

Article

Beyond Significance: Promoting Effect Size Measures in Comparing Metaheuristic Algorithms

Nourhan M. H. Ismail  and Mahamed G. H. Omran * 

College of Computer and Systems Engineering, Abdullah Al Salem University, Khaldiya 72303, Kuwait;
nourhan.mo.ismail@gmail.com

* Correspondence: omran.m@ieee.org

Abstract

This paper advocates the systematic use of effect size measures—such as Cohen’s d and Cliff’s δ —in the comparative evaluation of metaheuristic optimization methods. While statistical significance tests remain the dominant tool for determining whether observed performance differences are unlikely to be due to chance, they do not convey the magnitude or practical relevance of these differences. As a result, many studies report “significant” results that offer limited insight into the actual strength of one algorithm over another. Despite the clear value of effect size measures in addressing this gap, their adoption within the metaheuristics community remains limited. This paper highlights the role of effect sizes as complementary, not alternative, to traditional hypothesis testing. Through illustrative examples and discussion, we demonstrate how effect sizes provide a more informative and transparent basis for interpreting algorithmic superiority, contributing to more robust, reproducible, and meaningful empirical comparisons. The paper aims to encourage researchers to integrate effect size reporting into standard practice, thereby strengthening the methodological foundations of experimental research in metaheuristics.

Keywords: metaheuristics; optimization; effect size; Cohen’s d ; Cliff’s δ ; statistical tests

1. Introduction

1.1. Metaheuristics

Metaheuristics are optimization methods that combine local improvement procedures with higher-level strategies to guide the search through complex solution spaces. Their purpose is to escape local optima and enable robust search behavior in problems where simple optimization approaches fail. They work by selecting the best possible solutions per iteration to guide the search toward converging on an optimal solution, and by randomization to escape local optima and ensure the diversity of solutions [1,2]. Metaheuristic algorithms typically mimic stochastic search strategies inspired by natural processes, as seen with algorithms such as the Genetic Algorithm (GA), which simulates Charles Darwin’s evolutionary theory [3].

1.2. Use and Analysis of Metaheuristics

The effectiveness of metaheuristic algorithms is well-established across a diverse spectrum of complex fields. Their application spans critical engineering domains—from digital filter and solar cell design to mechanical, electrical, and industrial systems—as well as advanced computational areas like neural networks, data mining, software engineering, and cost optimization. This proven versatility has solidified their role as essential tools



Academic Editors: Alexander E.I. Brownlee and Frank Werner

Received: 19 January 2026

Revised: 23 February 2026

Accepted: 28 February 2026

Published: 4 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

for tackling optimization problems that are intractable for classical methods, driving their increasing adoption within the scientific community [1].

When evaluating the performance of these algorithms, the prevailing methodology relies on statistical hypothesis testing, where a p -value determines if observed differences are statistically significant. Commonly used parametric tests include t -tests and ANOVA, while non-parametric alternatives such as the Wilcoxon rank-sum, Friedman, and Kruskal–Wallis tests are also applied [4]. However, while the p -value can identify whether a difference exists, it fails to indicate how substantial that difference is. To provide a complete and practically meaningful comparison, it is essential to complement significance testing with effect size measures—such as Cohen’s d or Cliff’s δ —which quantify the magnitude of performance differences, offering deeper insight into their real-world relevance [5].

Comparative studies commonly examine several algorithms across benchmark suites, where each algorithm–instance combination may involve different numbers of independent runs. Algorithm performance is typically quantified using metrics such as solution quality, expressed through mean or median objective values [6], or computational efficiency, measured in CPU time or the number of function evaluations [7]. Statistical significance testing is then employed to determine whether one algorithm achieves a statistically significant advantage over another with respect to these performance metrics.

Although significance testing is widely used, effect size measures remain largely absent from the metaheuristics literature. Most studies neither compute nor report them. However, one of the few exceptions is [8], which reports Cohen’s d as a standardized effect size measure to determine practical significance in comparing metaheuristic algorithms used for hybrid energy systems. The lack of effect size reporting means that statistically significant findings may offer limited insight into whether performance differences are meaningful in practice.

Exclusive reliance on significance testing presents several well-documented limitations. First, statistical significance does not convey the magnitude or practical relevance of differences. A result may be statistically significant yet trivial, or practically important but fail to reach significance due to limited statistical power [4,9]. Traditional tests based on means are also sensitive to outliers, meaning that a statistically insignificant mean difference may still be practically important [4,9].

Second, significance testing is highly sensitive to sample size. A limited number of runs can obscure true performance differences by producing misleading rankings [10], increasing the risk of selecting a seemingly superior but actually suboptimal configuration [10]. Conversely, very large sample sizes can make trivial differences statistically significant potentially leading to misleading conclusions [11].

Third, the stochastic nature of metaheuristics introduces variability in performance measurements [12], which may violate assumptions of parametric tests, such as normality and homoscedasticity [10]. Although non-parametric methods exist, these assumptions are sometimes unchecked [13]. Additionally, multiple algorithm comparisons across many problems raise the issue of family-wise error.

Effect size measures address these limitations by quantifying the magnitude of performance differences in standardized, sample-size-independent units. Whereas significance tests determine whether a difference exists, effect sizes convey its magnitude and practical importance. Rather than replacing significance testing, effect sizes serve as a complementary tool; significance testing addresses the likelihood that results arose by chance, while effect sizes provide insight into the practical relevance of the observed differences. Used together, they provide a more comprehensive understanding of algorithmic performance, distinguishing between differences that are both statistically and practically significant,

statistically significant but trivial, or potentially meaningful but not statistically significant due to limited power.

This paper demonstrates the proper application of effect size measures for comparing metaheuristic algorithm performance. We illustrate this methodology through a comparative analysis of a recent algorithm, EnSO [14], against two other competitors. To robustly quantify the magnitude of observed performance differences, we employ both a parametric and a non-parametric effect size measure. Furthermore, we investigate and discuss the potential divergence in results obtained from these complementary statistical approaches.

The remainder of the paper is structured as follows. Section 2 provides an overview of the three metaheuristic approaches. Section 3 presents and discusses the used effect size measures. Experimental setup is described in Section 4. Experimental results are reported and discussed in Section 5. Finally, Section 7 concludes the paper.

2. Representative Metaheuristics

In comparative studies of metaheuristic optimization, it is common to include algorithms that reflect widely used search paradigms in order to provide meaningful reference points for evaluation [15,16]. Such algorithms typically rely on population-based search, stochastic exploration, and iterative refinement of candidate solutions, and are often inspired by natural or collective processes [1]. Including these methods allows new or modified approaches to be assessed against other optimization methods, offering insight into relative performance, robustness, and consistency across different problem types. In this context, baseline metaheuristics serve not as definitive standards, but as practical comparators that help contextualize experimental results and highlight the characteristics of alternative algorithmic designs.

Following this approach, COA [17] and HBA [18] are included as representative nature-inspired metaheuristics, providing reference baselines for evaluating the performance of EnSO on diverse functions.

2.1. Ensemble of Simple Optimizers (EnSO)

The Ensemble of Simple Optimizers (EnSO) [14] is an ensemble metaheuristic that combines ideas from Biotic Flower Pollination (BFPA) [19], Jaya [20], and the rime-ice-based optimization algorithm (RIME) [21] to exploit their complementary strengths and improve overall performance. It works with a group of candidate solutions, applying different strategies to parts of the group to balance exploring new regions and improving promising solutions. At each step, solutions are updated based on the best performers, random guidance, and comparisons with neighbors.

2.2. Coyote Optimization Algorithm (COA)

The Coyote Optimization Algorithm (COA) [17] is a recent population-based metaheuristic approach inspired by the behavior of coyotes. The algorithm begins by initializing multiple packs, each containing several coyotes, and evaluating their adaptation to the problem. Within each pack, an alpha coyote is identified to guide the group, and a social tendency is computed to influence individual behavior. Each coyote then updates its position based on the social condition of the pack and its own adaptation, followed by an evaluation of the new state. The algorithm incorporates birth, death, and aging processes to simulate natural population dynamics and encourage exploration. Coyotes may also move between packs, allowing information to be shared across the population. This iterative process continues until a predefined stopping criterion is met, after which the best-adapted coyote is selected as the solution.

2.3. Honey Badger Algorithm (HBA)

The Honey Badger Algorithm (HBA) [18] is inspired by the foraging behavior of honey badgers, which use both scent and digging to locate food. In “digging mode,” the animal explores areas guided by its sense of smell to find prey, while in “honey mode,” it follows cues from the honeyguide bird to directly locate beehives. This dual strategy forms the conceptual basis of the algorithm.

HBA translates this behavior into an optimization process by maintaining a population of candidate solutions that are iteratively refined. The algorithm alternates between exploration, analogous to the digging mode, and directed search toward promising regions, analogous to the honey mode. Through repeated application of these steps, solutions improve over successive iterations, reflecting the balance between exploration and exploitation inherent in the honey badger’s hunting strategy.

3. Effect Size

The effect size is an important metric that measures how large the difference is between groups, providing insight beyond mere statistical significance [5]. Unlike statistical tests, which only indicate whether an observed effect is statistically unlikely under a null hypothesis, effect sizes show how large or meaningful a difference is. In psychology and social sciences [22–24], measures like Cohen’s d [25] and Pearson’s r [26] are commonly used to evaluate interventions or behavioral differences, making it easier to compare results from different studies. In medicine and clinical trials [27–29], effect sizes help determine whether a treatment produces meaningful improvements beyond statistical significance, such as symptom reduction or disease detection. In machine learning research [30–32], effect sizes are used to compare algorithm performance across datasets, providing a clearer picture of the practical differences. Non-parametric measures, such as Cliff’s δ , are especially useful when data is skewed or non-normally distributed, as they focus on the typical size and direction of differences rather than relying on means alone. These measures make it possible to evaluate algorithms across multiple test functions, showing consistent trends in performance rather than isolated differences. In this study, using effect size metrics like Cliff’s δ and Cohen’s d allows for a clear, scale-aware comparison of algorithms, identifying which tends to perform better and making the results easier to interpret.

In the following sections, we will explore two common effect size measures, Cohen’s d and Cliff’s δ , and discuss their application in comparing metaheuristic algorithms.

3.1. Cohen’s d

Cohen’s d [25] is a standardized measure of effect size that quantifies the magnitude of the difference between two groups. It is calculated as the difference between the group means divided by the pooled standard deviation, making it independent of the units of measurement. In the context of comparing metaheuristic algorithms, Cohen’s d provides insight into how substantial the observed performance differences are, beyond what a simple statistical significance test can reveal [25,33].

Cohen’s d is calculated through the following formula [25]:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{pooled}} \quad (1)$$

In Equation (1), $\bar{x}_1$ and $\bar{x}_2$ denote the means of both groups and σ_{pooled} is the pooled standard deviation of both groups, which can be calculated using the following equation [25]:

$$\sigma_{pooled} = \sqrt{\frac{\sum_{i=1}^{n_1} (x_{1,i} - \bar{x}_1)^2 + \sum_{j=1}^{n_2} (x_{2,j} - \bar{x}_2)^2}{n_1 + n_2 - 2}} \quad (2)$$

In Equation (2), $x_{1,i}$ and $x_{2,j}$ are observations in the first and second group, respectively, and n_1 and n_2 refer to the number of observations in each group. The magnitude of the effect size can be interpreted using the thresholds shown in Table 1.

Table 1. Cohen's d effect size interpretation [25].

Effect Size Category	Cohen's d
Small	0.2
Medium	0.5
Large	0.8

Because Cohen's d relies on the pooled standard deviation, it implicitly assumes that the data are approximately normally distributed [25,33], and its magnitude is inherently sensitive to small or unstable variance estimates [33]. This assumption may not hold in the evaluation of metaheuristic algorithms [34]. As a result, while Cohen's d offers useful standardized comparisons, its interpretation should be made with caution in this context, and it is often beneficial to complement it with non-parametric measures of effect size, such as Cliff's δ , which do not rely on distributional assumptions [35].

3.2. Cliff's δ

Cliff's δ [35] is a non-parametric measure of effect size that quantifies the degree of stochastic dominance between two groups. Rather than relying on group means and variances, Cliff's δ measures the probability that a randomly selected observation from one group will have a larger value than a randomly selected observation from another group, minus the reverse probability. Its values range from minus one to plus one, where zero indicates no difference between groups, positive values indicate dominance of the first group, and negative values indicate dominance of the second group. The use of non-parametric effect sizes does not compromise analysis even with normally distributed data. Its only drawback is being more computationally intensive to calculate [35,36].

Cliff's δ can be calculated through the following formula:

$$\delta = \frac{\#(x_{1,i} > x_{2,j}) - \#(x_{1,i} < x_{2,j})}{n_1 n_2} \quad (3)$$

In Equation (3), $x_{1,i}$ denotes the i -th observation from the first group and $x_{2,j}$ denotes the j -th observation from the second group. The terms $\#(x_{1,i} > x_{2,j})$ and $\#(x_{1,i} < x_{2,j})$ represent the number of pairwise comparisons in which an observation from the first group is greater than or less than an observation from the second group, respectively. The quantities n_1 and n_2 correspond to the sample sizes of the first and second groups, and their product gives the total number of pairwise comparisons [36]. The resulting value, therefore, reflects the net probability that one group tends to produce larger values than the other. The absolute values can be interpreted through the thresholds in Table 2.

In the context of comparing metaheuristic algorithms, Cliff's δ is particularly suitable because it does not assume normality or equal variances [35]. Performance data obtained from stochastic optimization algorithms are often skewed, multimodal, or contain outliers, conditions under which parametric effect size measures may be unreliable [34]. By operating directly on pairwise rank comparisons, Cliff's δ provides a robust and non-parametric assessment of the magnitude of the difference in performance. Consequently, it serves

as an alternative measure to Cohen's d , allowing effect sizes to be evaluated under fewer statistical assumptions and offering a more reliable characterization of algorithm behavior on benchmark problems.

Table 2. Cliff's δ effect size interpretation [35,36].

Effect Size Category	Cliff's δ
Negligible	$ \delta < 0.147$
Small	$0.147 \leq \delta < 0.33$
Medium	$0.33 \leq \delta < 0.474$
Large	$ \delta \geq 0.474$

4. Experimental Setup

This section presents the experimental setup used to evaluate the three metaheuristic algorithms. It describes the benchmark problems, the algorithmic parameters, and the procedures followed to ensure a fair and consistent comparison. Additionally, the performance metrics, including both statistical significance and effect size measures, are introduced to provide a comprehensive assessment of algorithm behavior.

4.1. Congress of Evolutionary Computation (CEC) 2013 Benchmark

The Congress of Evolutionary Computation (CEC) 2013 benchmark suite [37] is a widely used collection of test functions designed specifically for evaluating the performance of metaheuristics under controlled and reproducible conditions [38,39]. It consists of 28 functions that include unimodal, multimodal, and composition functions (see Table 3), each with distinct characteristics intended to challenge different aspects of optimization algorithms. Unimodal functions (e.g., Figure 1) test an algorithm's ability to converge to a single global optimum efficiently, while multimodal functions (e.g., Figure 2) evaluate its capability to escape local optima. Composition functions (e.g., Figure 3) combine multiple types of landscapes with different properties around different local optima [37].

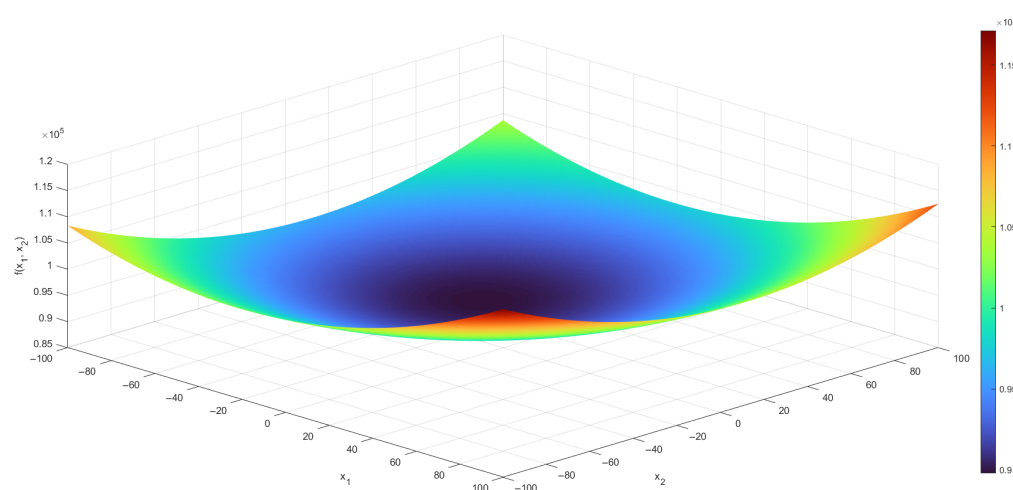


Figure 1. 3D plot showing function 1 (Sphere Function) [37], an example of a unimodal function.

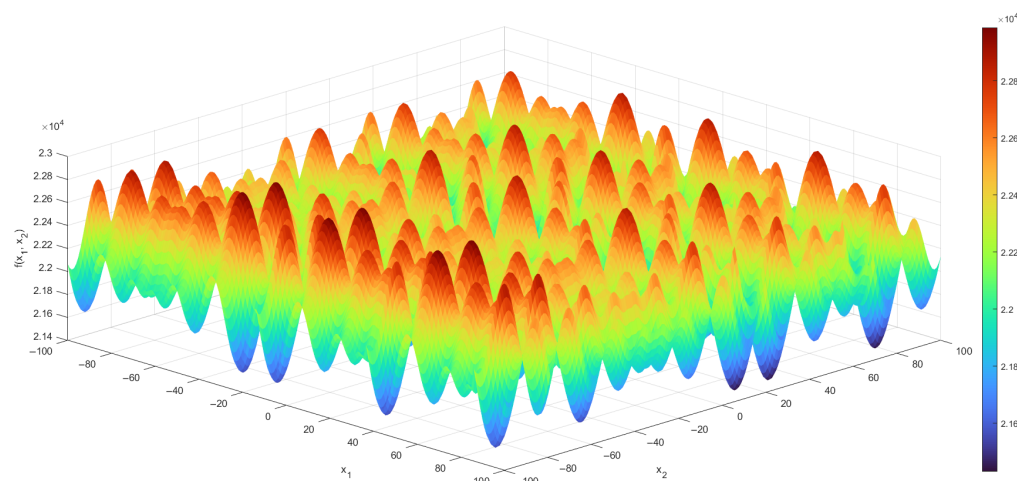


Figure 2. 3D plot showing function 14 (Schwefel's Function) [37], an example of a multimodal function.

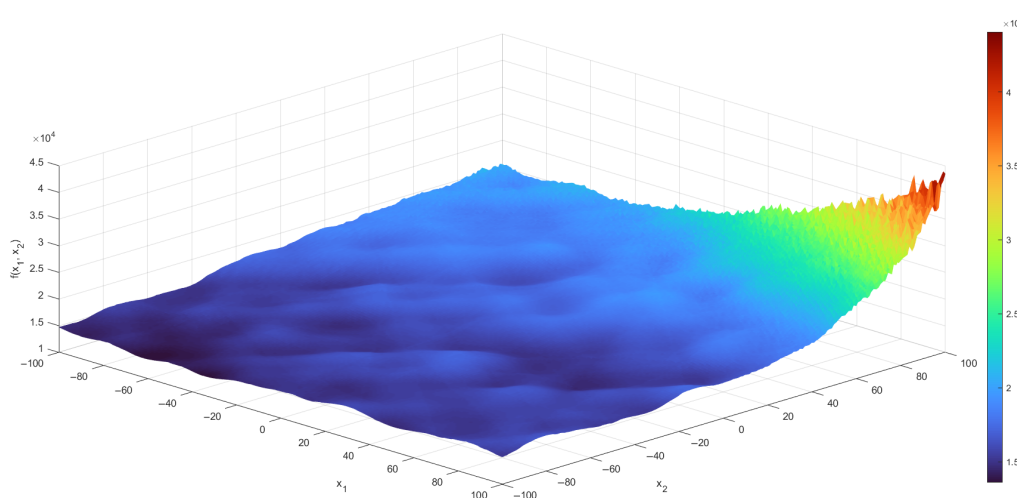


Figure 3. 3D plot showing function 28 (combines functions 1, 7, 15, 19, and 20) [37], an example of a composite function.

Table 3. CEC 2013 benchmark function categories.

Test Function Index	Category
1–5	Unimodal
6–20	Multimodal
21–28	Composite

4.2. Congress of Evolutionary Computation (CEC) 2011 Benchmark

The Congress of Evolutionary Computation (CEC) 2011 benchmark suite [40] is a collection of test problems designed to evaluate the performance of metaheuristic algorithms on real-world numerical optimization problems under standardized and reproducible conditions. Unlike the 2013 benchmarks, which are purely synthetic benchmark landscapes, the CEC 2011 suite focuses on problems derived from practical application domains such as engineering, data mining, and biological modeling, where optimization tasks often involve complex, irregular search spaces. The suite has been previously used [41–43] to provide a realistic assessment of algorithm robustness and practical applicability. These problems typically exhibit characteristics such as nonlinearity, multimodality, noise, and constraint interactions, which are representative of real industrial and scientific optimization scenarios. As such, functions T_1 , T_2 , T_4 , T_5 , and T_6 (see Table 4) from the CEC 2011 benchmark

were selected to complement the synthetic benchmark suite and to validate whether an algorithm's performance generalizes to real-world problem structures.

Table 4. CEC 2011 benchmark functions.

Test Function Index	Function
1	Parameter Estimation for Frequency-Modulated Sound Waves
2	Lennard–Jones Potential Problem
4	Optimal Control of a Non-Linear Stirred Tank Reactor
5	Tersoff Potential Function Minimization Problem
6	Spread Spectrum Radar Polly Phase Code Design

To illustrate a standard metaheuristics performance assessment, we will compare the EnSO algorithm against two methods: COA and HBA. This evaluation will be conducted through two paired analyses: EnSO versus COA and EnSO versus HBA. For each comparison, we will first apply an appropriate statistical test to determine the significance of any performance differences, followed by an effect size analysis to quantify the magnitude of those differences. All three algorithms, i.e., EnSO, COA, and HBA, were evaluated on the 28 functions of the CEC 2013 benchmark suite, as well as 5 functions from the CEC 2011 benchmark suite. Each algorithm was executed 30 times independently. Both the average and standard deviation of the final objective values across the 30 runs were computed to summarize algorithm performance. For the 2013 suite, the problem dimensionality, D , was fixed at 50, with all decision variables bounded between -100 and 100 . The stopping criterion was defined as reaching $10,000 \times D$ (500,000) function evaluations. In contrast, the 2011 suite had varying dimensionality and bounded variables depending on the problem, as shown in Table 5. In this case, the stopping criterion is reached at 150,000 function evaluations.

Table 5. CEC 2011 benchmark functions' dimensionality.

Test Function	Dimensions	Variable Bounds
T_1	6	$[-6.4, 6.35]$
T_2	30	$x = (x_1, \dots, x_{30})$ $x_1, x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ $x_i \in \left[-\left(4 + \frac{\lfloor (i-4)/3 \rfloor}{4}\right), 4 + \frac{\lfloor (i-4)/3 \rfloor}{4} \right], \quad i = 4, \dots, 30$
T_4	1	initialized at $[0, 5]$
T_5	30	$x = (x_1, \dots, x_{30})$ $x_1, x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ $x_i \in \left[-\left(4 + \frac{\lfloor (i-4)/3 \rfloor}{4}\right), 4 + \frac{\lfloor (i-4)/3 \rfloor}{4} \right], \quad i = 4, \dots, 30$

Table 5. Cont.

Test Function	Dimensions	Variable Bounds
T_6	30	$x = (x_1, \dots, x_{30})$ $x_1, x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ $x_i \in \left[-\left(4 + \frac{\lfloor (i-4)/3 \rfloor}{4}\right), 4 + \frac{\lfloor (i-4)/3 \rfloor}{4} \right], \quad i = 4, \dots, 30$

The parameters of the three methods have been set according to the original papers.

4.3. Statistical Significance

Permutation tests [6,44] are a non-parametric method used to assess whether the observed differences between two sets of measurements are statistically significant. Unlike traditional parametric tests, permutation tests do not assume any specific distribution of the data, making them suitable for experimental results that may be non-normal or have unequal variances. The test works by repeatedly shuffling the labels of the two samples and computing a test statistic, such as the difference in means, for each permutation. The proportion of permutations that produce a test statistic as extreme as or more extreme than the observed statistic gives an empirical p -value, quantifying the likelihood that the observed difference could occur by chance.

In this study, permutation tests were applied to the final objective values obtained from 30 independent runs of each algorithm for every benchmark function. For each comparison (EnSO vs. COA and EnSO vs. HBA), $n_{perm} = 100,000$ random permutations were used to calculate the empirical p -values. This approach allows for assessing the statistical significance of the observed performance differences between algorithms while avoiding assumptions about the underlying distribution of the performance data.

4.4. Performance Metrics

Statistical comparisons were conducted using permutation tests to compute p -values, indicating whether observed differences could have arisen by chance. To account for multiple comparisons, a Bonferroni correction [45] was applied, dividing the significance threshold by 2 to avoid inflated Type I errors, given that we are comparing three approaches, EnSO, COA, and HBA, with comparisons conducted between EnSO vs. COA and EnSO vs. HBA. Effect sizes were calculated using both Cohen's d and Cliff's δ . For both Cohen's d and Cliff's δ , the effect sizes are computed assuming that smaller values indicate better performance, since the optimization benchmarks are minimization problems. Consequently, the first group in each pairwise comparison is taken to be the algorithm that achieves the lower objective function value, effectively presuming the opposite result compared to the conventional interpretation, where higher values indicate improvement. This adjustment ensures that positive effect sizes consistently correspond to the superior algorithm in the context of minimization.

5. Experimental Results

5.1. Normality

The normality of the best values for each algorithm on the CEC 2013 benchmark functions was assessed using the Shapiro–Wilk test [46]. Table 6 shows the p -values for EnSO, COA, and HBA across all 28 functions, with values in bold indicating a rejection of the null hypothesis of normality at $\alpha = 0.05$. Table 7 presents an analogous analysis for the CEC 2011 benchmark functions, assessing the normality of the corresponding results

samples. The results reveal that most functions produce non-normal distributions for all three algorithms, particularly for EnSO and COA. HBA shows slightly fewer violations for CEC 2013 but still demonstrates non-normal behavior for most functions, whereas in CEC 2011, all samples are non-normal with the exception of EnSO on T_2 .

Given the frequent departure from normality as shown in Table 6, and especially Table 7, effect size measures that rely on parametric assumptions may not be consistently applicable across all functions. For this reason, Cliff's δ is employed as an alternative alongside Cohen's d to analyze the magnitude of difference in performance between the algorithms. While Cohen's d is informative when normality assumptions are satisfied [25], Cliff's δ provides a non-parametric alternative that remains valid regardless of distribution shape [35], allowing effect sizes to be interpreted consistently across both normal and non-normal cases.

Table 6. Shapiro–Wilk p -values showing normality for CEC 2013 benchmarks.

Function	EnSO	COA	HBA
T_1	2.7237×10^{-12}	1.4942×10^{-2}	4.5914×10^{-8}
T_2	4.8392×10^{-2}	1.6119×10^{-2}	7.5988×10^{-2}
T_3	7.2455×10^{-6}	7.6905×10^{-4}	4.9546×10^{-2}
T_4	6.5367×10^{-2}	9.9799×10^{-1}	3.8347×10^{-4}
T_5	1.1071×10^{-8}	3.5973×10^{-4}	9.7549×10^{-9}
T_6	3.3654×10^{-9}	1.3420×10^{-7}	2.6777×10^{-5}
T_7	4.1610×10^{-1}	6.2457×10^{-1}	3.8156×10^{-1}
T_8	3.8398×10^{-2}	8.9331×10^{-2}	1.9421×10^{-1}
T_9	1.1585×10^{-1}	6.8442×10^{-1}	2.3807×10^{-1}
T_{10}	2.0069×10^{-3}	2.4328×10^{-2}	2.5058×10^{-4}
T_{11}	3.7796×10^{-1}	1.5539×10^{-6}	7.6537×10^{-2}
T_{12}	1.5051×10^{-1}	2.6616×10^{-1}	3.8397×10^{-1}
T_{13}	6.6377×10^{-2}	7.2067×10^{-1}	4.7396×10^{-1}
T_{14}	1.8640×10^{-1}	3.0795×10^{-1}	6.2165×10^{-2}
T_{15}	4.0750×10^{-1}	3.9046×10^{-3}	1.2544×10^{-1}
T_{16}	3.5832×10^{-1}	8.4724×10^{-5}	1.4038×10^{-1}
T_{17}	3.6652×10^{-2}	4.7720×10^{-3}	5.0323×10^{-2}
T_{18}	4.7189×10^{-1}	8.6396×10^{-4}	3.8652×10^{-1}
T_{19}	2.4415×10^{-1}	2.3899×10^{-3}	3.1978×10^{-2}
T_{20}	2.1171×10^{-1}	4.2810×10^{-1}	4.3767×10^{-1}
T_{21}	5.0842×10^{-6}	1.0014×10^{-7}	3.9255×10^{-8}
T_{22}	5.4192×10^{-1}	1.4576×10^{-9}	4.4220×10^{-1}
T_{23}	3.8842×10^{-2}	2.8217×10^{-3}	8.2803×10^{-1}
T_{24}	3.3661×10^{-1}	1.7456×10^{-1}	8.1614×10^{-2}
T_{25}	5.2411×10^{-1}	8.3632×10^{-1}	4.6995×10^{-3}
T_{26}	3.0105×10^{-10}	4.5924×10^{-9}	1.1384×10^{-7}
T_{27}	3.0105×10^{-3}	1.6305×10^{-1}	5.0464×10^{-1}
T_{28}	1.0774×10^{-8}	1.7557×10^{-4}	3.8688×10^{-7}

Note: Values highlighted in bold indicate the distribution of the 30 best values per test function may violate the normality assumption at $\alpha = 0.05$.

Table 7. Shapiro–Wilk p -values showing normality for CEC 2011 benchmarks.

Function	EnSO	COA	HBA
T_1	1.5094×10^{-6}	5.7663×10^{-7}	1.6901×10^{-2}
T_2	4.4491×10^{-1}	2.3932×10^{-3}	1.9908×10^{-5}
T_3	8.9957×10^{-6}	3.3543×10^{-6}	3.9932×10^{-9}
T_4	8.0998×10^{-4}	4.2769×10^{-6}	8.2698×10^{-3}
T_5	3.1566×10^{-3}	2.9973×10^{-8}	1.8396×10^{-3}

Note: Values highlighted in bold indicate the distribution of the 30 best values per test function may violate the normality assumption at $\alpha = 0.05$.

5.2. Performance Statistics

Tables 8–10 summarize the performance of the three algorithms, EnSO, COA, and HBA, across the 28 benchmark functions from CEC 2013, whereas Table 11 summarizes their performance across the 5 functions from CEC 2011. For each function, the best fit values were recorded over 30 independent runs, including their minimum, maximum, median, mean, and standard deviation. The tables provide a detailed view of the algorithms' performance, allowing a preliminary comparison of their typical outcomes.

Table 8. Performance statistics for unimodal test functions T1–T5 using 500,000 FEs.

Function	EnSO				COA				HBA			
	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max
T_1	$-1.4000 \times 10^3 \pm 5.27 \times 10^{-13}$	-1.4000×10^3	-1.4000×10^3	-1.4000×10^3	$-1.4000 \times 10^3 \pm 3.97 \times 10^{-8}$	-1.4000×10^3	-1.4000×10^3	-1.4000×10^3	$-1.4000 \times 10^3 \pm 2.95 \times 10^{-11}$	-1.4000×10^3	-1.4000×10^3	-1.4000×10^3
T_2	$2.8126 \times 10^6 \pm 1.31 \times 10^6$	2.5618×10^6	7.6835×10^5	5.0313×10^6	$5.0311 \times 10^6 \pm 2.08 \times 10^6$	5.0188×10^6	2.2067×10^6	1.1626×10^7	$1.2038 \times 10^6 \pm 4.85 \times 10^5$	1.0971×10^6	3.8628×10^5	2.2295×10^6
T_3	$1.1624 \times 10^8 \pm 1.23 \times 10^8$	7.2816×10^7	8.5610×10^6	5.3139×10^8	$1.3465 \times 10^8 \pm 9.15 \times 10^8$	9.9335×10^8	2.5318×10^8	4.0546×10^9	$3.3514 \times 10^8 \pm 2.41 \times 10^8$	2.9910×10^8	2.6861×10^7	1.0040×10^9
T_4	$-3.6362 \times 10^2 \pm 2.76 \times 10^2$	-3.9881×10^2	-8.5517×10^2	2.3421×10^2	$5.8841 \times 10^4 \pm 1.28 \times 10^4$	5.9086×10^4	3.0784×10^4	8.7745×10^4	$2.0114 \times 10^3 \pm 2.37 \times 10^3$	1.6852×10^3	-7.8038×10^2	9.1834×10^3
T_5	$-1.0000 \times 10^3 \pm 1.91 \times 10^{-9}$	-1.0000×10^3	-1.0000×10^3	-1.0000×10^3	$-1.0000 \times 10^3 \pm 1.08 \times 10^{-4}$	-1.0000×10^3	-1.0000×10^3	-1.0000×10^3	$-1.0000 \times 10^3 \pm 6.06 \times 10^{-10}$	-1.0000×10^3	-1.0000×10^3	-1.0000×10^3

Table 9. Performance statistics for multimodal test functions T6–T20 using 500,000 FEs.

Function	EnSO				COA				HBA			
	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max
T_6	$-8.5270 \times 10^2 \pm 1.16 \times 10^1$	-8.5600×10^2	-8.5660×10^2	-8.0990×10^2	$-8.4450 \times 10^2 \pm 1.93 \times 10^1$	-8.5170×10^2	-8.5660×10^2	-7.8860×10^2	$-8.4070 \times 10^2 \pm 2.54 \times 10^1$	-8.5380×10^2	-8.8120×10^2	-7.9850×10^2
T_7	$-7.4440 \times 10^2 \pm 1.42 \times 10^1$	-7.4740×10^2	-7.6930×10^2	-7.1540×10^2	$-7.0830 \times 10^2 \pm 1.48 \times 10^1$	-7.0880×10^2	-7.3270×10^2	-6.7580×10^2	$-7.3350 \times 10^2 \pm 1.32 \times 10^1$	-7.3410×10^2	-7.5750×10^2	-7.1060×10^2
T_8	$-6.7887 \times 10^2 \pm 3.50 \times 10^{-2}$	-6.7887×10^2	-6.7895×10^2	-6.7882×10^2	$-6.7887 \times 10^2 \pm 3.50 \times 10^{-2}$	-6.7887×10^2	-6.7898×10^2	-6.7881×10^2	$-6.7885 \times 10^2 \pm 2.70 \times 10^{-2}$	-6.7885×10^2	-6.7891×10^2	-6.7880×10^2
T_9	$-5.4578 \times 10^2 \pm 3.65 \times 10^0$	-5.4533×10^2	-5.4448×10^2	-5.3907×10^2	$-5.4382 \times 10^2 \pm 4.26 \times 10^0$	-5.4355×10^2	-5.5420×10^2	-5.3345×10^2	$-5.4928 \times 10^2 \pm 8.06 \times 10^0$	-5.5031×10^2	-5.6196×10^2	-5.3111×10^2
T_{10}	$-4.9992 \times 10^2 \pm 5.70 \times 10^{-2}$	-4.9994×10^2	-4.9999×10^2	-4.9973×10^2	$-4.9177 \times 10^2 \pm 2.78 \times 10^0$	-4.9199×10^2	-4.9597×10^2	-4.8278×10^2	$-4.9987 \times 10^2 \pm 8.10 \times 10^{-2}$	-4.9989×10^2	-4.9995×10^2	-4.9962×10^2
T_{11}	$-3.3302 \times 10^2 \pm 1.90 \times 10^1$	-3.3384×10^2	-3.6517×10^2	-3.0048×10^2	$-3.9954 \times 10^2 \pm 5.68 \times 10^{-1}$	-4.0000×10^2	-4.0000×10^2	-3.9800×10^2	$-2.3510 \times 10^2 \pm 3.59 \times 10^1$	-2.4379×10^2	-2.8296×10^2	-1.6420×10^2
T_{12}	$-1.2972 \times 10^2 \pm 3.76 \times 10^1$	-1.3632×10^2	-1.9397×10^2	-2.3420×10^1	$-1.2586 \times 10^2 \pm 3.57 \times 10^1$	-1.2478×10^2	-1.9636×10^2	-6.0210×10^1	$-2.0389 \times 10^1 \pm 5.03 \times 10^1$	-1.5941×10^1	-1.4777×10^2	9.2010×10^1
T_{13}	$7.8666 \times 10^1 \pm 3.82 \times 10^1$	8.1948×10^1	3.5480×10^1	1.3465×10^2	$8.9187 \times 10^1 \pm 5.70 \times 10^1$	9.8509×10^1	-1.7591×10^1	2.2185×10^2	$1.8161 \times 10^2 \pm 5.03 \times 10^1$	1.8234×10^2	9.2946×10^1	2.6979×10^2
T_{14}	$1.9164 \times 10^3 \pm 5.78 \times 10^2$	1.8480×10^3	9.5682×10^2	2.9410×10^3	$-9.3376 \times 10^1 \pm 2.90 \times 10^0$	-9.3970×10^1	-9.7807×10^1	-8.6315×10^1	$5.1717 \times 10^3 \pm 1.37 \times 10^3$	4.7643×10^3	3.1140×10^3	7.7790×10^3
T_{15}	$8.8114 \times 10^3 \pm 9.75 \times 10^3$	8.8688×10^3	6.6656×10^3	1.0594×10^4	$1.1072 \times 10^4 \pm 2.51 \times 10^1$	1.0519×10^4	6.7270×10^3	1.4339×10^4	$9.7419 \times 10^3 \pm 2.08 \times 10^3$	9.3208×10^3	6.4140×10^3	1.4465×10^4
T_{16}	$2.0197 \times 10^2 \pm 4.60 \times 10^{-1}$	2.0205×10^2	2.0084×10^2	2.0275×10^2	$2.0322 \times 10^2 \pm 4.03 \times 10^{-1}$	2.0331×10^2	2.0176×10^2	2.0372×10^2	$2.0330 \times 10^2 \pm 3.60 \times 10^{-1}$	2.0340×10^2	2.0241×10^2	2.0386×10^2
T_{17}	$4.4378 \times 10^2 \pm 1.87 \times 10^1$	4.4034×10^2	4.0889×10^2	5.0350×10^2	$3.5331 \times 10^2 \pm 9.64 \times 10^{-1}$	3.5314×10^2	3.5203×10^2	3.5616×10^2	$5.4977 \times 10^2 \pm 7.17 \times 10^1$	5.3973×10^2	4.4377×10^2	7.2882×10^2
T_{18}	$7.3286 \times 10^2 \pm 4.32 \times 10^1$	7.3233×10^2	6.5770×10^2	8.4456×10^2	$5.9199 \times 10^2 \pm 5.56 \times 10^1$	5.8334×10^2	5.2187×10^2	7.8828×10^2	$7.3878 \times 10^2 \pm 8.28 \times 10^1$	7.3181×10^2	5.8994×10^2	9.0212×10^2
T_{19}	$5.1078 \times 10^2 \pm 2.54 \times 10^0$	5.1034×10^2	5.0667×10^2	5.1614×10^2	$5.0340 \times 10^2 \pm 8.38 \times 10^{-1}$	5.0326×10^2	5.0224×10^2	5.0638×10^2	$5.1227 \times 10^2 \pm 3.47 \times 10^0$	5.1186×10^2	5.0729×10^2	5.2318×10^2
T_{20}	$6.2157 \times 10^2 \pm 5.24 \times 10^{-1}$	6.2156×10^2	6.2071×10^2	6.2298×10^2	$6.2136 \times 10^2 \pm 5.46 \times 10^{-1}$	6.2152×10^2	6.2035×10^2	6.2244×10^2	$6.2052 \times 10^2 \pm 1.20 \times 10^0$	6.2082×10^2	6.1810×10^2	6.2273×10^2

Table 10. Performance statistics for composite test functions T21–T28 using 500,000 FEs.

Function	EnSO				COA				HBA			
	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max
T_{21}	$1.6083 \times 10^3 \pm 2.37 \times 10^2$	1.5364×10^3	9.0000×10^2	1.8224×10^3	$1.1481 \times 10^3 \pm 3.93 \times 10^2$	9.0000×10^2	9.0000×10^2	1.8224×10^3	$1.6222 \times 10^3 \pm 1.33 \times 10^2$	1.5364×10^3	1.5364×10^3	1.8224×10^3
T_{22}	$3.0457 \times 10^3 \pm 5.83 \times 10^2$	3.0545×10^3	1.7090×10^3	4.5150×10^3	$8.2820 \times 10^2 \pm 2.62 \times 10^1$	8.2310×10^2	8.1820×10^2	9.6495×10^2	$7.0331 \times 10^3 \pm 1.89 \times 10^3$	6.6440×10^3	3.7946×10^3	1.1281×10^4
T_{23}	$9.8273 \times 10^3 \pm 7.87 \times 10^2$	9.7256×10^3	8.6768×10^3	1.1979×10^4	$1.2232 \times 10^4 \pm 2.32 \times 10^3$	1.1484×10^4	7.9729×10^3	1.5096×10^4	$1.1222 \times 10^4 \pm 1.54 \times 10^3$	1.1008×10^4	8.1485×10^3	1.4262×10^4
T_{24}	$1.3227 \times 10^3 \pm 2.22 \times 10^1$	1.3202×10^3	1.2815×10^3	1.3613×10^3	$1.3552 \times 10^3 \pm 1.01 \times 10^1$	1.3567×10^3	1.3304×10^3	1.3702×10^3	$1.4081 \times 10^3 \pm 2.80 \times 10^1$	1.4016×10^3	1.3608×10^3	1.4904×10^3
T_{25}	$1.4548 \times 10^3 \pm 1.74 \times 10^1$	1.4506×10^3	1.4193×10^3	1.5001×10^3	$1.4747 \times 10^3 \pm 9.89 \times 10^0$	1.4753×10^3	1.4562×10^3	1.4952×10^3	$1.5315 \times 10^3 \pm 3.48 \times 10^1$	1.5264×10^3	1.4795×10^3	1.6071×10^3
T_{26}	$1.4088 \times 10^3 \pm 4.15 \times 10^1$	1.4010×10^3	1.4003×10^3	1.6287×10^3	$1.4251 \times 10^3 \pm 7.40 \times 10^1$	1.4009×10^3	1.4003×10^3	1.6533×10^3	$1.6422 \times 10^3 \pm 4.97 \times 10^1$	1.6471×10^3	1.4001×10^3	1.6903×10^3
T_{27}	$2.9588 \times 10^3 \pm 1.43 \times 10^2$	2.9646×10^3	2.4381×10^3	3.2012×10^3	$3.0637 \times 10^3 \pm 1.14 \times 10^2$	3.0763×10^3	2.7375×10^3	3.2582×10^3	$3.2200 \times 10^3 \pm 2.43 \times 10^2$	3.1924×10^3	2.7719×10^3	3.7078×10^3
T_{28}	$2.5359 \times 10^3 \pm 1.36 \times 10^3$	1.8000×10^3	1.8000×10^3	4.9820×10^3	$1.8000 \times 10^3 \pm 5.00 \times 10^{-3}$	1.8000×10^3	1.8000×10^3	1.8000×10^3	$3.1974 \times 10^3 \pm 1.75 \times 10^3$	1.8000×10^3	1.8000×10^3	5.5653×10^3

Table 11. Performance statistics for CEC 2011 test functions using 150,000 FEs.

Function	EnSO				COA				HBA			
	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max	Mean $\pm$ SD	Median	Min	Max
T_1	$4.1247 \times 10^0 \pm 5.46 \times 10^0$	1.1214×10^{-1}	2.4437×10^{-3}	1.4781×10^1	$4.4712 \times 10^0 \pm 6.04 \times 10^0$	2.0385×10^{-13}	3.4867×10^{-20}	1.4779×10^1	$1.3002 \times 10^1 \pm 7.21 \times 10^0$	1.4347×10^1	0.0000×10^0	2.4883×10^1
T_2	$-1.5443 \times 10^1 \pm 2.41 \times 10^0$	-1.5255×10^1	-2.0331×10^1	-1.0538×10^1	$-2.0181 \times 10^1 \pm 3.71 \times 10^0$	-1.8392×10^1	-2.7958×10^1	-1.5271×10^1	$-2.5557 \times 10^1 \pm 3.06 \times 10^0$	-2.6556×10^1	-2.8423×10^1	-1.6594×10^1
T_4	$1.3820 \times 10^1 \pm 6.48 \times 10^{-2}$	1.3776×10^1	1.3771×10^1	1.3951×10^1	$1.7541 \times 10^1 \pm 3.22 \times 10^0$	1.7229×10^1	1.3771×10^1	2.0957×10^1	$1.3827 \times 10^1 \pm 1.70 \times 10^{-1}$	1.3771×10^1	1.3771×10^1	$1.$

5.3. Statistical Analysis

To complement the descriptive statistics, the effect sizes of algorithm comparisons were first analyzed using Cohen's d , then by Cliff's δ . Both measures were visualized using bar plots for both Cohen's d and Cliff's δ . Separate plots were generated for EnSO vs. COA and EnSO vs. HBA, with each bar representing the effect size calculated for a specific benchmark function. Statistical significance (as measured by the permutation test) is indicated by an asterisk above the bars when the corresponding p -value is below the corrected threshold of 0.025, following the Bonferroni adjustment for multiple comparisons. These plots provide a clear visual representation of the magnitude and significance of performance differences across all benchmark problems.

5.3.1. EnSO vs. COA (CEC 2013)

As shown in Figure 4, Cohen's d indicates that EnSO shows a strong advantage over COA on unimodal benchmarks ($T_1 - T_5$), with large positive effect sizes across the board, reflecting a clear performance gap in simple, single-optimum problems. For multimodal benchmarks ($T_6 - T_{20}$), the effect sizes vary in both sign and magnitude, suggesting that either algorithm's advantage is problem-dependent when multiple local optima are present. On composite benchmarks ($T_{21} - T_{28}$), the same variance can be seen, indicating that the advantage of either algorithm is problem-dependent. Overall, in most cases, regardless of the function's landscape, EnSO generally dominates.

In Figure 5, Cliff's δ confirms the patterns observed with Cohen's d . On unimodal benchmarks, Cliff's δ approaches its maximum positive value (e.g., T_1 , T_4 , and T_5 at 1.0), showing near-complete dominance of EnSO over COA. For multimodal benchmarks, Cliff's δ alternates between positive and negative values, illustrating how dominance shifts across individual benchmarks. On composite benchmarks ($T_{21} - T_{28}$), Cliff's δ values vary in both directions as well, supporting the same resolution drawn from Cohen's d values.

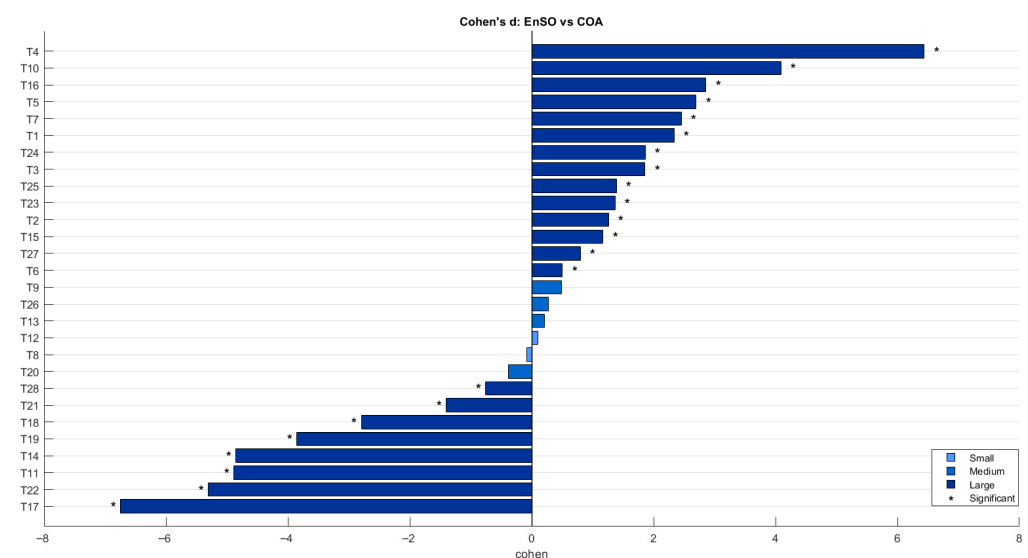


Figure 4. Cohen's d effect sizes for EnSO vs. COA across CEC 2013 functions. Positive values indicate EnSO superiority.

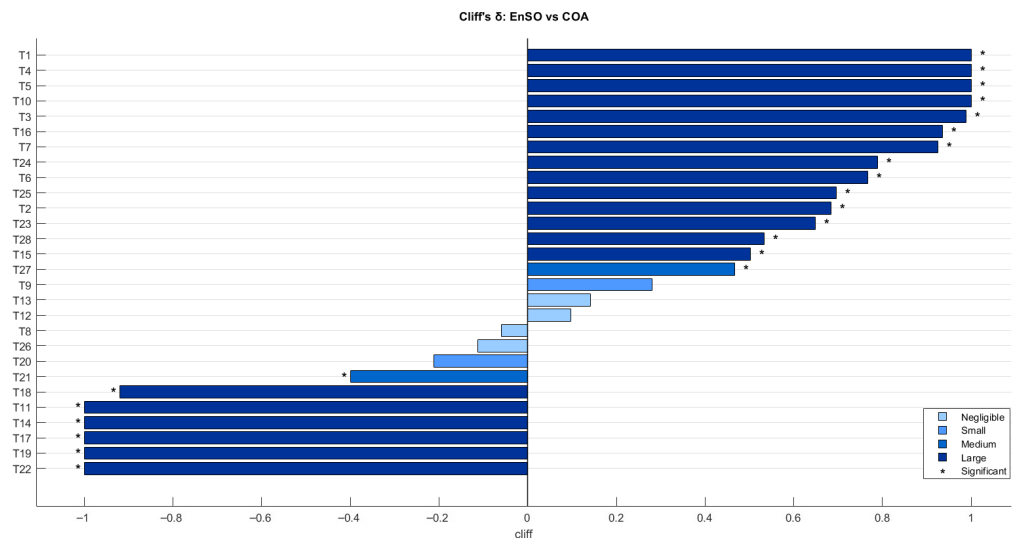


Figure 5. Cliff's δ effect sizes for EnSO vs. COA across CEC 2013 functions. Positive values indicate EnSO superiority.

The observed discrepancies between Cohen's d and Cliff's δ (e.g., T_9 , T_{12} , T_{13} , and T_{20}) arise from their fundamentally different definitions. Cohen's d quantifies standardized mean separation, while Cliff's δ measures the probability that one sample produces larger values than another. Even when both performance distributions are normal, substantial distributional overlap can result in small δ values despite noticeable mean differences, leading to different magnitude interpretations.

In comparisons where both distributions satisfy normality assumptions, Cohen's d is preferred because it directly measures standardized mean shift, which is fully characterized by the mean and standard deviation under normality. This provides greater sensitivity to average performance differences and allows direct comparison with widely established effect size interpretation thresholds in experimental research. Cliff's δ is therefore primarily emphasized for comparisons involving non-normal distributions, where parametric assumptions underlying Cohen's d may be violated.

Having characterized the performance differences between EnSO and COA, the analysis now turns to the comparison between EnSO and HBA using the same effect size measures.

5.3.2. EnSO vs. HBA (CEC 2013)

Comparing EnSO with HBA, Cohen's d in Figure 6 reveals consistently large positive effect sizes for most benchmarks, indicating a substantial and systematic advantage for EnSO. The only exception to this case are two unimodal functions (T_2 , T_5), one multimodal function (T_9), and one composite function (T_{20}). However, in several cases, effect sizes are extremely large (e.g., T_{26} and T_{11}), reflecting the sensitivity of Cohen's d to small or unstable variances [33] in stochastic optimization settings as well. Overall, this highlights EnSO as the ideal choice for most functions in comparison to HBA.

As in the previous comparison, Cohen's d is complemented by Cliff's δ to assess whether the observed mean differences correspond to any algorithm's dominance.

As seen in Figure 7, Cliff's δ supports and clarifies the trends seen with Cohen's d . Across most unimodal, multimodal, and composite benchmarks, Cliff's δ consistently approaches high positive values (e.g., T_{11}), indicating strong dominance of EnSO over HBA. Yet, the top difference in this case is not the same. Previously, it was shown to be T_{26} , whereas now it is T_{11} . This is due to the comparison in T_2 being between two non-normally distributed samples. In this case, Cliff's δ values are preferred to demonstrate the strength of one algorithm over the other since it is not sensitive to distributional scale or extreme

values. Given that the HBA results for this test function are non-normal, Cliff's δ offers a more reliable estimate of effect size. Overall, the Cliff's δ results emphasize EnSO's strength in most functions.

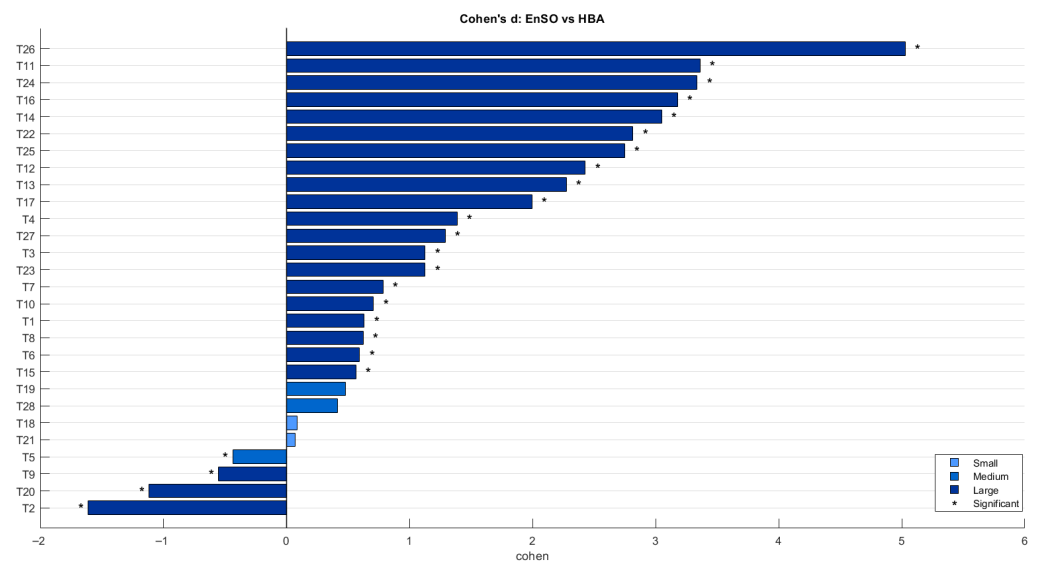


Figure 6. Cohen's d effect sizes for EnSO vs. HBA across CEC 2013 functions. Positive values indicate better performance by EnSO.

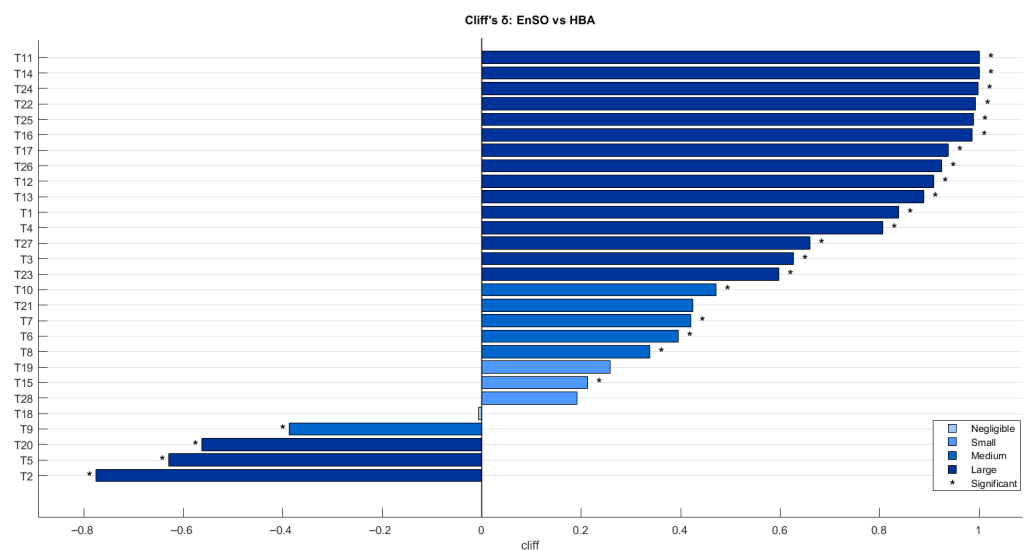


Figure 7. Cliff's δ effect sizes for EnSO vs. HBA across CEC 2013 functions. Positive values indicate EnSO dominance.

The combination of significance and effect sizes allows us to distinguish functions with moderate improvements, functions with very strong superiority, and the relative consistency of these differences, which was not visible from p -values alone. This gives a more complete and nuanced picture of algorithm performance than significance testing alone.

5.3.3. EnSO vs. COA (CEC 2011)

Figure 8 shows that with Cohen's d , EnSO demonstrates a small advantage over COA on T_3 , whereas for most other benchmarks (T_2 , T_4 , T_5) the effect sizes are large and negative, reflecting COA's consistent superiority. An exception is T_1 which is positive but non-significant and small. This indicates that, in general, COA outperforms EnSO on these CEC 2011 benchmarks.

As depicted in Figure 9, Cliff's δ confirms the general pattern observed with Cohen's d . For T_1 , δ indicates a medium yet non-significant effect favoring EnSO, whereas for the remaining functions, it highlights large effects in favor of COA. The differences in magnitude between Cohen's d and Cliff's δ reflect the sensitivity of Cohen's d to the scale of the data, while Cliff's δ focuses on the relative ordering of samples, which is particularly informative when distributions are non-normal, such as in this case.

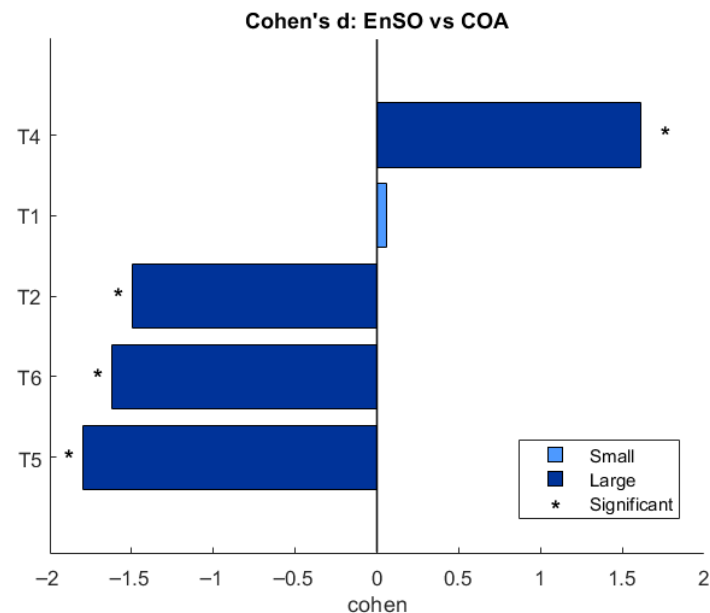


Figure 8. Cohen's d effect sizes for EnSO vs. COA across CEC 2011 functions. Positive values indicate EnSO dominance.

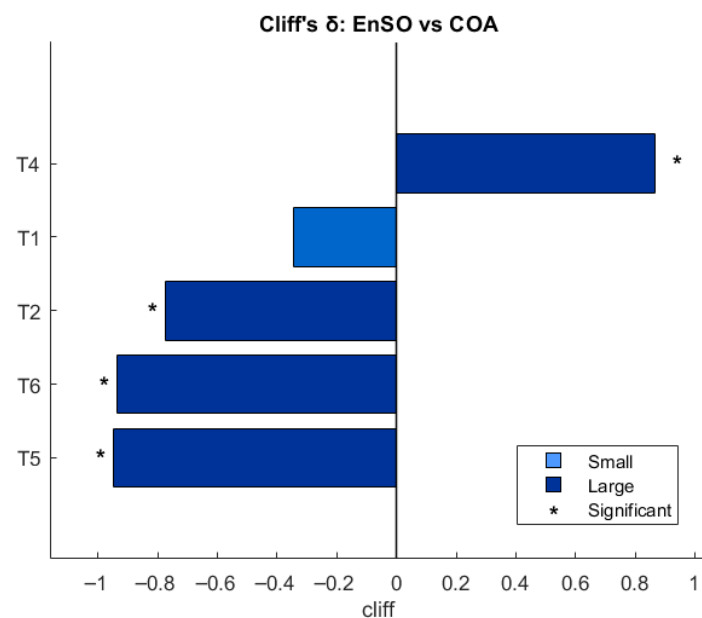


Figure 9. Cliff's δ effect sizes for EnSO vs. COA across CEC 2011 functions. Positive values indicate EnSO dominance.

5.3.4. EnSO vs. HBA (CEC 2011)

In Figure 10, Cohen's d indicates that EnSO shows large superiority over HBA for T_1 and T_5 , with only T_1 being a significant result, while the effect is non-significant and small for T_3 , making either of the algorithms equally suitable. The large significant negative effect

on T_2 suggests HBA outperforms EnSO, highlighting that algorithm performance is highly problem-dependent in these benchmarks.

Cliff's δ in Figure 11 shows a large effect for T_1 favoring EnSO, and small-to-large effects for other functions. Yet, only T_2 is significant, similarly to d , highlighting that algorithm superiority is problem-dependent. Given that most of the CEC 2011 samples are non-normal, Cliff's δ provides a more reliable estimate of effect size in these cases, reinforcing the importance of considering distributional characteristics when interpreting results.

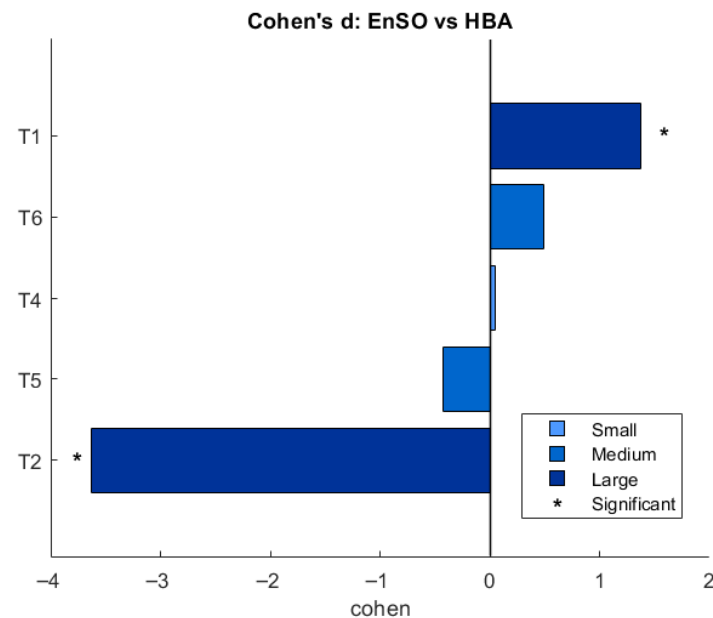


Figure 10. Cohen's d effect sizes for EnSO vs. HBA across CEC 2011 functions. Positive values indicate EnSO dominance.

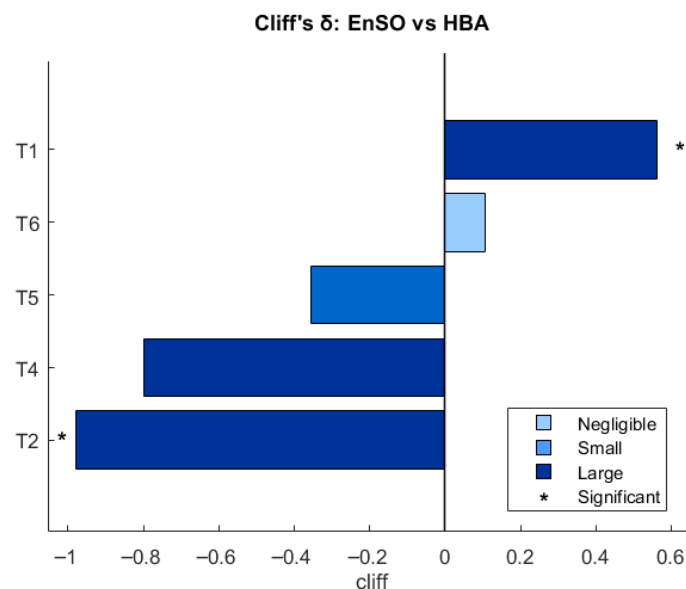


Figure 11. Cliff's δ effect sizes for EnSO vs. HBA across CEC 2011 functions. Positive values indicate EnSO dominance.

5.4. Overall Effect Size

Table 12 summarizes the median magnitude, and prevalence of meaningful effect sizes across all CEC 2013 benchmark functions. Median values for each metric signify the degree by which an algorithm performed better, with the sign denoting the direction of

dominance. For EnSO vs. COA, the median value of Cohen's d reflects a medium advantage for EnSO, whereas Cliff's δ suggests a large advantage instead for EnSO. The proportion of positive values labeled medium or higher for both measures indicates how often EnSO performed better. In this case, EnSO dominates the majority of the benchmark functions, irrespective of function type. When only statistically significant cases are considered, the proportion of medium-or-larger effects increases, and the median effect sizes also increase, showing that significant differences are more likely to be practically meaningful, and further demonstrating EnSO's superiority to COA.

Table 12. Average effect size values and percentages for CEC 2013. * Refers to significant values only.

Effect Size	Median	Median *	Values $\geq$ Medium (%)	Values $\geq$ Medium (%) *
Cohen's d (EnSO vs. COA)	0.4959	1.2157	60.7142	63.6363
Cliff's δ (EnSO vs. COA)	0.4844	0.6667	53.5714	68.1818
Cohen's d (EnSO vs. HBA)	0.9569	1.2100	78.5714	83.3333
Cliff's δ (EnSO vs. HBA)	0.6122	0.7333	71.4286	79.1667

For EnSO versus HBA, both effect size measures are large and positive, reflecting a strong and consistent dominance of EnSO over HBA across the benchmark functions. The relatively high percentages of positive medium-or-larger effects show that instances where HBA outperforms EnSO are rare, highlighting EnSO as the most suitable choice for most CEC 2013 benchmarks. Overall, the table highlights how effect size summaries complement p-values by revealing the magnitude and consistency of differences across functions.

Likewise, Table 13 summarizes the median magnitude and prevalence of meaningful effect sizes across selected CEC 2011 benchmark functions. For EnSO versus COA, both Cohen's d and Cliff's δ are generally large and negative. EnSO outperforms COA in only 20% of all cases, and in 25% of cases when considering only statistically significant results, highlighting COA as the superior algorithm for the majority of CEC 2011 functions.

Table 13. Average effect size values and percentages for CEC 2011. * Refers to significant values only.

Effect Size	Median	Median *	Values $\geq$ Medium (%)	Values $\geq$ Medium (%) *
Cohen's d (EnSO vs. COA)	−1.4930	−1.5555	20.0000	25.0000
Cliff's δ (EnSO vs. COA)	−0.7733	−0.8544	20.0000	25.0000
Cohen's d (EnSO vs. HBA)	0.0525	−1.1270	40.0000	50.0000
Cliff's δ (EnSO vs. HBA)	−0.3556	−0.2089	20.0000	50.0000

As for EnSO vs. HBA, the median value for Cohen's d is positive but very small, indicating that there is no meaningful difference between the two algorithms. Yet when only significant results are considered, HBA is shown to outperform EnSO greatly. In contrast, Cliff's δ indicates a medium effect overall, and only a small effect in significant cases, further highlighting the importance of reporting effect sizes alongside significance tests to accurately assess algorithmic differences. Given that nearly all samples from the CEC 2011 benchmarks are non-normally distributed, the effect sizes indicated by Cliff's δ provide a more reliable assessment of algorithmic differences.

6. Practical Tutorial: Using Effect Size Measures in Metaheuristics

To facilitate the adoption of effect size measures in comparative studies of meta-heuristic algorithms, this section provides a step-by-step tutorial on selecting, computing, and interpreting effect sizes alongside statistical significance tests.

6.1. Step 1: Assess the Nature of Your Data

Before choosing an effect size measure, it is important to evaluate your performance data. Start by checking whether the data is normally distributed using tests such as Shapiro–Wilk or by visually inspecting histograms. You should also assess the presence of outliers and skewness, as non-normal distributions, skewed data, or extreme values can influence the suitability and interpretation of parametric effect sizes.

6.2. Step 2: Choose the Appropriate Effect Size Measure

The choice of effect size measure should be guided by the characteristics of your data. Cohen’s d is appropriate when the data is approximately normally distributed and group variances are similar [47], as it provides a standardized measure of the mean difference. In contrast, Cliff’s δ is better suited for non-normal or skewed data, since it quantifies dominance without relying on distributional assumptions. Table 14 simplifies it further.

Table 14. Appropriate effect size per comparison.

Comparison Type	Recommended Effect Size
Normal vs. Normal	Cohen’s d
Normal vs. Non-normal	Cliff’s δ
Non-normal vs. Non-normal	Cliff’s δ

6.3. Step 3: Decide the Direction of Comparison

Since metaheuristics are typically evaluated on minimization problems, the ordering of comparison groups should be chosen carefully. The first group should correspond to the algorithm of interest, so that positive effect size values consistently indicate the superiority of the first algorithm over the second.

6.4. Step 4: Compute Statistical Significance

Before interpreting effect sizes, it is important to ensure the validity of the statistical analysis. If the data violate parametric assumptions, non-parametric tests such as permutation tests should be used. Empirical p -values should then be computed to assess whether the observed differences are likely due to chance. When more than two algorithms are compared, appropriate multiple comparison corrections, such as the Bonferroni adjustment [45], should be applied to control for inflated Type I error.

6.5. Step 5: Compute Effect Sizes

Effect sizes should be computed using the measure selected in the previous step. For Cohen’s d , apply Equations (1) and (2), and interpret the resulting values using the guidelines in Table 1. For Cliff’s δ , use Equation (3) and interpret the magnitude of the effect according to Table 2.

6.6. Step 6: Interpretation

Interpretation should integrate both statistical significance and effect size magnitude. Table 15 summarizes how to interpret significance and effect size metrics together.

Table 15. Guidelines for interpreting statistical significance and effect size.

Statistical Result	Interpretation
Significant + medium/large effect	Strong evidence of a practically meaningful performance difference.
Significant + negligible/small effect	Statistically detectable difference with limited practical relevance.
Not significant + medium/large effect	Potentially meaningful difference; more runs may be required.
Not significant + negligible/small effect	No meaningful performance difference detected.

6.7. Step 7: Reporting Recommendations

When reporting results, provide both p -values or significance results and effect sizes for all pairwise comparisons. Clearly specify the type of effect size used and the interpretation thresholds applied. Additionally, include summary tables or visualizations to convey algorithm performance across all benchmark functions effectively.

This tutorial provides a standardized framework that ensures empirical results are both statistically sound and practically meaningful, facilitating more reliable conclusions in metaheuristics research.

7. Conclusions

This paper highlights the need to move beyond relying solely on statistical significance when comparing metaheuristic methods. Tests can tell us whether a difference is real, but not how meaningful it is. Effect size measures such as Cohen's d and Cliff's δ help fill this gap by showing the actual strength of the difference. They offer clearer insight into how much better one method is compared to another, and their use remains rare in our field. By encouraging researchers to include effect sizes alongside traditional tests, we aim to make empirical comparisons more transparent, informative, and easier to interpret.

Author Contributions: Conceptualization, M.G.H.O.; methodology, M.G.H.O.; software, N.M.H.I.; writing—original draft preparation, N.M.H.I.; writing—review and editing, M.G.H.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable feedback, which significantly improved the clarity and rigor of this work.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EnSO	Ensemble of Simple Optimizers
COA	Coyote Optimization Algorithm
HBA	Honey Badger Algorithm
CEC	Congress of Evolutionary Computation
T_i	i -th test function
SD	Standard Deviation

References

1. Singh, P.; Choudhary, S.K. Introduction: Optimization and metaheuristics algorithms. In *Metaheuristic and Evolutionary Computation: Algorithms and Applications*; Springer: Singapore, 2021; pp. 3–33. [\[CrossRef\]](#)
2. Gendreau, M.; Potvin, J.Y. (Eds.) *Handbook of Metaheuristics*; Springer: New York, NY, USA, 2010. [\[CrossRef\]](#)
3. Alain, P.; Sana, B.H. *Evolutionary Algorithms*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2017; pp. 1–32. [\[CrossRef\]](#)
4. Amin, A.A.H.; Aladdin, A.M.; Hasan, D.O.; Mohammed-Taha, S.R.; Rashid, T.A. Enhancing Algorithm Selection through Comprehensive Performance Evaluation: Statistical Analysis of Stochastic Algorithms. *Computation* **2023**, *11*, 231. [\[CrossRef\]](#)
5. Sullivan, G.M.; Feinn, R. Using Effect Size—Or Why the P Value Is Not Enough. *J. Grad. Med Educ.* **2012**, *4*, 279–282. [\[CrossRef\]](#)
6. Omran, M.G.H.; Clerc, M.; Ghaddar, F.; Aldabagh, A.; Tawfik, O. Permutation tests for metaheuristic algorithms. *Mathematics* **2022**, *10*, 2219. [\[CrossRef\]](#)
7. Ramadas, G.C.V.; Fernandes, E.M.G.P.; Ramadas, A.M.V.; Rocha, A.M.A.C.; Costa, M.F.P. On Metaheuristics for Solving the Parameter Estimation Problem in Dynamic Systems: A Comparative Study. *J. Optim.* **2018**, *2018*, 3213484. [\[CrossRef\]](#)
8. Güven, A.F.; Yörükeren, N.; Mengi, O.O. Multi-objective optimization and sustainable design: A performance comparison of metaheuristic algorithms used for on-grid and off-grid hybrid energy systems. *Neural Comput. Appl.* **2024**, *36*, 7559–7594. [\[CrossRef\]](#)
9. Ronald, S.D.; Erik, M.A.; Juan, E.C. Statistical Methodology for Quality Comparison between Algorithms. *Int. J. Adv. Res. Comput. Eng. Technol.* **2013**, *3*, 469–472. Available online: https://www.ijera.com/papers/Vol3_issue6/CB36469472.pdf (accessed on 27 February 2026).
10. Vermetten, D.; Wang, H.; López-Ibañez, M.; Doerr, C.; Bäck, T. Analyzing the Impact of Undersampling on the Benchmarking and Configuration of Evolutionary Algorithms. In Proceedings of the GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference, Boston, MA, USA, 13–17 July 2022. [\[CrossRef\]](#)
11. Campelo, F.; Takahashi, F. Sample size estimation for power and accuracy in the experimental comparison of algorithms. *J. Heuristics* **2018**, *25*, 305–338. [\[CrossRef\]](#)
12. Dillen, W.; Lombaert, G.; Schevenels, M. Performance Assessment of Metaheuristic Algorithms for Structural Optimization Taking Into Account the Influence of Algorithmic Control Parameters. *Front. Built Environ.* **2021**, *7*, 618851. [\[CrossRef\]](#)
13. Eftimov, T.; Korošec, P.; Seljak, B.K. Deep statistical comparison of meta-heuristic stochastic optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*; GECCO '18; ACM: New York, NY, USA, 2018; pp. 15–16. [\[CrossRef\]](#)
14. Omran, M.; Salman, A.; Clerc, M. Ensemble of simple optimizers. *Memetic Comput.* **2025**, *17*, 41. [\[CrossRef\]](#)
15. Alorf, A. A survey of recently developed metaheuristics and their comparative analysis. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105622. [\[CrossRef\]](#)
16. Vadi, S.; Bildirici, M.; Kaplan, O. Performance Comparison of Metaheuristic and Hybrid Algorithms Used for Energy Cost Minimization in a Solar–Wind–Battery Microgrid. *Sustainability* **2025**, *17*, 8849. [\[CrossRef\]](#)
17. Pierezan, J.; Dos Santos Coelho, L. Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–8. [\[CrossRef\]](#)
18. Hassan, I.H.; Abdullahi, M.; Isuwa, J.; Yusuf, S.A.; Aliyu, I.T. A comprehensive survey of honey badger optimization algorithm and meta-analysis of its variants and applications. *Frankl. Open* **2024**, *8*, 100141. [\[CrossRef\]](#)
19. Kopciwicz, P.; Łukasik, S. Exploiting flower constancy in flower pollination algorithm: Improved biotic flower pollination algorithm and its experimental evaluation. *Neural Comput. Appl.* **2019**, *32*, 11999–12010. [\[CrossRef\]](#)
20. da Silva, L.S.A.; Lúcio, Y.L.S.; Coelho, L.d.S.; Mariani, V.C.; Rao, R.V. A comprehensive review on Jaya optimization algorithm. *Artif. Intell. Rev.* **2022**, *56*, 4329–4361. [\[CrossRef\]](#)
21. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physicsbased optimization. *Neurocomputing* **2023**, *532*, 183–214. [\[CrossRef\]](#)
22. Colak, M.; Bingol, O.S.; Dayi, A. Self-esteem and social media addiction level in adolescents: The mediating role of body image. *Indian J. Psychiatry* **2023**, *65*, 595–600. [\[CrossRef\]](#)
23. Andrews, D.; Bonta, J. *The Psychology of Criminal Conduct*; Routledge: London, UK, 2010. [\[CrossRef\]](#)
24. Gignac, G.E.; Szodorai, E.T. Effect size guidelines for individual differences researchers. *Personal. Individ. Differ.* **2016**, *102*, 74–78. [\[CrossRef\]](#)
25. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Routledge: London, UK, 1988. [\[CrossRef\]](#)
26. Pearson, K., VII. Note on regression and inheritance in the case of two parents. *Proc. R. Soc.* **1895**, *58*, 240–242. [\[CrossRef\]](#)
27. Goldberg, T.E.; Lee, S.; Devanand, D.P.; Schneider, L.S. Comparison of relative change with effect size metrics in Alzheimer's disease clinical trials. *J. Neurol. Neurosurg. Psychiatry* **2023**, *95*, 2–7. [\[CrossRef\]](#)
28. Rothwell, J.C.; Julious, S.A.; Cooper, C.L. A study of target effect sizes in randomised controlled trials published in the Health Technology Assessment journal. *Trials* **2018**, *19*, 544. [\[CrossRef\]](#) [\[PubMed\]](#)

29. Masino, N.; Quintero-Rincon, A. Effect sizes as a statistical feature-selector-based learning to detect breast cancer. *arXiv* **2024**, arXiv:2411.06868. [\[CrossRef\]](#)
30. Jollans, L.; Boyle, R.; Artiges, E.; Banaschewski, T.; Desrivieres, S.; Grigis, A.; Martinot, J.L.; Paus, T.; Smolka, M.N.; Walter, H.; et al. Quantifying performance of machine learning methods for neuroimaging data. *NeuroImage* **2019**, *199*, 351–365. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Santos, F.; Vargovich, J.; Trinkenreich, B.; Santos, I.; Penney, J.; Britto, R.; Pimentel, J.F.; Wiese, I.; Steinmacher, I.; Sarma, A.; et al. Tag that issue: Applying API-domain labels in issue tracking systems. *arXiv* **2023**, arXiv:2304.02877. [\[CrossRef\]](#)
32. Rajput, D.; Wang, W.J.; Chen, C.C. Evaluation of a decided sample size in machine learning applications. *BMC Bioinform.* **2023**, *24*, 48. [\[CrossRef\]](#)
33. Baguley, T. Standardized or simple effect size: What should be reported? *Br. J. Psychol.* **2009**, *100*, 603–617. [\[CrossRef\]](#)
34. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [\[CrossRef\]](#)
35. Cliff, N. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychol. Bull.* **1993**, *114*, 494–509. [\[CrossRef\]](#)
36. Meissel, K.; Yao, E.S. Using Cliff’s Delta as a Non-Parametric Effect Size Measure: An Accessible Web App and R Tutorial. *Pract. Assess. Res. Eval.* **2024**, *29*, 2. [\[CrossRef\]](#)
37. Liang, J.; Qu, B.; Suganthan, P.; Hernández-Díaz, A. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *PeerJ* **2013**, *34*, 281–295.
38. Loshchilov, I. CMA-ES with restarts for solving CEC 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*; IEEE: Piscataway, NJ, USA, 2013; pp. 369–376. [\[CrossRef\]](#)
39. Liao, T.; Stützle, T. Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation*; IEEE: Piscataway, NJ, USA, 2013; pp. 1938–1944. [\[CrossRef\]](#)
40. Al-Roomi, A.R. *IEEE Congresses on Evolutionary Computation Repository*; Dalhousie University, Electrical and Computer Engineering; Halifax, NS, Canada, 2015.
41. Agushaka, J.O.; Akinola, O.; Ezugwu, A.E.; Oyelade, O.N.; Saha, A.K. Advanced dwarf mongoose optimization for solving CEC 2011 and CEC 2017 benchmark problems. *PLoS ONE* **2022**, *17*, e0275346. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Singh, H.K.; Ray, T. Performance of a hybrid EA-DE-memetic algorithm on CEC 2011 real world optimization problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)*; IEEE: Piscataway, NJ, USA, 2011; pp. 1322–1326. [\[CrossRef\]](#)
43. Wagdy, A.; Hadi, A.; Khater, A. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1501–1529. [\[CrossRef\]](#)
44. Edgington, E.; Edgington, E.; Onghena, P. *Randomization Tests*; Informa; CRC: Boca Raton, FL, USA, 2007. [\[CrossRef\]](#)
45. Armstrong, R.A. When to use the Bonferroni correction. *Ophthalmic Physiol. Opt.* **2014**, *34*, 502–508. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Shapiro, S.S.; Wilk, M.B. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* **1965**, *52*, 591–611. [\[CrossRef\]](#)
47. Lakens, D. Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Front. Psychol.* **2013**, *4*, 863. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.