradiso.2019.108861.

[34] F. Mohr and J. N. van Rijn, "Learning curves for decision making in supervised machine learning: a survey," *Machine Learning*, vol. 113, no. 11, pp. 8371-8425, 2024, https://doi.org/10.1007/s10994-024-06619-7.