Supplementary Materials: An Improved Differential Evolution Algorithm Using Mahalanobis Distance and Cholesky Decomposition for Wrapper-based Feature Selection

This supplementary document provides additional methodological information and extended experimental results supporting the approach proposed in the manuscript.

Supplementary Note S1: An examination on different distance measures

In light of the considerations, the most salient distances for consideration are subsequently delineated. To begin with, the Euclidean distance is regarded as one of the most fundamental measures. This is due to the fact that it denotes the geometric distance between two points in a space of dimension d. This approach is based on the assumption that all features contribute equally and independently, thereby providing a simple method for quantifying similarity. The following Eq. S1 is employed to calculate the Euclidean distance between two points, x and x'.

$$d_E(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2} \tag{S1}$$

On the other hand, the Manhattan distance, also known as the L_1 norm, measures the sum of absolute differences between feature components. It is particularly robust in high-dimensional or sparse data scenarios, as it reduces the influence of outliers compared to the Euclidean distance. It is defined in the following Eq. S2.

$$d_{Man}(x, x') = \sum_{i=1}^{n} |x_i - x'_i| \tag{S2}$$

The Chebyshev distance focuses on the largest individual deviation among dimensions, emphasizing the maximum difference between feature values (see Eq. S3). This approach is useful when the greatest component-wise discrepancy determines dissimilarity.

$$d_C(x, x') = \max_i |x_i - x'_i| \tag{S3}$$

Finally, the Minkowski distance provides a generalization that encompasses both the Euclidean and Manhattan distances through the parameter p, which controls the order of the norm. This formulation, as shown in Eq. S4, allows flexible adaptation depending on the data distribution and the degree of sensitivity to large deviations.

$$d_{Min}(x, x') = \left(\sum_{i=1}^{n} |x_i - x'_i|^p \right)^{1/p} \tag{S4}$$

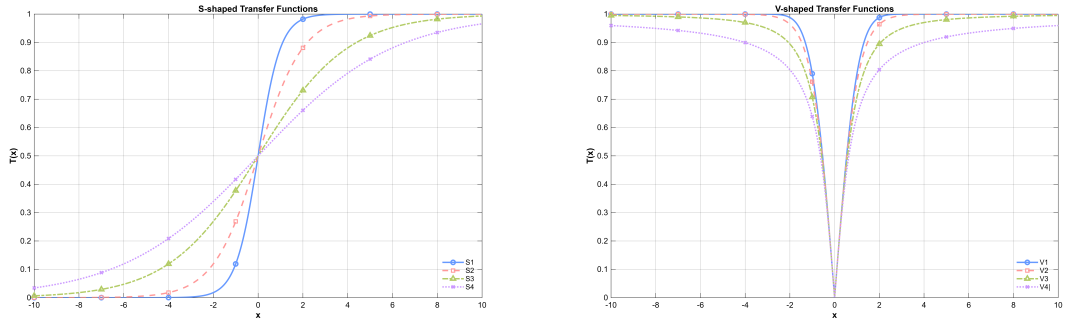
All these distances, though derived from distinct mathematical formulations, share the purpose of quantifying similarity or dissimilarity between feature vectors. Their selection depends on the underlying data characteristics, the computational cost, and the desired robustness to noise, each influencing the overall performance of typical applications in clustering, classification, and feature selection. Table S1 summarizes all the mathematical formulas for distances used in the manuscript.

Table S1. Summary of distance metrics analyzed

Distance	Equation	Description
Euclidean	$d_E(x,x') = \sqrt{\sum_{i=1}^{n}(x_i - x_i')^2}$	Measures the straight-line (geometric) distance between two points in a multidimensional space, assuming equal and independent contributions from all features.
Mahalanobis	$d_M(x,x') = \sqrt{(x-x')^\top \Sigma^{-1}(x-x')}$	Accounts for correlations among variables and normalizes feature scales using the covariance matrix; reduces to Euclidean distance when $\Sigma = I$.
Manhattan	$d_{Man}(x,x') = \sum_{i=1}^{n} \lvert x_i - x_i' \rvert$	Computes the sum of absolute differences between feature values; more robust to outliers and high-dimensional sparsity.
Chebyshev	$d_C(x,x') = \max_i \lvert x_i - x_i' \rvert$	Considers the maximum absolute difference between dimensions and highlights the most dominant deviation across features.
Minkowski	$d_{Min}(x,x') = \left(\sum_{i=1}^{n} \lvert x_i - x_i' \rvert^p\right)^{1/p}$	Generalizes to include Euclidean ($p = 2$) and Manhattan ($p = 1$) as special cases; allows control of sensitivity via the order p.

Supplementary Note S2: Analysis of various transfer functions

In this study, eight transfer functions are utilized to assess the proposed method. Four of these functions belong to the S-shaped family (S1-S4). The remaining four are classified as part of the V-shaped family (V1-V4). As demonstrated in Fig. S1, the graphical behaviors of the transfer functions are represented by their respective equations, summarized in Table S2. Transfer functions have been proposed to identify the most suitable basis for implementing the feature selection proposal.



(**a**) S-shaped (**b**) V-shaped

Figure S1. Transfer functions for binarization.

Table S2. S-shaped and V-shaped transfer functions.

S-shaped family	Transfer function	V-shaped family	Transfer function
S1	$T(x) = \frac{1}{1+e^{-2x}}$	V1	$T(x) = \left\lvert \mathrm{erf}\left(\sqrt{\frac{\pi}{2}}x\right)\right\rvert$
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \lvert \tanh(x) \rvert$
S3	$T(x) = \frac{1}{1+e^{-\frac{x}{2}}}$	V3	$T(x) = \left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$
S4	$T(x) = \frac{1}{1+e^{-\frac{x}{3}}}$	V4	$T(x) = \left\lvert \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right\rvert$

According to this process, the continuous positions of individuals in the population will be converted into binary values using the rules defined in the following Eq. S5 and Eq. S6, for S-shaped and V-shaped families, respectively. For the S-shaped family, a feature is selected when the uniformly distributed random number is less than the value of the transfer function; otherwise, it is discarded. For the V-shaped family, the current binary

value is complemented when the random number is less than the value of the transfer function; otherwise, it remains unchanged.

$$x_{i,d}^{g+1} = \begin{cases} 1, & \text{if } rand(0,1) < T(v_{i,d}^{g+1}), \\ 0, & \text{otherwise.} \end{cases} \tag{S5}$$

$$x_{i,d}^{g+1} = \begin{cases} 1 - x_{i,d}^{g}, & \text{if } rand(0,1) < T(v_{i,d}^{g+1}), \\ x_{i,d}^{g}, & \text{otherwise.} \end{cases} \tag{S6}$$

Supplementary Note S3: Overview of the classification models

The k-nearest neighbor classifier

The K-Nearest Neighbor classifier (KNN) is one of the most well-known and widely used supervised learning algorithms in the relevant literature for classification purposes. KNN endeavors to identify the closest training neighbors to the test sample based on the Euclidean distance. Subsequently, the test sample is classified based on the majority class of its closest neighbors. In KNN, the only parameter that must be determined is the number of neighbors (k) to be considered during the classification of a new test sample. In the context of machine learning, the classification of a sample to a specific category depends on the majority of its proximate neighbors in the feature space being classified into that category. The parameter k denotes the number of nearest neighbors that must be considered for determining the category to which a sample belongs. In the KNN algorithm, the selected neighbors are all objects that have been classified correctly. As demonstrated in Eq. S7, the distance is determined by quantifying the geometric separation between the feature vectors.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \tag{S7}$$

Subsequent to the identification of the nearest neighbors, the final class assignment is determined by a majority vote. Eq. S8 shows the manner in which the predicted class corresponds to the most frequent label among the k nearest neighbors.

$$\hat{y}(x) = \arg\max_{c} \sum_{i \in N_k(x)} I(y_i = c) \tag{S8}$$

Support vector machine classifier

The Support Vector Machine (SVM) is a machine learning model that is particularly adept at handling classification and regression problems. The model operates under the assumption that instances in the input space that are proximate to one another must possess identical output values. This process involves calculating an optimal hyperplane that effectively separates the classes in a multidimensional space. The class of a new instance is determined by its position on the same side of the hyperplane.

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \tag{S9}$$

As illustrated in Eq. S9, the constraints guarantee that correctly classified samples remain outside the margin boundaries, while the margin variables enable a regulated number of classification errors.

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{S10}$$

From Eq. S10, w denotes the weight vector defining the separating hyperplane, b is the bias term, C controls the balance between margin maximization and classification error, and x_i denotes the margin variables associated with misclassified samples. The KNN algorithm was configured with $k = 5$ uniform neighbor weights. The SVM was implemented using a radial basis function (RBF) kernel with $C = 1.0$ and the default scale-dependent value of γ. The classifiers were implemented within the MAFESE framework. No additional tuning of the classifiers' hyperparameters was performed, and the same configurations were used for all feature selection algorithms to ensure a fair comparison.

Supplementary Note S4: Detailed validation protocol for wrapper-based feature selection

An external validation protocol was adopted to clearly separate the feature selection phase from the final evaluation of the classification model's performance. For each dataset, the entire sample set was first divided into external training and test partitions in an 80/20 split. This split was generated using a fixed random seed and, whenever the class distribution allowed it, stratification was applied based on class labels. If stratification was not feasible, the same holdout ratio was used without stratification. The external test set was completely excluded from the optimization process and reserved exclusively for the final evaluation. Conversely, wrapper-based feature selection was performed exclusively on the external training partition. During the optimization process, this partition was divided internally into training and validation subsets using an additional holdout strategy. The internal training subset was used to tune the classifier for each candidate feature subset, and the internal validation subset was used to estimate classification performance as measured by the objective function. Consequently, the feature selection process followed a two-level holdout protocol rather than a k-fold cross-validation scheme, and the external test data were not used to guide the search process. For each candidate solution, the selected features were used to train the classifier on the internal training subset, and its predictive performance was evaluated using the internal validation subset. Classification accuracy was combined with the proportion of selected features to obtain the objective function values. Validation performance was used exclusively to guide the optimization process, with the external test set remaining untouched throughout. Once this was complete, the final classification performance was expressed in terms of Accuracy, F1-Score, Precision, and Recall.

Supplementary Note S5: Description of the CEC-2017 benchmark suite

Research on single-objective optimization algorithms has served as a foundational basis for the development of more complex algorithms that have been executed and applied to both real-world engineering problems and certain fields of machine learning. In recent years, there has been a growing tendency to test new algorithms and methodologies on single-objective reference problems. In order to evaluate the capabilities of the proposal outlined in this manuscript, it is necessary to consider the set of benchmark tests from the 2017 edition of the Congress on Evolutionary Computation (CEC). The set of benchmark tests under consideration comprises 29 independent optimization benchmark functions, grouped into the following categories: unimodal, multimodal, hybrid, and composite. This benchmark suite has been utilized for over a decade to illustrate the versatility and performance of optimization algorithms. A comprehensive overview of these functions is provided in Table S3, and a more detailed exposition of this information can be found in the original proposal in the following reference.

Table S3. Outline of the CEC-2017 benchmark functions

	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Zakharov Function	200
Simple Multimodal Functions	3	Shifted and Rotated Rosenbrock's Function	300
	4	Shifted and Rotated Rastrigin's Function	400
	5	Shifted and Rotated Expanded Schaffer's F6 Function	500
	6	Shifted and Rotated Lunacek Bi_Rastrigin Function	600
	7	Shifted and Rotated Non-Continuous Rastrigin's Function	700
	8	Shifted and Rotated Levy Function	800
	9	Shifted and Rotated Schwefel's Function	900
Hybrid Functions	10	Hybrid Function 1 (N=3)	1000
	11	Hybrid Function 2 (N=3)	1100
	12	Hybrid Function 3 (N=3)	1200
	13	Hybrid Function 4 (N=4)	1300
	14	Hybrid Function 5 (N=4)	1400
	15	Hybrid Function 6 (N=4)	1500
	16	Hybrid Function 7 (N=5)	1600
	17	Hybrid Function 8 (N=5)	1700
	18	Hybrid Function 9 (N=5)	1800
	19	Hybrid Function 10 (N=6)	1900
Composition Functions	20	Composition Function 1 (N=3)	2000
	21	Composition Function 2 (N=3)	2100
	22	Composition Function 3 (N=4)	2200
	23	Composition Function 4 (N=4)	2300
	24	Composition Function 5 (N=5)	2400
	25	Composition Function 6 (N=5)	2500
	26	Composition Function 7 (N=6)	2600
	27	Composition Function 8 (N=6)	2700
	28	Composition Function 9 (N=3)	2800
	29	Composition Function 10 (N=3)	2900
Search Range: $[-100, 100]^D$			

Supplementary Note S6: Extended comparative results of the CEC-2017 benchmark using different distance measures

A preliminary study was conducted to evaluate how different distance measures influence the behavior of MaCRO-DE. This was done before carrying out the full comparative study with other metaheuristic algorithms. Five distance measures were considered: Chebyshev, Euclidean, Mahalanobis-Cholesky, Manhattan, and Minkowski. The analysis used representative functions from the CEC-2017 benchmark suite with dimension $D = 30$. This experiment aimed to determine how using different distance metrics to identify neighboring candidate solutions affects the search process and the resulting optimization performance. The numerical results reveal that the choice of distance metric notably affects the behavior of MaCRO-DE; however, no specific metric consistently outperforms the others across all benchmark functions (see Table S4). Mahalanobis-Cholesky achieved the best average performance for function 10 and remained highly competitive for F3, F8, and F24, obtaining the second-best average values for these functions. Competitive performance was also observed for F15 and F29. Conversely, other distance measures perform better on specific functions. For instance, Chebyshev performs particularly well on F3, F8, F17, and F24, while Manhattan offers the best average performance on F15 and F29. These results suggest that the effectiveness of each metric depends on the characteristics of the optimization problem's landscape.

Despite the absence of a universal domain, the Mahalanobis-Cholesky formulation is still relevant to the proposed methodology because it considers the covariance structure of the population. Unlike coordinate-based distance measures, this formulation considers correlations among decision variables and transforms the search geometry via Cholesky decomposition. Consequently, the neighborhood used during mutation is defined by the current distribution and orientation of the population rather than by absolute coordinate

differences alone. This property provides an adaptive mechanism for identifying candidate solutions that are geometrically related to the current search region.

The convergence behavior of the five distance-based variants is illustrated in Fig. S2. Functions F1, F3, F8, F10, F15, F17, F24, and F29 illustrate the different convergence patterns observed during the experiment. Several functions, such as F3, F8, and F24, converge to similar final regions. However, clearer differences are observed among the distance measures for F1, F10, F15, F17, and F29. Notably, the Mahalanobis-Cholesky variant exhibits competitive convergence with respect to F10 and remains close to the best-performing alternatives for several other functions. Overall, these results confirm that modifying the distance metric directly influences MaCRO-DE's exploration and exploitation behavior. Therefore, the Mahalanobis-Cholesky distance is retained in the proposed algorithm for its competitive empirical performance and ability to incorporate population covariance-dependent geometry into the search process.

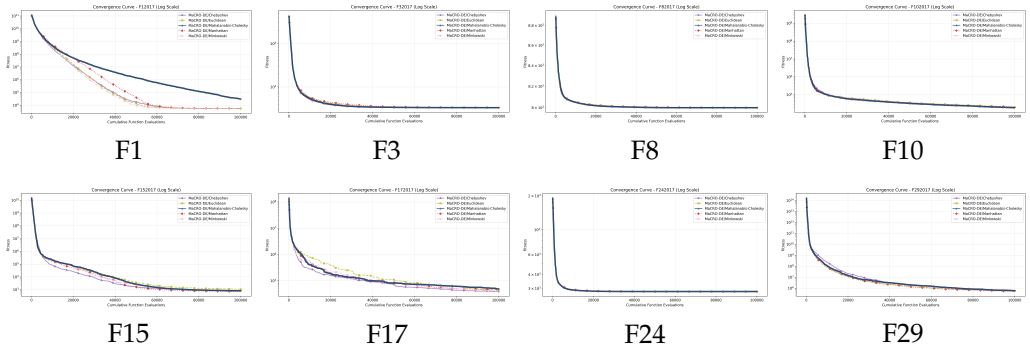


Figure S2. Convergence behavior of MaCRO-DE using Chebyshev, Euclidean, Mahalanobis-Cholesky, Manhattan, and Minkowski distance measures on representative CEC-2017 benchmark functions with $D = 30$.

Table S4. Comparison of Distance Measures for MaCRO-DE on Representative CEC-2017 Functions

Function	Statistic	MaCRO-DE/Chebyshev	MaCRO-DE/Euclidean	MaCRO-DE/Mahalanobis-Cholesky	MaCRO-DE/Manhattan	MaCRO-DE/Minkowski
F1	Best	1.03×10^{2}	5.55×10^{3}	1.06×10^{4}	3.83×10^{3}	2.39×10^{3}
	Worst	2.09×10^{4}	5.75×10^{3}	4.79×10^{4}	6.36×10^{3}	5.67×10^{3}
	Mean	5.51×10^{3}	5.59×10^{3}	2.99×10^{4}	5.55×10^{3}	5.48×10^{3}
	Std	3.32×10^{3}	3.22×10^{1}	9.07×10^{3}	3.55×10^{2}	5.84×10^{2}
F3	Best	3.32×10^{2}	3.32×10^{2}	3.32×10^{2}	3.32×10^{2}	3.32×10^{2}
	Worst	3.42×10^{2}	3.45×10^{2}	3.34×10^{2}	3.50×10^{2}	3.52×10^{2}
	Mean	3.33×10^{2}	3.33×10^{2}	3.32×10^{2}	3.35×10^{2}	3.33×10^{2}
	Std	2.43×10^{0}	2.67×10^{0}	5.35×10^{-1}	4.24×10^{0}	3.54×10^{0}
F8	Best	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}
	Worst	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}
	Mean	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}	8.00×10^{2}
	Std	2.20×10^{-7}	2.56×10^{-4}	4.08×10^{-8}	1.92×10^{-2}	5.89×10^{-5}
F10	Best	1.16×10^{4}	1.36×10^{4}	9.74×10^{3}	1.51×10^{4}	7.97×10^{3}
	Worst	2.80×10^{4}	3.10×10^{4}	2.92×10^{4}	2.74×10^{4}	3.44×10^{4}
	Mean	1.95×10^{4}	2.07×10^{4}	1.89×10^{4}	2.06×10^{4}	1.91×10^{4}
	Std	4.21×10^{3}	5.26×10^{3}	5.35×10^{3}	3.47×10^{3}	6.27×10^{3}
F15	Best	6.36×10^{2}	6.32×10^{2}	6.31×10^{2}	6.32×10^{2}	6.33×10^{2}
	Worst	1.42×10^{3}	1.03×10^{4}	1.68×10^{3}	9.58×10^{2}	2.00×10^{3}
	Mean	8.27×10^{2}	1.08×10^{3}	8.16×10^{2}	7.15×10^{2}	7.56×10^{2}
	Std	1.92×10^{2}	1.75×10^{3}	2.34×10^{2}	7.99×10^{1}	2.41×10^{2}
F17	Best	2.15×10^{4}	1.86×10^{4}	8.33×10^{3}	1.44×10^{4}	1.97×10^{4}
	Worst	1.15×10^{5}	1.43×10^{5}	6.53×10^{4}	7.35×10^{4}	7.97×10^{4}
	Mean	4.90×10^{4}	4.61×10^{4}	3.89×10^{4}	4.23×10^{4}	4.00×10^{4}
	Std	1.97×10^{4}	2.59×10^{4}	1.12×10^{4}	1.49×10^{4}	1.37×10^{4}
F24	Best	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}
	Worst	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}
	Mean	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}	2.82×10^{3}
	Std	7.86×10^{-2}	7.48×10^{-2}	6.23×10^{-2}	7.93×10^{-2}	9.26×10^{-2}
F29	Best	6.95×10^{4}	5.62×10^{4}	1.88×10^{5}	1.08×10^{5}	1.67×10^{5}
	Worst	1.47×10^{6}	1.91×10^{6}	1.28×10^{6}	1.11×10^{6}	1.08×10^{6}
	Mean	5.95×10^{5}	5.71×10^{5}	5.91×10^{5}	4.95×10^{5}	6.05×10^{5}
	Std	3.44×10^{5}	4.45×10^{5}	3.46×10^{5}	2.60×10^{5}	2.66×10^{5}

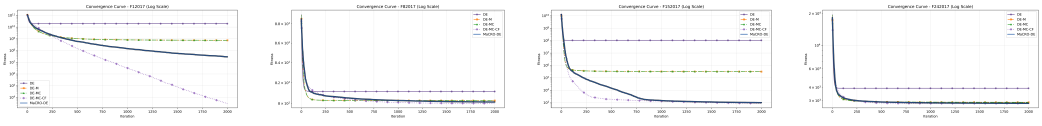
Supplementary Note S7: Ablation study of MaCRO-DE algorithm

To clarify the gradual incorporation of the mechanisms evaluated in the ablation study, the following variants are defined:

- DE: Standard differential evolution, which is used as the baseline configuration and does not include Mahalanobis distance-based neighborhood selection, Cholesky decomposition, or adaptive control mechanisms.
- DE-M: Differential evolution incorporating neighborhood selection based on Mahalanobis distance, which allows the search to consider the population's covariance structure.
- DE-MC: Differential evolution with Mahalanobis distance implemented via Cholesky decomposition. This provides a numerically efficient mechanism for identifying neighboring candidate solutions according to the population's geometry.
- DE-MC-CF: Extended DE-MC with adaptive control of differential evolution parameters, where the scale factor and crossover behavior adjust based on the population's state.
- **MaCRO-DE**: The proposed comprehensive method combines Mahalanobis-Cholesky neighborhood selection, adaptive parameter control, and the diversity-sensitive search mechanism introduced in this work.

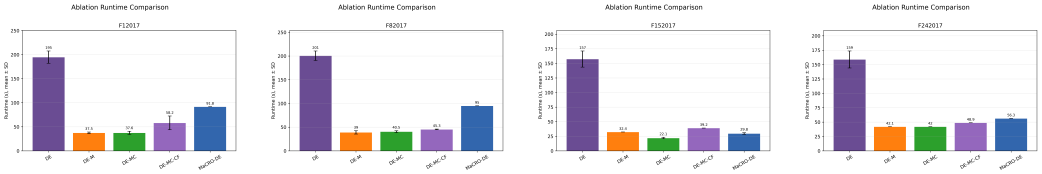
Regarding the IMPDE formulation developed earlier, the closest ablation variant is DE-MC, since it preserves the Mahalanobis distance mechanism with Cholesky decomposition while excluding the adaptive control and diversity-aware mechanisms introduced in MaCRO-DE. This ablation study was conducted using representative features from the CEC-2017: F1, F8, F15, and F24. The compared variants were DE, DE-M, DE-MC, DE-MC-CF, and full MaCRO-DE. The results demonstrate that progressively incorporating the proposed components substantially improves optimization performance. MaCRO-DE achieved better global fitness values for most of the selected functions. DE-MC-CF also showed clear improvement over the simpler variants.

The convergence curves further illustrate (in Fig. S3) the contribution of the incorporated mechanisms: MaCRO-DE and DE-MC-CF consistently reach lower fitness regions than DE, DE-M, and DE-MC. Regarding computational cost, the runtime analysis (see Fig. S4) shows that the additional mechanisms increase runtime compared to the simpler variants, DE-M and DE-MC. However, this increase is accompanied by a considerable improvement in solution quality. This indicates that the additional computational effort involved in the full formulation of MaCRO-DE is justified by its superior optimization performance.



(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

Figure S3. Convergence behavior of the MaCRO-DE ablation variants on representative CEC-2017 benchmark functions.



(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

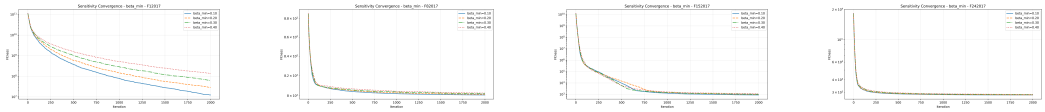
Figure S4. Runtime comparison of the MaCRO-DE ablation variants on representative CEC-2017 benchmark functions.

Table S5. Ablation Study of MaCRO-DE on Representative CEC-2017 Functions

Function	Statistic	DE	DE-M	DE-MC	DE-MC-CF	MaCRO-DE
F1	Best	7.35×10^{9}	1.34×10^{9}	1.34×10^{9}	1.06×10^{4}	1.00×10^{2}
	Worst	3.73×10^{10}	1.35×10^{10}	1.35×10^{10}	4.79×10^{4}	3.70×10^{2}
	Mean	1.97×10^{10}	4.97×10^{9}	4.97×10^{9}	2.99×10^{4}	1.14×10^{2}
	Std	7.71×10^{9}	3.15×10^{9}	3.15×10^{9}	9.07×10^{3}	5.05×10^{1}
F3	Best	7.48×10^{2}	4.96×10^{2}	4.96×10^{2}	3.32×10^{2}	3.32×10^{2}
	Worst	9.14×10^{3}	2.21×10^{3}	2.21×10^{3}	3.42×10^{2}	3.32×10^{2}
	Mean	3.39×10^{3}	8.51×10^{2}	8.51×10^{2}	3.33×10^{2}	3.32×10^{2}
	Std	1.82×10^{3}	4.06×10^{2}	4.06×10^{2}	2.43×10^{0}	1.47×10^{-7}
F8	Best	8.04×10^{2}	8.03×10^{2}	8.03×10^{2}	8.00×10^{2}	8.00×10^{2}
	Worst	8.23×10^{2}	8.09×10^{2}	8.09×10^{2}	8.00×10^{2}	8.00×10^{2}
	Mean	8.12×10^{2}	8.06×10^{2}	8.06×10^{2}	8.00×10^{2}	8.00×10^{2}
	Std	4.60×10^{0}	2.01×10^{0}	2.01×10^{0}	2.20×10^{-7}	1.34×10^{-8}
F10	Best	9.95×10^{4}	2.17×10^{4}	2.17×10^{4}	9.74×10^{3}	1.15×10^{3}
	Worst	3.40×10^{7}	1.48×10^{5}	1.48×10^{5}	2.92×10^{4}	1.95×10^{3}
	Mean	2.58×10^{6}	7.17×10^{4}	7.17×10^{4}	1.89×10^{4}	1.22×10^{3}
	Std	6.68×10^{6}	3.35×10^{4}	3.35×10^{4}	5.35×10^{3}	1.53×10^{2}
F15	Best	9.50×10^{4}	7.25×10^{3}	7.25×10^{3}	6.31×10^{2}	6.23×10^{2}
	Worst	2.05×10^{9}	7.36×10^{6}	7.36×10^{6}	1.68×10^{3}	1.13×10^{3}
	Mean	1.05×10^{8}	1.10×10^{6}	1.10×10^{6}	8.16×10^{2}	7.32×10^{2}
	Std	3.73×10^{8}	2.02×10^{6}	2.02×10^{6}	2.34×10^{2}	9.34×10^{1}
F17	Best	3.79×10^{4}	7.33×10^{3}	7.33×10^{3}	2.15×10^{4}	1.80×10^{3}
	Worst	1.36×10^{7}	8.26×10^{5}	8.26×10^{5}	1.15×10^{5}	6.19×10^{3}
	Mean	2.01×10^{6}	7.44×10^{4}	7.44×10^{4}	4.90×10^{4}	2.53×10^{3}
	Std	2.86×10^{6}	1.47×10^{5}	1.47×10^{5}	1.97×10^{4}	1.11×10^{3}
F24	Best	3.17×10^{3}	2.89×10^{3}	2.89×10^{3}	2.82×10^{3}	2.82×10^{3}
	Worst	5.43×10^{3}	3.20×10^{3}	3.20×10^{3}	2.82×10^{3}	2.82×10^{3}
	Mean	3.93×10^{3}	3.06×10^{3}	3.06×10^{3}	2.82×10^{3}	2.82×10^{3}
	Std	5.68×10^{2}	8.93×10^{1}	8.93×10^{1}	7.86×10^{-2}	8.81×10^{-2}

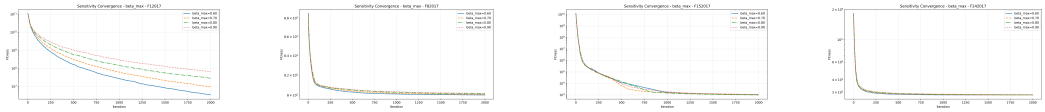
Supplementary Note S8: Parameter sensitivity analysis

A sensitivity analysis of the parameters was performed using representative CEC-2017 functions: F1, F8, F15, and F24. The following four MaCRO-DE parameters were evaluated independently: the minimum scaling parameter ($beta_{\min}$); the maximum scaling parameter ($beta_{\max}$); the crossing probability (p_{cr}); and the Mahalanobis quantile parameter (q). During each experiment, the remaining parameters were held constant to isolate the effect of the analyzed parameter. The numerical results are summarized in Table S6. Figs. S5 and S6 show the influence of $beta_{\min}$ and $beta_{\max}$ on the convergence behavior, respectively. Lower values generally result in faster convergence, especially for F1 and F15. In contrast, the differences between the tested configurations are considerably smaller for F8 and F15, where most parameter values converge toward similar final fitness regions. The effect of the crossover probability, p_{cr}, is shown in Fig. S7. Lower values of p_{cr} generally result in more favorable convergence behavior for F1 and F8, though the differences are less pronounced for F15 and F24. These results suggest that the impact of p_{cr} depends on the characteristics of the optimization landscape, and excessive crossover may reduce convergence efficiency for certain functions. Finally, Fig. S8 illustrates MaCRO-DE's sensitivity to the Mahalanobis quantile parameter q. The effect of q is particularly evident for F1, where $q = 0.9$ yields a substantially lower final fitness value. For the remaining functions, the differences among the tested values are smaller. Overall, the results in Table S6 and Figs. S5 to S8 indicate that MaCRO-DE exhibits relatively stable behavior across various parameter configurations. However, its convergence rate and the quality of the final solution may vary depending on the selected parameter values.



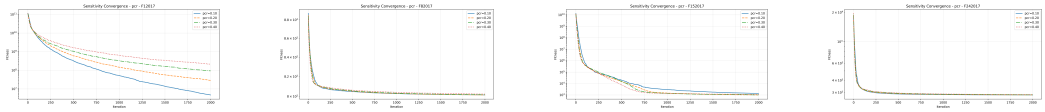
(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

Figure S5. Sensitivity analysis of $beta_{\min}$ on representative CEC-2017 benchmark functions.



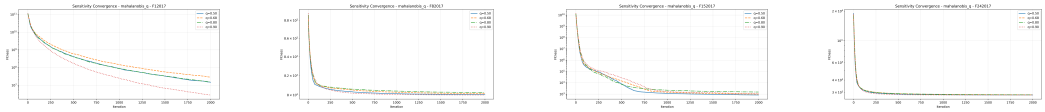
(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

Figure S6. Sensitivity analysis of $beta_{\max}$ on representative CEC-2017 benchmark functions.



(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

Figure S7. Sensitivity analysis of p_{cr} on representative CEC-2017 benchmark functions.



(**a**) F1 (**b**) F8 (**c**) F15 (**d**) F24

Figure S8. Sensitivity analysis of the Mahalanobis quantile parameter q on representative CEC-2017 benchmark functions.

Table S6. Parameter Sensitivity Analysis of MaCRO-DE on Representative CEC-2017 Functions

Function	Statistic	Mahalanobis q				$\beta_{\min}$				$\beta_{\max}$				p_{cr}			
		0.5	0.68	0.8	0.9	0.1	0.2	0.3	0.4	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4
F1	Best	6.06×10^2	4.88×10^4	8.50×10^3	1.06×10^4	1.06×10^4	7.65×10^4	2.49×10^5	1.31×10^6	1.06×10^4	1.18×10^5	3.81×10^5	8.04×10^5	1.06×10^4	4.90×10^4	6.33×10^4	1.76×10^5
	Worst	9.18×10^3	1.48×10^5	8.67×10^4	4.79×10^4	4.79×10^4	2.12×10^5	8.02×10^5	2.86×10^6	4.79×10^4	2.66×10^5	9.16×10^5	3.51×10^6	4.79×10^4	6.46×10^5	3.16×10^6	5.49×10^6
	Mean	2.94×10^3	1.02×10^5	3.66×10^4	2.99×10^4	2.99×10^4	1.38×10^5	5.05×10^5	1.87×10^6	2.99×10^4	1.78×10^5	6.65×10^5	2.27×10^6	2.99×10^4	3.30×10^5	1.44×10^6	2.67×10^6
	Std	2.00×10^3	2.01×10^4	1.40×10^4	9.07×10^3	9.07×10^3	3.39×10^4	1.38×10^5	3.78×10^5	9.07×10^3	3.94×10^4	1.51×10^5	6.14×10^5	9.07×10^3	1.59×10^5	8.51×10^5	1.50×10^6
F8	Best	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2
	Worst	8.04×10^2	8.01×10^2	8.03×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.01×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.01×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2
	Mean	8.00×10^2	8.00×10^2	8.01×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2	8.00×10^2
	Std	9.78×10^{-1}	1.85×10^{-1}	6.59×10^{-1}	2.20×10^{-7}	2.20×10^{-7}	5.71×10^{-4}	4.76×10^{-2}	1.42×10^{-1}	2.20×10^{-7}	3.62×10^{-4}	9.24×10^{-2}	1.38×10^{-1}	2.20×10^{-7}	2.93×10^{-8}	2.29×10^{-7}	3.99×10^{-6}
F15	Best	6.30×10^2	6.60×10^2	7.79×10^2	6.31×10^2	6.31×10^2	6.37×10^2	6.48×10^2	6.43×10^2	6.31×10^2	6.37×10^2	6.48×10^2	6.38×10^2	6.31×10^2	6.27×10^2	6.27×10^2	6.27×10^2
	Worst	1.69×10^3	2.01×10^3	3.29×10^3	1.68×10^3	1.68×10^3	1.76×10^3	1.69×10^3	1.60×10^3	1.68×10^3	1.42×10^3	1.50×10^3	2.11×10^3	1.68×10^3	2.16×10^3	1.44×10^3	1.29×10^6
	Mean	1.07×10^3	1.13×10^3	1.56×10^3	8.16×10^2	8.16×10^2	8.61×10^2	8.49×10^2	8.98×10^2	8.16×10^2	8.28×10^2	9.02×10^2	9.76×10^2	8.16×10^2	8.18×10^2	7.80×10^2	4.38×10^4
	Std	2.51×10^2	3.54×10^2	5.48×10^2	2.34×10^2	2.34×10^2	2.41×10^2	2.03×10^2	2.40×10^2	2.34×10^2	1.95×10^2	2.38×10^2	3.36×10^2	2.34×10^2	3.43×10^2	1.77×10^2	2.36×10^5
F24	Best	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3
	Worst	2.82×10^3	2.82×10^3	2.83×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3
	Mean	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.82×10^3
	Std	6.11×10^{-2}	8.87×10^{-2}	1.43×10^0	7.86×10^{-2}	7.86×10^{-2}	7.68×10^{-2}	1.08×10^{-1}	1.91×10^{-1}	7.86×10^{-2}	7.42×10^{-2}	1.02×10^{-1}	2.38×10^{-1}	7.86×10^{-2}	7.29×10^{-2}	8.03×10^{-2}	9.36×10^{-2}

Supplementary Note S9: Extended comparative results of the full CEC-2017 benchmark

To supplement the average fitness results presented in the main comparative analysis, this section includes additional descriptive statistics for the CEC-2017 baseline test. Tables S7 and S8 summarize the best, worst, and standard deviation values obtained by MaCRO-DE and the competing algorithms for functions F1–F29. These supplementary results provide a broader view of the algorithms' behavior across independent runs. This supports comparing average performance with information about extreme results and variability in results.

Table S7. Best, Worst, and Standard Deviation Fitness Values on CEC-2017 Functions F1–F15

Function	Statistic	MaCRO-DE	BRO	DBO	DE	DMOA	GWO	HHO	MFO	MGO	PSO	SHADE	WOA	JADE	FLA
F1	Best	1.00×10^{2}	1.87×10^{10}	9.14×10^{2}	7.35×10^{9}	1.36×10^{6}	7.82×10^{6}	9.74×10^{5}	9.22×10^{6}	1.18×10^{2}	1.39×10^{3}	1.00×10^{2}	5.84×10^{3}	1.00×10^{2}	8.20×10^{6}
	Worst	3.70×10^{2}	4.40×10^{10}	1.34×10^{7}	3.73×10^{10}	5.12×10^{6}	8.53×10^{8}	3.85×10^{6}	2.24×10^{9}	7.08×10^{3}	7.46×10^{7}	1.00×10^{2}	5.65×10^{9}	1.00×10^{2}	3.92×10^{7}
	Std	5.05×10^{1}	5.69×10^{9}	3.26×10^{6}	7.71×10^{9}	8.13×10^{5}	2.30×10^{8}	6.83×10^{5}	5.85×10^{8}	2.01×10^{3}	1.37×10^{7}	3.12×10^{-14}	1.21×10^{9}	5.90×10^{-15}	6.49×10^{6}
F2	Best	2.00×10^{2}	3.49×10^{4}	2.06×10^{2}	3.83×10^{4}	3.16×10^{3}	2.35×10^{2}	2.07×10^{2}	3.57×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	1.52×10^{3}	2.00×10^{2}	2.63×10^{2}
	Worst	2.00×10^{2}	7.32×10^{4}	4.93×10^{2}	1.02×10^{5}	1.02×10^{4}	8.40×10^{3}	2.50×10^{2}	1.60×10^{4}	2.00×10^{2}	2.01×10^{2}	2.00×10^{2}	3.23×10^{4}	2.00×10^{2}	3.49×10^{2}
	Std	1.21×10^{-5}	1.11×10^{4}	6.73×10^{1}	1.60×10^{4}	1.74×10^{3}	2.03×10^{3}	1.06×10^{1}	3.10×10^{3}	1.98×10^{-5}	2.67×10^{-1}	2.79×10^{-14}	6.85×10^{3}	2.36×10^{-14}	2.08×10^{1}
F3	Best	3.32×10^{2}	2.85×10^{3}	3.33×10^{2}	7.48×10^{2}	3.32×10^{2}	3.36×10^{2}	3.31×10^{2}	3.44×10^{2}	3.16×10^{2}	3.29×10^{2}	3.00×10^{2}	3.74×10^{2}	3.00×10^{2}	3.33×10^{2}
	Worst	3.32×10^{2}	9.62×10^{3}	8.35×10^{2}	9.14×10^{3}	3.33×10^{2}	4.74×10^{2}	5.61×10^{2}	5.62×10^{2}	4.16×10^{2}	4.64×10^{2}	3.32×10^{2}	8.91×10^{2}	3.32×10^{2}	4.94×10^{2}
	Std	1.47×10^{-7}	1.85×10^{3}	1.21×10^{2}	1.82×10^{3}	2.12×10^{-1}	4.08×10^{1}	5.79×10^{1}	4.78×10^{1}	2.35×10^{1}	4.40×10^{1}	1.31×10^{1}	1.21×10^{2}	9.67×10^{0}	4.85×10^{1}
F4	Best	5.96×10^{2}	2.36×10^{4}	5.90×10^{2}	8.91×10^{3}	6.04×10^{2}	5.51×10^{2}	8.87×10^{2}	1.79×10^{3}	4.68×10^{2}	6.71×10^{2}	4.20×10^{2}	3.64×10^{3}	4.59×10^{2}	5.99×10^{2}
	Worst	6.41×10^{2}	4.98×10^{4}	1.25×10^{3}	5.47×10^{4}	6.59×10^{2}	2.29×10^{3}	1.60×10^{3}	7.69×10^{3}	6.16×10^{2}	1.93×10^{3}	4.59×10^{2}	1.72×10^{4}	5.33×10^{2}	7.61×10^{2}
	Std	1.17×10^{1}	6.04×10^{3}	1.71×10^{2}	9.65×10^{3}	1.37×10^{1}	4.87×10^{2}	1.73×10^{2}	1.60×10^{3}	3.79×10^{1}	2.81×10^{2}	7.76×10^{0}	3.97×10^{3}	1.91×10^{1}	3.36×10^{1}
F5	Best	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}
	Worst	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}	5.00×10^{2}
	Std	1.66×10^{-3}	9.65×10^{-3}	3.17×10^{-3}	9.28×10^{-4}	2.64×10^{-4}	3.71×10^{-4}	3.55×10^{-3}	4.40×10^{-4}	8.04×10^{-5}	9.22×10^{-4}	6.19×10^{-6}	2.74×10^{-3}	5.68×10^{-4}	8.87×10^{-4}
F6	Best	8.30×10^{2}	7.15×10^{5}	8.54×10^{2}	3.05×10^{5}	9.36×10^{2}	3.05×10^{3}	1.19×10^{3}	3.54×10^{3}	7.64×10^{2}	1.08×10^{3}	6.30×10^{2}	9.42×10^{3}	6.91×10^{2}	1.25×10^{3}
	Worst	8.85×10^{2}	1.90×10^{6}	2.83×10^{3}	1.35×10^{6}	1.12×10^{3}	6.53×10^{4}	1.80×10^{3}	1.57×10^{5}	1.07×10^{3}	4.38×10^{3}	6.74×10^{2}	6.19×10^{5}	7.64×10^{2}	2.55×10^{3}
	Std	1.38×10^{1}	3.13×10^{5}	4.79×10^{2}	2.58×10^{5}	4.15×10^{1}	1.48×10^{4}	1.61×10^{2}	3.43×10^{4}	7.03×10^{1}	9.58×10^{2}	8.08×10^{0}	1.37×10^{5}	2.00×10^{1}	3.00×10^{2}
F7	Best	9.83×10^{2}	1.16×10^{3}	8.62×10^{2}	8.15×10^{2}	9.00×10^{2}	7.88×10^{2}	9.91×10^{2}	8.58×10^{2}	7.50×10^{2}	8.72×10^{2}	7.47×10^{2}	1.25×10^{3}	8.56×10^{2}	8.44×10^{2}
	Worst	1.16×10^{3}	1.42×10^{3}	1.28×10^{3}	1.08×10^{3}	1.09×10^{3}	1.12×10^{3}	1.32×10^{3}	1.16×10^{3}	9.58×10^{2}	1.16×10^{3}	7.90×10^{2}	1.73×10^{3}	9.56×10^{2}	1.08×10^{3}
	Std	3.98×10^{1}	6.61×10^{1}	9.86×10^{1}	6.73×10^{1}	4.33×10^{1}	9.85×10^{1}	8.17×10^{1}	6.89×10^{1}	4.64×10^{1}	7.42×10^{1}	1.10×10^{1}	1.11×10^{2}	2.76×10^{1}	5.61×10^{1}
F8	Best	8.00×10^{2}	8.12×10^{2}	8.01×10^{2}	8.04×10^{2}	8.00×10^{2}	8.00×10^{2}	8.06×10^{2}	8.02×10^{2}	8.01×10^{2}	8.02×10^{2}	8.00×10^{2}	8.03×10^{2}	8.00×10^{2}	8.00×10^{2}
	Worst	8.00×10^{2}	8.50×10^{2}	8.22×10^{2}	8.23×10^{2}	8.01×10^{2}	8.04×10^{2}	8.21×10^{2}	8.13×10^{2}	8.13×10^{2}	8.15×10^{2}	8.01×10^{2}	8.41×10^{2}	8.01×10^{2}	8.21×10^{2}
	Std	1.34×10^{-8}	8.36×10^{0}	4.01×10^{0}	4.60×10^{0}	8.81×10^{-2}	1.10×10^{0}	3.27×10^{0}	2.54×10^{0}	2.10×10^{0}	3.69×10^{0}	1.77×10^{-1}	6.29×10^{0}	3.01×10^{-1}	5.39×10^{0}
F9	Best	6.93×10^{3}	4.88×10^{3}	2.23×10^{3}	2.49×10^{3}	6.38×10^{3}	2.10×10^{3}	1.94×10^{3}	2.28×10^{3}	5.68×10^{2}	0.00×10^{0}	2.41×10^{2}	2.85×10^{3}	5.66×10^{3}	1.02×10^{1}
	Worst	1.03×10^{4}	9.84×10^{3}	7.89×10^{3}	6.45×10^{3}	9.98×10^{3}	1.15×10^{4}	1.06×10^{4}	7.27×10^{3}	4.15×10^{3}	7.87×10^{3}	2.49×10^{3}	8.41×10^{3}	9.28×10^{3}	4.98×10^{3}
	Std	9.27×10^{2}	1.22×10^{3}	1.26×10^{3}	1.02×10^{3}	7.20×10^{2}	2.62×10^{3}	2.02×10^{3}	1.17×10^{3}	9.18×10^{2}	1.51×10^{3}	4.89×10^{2}	1.58×10^{3}	8.21×10^{2}	1.32×10^{3}
F10	Best	1.15×10^{3}	1.20×10^{5}	3.99×10^{4}	9.95×10^{4}	3.02×10^{4}	4.49×10^{4}	3.92×10^{4}	7.68×10^{3}	2.88×10^{3}	1.91×10^{3}	1.04×10^{3}	1.95×10^{4}	1.05×10^{3}	3.03×10^{3}
	Worst	1.95×10^{3}	3.48×10^{6}	2.52×10^{5}	3.40×10^{7}	1.80×10^{5}	1.34×10^{5}	1.70×10^{5}	5.01×10^{4}	6.95×10^{4}	6.27×10^{4}	1.69×10^{3}	3.79×10^{6}	5.61×10^{3}	8.20×10^{4}
	Std	1.53×10^{2}	8.51×10^{5}	6.26×10^{4}	6.68×10^{6}	3.60×10^{4}	2.64×10^{4}	3.63×10^{4}	1.09×10^{4}	1.58×10^{4}	1.89×10^{4}	2.07×10^{2}	6.82×10^{5}	1.07×10^{3}	2.03×10^{4}
F11	Best	4.14×10^{4}	1.16×10^{9}	1.17×10^{5}	1.91×10^{8}	2.58×10^{6}	3.01×10^{5}	2.58×10^{6}	3.02×10^{4}	3.52×10^{3}	3.65×10^{3}	1.80×10^{3}	2.23×10^{4}	2.23×10^{3}	3.84×10^{5}
	Worst	3.70×10^{5}	3.26×10^{9}	1.03×10^{8}	6.37×10^{9}	1.02×10^{7}	5.64×10^{7}	5.66×10^{7}	1.70×10^{6}	1.05×10^{5}	1.33×10^{5}	9.40×10^{3}	5.45×10^{7}	1.11×10^{4}	2.36×10^{6}
	Std	6.88×10^{4}	6.13×10^{8}	1.98×10^{7}	1.40×10^{9}	2.02×10^{6}	1.22×10^{7}	1.64×10^{7}	3.99×10^{5}	2.07×10^{4}	2.29×10^{4}	1.69×10^{3}	9.85×10^{6}	2.15×10^{3}	5.00×10^{5}
F12	Best	1.44×10^{3}	6.09×10^{8}	9.19×10^{4}	7.16×10^{7}	1.33×10^{4}	1.49×10^{5}	2.43×10^{5}	5.75×10^{3}	1.67×10^{3}	2.03×10^{3}	1.26×10^{3}	2.36×10^{4}	1.32×10^{3}	2.41×10^{4}
	Worst	7.50×10^{3}	3.53×10^{9}	1.52×10^{7}	8.90×10^{9}	1.86×10^{5}	2.10×10^{7}	2.71×10^{6}	3.60×10^{6}	1.38×10^{4}	9.95×10^{4}	5.42×10^{3}	1.36×10^{7}	1.18×10^{4}	7.09×10^{5}
	Std	1.43×10^{3}	8.48×10^{8}	3.99×10^{6}	1.93×10^{9}	4.52×10^{4}	3.76×10^{6}	6.60×10^{5}	6.49×10^{5}	3.14×10^{3}	1.75×10^{4}	9.67×10^{2}	2.54×10^{6}	3.38×10^{3}	1.54×10^{5}
F13	Best	1.61×10^{3}	2.62×10^{5}	5.55×10^{4}	1.84×10^{5}	2.56×10^{5}	7.35×10^{4}	1.78×10^{5}	2.72×10^{4}	5.06×10^{3}	2.43×10^{3}	1.35×10^{3}	2.35×10^{4}	1.35×10^{3}	7.30×10^{3}
	Worst	6.49×10^{3}	3.26×10^{6}	9.62×10^{5}	2.02×10^{7}	5.83×10^{5}	4.74×10^{5}	4.62×10^{6}	8.90×10^{5}	3.02×10^{5}	4.37×10^{5}	4.35×10^{3}	4.20×10^{6}	2.74×10^{4}	2.71×10^{5}
	Std	8.94×10^{2}	7.84×10^{5}	2.37×10^{5}	4.43×10^{6}	7.40×10^{4}	8.88×10^{4}	9.15×10^{5}	2.19×10^{5}	7.68×10^{4}	9.96×10^{4}	5.67×10^{2}	7.56×10^{5}	7.06×10^{3}	6.28×10^{4}
F14	Best	1.97×10^{3}	1.48×10^{8}	8.88×10^{4}	7.26×10^{6}	3.80×10^{4}	6.98×10^{4}	8.05×10^{4}	1.53×10^{4}	4.09×10^{3}	3.81×10^{3}	1.49×10^{3}	2.11×10^{4}	1.50×10^{3}	2.24×10^{4}
	Worst	1.53×10^{4}	1.40×10^{9}	2.93×10^{6}	1.45×10^{9}	1.34×10^{5}	9.83×10^{5}	1.55×10^{6}	6.84×10^{4}	4.13×10^{4}	3.49×10^{4}	4.26×10^{3}	2.32×10^{8}	8.59×10^{3}	4.57×10^{5}
	Std	3.85×10^{3}	2.61×10^{8}	7.25×10^{5}	3.95×10^{8}	2.49×10^{4}	1.63×10^{5}	3.48×10^{5}	1.37×10^{4}	8.33×10^{3}	8.61×10^{3}	6.71×10^{2}	4.23×10^{7}	1.32×10^{3}	8.47×10^{4}
F15	Best	6.23×10^{2}	7.98×10^{5}	7.24×10^{3}	9.50×10^{4}	1.74×10^{3}	1.27×10^{4}	4.80×10^{3}	1.28×10^{3}	6.91×10^{2}	6.88×10^{2}	8.10×10^{2}	1.39×10^{3}	7.05×10^{2}	9.18×10^{2}
	Worst	1.13×10^{3}	3.04×10^{8}	1.84×10^{6}	2.05×10^{9}	1.93×10^{5}	4.87×10^{6}	1.12×10^{6}	1.67×10^{5}	8.79×10^{3}	9.68×10^{3}	1.66×10^{3}	1.98×10^{6}	2.12×10^{3}	1.12×10^{4}
	Std	9.34×10^{1}	6.30×10^{7}	5.34×10^{5}	3.73×10^{8}	4.00×10^{4}	1.42×10^{6}	2.61×10^{5}	3.37×10^{4}	1.46×10^{3}	2.05×10^{3}	2.45×10^{2}	6.33×10^{5}	2.84×10^{2}	3.11×10^{3}

Supplementary Note S10: Evaluation of Transfer Functions for Binarization

A sensitivity analysis was conducted to evaluate how different transfer functions influence the binary feature selection stage of MaCRO-DE. The analysis considered eight transfer functions, including four S-shaped variants ($S1$–$S4$) and four V-shaped variants ($V1$–$V4$). All configurations were evaluated under the same experimental conditions to ensure a fair comparison. The classification results are summarized in Table S9. Figs. S9 and S10 provide graphical comparisons of predictive performance, selected features, and computational runtime. The results in Table S9 demonstrate that, particularly, the $V4$ transfer function consistently achieves the highest average accuracy across all datasets. The V-shaped variants performed particularly well on BreastCancer, Tic-tac-toe, and Wine datasets, while the S-shaped variants remained highly competitive on the Ionosphere and Zoo datasets. These results confirm that the binarization mechanism directly influences the search process and that its effectiveness depends on the characteristics of the dataset.

The classification metrics shown in Fig. S9 provide a closer look at the differences between the transfer functions. V-shaped variants generally achieve higher Accuracy, F1-Score, Precision, and Recall. $V1$ and $V2$ demonstrate particularly competitive performance. However, the differences between the various configurations are relatively small, indicating that multiple transfer functions can produce adequate binary mappings. Finally, Fig. S10

Table S8. Best, Worst, and Standard Deviation Fitness Values on CEC-2017 Functions F16–F29

Function	Statistic	MaCRO-DE	BRO	DBO	DE	DMOA	GWO	HHO	MFO	MGO	PSO	SHADE	WOA	JADE	FLA
F16	Best	1.48×10^3	7.93×10^4	5.73×10^3	2.86×10^4	2.65×10^4	8.48×10^3	3.32×10^3	3.88×10^3	1.63×10^3	1.71×10^3	1.41×10^3	8.03×10^3	1.51×10^3	2.63×10^3
	Worst	1.77×10^3	6.81×10^9	1.12×10^5	2.15×10^{10}	2.94×10^5	6.66×10^4	4.94×10^4	3.34×10^4	3.25×10^3	5.79×10^3	3.34×10^3	1.67×10^5	3.45×10^3	4.64×10^3
	Std	9.17×10^1	1.46×10^9	2.96×10^4	4.48×10^9	5.82×10^4	2.00×10^4	1.06×10^4	5.95×10^3	4.13×10^2	1.04×10^3	3.59×10^2	4.29×10^4	4.74×10^2	5.05×10^2
F17	Best	1.80×10^3	1.26×10^5	2.95×10^4	3.79×10^4	1.01×10^5	5.08×10^4	5.83×10^4	2.58×10^4	2.06×10^4	1.30×10^4	1.74×10^3	2.29×10^4	1.78×10^3	2.35×10^4
	Worst	6.19×10^3	6.13×10^6	7.29×10^5	1.36×10^7	6.93×10^5	1.32×10^5	1.92×10^5	9.70×10^4	8.90×10^4	1.03×10^5	2.64×10^3	4.16×10^5	2.01×10^4	9.20×10^4
	Std	1.11×10^3	1.53×10^6	1.33×10^5	2.86×10^6	1.12×10^5	2.09×10^4	3.62×10^4	2.03×10^4	1.77×10^4	2.26×10^4	2.06×10^2	9.79×10^4	3.48×10^3	1.77×10^4
F18	Best	1.83×10^3	5.46×10^9	2.00×10^4	5.17×10^7	3.03×10^3	2.37×10^4	2.89×10^4	3.73×10^3	1.94×10^3	1.89×10^3	1.81×10^3	1.26×10^4	1.82×10^3	4.60×10^4
	Worst	1.86×10^3	1.56×10^{11}	2.97×10^5	1.68×10^{13}	5.76×10^4	4.75×10^8	2.70×10^5	1.15×10^5	3.59×10^4	6.29×10^4	1.83×10^3	1.23×10^8	1.86×10^3	6.07×10^5
	Std	6.38×10^0	3.15×10^{10}	7.42×10^4	4.21×10^{12}	1.69×10^4	8.67×10^7	5.77×10^4	2.89×10^4	6.88×10^3	1.84×10^4	5.57×10^0	2.24×10^7	8.66×10^0	1.46×10^5
F19	Best	1.39×10^3	4.02×10^3	1.62×10^3	2.18×10^3	1.34×10^3	1.50×10^3	3.34×10^3	2.00×10^3	1.47×10^3	1.58×10^3	1.45×10^3	4.77×10^3	1.48×10^3	1.42×10^3
	Worst	1.91×10^3	8.84×10^3	3.96×10^3	7.11×10^3	1.82×10^3	2.74×10^3	8.54×10^3	3.90×10^3	2.47×10^3	3.40×10^3	2.06×10^3	1.88×10^4	1.89×10^3	2.79×10^3
	Std	1.23×10^2	1.43×10^3	5.89×10^2	1.08×10^3	1.25×10^2	2.78×10^2	1.43×10^3	5.19×10^2	2.07×10^2	3.85×10^2	1.77×10^2	3.64×10^3	1.24×10^2	3.15×10^2
F20	Best	2.10×10^3	7.85×10^3	2.10×10^3	4.85×10^3	2.13×10^3	2.45×10^3	2.12×10^3	2.74×10^3	2.10×10^3	2.10×10^3	2.10×10^3	2.17×10^3	2.10×10^3	2.18×10^3
	Worst	2.44×10^3	2.17×10^4	3.21×10^3	3.59×10^4	2.46×10^3	4.25×10^3	3.84×10^3	8.24×10^3	2.46×10^3	3.72×10^3	2.25×10^3	1.92×10^4	2.35×10^3	2.53×10^3
	Std	1.60×10^2	4.18×10^3	2.94×10^2	8.11×10^3	1.26×10^2	3.66×10^2	4.92×10^2	1.39×10^3	1.23×10^2	4.22×10^2	6.77×10^1	3.82×10^3	9.89×10^1	1.24×10^2
F21	Best	1.52×10^4	1.16×10^4	1.07×10^4	1.49×10^4	1.12×10^4	1.51×10^4	1.16×10^4	1.45×10^4	3.77×10^3	6.55×10^3	1.71×10^2	1.01×10^4	0.00×10^0	8.36×10^2
	Worst	1.52×10^4	1.67×10^4	1.54×10^4	1.54×10^4	1.52×10^4	1.53×10^4	1.57×10^4	1.52×10^4	1.08×10^4	1.51×10^4	6.34×10^3	1.54×10^4	4.78×10^3	3.15×10^3
	Std	9.48×10^0	9.70×10^2	1.02×10^3	1.06×10^2	7.14×10^2	3.89×10^1	9.64×10^2	1.79×10^2	1.92×10^3	1.75×10^3	1.45×10^3	1.42×10^3	1.00×10^3	5.65×10^2
F22	Best	2.40×10^3	1.75×10^4	2.40×10^3	1.79×10^4	2.49×10^3	2.56×10^3	2.51×10^3	2.57×10^3	2.40×10^3	2.40×10^3	2.40×10^3	2.48×10^3	2.40×10^3	2.71×10^3
	Worst	2.74×10^3	4.10×10^4	3.15×10^3	3.98×10^4	2.57×10^3	6.24×10^3	3.44×10^3	9.90×10^3	2.95×10^3	4.02×10^3	2.54×10^3	2.02×10^4	2.62×10^3	3.20×10^3
	Std	1.03×10^2	5.03×10^3	2.21×10^2	5.81×10^3	2.08×10^1	7.74×10^2	3.32×10^2	2.15×10^3	1.31×10^2	4.27×10^2	4.78×10^1	5.11×10^3	7.63×10^1	1.14×10^2
F23	Best	2.50×10^3	1.56×10^4	2.50×10^3	9.25×10^3	2.56×10^3	2.53×10^3	2.58×10^3	2.62×10^3	2.50×10^3	2.50×10^3	2.50×10^3	2.55×10^3	2.50×10^3	2.74×10^3
	Worst	2.83×10^3	2.78×10^4	2.92×10^3	2.47×10^4	2.61×10^3	5.72×10^3	3.60×10^3	1.10×10^4	2.71×10^3	3.75×10^3	2.50×10^3	1.96×10^4	2.50×10^3	2.94×10^3
	Std	1.10×10^2	3.04×10^3	9.04×10^1	4.24×10^3	1.23×10^1	7.87×10^2	2.99×10^2	1.54×10^3	3.80×10^1	2.93×10^2	8.44×10^{-14}	4.61×10^3	0.00×10^0	4.37×10^1
F24	Best	2.82×10^3	3.43×10^3	2.83×10^3	3.17×10^3	2.82×10^3	2.83×10^3	2.83×10^3	2.83×10^3	2.82×10^3	2.82×10^3	2.82×10^3	2.90×10^3	2.82×10^3	2.82×10^3
	Worst	2.82×10^3	5.96×10^3	2.94×10^3	5.43×10^3	2.82×10^3	2.99×10^3	3.08×10^3	3.21×10^3	2.84×10^3	2.95×10^3	2.82×10^3	3.45×10^3	2.82×10^3	2.87×10^3
	Std	8.81×10^{-2}	5.70×10^2	3.55×10^1	5.68×10^2	5.54×10^{-1}	3.87×10^1	5.50×10^1	8.60×10^1	5.13×10^0	2.40×10^1	1.14×10^{-1}	1.17×10^2	1.10×10^{-1}	1.21×10^1
F25	Best	3.32×10^3	4.18×10^3	3.39×10^3	3.63×10^3	3.33×10^3	3.33×10^3	3.41×10^3	3.33×10^3	3.33×10^3	3.33×10^3	3.32×10^3	3.40×10^3	3.33×10^3	3.33×10^3
	Worst	3.33×10^3	7.90×10^3	3.88×10^3	6.48×10^3	3.33×10^3	3.42×10^3	4.24×10^3	4.36×10^3	3.38×10^3	3.59×10^3	3.33×10^3	8.32×10^3	3.34×10^3	3.38×10^3
	Std	3.09×10^{-1}	6.71×10^2	1.29×10^2	5.78×10^2	5.67×10^{-1}	2.20×10^1	2.08×10^2	1.85×10^2	1.66×10^1	6.68×10^1	2.50×10^0	1.35×10^3	2.96×10^0	1.60×10^1
F26	Best	3.09×10^3	3.37×10^3	3.12×10^3	3.18×10^3	3.08×10^3	3.05×10^3	3.24×10^3	3.12×10^3	3.07×10^3	3.08×10^3	3.05×10^3	3.23×10^3	3.08×10^3	3.08×10^3
	Worst	3.12×10^3	3.84×10^3	3.32×10^3	3.45×10^3	3.12×10^3	3.14×10^3	4.18×10^3	3.39×10^3	3.22×10^3	3.31×10^3	3.10×10^3	4.16×10^3	3.13×10^3	3.16×10^3
	Std	8.01×10^0	1.12×10^2	4.65×10^1	7.41×10^1	8.26×10^0	1.81×10^1	2.12×10^2	8.15×10^1	2.80×10^1	6.23×10^1	1.00×10^1	2.58×10^2	1.35×10^1	1.67×10^1
F27	Best	2.70×10^3	3.54×10^3	2.81×10^3	3.33×10^3	2.72×10^3	2.87×10^3	2.79×10^3	2.98×10^3	2.70×10^3	2.74×10^3	2.70×10^3	3.12×10^3	2.70×10^3	2.75×10^3
	Worst	2.70×10^3	5.75×10^3	3.26×10^3	4.38×10^3	3.16×10^3	3.24×10^3	3.22×10^3	3.21×10^3	2.83×10^3	3.11×10^3	2.85×10^3	4.98×10^3	2.86×10^3	3.17×10^3
	Std	5.85×10^{-2}	5.44×10^2	1.02×10^2	3.00×10^2	1.28×10^2	9.36×10^1	1.35×10^2	6.10×10^1	2.19×10^1	8.84×10^1	3.91×10^1	4.49×10^2	3.94×10^1	1.82×10^2
F28	Best	4.43×10^3	5.12×10^8	2.75×10^4	6.71×10^7	1.42×10^5	3.83×10^4	2.93×10^4	9.96×10^3	5.08×10^3	6.04×10^3	4.39×10^3	2.67×10^4	4.06×10^3	7.74×10^3
	Worst	7.32×10^3	2.93×10^{10}	1.79×10^9	9.27×10^{10}	3.50×10^7	1.65×10^7	5.10×10^6	3.39×10^5	3.06×10^4	2.86×10^8	9.67×10^3	2.92×10^9	9.88×10^3	2.09×10^5
	Std	7.51×10^2	5.47×10^9	4.40×10^8	1.68×10^{10}	7.76×10^6	3.77×10^6	9.25×10^5	6.44×10^4	5.39×10^3	5.39×10^7	1.21×10^3	5.99×10^8	1.54×10^3	4.95×10^4
F29	Best	5.76×10^3	9.49×10^8	1.28×10^5	1.07×10^8	2.95×10^4	1.35×10^5	1.48×10^5	1.82×10^4	8.92×10^3	8.24×10^3	4.92×10^3	4.19×10^4	5.29×10^3	1.01×10^5
	Worst	2.57×10^4	4.07×10^{10}	1.15×10^8	3.47×10^{10}	2.33×10^5	2.55×10^8	6.95×10^7	8.74×10^5	4.00×10^4	2.02×10^7	1.14×10^4	3.10×10^8	5.11×10^4	1.06×10^7
	Std	4.84×10^3	1.20×10^{10}	2.15×10^7	7.68×10^9	5.32×10^4	4.86×10^7	1.57×10^7	1.71×10^5	8.87×10^3	4.30×10^6	1.67×10^3	5.65×10^7	8.29×10^3	2.59×10^6

shows the relationship between the average number of selected features and computational runtime. S-shaped variants tend to retain larger subsets of features, while V-shaped variants achieve greater dimensionality reduction. $V4$ selects the fewest features and requires the shortest average execution time. Overall, the results suggest that V-shaped transfer functions offer an optimal balance of predictive performance, dimensionality reduction, and computational cost, making them suitable for subsequent feature selection experiments.

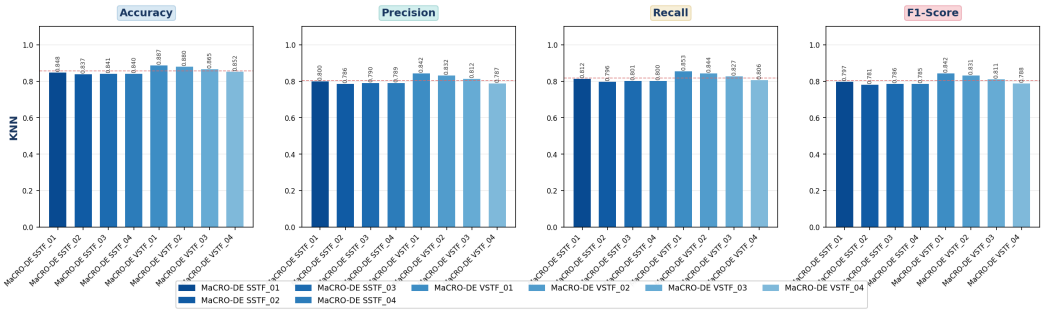


Figure S9. Average classification performance obtained by the evaluated transfer-function variants in terms of Accuracy, Precision, Recall, and F1-Score.

Supplementary Note S11: Parameter Sensitivity Analysis of α_1 and β_2

A sensitivity analysis was conducted to evaluate the influence of the fitness function weights α_1 and β_2 on the performance of MaCRO-DE's feature selection. Five complementary configurations that satisfied the condition $\alpha_1 + \beta_2 = 1$ were considered: $(0.70; 0.30)$, $(0.80; 0.20)$, $(0.90; 0.10)$, $(0.95; 0.05)$, and $(0.99; 0.01)$. The analysis was performed using KNN and SVM classifiers on the representative datasets BreastCancer, Ionosphere, Tic-tac-toe, and Wine. Figure S11 shows the resulting trade-off between the average F1 score and the average number of selected features. The effect of the weights varies depending on both the dataset and the classifier. In the case of BreastCancer and Ionosphere, several configurations achieve similar predictive performance, while more notable differences

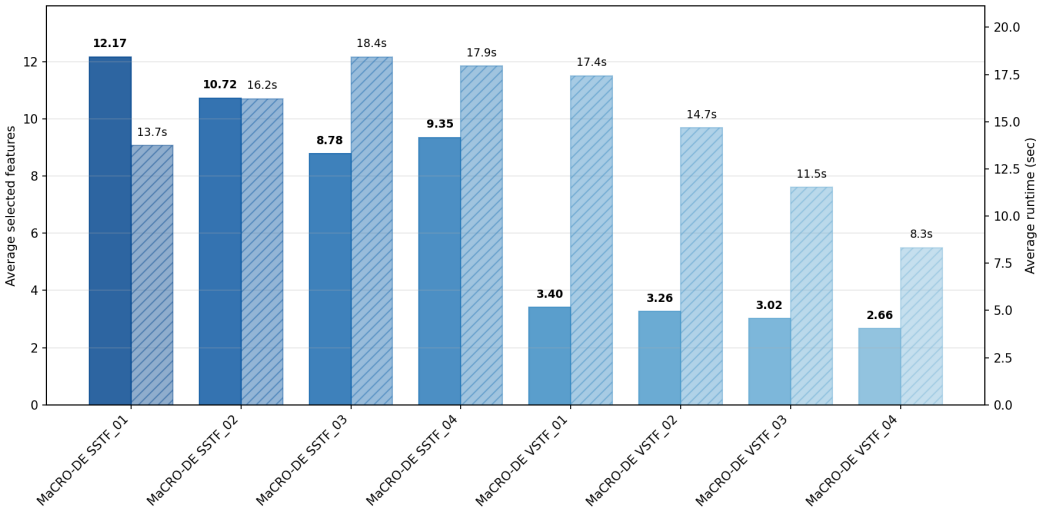


Figure S10. Trade-off between the average number of selected features and the average runtime obtained by the evaluated transfer-function variants.

Table S9. Classification Accuracy Obtained with Different Transfer Functions

Dataset	Statistic	V-shaped Transfer Functions				S-shaped Transfer Functions			
		V1	V2	V3	V4	S1	S2	S3	S4
BreastCancer	Best	96.4286	97.1429	96.4286	97.1429	97.1429	97.1429	**97.8571**	**97.8571**
	Worst	**65.7143**	**65.7143**	**65.7143**	**65.7143**	**65.7143**	**65.7143**	**65.7143**	**65.7143**
	Mean	90.5714	**91.7619**	90.7857	91.1190	72.8095	77.9286	80.8810	75.9286
	Std	9.2075	7.9909	8.8030	7.3748	13.0876	15.2245	**15.4429**	14.7046
Ionosphere	Best	**98.5915**	95.7746	94.3662	97.1831	**98.5915**	**98.5915**	**98.5915**	97.1831
	Worst	92.9577	78.8732	67.6056	73.2394	**94.3662**	92.9577	92.9577	80.2817
	Mean	96.2441	89.3427	86.9014	87.1362	**96.9953**	96.6667	96.3380	90.0469
	Std	1.5399	4.4182	**6.6388**	5.6062	1.2670	1.5017	1.3129	3.8248
Tic-tac-toe	Best	**83.8542**	82.8125	81.2500	75.0000	83.3333	82.8125	80.2083	82.8125
	Worst	63.5417	**66.1458**	64.5833	58.3333	62.5000	60.4167	61.4583	62.5000
	Mean	**74.8611**	74.8090	72.2569	67.3611	74.8438	73.9063	71.0069	73.5069
	Std	5.6493	**5.7210**	5.6856	3.7491	5.3468	5.3716	5.4201	5.5183
Wine	Best	98.1481	**98.3025**	97.9938	98.1481	94.2901	97.9938	97.9938	97.9938
	Worst	58.9506	64.3519	**70.3704**	58.9506	68.2099	63.4259	68.2099	70.0617
	Mean	**88.5288**	87.1965	86.7850	80.6636	74.4907	77.5720	76.1317	77.1039
	Std	10.5828	10.3532	8.4638	**10.8958**	4.8683	7.1099	5.9302	5.8986
Zoo	Best	**99.5465**	**99.5465**	99.0930	98.6395	99.0930	98.8662	98.8662	99.0930
	Worst	92.0635	83.9002	79.3651	77.5510	**93.8776**	**93.8776**	**93.8776**	73.4694
	Mean	95.9940	95.2608	93.8927	92.7135	**96.6818**	96.1451	95.7974	96.0771
	Std	2.1449	4.4568	**5.8461**	5.3953	1.6869	1.7156	1.6262	5.4640

Note: Bold values indicate the best-performing result for each statistical measure within each dataset.

are observed in Tic-tac-toe and Wine. Increasing α_1 generally places greater emphasis on predictive performance, while decreasing β_2 reduces the relative contribution of minimizing the subset size. This behavior is particularly evident in the case of Tic-tac-toe, where higher values of α_1 improve the F1-Score but are accompanied by an increase in the average number of selected features. In contrast, BreastCancer and Ionosphere show a more stable number of selected features across all tested configurations. In general, no single combination of weights on its own yields the best result for all datasets and classifiers. However, intermediate settings such as $\alpha_1 = 0.90$ and $\beta_2 = 0.10$ provide a favorable balance between classification performance and dimensionality reduction in all evaluated cases. These results confirm the importance of properly balancing both objectives in the wrapper-based fitness function.

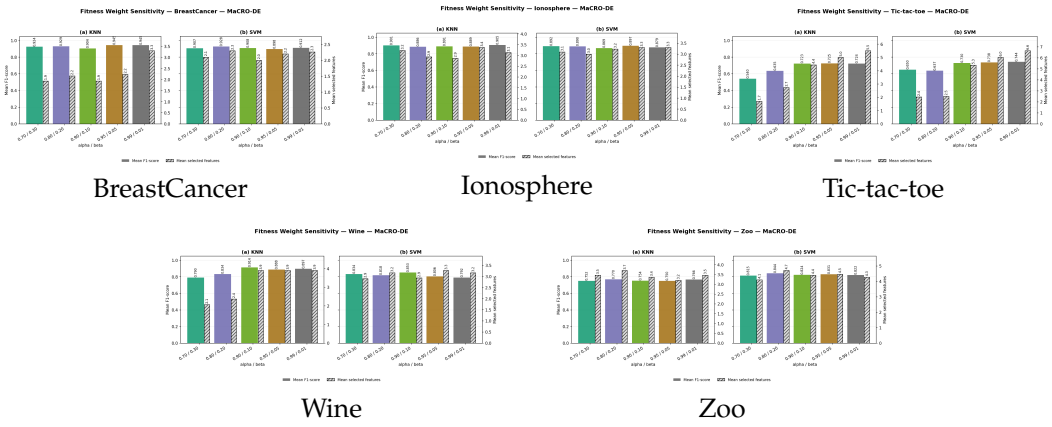


Figure S11. Sensitivity analysis of the fitness-function weights α_1 and β_2 on representative feature-selection datasets.

Supplementary Note S12: Extendend Feature Selection Results

This supplementary analysis provides a more detailed examination of the feature selection performance of MaCRO-DE and competing metaheuristic algorithms on the evaluated datasets. In addition to the aggregated numerical results presented in the main manuscript, this section illustrates the methods' behavior from complementary perspectives, including predictive performance, feature-reduction capacity, computational cost, convergence dynamics, and result variability. These visualizations provide a broader view of the methods' consistency and robustness across datasets with different characteristics.

Figure S12 summarizes the performance profiles of the evaluated metaheuristic algorithms per dataset in terms of Accuracy, F1-Score, Precision, Recall, and feature efficiency. Overall, the results indicate that the compared methods achieve competitive classification performance across datasets, though their relative performance varies with problem characteristics. MaCRO-DE generally maintains a balanced profile, achieving competitive predictive performance while preserving an appropriate balance with feature efficiency. Fig. S13 shows the average number of selected features and the computational runtime for each method and dataset. The results reveal notable differences in the compactness of the selected subsets and the associated runtimes. Some optimizers produce smaller subsets at the cost of longer execution times, while others provide faster execution with less aggressive dimensionality reduction. In this context, MaCRO-DE demonstrates a stable balance between subset size and computational cost across the evaluated datasets.

The convergence behavior shown in Fig. S14 further illustrates the search dynamics of the compared optimizers. Across several datasets, fitness values decrease rapidly during the first few iterations and then stabilize gradually as the search progresses. MaCRO-DE generally exhibits a smooth, consistent convergence pattern, suggesting the proposed search

strategy effectively balances exploration and exploitation during optimization. Finally, Fig. S15 shows the distribution of recall values obtained across all datasets and independent runs. The violin-shaped distributions demonstrate that most algorithms exhibit similar central tendencies, though variations in dispersion and robustness are evident. MaCRO-DE exhibits competitive mean and median recall values with moderate variability, supporting its suitability as a reliable wrapper-based feature selection approach under the KNN classifier.

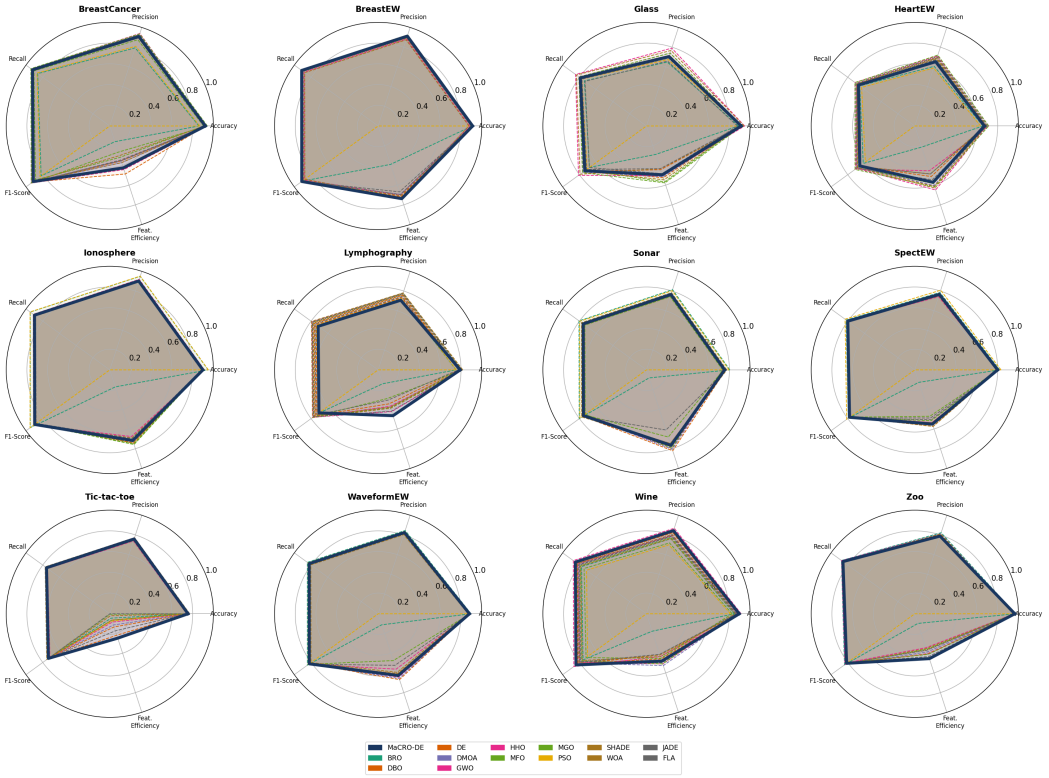


Figure S12. Detailed performance profiles of the evaluated metaheuristic algorithms across the feature-selection datasets considering Accuracy, Precision, Recall, F1-Score, and feature efficiency.

To complement the graphical analysis of feature selection presented above, Tables S10 and S11 show the best, worst, and standard deviation values obtained in the independent runs. These additional statistics expand on the average results presented in the main manuscript by showing the range and variability of classification accuracy and the number of features selected by the evaluated algorithms.

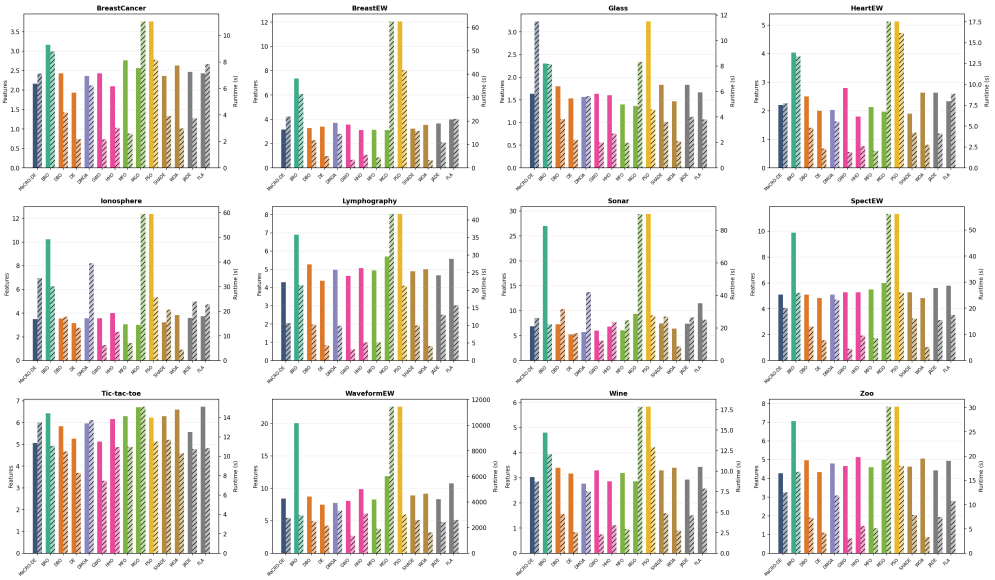


Figure S13. Average number of selected features and computational runtime obtained by the evaluated metaheuristic algorithms across the feature-selection datasets.

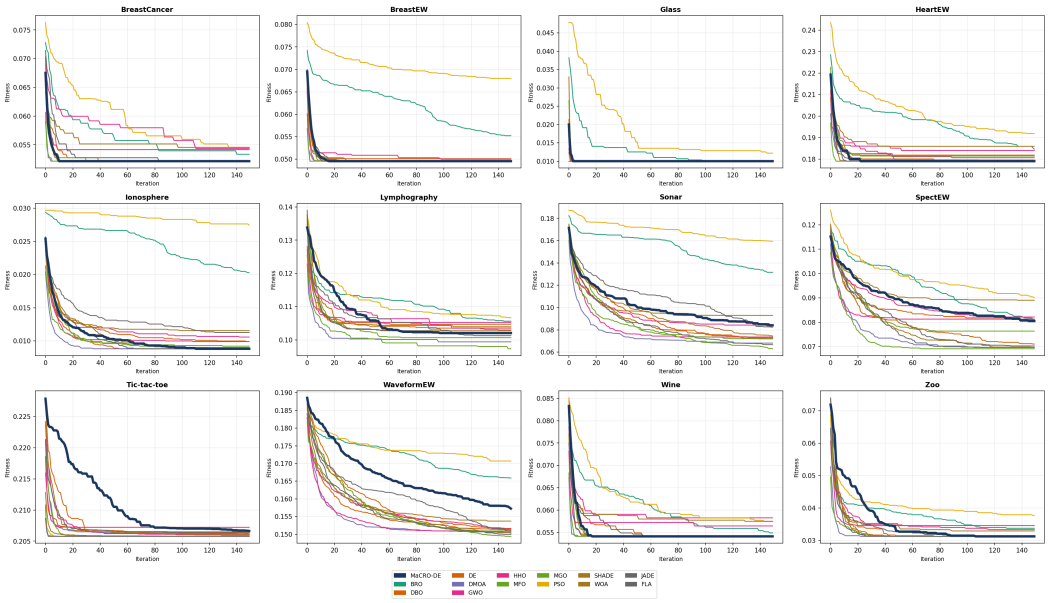


Figure S14. Convergence behavior of the evaluated metaheuristic algorithms across the feature-selection datasets.

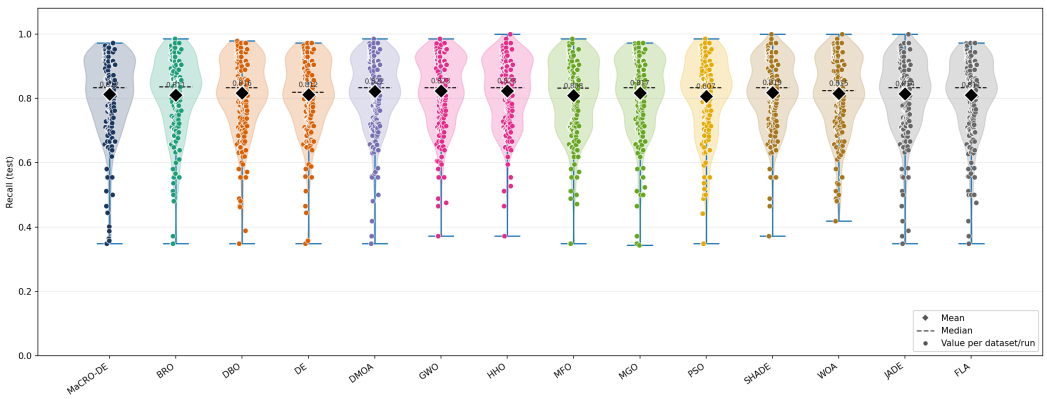


Figure S15. Distribution of Recall values obtained by the evaluated metaheuristic algorithms across all feature-selection datasets and independent runs.

Table S10. Best, worst, and standard deviation of the classification Accuracy (%) obtained by the evaluated algorithms across the feature-selection datasets.

Dataset	Stat.	MaCRO-DE	BRO	DBO	DE	DMOA	GWO	HHO	MFO	MGO	PSO	SHADE	WOA	JADE	FLA
BreastCancer	Best	97.1429	**97.8571**	**97.8571**	97.1429	97.1429	97.1429	**97.8571**	**97.8571**	97.1429	97.1429	97.1429	97.1429	**97.8571**	97.1429
	Worst	65.7143	65.7143	65.7143	65.7143	65.7143	65.7143	**72.1429**	65.7143	65.7143	65.7143	65.7143	65.7143	65.7143	65.7143
	Std	6.7270	11.7111	7.0414	5.8388	5.9010	6.7627	**3.9063**	7.0489	9.7023	10.6467	6.2007	7.5475	6.3788	8.6481
BreastEW	Best	95.6140	95.6140	95.6140	95.6140	95.6140	95.6140	95.6140	95.6140	95.6140	**97.3684**	95.6140	95.6140	95.6140	95.6140
	Worst	68.4211	71.0526	59.6491	59.6491	68.4211	60.5263	68.4211	58.7719	57.8947	**78.9474**	68.4211	57.0175	57.0175	57.0175
	Std	7.1458	4.8852	8.5939	9.1955	8.0238	8.8788	7.8764	10.4023	8.1281	**2.8090**	9.1854	8.5101	9.8813	7.8522
Glass	Best	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**	**99.6755**
	Worst	63.5479	58.8967	58.8967	58.8967	63.5479	58.8967	58.8967	63.5479	58.8967	63.5479	65.5489	**66.6306**	63.5479	58.8967
	Std	12.6252	12.4691	12.5258	11.8754	11.0835	11.3488	12.8291	12.8143	12.4850	**10.7351**	11.0414	11.5218	11.2466	12.6966
HeartEW	Best	**81.4815**	**81.4815**	79.6296	77.7778	77.7778	79.6296	79.6296	75.9259	79.6296	75.9259	79.6296	77.7778	77.7778	**81.4815**
	Worst	44.4444	48.1481	46.2963	44.4444	48.1481	50.0000	44.4444	**55.5556**	42.5926	50.0000	50.0000	42.5926	44.4444	42.5926
	Std	6.7024	7.4961	6.5572	5.7103	5.3942	6.2945	5.7703	**4.0413**	6.9988	7.7126	5.7479	7.7522	6.0173	7.4032
Ionosphere	Best	95.7746	98.5915	97.1831	97.1831	98.5915	98.5915	98.5915	98.5915	97.1831	98.5915	**100.0000**	98.5915	97.1831	97.1831
	Worst	73.2394	**83.0986**	78.8732	73.2394	80.2817	70.4225	**83.0986**	78.8732	78.8732	81.6901	73.2394	77.4648	80.2817	76.0563
	Std	5.4925	3.8031	4.8684	5.2488	4.4425	5.4400	**3.7854**	4.5315	4.9777	4.1179	5.3423	5.3457	4.2131	4.9010
Lymphography	Best	91.5556	91.5556	90.3333	87.1111	93.5556	**96.7778**	93.5556	93.5556	93.5556	**96.7778**	92.1111	93.5556	94.7778	94.7778
	Worst	59.3333	59.3333	60.0000	56.1111	**46.4444**	43.7778	59.3333	**61.3333**	59.8889	53.4444	**61.3333**	52.8889	56.1111	52.8889
	Std	7.1809	7.5481	7.6812	8.0615	8.7563	9.9643	7.7904	7.1137	**6.8170**	8.4339	7.5186	9.0200	7.1142	9.4224
Sonar	Best	90.4762	**92.8571**	90.4762	90.4762	85.7143	**92.8571**	85.7143	88.0952	**92.8571**	**92.8571**	88.0952	88.0952	88.0952	90.4762
	Worst	52.3810	**71.4286**	59.5238	59.5238	54.7619	59.5238	59.5238	59.5238	50.0000	**71.4286**	57.1429	57.1429	61.9048	57.1429
	Std	7.4843	5.1364	6.9703	5.8336	7.8683	7.8796	6.3141	6.9602	7.7732	**4.5786**	6.4886	6.7812	6.2687	6.6118
SpectEW	Best	85.1852	85.1852	85.1852	83.3333	85.1852	85.1852	85.1852	85.1852	85.1852	**88.8889**	85.1852	83.3333	87.0370	85.1852
	Worst	72.2222	72.2222	**75.9259**	74.0741	**75.9259**	**75.9259**	74.0741	66.6667	70.3704	74.0741	72.2222	**75.9259**	74.0741	74.0741
	Std	2.3116	3.3730	1.8350	1.9277	1.6873	1.6919	1.7541	2.5355	2.0765	3.6701	2.1372	**1.4412**	2.2725	2.0690
Tic-tac-toe	Best	82.8125	**83.8542**	83.3333	82.8125	83.3333	**83.8542**	**83.8542**	83.3333	**83.8542**	**83.8542**	**83.8542**	83.3333	82.8125	**83.8542**
	Worst	59.8958	**64.0625**	41.1458	58.8542	61.9792	41.1458	51.5625	59.3750	60.4167	56.2500	54.1667	59.8958	59.8958	63.0208
	Std	5.5589	4.8216	7.0284	5.5246	4.9178	7.2584	6.9603	5.3014	5.2789	5.5436	6.7473	6.1248	5.6960	**4.7019**
Wine	Best	**100.0000**	**100.0000**	98.3025	98.1481	98.1481	**100.0000**	**100.0000**	**100.0000**	**100.0000**	**100.0000**	**100.0000**	**100.0000**	**100.0000**	**100.0000**
	Worst	64.3519	**70.3704**	58.9506	**70.3704**	64.3519	**70.3704**	66.2037	63.4259	64.3519	70.0617	**70.3704**	**70.3704**	58.9506	64.3519
	Std	7.7872	9.2510	9.1475	7.7610	8.5340	**6.2761**	7.5632	9.9405	9.6158	8.6279	8.7654	8.9786	9.9054	8.9361
WaveformEW	Best	**91.5289**	91.0620	90.4613	91.4641	91.1289	90.7254	91.2648	90.8014	**91.5289**	91.4606	91.3289	90.4627	90.7303	90.7944
	Worst	80.9944	78.8501	**82.1965**	82.1128	81.3762	76.4193	79.7177	81.6021	**82.3797**	81.1776	78.9755	75.6396	80.6641	76.4221
	Std	2.5205	2.6689	2.1343	2.1368	2.2932	2.6722	2.6712	2.0840	**1.9676**	2.2653	2.5415	3.0974	2.4815	2.6920
Zoo	Best	99.0930	**99.5465**	**99.5465**	99.0930	99.0930	99.0930	99.0930	99.0930	99.0930	**99.5465**	98.6395	98.6395	99.3197	98.6395
	Worst	81.6327	**88.8889**	79.3651	77.5510	69.8413	75.7370	75.7370	79.3651	78.0045	78.9116	78.9116	78.9116	86.1678	75.7370
	Std	3.3900	**2.3690**	2.8604	4.0647	4.3057	4.7792	5.6940	4.2939	5.0252	3.0507	3.9353	3.2832	3.4795	3.8224

Note: Bold values indicate the best-performing result for each statistical measure within each dataset.

Table S11. Best, worst, and standard deviation of the number of selected features obtained by the evaluated algorithms across the feature-selection datasets.

Dataset	Stat.	MaCRO-DE	BRO	DBO	DE	DMOA	GWO	HHO	MFO	MGO	PSO	SHADE	WOA	JADE	FLA
BreastCancer	Best	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Worst	**4.0000**	5.0000	**4.0000**	**4.0000**	**4.0000**	5.0000	**4.0000**	5.0000	7.0000	7.0000	5.0000	6.0000	**4.0000**	5.0000
	Std	**0.7507**	1.1850	0.8294	0.8629	0.9892	1.0591	0.8862	1.1273	1.1263	1.4066	0.9806	1.1570	0.9654	1.1026
BreastEW	Best	**1.0000**	2.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	6.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Worst	**6.0000**	16.0000	**6.0000**	**6.0000**	**6.0000**	**6.0000**	7.0000	**6.0000**	**6.0000**	22.0000	**6.0000**	5.0000	**6.0000**	7.0000
	Std	**1.0279**	3.2519	1.1423	1.1213	1.1658	1.2688	1.0715	1.1061	1.1802	3.4996	1.2849	1.1687	1.2812	1.4878
Glass	Best	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Worst	5.0000	5.0000	5.0000	4.0000	**3.0000**	4.0000	4.0000	4.0000	5.0000	8.0000	7.0000	**3.0000**	4.0000	4.0000
	Std	0.7912	0.9794	0.9892	0.7287	**0.6223**	0.7433	0.7477	0.8102	0.9478	1.4186	1.2469	0.6369	0.8397	0.8227
HeartEW	Best	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Worst	5.0000	9.0000	5.0000	5.0000	**4.0000**	5.0000	**4.0000**	5.0000	5.0000	10.0000	5.0000	6.0000	6.0000	5.0000
	Std	0.9771	1.8910	1.0301	0.9259	0.8727	0.9806	**0.8045**	0.8610	0.8995	1.6823	0.9418	1.1687	1.1560	0.9543
Ionosphere	Best	**1.0000**	6.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	2.0000	**1.0000**	8.0000	**1.0000**	**1.0000**	2.0000	**1.0000**
	Worst	7.0000	19.0000	9.0000	7.0000	**6.0000**	**6.0000**	7.0000	**5.0000**	**6.0000**	24.0000	**6.0000**	9.0000	**6.0000**	10.0000
	Std	1.1542	3.3629	1.4359	1.1601	1.2277	1.2139	1.3031	**0.9543**	1.0442	3.7199	1.0495	1.6270	1.0809	1.9473
Lymphography	Best	2.0000	3.0000	2.0000	2.0000	**1.0000**	**1.0000**	2.0000	2.0000	3.0000	4.0000	2.0000	3.0000	**1.0000**	2.0000
	Worst	7.0000	13.0000	9.0000	**6.0000**	8.0000	8.0000	11.0000	11.0000	12.0000	14.0000	9.0000	10.0000	9.0000	14.0000
	Std	1.1493	2.3060	1.5455	**0.9020**	1.6971	1.6221	2.0592	2.1070	2.1240	2.1691	1.6030	1.6464	1.6166	2.1191
Sonar	Best	**2.0000**	15.0000	**2.0000**	**2.0000**	3.0000	**2.0000**	3.0000	3.0000	**2.0000**	16.0000	**2.0000**	**2.0000**	3.0000	3.0000
	Worst	17.0000	39.0000	18.0000	10.0000	**9.0000**	14.0000	15.0000	14.0000	20.0000	40.0000	14.0000	19.0000	13.0000	25.0000
	Std	2.3003	5.2201	2.9016	1.6413	**1.3830**	1.8802	2.0333	2.4298	5.1265	5.9155	2.2599	2.8274	2.2388	4.5387
SpectEW	Best	**1.0000**	4.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	5.0000	2.0000	**1.0000**	**1.0000**	**1.0000**
	Worst	8.0000	16.0000	7.0000	8.0000	**7.0000**	9.0000	10.0000	8.0000	9.0000	17.0000	**7.0000**	10.0000	9.0000	10.0000
	Std	1.5731	2.8310	1.6689	1.5348	1.4786	1.7153	1.9641	1.8964	2.0541	2.6320	**1.3302**	1.8338	1.7914	2.1579
Tic-tac-toe	Best	2.0000	3.0000	**1.0000**	**1.0000**	3.0000	**1.0000**	2.0000	2.0000	3.0000	**1.0000**	2.0000	2.0000	2.0000	2.0000
	Worst	**7.0000**	9.0000	8.0000	**7.0000**	8.0000	8.0000	8.0000	9.0000	8.0000	9.0000	9.0000	9.0000	**7.0000**	9.0000
	Std	1.2738	1.3415	1.4783	1.4162	1.2936	1.5741	1.5013	1.5205	1.5499	1.4540	1.6454	1.6257	**1.2469**	1.6351
Wine	Best	2.0000	3.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	2.0000	2.0000	**1.0000**	**1.0000**	**1.0000**
	Worst	7.0000	11.0000	9.0000	7.0000	**6.0000**	**6.0000**	**6.0000**	7.0000	7.0000	9.0000	7.0000	8.0000	**6.0000**	9.0000
	Std	1.2818	1.6121	1.7647	1.3339	**1.1716**	1.1802	1.3715	1.3144	1.4714	1.7476	1.4391	1.7656	1.2469	1.6408
WaveformEW	Best	**4.0000**	15.0000	**4.0000**	**4.0000**	**4.0000**	5.0000	**4.0000**	**4.0000**	7.0000	10.0000	5.0000	**4.0000**	**4.0000**	5.0000
	Worst	14.0000	27.0000	17.0000	**11.0000**	12.0000	13.0000	17.0000	16.0000	23.0000	28.0000	15.0000	19.0000	14.0000	19.0000
	Std	2.1270	2.8214	2.5565	1.7720	**1.6382**	1.8365	2.7202	2.5569	3.3653	2.7862	1.8142	2.7002	2.0773	3.2337
Zoo	Best	2.0000	3.0000	2.0000	2.0000	**1.0000**	2.0000	**1.0000**	**1.0000**	2.0000	3.0000	2.0000	3.0000	2.0000	**1.0000**
	Worst	7.0000	11.0000	7.0000	7.0000	8.0000	7.0000	8.0000	7.0000	8.0000	11.0000	**6.0000**	7.0000	7.0000	10.0000
	Std	**1.1026**	2.0724	1.2671	1.1813	1.2870	1.2599	1.8633	1.2527	1.3700	1.9968	1.1998	1.2679	1.1716	1.5884

Note: Bold values indicate the best-performing result for each statistical measure within each dataset.