

Article

# Computational Cost Reduction in Multi-Objective Feature Selection Using Permutational-Based Differential Evolution

Jesús-Arnulfo Barradas-Palmeros <sup>1,\*</sup> , Efrén Mezura-Montes <sup>1</sup> , Rafael Rivera-López <sup>2</sup> ,  
Hector-Gabriel Acosta-Mesa <sup>1</sup>  and Aldo Márquez-Grajales <sup>3</sup> 

<sup>1</sup> Artificial Intelligence Research Institute, Universidad Veracruzana, Xalapa 91097, Veracruz, Mexico; emezura@uv.mx (E.M.-M.); heacosta@uv.mx (H.-G.A.-M.)

<sup>2</sup> Departamento de Sistemas y Computación, Instituto Tecnológico de Veracruz, Veracruz 91897, Veracruz, Mexico; rafael.rl@veracruz.tecnm.mx

<sup>3</sup> INFOTEC Center for Research and Innovation in Information and Communication Technologies, Pocitos, Aguascalientes 20326, Aguascalientes, Mexico; li.aldomg@gmail.com

\* Correspondence: zs23000652@estudiantes.uv.mx

**Abstract:** Feature selection is a preprocessing step in machine learning that aims to reduce dimensionality and improve performance. The approaches for feature selection are often classified according to the evaluation of a subset of features as filter, wrapper, and embedded approaches. The high performance of wrapper approaches for feature selection is associated at the same time with the disadvantage of high computational cost. Cost-reduction mechanisms for feature selection have been proposed in the literature, where competitive performance is achieved more efficiently. This work applies the simple and effective resource-saving mechanisms of the fixed and incremental sampling fraction strategies with memory to avoid repeated evaluations in multi-objective permutational-based differential evolution for feature selection. The selected multi-objective approach is an extension of the DE-FS<sup>PM</sup> algorithm with the selection mechanism of the GDE3 algorithm. The results showed high resource savings, especially in computational time and the number of evaluations required for the search process. Nonetheless, it was also detected that the algorithm's performance was diminished. Therefore, the results reported in the literature on the effectiveness of the strategies for cost reduction in single-objective feature selection were only partially sustained in multi-objective feature selection.

**Keywords:** feature selection; differential evolution; cost reduction; multi-objective optimization



**Citation:** Barradas-Palmeros, J.-A.; Mezura-Montes, E.; Rivera-López, R.; Acosta-Mesa, H.-G.; Márquez-Grajales, A. Computational Cost Reduction in Multi-Objective Feature Selection Using Permutational-Based Differential Evolution. *Math. Comput. Appl.* **2024**, *29*, 56. <https://doi.org/10.3390/mca29040056>

Academic Editor: Oliver Schütze

Received: 3 June 2024

Revised: 7 July 2024

Accepted: 9 July 2024

Published: 13 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Feature selection (FS) is a dimensionality reduction preprocessing task in machine learning (ML) that deals with selecting the most relevant features in a dataset and discarding the noisy and irrelevant ones. By preferring only the features with meaningful information, the learning process complexity is reduced, and an ML algorithm's performance in classification or clustering tasks is expected to be improved [1]. FS is an NP-Hard combinatorial problem with an exponentially growing search space of  $2^n$  possible solutions for a dataset with  $n$  features. Given the complexity of the problem, nature-inspired metaheuristics, including Evolutionary Computation (EC) algorithms, constitute popular and effective methods for FS [2].

Dimensionality reduction methods have gained attention given the growing volumes of data with high dimensions generated nowadays in applications from different fields. By reducing the data dimensions, the ML algorithm training process is expected to be more efficient and fruitful [3,4]. An alternative feature reduction approach is feature extraction or feature reduction. It uses an algorithm to generate new features that combine and represent the information from the dataset features. In contrast, FS consists of selecting some of the

original features in the dataset. An advantage of FS is that the interpretability of the data is not affected [5,6]. This work focuses on FS.

As presented in [3,7,8], the FS mechanism of assessing and comparing the goodness of a subset of features allows the classification of FS approaches into filter, wrapper, and embedded. Filters are the fastest approach, using an evaluation metric independent of the classification or clustering algorithm. Filters usually focus on the relevance of each feature or maximize the interaction between selected features. Wrapper approaches use an ML algorithm to determine the quality of a feature subset. The features that increase the performance of the ML algorithm are expected to be selected. Wrappers are the most resource-demanding approach, given the required several runs of the ML algorithm. Nonetheless, wrappers generally obtain the highest performance. Finally, embedded approaches incorporate the FS process in the training process of the ML algorithm. Decision trees are an example of an embedded approach that selects the most relevant features for classification in the training process. Embedded approaches are expected to be more complex than filters and faster than wrappers.

Differential Evolution (DE) is an evolutionary algorithm (EA) that is highlighted for its simplicity and robustness when applied to optimization problems [9]. As presented in [10], various works have extended the DE algorithm's classic single-objective form to a multi-objective one. For the FS problem, various multi-objective DE proposals can be found in the literature using two common types of solution representation: a real-value codification with a threshold to determine if a feature is selected and binary adaptations of DE. Binary DE versions require changes in the mutation operator of DE, whereas using a real-value representation with a threshold allows the application of basic operators without changes. Examples of multi-objective DE for feature selection are found in [11,12] for filters with real-value codification, in [13,14] for wrappers with real-value codification, and in [15,16] for wrappers with binary representation. Alternative multi-objective DE approaches for FS can be found in [17] for unsupervised FS and [18] for FS using reinforcement learning.

An alternative and effective solution representation for DE applied for the FS problem is presented in [19] where the permutational-based DE algorithm for FS (DE-FS<sup>PM</sup>) is proposed. Permutations represent the individuals, and the mutation operator is modified to work with the selected representation. The reported results from the DE-FS<sup>PM</sup> show that it outperformed other metaheuristic and classic approaches for FS. In this manner, the effectiveness of using the DE adaptation to the permutational space was proved. In [20], the DE-FS<sup>PM</sup> algorithm is extended to a multi-objective version by combining it with the Generalized Differential Evolution 3 (GDE3) algorithm. The results show that the proposal effectively found a set of solutions that represent a trade-off between the objectives of minimizing the prediction error of a classifier and the number of selected features. Nonetheless, the single objective version of the DE-FS<sup>PM</sup> algorithm finds subsets with lower classification errors.

Cost-reduction approaches for the FS process are proposed in [21–23], where the search process for the most relevant subset of features is conducted using a fraction of the dataset instances, selected with random sampling, to reduce the evaluation cost. In [21], using the randomly selected group of dataset instances is tested in filter approaches for FS and feature extraction. The authors define a fixed number of instances to be sampled (100, 250, 500, 1500, and 2000) and select a subset of features with a predefined size of ten. Six methods are used with large-scale datasets, resulting in an execution time reduction when using the reduced subset of instances with minimal impact on the performance of the feature reduction method.

In [22], the approach of using random sampling for reducing the number of instances in the search process for feature selection, and therefore the computational cost, is extended for wrapper approaches for FS. The authors proposed three different random sampling-based strategies to reduce the number of dataset instances in the search process: the fixed, the incremental, and the evolving sampling fraction. In addition, the Success History parameter adaptation for DE from [24] was adapted to the FS problem with the DE-FS<sup>PM</sup>

algorithm. The results were promising in maintaining the performance of the FS procedure, but the resource savings were scarce in computational time.

An extension of the work from [22] is presented in [23]. A memory mechanism to avoid repeated evaluations is added to work with the fixed and incremental sampling fraction strategies applied to the DE-FS<sup>PM</sup> algorithm. The memory attempts to solve the problem of wasted resources associated with evaluating duplicated individuals in DE presented in [25]. The method could perform similarly to the DE-FS<sup>PM</sup> original algorithm using the fixed sampling fraction strategy with memory to avoid repeated evaluations. The proposed approach reduced the average of 35.15% of the computational time and detected that an average of 35.35% of the evaluations could be avoided.

This work is based on the future work stated in [22,23], where the resource-saving mechanisms of the sampling strategies and memory to avoid repeated evaluations were proposed. The main contributions of this paper are the following:

- We incorporate sampling strategies and memory to avoid repeated evaluations into the GDE3-based multi-objective version of the DE-FS<sup>PM</sup> algorithm.
- We introduce two novel proposals: GDE3fix and GDE3inc. The former utilizes the fixed sampling fraction strategy, while the latter incorporates the incremental sampling fraction strategy. Both proposals use memory to avoid repeated evaluations.
- We test the robustness of the cost-reduction mechanisms in multi-objective FS. The main goal was to determine if the results from the single-objective approach, where the computational cost was reduced without diminishing algorithm performance, are maintained in the multi-objective version of the algorithm.
- We thoroughly analyze the effects of the proposals in terms of computational time consumption, the number of evaluations performed by the algorithm, and the number of instances used for an evaluation. Additionally, future work is described, including possible areas for improvement.

The rest of this document is organized into four sections. In Section 2, the details of the multi-objective FS process are introduced. After that, the DE algorithm is described along with its adaptation to the permutational codification of solutions and its extension to multi-objective optimization. Additionally, the cost-reduction mechanisms are presented in detail. Section 3 presents the experimentation details and results. Finally, Sections 4 and 5 include the analysis of the results, the conclusions, and future research directions.

## 2. Materials and Methods

### 2.1. Multi-Objective Feature Selection

As presented in [6], the FS process is guided by two main objectives: maximizing the classification accuracy and minimizing the number of selected features. These objectives conflict and different feature subsets can represent trade-offs between obtaining a higher accuracy performance or a smaller subset of features. Usually, the classification accuracy maximization is transformed into the minimization of the classification error. This way, the goal is to minimize both objectives. Given that there are no constraints to the FS problem, in [26], the multi-objective FS problem is modeled following Equation (1).  $x$  represents a subset of selected features.

$$\text{minimize } F(x) = [f_1(x), f_2(x)] \quad (1)$$

$f_1(x)$  represents the classification error and is calculated using Equation (2). In the equation, the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values from the confusion matrix are required. A cross-validation (CV) approach is followed in our proposal to calculate the error rate.  $f_2(x)$  corresponds to the number of selected features as presented in Equation (3).  $d$  is the number of selected features, and  $n$  is the total number of features in the dataset.  $f_1(x)$  and  $f_2(x)$  are in the range of [0, 1].

$$f_1(x) = \frac{FP + FN}{TP + FP + TN + FN} \quad (2)$$

$$f_2(x) = \frac{d}{n} \quad (3)$$

As presented in [27], a multi-objective optimization algorithm returns a set of solutions instead of a single solution as in single-objective optimization. The concept of Pareto optimal solutions is presented in multi-objective optimization. A solution is Pareto optimal if it is not dominated by any other solution. The weak dominance ( $\preceq$ ) is defined in Equation (4), where a solution  $x_1$  weakly dominates  $x_2$  ( $x_1 \preceq x_2$ ) if, and only if, all its objective values are less than or equal to the objective values of  $x_2$ . The dominance ( $\prec$ ) of a solution  $x_1$  over  $x_2$  ( $x_1 \prec x_2$ ) is given by Equation (5), adding to the weak dominance the condition that at least one objective value  $f_m(x_1)$  must be less than  $f_m(x_2)$ . In the case of the previously stated modeling of the FS problem,  $m = 2$  since the objectives are the classification error and the normalized number of selected features. The Pareto optimal solutions compose the Pareto optimal set. The visualization of the solutions in the Pareto optimal set is the Pareto front.

$$x_1 \preceq x_2 \text{ iff } \forall m \in \{1, 2, \dots, M\} : f_m(x_1) \leq f_m(x_2) \quad (4)$$

$$x_1 \prec x_2 \text{ iff } x_1 \preceq x_2 \wedge \exists m \in \{1, 2, \dots, M\} : f_m(x_1) < f_m(x_2) \quad (5)$$

## 2.2. Differential Evolution

DE is a population-based metaheuristic algorithm for continuous optimization proposed in [28].  $N$  individuals constitute the population. A vector of the size of the dimensions of the problem represents an individual ( $x_i$ ) in the population. The initial population consists of  $N$  individuals randomly created. Each individual has an associated fitness value provided by an evaluation using the selected fitness function that will guide the search process. The population is evolved in an iterative process where in each generation  $g$ , the mutation and crossing procedures are applied for each target vector  $x_i$  to generate the noise ( $v_i$ ) and trial ( $u_i$ ) vectors. A user-defined parameter is the maximum number of generations  $G$  to evolve the population.

As presented in [29], the basic DE version is called *DE/rand/1/bin*. *rand* and 1 come from calculating  $v_i$ , where Equation (6) is applied. *rand* refers to the random selection of the  $r_0$ ,  $r_1$ , and  $r_2$  individuals from the population different from themselves and the  $x_i$  that is being considered for mutation and crossing. The value 1 is given by the calculation of one vector difference. The scaling factor  $F$  is a scalar value defined by the user. *bin* comes from the binomial distribution associated with the uniform crossover operator presented in Equation (7). The crossover rate  $CR$  is another user-defined parameter that controls the probability of choosing  $u_{i,j}$  from  $v_i$  or  $x_i$ . To ensure that  $u_i$  is not copied from  $x_i$ , a random position  $J_{rand}$  is chosen, where  $u_{i,j}$  is guaranteed to be  $v_{i,j}$ . Alternative DE variants have been proposed in the literature, changing the calculation of  $v_i$  [9]. Some examples include *DE/best/1/*, *DE/rand/2/*, and *DE/current-to-best/*. An alternative crossover mechanism is the exponential crossover (*exp*).

$$v_i = r_0 + F(r_1 - r_2) \quad (6)$$

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } (rand_j \leq CR) \text{ or } (j = J_{rand}); j = 1, \dots, |x_i| \\ x_{i,j} & \text{otherwise} \end{cases} \quad (7)$$

After  $u_i$  is computed and evaluated, a binary tournament is performed to determine the individual with better fitness value comparing  $x_i$  and  $u_i$ . The winner is included in the population for the next generation of the evolutionary process. The previous process is an elitist selection mechanism that guarantees that the best solution found in the search process is never discarded. Once the  $G$  generations are computed, the process ends, and the individual with the highest fitness value in the population is returned as the best solution for the problem.

### 2.3. Permutational-Based Differential Evolution for Feature Selection

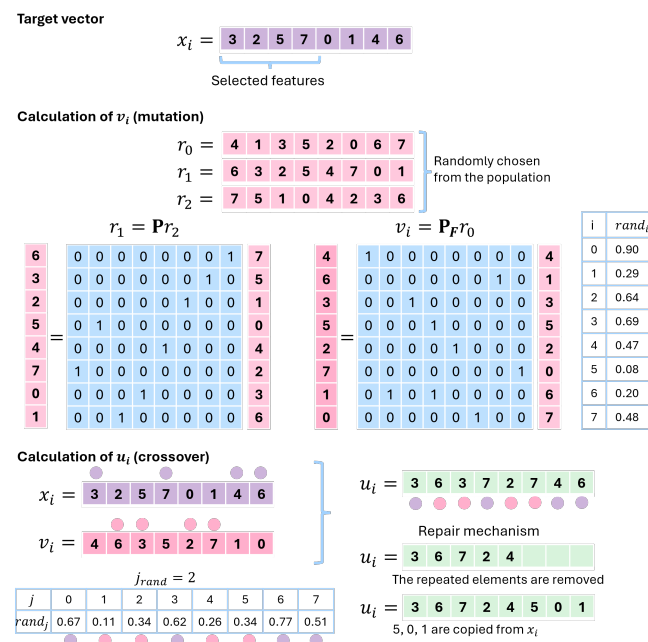
In [19], the DE-FS<sup>PM</sup> algorithm is proposed, modifying the DE algorithm to be applied to the FS problem. The first modification is that each individual is represented by a permutation with the indexes of the features in the dataset and a number zero used in the decoding process. The indexes that appear before the zero in the permutation are considered the selected features by the individual. Given the alternative encoding of the individuals, the DE procedure is adapted to the permutational space. The main changes are applied to the  $v_i$  calculation. Equations (8) and (9) are used instead of Equation (6).

$$r_1 \leftarrow \mathbf{P}r_2 \quad (8)$$

$$v_i \leftarrow \mathbf{P}_F r_0 \quad (9)$$

Equation (8) presents the calculation of the permutation matrix  $\mathbf{P}$  that maps  $r_1$  and  $r_2$ . After calculating  $\mathbf{P}$ , a scaled permutation matrix  $\mathbf{P}_F$  is required.  $\mathbf{P}_F$  is used to apply some changes to  $r_0$  in Equation (9); the parameter  $F$  controls how disruptive the mutation is. The process of calculating  $\mathbf{P}_F$  is presented in [30]. For each row  $i$  in  $\mathbf{P}$ , if there is a 0 in the position  $\mathbf{P}[i, i]$  and a random number  $rand_i$  is greater than  $F$ , the row  $i$  is swapped with the row  $j$  where the position  $\mathbf{P}[j, i]$  is a 1. When the swap is produced, a number 1 is included in the diagonal of the matrix. The positions in the diagonal of  $\mathbf{P}_F$  with a value of 1 represent no changes to  $r_0$  when Equation (9) is applied. Greater  $F$  values will produce little change to  $\mathbf{P}$  due to the low probability of  $rand_i$  being greater than  $F$ . If  $F = 1$ , no changes will be applied to  $\mathbf{P}$ . By contrast, smaller  $F$  values will allow the setting of more elements in the  $\mathbf{P}_F$  diagonal as 1. If  $F = 0$ , the resulting  $\mathbf{P}_F$  will be the diagonal matrix and  $v_i = r_0$ .

The uniform crossover operator of DE presented in Equation (7) is maintained in the DE-FS<sup>PM</sup> algorithm. This way, some elements in the permutation of  $u_i$  come from  $v_i$  and others from  $x_i$ . As a result,  $u_i$  could possess repeated elements requiring the application of a repair mechanism. The repair mechanism proposed in [19] removes all the repeated elements in  $u_i$ ; the remaining elements are moved to the left to take the empty spaces left by the removed elements. Finally, the permutation is completed with the missing elements in the order they appear in  $x_i$ . The mutation and crossover process of the DE-FS<sup>PM</sup> algorithm is illustrated with an example in Figure 1.



**Figure 1.** An example of the DE-FS<sup>PM</sup> algorithm mutation and crossover procedures. The decoding of an individual  $x_i$  is presented with an example of calculating  $u_i$  and applying the repair mechanism to maintain a valid permutation. The random values required by the process are also shown. The parameters



considered for the example are  $F = 0.6$  and  $CR = 0.4$ . After calculating  $u_i$ , it is evaluated, and its associated fitness value is compared with the  $x_i$  fitness value. The most fitted one is included in the next-generation population. Different colors are used in the figure to represent the  $x_i$ ,  $v_i$ , and  $u_i$  vectors.

The accuracy metric guides the FS process in the DE-FS<sup>PM</sup> algorithm. A five-fold stratified cross-validation (CV) evaluation using the k-nearest-neighbors (KNN) algorithm is used as the fitness function. To determine the  $k$  value for the KNN algorithm, at the beginning of the search process, different values of  $k$  from 1 to 20 with a step of 2 are tested using all the dataset features, selecting the  $k$  with the highest performance in a ten-fold CV. After selecting  $k$ , the dataset is evaluated using all features with a ten-fold stratified CV. The ten-fold stratified CV evaluation is conducted again at the end of the search process with the selected subset of features.

The DE-FS<sup>PM</sup> algorithm requires a preprocessing step for the datasets used for the feature selection process. First, the missing values are imputed, considering the mean for numerical features and the mode for categorical features. The next step is converting the categorical features into numerical features. Finally, all features are normalized following a min-max normalization.

#### 2.4. Generalized Differential Evolution 3 Algorithm

The GDE3 algorithm was proposed in [31], extending the capabilities of the DE algorithm to deal with problems with  $M$  objectives and  $K$  constraints. As presented in [10], the changes to the DE algorithm are applied in the selection procedure. Earlier GDE versions considered the Pareto dominance and crowding distance measurement when comparing  $x_i$  and  $u_i$ . GDE3's population can grow temporally when specific criteria are met (as explained later), keeping both  $x_i$  and  $u_i$ . At the end of each generation, if the population grew, the Fast-Non-Dominated Sort and Crowding Distance (CD) assignment algorithms from [32] are applied to reduce the population to the size  $N$ . The Fast-Non-Dominated Sort assigns the non-dominated solutions to the front number 1. Front number 2 comprises the non-dominated solutions from the rest, and so on until all solutions are included in a front. The fronts formed by the algorithm are progressively added to the population for the next generation until one front does not fit entirely due to a surpass of  $N$ . Then, CD is applied to rank the individuals in the front. The ones with the highest CD value are selected.

In [27], it is explained that the changes applied to the DE algorithm in the generalized versions were intended to be as little as possible. As mentioned earlier, the modification is presented in the selection mechanism, and the rest is the same as the  $DE/rand/1/bin$  procedure. The GDE3 algorithm is equivalent to the DE original procedure when applied to problems with  $M = 1$  and  $K = 0$ , i.e., problems with one objective and no constraints. To deal with constraint satisfaction problems ( $K > 0$ ), the selection mechanism of GDE3 is based on the concept of constraint-domination ( $\prec_c$ ), a variant of the domination ( $\prec$ ) concept presented in Section 2.1.  $x_1 \prec_c x_2$  when:

- $x_1$  and  $x_2$  are unfeasible, but  $x_1$  violates the constraints less.
- $x_1$  is feasible and  $x_2$  is unfeasible.
- $x_1$  and  $x_2$  are feasible and  $x_1 \prec x_2$ .

The selection mechanism defined in [31] for the GDE3 algorithm consists of the following three cases:

1. When  $x_i$  and  $u_i$  are infeasible. The one that violates the least number of constraints is selected.
2. If  $x_i$  is feasible and  $u_i$  is unfeasible,  $x_i$  is selected. On the other hand, if  $u_i$  is feasible and  $x_i$  is unfeasible,  $u_i$  is selected.
3. When  $x_i$  and  $u_i$  are feasible,  $u_i$  is selected if  $u_i \preceq x_i$ ;  $x_i$  is selected if  $x_i \prec u_i$ ; and both  $x_i$  and  $u_i$  are selected if  $x_i \not\prec u_i$  and  $u_i \not\prec x_i$ .

The third case is the only one in which the GDE3 algorithm allows population growth, given that both  $x_i$  and  $u_i$  can be selected. Since the FS problem presents no constraints ( $k = 0$ ), only the third case in the GDE3 selection mechanism is applied. In [20], the DE- $FS^{PM}$  algorithm is extended to work with two objectives using the optimization framework of the GDE3 algorithm. The objectives considered are the classification error and the number of selected features. The selection mechanism is the only modification to the DE- $FS^{PM}$  algorithm. In our research proposal, the number of selected features is normalized by dividing it by the total number of features in the dataset as presented in Section 2.1. The GDE3-based multi-objective version of the DE- $FS^{PM}$  algorithm is presented in Algorithm 1. As observed in Algorithm 1, the procedure returns the Pareto optimal solutions. Hence, a mechanism must be established to choose one of the returned solutions.

---

**Algorithm 1** The multi-objective DE –  $FS^{PM}$  algorithm

---

**Require:** DE –  $FS^{PM}$  – GDE3 ( $CR, F, N, G$ )

**Input:** The crossover rate ( $CR$ ), the scale factor ( $F$ ), the population size ( $N$ ), and the number of generations ( $G$ ).

**Output:** Pareto optimal solutions in the current population.

```

 $\mathbb{X}_0 \leftarrow \emptyset$ 
for each  $i \in \{1, \dots, N\}$  do
     $x_i \leftarrow$  A permutation chosen at random from the solution space.
     $\mathbb{X}_0 \leftarrow \mathbb{X}_0 \cup \{x_i\}$ 
end for
for each  $g \in \{1, \dots, G\}$  do
     $\mathbb{X}_g \leftarrow \emptyset$ 
     $m \leftarrow 0$ 
    for each  $x_i \in \mathbb{X}_g$  do
         $v_i \leftarrow$  Mutated vector using Equations (8) and (9).
         $u_i \leftarrow$  Trial vector calculated using Equation (7) and the repair procedure.
         $\mathbb{X}_g \leftarrow \mathbb{X}_g \cup \begin{cases} \{u_i\} & \text{if } u_i \prec x_i \\ \{x_i\} & \text{otherwise} \end{cases}$ 
        if  $u_i \not\prec x_i \wedge x_i \not\prec u_i$  then
             $\mathbb{X}_g \leftarrow \mathbb{X}_g \cup \{u_i\}$ 
             $m \leftarrow m + 1$ 
        end if
    end for
    if  $m > 0$  then
         $FR \leftarrow$  fast-non-dominated-sort( $\mathbb{X}_g$ )
         $\mathbb{X}_g \leftarrow \emptyset$ 
         $j \leftarrow 1$ 
        while  $|\mathbb{X}_g| < N$  do
            if  $|\mathbb{X}_g| + |FR_j| \leq N$  then
                 $\mathbb{X}_g \leftarrow \mathbb{X}_g \cup FR_j$ 
            else
                 $CD_j \leftarrow$  Crowding-distance( $FR_j$ )
                 $sort(FR_j, CD_j)$ 
                 $\mathbb{X}_g \leftarrow \mathbb{X}_g \cup FR_j[0 : N - |\mathbb{X}_g|]$ 
                end if
             $j \leftarrow j + 1$ 
        end while
    end if
    end for
     $FR \leftarrow$  fast-non-dominated-sort( $\mathbb{X}_g$ )
    return  $FR_1$ 

```

▷ Sort  $FR_j$  in descending  $CD_j$  order

---

## 2.5. Cost-Reduction Mechanisms for Feature Selection

The considered mechanisms for computational cost reduction are the fixed and incremental sampling fraction strategies from [22] and their incorporation with memory to avoid repeated evaluations from [23]. Both proposals were applied to the DE-FS<sup>PM</sup> algorithm that considers only one objective in its optimization process. Nonetheless, the mechanisms can also be applied to multi-objective optimization processes. The incorporation of both mechanisms into the multi-objective feature selection process is described next.

### 2.5.1. Fixed Sampling Fraction with Memory

The FS search process is conducted with a fraction of the dataset instances in the fixed sampling fraction strategy. The user defines two parameters: the initial sampling fraction  $S$  and the number of blocks in the search  $B$ .  $S$  is used at the beginning of the search process to apply random sampling and select part of the dataset instances. At that step, the memory used to avoid repeated evaluations is empty. When an individual is decoded for evaluation, the procedure searches in memory if there is a stored fitness value associated with the subset of features represented by the individual. The stored fitness value is returned if a coincidence is found in memory. Otherwise, the evaluation is performed using the fitness function. In the case of a multi-objective process, the memory stores the value obtained from the assessment of the individual in each of the considered objectives.

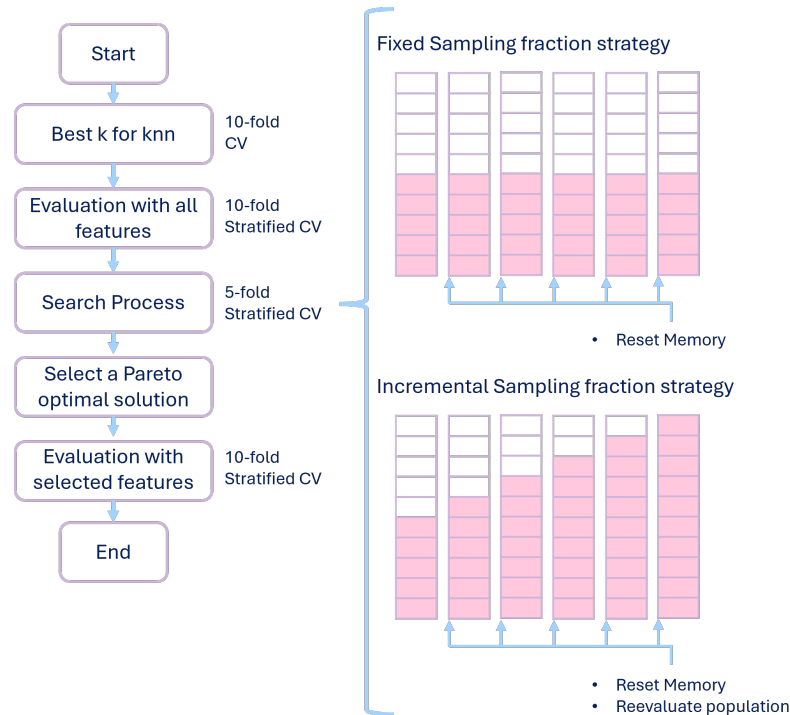
The parameter  $B$  divides the search process's generations into groups. In the fixed sampling fraction, the memory is reset at the beginning of each block. The previous block is used to control the size of the memory. For example, if  $G$  is set as 100,  $B$  is set as five. The memory is reset at generations 1, 21, 41, 61, and 81. The memory mechanism is expected to have a more considerable impact on the last blocks of the search process due to the algorithm's convergence, where population diversity diminishes and the probability of finding a duplicate individual increases.

### 2.5.2. Incremental Sampling Fraction with Memory

When the incremental sampling fraction strategy is applied, the process has the same start conditions as the fixed sampling fraction strategy. The user must also define the parameters  $S$  and  $B$ . Random sampling is applied at the beginning of the process, considering  $S$  to select part of the dataset instances. The memory mechanism is also used to avoid repeated evaluations. The characteristic aspect of the incremental sampling fraction strategy is that, at the beginning of each block of generations, not only is the memory reset, but more instances are proportionally added to the search process. In the last block of generations, all dataset instances are used. This way, fewer instances are considered in early generations, expecting that the algorithm will find promising areas of the search space with less costly evaluations. Then, late generations will likely be able to consider feature subsets with better generalization capabilities and avoid overfitting the selected feature subset to a fraction of dataset instances.

Given that the conditions of the FS problem change when more instances are considered, the population is reevaluated at the beginning of each block. The previous point is a drawback of the strategy due to the extra evaluations. Figure 2 presents a diagram of the multi-objective version of the DE-FS<sup>PM</sup> algorithm with the resource-saving modifications in the search process. It is seen that the incremental sampling fraction strategy requires the reevaluation process when a new block starts, while the fixed sampling fraction only requires resetting the memory.





**Figure 2.** Diagram of the multi-objective DE-FS<sup>PM</sup> algorithm with the cost-reduction mechanisms applied in the search process.  $S$  and  $B$  are considered 0.5 and 6, respectively, as examples of their effect on managing the dataset instances in the search process.

### 3. Results

To test the effectiveness of the cost-reduction mechanisms from Section 2.5, the experimentation approach from [23] is adopted. The algorithm proposals are run 30 times for each of the eighteen datasets selected for experimentation. The code was implemented using Python, and the experiments were run in the virtual environments provided by Google Colab Pro. The datasets are selected from the UCI Machine Learning Repository [33], and their details are presented in Table 1.

**Table 1.** Details of the datasets selected for experimentation.

| Dataset     | Features | Instances | Classes |
|-------------|----------|-----------|---------|
| Arrhythmia  | 279      | 452       | 16      |
| Audiology   | 69       | 226       | 24      |
| Australian  | 14       | 690       | 2       |
| Cylinder-b  | 39       | 540       | 2       |
| CRX         | 15       | 690       | 2       |
| Dermatology | 34       | 366       | 6       |
| German-c    | 20       | 1000      | 2       |
| Hill valley | 100      | 1212      | 2       |
| Ionosphere  | 34       | 351       | 2       |
| M-libras    | 90       | 360       | 15      |
| Musk 1      | 168      | 476       | 2       |
| Parkinsons  | 22       | 195       | 2       |
| Sonar       | 60       | 208       | 2       |
| Soybean     | 35       | 683       | 19      |
| SPECTF      | 44       | 267       | 2       |
| Vehicle     | 18       | 846       | 4       |
| Vote        | 16       | 435       | 2       |
| WDBC        | 30       | 569       | 2       |

The proposals are identified as:

1. **GDE3**: for the GDE3 version of the DE-FS<sup>PM</sup> algorithm.
2. **GDE3fix**: for the GDE3 version of the DE-FS<sup>PM</sup> algorithm using the fixed sampling fraction strategy with memory to avoid repeated evaluations.
3. **GDE3inc**: for the GDE3 version of the DE-FS<sup>PM</sup> algorithm using the incremental sampling fraction strategy with memory to avoid repeated evaluations.

The parameter configuration is adopted from [19], where the parameters  $G$  and  $N$  are defined for the DE-FS<sup>PM</sup> algorithm.  $F$  and  $CR$  are obtained in the fine-tuning of the GDE3 version of the DE-FS<sup>PM</sup> algorithm from [20]. Finally, the parameters  $S$  and  $B$  for the cost-reduction mechanisms are presented in [23]. The details are provided next.

- $G$ : 200 as the maximum number of generations.
- $N$ : 5 times the number of features in the dataset. The value is bound to have at least 200 individuals and at most 450.
- $F$ : 0.8305.
- $CR$ : 0.7049.
- $S$ : 0.6 for GDE3fix and GDE3inc.
- $B$ : 10 for GDE3fix and GDE3inc.

Three criteria were considered when selecting a solution from the resulting Pareto front of each method. First, the solution with the best performance in  $f_1(x)$  represents the lowest classification error obtained by a solution. After that, the solution in the knee point of the front is considered a middle point in the trade-off between  $f_1(x)$  and  $f_2(x)$ . The knee point is calculated as the solution in the Pareto front with the smallest Euclidian distance concerning the point (0,0). Finally, the solution with better performance minimizing  $f_2(x)$  is selected.

Three key aspects were monitored to assess the computational resources saved by the cost-reduction mechanism: the execution time, the number of evaluations used by the method, and the number of instances avoided in evaluation. The execution time is the time required on average by the FS procedure in the runs of each one of the proposals. The number of evaluations refers to the times an individual is evaluated with  $f_1(x)$  and  $f_2(x)$ . In the GDE3fix and GDE3inc proposals, the memory to avoid repeated evaluations is used to reduce the number of evaluations. Finally, the effect of using the fixed and incremental sampling strategies and the memory mechanism is measured with the sum of the number of instances used while evaluating an individual. Additionally, the hypervolume and spacing indicators for multi-objective optimization are reported.

The results are divided into three subsections of the document. First, in Section 3.1, the proposals are compared in terms of accuracy and the number of selected features. Then, in Section 3.2, the resource savings achieved by the GDE3fix and GDE3inc proposals are presented. Finally, Section 3.3 shows the performance of the proposals measured by the selected multi-objective indicators.

### 3.1. Accuracy Performance and Number of Selected Features

Table 2 presents the average results of the proposals when the solution selected from the Pareto front is the one with the minimum classification error. For this selection, the priority is given to  $f_1(x)$ . For comparison, the table includes the accuracy and the number of selected features performance of the data without FS, and the results reported in [23] for the DE-FS<sup>PM</sup> algorithm. It is seen that the single-objective version of the DE-FS<sup>PM</sup> algorithm achieves a higher accuracy performance. Nonetheless, in almost all cases, the GDE3-based proposals attained the goal of reducing the data dimensionality and increasing the accuracy of using the dataset without feature selection.

The next comparison point is when the solution is selected from the Pareto front in the knee point. These solutions represent the trade-off between the accuracy performance and the number of selected features. Table 3 presents the results. In this case, the accuracy performance of the dataset without FS is presented as a reference. In all cases, the procedure

reduced the dimensionality of the data. However, in most cases, the accuracy results are poorer than the accuracy performance achieved by the method without FS.

**Table 2.** Accuracy and feature selection results of selecting the solution with the best performance in accuracy from the Pareto front. The best result is marked in bold, and the best result considering only the GDE3, GDE3fix, and GDE3inc proposals is underlined.

| Dataset     | WoFS |       | DE-FS(PM)   |              | GDE3        |              | GDE3fix     |       | GDE3inc    |       |
|-------------|------|-------|-------------|--------------|-------------|--------------|-------------|-------|------------|-------|
|             | SF   | Acc   | SF          | Acc          | SF          | Acc          | SF          | Acc   | SF         | Acc   |
| Arrhythmia  | 279  | 58.79 | 10.9        | <b>75.66</b> | <b>10.5</b> | 70.73        | 16.0        | 66.37 | 12.1       | 66.75 |
| Audiology   | 69   | 76.47 | <b>25.1</b> | <b>85.63</b> | <u>42.9</u> | <u>82.43</u> | 43.7        | 79.72 | 44.9       | 81.00 |
| Australian  | 14   | 85.98 | <b>6.7</b>  | <b>86.82</b> | <u>9.5</u>  | <u>86.48</u> | <b>6.7</b>  | 86.29 | 7.6        | 86.47 |
| Cylinder-b  | 39   | 74.51 | 3.9         | <b>84.19</b> | <b>2.2</b>  | <u>83.51</u> | 10.2        | 80.97 | 2.3        | 82.65 |
| CRX         | 15   | 86.13 | 9.0         | <b>87.49</b> | <u>10.8</u> | <u>87.17</u> | <b>8.0</b>  | 86.16 | 9.0        | 86.94 |
| Dermatology | 34   | 96.89 | <b>19.1</b> | <b>98.35</b> | 25.7        | <u>98.02</u> | <u>23.9</u> | 97.17 | 26.6       | 97.65 |
| German-c    | 20   | 75.35 | <b>9.1</b>  | <b>77.53</b> | 15.4        | <u>76.14</u> | <u>12.6</u> | 74.34 | 14.8       | 75.65 |
| Hill valley | 100  | 64.54 | <b>9.9</b>  | <b>71.64</b> | 10.5        | <u>69.52</u> | <u>14.9</u> | 67.24 | 16.6       | 68.48 |
| Ionosphere  | 34   | 86.97 | 6.5         | <b>94.82</b> | <u>6.6</u>  | <u>93.45</u> | <b>5.7</b>  | 90.52 | 8.0        | 92.13 |
| M-libras    | 90   | 86.06 | <b>24.0</b> | <b>88.89</b> | 24.9        | <u>87.86</u> | <u>39.5</u> | 86.23 | 47.3       | 87.19 |
| Musk 1      | 168  | 85.65 | <b>36.1</b> | <b>94.23</b> | <u>42.4</u> | <u>91.23</u> | 52.5        | 87.98 | 60.5       | 89.45 |
| Parkinsons  | 22   | 95.84 | 10.9        | <b>98.84</b> | <b>9.6</b>  | <u>98.24</u> | 13.3        | 95.44 | 12.8       | 97.39 |
| Sonar       | 60   | 86.70 | <b>22.4</b> | <b>93.79</b> | 25.2        | <u>90.56</u> | 25.1        | 87.00 | 34.3       | 89.03 |
| Soybean     | 35   | 91.85 | <b>17.3</b> | <b>94.91</b> | <u>21.3</u> | <u>94.30</u> | <u>23.3</u> | 92.69 | 22.6       | 93.61 |
| SPECTF      | 44   | 77.68 | <b>5.5</b>  | <b>84.22</b> | <u>6.9</u>  | <u>83.00</u> | 15.3        | 79.34 | 9.2        | 79.93 |
| Vehicle     | 18   | 70.08 | <b>9.9</b>  | <b>74.31</b> | <u>11.1</u> | <u>73.19</u> | 11.6        | 71.62 | 11.9       | 72.15 |
| Vote        | 16   | 93.50 | 5.2         | <b>96.59</b> | <u>4.7</u>  | <u>96.54</u> | 4.3         | 95.38 | <b>2.5</b> | 95.93 |
| WDBC        | 30   | 97.07 | 18.2        | <b>97.55</b> | 23.5        | <u>97.26</u> | <u>17.6</u> | 96.82 | 22.5       | 97.17 |

**Table 3.** Accuracy and feature selection results of selecting the solution in the knee point from the Pareto front. The best result is marked in bold, and the best result considering only the GDE3, GDE3fix, and GDE3inc proposals is underlined.

| Dataset     | WoFS |       | DE-FS(PM) |              | GDE3       |              | GDE3fix     |              | GDE3inc    |              |
|-------------|------|-------|-----------|--------------|------------|--------------|-------------|--------------|------------|--------------|
|             | SF   | Acc   | SF        | Acc          | SF         | Acc          | SF          | Acc          | SF         | Acc          |
| Arrhythmia  | 279  | 58.79 | 10.9      | <b>75.66</b> | 9.2        | <u>70.54</u> | 11.0        | 66.12        | <b>7.7</b> | 66.81        |
| Audiology   | 69   | 76.47 | 25.1      | <b>85.63</b> | <b>9.8</b> | <u>73.08</u> | 11.4        | 68.08        | 13.0       | 72.95        |
| Australian  | 14   | 85.98 | 6.7       | <b>86.82</b> | <b>1.0</b> | <u>85.51</u> | <b>1.0</b>  | <u>85.51</u> | <b>1.0</b> | 85.42        |
| Cylinder-b  | 39   | 74.51 | 3.9       | <b>84.19</b> | <b>2.0</b> | <u>83.30</u> | 2.2         | 81.69        | 2.1        | 82.41        |
| CRX         | 15   | 86.13 | 9.0       | <b>87.49</b> | <b>1.0</b> | <u>84.57</u> | <b>1.0</b>  | 83.82        | <b>1.0</b> | <u>84.86</u> |
| Dermatology | 34   | 96.89 | 19.1      | <b>98.35</b> | <b>4.4</b> | 90.42        | 4.5         | 87.98        | <u>4.7</u> | <u>90.87</u> |
| German-c    | 20   | 75.35 | 9.1       | <b>77.53</b> | <b>1.8</b> | <u>71.13</u> | 1.9         | 70.84        | 1.9        | 71.69        |
| Hill valley | 100  | 64.54 | 9.9       | <b>71.64</b> | 6.9        | <u>69.46</u> | <b>5.5</b>  | 66.15        | 6.8        | 68.11        |
| Ionosphere  | 34   | 86.97 | 6.5       | <b>94.82</b> | <b>2.2</b> | <u>89.17</u> | 2.4         | 87.33        | 2.7        | 89.20        |
| M-libras    | 90   | 86.06 | 24.0      | <b>88.89</b> | <b>8.5</b> | <u>86.92</u> | 9.6         | 84.59        | 8.8        | 85.44        |
| Musk 1      | 168  | 85.65 | 36.1      | <b>94.23</b> | 12.3       | <u>88.69</u> | <b>11.5</b> | 85.16        | 12.8       | 86.91        |
| Parkinsons  | 22   | 95.84 | 10.9      | <b>98.84</b> | 2.0        | <u>93.48</u> | <b>1.9</b>  | 89.47        | 2.1        | 92.04        |
| Sonar       | 60   | 86.70 | 22.4      | <b>93.79</b> | 5.8        | <u>86.32</u> | <b>5.0</b>  | 79.35        | 5.9        | 83.19        |
| Soybean     | 35   | 91.85 | 17.3      | <b>94.91</b> | <b>5.5</b> | <u>85.94</u> | 5.9         | 83.55        | 5.9        | 83.96        |
| SPECTF      | 44   | 77.68 | 5.5       | <b>84.22</b> | <u>2.3</u> | <u>83.30</u> | 2.3         | 79.76        | <b>2.0</b> | 79.95        |
| Vehicle     | 18   | 70.08 | 9.9       | <b>74.31</b> | <b>2.2</b> | <u>65.88</u> | 2.4         | 65.82        | <u>2.8</u> | <u>66.93</u> |
| Vote        | 16   | 93.50 | 5.2       | <b>96.59</b> | <b>1.0</b> | <u>95.13</u> | <b>1.0</b>  | <u>95.63</u> | <b>1.0</b> | <u>95.63</u> |
| WDBC        | 30   | 97.07 | 18.2      | <b>97.55</b> | 2.0        | <u>95.31</u> | <b>1.8</b>  | 93.86        | 2.0        | 94.79        |

The third selection strategy was to prioritize the dimensional reduction and select the solution in the Pareto front with better performance in  $f_2(x)$ . Table 4 presents the results of

choosing the solution with the smallest feature subset. The accuracy performance of the dataset without FS is presented as a reference. It is seen that, in this case, the dimensionality reduction is maximum. All the proposals found a subset with only one feature, but the accuracy was severely affected. It is worth noticing the cases where only one feature obtains a performance close to the accuracy using all features, such as the arrhythmia, Australian, cylinder-b, CRX, and vote datasets.

**Table 4.** Accuracy and feature selection results of selecting the solution with the best performance in the number of selected features from the Pareto front. The best result is marked in bold, and the best result considering only the GDE3, GDE3fix, and GDE3inc proposals is underlined.

| Dataset     | WoFS |       | DE-FS(PM) |              | GDE3     |              | GDE3fix  |       | GDE3inc  |       |
|-------------|------|-------|-----------|--------------|----------|--------------|----------|-------|----------|-------|
|             | SF   | Acc   | SF        | Acc          | SF       | Acc          | SF       | Acc   | SF       | Acc   |
| Arrhythmia  | 279  | 58.79 | 10.9      | <b>75.66</b> | <u>1</u> | <u>57.64</u> | <u>1</u> | 56.04 | <u>1</u> | 56.42 |
| Audiology   | 69   | 76.47 | 25.1      | <b>85.63</b> | <u>1</u> | <u>29.13</u> | <u>1</u> | 27.68 | <u>1</u> | 27.57 |
| Australian  | 14   | 85.98 | 6.7       | <b>86.82</b> | <u>1</u> | <u>85.51</u> | <u>1</u> | 85.51 | <u>1</u> | 85.29 |
| Cylinder-b  | 39   | 74.51 | 3.9       | <b>84.19</b> | <u>1</u> | <u>74.88</u> | <u>1</u> | 74.56 | <u>1</u> | 74.86 |
| CRX         | 15   | 86.13 | 9.0       | <b>87.49</b> | <u>1</u> | <u>84.56</u> | <u>1</u> | 83.28 | <u>1</u> | 84.78 |
| Dermatology | 34   | 96.89 | 19.1      | <b>98.35</b> | <u>1</u> | <u>47.10</u> | <u>1</u> | 43.27 | <u>1</u> | 46.09 |
| German-c    | 20   | 75.35 | 9.1       | <b>77.53</b> | <u>1</u> | <u>68.08</u> | <u>1</u> | 67.65 | <u>1</u> | 69.60 |
| Hill valley | 100  | 64.54 | 9.9       | <b>71.64</b> | <u>1</u> | <u>55.56</u> | <u>1</u> | 53.45 | <u>1</u> | 54.77 |
| Ionosphere  | 34   | 86.97 | 6.5       | <b>94.82</b> | <u>1</u> | <u>79.17</u> | <u>1</u> | 77.97 | <u>1</u> | 78.77 |
| M-libras    | 90   | 86.06 | 24.0      | <b>88.89</b> | <u>1</u> | <u>25.88</u> | <u>1</u> | 23.49 | <u>1</u> | 25.29 |
| Musk 1      | 168  | 85.65 | 36.1      | <b>94.23</b> | <u>1</u> | <u>63.07</u> | <u>1</u> | 61.74 | <u>1</u> | 62.44 |
| Parkinsons  | 22   | 95.84 | 10.9      | <b>98.84</b> | <u>1</u> | <u>80.61</u> | <u>1</u> | 78.30 | <u>1</u> | 79.46 |
| Sonar       | 60   | 86.70 | 22.4      | <b>93.79</b> | <u>1</u> | <u>67.07</u> | <u>1</u> | 65.22 | <u>1</u> | 67.20 |
| Soybean     | 35   | 91.85 | 17.3      | <b>94.91</b> | <u>1</u> | <u>29.56</u> | <u>1</u> | 30.51 | <u>1</u> | 28.89 |
| SPECTF      | 44   | 77.68 | 5.5       | <b>84.22</b> | <u>1</u> | <u>78.33</u> | <u>1</u> | 78.15 | <u>1</u> | 78.50 |
| Vehicle     | 18   | 70.08 | 9.9       | <b>74.31</b> | <u>1</u> | <u>50.43</u> | <u>1</u> | 48.79 | <u>1</u> | 49.58 |
| Vote        | 16   | 93.50 | 5.2       | <b>96.59</b> | <u>1</u> | <u>95.38</u> | <u>1</u> | 95.41 | <u>1</u> | 95.54 |
| WDBC        | 30   | 97.07 | 18.2      | <b>97.55</b> | <u>1</u> | <u>91.07</u> | <u>1</u> | 90.35 | <u>1</u> | 90.73 |

### 3.2. Computational Cost Reduction

The effects of the cost-reduction mechanisms on the computational time, the number of instances used for evaluation, and the number of evaluations are presented in Tables 5 and 6. Table 5 presents the average execution time of the three proposals with the percentage of reduction obtained using cost-reduction mechanisms. Both cost-reduction proposals could significantly reduce the computational time, but GDE3fix achieved a slightly higher time reduction than GDE3inc.

Table 6 shows that the memory mechanism can avoid around 80% of the evaluations in the search process. In reducing the number of instances used for evaluation, it is seen that GDE3fix achieves a higher reduction than GDE3inc. This behavior is expected given that GDE3inc proportionally incorporates more instances in the search process.

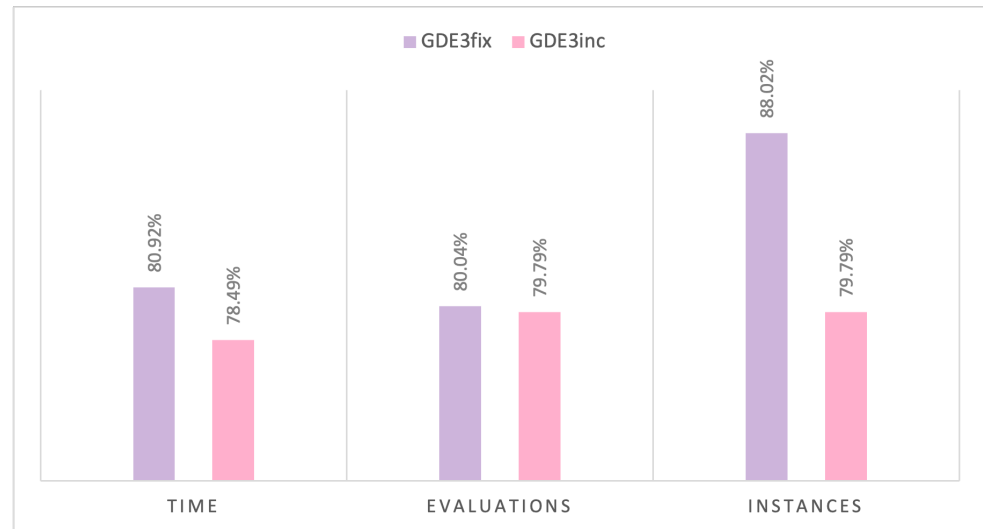
Figure 3 summarizes the effect of incorporating the fixed and incremental sampling fraction strategies with memory to avoid repeated evaluations in the multi-objective version of the DE-FS<sup>PM</sup> algorithm. It is seen that, despite reducing the required evaluations in a similar percentage, the GDE3inc proposal achieved a higher reduction in the computational cost of the FS procedure and in the number of instances used during evaluation. A higher reduction in the number of instances was expected, given that GDE3fix uses the same amount of instances throughout the FS process, while GDE3inc uses more instances in the final generations.

**Table 5.** Time reduction obtained when applying the cost-reduction mechanisms to the multi-objective feature selection process of the DE-FS<sup>PM</sup> algorithm with GDE3. The larger reduction in computational time between the GDE3fix and GDE3inc proposals is marked in bold.

| Dataset     | GDE3       | GDE3fix  |               | GDE3inc  |               |
|-------------|------------|----------|---------------|----------|---------------|
| Arrhythmia  | 8038.508   | 2465.115 | <b>69.33%</b> | 2514.494 | 68.72%        |
| Audiology   | 3991.253   | 1035.456 | <b>74.06%</b> | 1170.440 | 70.67%        |
| Australian  | 3032.478   | 233.618  | 92.30%        | 228.902  | <b>92.45%</b> |
| Cylinder-b  | 2696.103   | 429.128  | 84.08%        | 364.854  | <b>86.47%</b> |
| CRX         | 2939.885   | 287.545  | 90.22%        | 239.618  | <b>91.85%</b> |
| Dermatology | 2561.947   | 544.807  | <b>78.73%</b> | 818.316  | 68.06%        |
| German-c    | 4036.628   | 423.846  | <b>89.50%</b> | 524.601  | 87.00%        |
| Hill valley | 11,660.199 | 2057.723 | <b>82.35%</b> | 2662.471 | 77.17%        |
| Ionosphere  | 2171.877   | 353.161  | <b>83.74%</b> | 372.972  | 82.83%        |
| M-libras    | 5755.865   | 1765.322 | <b>69.33%</b> | 2640.832 | 54.12%        |
| Musk 1      | 7632.818   | 3085.924 | <b>59.57%</b> | 3704.495 | 51.47%        |
| Parkinsons  | 1853.851   | 299.767  | <b>83.83%</b> | 348.367  | 81.21%        |
| Sonar       | 3417.529   | 782.014  | <b>77.12%</b> | 849.622  | 75.14%        |
| Soybean     | 2844.210   | 820.758  | 71.14%        | 700.032  | <b>75.39%</b> |
| SPECTF      | 2627.962   | 315.105  | <b>88.01%</b> | 331.941  | 87.37%        |
| Vehicle     | 3228.412   | 423.969  | 86.87%        | 420.571  | <b>86.97%</b> |
| Vote        | 2278.412   | 173.897  | 92.37%        | 164.088  | <b>92.80%</b> |
| WDBC        | 3075.011   | 492.209  | <b>83.99%</b> | 518.768  | 83.13%        |
| Average     |            |          | <b>80.92%</b> |          | 78.49%        |

**Table 6.** Percentages of avoided evaluations and reduction in the number of instances used for evaluation. The proposal representing more significant savings in each case is marked in bold.

| Dataset     | Evaluations   |               | Instances     |         |
|-------------|---------------|---------------|---------------|---------|
|             | GDE3fix       | GDE3inc       | GDE3fix       | GDE3inc |
| Arrhythmia  | 73.05%        | <b>75.94%</b> | <b>83.84%</b> | 75.94%  |
| Audiology   | <b>77.64%</b> | 74.12%        | <b>86.59%</b> | 74.12%  |
| Australian  | 90.91%        | <b>93.56%</b> | <b>94.55%</b> | 93.56%  |
| Cylinder-b  | 83.24%        | <b>88.07%</b> | <b>89.94%</b> | 88.07%  |
| CRX         | 89.43%        | <b>93.00%</b> | <b>93.66%</b> | 93.00%  |
| Dermatology | <b>75.43%</b> | 68.12%        | <b>85.23%</b> | 68.12%  |
| German-c    | <b>86.74%</b> | 86.16%        | <b>92.04%</b> | 86.16%  |
| Hill valley | <b>77.46%</b> | 74.32%        | <b>86.48%</b> | 74.32%  |
| Ionosphere  | 84.12%        | <b>84.37%</b> | <b>90.46%</b> | 84.37%  |
| M-libras    | <b>69.98%</b> | 57.58%        | <b>81.99%</b> | 57.58%  |
| Musk 1      | <b>51.83%</b> | 51.52%        | <b>71.06%</b> | 51.52%  |
| Parkinsons  | <b>84.35%</b> | 84.11%        | <b>90.61%</b> | 84.11%  |
| Sonar       | 77.29%        | <b>78.07%</b> | <b>86.36%</b> | 78.07%  |
| Soybean     | 71.12%        | <b>73.27%</b> | <b>82.66%</b> | 73.27%  |
| SPECTF      | 89.09%        | <b>89.84%</b> | <b>93.46%</b> | 89.84%  |
| Vehicle     | 85.66%        | <b>86.55%</b> | <b>91.39%</b> | 86.55%  |
| Vote        | 92.38%        | <b>95.01%</b> | <b>95.43%</b> | 95.01%  |
| WDBC        | 81.10%        | <b>82.55%</b> | <b>88.67%</b> | 82.55%  |
| Average     | <b>80.04%</b> | 79.79%        | <b>88.02%</b> | 79.79%  |



**Figure 3.** Resource consumption reduction achieved by the proposals GDE3fix and GDE3inc. The plot presents the average reductions concerning the GDE3 procedure in terms of computational time, the number of evaluations, and the number of instances used when evaluating the individuals.

### 3.3. Multi-Objective Optimization Indicators

Finally, Table 7 presents the performance of the proposals in the hypervolume and spacing indicators for multi-objective optimization. In most cases, the GDE3 proposal achieved a larger hypervolume value, but the GDE3fix and GDE3 proposals presented a better spacing value. The values from the spacing are expected to be small in this problem due to the discrete nature of the Pareto front. The solutions select feature subsets with integer sizes, distributing them in the front.

**Table 7.** Proposal's results for the hypervolume and spacing indicators. The method with the best performance in each case is marked in bold.

| Dataset     | Hypervolume   |               |         | Spacing |               |               |
|-------------|---------------|---------------|---------|---------|---------------|---------------|
|             | GDE3          | GDE3fix       | GDE3inc | GDE3    | GDE3fix       | GDE3inc       |
| Arrhythmia  | <b>0.7093</b> | 0.6816        | 0.6660  | 0.0002  | <b>0.0000</b> | 0.0001        |
| Audiology   | <b>0.7870</b> | 0.7352        | 0.7422  | 0.0027  | <b>0.0003</b> | <b>0.0003</b> |
| Australian  | 0.8102        | <b>0.8141</b> | 0.8061  | 0.0005  | <b>0.0002</b> | 0.0004        |
| Cylinder-b  | <b>0.8348</b> | 0.7809        | 0.7953  | 0.0006  | 0.0008        | <b>0.0000</b> |
| CRX         | 0.8167        | <b>0.8201</b> | 0.8105  | 0.0005  | <b>0.0003</b> | 0.0007        |
| Dermatology | <b>0.9182</b> | 0.9122        | 0.9083  | 0.0030  | 0.0004        | <b>0.0003</b> |
| German-c    | 0.7282        | <b>0.7301</b> | 0.7130  | 0.0161  | <b>0.0004</b> | 0.0005        |
| Hill valley | <b>0.6926</b> | 0.6457        | 0.6712  | 0.0003  | <b>0.0001</b> | <b>0.0001</b> |
| Ionosphere  | <b>0.9086</b> | 0.9028        | 0.8876  | 0.0008  | <b>0.0001</b> | 0.0002        |
| M-libras    | <b>0.8663</b> | 0.8203        | 0.8508  | 0.0053  | <b>0.0004</b> | 0.0005        |
| Musk 1      | <b>0.9044</b> | 0.8822        | 0.8852  | 0.0012  | 0.0008        | <b>0.0004</b> |
| Parkinsons  | <b>0.9366</b> | 0.9208        | 0.9151  | 0.0037  | <b>0.0003</b> | 0.0011        |
| Sonar       | <b>0.8955</b> | 0.8830        | 0.8638  | 0.0019  | <b>0.0002</b> | 0.0004        |
| Soybean     | <b>0.8707</b> | 0.8531        | 0.8501  | 0.0041  | <b>0.0004</b> | <b>0.0004</b> |
| SPECTF      | 0.8337        | <b>0.8401</b> | 0.7936  | 0.0026  | 0.0009        | <b>0.0002</b> |
| Vehicle     | <b>0.6832</b> | 0.6722        | 0.6618  | 0.0033  | 0.0005        | <b>0.0003</b> |
| Vote        | <b>0.9048</b> | 0.9028        | 0.8992  | 0.0001  | 0.0001        | <b>0.0000</b> |
| WDBC        | <b>0.9442</b> | 0.9436        | 0.9385  | 0.0038  | 0.0007        | <b>0.0003</b> |

## 4. Discussion

Statistical tests were conducted as suggested in [34] to evaluate if the resource-saving mechanisms could maintain the performance of the GDE3 algorithm. First, the Friedman



test was applied with the accuracy results of the GDE3, GDE3fix, and GDE3inc proposals from Table 2, obtaining a  $p$ -value of  $1.52 \times 10^{-8}$  that indicates significant differences in the means of the proposals. The post hoc Nemenyi test was then applied to compare the proposals, and the results showed that significant differences were presented in all cases.

An alternative comparison is performed using the hypervolume indicator results to see if the cost-saving mechanisms are affecting the multi-objective search capabilities of the algorithm. The statistical tests were rerun considering the hypervolume values of the GDE3, GDE3fix, and GDE3inc proposals from Table 7. The Friedman test resulted in a  $p$ -value of  $3.84 \times 10^{-5}$ , showing significant differences among the proposals. The Nemenyi post hoc test indicated significant differences for the GDE3 and the GDE3fix proposals and for the GDE3 and the GDE3inc procedures.

In contrast with the findings from [23], incorporating the fixed sampling fraction proposal with memory to avoid repeated evaluations resulted in diminishing the performance of the multi-objective FS method. Nonetheless, the fixed and incremental sampling strategies with memory achieved considerably higher resource savings. The time reduction observed in the single-objective approach was reported as 35.16% and 48.61%, respectively, for the fixed and incremental proposals. Whereas in the multi-objective approach, the time reduction was 80.92% and 78.49%, for the fixed and incremental proposals, respectively. The fixed proposal achieved a higher reduction in this case.

An interesting thing to analyze is the number of evaluations the memory mechanism avoids. The reduction of 80% in the number of evaluations clearly indicates that the adaptation of the DE-FS<sup>PM</sup> algorithm to multi-objective optimization using GDE3 is finding a high number of duplicate individuals. Population diversity and the crowdedness of the solutions are aspects to be considered in future attempts to improve the algorithm's performance.

Another future direction of implementing the cost-reduction mechanisms with the multi-objective version of the DE-FS<sup>PM</sup> algorithm is to find an adequate compromise between performance and resource savings. As seen in [23], the algorithm's performance is not affected, but the reported resource savings are fewer than the ones found in this work despite using the same configuration of the mechanisms. The previous observation suggests that the cost-reduction mechanisms require a more specific parameter configuration for their application in different FS approaches. Additionally, a comparison among the subsets of features selected by each proposal can provide valuable insights into the differences in the results of the FS process. Applying the FS processes in a real-life application, like the one from [35], will result in a deeper comparison of the results when collaborating with an expert in the field.

## 5. Conclusions

In this work, fixed and incremental sampling fraction strategies with memory to avoid repeated evaluations were implemented as cost-reduction mechanisms in a multi-objective permutational-based DE approach for FS. The proposed approach for multi-objective FS presented some limitations. The main one is in the method's performance in the accuracy and multi-objective indicators, which were reduced by the use of cost-reduction mechanisms. The proposals presented limited exploration capabilities, evidenced by the number of evaluations they avoid due to the finding of repeated individuals. The results exhibit high savings of computational resources at the expense of reducing the method's performance. The previous observation shows that the findings in the single-objective version of the algorithm, where the fixed sampling fraction proposal achieved the goal of reducing the computational cost of a wrapper approach for FS without diminishing its performance, is not maintained in a multi-objective FS approach.

The multi-objective approach for FS using the permutational version of GDE3 does not achieve accuracy results as high as the single-objective version of the DE-FS<sup>PM</sup> algorithm. Nevertheless, the advantage of using a multi-objective approach is that more than one solution to the problem is found, allowing the user to choose the most convenient subset of

features among the options in the Pareto front. The GDE3-based proposals found solutions with smaller feature subsets and involved different compromises between the accuracy performance and the number of selected features. The previous results are an advantage of the multi-objective approach. However, it requires an additional step in the process that can be performed with an automatic technique or by following the recommendation of an expert in the data field in the context of using the FS technique in a real-life application.

The parameter configuration used affects the algorithm's search capabilities. In DE, the  $F$  and  $CR$  parameters control the algorithm's exploration/exploitation capabilities. Finding an adequate set of parameters for every dataset in the FS process is a complicated task. A parameter adaptation scheme can help increase the algorithm's search capabilities in future work. Another critical aspect for future improvements is considering mechanisms to improve the population's diversity. Given the high number of repeated individuals found, the procedure appears to suffer from stagnation. Following the previous point, another area to explore in future changes to the algorithm is the mechanism used to reduce the population when it grows with the selection mechanism of the GDE3 algorithm. Finding more effective selection criteria for the FS problem will be helpful.

Future experimentation, in which the resource-saving mechanisms are applied to different single- and multi-objective approaches for FS, will provide more insights into how robust the mechanisms are for computational cost reduction. As presented in this work, finding a balance in the severity of the proposed resource savings is necessary if the goal is to maintain the algorithm's performance. Consequently, additional experimentation focusing on the effect of the parameters  $S$  and  $B$  will be helpful. A further aspect to be considered in future experimentation is the mechanism used to select a solution among the Pareto optimal ones that the algorithm is returning. A more complex selection procedure can provide options for considering a different trade-off between the objectives than just selecting the solution that performs best in one objective or the knee point in the front.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available in the UCI Machine Learning repository at <https://archive.ics.uci.edu/>, reference number [33].

**Acknowledgments:** The first author (CVU 1142850) acknowledges support from the Mexican National Council of Humanities, Science, and Technology (CONAHCYT) with a scholarship to pursue PhD studies at Universidad Veracruzana.

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

## Abbreviations

The following abbreviations are used in this manuscript:

|                     |                                                                            |
|---------------------|----------------------------------------------------------------------------|
| FS                  | Feature Selection                                                          |
| DE                  | Differential Evolution                                                     |
| EA                  | Evolutionary Algorithm                                                     |
| CV                  | Cross-validation                                                           |
| GDE3                | Generalized Differential Evolution 3                                       |
| KNN                 | K-nearest-neighbors                                                        |
| DE-FS <sup>PM</sup> | Permutational-based Differential Evolution Algorithm for Feature Selection |

## References

- Sharma, M.; Kaur, P. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127. [\[CrossRef\]](#)
- Dokeroglu, T.; Deniz, A.; Kiziloğlu, H.E. A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing* **2022**, *494*, 269–296. [\[CrossRef\]](#)
- Abdulwahab, H.M.; Ajitha, S.; Saif, M.A.N. Feature selection techniques in the context of big data: Taxonomy and analysis. *Appl. Intell.* **2022**, *52*, 13568–13613. [\[CrossRef\]](#)
- Brezočnik, L.; Fister, I.; Podgorelec, V. Swarm Intelligence Algorithms for Feature Selection: A Review. *Appl. Sci.* **2018**, *8*, 1521. [\[CrossRef\]](#)
- Agrawal, P.; Abutarboush, H.F.; Ganesh, T.; Mohamed, A.W. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009–2019). *IEEE Access* **2021**, *9*, 26766–26791. [\[CrossRef\]](#)
- Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **2016**, *20*, 606–626. [\[CrossRef\]](#)
- Dhal, P.; Azad, C. A comprehensive survey on feature selection in the various fields of machine learning. *Appl. Intell.* **2022**, *52*, 4543–4581. [\[CrossRef\]](#)
- Theng, D.; Bhoyar, K.K. Feature selection techniques for machine learning: A survey of more than two decades of research. *Knowl. Inf. Syst.* **2024**, *66*, 1575–1637. [\[CrossRef\]](#)
- 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. *Alex. Eng. J.* **2022**, *61*, 3831–3872. [\[CrossRef\]](#)
- Mezura-Montes, E.; Reyes-Sierra, M.; Coello, C.A.C. Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art. In *Advances in Differential Evolution*; Chakraborty, U.K., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 173–196. [\[CrossRef\]](#)
- Hancer, E.; Xue, B.; Zhang, M. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowl.-Based Syst.* **2018**, *140*, 103–119. [\[CrossRef\]](#)
- Hancer, E.; Xue, B.; Zhang, M. An evolutionary filter approach to feature selection in classification for both single- and multi-objective scenarios. *Knowl.-Based Syst.* **2023**, *280*, 111008. [\[CrossRef\]](#)
- Xue, B.; Fu, W.; Zhang, M. Multi-objective Feature Selection in Classification: A Differential Evolution Approach. In *Proceedings of the Simulated Evolution and Learning*; Dick, G., Browne, W.N., Whigham, P., Zhang, M., Bui, L.T., Ishibuchi, H., Jin, Y., Li, X., Shi, Y., Singh, P., et al., Eds.; Springer: Cham, Switzerland, 2014; pp. 516–528.
- Wang, P.; Xue, B.; Liang, J.; Zhang, M. Differential Evolution-Based Feature Selection: A Niching-Based Multiobjective Approach. *IEEE Trans. Evol. Comput.* **2023**, *27*, 296–310. [\[CrossRef\]](#)
- Bidgoli, A.A.; Ebrahimpour-Komleh, H.; Rahnamayan, S. A Novel Multi-objective Binary Differential Evolution Algorithm for Multi-label Feature Selection. In *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, 10–13 June 2019; pp. 1588–1595. [\[CrossRef\]](#)
- Wang, P.; Xue, B.; Liang, J.; Zhang, M. Feature Selection Using Diversity-Based Multi-objective Binary Differential Evolution. *Inf. Sci.* **2023**, *626*, 586–606. [\[CrossRef\]](#)
- Hancer, E. A new multi-objective differential evolution approach for simultaneous clustering and feature selection. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103307. [\[CrossRef\]](#)
- Yu, X.; Hu, Z.; Luo, W.; Xue, Y. Reinforcement learning-based multi-objective differential evolution algorithm for feature selection. *Inf. Sci.* **2024**, *661*, 120185. [\[CrossRef\]](#)
- Rivera-López, R.; Mezura-Montes, E.; Canul-Reich, J.; Cruz-Chávez, M.A. A permutational-based Differential Evolution algorithm for feature subset selection. *Pattern Recognit. Lett.* **2020**, *133*, 86–93. [\[CrossRef\]](#)
- Mendoza-Mota, J.A. Selección de Atributos con un Enfoque Evolutivo Multiobjetivo. Master's Thesis, Laboratorio Nacional de Informática Avanzada, Xalapa-Enríquez, Mexico, 2021.
- Malekipirbazari, M.; Aksakalli, V.; Shafqat, W.; Eberhard, A. Performance comparison of feature selection and extraction methods with random instance selection. *Expert Syst. Appl.* **2021**, *179*, 115072. [\[CrossRef\]](#)
- Barradas-Palmeros, J.A.; Rivera-López, R.; Mezura-Montes, E.; Acosta-Mesa, H.G. Experimental Study of the Instance Sampling Effect on Feature Subset Selection Using Permutational-Based Differential Evolution. In *Proceedings of the Advances in Computational Intelligence, MICAI 2023 International Workshops*; Calvo, H., Martínez-Villaseñor, L., Ponce, H., Zatarain Cabada, R., Montes Rivera, M., Mezura-Montes, E., Eds.; Springer: Cham, Switzerland, 2024; pp. 409–421. [\[CrossRef\]](#)
- Barradas-Palmeros, J.A.; Mezura-Montes, E.; Rivera-López, R.; Acosta-Mesa, H.G. Computational Cost Reduction in Wrapper Approaches for Feature Selection: A Case of Study Using Permutational-Based Differential Evolution (In press). In *Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC)*, Yokohama, Japan, 30 June–5 July 2024.
- Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for Differential Evolution. In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, Cancun, Mexico, 20–23 June 2013; pp. 71–78. [\[CrossRef\]](#)
- Kitamura, T.; Fukunaga, A. Duplicate Individuals in Differential Evolution. In *Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC)*, Padua, Italy, 18–23 July 2022; pp. 1–8. [\[CrossRef\]](#)
- Al-Tashi, Q.; Abdulkadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Approaches to Multi-Objective Feature Selection: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 125076–125096. [\[CrossRef\]](#)

27. Kukkonen, S.; Coello, C.A. Generalized Differential Evolution for Numerical and Evolutionary Optimization. In *NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop NEO 2015 Held at September 23–25 2015 in Tijuana, Mexico*; Schütze, O., Trujillo, L., Legrand, P., Maldonado, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 253–279. [\[CrossRef\]](#)
28. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
29. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2015. [\[CrossRef\]](#)
30. Price, K.V.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer: Berlin/Heidelberg, Germany, 2005. [\[CrossRef\]](#)
31. Kukkonen, S.; Lampinen, J. GDE3: The third evolution step of generalized differential evolution. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Edinburgh, UK, 2–5 September 2005; Volume 1, pp. 443–450. [\[CrossRef\]](#)
32. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
33. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu> (accessed on 25 July 2023).
34. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [\[CrossRef\]](#)
35. Vargas-Moreno, I.; Rodríguez-Landa, J.F.; Acosta-Mesa, H.G.; Fernández-Demeneghi, R.; Oliart-Ros, R.; Hernández Baltazar, D.; Herrera-Meza, S. Effects of Sterculia Apetala Seed Oil on Anxiety-like Behavior and Neuronal Cells in the Hippocampus in Rats. *J. Food Nutr. Res.* **2023**, *11*, 211–222. [\[CrossRef\]](#)

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