2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Improving metaheuristic algorithm design through inequality and diversity analysis: A novel multi-population Differential Evolution

Alfonso Ramos-Michel, Mario A. Navarro, Diego Oliva, Bernardo Morales-Castañeda, Angel Casas-Ordaz, Arturo Valdivia
Universidad de Guadalajara, CUCEI
Guadalajara, Mexico
{juanbernardo.morales, mario.navarro, arturo.valdivia, alfonso.rmichel}@academicos.udg.mx
angel.casas5699@alumnos.udg.mx
diego.oliva@cucei.udg.mx

Erick Rodríguez-Esparza
DeustoTech,
University of Deusto,
Bilbao, Spain
erick.rodriguez@deusto.es

Seyed Jalaleddin Mousavirad
Mid Sweden University,
Sundsvall, Sweden
seyedjalaleddin.mousavirad@miun.se

***Abstract*—In evolutionary algorithms and metaheuristics, defining when applying a specific operator is important. Besides, in complex optimization problems, multiple populations can be used to explore the search space simultaneously. However, one of the main problems is extracting information from the populations and using it to evolve the solutions. This article presents the inequality-based multi-population differential evolution (IMDE). This algorithm uses the K-means to generate subpopulations (settlements). Two variables are extracted from the settlements, the diversity and the Gini index, which measure the solutions' distribution and the solutions' inequality regarding fitness. The Gini index and the diversity are used in the IMDE to dynamically modify the scalation factor and the crossover rate. Experiments over a set of benchmark functions with different degrees of complexity validate the performance of the IMDE. Besides comparisons, statistical and ranking average validate the search capabilities of the IMDE.**

***Index Terms*—Differential evolution, Multi-population, Gini index, Diversity, K-means**

I. INTRODUCTION

Evolutionary algorithms (EA) are essential for solving complex optimization problems in multidimensional search spaces [1]. EAs employ operators and rules to explore and exploit the search space for optimal solutions. The Differential Evolution (DE) [2] is an EA that uses crossover, mutation, and selection to modify candidate solutions.

DE is widely used; it has applications in feature selection for classification [3], controller design in power electronics, and parameter extraction of photovoltaic models [4]. However, like other EAs, DE faces premature convergence, resulting in suboptimal solutions [5]. This is often due to a loss of diversity in the population, which is crucial for proper search space exploration [6]. Diversity provides essential information and guides exploration decisions during iteration.

The Gini index (GI) is a statistical tool used to measure equity in the distribution of resources in a population [7]. It ranges from 0 (equality) to 1 (inequality). The GI is used to verify inequality in the values of a variable [8]. It is represented graphically by the Lorenz curve. It has several applications, such as geostatistics. Although it is a powerful tool, it has not been used in EA.

Regarding the DE, it has two main parameters that affect its behavior: the amplification or scaling factor (F) used for the mutation, the most studied DE parameter due to the importance of the mutation stage in the algorithm [9]. The crossover rate (pCR) controls when to apply the crossover operator, permitting the creation of a new solution from a mutated individual.

The pCR is a parameter commonly studied [10]. All iterations use A fixed value for pCR and F. This paper proposes a version of the DE with the main population divided into subpopulations. The parameters pCR and F are adapted at each iteration using the subpopulations' diversity and the GI. The proposal, called inequality-based multi-population DE (IMDE), creates the subpopulations (settlements) using the K-means algorithm. The Gini index is then computed for each settlement based on the fitness of its elements. Each settlement's diversity and GI are then used in the DE process as a dynamic F and pCR. The GI then measures the inequality of each settlement, and the diversity analyzes the distribution of solutions. If a subpopulation has a lower GI index, they are more probability of applying the crossover.

A number of well-known benchmark functions were used to test the IMDE [11]. For comparison purposes, other state-of-the-art EAs have also been under test. Statistical analysis and average rank test were performed to validate the effectiveness of the IMDE . The remainder sections are organized as follows: Section II provides the concepts related to DE, Gini index, diversity, and K-means. Section III explains the methodology and inspiration of the IMDE. Section IV presents the experiments and results. Finally, Section V discusses some conclusions and future work.

II. Background

A. Differential evolution algorithm

Differential Evolution (DE) has a population containing N solution vectors with d dimensions [2]. This EA consists of four main stages: initialization, crossover, mutation, and selection.

1) Initialization: N individuals are randomly generated by Eq. 1.

$$x_i^t = \{x_{1,i}^t, x_{2,i}^t \dots, x_{d,i}^t\}, \in R^D \tag{1}$$

2) Mutation: The most popular mutation for the DE is the DE/rand/1, defined in Eq. 2.

$$v_i^{t+1} = x_p^t + F(x_q^t - x_r^t) \tag{2}$$

At the generation t for each vector x_i, initially, three vectors are randomly selected x_p, x_q, and x_r. Notice that p, q, and r are random indexes. A donor vector is constructed, where $F \in [0, 2]$ is the scale or amplification factor.

3) Crossover: This stage is guided by the crossover rate (pCR), which takes a constant value in the range of [0,1]. The crossover comprises every d component of the vector v_i. Considering a random variable $r_i \in [0, 1]$. Then the j-th components of v_i are modified as follows:

$$u_{j,i}^{t+1} = \begin{cases} v_{i,j}^t & \text{if } r_i \leq pCR, \\ x_{i,j}^t & \text{otherwise,} \end{cases} \tag{3}$$

where $j = 1, 2, \dots, d$. Then, each component has a probability of the be from the donor vector of the x_i to construct the mutated vector u.

4) Selection: Here, the performance of the solutions u is evaluated. The fitness of u_i is compared with x_i, where the vector with the fittest continues in the population in the next generation.

$$x_i^{t+1} = \begin{cases} u_{i,j}^t & \text{if } f(u_i^{t+1}) \leq f(x_i^t), \\ x_{i,j}^t & \text{otherwise,} \end{cases} \tag{4}$$

B. K-means algorithm

The K-means algorithm works by dividing a set of particles by the quadratic error criterion, minimizing the performance index of the clusters. This division is an unsupervised learning clustering mechanism proposed in [12]. Given a set of d-dimensional observations $(x_1, x_2, \dots, x_n)$, the algorithm divides the n observations into k sets $S = (S_i, S_{i+1}, \dots, S_k)$; where k refers to the number of centroids set to group the data. The condition of the algorithm is that each data point can pertain to only one group. To minimize the sum of squares within the cluster S_i, (i.e., the variance). The objective is to find the equality shown in Eq. 5, where μ_i is the mean of the data in S_i.

$$argmin(S) \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = argmin(S) \sum_{i=1}^{k} |S_i| \, VarS_i \tag{5}$$

C. Dimension-wise diversity

Balancing the exploration and exploitation phases is crucial in EAs. The exploration disperses candidate solutions across the search space. In contrast, the exploitation concentrates solutions in a promising area and performs an exhaustive search. There is no established balance, as different problems require different approaches. Improving this balance involves analyzing the problem and the population's behavior toward potential solutions. Population diversity gives us a measurement of how concentrated or sparsed are the search agents along the search space. In this paper, Dimension-wise diversity [13] measurement is employed to know the behavior of the search agents. Population diversity is calculated by Eq. 6, where N is the total number of search agents, $median(x_j)$ represents the median of the j-th dimension in the entire population, and (x_{ij}) is the j-th dimension of the i-th search agent.

$$Div_j = \frac{1}{N} \sum_{i=1}^{N} |median(x_j) - x_{ij}| \tag{6}$$

After calculating the dimension-wise distance of each particle i, the average Div_j of all particles is obtained. Averaging Div_j in each dimension brings the whole population diversity, calculated through Eq. 7, where D is the problem dimensionality.

$$Div = \frac{1}{D} \sum_{j=1}^{D} Div_j \tag{7}$$

Eq. 8 is used to calculate the total percentage of diversity, where Div is the total diversity of the present iteration, and Div_{max} is the maximum diversity obtained since the first diversity measurement.

$$Div_{\%} = \frac{Div}{Div_{max}} \tag{8}$$

D. Inequality measurement

1) Gini index (GI): It is a coefficient presented by Corrado Gini in 1912 [14]. The GI allows for measuring the equity in distributing some goods among the population. It has been proved that the GI is the numerical representation of the double area between the Lorenz Curve and the straight line representing perfect equity. Besides, the GI can be formulated in different ways [14]. Among all the formulations, there is a version suitable for non-clustered data, which can be seen in Eq. 9 [15].

$$GI = \frac{\sum_{i=1}^{n-1} (P_i - Y_i)}{\sum_{i=1}^{n-1} P_i} \tag{9}$$

Assume the existence of a resource y distributed among the n individuals of a given population p. The individuals are ordered in increasing order concerning the value of the distribution of the good: $y_1 \leq y_2, \dots, \leq y_n$. A simple (p_i) and a cumulative (P_i) relative frequency distribution is constructed for the population under study and another for the distributed variable (y_i and Y_i). The GI is based on the sum of the differences ($P_i - Y_i$) and is divided by $\sum_{i=1}^{n-1} P_i$ to normalize the output values to the interval $[0, 1]$. For calculating the cumulative relative frequency of the population, we use Eq. 10, where X_i is the cumulative absolute frequency of the population.

$$P_i = \frac{X_i}{n} * 100 \tag{10}$$

The relative frequency of the distributed variable is constructed by Eq. 11, where x_i is the number of individuals in the population to which the value y_i of the distributed variable corresponds.

$$Y_i = \frac{\sum_{i=1}^{n} y_i x_i}{w_n} * 100 \tag{11}$$

III. Methodology and proposal

A. Inequality-based multi-population DE

The original concept of the inequality-based multi-population DE (IMDE) is based on the idea that a community improves when each community member can access enough goods produced as a society to have a good living. It is not a secret that where there are resources, communities of individuals are more likely to prosper. Since ancient times, humans have founded populations depending on the availability of resources in determining areas. The approach of the resources by a community has led to the growth or eventual disappearance of the population.

In this article, the general structure of the original DE algorithm is used. However, instead of using a single population, multiple populations (settlements) are created by the K-means. Similar to human settlements, these populations are expected to evolve in the pursuit of better production, measured by each individual's fitness. The GI assesses the concentration or dispersion of production within a population.

B. IMDE Methodology

The methodology used to design the proposed algorithm is based on three main points:

- Modify the DE structure using the K-means to create multiple populations in the initialization stage. The DE then operates the populations independently.
- Analyze the evolution of the scaling factor (F) and crossover rate (pCR) values using dimension-wise diversity and the GI, respectively.
- Compare the proposed IMDE with relevant state-of-the-art DE modifications and other EAs.

C. IMDE algorithm: general structure

The flowchart of Fig.1 presents the structure of the IMDE. The initialization and the iterative process are explained in this subsection.

1) Initialization : Two critical parameters for the operation of IMDE are the number of particles N, and the number of settlements k. By setting the population, particles normally distributed over the search space are randomly generated and clustered into the k settlements (not uniformly sized) by K-means. For this purpose, we perform the sensitivity analysis of the Subsection III-D. The last important step in the initialization stage is calculating each settlement's diversity and best solution.

2) Iterative process : The first brings a significant change concerning the original DE because the scaling factor (F) is no longer a constant but a dynamic parameter calculated using Eq. 12. The scaling factor is calculated at each iteration for each settlement. After that, the GI is computed using Eq. 9. The algorithm sets $pCR = GI$ or a random value (Eq. 13), depending on the GI value previously obtained. IMDE performs the standard DE stages for each settlement when both parameters are calculated. This process continues until the stop criteria is reached and the best solution is selected.

D. Sensitivity tests for adjusting initialization parameters

In the IMDE the K-means generate the settlements. However, k is an extra parameter and should be properly tuned. A sensitivity test has been performed to identify the best k. For this test, the population is set as $N = 100$, and four functions were selected, one of each group (multimodal, unimodal, hybrid, and shifted), all in 10 dimensions. The value of k changes along the test from 1 to 10 one by one.

To numerically analyze the influence of the k, 30 independent runs were performed. Table I shows the results of the average (AVG) and standard deviation (SD) of the fitness. In Table I sensitivity tests, it is evident that $k = 2$ is the best settling value, as this number of clusters provides the lowest AVG and STD across all four functions. $k = 3$ is also a good choice for the multimodal and shifted functions. However, the behavior is to be expected since the multimodal and shifted functions have complex landscapes where more clusters benefit the search by escaping local optima. However, the excessive use of populations also affects the performance of the proposed approach. For this reason, $k = 2$ is the initialization parameter used in the IMDEto perform the experiments.

TABLE I: Statistical analysis of the sensitivity test.

Function	IMDE	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Multimodal f_6	AVG	3.80E+00	**1.07E-15**	**1.07E-15**	1.42E-15	1.78E-15	7.11E-15	3.14E-10	7.92E-13	2.89E-08	9.23E-03
	SD	6.16E+00	**1.72E-15**	**1.72E-15**	1.83E-15	1.87E-15	1.51E-14	9.93E-10	1.83E-12	9.04E-08	2.40E-02
Uni-modal f_1	AVG	1.53E+02	**8.34E-46**	1.11E-32	1.32E-26	1.43E-21	7.01E-17	9.23E-12	1.25E-08	4.96E-03	6.02E-02
	SD	4.85E+02	**2.62E-45**	3.50E-32	3.75E-26	4.03E-21	2.18E-16	1.97E-11	3.89E-08	1.10E-02	1.90E-01
Composite f_{31}	AVG	1.22E+02	**1.30E-26**	1.29E-25	7.10E-25	1.46E-20	9.06E-17	6.54E-13	2.39E-01	6.96E-12	1.23E-02
	SD	3.85E+02	**1.20E-26**	1.77E-25	8.26E-25	4.26E-20	2.86E-16	2.06E-12	7.54E-01	1.43E-11	3.62E-02
Shifted f_{35}	AVG	4.19E+00	**0.00E+00**	**0.00E+00**	1.07E-15	1.07E-15	6.06E-13	4.58E-14	6.53E-11	2.63E-03	1.65E-01
	SD	6.79E+00	**0.00E+00**	**0.00E+00**	1.72E-15	1.72E-15	1.91E-12	9.07E-14	2.06E-10	8.32E-03	5.20E-01

E. Evolution of scaling factor using dimension-wise diversity

Typically, the scaling factor F value is commonly set with values of 0.8, 0.9, or 1. However, for a particular class of problems fixed, this parameter causes premature convergence. In this work, dimension-wise diversity is used to determine the diversity among the elements of a specific population [13]. We use this criterion to determine each settlement's dimensional diversity in each iteration. The diversity is then

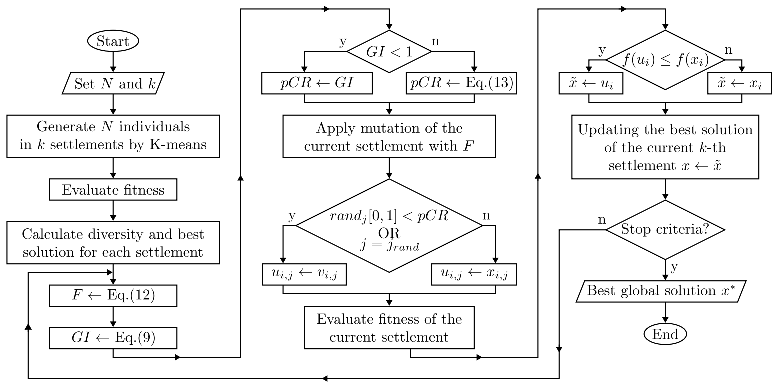


Fig. 1: Flowchart of the proposed IMDE algorithm.

normalized with values from 0 to 1, which allows us to work with div_{max}, which is equal to the sum of the diversities of all settlements. This value is normalized as 1 or the maximum diversity reached in the initialization, representing 100% of the diversity. The second value is the div_{min}, corresponding to a percentage of the maximum diversity, also expressed in values between 0 and 1. After this, a scaling factor F is calculated as:

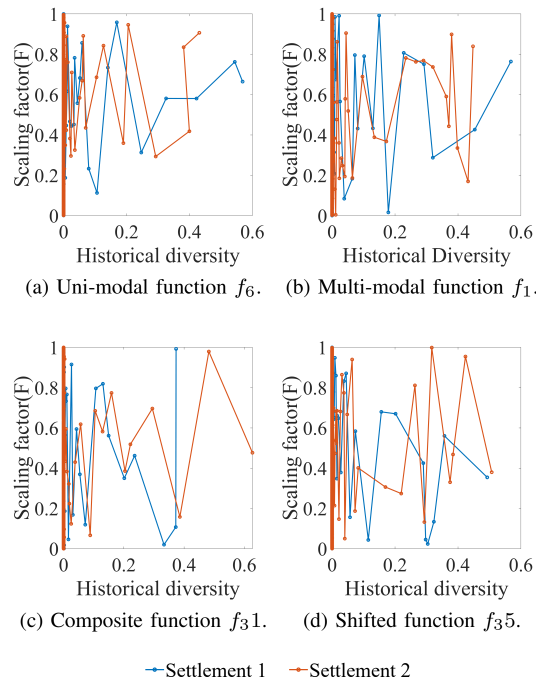
$$F = div_{min} + (div_{max} - div_{min}) \cdot rand \tag{12}$$

Where $rand$ is a random number between 0 and 1, allowing to set a value for F between the range of div_{min} and div_{max}. Fig. 2 (a-d) shows four graphs corresponding to each of the chosen functions (same as the sensitive analysis). In blue and orange, each of the settlements is differentiated; these were plotted based on the optimal parameter $k = 2$. The Y axis represents the values of the parameter F as a function of the historical of the diversity for each settlement represented on the X axis.

In Fig. 2, we can observe in both settlements that, when the diversity is higher, commonly in the first stages of the iterative process, the scaling factor F tends to have high values which is very convenient in the initial stages. In contrast, when the diversity decreases, the range increases, propitiating that the scaling factor F has more diverse values between the range of $0.1 - 0.9$ approximately. Finally, when the diversity tends to 0, i.e., in the final stage of the iterative process with diversities inferior to 0.1 that represents 10% of the total diversity, the values of F commute quickly between 0 and 1. This means that the algorithm in the exploitation stage can continue finding the diversity of solutions or intensify and refine the solutions.

F. Evolution of crossover probability using the GI

Once the mutation is performed and depending on whether the pCR is high or low, this operator will create a new solution based on the elements selected during the mutation process. The values for the pCR should be low, typically fixed to 0.2 or 0.3. However, here is implemented a pCR that varies dynamically according to the GI. The process to establish the pCR first involves calculating the GI, normalized from 0 to 1, and determining using the fitness of each



(a) Uni-modal function f_6. (b) Multi-modal function f_1.

(c) Composite function f_31. (d) Shifted function f_35.

Fig. 2: Dynamic adjustment of F based on the diversity of each settlement.

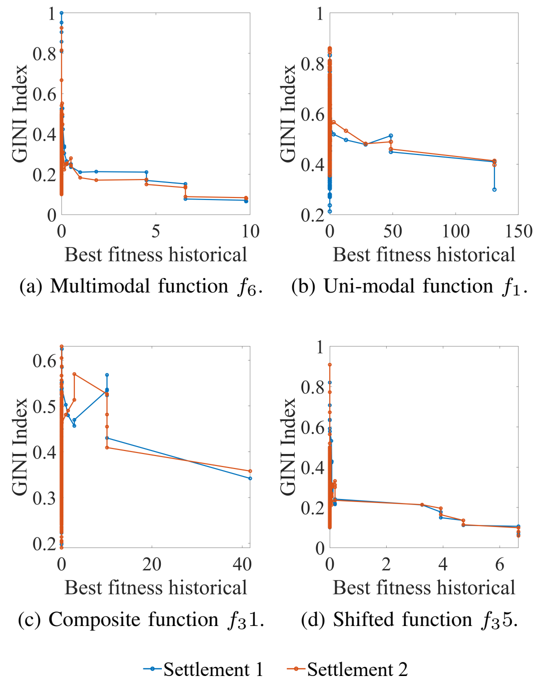
and the number of individuals per settlement using the Eq. 9. Once the GI has been computed, there are two cases:

Case 1: When the GI is less than or equal to 1, the value is equated to the crossover probability (pCR), i.e., $pCR = GI$.

Case 2: When the GI cannot be calculated due to the nature of the data, the crossover probability pCR will be chosen from a range of $0.1 - 0.5$ with the following expression:

$$pCR = 0.1 + (0.5 - 0.1) \cdot rand \tag{13}$$

The analysis of the evolution of the pCR parameter as a function of the GI shows interesting results. In Fig. 3, the four graphs of different classes of functions show similar behaviors although, with different values. It is evident that while the fitness value is high in the initial stages of the algorithm, the GI is low, which indicates a good diversity



(a) Multimodal function f_6. (b) Uni-modal function f_1.

(c) Composite function f_31. (d) Shifted function f_35.

Fig. 3: Dynamic adjustment of pCR based on the Gini index of each settlement.

of solutions and different fitness values between each settlement, obtaining pCR values between the ranges of $[0.1-0.3]$ for Fig. 3 (a), values of $[0.1-0.6]$ for Fig. 3 (b), pCR between $[0.35-0.55]$ approximately for Fig. 3 (c) and, finally pCR with a range of $[0.1-0.3]$ for Fig. 3 (d), these ranges of values are very appropriate considering that the ideal values for pCR should be low as evidenced by several articles of the state of the art. On the other hand, all the plots show that when the fitness diversity is very low in the final stages of the algorithm, the GI tends to 1, but for a very short time. This could benefit if, in some complex functions where the algorithm suffers from premature convergence, allowing it to escape from local optima. The direct relationship of the GI with the pCR is then given by the fitness in each settlement, which allows a lower probability of crossover in the early stages of the algorithm. At the same time, in the algorithm's late steps, the PCR can be more random with high values, which would provide diversity in the solutions of promising zones.

IV. Experiments Setup and Results

We choose four types of benchmark functions that test different aspects of the algorithms. The test set consists of 43 functions divided into five unimodal, 25 multimodal, four composite, and nine shifted. The obtained results from testing IMDE are compared against some DE and Particle Swarm Optimization (PSO)variants. The algorithms used to compare our proposal are Self-adaptive DE (JADE) [16], Linear Population Size Reduction with Adaptive DE (L-SHADE) [17], Differential Evolution (DE) [2], Estimation of Distribution Algorithm (EDA) [18], Autonomous Particles Groups for PSO (AGPSO) [19], Improved PSO (IPSO) [20], PSO with asymmetric time-varying (MPSO) [21], and Particle Swarm Optimization (PSO) [22]. The experiments were performed under the same conditions setting as stopping criteria 50,000 function accesses for 30 dimensions, the population size for each algorithm was 50 search agents. In the case of algorithms working with dynamically changing populations, the initialization parameters were set as suggested by the authors in the original publications.

Tables II and III present the statistical results of the AVG and the SD after 30 executions of the nine algorithms for each problem considering 30 dimensions. The algorithms are evaluated in each run until the maximum number of function accesses is reached. The results that achieved the minimal AVG value and SD are highlighted in bold. Table III shows that the IMDE algorithm obtained the best results in four out of five unimodal functions using 30 dimensions for both statistical results. The algorithm that won in the F5 instance is the IPSO.

The study conducted statistical tests and used a ranking average to compare different algorithms on 43 optimization functions. Tables II and III show that the IMDE outperformed other algorithms, achieving the best results in most functions and obtaining the highest rank in 28 of them. In the case of composite functions (Table II), the proposed algorithm performed well, obtaining the best results for two out of four instances. However, the LSHADE algorithm outperformed in the remaining instances. For shifted functions, the proposed algorithm achieved minimum values in more than half of the tested instances. Overall, the results demonstrate the efficiency and effectiveness of the proposed method in optimizing various types of problems and finding optimal solutions with stability.

TABLE II: Results of Composite and Shifted functions considering 30 dimensions.

Function		IMDE	DE	JADE	LSHADE	EDA	PSO	AGPSO	IPSO	MPSO
F31.fx15	AVG	**1.41E-13**	5.67E+03	7.69E-01	1.65E-05	6.14E+03	4.86E-04	5.57E-04	4.15E-04	3.68E+03
	SD	**4.94E-14**	3.60E+03	3.93E+00	1.55E-05	5.91E+03	3.05E-04	1.40E-03	8.05E-04	7.21E+03
	Rank AVG	1	8	6	2	9	4	5	3	7
F32.fx16	AVG	**2.90E+01**	2.39E+02	3.33E+01	2.90E+01	2.57E+02	6.29E+01	8.15E+01	1.00E+02	9.45E+01
	SD	**3.61E-10**	4.30E+01	3.66E+00	1.91E-08	7.38E+01	1.69E+01	1.42E+01	2.16E+01	2.38E+01
	Rank AVG	1	7	2	1	8	3	4	6	5
F33.fx17	AVG	1.53E+02	1.30E+06	1.63E+02	**3.20E+01**	1.31E+07	9.01E+01	4.26E+01	3.55E+01	5.07E+01
	SD	2.97E+01	2.48E+06	9.82E+01	**4.40E-05**	1.13E+07	2.37E+01	8.58E+00	3.05E+00	1.74E+01
	Rank AVG	6	8	7	1	9	5	3	2	4
F34.fx18	AVG	4.41E+01	2.93E+02	3.56E+01	**2.90E+01**	4.57E+02	4.96E+01	8.10E+01	9.97E+01	3.22E+02
	SD	2.62E+01	5.59E+01	7.61E+00	**8.40E-05**	5.80E+02	1.54E+01	1.71E+01	2.38E+01	2.66E+02
	Rank AVG	3	7	2	1	9	4	5	6	8
F35.ShiftedAckley	AVG	**2.63E-11**	1.09E+01	5.69E+00	1.64E-06	1.07E+01	6.60E-04	1.78E+00	9.00E-01	1.17E+00
	SD	**2.27E-11**	1.34E+00	1.24E+00	1.00E-06	1.86E+00	4.55E-04	5.19E-01	7.87E-01	2.56E+00
	Rank AVG	1	9	7	2	8	3	6	4	5
F36.ShiftedPowell	AVG	2.43E-01	1.29E+02	9.40E-01	**2.92E-06**	6.12E+02	5.96E-03	9.91E+00	6.09E+01	2.81E+02
	SD	2.13E-01	7.50E+01	5.15E+00	**1.92E-06**	4.73E+02	2.82E-03	2.69E+01	7.69E+01	3.70E+02
	Rank AVG	3	7	4	1	9	2	5	6	8
F37.ShiftedRastringin	AVG	**2.67E-12**	1.71E+02	1.74E+01	1.12E+01	8.55E+01	4.47E+01	4.99E+01	8.08E+01	9.79E+01
	SD	**3.01E-12**	3.29E+01	3.26E+00	2.39E+00	1.98E+01	1.26E+01	1.91E+01	2.28E+01	2.53E+01
	Rank AVG	1	9	3	2	7	4	5	6	8
F38.ShiftedRosenbrock	AVG	4.13E+02	7.66E+04	2.36E+01	**2.11E+01**	5.03E+04	4.79E+01	2.17E+03	2.14E+02	4.90E+04
	SD	6.24E+02	4.36E+04	1.04E+01	**6.31E-01**	5.09E+04	3.68E+01	8.48E+03	4.22E+02	3.61E+04
	Rank AVG	5	9	2	1	8	3	6	4	7
F39.ShiftedRothyp	AVG	**5.80E-22**	1.97E+04	2.92E-06	3.85E-10	1.93E+04	5.77E-05	1.43E+02	2.00E+03	1.03E+04
	SD	**5.59E-22**	7.49E+03	9.47E-06	3.51E-10	9.13E+03	7.33E-05	7.84E+02	4.61E+03	1.10E+04
	Rank AVG	1	9	3	2	8	4	5	6	7
F40.ShiftedSchwefel2	AVG	**1.79E-20**	6.99E+05	3.47E-03	1.67E-08	7.42E+05	6.67E+02	3.40E+04	1.54E+05	5.41E+05
	SD	**2.51E-20**	3.44E+05	1.89E-02	1.35E-08	4.40E+05	2.54E+03	8.59E+04	2.00E+05	6.20E+05
	Rank AVG	1	8	3	2	9	4	5	6	7
F41.ShiftedSchwefel22	AVG	9.77E+00	6.65E+10	7.48E+00	**1.40E-01**	2.67E+02	5.43E+01	8.13E+01	1.62E+02	4.47E+02
	SD	1.69E+01	1.76E+11	4.87E+00	**2.21E-01**	6.90E+01	1.65E+02	1.32E+02	1.07E+02	2.68E+02
	Rank AVG	3	9	2	1	7	4	5	6	8
F42.ShiftedSphere	AVG	**3.43E-26**	1.17E+01	1.34E-09	1.34E-13	9.59E+00	3.57E-08	9.09E-11	4.17E-11	8.33E-10
	SD	**1.10E-26**	3.81E+00	3.28E-09	1.12E-13	4.49E+00	4.87E-08	1.64E-10	1.30E-10	2.03E-09
	Rank AVG	1	9	6	2	8	7	4	3	5
F43.ShiftedSum2	AVG	**5.03E-24**	4.47E+02	1.16E-07	8.12E-12	4.93E+02	8.57E-07	1.33E+01	1.67E+01	1.57E+02
	SD	**5.64E-24**	1.30E+02	3.02E-07	8.43E-12	2.87E+02	1.24E-06	4.34E+01	4.61E+01	2.31E+02
	Rank AVG	1	8	3	2	9	4	5	6	7

V. Conclusions

In this paper, the DE algorithm has been improved with two specific changes: a multi-population scheme and a strategy to dynamically adjust the pCR and F tuning parameters according to the diversity of the subpopulations (settlements) and with the Gini index. The proposed algorithm, called IMDE, creates the settlements using the K-means. The Gini index is then computed for each settlement based on the fitness of its elements. Each settlement's diversity and Gini index are then used in the DE process as a dynamic F and pCR. First, as the scaling factor, F governs the magnitude of the changes in the mutation. Dynamically adjusting it based on the population diversity allows the algorithm to better adapt to the specific challenges that each optimization function entails. On the other hand, the pCR parameter greatly influences the crossover. Adjusting it based on the Gini index allows the algorithm to maintain a healthy population along the search. The experiments and comparison validate the good capabilities of the IMDE to overcome different optimization problems.

TABLE III: Results of Unimodal and Multimodal functions considering 30 dimensions.

Function		IMDE	DE	JADE	LSHADE	EDA	PSO	AGPSO	IPSO	MPSO
F1.Rothyp	AVG	**1.83E-22**	1.98E+04	7.92E-06	4.01E-10	2.05E+04	7.23E-05	3.78E-06	1.15E+03	6.59E+03
	SD	**3.66E-23**	9.22E+03	3.94E-05	2.65E-10	1.20E+04	1.15E-04	1.44E-05	2.51E+03	7.46E+03
	Rank AVG	1	8	4	2	9	5	3	6	7
F2.Schwefel2	AVG	**2.46E-20**	7.32E+05	7.54E-04	3.20E-08	8.46E+05	2.68E-03	7.00E+03	8.20E+04	4.98E+05
	SD	**2.75E-20**	3.00E+05	3.37E-03	2.73E-08	3.86E+05	5.37E-03	1.88E+04	1.52E+05	3.71E+05
	Rank AVG	1	8	3	2	9	4	5	6	7
F3.Sphere	AVG	**1.03E-25**	1.09E+01	1.55E-09	2.06E-13	9.89E+00	3.60E-08	3.44E-11	5.04E-12	4.25E-09
	SD	**9.51E-26**	4.09E+00	6.46E-09	4.35E-13	6.26E+00	8.50E-08	7.09E-11	1.59E-11	2.06E-08
	Rank AVG	1	9	5	2	8	7	4	3	6
F4.Sum2	AVG	**2.72E-24**	4.75E+02	2.29E-07	1.06E-11	5.12E+02	1.71E-06	6.73E-07	1.67E+01	1.73E+02
	SD	**2.40E-24**	2.41E+02	8.04E-07	2.60E-11	2.10E+02	2.25E-06	3.66E-06	4.61E+01	2.20E+02
	Rank AVG	1	8	3	2	9	5	4	6	7
F5.Sumpow	AVG	9.31E-29	3.34E-02	6.04E-18	8.77E-32	2.07E-02	1.18E-28	3.97E-27	**8.05E-45**	2.01E-30
	SD	1.61E-28	1.81E-01	1.43E-17	3.31E-31	3.33E-02	2.80E-28	1.62E-26	**2.03E-44**	1.10E-29
	Rank AVG	4	9	7	2	8	5	6	1	3
F6.Ackley	AVG	**1.90E-11**	4.82E+00	6.40E+00	1.54E-06	1.02E+01	7.80E-04	1.75E+00	1.07E+00	9.79E-01
	SD	**8.07E-12**	1.80E+00	3.01E+00	7.53E-07	1.72E+00	5.16E-04	5.83E-01	7.37E-01	8.59E-01
	Rank AVG	1	7	8	2	9	3	6	5	4
F7.Dixon	AVG	7.57E-01	2.12E+03	6.67E-01	**6.67E-01**	3.80E+04	7.92E-01	3.61E+00	3.36E+01	6.72E+01
	SD	1.57E-01	2.89E+03	2.52E-03	**7.94E-11**	2.28E+04	2.85E-01	1.54E+01	8.35E+01	1.08E+02
	Rank AVG	2	7	1	1	8	3	4	5	6
F8.Griewank	AVG	**0.00E+00**	6.06E+00	1.04E-03	2.22E-10	3.35E+01	1.33E-02	4.72E-02	1.64E-02	1.30E-02
	SD	**0.00E+00**	4.64E+00	2.32E-03	2.14E-10	1.36E+01	1.93E-02	3.62E-02	2.08E-02	1.46E-02
	Rank AVG	1	8	3	2	9	5	7	6	4
F9.Infinity	AVG	**8.69E-42**	1.94E-03	2.11E-22	1.06E-34	5.01E-02	6.10E-18	1.05E-28	4.46E-33	8.44E-28
	SD	**1.50E-41**	2.52E-03	1.12E-21	1.88E-34	4.98E-02	8.55E-18	3.91E-28	1.27E-32	2.64E-27
	Rank AVG	1	8	6	2	9	7	4	3	5
F10.Levy	AVG	**1.51E-24**	1.97E+00	3.34E-02	3.79E-12	1.13E+01	5.33E-01	3.54E-01	1.54E+00	3.63E+00
	SD	**8.11E-25**	1.30E+00	1.16E-01	3.77E-12	4.73E+00	8.33E-01	5.74E-01	1.31E+00	4.30E+00
	Rank AVG	1	7	3	2	9	5	4	6	8
F11.Mishra1	AVG	2.00E+00	4.68E+00	2.00E+00	2.00E+00	**1.83E-01**	2.53E+01	2.00E+00	2.00E+00	2.70E+00
	SD	**0.00E+00**	2.18E+00	**0.00E+00**	**0.00E+00**	3.98E-01	1.13E+02	**0.00E+00**	**0.00E+00**	2.14E+00
	Rank AVG	2	4	2	2	1	5	2	2	3
F12.Mishra2	AVG	2.00E+00	4.69E+00	2.00E+00	2.00E+00	**4.22E-01**	1.06E+01	2.00E+00	2.00E+00	2.60E+00
	SD	**0.00E+00**	2.00E+00	**0.00E+00**	**0.00E+00**	4.99E-01	1.58E+01	**0.00E+00**	**0.00E+00**	1.81E+00
	Rank AVG	2	4	2	2	1	5	2	2	3
F13.Mishra11	AVG	**0.00E+00**	3.23E-05	**0.00E+00**	1.48E-05	4.02E-02	9.96E-20	1.33E-06	3.24E-11	**0.00E+00**
	SD	**0.00E+00**	7.79E-05	**0.00E+00**	5.00E-05	1.03E-02	1.81E-19	7.27E-06	1.78E-10	**0.00E+00**
	Rank AVG	1	6	1	5	7	2	4	3	1
F14.MultiModal	AVG	**0.00E+00**	**0.00E+00**	6.94E-49	9.23E-79	1.79E-232	1.80E-64	9.81E-97	4.72E-100	1.23E-107
	SD	**0.00E+00**	**0.00E+00**	3.63E-48	4.81E-78	**0.00E+00**	9.86E-64	5.37E-96	2.59E-99	6.74E-107
	Rank AVG	1	1	8	6	2	7	5	4	3
F15.Penalty1	AVG	**7.28E+01**	4.09E+05	2.78E+02	7.45E+01	1.13E+07	8.13E+01	8.22E+01	8.38E+01	8.38E+01
	SD	**1.78E+00**	5.50E+05	1.49E+02	9.64E+00	8.83E+06	3.28E+00	3.26E+00	2.80E+00	3.04E+00
	Rank AVG	1	7	6	2	8	3	4	5	5
F16.Penalty2	AVG	**9.85E+01**	8.10E+05	1.32E+02	1.10E+02	1.43E+07	1.04E+02	1.00E+02	1.02E+02	1.02E+02
	SD	**1.61E+00**	1.43E+06	1.69E+01	3.77E+00	1.67E+07	4.11E+00	5.41E+00	5.17E+00	5.03E+00
	Rank AVG	1	7	6	5	8	4	2	3	3
F17.Plateau	AVG	**3.00E+01**	3.06E+01	**3.00E+01**	**3.00E+01**	3.69E+01	**3.00E+01**	**3.00E+01**	3.01E+01	3.09E+01
	SD	**0.00E+00**	8.90E-01	**0.00E+00**	**0.00E+00**	3.02E+00	**0.00E+00**	**0.00E+00**	7.30E-01	1.78E+00
	Rank AVG	1	3	1	1	5	1	1	2	4
F18.Powell	AVG	1.33E+01	5.57E+01	1.86E-01	**2.73E-06**	7.51E+02	5.01E-03	9.38E+00	2.43E+01	3.30E+02
	SD	2.11E+01	4.95E+01	1.02E+00	**2.46E-06**	5.62E+02	1.77E-03	2.23E+01	4.02E+01	1.21E+02
	Rank AVG	5	7	3	1	9	2	4	6	8
F19.Qing	AVG	**9.39E-14**	4.38E+08	9.98E+00	9.05E-01	3.09E+09	3.35E-01	1.70E-04	9.83E-06	3.99E-03
	SD	**1.04E-13**	6.04E+08	9.05E+00	3.87E+00	2.34E+09	5.18E-01	3.85E-04	4.24E-05	8.78E-03
	Rank AVG	1	8	7	6	9	5	3	2	4
F20.Quartic	AVG	1.06E+01	9.55E+00	1.11E+01	**9.16E+00**	1.34E+01	9.58E+00	1.04E+01	1.03E+01	1.11E+01
	SD	**1.79E-01**	5.34E-01	4.80E-01	3.87E-01	2.41E+00	3.85E-01	7.79E-01	7.00E-01	1.21E+00
	Rank AVG	6	2	7	1	8	3	5	4	7
F21.Quintic	AVG	**4.33E-08**	1.98E+02	2.03E+00	1.62E-03	3.65E+03	2.40E-03	1.17E-01	6.87E-02	7.07E-01
	SD	**7.40E-08**	1.92E+02	1.08E+00	1.24E-03	2.54E+03	1.65E-03	2.98E-01	1.64E-01	1.80E+00
	Rank AVG	1	8	7	2	9	3	5	4	6
F22.Rastringin	AVG	**3.80E-08**	1.38E+01	1.77E+01	1.07E+01	8.52E+01	4.34E+01	4.46E+01	7.73E+01	1.00E+02
	SD	**6.58E-08**	4.19E+00	2.82E+00	3.00E+00	1.81E+01	1.27E+01	1.31E+01	2.50E+01	3.15E+01
	Rank AVG	1	3	4	2	8	5	6	7	9
F23.Rosenbrock	AVG	4.20E+01	3.68E+03	2.84E+01	**2.08E+01**	4.12E+04	5.54E+01	8.68E+01	2.06E+02	1.09E+04
	SD	3.90E+01	3.48E+03	2.10E+01	**7.87E-01**	4.31E+04	3.97E+01	1.90E+02	5.03E+02	2.57E+04
	Rank AVG	3	7	2	1	9	4	5	6	8
F24.Schwefel22	AVG	**1.07E-09**	1.32E+01	7.74E+00	1.36E-01	3.17E+02	1.04E+02	3.96E+01	1.68E+02	4.28E+02
	SD	**8.13E-10**	2.39E+01	4.86E+00	1.87E-01	1.06E+02	2.13E+02	5.52E+01	1.27E+02	2.08E+02
	Rank AVG	1	4	3	2	8	6	5	7	9
F25.Schwefel26	AVG	-1.25E+04	-1.16E+04	-9.68E+03	-1.20E+04	**-5.73E+04**	-6.77E+03	-1.01E+04	-9.78E+03	-8.68E+03
	SD	**6.84E+01**	2.61E+02	5.80E+02	1.66E+02	1.04E+05	7.16E+02	6.08E+02	4.82E+02	6.89E+02
	Rank AVG	2	4	7	3	1	9	5	6	8
F26.Step	AVG	**0.00E+00**	7.24E+02	**0.00E+00**	**0.00E+00**	3.38E+03	3.33E-02	3.00E-01	2.43E+00	2.50E+00
	SD	**0.00E+00**	7.04E+02	**0.00E+00**	**0.00E+00**	1.89E+03	1.83E-01	5.96E-01	6.22E+00	2.70E+00
	Rank AVG	1	6	1	1	7	2	3	4	5
F27.Stybtang	AVG	**-1.17E+03**	-1.13E+03	-1.16E+03	-1.17E+03	-8.73E+02	-1.02E+03	-1.07E+03	-1.05E+03	-1.06E+03
	SD	**1.61E-13**	2.41E+01	1.46E+01	4.31E+00	4.66E+01	3.31E+01	2.77E+01	3.58E+01	3.59E+01
	Rank AVG	1	3	2	1	8	7	4	6	5
F28.Trid	AVG	5.64E+03	7.03E+04	-3.95E+03	**-4.93E+03**	2.20E+05	-1.33E+03	4.90E+03	5.88E+03	2.20E+05
	SD	1.24E+04	5.07E+04	8.96E+02	**5.22E-02**	8.60E+04	1.83E+03	2.94E+04	2.04E+04	1.91E+05
	Rank AVG	5	7	2	1	8	3	4	6	8
F29.Vincent	AVG	**-3.00E+01**	-2.99E+01	-2.99E+01	-2.98E+01	-1.04E+01	-3.00E+01	-3.00E+01	-2.97E+01	-2.95E+01
	SD	**2.98E-10**	6.16E-02	3.84E-02	7.16E-02	1.60E+00	6.78E-08	8.40E-10	6.52E-01	8.48E-01
	Rank AVG	1	2	2	3	6	1	1	4	5
F30.Zakharov	AVG	1.32E+02	2.13E+01	4.09E+02	**4.70E-04**	5.63E+02	8.57E+00	4.03E+01	2.76E+01	1.13E+02
	SD	1.72E+01	1.18E+01	9.45E+01	**5.12E-04**	1.70E+02	1.44E+01	5.65E+01	4.41E+01	6.94E+01
	Rank AVG	7	3	8	1	9	2	5	4	6

References

[1] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 59–110, 2022.

[2] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[3] P. Wang, B. Xue, J. Liang, and M. Zhang, "Differential evolution with duplication analysis for feature selection in classification," *IEEE Transactions on Cybernetics*, 2022.

[4] Z. Hu, W. Gong, and S. Li, "Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models," *Energy Reports*, vol. 7, pp. 916–928, 2021.

[5] B. Morales-Castaneda, O. Maciel-Castillo, M. A. Navarro, I. Aranguren, A. Valdivia, A. Ramos-Michel, D. Oliva, and S. Hinojosa, "Handling stagnation through diversity analysis: A new set of operators for evolutionary algorithms," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–7.

[6] D. Oliva, E. Rodriguez-Esparza, M. S. Martins, M. Abd Elaziz, S. Hinojosa, A. A. Ewees, and S. Lu, "Balancing the influence of evolutionary operators for global optimization," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[7] F. A. Farris, "The gini index and measures of inequality," *The American Mathematical Monthly*, vol. 117, no. 10, pp. 851–864, 2010.

[8] B. Crawford, H. Khayyam, A. S. Milani, and R. N. Jazar, "Big data modeling approaches for engineering applications," in *Nonlinear approaches in engineering applications*. Springer, 2020, pp. 307–365.

[9] K. Opara and J. Arabas, "Comparison of mutation strategies in differential evolution–a probabilistic perspective," *Swarm and Evolutionary Computation*, vol. 39, pp. 53–69, 2018.

[10] D. Bajer, "Parameter control for differential evolution by storage of successful values at an individual level," *Journal of Computational Science*, vol. 68, p. 101985, 2023.

[11] W. Wong and C. I. Ming, "A review on metaheuristic algorithms: Recent trends, benchmarking and applications," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, 2019, pp. 1–5.

[12] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, 1967, pp. 281–297.

[13] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "On the exploration and exploitation in popular swarm-based metaheuristic algorithms," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7665–7683, 2019.

[14] L. Ceriani and P. Verme, "The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini," *The Journal of Economic Inequality*, vol. 10, no. 3, pp. 421–443, 2012.

[15] M. Fernando, "Consideraciones sobre el índice de gini para medir la concentración del ingreso," *UN, CEPAL, ECLAC. Serie de estudios estadísticos y prospectivos. Santiago de Chile, www. eclac. cl*, 2001.

[16] J. Zhang and A. C. Sanderson, "Jade: Self-adaptive differential evolution with fast and reliable convergence performance," in *2007 IEEE congress on evolutionary computation*. IEEE, 2007, pp. 2251–2258.

[17] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[18] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *International conference on parallel problem solving from nature*. Springer, 1996, pp. 178–187.

[19] S. Mirjalili, A. Lewis, and A. S. Sadiq, "Autonomous particles groups for particle swarm optimization," *Arabian Journal for Science and Engineering*, vol. 39, no. 6, pp. 4683–4697, 2014.

[20] Z. Cui, J. Zeng, and Y. Yin, "An improved pso with time-varying accelerator coefficients," in *2008 Eighth International Conference on Intelligent Systems Design and Applications*, vol. 2, 2008, pp. 638–643.

[21] G. Q. Bao and K. F. Mao, "Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 2134–2139.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.