

Article

Not peer-reviewed version

---

# HE-DEPSO algorithm for textures optimization with application in particle physics

---

[Emmanuel Martínez-Guerrero](#) , [Pedro Lagos Eulogio](#) , [Pedro Miranda-Romagnoli](#) <sup>\*</sup> ,  
[Roberto Noriega-Papaqui](#) , [Juan Carlos Seck-Tuoh-Mora](#) <sup>\*</sup>

Posted Date: 16 August 2024

doi: 10.20944/preprints202408.1171.v1

Keywords: Optimization; High energy physics; Differential evolution; Particle swarm optimization



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# HE-DEPSO Algorithm for Textures Optimization with Application in Particle Physics

Emmanuel Martínez-Guerrero <sup>1,†</sup> , Pedro Lagos-Eulogio <sup>2</sup> , Pedro Miranda-Romagnoli <sup>2,\*,†</sup> , Roberto Noriega-Papaqui <sup>2</sup> and Juan Carlos Seck-Tuoh-Mora <sup>3,\*</sup> 

<sup>1</sup> Área Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, Pachuca, Mexico

<sup>2</sup> Área Académica de Matemáticas y Física, Universidad Autónoma del Estado de Hidalgo, Pachuca, Mexico

<sup>3</sup> Área Académica de Ingeniería Industrial, Universidad Autónoma del Estado de Hidalgo, Pachuca, Mexico

\* Correspondence: pmiranda@uaeh.edu.mx (P.M.-R.); jseck@uaeh.edu.mx (J.C.S.-T.-M.)

† These authors contributed equally to this work.

**Abstract:** Within the phenomenology of particle physics, the theoretical model of 4-zero textures is validated using a chi-square criterion that compares experimental data with the computational results of the model. Traditionally, analytical methods that often imply simplifications, combined with computational analysis, have been used to validate texture models. In this paper, we propose a new meta-heuristic variant of the differential evolution algorithm that incorporates aspects of the particle swarm optimization algorithm called "HE-DEPSO" to obtain chi-squared values are less than a bound value, which exhaustive and traditional algorithms cannot obtain. The results show that the proposed algorithm can optimize the chi-square function according to the required criteria. We compare simulated data with experimental data in the allowed search region, thereby validating the 4-zero texture model.

**Keywords:** optimization; high energy physics; differential evolution; particle swarm optimization

## 1. Introduction

Broadly speaking, physics can be classified into three areas: the theoretical, which, through a mathematical formalism, explains and understands the physical phenomena; the experimental part, which groups disciplines related to the acquisition of data, the methods of data acquisition, the design and performance of experiments; finally, the aspect that links these two is known as phenomenology, which is in charge of 1) validating theoretical models, 2) how to measure their parameters, 3) how to distinguish one model from another and 4) studying the experimental consequences of these models. This paper addresses a phenomenological approach by providing a solution to the validation of a theoretical model of physics. More specifically, from particle physics to unveil the mechanisms of fermion mass generation and to reproduce the elements of the  $V_{CKM}$  matrix known as the mixing matrix.

The beginning of these models goes back to the first years of the 1970s, shortly after the establishment of the Standard Model (SM) of Particle Physics. Since then, different approaches have been developed in the context of theoretical and phenomenological models, which can be broadly classified as follows: Radiative mechanisms [1,2]; Textures [3–5]; Symmetries between families [6,7]; and Seesaw mechanisms [8–10]. These approaches are interrelated.

The texture formalism was born by considering that certain entries of the mass matrix are zero, such that we can compute analytically the matrix that diagonalizes it and hence the matrix  $V_{CKM}$ . In 1977 Harald Fritzsch created this formalism by using 6-zero hermetic textures as a viable model [11], in 2005 with experimental data from those years, he found 4-zero hermetic textures viable to generate the quark masses and the mixing matrix [12], however, given the precision of the current experimental data, it is worthwhile to assess the feasibility of these texture models.

There are numerical works on 4-zero texture models; however, the detailed numerical description of the techniques and algorithms used are not described by their authors [13]. The numerical analysis of texture models requires a  $\chi^2$  criterion that establishes when the theoretical part of the model agrees

with the experimental part [14]. That is, to validate the model, a function (which we will call  $\chi^2(X)$ ) is built, and permissible values of the free parameters  $X$  of the model must be found such that this function takes the minimum value possible greater than 0 but less than 1. In other words, it is necessary to optimize said function under a certain threshold. For the particular case of the function  $\chi^2(X)$  obtained from the texture formalism, the difficulty comes from the cumbersome algebraic manipulation of the expressions involved, which complicate and entangle the application of classic optimization techniques; thus, alternative optimization techniques are required.

To the best of the authors' knowledge, the works where bio-inspired optimization algorithms have been used within particle physics are the following: In experimental contexts, the particle swarm optimization algorithm (PSO) as well as genetic algorithms (GA), have been implemented within the hyperparameter optimization of machine learning (ML) models used in the analysis of data obtained in high-energy physics experiments [15]; The optimization of the design of particle accelerators where the differential evolution (DE) algorithm is quite effective [16,17]; Regarding phenomenology, genetic algorithms have been used to discriminate models from supersymmetric theories [18].

As can be seen, the incursion of bio-inspired optimization algorithms in particle physics has been very limited, even when the results are favorable. For this reason, further application of these techniques and algorithms in this type of particle physics area, especially in texture formalism, is of great interest.

The DE algorithm is one of the evolutionary algorithms that has stood out the most in recent times due to its simplicity, power, and efficiency [19,20]; however, like other evolutionary algorithms, it is likely to suffer premature convergence and stagnation in local minima [19,21]. The strategies implemented to solve these problems can be classified as follows [22]:

1. *Change the mutation strategy.* The mutation phase of the DE algorithm is important since it allows the integration of new individuals into the existing population, thus influencing its performance. Algorithms such as CoDE [23] have introduced new mutation strategies and implemented different mutation strategies, respectively, in order to improve the efficiency of the DE algorithm.
2. *Change the parameter control.* The DE algorithm is sensitive to the configuration of its main parameters: the scale factor  $F$  and the crossover probability  $Cr$  [19,24]. Self-adaptive control of these parameters has been shown to improve the performance of the DE algorithm significantly. In this sense, the SHADE [25] and SaDE [26] algorithms represent two fairly well-known variants.
3. *Incorporation of population mechanisms.* The way in which the population is handled in the DE algorithm can improve its performance. Techniques such as genotypic topology [27], opposition-based initialization [28], and external population archives, as seen in JADE [29], have shown positive effects.
4. *Hybridization with other optimization algorithms.* One way to improve the performance of the DE algorithm is to take advantage of the operators' strengths from other algorithms and incorporate them into the structure of the DE algorithm through a hybridization process [20,30]. Hybridization with other computational intelligence algorithms, such as Artificial Neural Networks (ANN), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO), has marked a trend within the last decade [20,31].

The self-adaptive Differential Evolution algorithm based on Particle Swarm Optimization (DEPSO) [32] is a recent variant of the DE algorithm that integrates the PSO mutation strategy within its structure. DEPSO employs a probabilistic selection technique to choose between two mutation strategies: a new mutation strategy with an elite file called DE/e-rand/1, a modification of the DE/rand/1 strategy, and the mutation strategy of the PSO algorithm. This probabilistic selection technique enables DEPSO to

improve the balance between exploitation and exploration, resulting in significantly better performance compared to both DE and PSO on various single-objective optimization problems.

This paper proposes a new variant of the DEPSO algorithm called Historical Elite Differential Evolution Based on Particle Swarm Optimization (HE-DEPSO) to improve optimization performance in solving complex single-objective problems, with a specific focus on addressing the problem of optimizing the  $\chi^2$  criteria for the 4-zero texture model of high energy physics. The proposed variant aims to explore areas of opportunity encountered to enhance DEPSO, specifically introducing a new mutation strategy named DE/current-to-EHE/1, which utilizes information from the elite individuals of the population and incorporates historical data from the evolutionary process to improve the balance between exploration and exploitation by leveraging information from elite individuals and historical data, particularly during the early stages of the evolutionary process. Additionally, HE-DEPSO employs the self-adaptive parameter control mechanism from the SHADE algorithm to reduce the sensitivity of the algorithm's parameters. To test the HE-DEPSO algorithm's performance, it was compared against other optimization algorithms, including DE, PSO, CoDE, SHADE, and DEPSO, using the CEC 2017 single-objective benchmark function set. The HE-DEPSO algorithm outperformed the other algorithms in terms of solution quality. Finally, the validation of the 4-zeros texture model was conducted, optimizing the  $\chi^2$  function and, at the same time, comparing the performance of our proposal against DE, PSO, CoDE, SHADE, and DEPSO algorithms for this particular application. The results are encouraging and expand the use of bio-inspired methods in high-energy physics by integrating a metaheuristic approach.

The remainder of the paper is structured as follows: Chapter 2 contains the definition of the problem to be solved, including the definition of the  $\chi^2$  criterion; Chapter 3 reviews the versions of the PSO and DEPSO algorithms used here. Chapter 4 explains our proposal, the HE-DEPSO algorithm; Chapter 5 subjects our algorithm to benchmark tests, while Chapter 6 presents the validation problem of the 4-zero texture model. Finally, in chapter 7, the conclusions are presented.

## 2. Problem Definition

In the SM scope, the masses of the quarks come from hermetic  $3 \times 3$  matrices known as mass matrices (one for u-type quarks and another for d-type quarks)[33–35], as the absolute values of their eigenvalues and the matrix  $V_{CKM}$  as the product of the matrix that diagonalizes the u-type quarks and the matrix that diagonalizes the d-type quarks, however, due to the mathematical formalism used, the mass matrices remain entirely unknown and consequently, the masses of the quarks and the mixing matrix  $V_{CKM}$  cannot be theoretically predicted. The experiment provides us with the numerical value of these quantities.

The mixing matrix  $V_{CKM}$ , can be written in general way as:

$$V_{CKM} = \begin{pmatrix} (V_{CKM})_{ud} & (V_{CKM})_{us} & (V_{CKM})_{ub} \\ (V_{CKM})_{cd} & (V_{CKM})_{cs} & (V_{CKM})_{cb} \\ (V_{CKM})_{td} & (V_{CKM})_{ts} & (V_{CKM})_{tb} \end{pmatrix}, \quad (1)$$

and is a unitary matrix containing information about the probability of transition between u-type quarks (left-hand index) and d-type quarks (right-hand index), through the weak interaction [36]. That is:  $|(V_{CKM})_{us}|$  quantifies the transition probability between the quark  $u$  and the quark  $s$  through the interaction with the boson  $W^\pm$ . Over many years and different collaborations, the magnitudes of the elements of the mixing matrix  $V_{CKM}$  have been experimentally measured with an accuracy up to  $10^{-5}$  and it is well known that four quantities (three angles and one phase) are needed to have a parameterization that adequately describes the matrix  $V_{CKM}$ [37,38]. In this work we choose the Chau-Keng parametrization [37] and the three corresponding angles  $\theta_{13}$ ,  $\theta_{12}$ ,  $\theta_{23}$  are obtained by choosing the three magnitudes of the elements  $|(V_{CKM})_{us}|$ ,  $|(V_{CKM})_{ub}|$  and  $|(V_{CKM})_{cb}|$  and the phase  $\delta_{13}$  is obtained from the Jarlslog invariant ( $J$ ) through the following relations:

$$\sin \theta_{13} = |(V_{CKM})_{ub}|, \quad (2)$$

$$\sin \theta_{12} = \frac{|(V_{CKM})_{us}|}{\sqrt{1 - |(V_{CKM})_{ub}|^2}}, \quad (3)$$

$$\sin \theta_{23} = \frac{|(V_{CKM})_{cb}|}{\sqrt{1 - |(V_{CKM})_{ub}|^2}}, \quad (4)$$

$$\sin \delta_{13} = \frac{J}{\sqrt{1 - (\sin \theta_{12})^2} \sin \theta_{12} [1 - (\sin \theta_{13})^2] \sin \theta_{13} \sqrt{1 - (\sin \theta_{23})^2} \sin \theta_{23}}, \quad (5)$$

where  $J = \text{Im}(V_{us} V_{cb} V_{ub}^* V_{cs}^*)$ . Hence, we choose  $|(V_{CKM})_{us}|$ ,  $|(V_{CKM})_{ub}|$ ,  $|(V_{CKM})_{cb}|$  and  $J$  as independent quantities.

Within the SM context, the mixing matrix,  $V_{CKM}$ , is defined by [37]:

$$V_{CKM} = U_u^\dagger U_d,$$

where  $U_u$  is the matrix that diagonalizes the mass matrix of the u-type quarks, and  $U_d$  is the matrix that diagonalizes the d-type quarks. It is here where the texture formalism is born, which consists of proposing structures with zeros in some entries of the mass matrix in such a way that we can find the matrices that diagonalize it and be able to calculate analytically the mixing matrix  $V_{CKM}$  and validate the chosen texture structure.

Without loss of generality, the mass matrices  $M_u$  and  $M_d$  are considered hermitian so that the general mass structure has the form:

$$M_q = \begin{pmatrix} E_q & D_q & F_q \\ D_q^* & C_q & B_q \\ F_q^* & B_q^* & A_q \end{pmatrix}, \quad (6)$$

where the index  $q$  runs over the labels  $u, d$ . The elements  $A_q, C_q$  and  $E_q$  are real, while  $B_q, D_q$  and  $F_q$  are complex and are usually written in their polar form  $Z_q = |Z_q| e^{i\phi_{Z_q}}$  where  $|Z_q|$  is the magnitude and  $\phi_{Z_q}$  its angular phase ( $Z = B, D, F$ ).

A matrix of the type four-zero textures [12] is formed from the above matrix by taking the entries  $(1,1)$ ,  $(1,3)$  and  $(3,1)$  equal to zero. Thus, we arrive at the following matrix structure:

$$M_q = \begin{pmatrix} 0 & D_q & 0 \\ D_q^* & C_q & B_q \\ 0 & B_q^* & A_q \end{pmatrix}. \quad (7)$$

In references [3,12] it is shown that this matrix can be diagonalized by a unitary matrix as follows:

$$\begin{aligned} U_q M_q U_q^\dagger &= \text{diag}\{\lambda_1^q, \lambda_2^q, \lambda_3^q\}, \\ (O_q^T P_q) M_q (O_q^T P_q)^\dagger &= \text{diag}\{\lambda_1^q, \lambda_2^q, \lambda_3^q\}, \end{aligned}$$

where  $\text{diag}\{\lambda_1^q, \lambda_2^q, \lambda_3^q\}$  denotes a diagonal matrix and  $\lambda_i^q$  denotes each of the three eigenvalues of  $M_q$ . The matrices  $P_q$  and  $O_q$  are given by:

$$P_q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\phi_{D_q}} & 0 \\ 0 & 0 & e^{i(\phi_{D_q} + \phi_{B_q})} \end{pmatrix}, \quad (8)$$

and

$$O_q = \begin{pmatrix} (O_q)_{11} & (O_q)_{12} & (O_q)_{13} \\ (O_q)_{21} & (O_q)_{22} & (O_q)_{23} \\ (O_q)_{31} & (O_q)_{32} & (O_q)_{33} \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{\lambda_2^q \lambda_3^q (A_q - \lambda_1^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & \eta_q \sqrt{\frac{\lambda_1^q \lambda_3^q (\lambda_2^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_1^q \lambda_2^q (A_q - \lambda_3^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ -\eta_q \sqrt{\frac{\lambda_1^q (\lambda_1^q - A_q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & \sqrt{\frac{\lambda_2^q (A_q - \lambda_2^q)}{(\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_3^q (\lambda_3^q - A_q)}{(\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \\ \eta_q \sqrt{\frac{\lambda_1^q (A_q - \lambda_2^q) (A_q - \lambda_3^q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_1^q)}} & -\sqrt{\frac{\lambda_2^q (A_q - \lambda_1^q) (\lambda_3^q - A_q)}{A_q (\lambda_2^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} & \sqrt{\frac{\lambda_3^q (A_q - \lambda_1^q) (A_q - \lambda_2^q)}{A_q (\lambda_3^q - \lambda_1^q) (\lambda_3^q - \lambda_2^q)}} \end{pmatrix}, \quad (9)$$

Taking  $\lambda_3^q > 0$  and  $A_q > 0$ , the relations between  $\lambda_i^q$  with the physical masses of the quarks are:

$$(\lambda_1^u, \lambda_2^u, \lambda_3^u) = (-\eta_u m_u, \eta_u m_c, m_t), \quad (10)$$

$$(\lambda_1^d, \lambda_2^d, \lambda_3^d) = (-\eta_d m_d, \eta_d m_s, m_b), \quad (11)$$

where  $m_u$  is the u quark mass,  $m_c$  is the c quark mass,  $m_t$  is the t quark mass,  $m_d$  is the d quark mass,  $m_s$  is the s quark mass, and  $m_b$  is the b quark mass and its experimental value is presented in Appendix A. In this work, the same four-zero texture structure is considered for the u-quark mass matrix and the d-type quark mass matrix (a parallel mass structure). The  $q$ -index takes two values  $q = u$  and  $q = d$ .

From the Equation (9), it is noticed that the elements of the matrix  $O_q$  depend on the free parameters  $\eta_q$  and  $A_q$ , the first parameter takes values of +1 and -1 and tells us that eigenvalue of the mass matrix is negative, when  $\eta_q = 1$  the first eigenvalue  $\lambda_1^q$  is negative and the second eigenvalue  $\lambda_2^q$  is positive, when  $\eta_q = -1$  the first eigenvalue  $\lambda_1^q$  is positive and the second eigenvalue  $\lambda_2^q$  is negative. The combinations of signs of the  $\eta_u$  and  $\eta_d$  parameters define the different cases of study to be considered (see Table 1). The second pair of free parameters are  $A_u$  and  $A_d$  whose values are restricted to the intervals  $m_c < A_u < m_t$  and  $m_s < A_d < m_b$  to ensure that the elements of the  $O_u$  and  $O_d$  matrices are real.

**Table 1.** Cases of study.

| Case | $\eta_u$ | $\eta_d$ |
|------|----------|----------|
| 1    | +1       | +1       |
| 2    | +1       | -1       |
| 3    | -1       | +1       |
| 4    | -1       | -1       |

From the above, the mixing matrix  $V_{CKM}$ , predicted by the four-texture model, is given by:

$$V_{CKM}^{th} \equiv O_u^\dagger P_u P_d^\dagger O_d, \quad (12)$$

in an explicit form:

$$(V_{CKM}^{th})_{ij} = (O_u)_{1i}(O_d)_{1j} + (O_u)_{2i}(O_d)_{2j}e^{i\phi_1} + (O_u)_{3i}(O_d)_{3j}e^{i(\phi_1+\phi_2)}, \quad (13)$$

where the phases  $\phi_1$  and  $\phi_2$  are defined as:

$$\phi_1 = \phi_{D_u} - \phi_{D_d}, \quad \phi_2 = \phi_{B_u} - \phi_{B_d}. \quad (14)$$

These are considered to be measured in radians with principal argument  $[0, 2\pi]$ , and the indices  $i$  and  $j$  correspond, respectively, to the indices  $(u, c, t)$  and  $(d, s, b)$ .

The magnitude of the elements of the mixing matrix is given by:

$$\begin{aligned}
|(V_{CKM})_{ij}^{th}| &= \left\{ [(O_u)_{1i}(O_d)_{1j}]^2 + [(O_u)_{2i}(O_d)_{2j}]^2 + [(O_u)_{3i}(O_d)_{3j}]^2 \right. \\
&+ 2[(O_u)_{1i}(O_d)_{1j}][(O_u)_{2i}(O_d)_{2j}] \cos(\phi_1) \\
&+ 2[(O_u)_{1i}(O_d)_{1j}][(O_u)_{3i}(O_d)_{3j}] \cos(\phi_1 + \phi_2) \\
&\left. + 2[(O_u)_{2i}(O_d)_{2j}][(O_u)_{3i}(O_d)_{3j}] \cos(\phi_2) \right\}^{1/2}.
\end{aligned} \tag{15}$$

At this point, it is essential to emphasize that analytical expressions are obtained for the elements of the mixing matrix predicted by the four-zero texture formalism; with this information, it is possible to construct a more complete theory of the Standard Model where the origin of the mixing matrix  $V_{CKM}$  is explained.

It only remains to validate the theoretical model of textures with 4-zeros, that is, to find the range of the free parameters  $A_u$ ,  $A_d$ ,  $\phi_1$  y  $\phi_2$  that agree with the experimental values of the matrix  $V_{CKM}$  and for this, we define a function  $\chi^2(X)$  and use a chi-square criterion [39,40] established by:

$$0 < \frac{\chi^2(X)}{4} < 1, \tag{16}$$

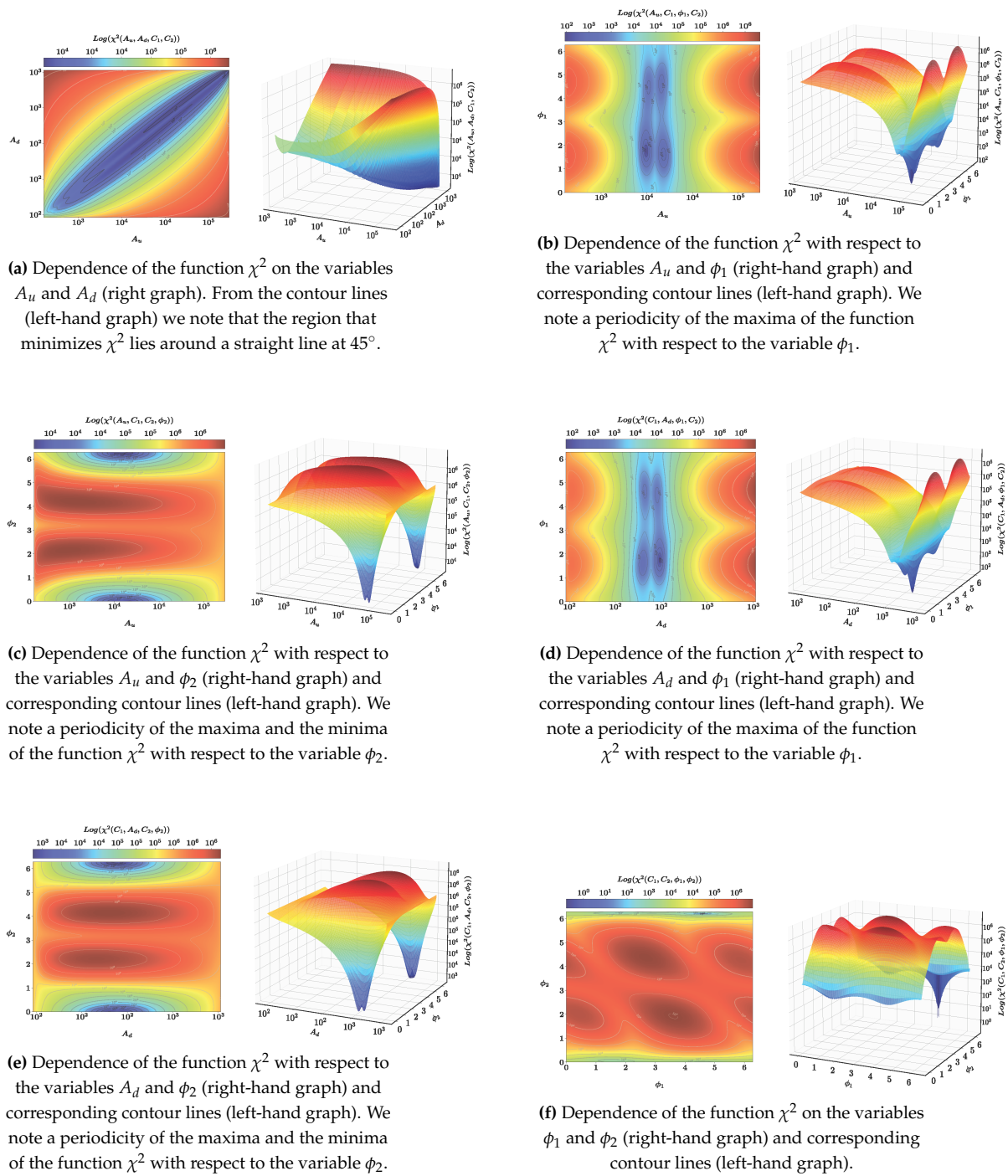
where  $X = (A_u, A_d, \phi_1, \phi_2)$  and

$$\begin{aligned}
\chi^2(X) &= \frac{\left( |(V_{CKM})_{us}^{th}(X)| - |V_{us}| \right)^2}{\sigma_{V_{us}}^2} + \frac{\left( |(V_{CKM})_{ub}^{th}(X)| - |V_{ub}| \right)^2}{\sigma_{V_{ub}}^2} + \frac{\left( |(V_{CKM})_{cb}^{th}(X)| - |V_{cb}| \right)^2}{\sigma_{V_{cb}}^2} \\
&+ \frac{\left( J^{th}(X) - J \right)^2}{\sigma_{J_q}^2},
\end{aligned} \tag{17}$$

the super-indexes “th” are given by the Equation (15) and the quantities without super-index are the experimental data with uncertainty  $\sigma_{V_{kl}}^2$  (see Equations (A1) and (A2)).

Although at first sight, the mathematically constructed function  $\chi^2(X)$  turns out to have a simple structure and the amount of free parameters is small, the difficulty of finding the numerical range of the same ones that fulfill the condition given in the Equation (16) comes from the cumbersome composition of functions to evaluate that constitute it. In order to have a notion of the topographic relief of this function, different projections of the function  $\chi^2(X)$  in different planes are shown in Figure 1. Figure 1a shows the projection onto the plane  $A_u$  and  $A_d$  (held fixed values for  $\phi_1 = c_1 = \text{const}$  and  $\phi_2 = c_2 = \text{const}$ ), that is; the dependence of the function  $\chi^2(X)$  on the variables  $A_u$  and  $A_d$  is shown (right graph) and the corresponding contour lines are illustrated (left graph). Similarly, Figure 1b corresponds to the projection of the function  $\chi^2(X)$  onto  $A_u$  and  $\phi_1$  (setting  $A_d = c_3 = \text{const}$ . and  $\phi_2 = c_4 = \text{const}$ ). Analogously in the graphical displays we show the behavior of the function  $\chi^2(X)$  in the plane  $A_u$  and  $\phi_2$  (Figure 1c), in the plane  $A_d$  and  $\phi_1$  (Figure 1d), in the plane  $A_d$  and  $\phi_2$  (Figure 1e), and in the  $\phi_1$  and  $\phi_2$  planes (Figure 1f).

The following color code has been established, regions towards the intense red color mean larger values of  $\chi^2(X)$ , while regions towards the intense blue color correspond to smaller values of  $\chi^2(X)$ . We point out that the function to optimize  $\chi^2(X) : R^4 \rightarrow R$  is defined on (17) with variables  $A_u$ ,  $A_d$ ,  $\phi_1$ ,  $\phi_2$  bounded on the boundaries  $[m_c, m_t]$ ,  $[m_s, m_b]$ ,  $[0, 2\pi]$ ,  $[0, 2\pi]$  and as can be observed from the graphs (Figure 1), the topography in which it is intended to optimize is rugged. With this, one sees the opportunity to explore the use of alternative optimization techniques such as bio-inspired algorithms.



**Figure 1.**  $\chi^2$  function projections over variables  $A_u$ ,  $A_d$ ,  $\phi_1$ ,  $\phi_2$ .

### 3. Review of PSO, DE and DEPSO

#### 3.1. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm [41] is one of the most popular swarm intelligence-based algorithms. It models the behavior of groups of individuals, such as flocks, to solve complex optimization problems in D-dimensional space. The algorithm performs a search for the global optimum using a swarm of  $NP$  particles. For the  $i$ -th particle of the swarm  $X_{i,t}$  ( $i = 1, 2, \dots, NP$ ), its position and velocity at iteration  $t$  are represented by  $X_{i,t} = \{x_{i,t}^1, x_{i,t}^2, \dots, x_{i,t}^D\}$  and  $V_{i,t} = \{v_{i,t}^1, v_{i,t}^2, \dots, v_{i,t}^D\}$  respectively. At each iteration the position and velocity of each particle is

updated taking into account the information of the best solution found by the swarm  $X_{gbest,t}$ , and the best solution found individually  $X_{pbest,t}$  by each particle. In the standard version of the PSO algorithm, the way in which the updating of the position and velocity of each particle is performed is carried out according to the following expressions [42]:

$$V_{i,t+1} = \omega \cdot V_{i,t} + c_1 \cdot r_1 \cdot (X_{gbest,t} - X_{i,t}) + c_2 \cdot r_2 \cdot (X_{pbest,t} - X_{i,t}), \quad (18)$$

$$X_{i,t+1} = X_{i,t} + V_{i,t+1}, \quad (19)$$

where  $c_1$  and  $c_2$  are the social and cognitive coefficients respectively, which commonly take the values  $c_1 = c_2 = 2$ .  $r_1$  and  $r_2$  are two randomly selected values within the interval  $[0, 1]$ .  $\omega$  is known as the inertia parameter and its aims to provide a better balance between global and local search. The inertia parameter  $\omega$  is updated at each iteration as follows:

$$\omega = \omega_{max} - \frac{t \cdot (\omega_{max} - \omega_{min})}{t_{max}}, \quad (20)$$

where  $t$  refers to the current iteration,  $t_{max}$  is the maximum number of iterations,  $t_{min}$  and  $t_{max}$  are the minimum and maximum values of the inertia factor, which are generally set to values  $t_{min} = 0.4$  and  $t_{max} = 0.9$ .

### 3.2. Differential Evolution

The DE algorithm proposed by Storn and Price [43,44], is one of the most representative evolutionary algorithms, which, due to its ease of implementation, effectiveness and robustness, has been widely used in the solution of several complex optimization problems [45,46]. This algorithm makes use of a population  $P$  constituted from  $NP$  individuals or parameter vectors:

$$P_t = \{X_{1,t}, X_{2,t}, \dots, X_{i,t}\}, \quad i = 1, \dots, NP, \quad (21)$$

where  $t$  refers to the current iteration, and each individual  $X_{i,t} = \{x_{i,t}^1, x_{i,t}^2, \dots, x_{i,t}^D\}$  has  $D$  dimensions. At the beginning of the execution of this algorithm, the population is initialized randomly within the search space of the problem.

The DE algorithm consists mainly of three operators: mutation, crossover and selection. These operators allow the algorithm to perform the search for the global optimum. Mutation, crossover and selection are applied consecutively to each individual  $X_{i,t}$  of the population  $P_t$  in each of the  $t$  iterations until a certain stopping criterion is satisfied. Within the mutation, in each iteration and for each individual, a mutated vector  $V_{i,t}$  is generated by means of the information of the current population  $P_t$  and the application of a mutation scheme. In the standard version of the DE algorithm the mutated vector is generated following the DE/rand/1 mutation scheme, described as follows:

$$V_{i,t} = X_{r1,t} + F \cdot (X_{r2,t} - X_{r3,t}), \quad (22)$$

where the indices  $r_1, r_2$  and  $r_3$  are randomly selected within the range  $[1, NP]$  such that they are different from each other and different from the index  $i$ .  $F > 0$  is the scaling factor and is one of the parameters of the algorithm. The value of parameter  $F$  is typically within the interval  $[0, 1]$ .

The next stage within the DE algorithm consists of the crossover, in which a test vector  $U_{i,t}$  is generated, that is, once the mutated vector  $V_{i,t}$  is generated for the individual  $X_{i,t}$ , the information crossover between the vector  $X_{i,t}$  and the vector  $V_{i,t}$  is performed. This crossover operation is performed consistently following the binomial crossover:

$$U_{i,t}^j = \begin{cases} V_{i,t}^j & \text{if } \text{rand}[0, 1] < Cr \text{ or } j = j_{\text{rand}} \\ X_{i,t}^j & \text{otherwise} \end{cases}, \quad (23)$$

where  $rand[0, 1]$  is a number uniformly selected within the interval  $[0, 1]$ , and  $j_{rand}$  is an index corresponding to a variable which is uniformly selected in the interval  $[1, D]$ . Here  $Cr \in [0, 1]$  is known as the crossover probability, and like  $F$ , it is another parameter of the algorithm.

Once the corresponding test vector  $U_{i,t}$  is generated for the individual  $X_{i,t}$ , a selection procedure is carried out from which the population for iteration  $t + 1$  is constructed. The standard way to perform this selection procedure consists of comparing the fit value of the test vector  $U_{i,t}$  against the fit value of the individual  $X_{i,t}$ , always keeping the individual with the best fit value. This procedure is performed as follows:

$$X_{i,t+1} = \begin{cases} U_{i,t} & \text{if } f(U_{i,t}) < f(X_{i,t}) \\ X_{i,t} & \text{otherwise} \end{cases} \quad (24)$$

### 3.2.1. Improvements on DE Parameters Control

One of the components that influence the performance of the DE algorithm is the control of the  $F$  and  $Cr$  parameters [25,47]. The standard version of the DE algorithm uses fixed values for these two parameters, however, the selection of these parameters can be done deterministically (following a certain rule that modifies these values after a certain number of iterations), adaptively (according to the feedback of the search process) and in a self-adaptive way (by means of the evolution information provided by the individuals of the population) [30]. One of the most representative variants concerning the improvement of the control of the parameters  $F$  and  $Cr$  of the DE algorithm was proposed in the SHADE algorithm [25].

Precisely, in each iteration of the SHADE algorithm, the parameters  $F$  and  $Cr$  associated with test vectors with a better fit than their parent vector are stored in two sets:  $S_F$  and  $S_{Cr}$  respectively. Two archives or memories  $M_F$  and  $M_{Cr}$  with a defined time  $H$ , and whose contents  $M_{Cr,k}$  and  $M_{F,k}$  ( $k = 1, \dots, H$ ) are initialized with a value of 0.5, serve to generate the parameters  $F_i$  and  $Cr_i$  of the  $i$ -th individual  $X_{i,t}$  at each iteration  $t$ . The way to generate the values of these parameters requires the uniform selection of an index  $r_i \in [1, H]$  and is carried out according to the following expressions:

$$Cr_i = \text{randn}_i(M_{Cr,r_i}, 0.1), \quad (25)$$

$$F_i = \text{randc}_i(M_{F,r_i}, 0.1), \quad (26)$$

where  $M_{Cr,r_i}$  is the element selected from  $M_{Cr}$  to generate the value of  $Cr_i$ , while  $M_{F,r_i}$  is the element selected from  $M_F$  to generate the value of  $F_i$ ;  $\text{randn}_i(M_{Cr,r_i}, 0.1)$  represents a normal distribution with mean  $M_{Cr,r_i}$  and standard deviation 0.1, while  $\text{randc}_i(M_{F,r_i}, 0.1)$  represents a Cauchy distribution with parameter position  $M_{F,r_i}$  and scale factor 0.1. When generating the value of the parameter  $Cr_i$ , it must be verified that it is within the range  $[0, 1]$ , otherwise, if  $Cr_i < 0$  then  $Cr_i$  is truncated to 0 and if  $Cr_i > 1$  then  $Cr_i$  is truncated to 1. For the parameter  $F_i$ , if  $F_i > 1$  then  $F_i$  is truncated to 1, otherwise, if  $F_i < 0$  then  $F_i$  is regenerated until  $F_i > 0$ .

At the end of each iteration, the contents of the  $M_{Cr}$  and  $M_F$  memories are updated as follows:

$$M_{Cr,k,t+1} = \begin{cases} \text{mean}_{WL}(S_{Cr}), & \text{si } S_{Cr} \neq \emptyset \\ M_{Cr,k,t}, & \text{otherwise} \end{cases} \quad (27)$$

$$M_{F,k,t+1} = \begin{cases} \text{mean}_{WL}(S_F), & \text{si } S_F \neq \emptyset \\ M_{F,k,t}, & \text{otherwise} \end{cases} \quad (28)$$

where  $\text{mean}_{WL}(S)$  is the Lehmer weighted mean defined by means of the Equation (29), and  $S$  refers to  $S_{Cr}$  or  $S_F$ .

$$mean_{WL} = \frac{\sum_{k=1}^{|S|} w_k \cdot S_k^2}{\sum_{k=1}^{|S|} w_k \cdot S_k} \quad (29)$$

$$w_k = \frac{\Delta f_k}{\sum_{l=1}^{|S|} \Delta f_l} \quad (30)$$

$$\Delta f_k = |f(X_{p,k,t}) - f(X_{k,t})| \quad (31)$$

In Equation (27) and Equation (28),  $k \in [1, H]$  determines the position in memory to be updated. At iteration  $t$ , the  $k$ -th element in memory is updated. At the beginning of the optimization process  $k = 1$ , and is incremented each time a new element is added to memory. If  $k > H$ ,  $k$  is assigned equal to 1.

Due to the good performance obtained by the SHADE algorithm when solving optimization problems using this strategy, this paper takes advantage of the adaptive control of the parameters provided by the SHADE algorithm.

### 3.2.2. DEPSO Algorithm

The self-adaptive differential evolution algorithm based on particle swarm optimization (DEPSO) [32] is a recent method, which incorporates characteristics of the PSO algorithm within the structure of the DE algorithm for solving numerical optimization problems. In this algorithm, the top  $\alpha \cdot NP$  ( $\alpha \in [0.1, 0.9]$ ) of the best particles in the current iteration is used to form an elite sub-swarm ( $P$ ), while the rest, is used to form a non-elite sub-swarm ( $Q$ ).

The DEPSO algorithm follows a scheme similar to the standard version of the DE algorithm, i.e., it uses mutation, crossover and selection operators within the evolutionary process of the algorithm. Within the mutation, a mutation strategy is implemented which, in a self-adaptive way, selects between two mutation schemes to generate in each iteration a mutated vector  $V_{i,t+1}$  for the  $i$ -th particle  $X_{i,t}$  as follows:

$$V_{i,t+1} = \begin{cases} X_{r1,t}^P + F_i \cdot (X_{r2,t}^P - X_{r3,t}^Q) & \text{if } rand[0,1] < SP_t \\ \omega \cdot X_{i,t} + c_1 \cdot r_1 \cdot (X_{gbest,t} - X_{i,t}) \\ + c_2 \cdot r_2 \cdot (X_{pbest,t} - X_{i,t}) & \text{otherwise} \end{cases} \quad (32)$$

where  $rand[0,1]$  is a random number selected uniformly within the interval  $[0,1]$ , and  $SP_t$  represents the probability of selecting between one of the two mutation schemes. In Equation (32), the case in which  $rand[0,1] < SP_t$  represents the form in which a novel mutation scheme denoted DE/e-rand/1 generates a mutated vector  $V_{i,t+1}$ . In this scheme two optimal solutions of the elite sub-swarm ( $X_{r1,t}^P$  and  $X_{r2,t}^P$ ) and one solution of the non-elite sub-swarm ( $X_{r3,t}^Q$ ) are required, and a scaling factor  $F_i$  is used, which acts at the level of each particle. The remaining case represents the way in which the mutation scheme of the standard PSO algorithm is used to generate a mutated vector  $V_{i,t+1}$ .

The selection probability  $SP_t$  of the Equation (32) changes adaptively within the evolution of the algorithm as follows:

$$SP_t = \frac{1}{1 + e^{1-(t_{max}/t+1)^\tau}}, \quad (33)$$

where  $t_{max}$  is the maximum number of iterations and  $t$  the current iteration.  $\tau$  is a positive constant.

Within the crossover operator, a test vector  $U_{i,t}$  is generated by combining the information of the current particle  $X_{i,t}$  and the mutated vector  $V_{i,t+1}$ , following the binomial crossover (see Equation (23)). The selection of the surviving particle to the next iteration is carried out by the competition between

the current particle  $X_{i,t}$  and the test vector  $U_{i,t}$ , the one with the best fit value according to the objective function is selected to remain within the next iteration.

When a particle remains stagnant over a maximum number of iterations (e.g. 5), the values of its corresponding scale factor  $F_i$  and crossover probability  $Cr_i$  are reset in order to increase diversity. The way in which the values of these parameters are reset is as follows:

$$F_{i,t+1} = \begin{cases} F_{i,t} & \text{if } NS_i < NS_{max} \\ F_l + rand[0,1] \cdot (F_u - F_l) & \text{otherwise} \end{cases}, \quad (34)$$

$$Cr_{i,t+1} = \begin{cases} Cr_{i,t} & \text{if } NS_i < NS_{max} \\ Cr_l + rand[0,1] \cdot (Cr_u - Cr_l) & \text{otherwise} \end{cases}, \quad (35)$$

where  $F_i$  and  $Cr_i$  are the scaling factor and the crossover probability, respectively, of the particle  $X_{i,t}$  at iteration  $t$ .  $F_l = 0.1$  and  $F_u = 0.8$  are the lower and upper bounds respectively of the scaling factor.  $Cr_l = 0.3$  and  $Cr_u = 1.0$  are the lower and upper bounds respectively of the crossover probability and  $rand[0,1]$  is a random number selected within the interval  $[0,1]$ .  $NS_i$  is a stagnation counter for each particle, and  $NS_{max}$  is the maximum number of iterations with stagnation.

To avoid stagnation, the DEPSO algorithm randomly updates a sub-dimension of individuals within the non-elite population ( $Q$ ) in which  $NS_i > NS_{max}$ , to reset them as follows:

$$X_{i,t}^j = \begin{cases} X_{min}^j + rand[0,1] \cdot (X_{max}^j - X_{min}^j) & \text{if } rand[0,1]_j \leq \gamma, \quad j = 1, 2, \dots, D \\ X_{i,t}^j & \text{otherwise} \end{cases} \quad (36)$$

where  $rand[0,1]_j$  is a random number selected within the interval  $[0,1]$ .  $\gamma$  is a probability fixed value.  $X_{min}^j$  and  $X_{max}^j$  are the lower and upper limits, respectively, of the variable  $j$ .

In general, the DEPSO algorithm manages to be superior to other variants of the DE algorithm in different optimization problems. The performance of this algorithm is due to the fact that it maintains a good balance between exploration and exploitation, thanks to the use of the self-adaptive mutation strategy, in which the "DE/e-rand/1" scheme has better exploration abilities, while the mutation scheme of the PSO algorithm achieves better convergence abilities. With this strategy, the population manages to maintain a good diversity in the first stages of the evolutionary process, and a faster convergence towards the last stages of the process.

#### 4. Proposed HE-DEPSO Algorithm

Despite the development of several advanced versions of DE algorithm in recent years, its performance still needs to improve in optimization problems with multiple local minima. To address these issues, the design of effective mutation operators and parameter control are two key aspects to improve the performance of the DE algorithm. In the proposed HE-DEPSO algorithm, an adaptive hybrid mutation operator is developed, which takes the self-adaptive mutation strategy from the DEPSO algorithm [32] as a basis and incorporates the parameter control mechanism from the SHADE algorithm [25]. This adaptive hybrid mutation operator introduces a new mutation strategy called "DE/current-to-EHE/1", which utilizes the historical information of the elite individuals in the population to enhance the optimization of the  $\chi^2$  function.

##### 4.1. Adaptive Hybrid Mutation Operator

In order to achieve a better balance between exploration and exploitation, the mutation operator of the HE-DEPSO algorithm adopts a dual mechanism in which it adaptively selects between two mutation strategies. First, a new mutation strategy called "DE/current-to-EHE/1" is presented, which is oriented to improve the balance between exploration and exploitation abilities within the early stages of the evolutionary process. On the other hand, to improve the exploitation capacity within the more advanced stages of the evolutionary process, the mutation scheme of the PSO algorithm

(Equation (32)) is incorporated in a similar way to the self-adaptive mutation strategy of the DEPSO algorithm (Equation (18)).

Unlike the “DE/e-rand/1” scheme used within the self-adaptive mutation strategy of the DEPSO algorithm, in which only the information of the individuals of the current iteration is used, the “DE/current-to-EHE/1” strategy uses the historical information of the elite and obsolete individuals of the evolutionary process to improve the exploration capability of the algorithm. This strategy is described below.

#### 4.1.1. DE/Current-to-EHE/1 Mutation Strategy

Within evolutionary algorithms, the best individuals in the population, also known as elite individuals, retain valuable evolutionary information to guide the population to promising regions [32,48–50]. However, many of the proposals made use of the information from the elite individuals of the current iteration, completely forgetting the information from the previous elite individuals. This absence of information within subsequent iterations may limit the ability of new individuals to explore the search space. Because of this, historical evolution information is used to improve the ability to explore new individuals.

Before applying the mutation operator, all individuals in the current population  $P_t$  are sorted in ascending order based on their fitness values. After reordering the population, two partitions of individuals are created. The first partition, denoted by  $E_t$ , consists of the top  $pb\% \in [0.1, 1]$  best individuals or elite individuals. The second partition, denoted by  $NE_t$  groups all non-elite individuals, i.e.,  $NP - NP \cdot pb\%$ . It is important to note that  $E_t \cup NE_t = P_t$ .

Unlike other mutation strategies [29,32,51], in which only elite individuals corresponding to the current iteration are used, this strategy makes use of the evolution history by incorporating two external archives of size  $NP$ . The first archive, denoted by  $HE$ , stores at each iteration the elite individuals belonging to the  $E_t$  partition. If the size of the archive  $HE$  is greater than  $NP$  then its size is readjusted. The second archive used, denoted by  $HL$ , stores the obsolete individuals (individuals discarded within the selection process) and is updated at each iteration. Similar to the  $HE$  archive, if the size of the  $HL$  file is larger than  $NP$  then its size is reset. With the information of the individuals belonging to  $E_t$  and  $HE$ , a group of candidates can be formed using  $E_t \cup HE$  to mutate individuals in the population. On the other hand, the individuals of  $NE_t$  and  $HL$  make up a group of individuals  $NE_t \cup HL$  whose information contributes to the exploration of the search space. The way in which a mutated vector is generated by the “DE/current-to-EHE/1” strategy is as follows:

$$V_{i,t} = X_{i,t} + F_i \cdot (X_{E_r,t} - X_{i,t}) + F_i \cdot (X_{P_r,t} - X_{HL_r,t}), \quad (37)$$

where  $X_{i,t}$  is the  $i$ -th individual at iteration  $t$ ,  $X_{E_r,t}$  is a randomly selected individual from among the individuals in the group  $E_t \cup HE$ ,  $X_{P_r,t}$  is an individual randomly selected from within the current population  $P_t$ ,  $X_{HL_r,t}$  is an individual randomly selected from the individuals comprising the group  $NE_t \cup HL$ , and  $F_i$  is the scaling factor corresponding to the  $i$ -th individual.

Following Equation (37), it is possible to observe that the “DE/current-to-EHE/1” strategy can help the HE-DEPSO algorithm to maintain a good exploration capability, and direct the mutated individuals to promising regions without leading to stagnation in local minima. This can be explained by the following reasons:

- The use of a randomly selected individual within the group  $E_t \cup HE$ , i.e.,  $X_{E_r,t}$ , helps to guide mutated individuals to more promising regions. However, due to the presence of the historical information of the elite individuals ( $HE$ ), the mutated individuals are prevented from being directed towards the best regions found, thus allowing the maintenance of good diversity in the population and increase the chances of targeting optimal regions.

- The participation of two individuals,  $X_{P_r,t}$  and  $X_{L_r,t}$ , randomly selected from  $P_t$  and **HL** respectively, promotes the diversity of mutated individuals. Consequently, the search diversity of the HE-DEPSO algorithm is considerably improved, which is beneficial for escaping from possible local minima.

At the beginning of the execution of the HE-DEPSO algorithm, the archives **HE** and **HL** are defined empty. Through the evolutionary process, they store the elite and obsolete individuals, respectively. When analyzing the effect of the parameter  $pb\%$ , which determines the percentage of individuals within the participation of  $E_t$ , it was found that values close to 0.9 promote the participation of a more significant number of elite individuals, causing a greater diversity in the population. However, this diversity increase may affect the HE-DEPSO algorithm's convergence capability. On the other hand, values close to 0.1 may restrict the number of elite individuals to be considered, which improves the convergence of the algorithm but may cause stagnation at local minima. Due to the above, within the HE-DEPSO algorithm, we choose to update in each iteration the value of  $pb\%$  following a linear decrement given as follows:

$$pb\% = p_{max} - \frac{t \cdot (p_{max} - p_{min})}{t_{max}}, \quad (38)$$

where  $t$  refers to the current iteration,  $t_{max}$  is the maximum number of iterations,  $p_{min}$  and  $p_{max}$  are the minimum and maximum values of the interval assigned for the percentage of individuals within the partition  $E_t$ . In this work, we have chosen to use the values  $p_{max} = 0.4$  and  $p_{min} = 0.1$ . In this way, the sensitivity of the  $pb\%$  parameter is reduced and, at the same time, a good balance between exploitation and exploration is maintained.

#### 4.1.2. Selection of Mutation Strategy for Adaptive Hybrid Mutation Operator

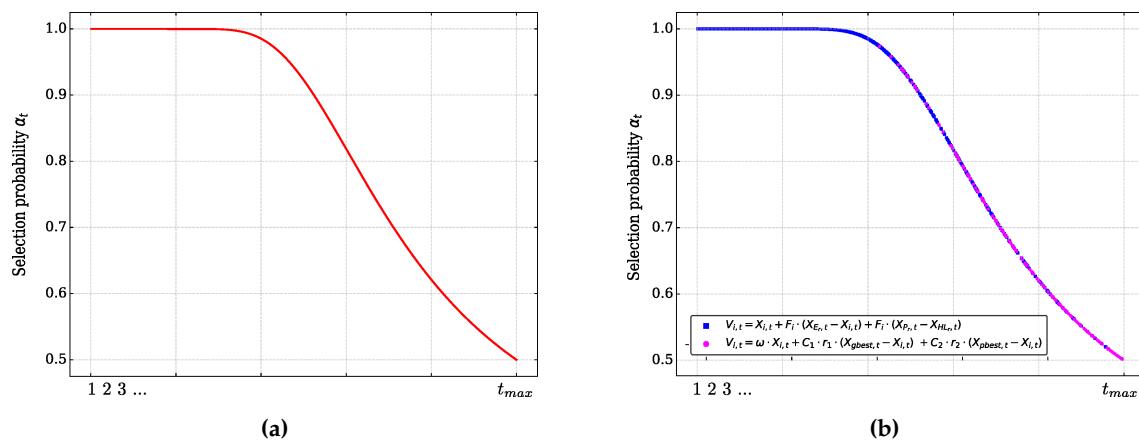
The adaptive hybrid mutation operator implemented in the HE-DEPSO algorithm uses two mutation strategies that aim to improve the exploration and exploitation abilities of the HE-DEPSO algorithm at different stages of the evolutionary process. Concretely, for each individual  $X_{i,t}$  in the population, this operator generates a mutated vector  $V_{i,t}$  as follows:

$$V_{i,t} = \begin{cases} X_{i,t} + F_i \cdot (X_{E_r,t} - X_{i,t}) + F_i \cdot (X_{P_r,t} - X_{HL_r,t}) & \text{if } rand[0,1] < \alpha_t \\ \omega \cdot X_{i,t} + c_1 \cdot r_1 \cdot (X_{gbest,t} - X_{i,t}) & \\ + c_2 \cdot r_2 \cdot (X_{pbest,t} - X_{i,t}) & \text{otherwise} \end{cases}, \quad (39)$$

where  $rand[0,1]$  is a random number selected uniformly within the interval  $[0,1]$ , and  $\alpha_t$  represents the probability of selecting between the “DE/current-to-EHE/1” mutation strategy and the adopted mutation strategy of the PSO algorithm (Equation (18)).

The probability of selection  $\alpha_t$ , is updated at each iteration according to the Equation (33) taking a value of  $\tau = 1.8$ . Figure 2 shows the mutation strategy selection curve followed by the adaptive hybrid mutation operator of the HE-DEPSO algorithm. The probability selection curve described by the Equation (33) with a value of  $\tau = 1.8$ , for a maximum number of iterations  $t_{max}$ , is illustrated in red in Figure 2a. On the other hand, Figure 2b shows, as an example, a possible distribution of occurrences among which is selected the mutation strategy “DE/current-to-EHE/1” (squares in blue color) or the adopted strategy of the PSO algorithm (circles in magenta color). According to these graphs, it can be observed that within the first stages of evolution of the HE-DEPSO algorithm, the probability of selection  $\alpha_t$  tends towards values close to 1, which increases the probability that the mutation strategy “DE/current-to-EHE/1” is selected, and the balance between exploration and exploitation provided by this strategy is taken advantage of. On the other hand, towards the advanced stages of the algorithm evolution, the mutation strategy adopted from the PSO algorithm is selected more frequently; this implies that, during the advanced stages, there is a higher probability that the HE-DEPSO algorithm

increases its exploitation ability due to more frequent use of the information of the best positions of each individual, as well as the information of the position of the best individual in the population.



**Figure 2.** Strategy selection curve; (a) Selection probability  $\alpha_t$ , (b) Distribution of strategy selection.

#### 4.2. The Complete HE-DEPSO Algorithm

The HE-DEPSO algorithm follows the usual structure of the DE algorithm in general, in which we have the stages of mutation, crossover, and selection. After generating a mutated vector  $V_{i,t}$  for the  $i$ -th individual of the population as described in the previous section, we proceed to generate a test vector  $U_{i,t}$  for the individual  $X_{i,t}$  using the binomial crossover (Equation (23)). In order to reduce the sensitivity of the  $F$  and  $C_r$  parameter control, the HE-DEPSO algorithm takes advantage of the individual-level adaptive parameter control of the SHADE algorithm to generate the parameter configuration of  $F$  and  $C_r$  for each individual. Finally, in the selection stage, the individuals that would make up the population for the next iteration are identified; this is done according to Equation (24). Based on the above description, the pseudo-code of the HE-DEPSO algorithm is reported in Figure (3).

**Algorithm: HE-DEPSO**


---

**Input** : Population size:  $NP$ , Maximum number of iterations:  $t_{max}$ , Dimensionality of the search space:  $D$

// Initialization

- 1 Iteration number:  $t = 0$
- 2 Randomly initialize the population:  $P_0 = \{X_{t,0}\}, i = 1, 2, \dots, NP$
- 3 Fitness evaluation of initial population:  $\{f_{i,0}\}$
- 4 Set all elements in  $M_F$  and  $M_{Cr}$  to 0.8 y 0.5 respectively
- 5 Set  $HE = \emptyset, HL = \emptyset$
- 6 Set  $p_{min} = 0.1, p_{max} = 0.25, k = 0, c1 = c2 = 2$
- 7 Initialize pbest and gbest
- // Main loop
- 8 **while**  $t < t_{max}$  **do**
- 9    $SCr = \emptyset; SF = \emptyset$
- 10   Update pb%,  $\alpha_t$  and  $\omega$
- 11   Sort individuals in  $P_t$
- 12   Generate  $E_t$  and  $NE_t$  from sets  $\{100pb\%\}$  and  $\{NP - (100pb\%\}$
- 13   Directly add  $E_t$  to  $HE$  (Readjust size of  $HE$  if necessary)
- 14   Randomly select a pair of values from  $M_F$  and  $M_{Cr}$  to generate  $F$  and  $Cr$  for all individuals
- 15   **for**  $i = 1 : NP$  **do**
- 16     Randomly select an index from  $[1, NP]$  to obtain the associated  $F_{i,t}$  and  $Cr_{i,t}$  of  $X_{i,t}$
- 17     Randomly select an individual  $X_{E_r,t}$  from  $E_t \cup HE$
- 18     Randomly select an individual  $X_{P_r,t}$  from  $P_t$
- 19     Randomly select an individual  $X_{HL_r,t}$  from  $NE_t \cup HL$
- 20     Generate a random number  $rand_i$  within  $[0, 1]$
- 21     Generate the mutation vector  $V_{i,t}$  by:
- 22     **if**  $rand_i < \alpha_t$  **then**
- 23        $V_{i,t} = X_{i,t} + F_i \cdot (X_{E_r,t} - X_{i,t}) + F_i \cdot (X_{P_r,t} - X_{HL_r,t})$
- 24     **else**
- 25        $V_{i,t} = \omega \cdot X_{i,t} + c1 \cdot r1 \cdot (X_{gbest,t} - X_{i,t}) + c2 \cdot r2 \cdot (X_{pbest,t} - X_{i,t})$
- 26     **end**
- 27     Generate the trial vector  $U_{i,t}$  using binomial crossover and calculate its fitness
- 28     **if**  $f(U_{i,t}) < f(X_{i,t})$  **then**
- 29        $X_{i,t+1} = U_{i,t}$
- 30        $F_{i,t} \rightarrow SF; Cr_{i,t} \rightarrow SCr$
- 31     **else**
- 32       Directly add  $X_{i,t}$  to  $HL$  (Readjust size of  $HL$  if necessary)
- 33        $X_{i,t+1} = X_{i,t}$
- 34     **end**
- 35     **if**  $f(U_{i,t}) < f(X_{pbest,i})$  **then**
- 36        $X_{pbest,i} = U_{i,t}$
- 37     **end**
- 38     **if**  $f(U_{i,t}) < f(X_{gbest,i})$  **then**
- 39        $X_{gbest,i} = U_{i,t}$
- 40     **end**
- 41   **end**
- 42   Update  $M_{Cr}$  and  $M_F$
- 43    $k = k + 1$  (if  $K > NP \rightarrow k = 1$ )
- 44    $t = t + 1$
- 45 **end**

**Output:**  $gbest$  and  $f(gbest)$

---

**Figure 3.** The pseudocode of HE-DEPSO**5. Experimental Results and Analysis**

This section verifies the performance of the proposed HE-DEPSO algorithm in solving different optimization problems. First, the set of test problems used is presented, then, the algorithms selected to carry out comparisons are presented, as well as the configuration of their parameters. Then, the performance of the HE-DEPSO algorithm is compared against the selected algorithms within the test problem set. Finally, a discussion of the results obtained is presented.

**5.1. Benchmark Functions**

In order to validate the performance of the HE-DEPSO algorithm in solving different optimization problems, the set of single-objective test functions with boundary constraints CEC 2017 [52] was used. This set consists of 29 test functions whose global optimum is known. We have two unimodal functions ( $F_1$  and  $F_3$ ), seven simple multimodal functions ( $F_4 - F_{10}$ ), ten hybrid functions ( $F_{11} - F_{20}$ ) and ten composite functions ( $F_{21} - F_{30}$ ). In all these optimization problems, we seek to find the global

minimum ( $F_{min}$ ) within the search space bounded in each dimension ( $D$ ) by the interval  $[-100, 100]$ . Information about this set of test functions is briefly presented in Table 2.

Table 2. Details of benchmark functions used in the experiments.

| Function Type               | Index | Function Name                                | Optimum ( $F_{min}$ ) |
|-----------------------------|-------|----------------------------------------------|-----------------------|
| Unimodal Functions          | 1     | Shifted and Rotated Bent Cigar               | 100                   |
|                             | 3     | Shifted and Rotated Zakharov                 | 300                   |
|                             | 4     | Shifted and Rotated Rosenbrock               | 400                   |
| Simple Multimodal Functions | 5     | Shifted and Rotated Rastrigin                | 500                   |
|                             | 6     | Shifted and Rotated Expanded Schaffer F6     | 600                   |
|                             | 7     | Shifted and Rotated Lunacek Bi-Rastrigin     | 700                   |
|                             | 8     | Shifted and Rotated Non-Continuous Rastrigin | 800                   |
|                             | 9     | Shifted and Rotated Levy                     | 900                   |
|                             | 10    | Shifted and Rotated Schwefel                 | 1000                  |
|                             | 11    | Hybrid Problem 1 (N = 3)                     | 1100                  |
| Hybrid Functions            | 12    | Hybrid Problem 2 (N = 3)                     | 1200                  |
|                             | 13    | Hybrid Problem 3 (N = 3)                     | 1300                  |
|                             | 14    | Hybrid Problem 4 (N = 4)                     | 1400                  |
|                             | 15    | Hybrid Problem 5 (N = 4)                     | 1500                  |
|                             | 16    | Hybrid Problem 6 (N = 4)                     | 1600                  |
|                             | 17    | Hybrid Problem 7 (N = 5)                     | 1700                  |
|                             | 18    | Hybrid Problem 8 (N = 5)                     | 1800                  |
|                             | 19    | Hybrid Problem 9 (N = 5)                     | 1900                  |
|                             | 20    | Hybrid Problem 10 (N = 6)                    | 2000                  |
| Composition Functions       | 21    | Composition Problem 1 (N = 3)                | 2100                  |
|                             | 22    | Composition Problem 2 (N = 3)                | 2200                  |
|                             | 23    | Composition Problem 3 (N = 4)                | 2300                  |
|                             | 24    | Composition Problem 4 (N = 4)                | 2400                  |
|                             | 25    | Composition Problem 5 (N = 5)                | 2500                  |
|                             | 26    | Composition Problem 6 (N = 5)                | 2600                  |
|                             | 27    | Composition Problem 7 (N = 6)                | 2700                  |
|                             | 28    | Composition Problem 8 (N = 6)                | 2800                  |
|                             | 29    | Composition Problem 9 (N = 9)                | 2900                  |
|                             | 30    | Composition Problem 10 (N = 3)               | 3000                  |

5.2. Algorithms and Parameter Settings

In this subsection, HE-DEPSO is compared with five algorithms, including the PSO and DE algorithms, as well as three representative and advanced variants of the DE algorithm, named CoDE [23], SHADE [25] and DEPSO [32]. Among these algorithms, CoDE and DEPSO incorporate modifications mainly within the mutation operator of the DE algorithm, while SHADE modifies the control of  $F$  and  $Cr$  parameters. The performance comparison of the HE-DEPSO algorithm with the previously mentioned algorithms was performed in two dimensions,  $D = 10$  and  $D = 30$ , on the CEC 2017 test function set.

In order to ensure a fair comparison, the parameter settings in common to all algorithms were assigned identically: the maximum number of iterations  $t_{max}$  was set to 1000, the population size  $NP$  to 100, and 31 independent runs were performed. The configuration of each algorithm’s parameters followed the values suggested by their respective authors, as presented in Table 3. The proposed HE-DEPSO algorithm was implemented in Python version 3.11, and all algorithms were executed on the same computer with 8 GB of RAM and a six-core 3.6 GHz processor.

Table 3. Parameters settings of all algorithms.

| Algorithm | Parameters                                                                                                                         |
|-----------|------------------------------------------------------------------------------------------------------------------------------------|
| HE-DEPSO  | $p_{min} = 0.1, p_{max} = 0.4, \tau = 1.8, c1 = c2 = 2, \omega \in [0.4, 0.9],$<br>$H = NP, M_F(1 : H) = 0.5, M_{Cr}(1 : H) = 0.8$ |
| DEPSO     | $c1 = c2 = 2, \omega \in [0.4, 0.9], Cr[0.3, 1.0], F[0.1, 0.8],$<br>$NS_{max} = 5, \gamma = 0.001, \tau = 1.8, SEP = 0.3 \cdot NP$ |
| SHADE     | $H = NP, M_F(1 : H) = M_{Cr}(1 : H) = 0.5$                                                                                         |
| CoDE      | 1) $[F = 1.0, Cr = 0.1], 2) [F = 1.0, Cr = 0.9], 3) [F = 0.8, Cr = 0.2]$                                                           |
| DE        | $Cr = 0.9, F = 0.5$                                                                                                                |
| PSO       | $\omega \in [0.4, 0.9], c1 = c2 = 2$                                                                                               |

5.3. Comparison with DE, PSO, and Three State-of-the-Art DE Variants

In order to evaluate the performance of the HE-DEPSO algorithm, its optimization results and convergence properties are compared and analyzed with respect to those obtained by the algorithms: DE, PSO, CoDE, SHADE and DEPSO, on the set of CEC 2017 test functions.

Tables 4 and 5 report the statistical results for each algorithm at  $D = 10$  and  $D = 30$  respectively. The error measure of the solution  $|F_{min} - F(X^*)|$ , where  $F_{min}$  is the known solution of the problem and  $X^*$  is the best solution found in each iteration of each algorithm, was used to obtain these results. These tables show the mean and standard deviation (Std) of the solution error measure for each function and algorithm over the 31 independent runs and the ranking achieved based on the mean value obtained. The best results are highlighted in bold.

A non-parametric Wilcoxon rank sum test, with a confidentiality level of  $\alpha = 0.05$ , was performed to identify statistically significant differences between the results obtained by the HE-DEPSO algorithm and the results obtained by the other algorithms. In these tables, the statistical significance related to the performance of the HE-DEPSO algorithm is represented by the symbols “+”, “≈” and “−”, which indicate that the performance of the HE-DEPSO algorithm is “better/similar/worse” than the algorithm to be compared. The row “W/T/L” counts the total number of “+”, “≈” and “−”, respectively.

Table 4. Comparison of results between HE-DEPSO, DE, PSO, and three advanced variants of the DE algorithm on the set of CEC 2017 test functions with  $D = 10$ .

| Function | Metrics      | HE-DEPSO          | DEPSO             | SHADE             | CoDE       | DE         | PSO        |
|----------|--------------|-------------------|-------------------|-------------------|------------|------------|------------|
| $F_1$    | Mean         | <b>0.0000E+00</b> | 1.8337E-15        | <b>0.0000E+00</b> | 1.1580E-10 | 3.5103E-05 | 1.1366E+10 |
|          | Std          | <b>0.0000E+00</b> | 4.8427E-15        | <b>0.0000E+00</b> | 6.0507E-11 | 2.0793E-05 | 9.7631E+09 |
|          | Rank         | <b>1</b>          | 4                 | <b>1</b>          | 3          | 2          | 5          |
|          | Significance |                   | ≈                 | ≈                 | +          | +          | +          |
| $F_3$    | Mean         | <b>9.1683E-16</b> | 1.3752E-14        | 1.6503E-14        | 2.0001E-09 | 1.7359E-04 | 1.5886E+10 |
|          | Std          | <b>0.0000E+00</b> | <b>0.0000E+00</b> | <b>0.0000E+00</b> | 2.7390E-11 | 4.8441E-04 | 6.2297E+02 |
|          | Rank         | <b>1</b>          | <b>1</b>          | <b>1</b>          | 2          | 3          | 4          |
|          | Significance |                   | ≈                 | ≈                 | +          | +          | +          |
| $F_4$    | Mean         | <b>0.0000E+00</b> | 2.3838E-14        | 2.0805E+00        | 2.3103E-09 | 1.0409E-03 | 5.8071E+01 |
|          | Std          | <b>0.0000E+00</b> | 2.8513E-14        | 6.7544E-01        | 1.2601E-09 | 6.4578E-04 | 6.9704E+01 |
|          | Rank         | <b>1</b>          | 2                 | 5                 | 3          | 4          | 6          |
|          | Significance |                   | +                 | +                 | +          | +          | +          |

Table 4. Cont.

| Function        | Metrics      | HE-DEPSO          | DEPSO             | SHADE             | CoDE              | DE                | PSO        |
|-----------------|--------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------|
| F <sub>5</sub>  | Mean         | <b>2.8439E+00</b> | 4.0119E+00        | 3.8489E+00        | 8.5594E+00        | 2.9710E+01        | 3.0728E+01 |
|                 | Std          | <b>7.2094E-01</b> | 1.6940E+00        | 1.0401E+00        | 1.4246E+00        | 3.9355E+00        | 1.2073E+01 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>6</sub>  | Mean         | <b>0.0000E+00</b> | 4.6321E-03        | 2.7530E-04        | 4.6895E-03        | 4.5748E-05        | 1.1970E+01 |
|                 | Std          | <b>0.0000E+00</b> | 2.5790E-02        | 5.5026E-04        | 2.4954E-03        | 2.2095E-05        | 9.6167E+00 |
|                 | Rank         | <b>1</b>          | 4                 | 3                 | 5                 | 2                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>7</sub>  | Mean         | <b>1.2678E+01</b> | 2.1067E+01        | 1.3953E+01        | 2.0678E+01        | 3.9143E+01        | 3.0657E+01 |
|                 | Std          | <b>7.0313E-01</b> | 6.7134E+00        | 7.6551E-01        | 2.2560E+00        | 3.7389E+00        | 8.6791E+00 |
|                 | Rank         | <b>1</b>          | 4                 | 2                 | 3                 | 6                 | 5          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>8</sub>  | Mean         | <b>2.2853E+00</b> | 4.2045E+00        | 3.8070E+00        | 8.6884E+00        | 2.6037E+01        | 2.4833E+01 |
|                 | Std          | 8.1114E-01        | 1.9346E+00        | <b>7.3957E-01</b> | 1.8765E+00        | 5.5290E+00        | 1.1493E+01 |
|                 | Rank         | 1                 | 3                 | 2                 | 4                 | 6                 | 5          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>9</sub>  | Mean         | <b>0.0000E+00</b> | 7.3346E-15        | <b>0.0000E+00</b> | <b>0.0000E+00</b> | <b>0.0000E+00</b> | 6.8233E+01 |
|                 | Std          | <b>0.0000E+00</b> | 2.8391E-14        | <b>0.0000E+00</b> | <b>0.0000E+00</b> | <b>0.0000E+00</b> | 1.9812E+02 |
|                 | Rank         | <b>1</b>          | 2                 | <b>1</b>          | <b>1</b>          | <b>1</b>          | 3          |
|                 | Significance |                   | ≈                 | ≈                 | ≈                 | +                 | +          |
| F <sub>10</sub> | Mean         | 2.4499E+02        | 8.9742E+02        | <b>1.5812E+02</b> | 7.4438E+02        | 1.4730E+03        | 9.1356E+02 |
|                 | Std          | 1.1864E+02        | 1.6890E+02        | <b>7.6565E+01</b> | 1.1745E+02        | 1.7509E+02        | 3.3829E+02 |
|                 | Rank         | 2                 | 4                 | <b>1</b>          | 3                 | 6                 | 5          |
|                 | Significance |                   | +                 | –                 | +                 | ≈                 | +          |
| F <sub>11</sub> | Mean         | <b>2.5687E-07</b> | 8.6658E-01        | 2.3208E+00        | 1.2502E+00        | 6.7909E+00        | 8.1368E+01 |
|                 | Std          | <b>1.4103E-06</b> | 6.1559E-01        | 4.7869E-01        | 4.6467E-01        | 9.9142E-01        | 8.3317E+01 |
|                 | Rank         | <b>1</b>          | 2                 | 4                 | 3                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>12</sub> | Mean         | 8.1966E+01        | <b>1.1535E+01</b> | 3.5298E+01        | 2.7652E+02        | 5.3458E+02        | 8.7444E+07 |
|                 | Std          | 7.0615E+01        | 2.8045E+01        | <b>1.4666E+01</b> | 6.4786E+01        | 1.2345E+02        | 3.8318E+08 |
|                 | Rank         | 3                 | <b>1</b>          | 2                 | 4                 | 5                 | 6          |
|                 | Significance |                   | –                 | –                 | +                 | +                 | +          |
| F <sub>13</sub> | Mean         | <b>3.8882E+00</b> | 4.6716E+00        | 5.9948E+00        | 1.1231E+01        | 1.2796E+01        | 6.3448E+03 |
|                 | Std          | 2.4042E+00        | 9.7586E-01        | <b>6.0682E-01</b> | 2.0901E+00        | 1.7030E+00        | 1.2009E+04 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | ≈                 | +                 | +                 | +                 | +          |
| F <sub>14</sub> | Mean         | <b>2.4091E+00</b> | 6.8868E+00        | 1.9065E+01        | 2.0295E+01        | 2.6185E+01        | 1.7551E+02 |
|                 | Std          | 5.1573E+00        | 8.0249E+00        | 3.0340E+00        | <b>1.2717E-01</b> | 1.2365E+00        | 9.5615E+01 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>15</sub> | Mean         | <b>3.6646E-01</b> | 5.7132E-01        | 1.0838E+00        | 1.3990E+00        | 2.1239E+00        | 4.6041E+02 |
|                 | Std          | <b>2.3145E-01</b> | 4.9456E-01        | 3.3723E-01        | 5.4455E-01        | 1.2934E+00        | 1.0508E+03 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>16</sub> | Mean         | <b>1.4450E+00</b> | 1.6357E+00        | 4.5365E+00        | 4.3009E+00        | 1.9028E+01        | 1.2496E+02 |
|                 | Std          | <b>3.2202E-01</b> | 3.2942E+00        | 2.2215E+00        | 2.5242E+00        | 1.6484E+01        | 1.0780E+02 |
|                 | Rank         | <b>1</b>          | 2                 | 4                 | 3                 | 5                 | 6          |
|                 | Significance |                   | –                 | +                 | +                 | +                 | +          |
| F <sub>17</sub> | Mean         | <b>7.2016E+00</b> | 1.2075E+01        | 2.1001E+01        | 2.0112E+01        | 2.8168E+01        | 5.7710E+01 |
|                 | Std          | 3.0221E+00        | 8.5311E+00        | 3.2211E+00        | 2.8326E+00        | <b>1.8397E+00</b> | 2.4017E+01 |
|                 | Rank         | <b>1</b>          | 2                 | 4                 | 3                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |

Table 4. Cont.

| Function        | Metrics      | HE-DEPSO          | DEPSO             | SHADE             | CoDE              | DE                | PSO        |
|-----------------|--------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------|
| F <sub>18</sub> | Mean         | <b>3.8770E-01</b> | 8.2949E+00        | 1.9823E+01        | 2.0117E+01        | 2.0542E+01        | 2.7241E+04 |
|                 | Std          | 1.4583E-01        | 6.7399E+00        | 1.9348E+00        | 3.4569E-02        | <b>1.0532E-01</b> | 1.8731E+04 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>19</sub> | Mean         | <b>9.7895E-02</b> | 5.4625E-01        | 1.0349E+00        | 5.9210E-01        | 1.2208E+00        | 5.0978E+03 |
|                 | Std          | <b>1.2871E-01</b> | 3.7564E-01        | 1.5715E-01        | 4.4716E-02        | 2.0310E-01        | 1.1584E+04 |
|                 | Rank         | <b>1</b>          | 2                 | 4                 | 3                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>20</sub> | Mean         | <b>5.2046E+00</b> | 6.9310E+00        | 1.7787E+01        | 2.0000E+01        | 2.5137E+01        | 5.9825E+01 |
|                 | Std          | 3.1153E+00        | 6.8469E+00        | 3.9314E+00        | <b>1.4932E-04</b> | 1.9501E+00        | 4.0129E+01 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | ≈                 | +                 | +                 | +                 | +          |
| F <sub>21</sub> | Mean         | 1.7676E+02        | <b>1.0695E+02</b> | 1.5575E+02        | 1.2554E+02        | 1.9598E+02        | 1.9045E+02 |
|                 | Std          | 4.6029E+01        | <b>2.0972E+01</b> | 4.2795E+01        | 4.8087E+01        | 5.7606E+01        | 6.2148E+01 |
|                 | Rank         | 4                 | <b>1</b>          | 3                 | 2                 | 6                 | 5          |
|                 | Significance |                   | —                 | —                 | —                 | +                 | +          |
| F <sub>22</sub> | Mean         | 1.0000E+02        | 9.7428E+01        | 9.9647E+01        | <b>7.4280E+01</b> | 1.0070E+02        | 1.2711E+02 |
|                 | Std          | <b>0.0000E+00</b> | 1.5939E+01        | 2.1997E+00        | 4.4524E+01        | 5.7442E-01        | 5.6792E+01 |
|                 | Rank         | 4                 | 2                 | 3                 | <b>1</b>          | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | ≈                 | ≈                 | +          |
| F <sub>23</sub> | Mean         | <b>3.0613E+02</b> | 3.0845E+02        | 3.0859E+02        | 3.2072E+02        | 3.3600E+02        | 3.4479E+02 |
|                 | Std          | 1.4783E+00        | 2.1842E+00        | <b>1.4096E+00</b> | 3.4098E+00        | 5.2784E+00        | 1.6592E+01 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |
| F <sub>24</sub> | Mean         | 3.2397E+02        | <b>2.6066E+02</b> | 3.0017E+02        | 3.0685E+02        | 3.6119E+02        | 3.7743E+02 |
|                 | Std          | 4.1615E+01        | 1.1813E+02        | 6.5109E+01        | 9.2286E+01        | <b>4.7582E+00</b> | 1.5025E+01 |
|                 | Rank         | 4                 | <b>1</b>          | 2                 | 3                 | 5                 | 6          |
|                 | Significance |                   | ≈                 | —                 | ≈                 | +                 | +          |
| F <sub>25</sub> | Mean         | 4.1869E+02        | 4.2783E+02        | 4.0980E+02        | <b>3.9783E+02</b> | 4.0247E+02        | 4.7998E+02 |
|                 | Std          | 2.3095E+01        | 2.2521E+01        | 2.0463E+01        | <b>1.5694E-01</b> | 1.3981E+01        | 6.1710E+01 |
|                 | Rank         | 4                 | 5                 | 3                 | <b>1</b>          | 2                 | 6          |
|                 | Significance |                   | ≈                 | —                 | —                 | +                 | +          |
| F <sub>26</sub> | Mean         | 3.0304E+02        | 3.0849E+02        | 3.0085E+02        | <b>2.9625E+02</b> | 3.8419E+02        | 5.6150E+02 |
|                 | Std          | <b>1.1775E+01</b> | 3.2854E+01        | 1.7405E+01        | 6.3643E+01        | 1.6020E+02        | 2.5242E+02 |
|                 | Rank         | 3                 | 4                 | 2                 | <b>1</b>          | 5                 | 6          |
|                 | Significance |                   | +                 | ≈                 | +                 | ≈                 | +          |
| F <sub>27</sub> | Mean         | 3.7584E+02        | <b>3.7153E+02</b> | 3.7683E+02        | 4.1575E+02        | 3.9004E+02        | 4.1060E+02 |
|                 | Std          | 2.3098E+01        | <b>5.3891E-01</b> | 7.7935E+00        | 1.8031E+01        | 3.4010E+01        | 2.0006E+01 |
|                 | Rank         | 2                 | <b>1</b>          | 3                 | 5                 | 4                 | 6          |
|                 | Significance |                   | ≈                 | +                 | +                 | +                 | +          |
| F <sub>28</sub> | Mean         | 4.7251E+02        | <b>4.7218E+02</b> | 4.7338E+02        | 4.7816E+02        | 4.7253E+02        | 6.1933E+02 |
|                 | Std          | <b>3.0463E-02</b> | 1.8112E+00        | 2.7252E+00        | 6.1120E+00        | 1.6134E-01        | 1.5566E+02 |
|                 | Rank         | 2                 | <b>1</b>          | 4                 | 5                 | 3                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | ≈                 | +          |
| F <sub>29</sub> | Mean         | 2.3616E+02        | <b>2.3508E+02</b> | 2.4658E+02        | 2.5569E+02        | 2.8284E+02        | 3.4235E+02 |
|                 | Std          | 6.1991E+00        | 8.2538E+00        | <b>6.1477E+00</b> | 9.0010E+00        | 1.4685E+01        | 9.4838E+01 |
|                 | Rank         | 2                 | <b>1</b>          | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | ≈                 | +                 | +                 | +                 | +          |
| F <sub>30</sub> | Mean         | <b>2.0061E+02</b> | 2.0113E+02        | 2.0227E+02        | 2.0313E+02        | 2.0320E+02        | 1.6044E+06 |
|                 | Std          | <b>1.5718E-01</b> | 5.2187E-01        | 3.8772E-01        | 4.7938E-01        | 8.5716E-01        | 2.1981E+06 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4                 | 5                 | 6          |
|                 | Significance |                   | +                 | +                 | +                 | +                 | +          |

Table 4. Cont.

| Function     | Metrics | HE-DEPSO | DEPSO  | SHADE  | CoDE   | DE     | PSO    |
|--------------|---------|----------|--------|--------|--------|--------|--------|
| Average rank |         | 1.68     | 2.27   | 2.75   | 3.24   | 4.48   | 5.48   |
| Final rank   |         | 1        | 2      | 3      | 4      | 5      | 6      |
| W/T/L        |         | -/-/-    | 17/9/3 | 20/4/5 | 24/3/2 | 25/4/0 | 29/0/0 |

Table 5. Comparison of results between HE-DEPSO, DE, PSO, and three advanced variants of the DE algorithm on the CEC 2017 test function set with  $D = 30$ .

| Function | Metrics      | HE-DEPSO   | DEPSO      | SHADE      | CoDE       | DE         | PSO        |
|----------|--------------|------------|------------|------------|------------|------------|------------|
| $F_1$    | Mean         | 1.6045E-14 | 2.0034E-04 | 2.3598E+03 | 1.2860E+08 | 2.0817E+09 | 6.3600E+10 |
|          | Std          | 6.0758E-15 | 4.9086E-04 | 1.2985E+03 | 3.5599E+07 | 7.3074E+08 | 5.8100E+10 |
|          | Rank         | 1          | 2          | 3          | 4          | 5          | 6          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_3$    | Mean         | 1.4486E-13 | 1.3343E+02 | 6.1189E+04 | 7.0539E+04 | 4.1473E+05 | 2.1200E+04 |
|          | Std          | 4.1092E-14 | 8.8712E+01 | 2.2749E+04 | 2.6622E+04 | 1.3053E+05 | 2.9800E+04 |
|          | Rank         | 1          | 2          | 4          | 5          | 6          | 3          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_4$    | Mean         | 2.8405E+00 | 1.2605E+01 | 7.7040E+01 | 9.6061E+01 | 3.5736E+01 | 1.7300E+03 |
|          | Std          | 2.1842E+00 | 8.6176E+00 | 3.8595E+00 | 2.0503E+01 | 3.2288E+00 | 1.3000E+03 |
|          | Rank         | 1          | 2          | 4          | 5          | 3          | 6          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_5$    | Mean         | 2.3850E+01 | 1.8175E+02 | 9.3463E+01 | 1.8418E+02 | 2.4169E+02 | 1.8500E+02 |
|          | Std          | 5.0701E+00 | 1.2845E+01 | 1.0871E+01 | 1.2630E+01 | 1.2302E+01 | 4.4900E+01 |
|          | Rank         | 1          | 3          | 2          | 4          | 6          | 5          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_6$    | Mean         | 4.6793E-01 | 5.7971E-03 | 3.7760E+01 | 5.8751E+01 | 2.4651E+01 | 5.3800E+01 |
|          | Std          | 3.5101E-01 | 2.9636E-02 | 5.8394E+00 | 5.2672E+00 | 3.3305E+00 | 1.2000E+01 |
|          | Rank         | 2          | 1          | 4          | 6          | 3          | 5          |
|          | Significance |            | —          | +          | +          | +          | +          |
| $F_7$    | Mean         | 5.2487E+01 | 2.3377E+02 | 1.1601E+02 | 2.1572E+02 | 2.6887E+02 | 2.9100E+02 |
|          | Std          | 3.3716E+00 | 1.4172E+01 | 8.0551E+00 | 1.2272E+01 | 1.3070E+01 | 1.1300E+02 |
|          | Rank         | 1          | 4          | 2          | 3          | 5          | 6          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_8$    | Mean         | 2.0726E+01 | 1.7779E+02 | 9.3771E+01 | 1.8161E+02 | 2.3844E+02 | 1.7100E+02 |
|          | Std          | 4.6475E+00 | 1.1810E+01 | 9.9764E+00 | 1.3211E+01 | 1.6657E+01 | 3.8700E+01 |
|          | Rank         | 1          | 4          | 2          | 5          | 6          | 3          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_9$    | Mean         | 1.1236E+00 | 2.8880E-03 | 7.6008E+02 | 4.1017E+03 | 4.8783E+02 | 2.8400E+03 |
|          | Std          | 9.0322E-01 | 1.6080E-02 | 2.7907E+02 | 8.3155E+02 | 1.5998E+02 | 1.2400E+03 |
|          | Rank         | 2          | 1          | 4          | 6          | 3          | 5          |
|          | Significance |            | —          | +          | +          | +          | +          |
| $F_{10}$ | Mean         | 2.6526E+03 | 6.7636E+03 | 3.4600E+03 | 6.3166E+03 | 7.9631E+03 | 5.2400E+03 |
|          | Std          | 3.0794E+02 | 2.5866E+02 | 2.3363E+02 | 2.0917E+02 | 3.0010E+02 | 8.1500E+02 |
|          | Rank         | 1          | 5          | 2          | 4          | 6          | 3          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_{11}$ | Mean         | 3.2932E+01 | 7.9240E+01 | 7.3961E+01 | 2.7578E+02 | 2.1665E+02 | 9.3600E+02 |
|          | Std          | 1.7540E+01 | 9.4859E+00 | 1.6453E+01 | 6.0425E+01 | 4.1872E+01 | 1.7000E+03 |
|          | Rank         | 1          | 3          | 2          | 5          | 4          | 6          |
|          | Significance |            | +          | +          | +          | +          | +          |
| $F_{12}$ | Mean         | 6.0704E+03 | 7.4374E+04 | 1.7001E+06 | 5.9649E+08 | 2.5500E+09 | 1.1600E+10 |
|          | Std          | 5.2411E+03 | 7.3297E+04 | 5.7892E+05 | 1.8156E+08 | 8.9672E+08 | 1.5200E+10 |
|          | Rank         | 1          | 2          | 3          | 4          | 5          | 6          |

Table 5. Cont.

| Function        | Metrics      | HE-DEPSO          | DEPSO             | SHADE             | CoDE       | DE         | PSO        |
|-----------------|--------------|-------------------|-------------------|-------------------|------------|------------|------------|
| F <sub>13</sub> | Significance |                   | +                 | +                 | +          | +          | +          |
|                 | Mean         | <b>5.3737E+01</b> | 2.3200E+03        | 7.2862E+03        | 3.3010E+08 | 1.0201E+08 | 1.0300E+10 |
|                 | Std          | <b>4.9535E+01</b> | 5.3868E+03        | 2.4057E+04        | 8.7976E+07 | 3.9539E+07 | 1.5000E+10 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 5          | 4          | 6          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>14</sub> | Mean         | <b>5.1892E+01</b> | 9.7833E+01        | 7.5446E+01        | 1.5942E+04 | 4.0951E+03 | 2.2100E+05 |
|                 | Std          | 1.3639E+01        | 2.2150E+01        | <b>1.0230E+01</b> | 4.5198E+03 | 8.1599E+02 | 4.9000E+05 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 5          | 4          | 6          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>15</sub> | Mean         | <b>6.7166E+01</b> | 7.9018E+01        | 1.0487E+02        | 3.0409E+07 | 3.1028E+06 | 2.9100E+08 |
|                 | Std          | 4.6957E+01        | 3.4973E+01        | <b>2.4704E+01</b> | 9.9730E+06 | 1.2198E+06 | 1.6200E+09 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 5          | 4          | 6          |
|                 | Significance |                   | ≈                 | ≈                 | +          | +          | +          |
| F <sub>16</sub> | Mean         | <b>2.7544E+02</b> | 1.4323E+03        | 8.5936E+02        | 1.6794E+03 | 2.3141E+03 | 1.7100E+03 |
|                 | Std          | 1.4178E+02        | 1.6353E+02        | <b>1.1941E+02</b> | 1.7833E+02 | 1.8579E+02 | 6.0700E+02 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 4          | 6          | 5          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>17</sub> | Mean         | <b>6.6918E+01</b> | 3.8170E+02        | 1.9226E+02        | 6.8169E+02 | 1.1646E+03 | 8.4800E+02 |
|                 | Std          | <b>1.3274E+01</b> | 1.0402E+02        | 6.2540E+01        | 9.4653E+01 | 1.4710E+02 | 3.2100E+02 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 4          | 6          | 5          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>18</sub> | Mean         | <b>6.1722E+01</b> | 2.4655E+02        | 2.3829E+05        | 3.4473E+05 | 7.9235E+05 | 1.9200E+06 |
|                 | Std          | <b>2.7440E+01</b> | 2.4587E+02        | 1.4060E+05        | 1.1610E+05 | 1.9796E+05 | 4.8700E+06 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4          | 5          | 6          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>19</sub> | Mean         | 4.0473E+01        | 3.7316E+01        | <b>3.2834E+01</b> | 5.0014E+03 | 3.7873E+04 | 4.2500E+08 |
|                 | Std          | 2.2587E+01        | 7.4498E+00        | <b>6.1811E+00</b> | 2.4562E+03 | 2.1473E+04 | 7.1600E+08 |
|                 | Rank         | 3                 | 2                 | <b>1</b>          | 4          | 5          | 6          |
|                 | Significance |                   | ≈                 | ≈                 | +          | +          | +          |
| F <sub>20</sub> | Mean         | <b>8.7890E+01</b> | 2.6052E+02        | 3.3149E+02        | 4.8347E+02 | 1.0591E+03 | 6.0400E+02 |
|                 | Std          | <b>3.2291E+01</b> | 9.0720E+01        | 7.0732E+01        | 7.4079E+01 | 1.1750E+02 | 2.3100E+02 |
|                 | Rank         | <b>1</b>          | 2                 | 3                 | 4          | 6          | 5          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>21</sub> | Mean         | <b>2.2047E+02</b> | 3.7820E+02        | 2.7893E+02        | 3.9662E+02 | 4.4351E+02 | 3.8400E+02 |
|                 | Std          | <b>4.7309E+00</b> | 1.3317E+01        | 3.7140E+01        | 7.1300E+00 | 1.3511E+01 | 5.0800E+01 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 5          | 6          | 4          |
|                 | Significance |                   | ≈                 | ≈                 | ≈          | +          | ≈          |
| F <sub>22</sub> | Mean         | <b>1.0186E+02</b> | 2.9760E+03        | 2.2144E+02        | 6.5363E+03 | 8.0766E+03 | 4.5500E+03 |
|                 | Std          | <b>2.0196E+00</b> | 3.4464E+03        | 2.9380E+01        | 5.3749E+02 | 2.9236E+02 | 2.0400E+03 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 5          | 6          | 4          |
|                 | Significance |                   | +                 | ≈                 | +          | +          | +          |
| F <sub>23</sub> | Mean         | <b>3.7543E+02</b> | 5.4924E+02        | 4.4917E+02        | 5.5419E+02 | 5.8732E+02 | 7.2800E+02 |
|                 | Std          | <b>8.3139E+00</b> | 1.3235E+01        | 9.7459E+00        | 1.2816E+01 | 1.9032E+01 | 7.9500E+01 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 4          | 5          | 6          |
|                 | Significance |                   | +                 | +                 | +          | +          | +          |
| F <sub>24</sub> | Mean         | <b>4.4893E+02</b> | 6.2854E+02        | 5.3810E+02        | 6.9034E+02 | 6.7462E+02 | 8.2800E+02 |
|                 | Std          | <b>6.6585E+00</b> | 1.3484E+01        | 1.6953E+01        | 1.5747E+01 | 1.2401E+01 | 1.0000E+02 |
|                 | Rank         | <b>1</b>          | 3                 | 2                 | 5          | 4          | 6          |
|                 | Significance |                   | ≈                 | ≈                 | +          | ≈          | +          |
| F <sub>25</sub> | Mean         | 3.8051E+02        | <b>3.7837E+02</b> | 3.7882E+02        | 4.1044E+02 | 3.9396E+02 | 5.6400E+02 |
|                 | Std          | 1.1308E+01        | <b>6.0195E-02</b> | 1.1887E+00        | 6.6423E+00 | 4.4183E+00 | 2.0000E+02 |
|                 | Rank         | 3                 | <b>1</b>          | 2                 | 5          | 4          | 6          |

Table 5. Cont.

| Function     | Metrics      | HE-DEPSO          | DEPSO             | SHADE             | CoDE              | DE                | PSO        |
|--------------|--------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------|
| $F_{26}$     | Significance |                   | —                 | ≈                 | +                 | ≈                 | +          |
|              | Mean         | <b>1.3469E+03</b> | 2.6899E+03        | 1.4149E+03        | 2.6587E+03        | 3.0652E+03        | 4.9700E+03 |
|              | Std          | 9.6715E+01        | 2.2460E+02        | 5.7298E+02        | <b>9.0222E+01</b> | 1.2139E+02        | 9.0600E+02 |
|              | Rank         | <b>1</b>          | 4                 | 2                 | 3                 | 5                 | 6          |
|              | Significance |                   | +                 | ≈                 | +                 | +                 | +          |
| $F_{27}$     | Mean         | <b>5.0001E+02</b> | <b>5.0001E+02</b> | <b>5.0001E+02</b> | <b>5.0001E+02</b> | <b>5.0001E+02</b> | 7.1600E+02 |
|              | Std          | 1.5786E-04        | <b>1.2917E-04</b> | 9.3150E-05        | 5.9917E-05        | 8.1534E-05        | 1.0700E+02 |
|              | Rank         | <b>1</b>          | <b>1</b>          | <b>1</b>          | <b>1</b>          | <b>1</b>          | 2          |
|              | Significance |                   | ≈                 | ≈                 | ≈                 | ≈                 | ≈          |
| $F_{28}$     | Mean         | 4.9902E+02        | <b>4.9059E+02</b> | 4.9845E+02        | 5.0001E+02        | 5.0001E+02        | 1.2300E+03 |
|              | Std          | 3.0735E+00        | 4.2116E+00        | 1.3995E+00        | 6.6494E-05        | <b>6.6473E-05</b> | 8.4400E+02 |
|              | Rank         | 3                 | <b>1</b>          | 2                 | 4                 | 4                 | 5          |
|              | Significance |                   | —                 | ≈                 | ≈                 | ≈                 | +          |
| $F_{29}$     | Mean         | <b>3.8546E+02</b> | 9.3919E+02        | 4.6431E+02        | 1.6187E+03        | 2.0724E+03        | 1.9200E+03 |
|              | Std          | 4.5529E+01        | 2.1194E+02        | <b>4.2813E+01</b> | 1.8521E+02        | 1.8256E+02        | 6.1300E+02 |
|              | Rank         | <b>1</b>          | 3                 | 2                 | 4                 | 6                 | 5          |
|              | Significance |                   | +                 | +                 | +                 | +                 | +          |
| $F_{30}$     | Mean         | 2.8659E+02        | <b>2.3933E+02</b> | 1.7439E+04        | 1.4650E+07        | 2.4426E+06        | 4.7800E+07 |
|              | Std          | 5.8034E+01        | <b>6.7166E+00</b> | 2.1431E+04        | 6.8820E+06        | 1.5971E+06        | 8.9800E+07 |
|              | Rank         | 2                 | <b>1</b>          | 3                 | 5                 | 4                 | 6          |
|              | Significance |                   | —                 | +                 | +                 | +                 | +          |
| Average rank |              | <b>1.31</b>       | 2.44              | 2.44              | 4.37              | 4.75              | 5.13       |
| Final rank   |              | <b>1</b>          | 2                 | 2                 | 3                 | 4                 | 5          |
| W/T/L        |              | -/-/-             | 19/5/5            | 20/9/0            | 26/3/0            | 25/4/0            | 27/2/0     |

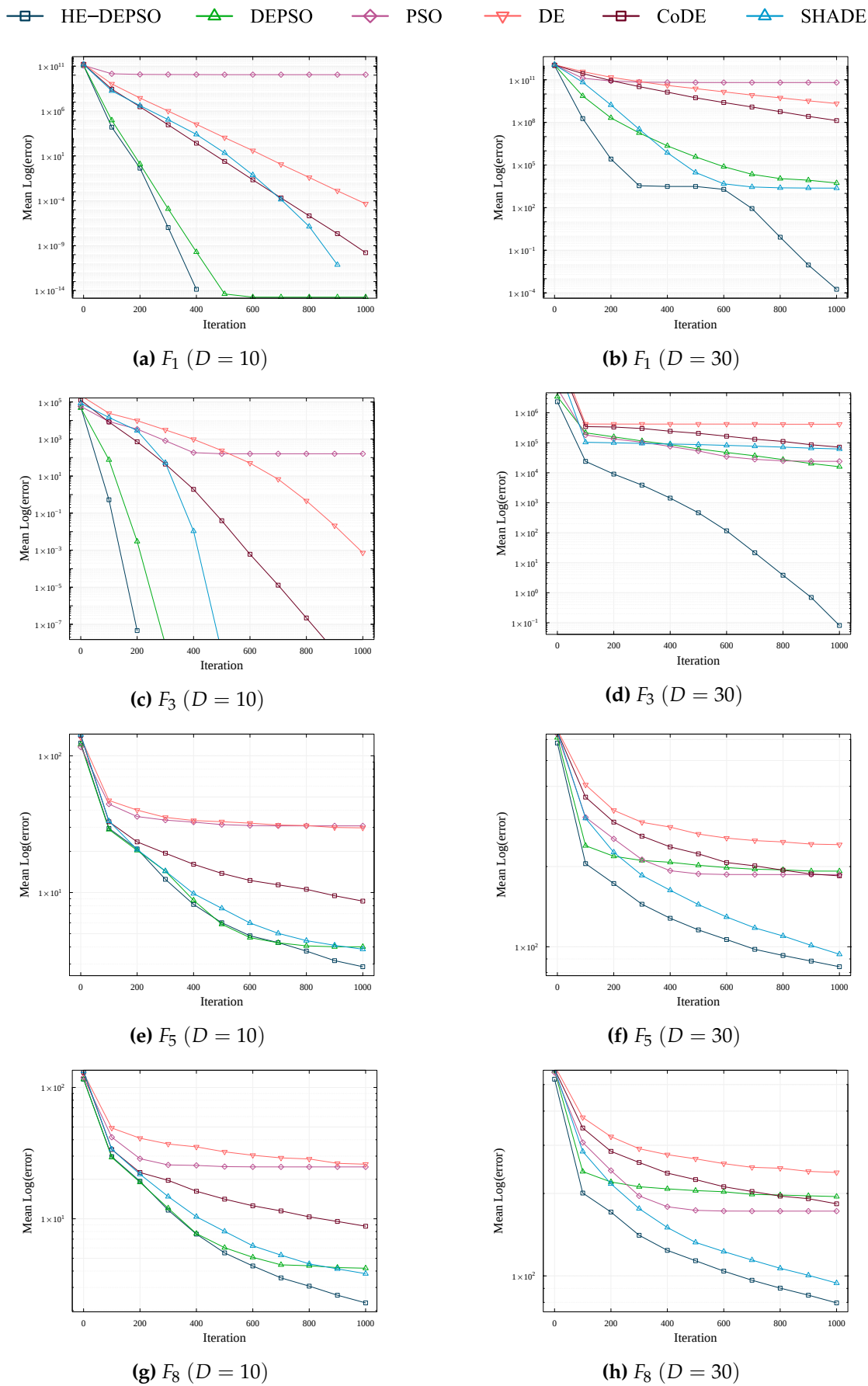
5.3.1. Optimization Results

According to the results reported in Table 4, for  $D = 10$ , the proposed HE-DEPSO algorithm obtains the global optimal solution for functions  $F_1 - F_4$ ,  $F_6$  and  $F_9$ . For the unimodal function  $F_3$ , DEPSO and SHADE obtain the global optimal solution. On the other hand, SHADE, CoDE, and DE obtain the global optimal solution in the simple multimodal function  $F_9$ . For functions  $F_5$ ,  $F_7 - F_9$ ,  $F_{11}$ ,  $F_{13} - F_{20}$ ,  $F_{23}$  and  $F_{30}$  the HE-DEPSO algorithm achieves the best result among all algorithms. DEPSO is the best in the functions  $F_{12}$ ,  $F_{21}$ ,  $F_{24}$  and  $F_{27} - F_{29}$ . SHADE obtains the best results on function  $F_{10}$ , while CoDE obtains the best results on functions  $F_{22}$ ,  $F_{25}$  and  $F_{26}$ . The PSO does not show superior results in any function. The results of this table indicate that, for these tests, the proposed HE-DEPSO algorithm obtains the best ranking according to the mean value among all the algorithms. On the other hand, the results of the Wilcoxon rank sum test show that HE-DEPSO is superior to DEPSO, SHADE, CoDE, DE and PSO in 17, 20, 24, 25, and 29 functions out of a total of 29 test functions respectively.

For  $D = 30$ , the statistical results presented in Table 5 show that the HE-DEPSO algorithm achieves the best results on the functions  $F_1 - F_5$ ,  $F_7 - F_8$ ,  $F_{10} - F_{18}$ ,  $F_{20} - F_{24}$ ,  $F_{26} - F_{27}$  and  $F_{29}$ . On the other hand, DEPSO, SHADE, CoDE, and DE achieve the best results in the  $F_{27}$  function. DEPSO is the best in the functions  $F_6$ ,  $F_9$ ,  $F_{25}$ ,  $F_{28}$  and  $F_{30}$ , while SHADE, shows to be the best in the function  $F_{19}$  in comparison to the rest of the algorithms. With these results, it is possible to identify that HE-DEPSO obtains the best ranking among all the algorithms for these tests. According to the Wilcoxon rank sum test results, among the 29 test functions, HE-DEPSO has 19, 20, 26, 26, 25, and 27 items that are better than DEPSO, SHADE, CoDE, DE, and PSO, respectively.

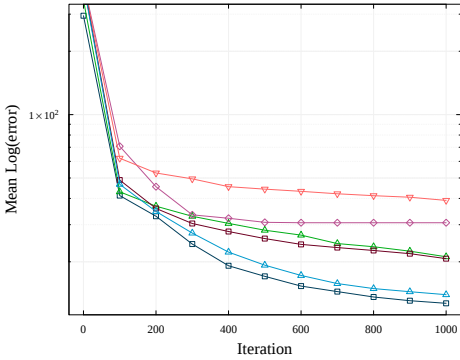
5.3.2. Convergence Properties

The convergence properties can be summarized into four types, which are represented by the graphs presented in Figures 4 and 5.

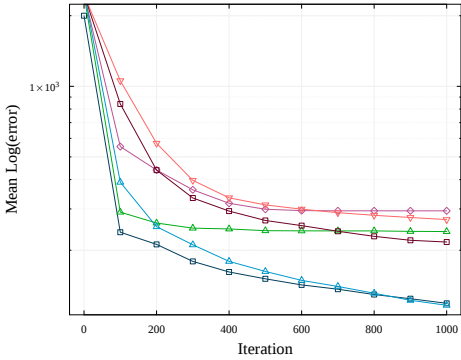


**Figure 4.** Convergence curves for the functions  $F_1$ ,  $F_3$ ,  $F_5$  and  $F_8$ , with  $D = 10, 30$ . The horizontal and vertical axis represent the iterations and the mean error values for the 31 independent repetitions.

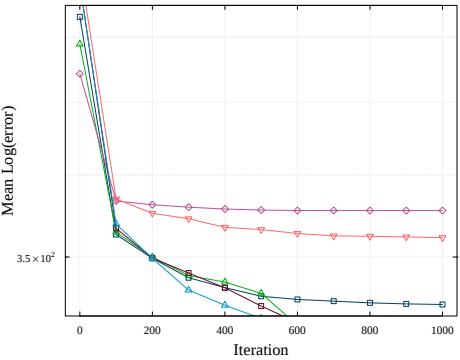
HE-DEPSO    DEPSO    PSO    DE    CoDE    SHADE



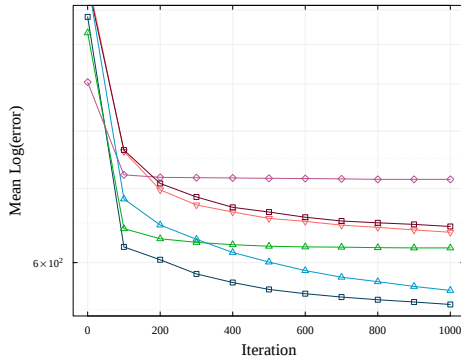
(a)  $F_7$  ( $D = 10$ )



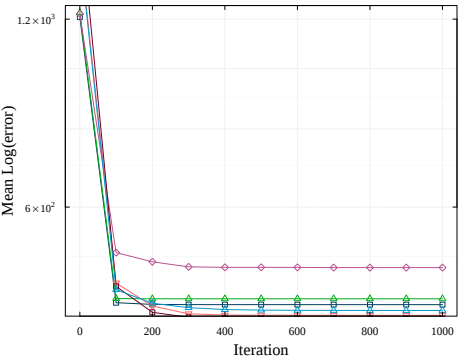
(b)  $F_7$  ( $D = 30$ )



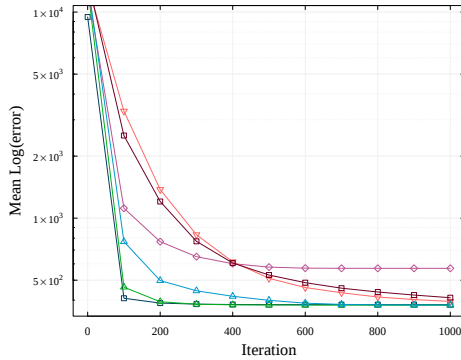
(c)  $F_{24}$  ( $D = 10$ )



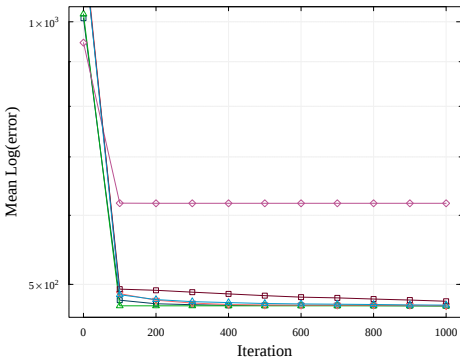
(d)  $F_{24}$  ( $D = 30$ )



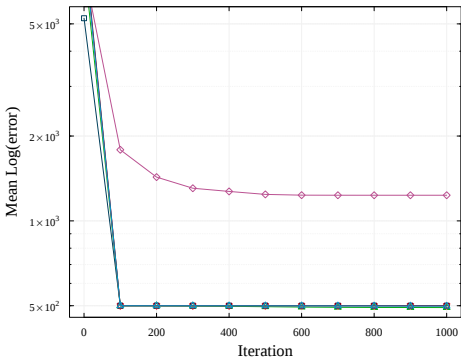
(e)  $F_{25}$  ( $D = 10$ )



(f)  $F_{25}$  ( $D = 30$ )



(g)  $F_{28}$  ( $D = 10$ )



(h)  $F_{28}$  ( $D = 30$ )

**Figure 5.** Convergence curves for the functions  $F_1, F_3, F_5$  and  $F_8$ , with  $D = 10, 30$ . The horizontal and vertical axis represent the iterations and the mean error values for the 31 independent repetitions.

1. The convergence properties of the functions  $F_1 - F_4$ ,  $F_6$  and  $F_9$  can be identified within the same class; this can be observed with the help of Figure 4(a)–4(d). In this type, the HE-DEPSO algorithm shows faster convergence than the other algorithms at  $D = 10$ , while at  $D = 30$ , it obtains a better mean error value.
2. In Figure 4(e)–4(h), the convergence curves of functions  $F_5$  and  $F_8$  are presented, which are similar to the corresponding ones for functions  $F_{10}$ ,  $F_{16}$ ,  $F_{17}$  and  $F_{20}$ . All algorithms have different degrees of evolutionary stagnation or exhibit slow evolution for these functions.
3. The convergence curves of the functions  $F_{12} - F_{15}$ ,  $F_{18} - F_{19}$ ,  $F_{23}$  and  $F_{29} - F_{30}$ , are similar to the curves of the functions  $F_7$  and  $F_{24}$  (see Figure 5(a)–4(d)) obtained for the HE-DEPSO algorithm. In this type, HE-DEPSO presents a fast convergence at the beginning, and subsequently, it keeps evolving downward.
4. The evolutionary process presented in the convergence curves of the functions  $F_{22}$  and  $F_{27}$  follows the same trend as the convergence curves of the functions  $F_{25}$  and  $F_{28}$  as can be seen in Figure 5(e)–4(h). Here, most of the algorithms present stagnation in an accelerated way, although some of them manage to keep evolving downward.

The experiments performed above prove the superiority of the proposed HE-DEPSO algorithm in solving different optimization problems. The reasons why the HE-DEPSO algorithm obtains such superior performance can be summarized as follows:

1. The adaptive hybrid mutation operator implemented in HE-DEPSO is based on the self-adaptive mutation strategy of the DEPSO algorithm, which has proven to be helpful in tackling several complex optimization problems. On the other hand, the collaborative work between the “DE/current-to-EHE/1” mutation strategy with historical information of the elite individuals and the PSO algorithm strategy generates a good balance between exploration and exploitation at different stages of the evolutionary process.
2. The self-adaptive control of  $F$  and  $Cr$  parameters adopted from the SHADE algorithm allows mitigating parameter sensitivity. In this way, the crossover probability and the scale factor are dynamically adjusted during the evolutionary process at the level of each individual, which can make the proposed algorithm suitable for a wider variety of optimization problems.

## 6. 4-Zeros Texture Model Validation

In order to verify the physical feasibility of the 4-zero texture model, it is essential to find sets of numerical values for the parameters  $A_u$ ,  $A_d$ ,  $\phi_1$  and  $\phi_2$  that minimize the  $\chi^2(X)$  function to a value less than one (see Equation (16)), and also that, the numerical value for all  $|V_{ckm}|$  predicted for the 4-zeros texture model (evaluated in these sets of values) are in agreement with the experimental data. Firstly, an analysis of the HE-DEPSO algorithm's behavior when optimizing the  $\chi^2(X)$  function will be conducted. The study will assess the algorithm's optimization, convergence, and stability properties, providing insights into its performance when dealing with the  $\chi^2$  optimization problem.

### 6.1. HE-DEPSO Performance in $\chi^2$ Optimization

The given optimization problem seeks to minimize the function  $\chi^2(X)$  given by the Equation (17). The search space is constrained by the possible signs of the parameters  $\eta_u$  and  $\eta_d$ , as shown in Table 1. The phases  $\phi_1$  and  $\phi_2$  can take values between 0 and  $2\pi$ . The ranges of parameters  $A_u$  and  $A_d$  depend on the signs of  $\eta_u$  and  $\eta_d$ . Specifically considering the case of study 1,  $A_u$  can take values in the interval  $[m_c, m_t]$ , while  $A_d$  in  $[m_s, m_b]$ .

#### 6.1.1. Experiment Configuration

Through an experiment, an evaluation of the performance of the HE-DEPSO algorithm in the optimization of the  $\chi^2(X)$  function will be carried out in order to determine its ability to find the global minimum. This study will be carried out in the search space defined in case 1 previously mentioned. The performance of HE-DEPSO will be compared with the PSO and DE algorithms, as

well as with advanced variants of the DE algorithm such as CoDE, SHADE, and DEPSO. To ensure a fair comparison, standard parameters have been set for all algorithms:  $t_{max} = 1000$ ,  $NP = 100$ , and 31 independent runs have been performed. The settings of the other parameters of each algorithm have been made following the indications in Table 3.

6.1.2. Experiment Results

To evaluate the effectiveness of the proposed HE-DEPSO algorithm, comparisons and analysis of its optimization results were performed with the DE, PSO, and three advanced variants of DE algorithms applied to the  $\chi^2(X)$  function. Using an approach similar to that described in Section 5, Table 6 presents the statistical data for each algorithm evaluated.

**Table 6.** Comparison of results between HE-DEPSO, DE, PSO and three advanced variants of the DE algorithm, on the function  $\chi^2(X)$ .

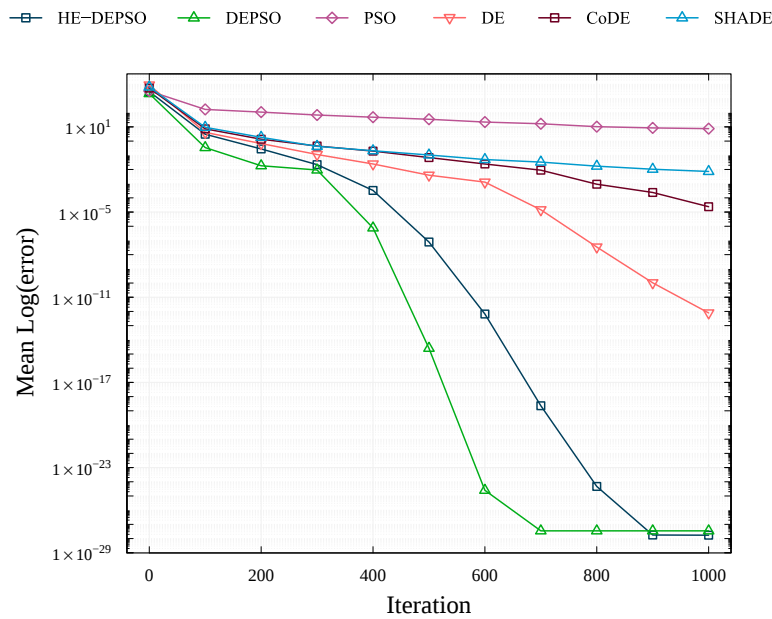
| Function             | Metrics        | HE-DEPSO          | DEPSO      | SHADE      | CoDE       | DE         | PSO        |
|----------------------|----------------|-------------------|------------|------------|------------|------------|------------|
| $\chi^2(X)$          | Mean           | <b>1.7267E-28</b> | 3.4971E-28 | 7.2296E-03 | 2.3720E-05 | 6.1610E-13 | 7.4247E+00 |
|                      | Std            | <b>1.4496E-28</b> | 5.6400E-28 | 8.7102E-03 | 7.7748E-05 | 3.4294E-12 | 1.6815E+01 |
|                      | Classification | <b>1</b>          | 2          | 5          | 4          | 3          | 6          |
|                      | Significance   |                   | +          | +          | +          | +          | +          |
| Final classification |                | <b>1</b>          | 2          | 5          | 4          | 3          | 6          |
| W/T/L                |                | -/-/-             | 1/0/0      | 1/0/0      | 1/0/0      | 1/0/0      | 1/0/0      |

According to the data presented in Table 6, the HE-DEPSO algorithm excels in obtaining the best results in optimizing the  $\chi^2(X)$  function, outperforming DEPSO. Although DEPSO comes close to the results of HE-DEPSO, its accuracy is not matched. On the other hand, the DE algorithm also demonstrated acceptable accuracy, although lower than that achieved by HE-DEPSO and DEPSO. In contrast, the SHADE, CoDE, and PSO algorithms obtained the lowest results compared to the others. In terms of average ranking, HE-DEPSO is positioned as the leader among all the analyzed algorithms. Furthermore, the results of the Wilcoxon test confirm the superiority of HE-DEPSO over DEPSO, SHADE, CoDE, DE, and PSO in the optimization of the  $\chi^2(X)$  function.

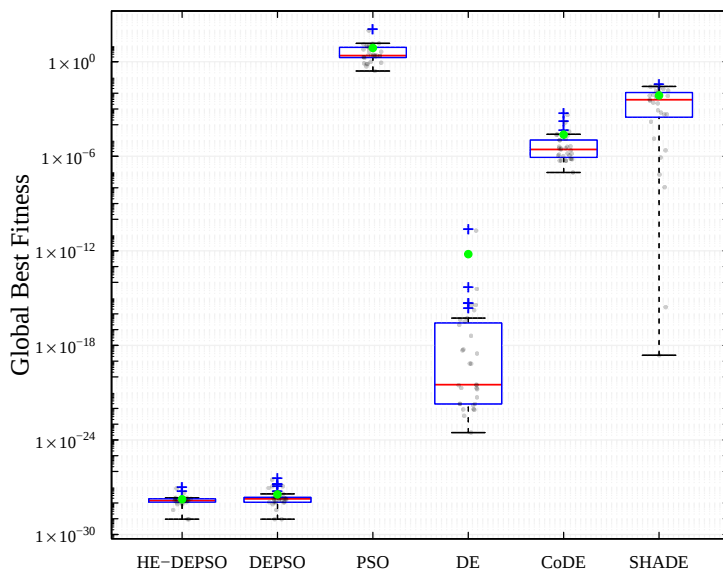
Figure 6 presents the curves obtained over 31 independent runs by the HE-DEPSO, DEPSO, SHADE, CoDE, DE, and PSO algorithms. It is observed that the HE-DEPSO algorithm exhibits excellent convergence behavior, consistently outperforming SHADE, CoDE, DE, and PSO and achieving a lower mean error value. Although DEPSO converges faster starting from the 300 iteration, the performance of HE-DEPSO stands out for its consistency and exploration capability, which could explain its more extended convergence compared to DEPSO.

In Figure 7, box and whisker plots show the results of the HE-DEPSO, DEPSO, SHADE, CoDE, DE, and PSO algorithms, based on the best overall fits of 31 independent runs to the chi-square function, are presented. These graphs illustrate the distribution of these values into quartiles, with the median highlighted by a red horizontal line within a blue box. The box boundaries correspond to the upper (Q3) and lower (Q1) quartiles. Lines extending from the box indicate the maximum and minimum values, excluding outliers represented by the “+” symbol in blue. The green circles represent the mean of the overall best fits of the 31 independent runs.

According to the box and whisker plots, the HE-DEPSO algorithm demonstrated superior performance to the other evaluated algorithms regarding stability and solution quality, based on 31 independent runs. The distribution of the best global fitness values achieved by HE-DEPSO presents a higher concentration around an lower optimum, reflected in a lower average than the other algorithms. Consequently, the HE-DEPSO offers a more robust and stable performance.



**Figure 6.** Convergence curves of the error measure in the solution for the  $\chi^2(X)$  function. The horizontal and vertical axis represent the iterations and the mean error values for the 31 independent iterations.



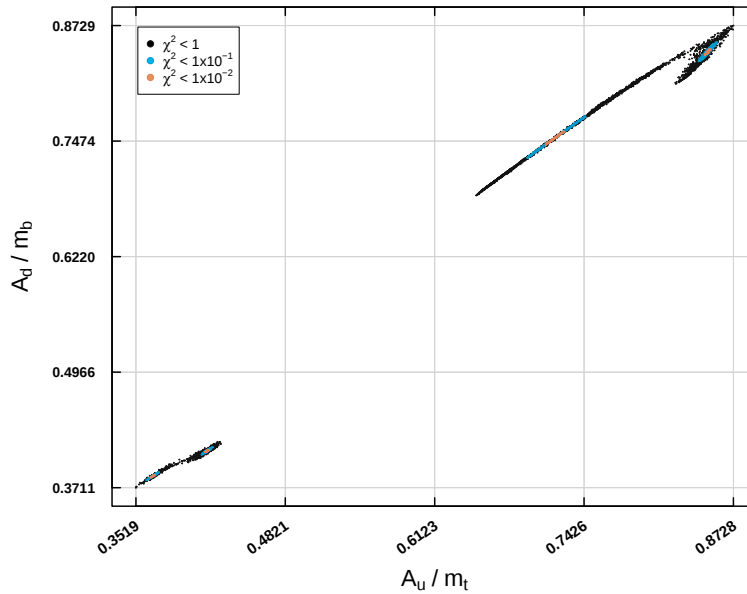
**Figure 7.** Box and whisker plots of the best global fit values obtained by HE-DEPSO and the DEPSO, SHADE, CoDE, DE, and PSO algorithms in the 31 independent repetitions in the  $\chi^2(X)$  function. The horizontal axis shows the algorithms to be compared, and the vertical axis shows the global fit values.

## 6.2. Valid Regions for Parameters $A_u, A_d, \phi_1$ and $\phi_2$

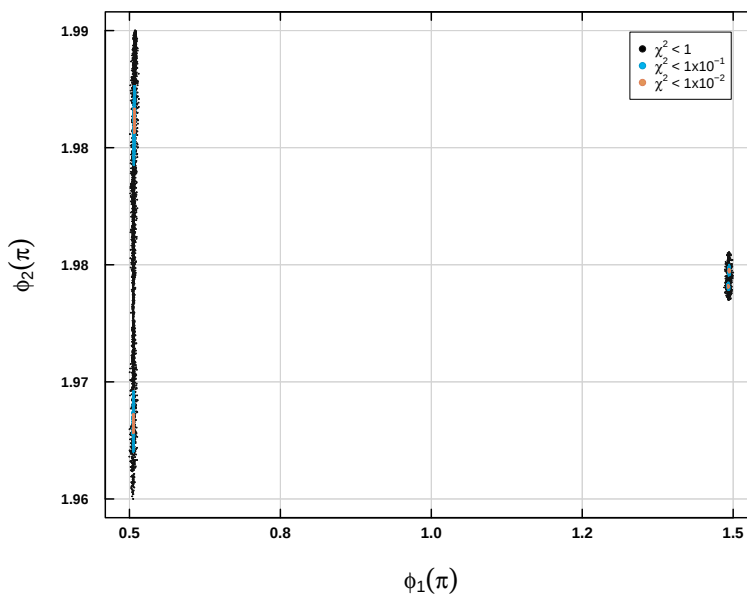
In this subsection, we present the results obtained after applying the HE-DEPSO algorithm to search for valid regions of the  $A_u, A_d, \phi_1$  and  $\phi_2$  parameters. This process was carried out using the most recent experimental values of the  $V_{CKM}$  matrix elements, the quark masses, and the Jarlskog invariant (see Appendix A).

Figures 8 and 9 present the allowed regions for the parameters  $A_u, A_d, \phi_1$  and  $\phi_2$ , scaled as  $A_u/m_t, A_d/m_b, \phi_1/\pi$  and  $\phi_2/\pi$  respectively, in the first case study of Table 1. These regions are classified by the solutions found using the HE-DEPSO algorithm and are represented by black, blue, and orange dots, which correspond to different levels of precision of the  $\chi^2(X)$  function. The orange

region, the smallest, indicates the region of highest precision, where experimental data agree best with theoretical predictions. These solutions were found by iteratively executing the algorithm until a total of 5000 solutions were collected at each of the precisions ( $\chi^2 < 1$ ,  $\chi^2 < 1 \times 10^{-1}$  and  $\chi^2 < 1 \times 10^{-2}$ ).



**Figure 8.** Allowed regions for  $A_u$  and  $A_d$  constrained based on current experimental data with different levels of precision:  $\chi^2 < 1$  (black dots),  $\chi^2 < 1 \times 10^{-1}$  (blue dots) and  $\chi^2 < 1 \times 10^{-2}$  (orange dots).



**Figure 9.** Allowed regions for  $\phi_1$  and  $\phi_2$  constrained based on current experimental data with different levels of precision:  $\chi^2 < 1$  (black dots),  $\chi^2 < 1 \times 10^{-1}$  (blue dots) and  $\chi^2 < 1 \times 10^{-2}$  (orange dots).

It is worth noting that for the rest of the case studies (see Table 1), the found regions for the parameters  $A_u$ ,  $A_d$ ,  $\phi_1$  and  $\phi_2$  were similar.

### 6.3. Predictions for the $V_{CKM}$ Elements

The second part of model validation is the following: with the data obtained from the valid regions for the four free parameters of the 4-zeros texture model:  $A_u$ ,  $A_d$ ,  $\phi_1$  and  $\phi_2$ , found with the help of HE-DEPSO, we can evaluate the feasibility of the model according to the latest experimental values of the  $V_{CKM}$  matrix and the Jarlskog invariant. This can be achieved by checking the model's predictive

power when predicting the remaining elements of the  $V_{CKM}$  matrix that were not used within the  $\chi^2$  fit, i.e.,  $|V_{cd}|$ ,  $|V_{ud}|$ ,  $|V_{cs}|$ ,  $|V_{tb}|$ ,  $|V_{td}|$ , and  $|V_{ts}|$ . Using Equations (2), (3), (4), and (5), the Chau-Keng parametrization [37], and the solutions found, we can calculate the numerical predictions for those elements. These predictions are then presented using scatter plots. Each graph includes the experimental central value of each element with its associated experimental error value. The points that fall within the region delimited by the experimental values are considered good predictions, while those outside the region are considered poor predictions.

In Figure 10, the predictions for the elements  $|V_{ud}|$  and  $|V_{cd}|$  in the first case study are presented. The black, blue, and orange points indicate a precision of  $\chi^2 < 1$ ,  $\chi^2 < 1 \times 10^{-1}$ , and  $\chi^2 < 1 \times 10^{-2}$ , respectively. For  $|V_{cd}|$ , the experimental central value (0.22636) is indicated with a solid red vertical line on the corresponding axis, while its experimental error ( $\pm 0.00048$ ) is shown with two red dashed vertical lines on either side of the central value. In the case of  $|V_{ud}|$ , the experimental central value (0.97401) is represented with a solid blue horizontal line on the  $|V_{ud}|$  axis, and its experimental error ( $\pm 0.00011$ ) is indicated with two blue dashed horizontal lines on either side of the central value. The predictions for the two elements show that the data with a precision of  $\chi^2 < 1$  can be outside the experimental error region, suggesting a poor prediction. On the other hand, the predictions with  $\chi^2 < 1 \times 10^{-1}$  and  $\chi^2 < 1 \times 10^{-2}$  remain within this region, indicating accurate predictions. This pattern is repeated in the rest of the analyzed case studies.

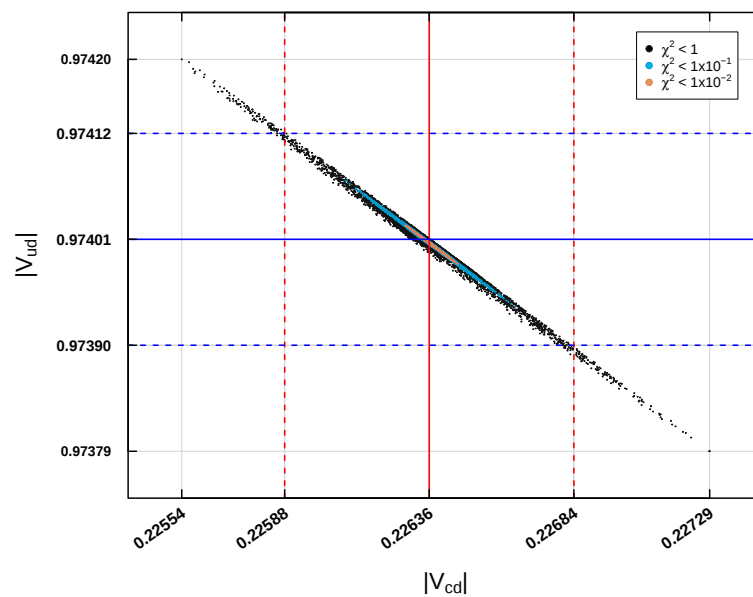
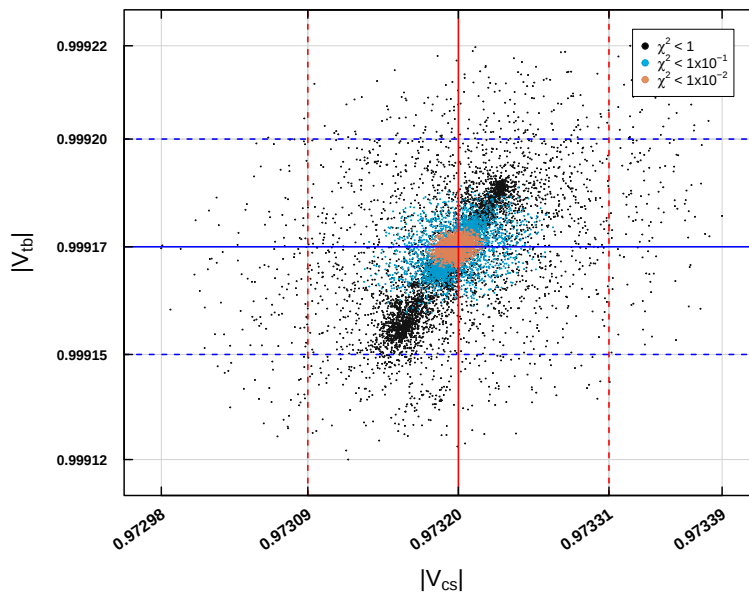


Figure 10. Predictions for  $|V_{cd}|$  and  $|V_{ud}|$  for  $\eta_u = +1$  and  $\eta_d = +1$ .

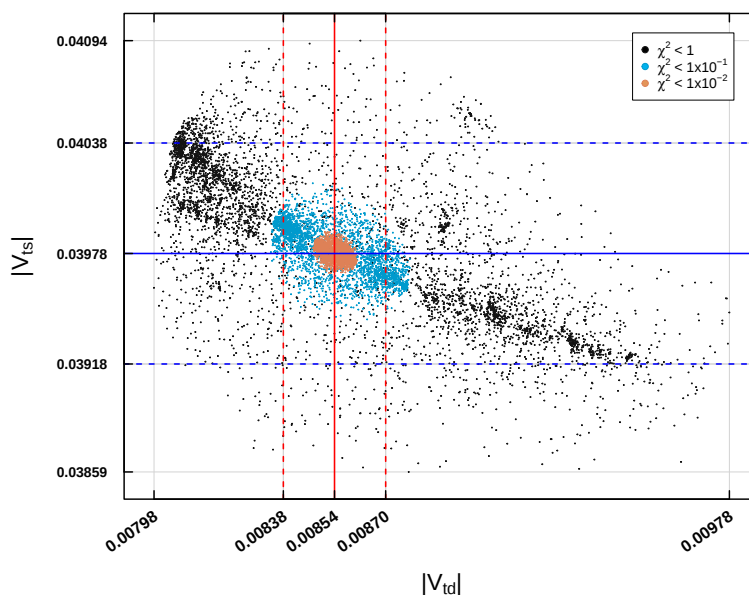
The predictions for the elements  $|V_{cs}|$  and  $|V_{tb}|$  are shown in Figure 11 for the first case study. The experimental central value of  $|V_{cs}|$  is represented by a solid red vertical line at the center of the  $|V_{cs}|$  axis at 0.97320, while its experimental error of  $\pm 0.00011$  is indicated by red dashed vertical lines on either side. The experimental central value of  $|V_{tb}|$  is shown by a solid blue horizontal line at the center of the  $|V_{tb}|$  axis at 0.999172, and its experimental error of  $\pm 0.000024$  is represented by blue dashed horizontal lines on either side. According to the predictions shown in Figure 11, the data obtained with  $\chi^2 < 1$  do not represent good predictions, as they are outside the experimental errors. On the other hand, the predictions with  $\chi^2 < 1 \times 10^{-1}$  and  $\chi^2 < 1 \times 10^{-2}$  remain within the error region, indicating that they are good predictions. This pattern is repeated in the four case studies.

Finally, in Figure 12, the predictions for the elements  $|V_{td}|$  and  $|V_{ts}|$  are shown in the four case studies. For  $|V_{td}|$ , the experimental central value is represented by a solid vertical line at  $|V_{td}| = 0.00854$ , and its experimental error with two red dashed vertical lines at  $\pm 0.00016$ . For  $|V_{ts}|$ , the experimental central value is shown with a solid blue horizontal line at  $|V_{ts}| = 0.03978$ , and its experimental error



**Figure 11.** Predictions for  $|V_{cs}|$  and  $|V_{tb}|$  for  $\eta_u = +1$  and  $\eta_d = +1$ .

with two blue dashed horizontal lines at  $\pm 0.00060$ . The predictions for these two elements indicate that the data obtained with precisions of  $\chi^2 < 1$  and  $\chi^2 < 1 \times 10^{-1}$  may be outside the experimental errors, so they are not good predictions. In contrast, the prediction with  $\chi^2 < 1 \times 10^{-2}$  remains within the errors, being a good prediction. These characteristics are observed in the four case studies.



**Figure 12.** Predictions for  $|V_{td}|$  and  $|V_{ts}|$  for  $\eta_u = +1$  and  $\eta_d = +1$ .

When analyzing the predictions of the elements of the  $V_{CKM}$  matrix ( $|V_{ud}|$ ,  $|V_{cd}|$ ,  $|V_{cs}|$ ,  $|V_{tb}|$ ,  $|V_{td}|$  and  $|V_{ts}|$ ), it was observed that only the solutions of the HE-DEPSO algorithm with an accuracy of  $\chi^2 < 1 \times 10^{-2}$  fit within the limits of experimental error. These solutions are considered valid and capable of reproducing the  $V_{CKM}$  matrix in all case studies and elements. Thus, it can be established that the 4-zero texture model is physically viable. It would be interesting, however, to explore the implications of this information obtained within the texture formalism.

## 7. Conclusions

This paper explored the feasibility of the 4-zeros texture model using the most recent experimental values. To gather data that can inform the decision on the model's viability, we employ a Chi-square fit to compare the theoretical expressions of the model with experimental values of well-measured physical observables. We developed a single-objective optimization model by defining a  $\chi^2(X)$  function to identify the allowed regions for the free parameters of the 4-zeros texture model that are consistent with the experimental data. To address the challenge of optimizing the  $\chi^2(X)$  function within the Chi-square fit, we propose a new DEPSO algorithm variant, HE-DEPSO.

The proposed algorithm has demonstrated its ability to efficiently optimize different functions, particularly those from the CEC 2017 single-objective benchmark functions set. The convergence properties of HE-DEPSO show a good balance between solution precision and convergence speed in most cases. Regarding optimizing the  $\chi^2(X)$  function, our findings indicate that HE-DEPSO and DEPSO are more than adequate for solving the optimization problem, with HE-DEPSO providing the best solution quality and consistency. At the same time, DEPSO maintains a faster convergence rate. This part of the investigation also highlights the difficulty that some algorithms can face when optimizing the  $\chi^2(X)$  function, as algorithms like SHADE and CoDE, despite exhibiting good performance on the CEC 2017 test set, face some difficulties in optimizing the  $\chi^2(X)$  function. We can also conclude that optimization algorithms such as SHADE, CoDE, and DE would be worth considering for optimization problems in high-energy physics. Finally, with the use of HE-DEPSO algorithm we show that the 4-zeros texture model is compatible with current experimental data and thus we can affirm that this model is physically feasible.

Future work will focus on enhancing the convergence speed of the HE-DEPSO algorithm and evaluating its effectiveness across diverse problem domains, exploring its ability to tackle more complex optimization challenges. An extended comparative analysis against other bio-inspired algorithms or other advanced DE variants could also be conducted. Furthermore, this research could be expanded by analyzing additional texture models and determining their validity based on current experimental data.

**Author Contributions:** Conceptualization, P.M.-R. and R.N.-P.; methodology, P.L.-E., P.M.-R. and R.N.-P.; software, E.M.-G.; validation, E.M.-G. and P.L.-E.; formal analysis, P.L.-E. and P.M.-R.; investigation, E.M.-G. and P.L.-E.; resources, P.M.-R. and P.L.-E.; writing—original draft preparation, E.M.-G., P.M.-R. and R.N.-P.; writing—review and editing, P.M.-R. and J.C.S.-T.-M. visualization, P.M.-R. and J.C.S.-T.-M.; supervision, P.M.-R. and J.C.S.-T.-M.; funding acquisition, J.C.S.-T.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the Autonomous University of Hidalgo (UAEH) and the National Council for Humanities, Science and Technology (CONAHCYT) with project number F003-320109.

**Data Availability Statement:** The Chi-Square function code is available on Github: [https://github.com/EmX-Mtz/Chi-Square\\_Function](https://github.com/EmX-Mtz/Chi-Square_Function)

**Acknowledgments:** The first author acknowledges the National Council for Humanities, Science and Technology (CONAHCYT) for the financial support received in pursuing graduate studies, which has been essential for the completion of this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A.

In this work, the following experimental values for the elements of the  $V_{CKM}$ , the Jarlskog invariant, and the masses of the quarks (at the scale of  $M_z$ ) are considered [53]:

$$|V_{CKM}| = \begin{pmatrix} 0.97401 \pm 0.00011 & 0.22650 \pm 0.00048 & 0.00361^{+0.00011}_{-0.00009} \\ 0.22636 \pm 0.00048 & 0.97320 \pm 0.00011 & 0.04053^{+0.00083}_{-0.00061} \\ 0.00854^{+0.00023}_{-0.00016} & 0.03978^{+0.00082}_{-0.00060} & 0.999172^{+0.000024}_{-0.000035} \end{pmatrix}, \quad (A1)$$

$$\mathcal{J} = (3.00^{+0.15}_{-0.09}) \times 10^{-5}, \quad (A2)$$

$$\begin{aligned} m_u &= 1.23 \pm 0.21 \text{ MeV}, & m_d &= 2.67 \pm 0.19 \text{ MeV}, & m_s &= 53.16 \pm 4.61 \text{ MeV}, \\ m_c &= 0.620 \pm 0.017 \text{ GeV}, & m_b &= 2.839 \pm 0.026 \text{ GeV}, & m_t &= 168.26 \pm 0.75 \text{ GeV}. \end{aligned} \quad (\text{A3})$$

## References

- Altarelli, G.; Feruglio, F. Neutrino masses and mixings: A theoretical perspective. *Phys. Rept.* **1999**, *320*, 295–318. doi:10.1016/S0370-1573(99)00067-8.
- Altarelli, G.; Feruglio, F. Models of neutrino masses and mixings. *New J. Phys.* **2004**, *6*, 106, [hep-ph/0405048]. doi:10.1088/1367-2630/6/1/106.
- Fritzsch, H.; Xing, Z.Z. Mass and flavor mixing schemes of quarks and leptons. *Prog. Part. Nucl. Phys.* **2000**, *45*, 1–81, [hep-ph/9912358]. doi:10.1016/S0146-6410(00)00102-2.
- Xing, Z.Z. Flavor structures of charged fermions and massive neutrinos. *Phys. Rept.* **2020**, *854*, 1–147, [arXiv:hep-ph/1909.09610]. doi:10.1016/j.physrep.2020.02.001.
- Gupta, M.; Ahuja, G. Flavor mixings and textures of the fermion mass matrices. *Int. J. Mod. Phys. A* **2012**, *27*, 1230033, [arXiv:hep-ph/1302.4823]. doi:10.1142/S0217751X12300335.
- Froggatt, C.D.; Nielsen, H.B. Hierarchy of Quark Masses, Cabibbo Angles and CP Violation. *Nucl. Phys. B* **1979**, *147*, 277–298. doi:10.1016/0550-3213(79)90316-X.
- King, S.F.; Luhn, C. Neutrino Mass and Mixing with Discrete Symmetry. *Rept. Prog. Phys.* **2013**, *76*, 056201, [arXiv:hep-ph/1301.1340]. doi:10.1088/0034-4885/76/5/056201.
- Borzumati, F.; Nomura, Y. Low scale seesaw mechanisms for light neutrinos. *Phys. Rev. D* **2001**, *64*, 053005, [hep-ph/0007018]. doi:10.1103/PhysRevD.64.053005.
- Lindner, M.; Ohlsson, T.; Seidl, G. Seesaw mechanisms for Dirac and Majorana neutrino masses. *Phys. Rev. D* **2002**, *65*, 053014, [hep-ph/0109264]. doi:10.1103/PhysRevD.65.053014.
- Flieger, W.; Gluza, J. General neutrino mass spectrum and mixing properties in seesaw mechanisms. *Chin. Phys. C* **2021**, *45*, 023106, [arXiv:hep-ph/2004.00354]. doi:10.1088/1674-1137/abcd2f.
- Fritzsch, H. Calculating the Cabibbo angle. *Physics Letters B* **1977**, *70*, 436–440. doi:https://doi.org/10.1016/0370-2693(77)90408-7.
- Fritzsch, H.; Xing, Z.Z. Four zero texture of Hermitian quark mass matrices and current experimental tests. *Phys. Lett. B* **2003**, *555*, 63–70, [hep-ph/0212195]. doi:10.1016/S0370-2693(03)00048-0.
- zhong Xing, Z.; hua Zhao, Z. On the four-zero texture of quark mass matrices and its stability. *Nuclear Physics B* **2015**, *897*, 302–325. doi:10.1016/j.nuclphysb.2015.05.027.
- Gómez-Ávila, S.; López-Lozano, L.; Miranda-Romagnoli, P.; Noriega-Papaqui, R.; Lagos-Eulogio, P. 2-zeroes texture and the Universal Texture Constraint, 2021, [arXiv:hep-ph/2105.01554].
- Tani, L.; Rand, D.; Veelken, C.; Kadastik, M. Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics. *The European Physical Journal C* **2021**, *81*. doi:10.1140/epjc/s10052-021-08950-y.
- Zhang, Y.; Zhou, D.; others. Application of differential evolution algorithm in future collider optimization. 7th International Particle Accelerator Conference (IPAC'16), Busan, Korea, May 8-13, 2016, pp. 1025–1027.
- Wu, J.; Zhang, Y.; Qin, Q.; Wang, Y.; Yu, C.; Zhou, D. Dynamic aperture optimization with diffusion map analysis at CEPC using differential evolution algorithm. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2020**, *959*, 163517. doi:10.1016/j.nima.2020.163517.
- Allanach, B.C.; Grellscheid, D.; Quevedo, F. Genetic algorithms and experimental discrimination of SUSY models. *Journal of High Energy Physics* **2004**, *2004*, 069. doi:10.1088/1126-6708/2004/07/069.
- Bilal, M.; Pant, M.; Zaheer, H.; Garcia-Hernandez, L.; Abraham, A. Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence* **2020**, *90*, 103479. doi:10.1016/j.engappai.2020.103479.
- Ahmad, M.F.; Isa, N.A.M.; Lim, W.H.; Ang, K.M. Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal* **2022**, *61*, 3831–3872. doi:10.1016/j.aej.2021.09.013.
- Pan, J.S.; Liu, N.; Chu, S.C. A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning. *IEEE Access* **2020**, *8*, 17691–17712. doi:10.1109/ACCESS.2020.2968119.

22. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm and Evolutionary Computation* **2019**, *44*, 546–558. doi:<https://doi.org/10.1016/j.swevo.2018.06.010>.
23. Wang, Y.; Cai, Z.; Zhang, Q. Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters. *IEEE Transactions on Evolutionary Computation* **2011**, *15*, 55–66. doi:10.1109/TEVC.2010.2087271.
24. Peñuñuri, F.; Cab, C.; Carvente, O.; Zambrano-Arjona, M.; Tapia, J. A study of the Classical Differential Evolution control parameters. *Swarm and Evolutionary Computation* **2016**, *26*, 86–96. doi:10.1016/j.swevo.2015.08.003.
25. Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for Differential Evolution. 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78. doi:10.1109/CEC.2013.6557555.
26. Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation* **2009**, *13*, 398–417. doi:10.1109/TEVC.2008.927706.
27. Das, S.; Abraham, A.; Chakraborty, U.K.; Konar, A. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation* **2009**, *13*, 526–553. doi:10.1109/TEVC.2008.2009457.
28. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M. Opposition-based differential evolution. *IEEE Transactions on Evolutionary computation* **2008**, *12*, 64–79.
29. Zhang, J.; Sanderson, A. JADE: Adaptive Differential Evolution With Optional External Archive. *Evolutionary Computation, IEEE Transactions on* **2009**, *13*, 945 – 958. doi:10.1109/TEVC.2009.2014613.
30. Chakraborty, S.; Saha, A.K.; Sharma, S.; Sahoo, S.K.; Pal, G. Comparative Performance Analysis of Differential Evolution Variants on Engineering Design Problems. *Journal of Bionic Engineering* **2022**, *19*, 1140–1160. doi:10.1007/s42235-022-00190-4.
31. Lagos-Eulogio, P.; Miranda-Romagnoli, P.; Seck-Tuoh-Mora, J.C.; Hernández-Romero, N. Improvement in Sizing Constrained Analog IC via Ts-CPD Algorithm. *Computation* **2023**, *11*, 230. doi:10.3390/computation11110230.
32. Wang, S.; Li, Y.; Yang, H. Self-adaptive mutation differential evolution algorithm based on particle swarm optimization. *Applied Soft Computing* **2019**, *81*, 105496. doi:10.1016/j.asoc.2019.105496.
33. Novaes, S.F. Standard model: An Introduction. 10th Jorge Andre Swieca Summer School: Particle and Fields, 1999, pp. 5–102, [[hep-ph/0001283](#)].
34. Langacker, P. Introduction to the Standard Model and Electroweak Physics. Theoretical Advanced Study Institute in Elementary Particle Physics: The Dawn of the LHC Era, 2010, pp. 3–48, [[arXiv:hep-ph/0901.0241](#)]. doi:10.1142/9789812838360\_0001.
35. Isidori, G.; Nir, Y.; Perez, G. Flavor Physics Constraints for Physics Beyond the Standard Model. *Ann. Rev. Nucl. Part. Sci.* **2010**, *60*, 355, [[arXiv:hep-ph/1002.0900](#)]. doi:10.1146/annurev.nucl.012809.104534.
36. Antonelli, M.; others. Flavor Physics in the Quark Sector. *Phys. Rept.* **2010**, *494*, 197–414, [[arXiv:hep-ph/0907.5386](#)]. doi:10.1016/j.physrep.2010.05.003.
37. Branco, G.C.; Lavoura, L.; Silva, J.P. *CP Violation*; Vol. 103, Int.Ser.Monogr.Phys., 1999.
38. Eidelman, S.; others. Review of particle physics. Particle Data Group. *Phys. Lett. B* **2004**, *592*, 1. doi:10.1016/j.physletb.2004.06.001.
39. Félix-Beltrán, O.; González-Canales, F.; Hernández-Sánchez, J.; Moretti, S.; Noriega-Papaqui, R.; Rosado, A. Analysis of the quark sector in the 2HDM with a four-zero Yukawa texture using the most recent data on the CKM matrix. *Phys. Lett. B* **2015**, *742*, 347–352, [[arXiv:hep-ph/1311.5210](#)]. doi:10.1016/j.physletb.2015.02.003.
40. Barranco, J.; Delepine, D.; Lopez-Lozano, L. Neutrino mass determination from a four-zero texture mass matrix. *Phys. Rev. D* **2012**, *86*, 053012. doi:10.1103/PhysRevD.86.053012.
41. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* **1995**, pp. 39–43.
42. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 1998, pp. 69–73. doi:10.1109/ICEC.1998.699146.
43. Storn, R.; Price, K. Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. *Journal of Global Optimization* **1995**, *23*.
44. Storn, R.; Price, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization* **1997**, *11*, 341–359. doi:10.1023/A:1008202821328.
45. Sánchez Vargas, O.; De León Aldaco, S.E.; Aguayo Alquicira, J.; Vela Valdés, L.G.; Mina Antonio, J.D. Differential Evolution Applied to a Multilevel Inverter—A Case Study. *Applied Sciences* **2022**, *12*. doi:10.3390/app12199910.

46. Batool, R.; Bibi, N.; Muhammad, N.; Alhazmi, S. Detection of Primary User Emulation Attack Using the Differential Evolution Algorithm in Cognitive Radio Networks. *Applied Sciences* **2023**, *13*. doi:10.3390/app13010571.
47. Elsayed, S.M.; Sarker, R.A.; Ray, T. Differential evolution with automatic parameter configuration for solving the CEC2013 competition on Real-Parameter Optimization. 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 1932–1937. doi:10.1109/CEC.2013.6557795.
48. Zhong, X.; Cheng, P. An elite-guided hierarchical differential evolution algorithm. *Applied Intelligence* **2021**, *51*, 4962–4983. doi:10.1007/s10489-020-02091-7.
49. Yang, Q.; Guo, X.; Gao, X.D.; Xu, D.D.; Lu, Z.Y. Differential Elite Learning Particle Swarm Optimization for Global Numerical Optimization. *Mathematics* **2022**, *10*. doi:10.3390/math10081261.
50. Huang, Y.; Yu, Y.; Guo, J.; Wu, Y. Self-adaptive Artificial Bee Colony with a Candidate Strategy Pool. *Applied Sciences* **2023**, *13*. doi:10.3390/app131810445.
51. Wang, S.; Li, Y.; Yang, H.; Liu, H. Self-adaptive differential evolution algorithm with improved mutation strategy. *Soft Computing* **2017**, *22*, 3433–3447. doi:10.1007/s00500-017-2588-5.
52. Kumar, A.; Price, K.V.; Mohamed, A.W.; Hadi, A.A.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*; Technical Report, 2016; pp. 1–34.
53. Fritzsche, H.; Xing, Z.z.; Zhang, D. Correlations between quark mass and flavor mixing hierarchies. *Nuclear Physics B* **2022**, *974*, 115634. doi:10.1016/j.nuclphysb.2021.115634.

**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.