Boosting initial population in multiobjective feature selection with knowledge-based partitioning

DENIZ, Ayca and KIZILOZ, Hakan Ezgi

Boosting Initial Population in Multiobjective Feature Selection with Knowledge-Based Partitioning

Ayça Deniz
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
ayca.deniz@metu.edu.tr

Hakan Ezgi Kiziloz
Department of Computing
Sheffield Hallam University
Sheffield, United Kingdom
hakanezgi@etu.edu.tr

***Abstract*—The quality of features is one of the main factors that affect classification performance. Feature selection aims to remove irrelevant and redundant features from data in order to increase classification accuracy. However, identifying these features is not a trivial task due to a large search space. Evolutionary algorithms have been proven to be effective in many optimization problems, including feature selection. These algorithms require an initial population to start their search mechanism, and a poor initial population may cause getting stuck in local optima. Diversifying the initial population is known as an effective approach to overcome this issue; yet, it may not suffice as the search space grows exponentially with increasing feature sizes. In this study, we propose an enhanced initial population strategy to boost the performance of the feature selection task. In our proposed method, we ensure the diversity of the initial population by partitioning the candidate solutions according to their selected number of features. In addition, we adjust the chances of features being selected into a candidate solution regarding their information gain values, which enables wise selection of features among a vast search space. We conduct extensive experiments on many benchmark datasets retrieved from UCI Machine Learning Repository. Moreover, we apply our algorithm on a real-world, large-scale dataset, i.e., Stanford Sentiment Treebank. We observe significant improvements after the comparisons with three off-the-shelf initialization strategies.**

***Index Terms*—feature selection, evolutionary computation, initial population, multiobjective optimization, binary classification**

I. Introduction

Dimensions of the available data have increased massively with the advances in data acquisition and processing techniques. In a classification task, it is desirable to have a high amount of data to improve learning performance. However, some insignificant features in the data may degrade the classification performance. Moreover, a high amount of data may cause the curse of dimensionality [1]. Feature selection is an effective preprocessing technique that opts for a subset of features that represent the data most informatively. It aims to remove redundant and irrelevant features from data with a constraint of not reducing the learning performance [2]. Therefore, it requires multiobjective optimization, as it has two conflicting objectives [3].

Feature selection contributes to the classification task in many ways [4]. First, it improves learning performance, e.g., classification accuracy. Other than that, it decreases the size of the search space, which reduces the computation time and space requirements. Moreover, it increases the interpretability of the model by converting a complex model to a simpler one. It plays an essential role in many real-world problems such as gene selection [5], natural language processing [6], and image analysis [7].

In the literature, feature selection methods are divided into two main categories: filter- and wrapper-based [1]. Filters calculate each feature's value with a designated function such as chi-square. On the other hand, wrappers assess feature subsets iteratively by incorporating the learning performance of a classifier. Therefore, filters are computationally cheaper than wrappers. However, wrappers generally achieve higher learning performance. Furthermore, there exist hybrid methods that combine filters and wrappers to leverage their strengths [8].

Wrapper-based feature selection is a combinatorial optimization problem with two objectives: minimizing the number of features and maximizing classification performance. When there are n features in a dataset, finding the best subset of features requires 2^n computations. Therefore, exhaustive searches are not feasible for medium or large-scale datasets. For this reason, many heuristic methods have been proposed for the feature selection problem in the literature [9]. Recent studies presented that metaheuristic algorithms are also very effective for the feature selection task [10]. They provide near-optimal solutions in a timely manner. These algorithms begin their search with an initial population. Therefore, their performance relies on the quality of the initial population. It has been shown that wise selection of the initial population leads to better results [11]. In this study, we leverage this information to improve the performance of the feature selection task.

The main contributions of this study are as follows. We propose a new initial population generation mechanism that combines randomness with information gain. Moreover, we enforce a more diverse search space exploration with initial population segmentation. We carry out experiments on the well-known UCI benchmark datasets. In addition, we further experiment on a text-based classification dataset with a large number of features. Finally, we analyze our proposed method, along with three off-the-shelf initialization strategies and compare them with empirical results.

978-1-7281-8671-9/22/$31.00 © 2022 IEEE
doi:10.1109/IJCNN55064.2022.9892123

The rest of this paper is arranged as follows. Section II includes a formal description of the multiobjective feature selection problem along with available initial population generation methods. Section III provides a detailed description of our proposed algorithm. We describe the experimental environment in Section IV. Section V includes the experimental results of our model and its comparison with other methods. Finally, we summarize our findings and share our future work plans in Section VI.

II. Background

A. Multiobjective Feature Selection

Dash and Liu [12] defined feature selection as choosing the minimal feature subset that has remarkably similar representation with the original dataset. Formally, given a dataset D with the features $F = \langle f_1, f_2, ...f_n \rangle$, feature selection finds the most promising subset F_s, i.e., $F_s \subseteq F$, that improves the classification performance for D. Therefore, the feature selection task requires optimizing two objectives: minimizing the number of features and maximizing classification performance. In this task, one solution dominates another one if and only if it provides better value for at least one objective while providing no worse value for the other one. Formal representation of a solution S_i dominating another solution S_j in the solution set SS is presented below:

$$\begin{aligned} S_i \prec S_j \Leftrightarrow |S_i| < |S_j| \ and \ acc(S_i) \geq acc(S_j) \\ or \ |S_i| \leq |S_j| \ and \ acc(S_i) > acc(S_j) \\ where \ S_i, S_j \in SS \end{aligned} \tag{1}$$

Consequently, in multiobjective optimization problems, there exist multiple solutions which any other solution cannot dominate [13]. These solutions are called non-dominated solutions. Non-dominated solutions fit a Pareto-curve [14]. A non-dominated solution S_i is formally presented as given below:

$$S_k \prec S_i \qquad \neg \exists S_k \in SS \tag{2}$$

All the non-dominated solutions in the solution set constitute the candidate solutions. The dataset domain plays a vital role in determining the final solution among the candidate solutions. For the feature selection task, some domains prioritize classification performance over feature size, while minimizing the feature size is more critical for some others.

B. Metaheuristic Algorithms for Feature Selection

In 1998, Yang and Honavar [15] presented that the genetic algorithm has merit for the solution of the feature selection problem. Since then, various metaheuristic approaches have been proposed for the feature selection problem in the literature [16], including Particle Swarm Optimization [8], Tabu Search [17], Artificial Bee Colony Optimization [18]. Teaching-Learning-Based Optimization (TLBO) is another metaheuristic optimization algorithm proposed by Rao et al. [19]. It mimics the information conveyed between teachers and students to solve global optimization problems. Recently, Zou et al. [20] elaborated on TLBO by providing a detailed description of the algorithm, different variants, hybridized versions, and application areas. TLBO has been applied to many different problem domains, including feature selection [21].

TLBO consists of two phases, namely teacher and learner. In the teacher phase, the best individual in the population is selected as the teacher, and the remaining are named students. Then, the teacher shares its information with the students with the aim of improving the knowledge of the whole population. After the teacher phase ends, the learner phase begins. In the learner phase, the students interact with each other in order to enhance the performance of all the students in the population.

An individual in the population is represented with a bit string of length equal to the number of features. In this bit string, 1s and 0s indicate that the respective feature is in the selected feature subset or not, respectively.

In order to exchange knowledge between individuals and enhance the exploration and exploitation capability, we utilize crossover and mutation operators. In the teacher phase, the half uniform crossover operation is applied between the teacher and every student in the population regarding the equation below:

$$x_i^S = \begin{cases} x_i^T, & \text{if } x_i^T = x_i^S \\ rand(0,1), & \text{otherwise} \end{cases} \qquad \forall i \in x^T \tag{3}$$

where x^T is the teacher, x^S is the student, and $rand(0,1)$ is a function that randomly returns 0 or 1. The subscript i indicates the i-th feature of the individual, e.g., x_i^S is the i-th feature of the student. Then, in the learner phase, both crossover and mutation operators are utilized to increase the algorithm's search capability. First, the crossover is applied among students with regard to the equation below:

$$x_i^{S_2} = \begin{cases} x_i^{S_2}, & \text{if } x_i^{S_1} = x_i^{S_2} \\ rand(0,1), & \text{otherwise} \end{cases} \qquad \forall i \in x^{S_1} \tag{4}$$

where x^{S_1} and x^{S_2} represent two randomly selected students. After crossover, all the students are altered regarding the bit-flip mutation equation below:

$$x_i^S = \{1 - x_i^S : rand \leq MP\} \qquad \forall i \in x^S \tag{5}$$

where $rand$ is a random number between 0 and 1 to represent the probability of the feature i being mutated, and MP is the predefined mutation probability.

As there is no single best solution in a multiobjective environment, determining the teacher is not a trivial task. Kiziloz et al. [22] proposed three variants of TLBO for multiobjective optimization. In this study, we utilized multiobjective TLBO with minimum distance. In this version of the algorithm, the teacher is decided by calculating the distance between the candidate solution and the ideal point, (1,1), where the number of features is minimum and accuracy is maximum. The shorter the distance, the better the candidate is.

Algorithm 1: RANDOM INITIALIZATION

Input: total feature size D; population size N; probability P
Output: initial population Pop

Function `RandomInit(`D, N, P`)`**:**
  for *i=1,...,N* **do**
    for *j=1,...,D* **do**
      if $rand < P$ **then**
        $x^i_j \leftarrow 1$
      else
        $x^i_j \leftarrow 0$
    $Pop(i) \leftarrow x^i$
  return Pop

Algorithm 2: INFORMATION GAIN RANKING INITIALIZATION [26]

Input: total feature size D; population size N; sorted information gain ranks of features R
Output: initial population Pop

Function `IGRInit(`D, N, R`)`**:**
  $size \leftarrow min(D, N)$
  $x^0 \leftarrow \langle 0, 0, \ldots, 0 \rangle$
  for *i=1,...,*$size$ **do**
    $x^i \leftarrow x^{i-1}$
    $x^i_{R[i]} \leftarrow 1$
    $Pop(i) \leftarrow x^i$
  return Pop

Algorithm 3: SEGMENTED INITIALIZATION [27]

Input: total feature size D; population size N
Output: initial population Pop

Function `SegmentedInit(`D, N`)`**:**
  $\tau \leftarrow \lceil N/3 \rceil$
  $subPop_1 \leftarrow RandomInit(D, \tau, 0.25)$
  $subPop_2 \leftarrow RandomInit(D, \tau, 0.5)$
  $subPop_3 \leftarrow RandomInit(D, N - 2\tau, 0.75)$
  $Pop \leftarrow subPop_1 \cup subPop_2 \cup subPop_3$
  return Pop

Algorithm 4: PROPOSED INITIALIZATION

Input: total feature size D; population size N; information gain values of features V; number of sub-groups M; maximum ratio of selected features K; deviation size of feature ratio δ
Output: initial population Pop

Function `ProposedInit(`D, N, V, M, K, δ`)`**:**
  $Bins \leftarrow FindBins(D, V)$
  $\tau \leftarrow \lceil N/M \rceil$, $\tau' \leftarrow N - (M-1)\tau$
  $\varphi \leftarrow \lceil K/M \rceil$
  for *i=1,...,*$M-1$ **do**
    $\beta_i \leftarrow i \times \varphi$
    $subPop_i \leftarrow CreateSub(D, \tau, Bins, \beta_i, \delta)$
  $subPop_M \leftarrow CreateSub(D, \tau', Bins, K, \delta)$
  $Pop \leftarrow subPop_1 \cup subPop_2 \cup \ldots \cup subPop_M$
  return Pop

C. Initialization Methods

Population initialization is one of the critical factors that affect the performance of evolutionary computation techniques. In the literature, it has been shown that a better initial population speeds up the convergence and improves the quality of the solutions [11], [23]–[25]. In the following subsections, we share three available initial population generation methods that we compare our results with.

a) Random initialization (Rnd): This method is a commonly applied initial population generation technique. All the individuals are populated in a completely random way, each feature in each individual having equal chances of being selected or not (see Algorithm 1).

b) Information Gain Ranking (IGR): This method was proposed by Deniz and Kiziloz [26]. In this method, first, the information gain values of each feature are calculated and sorted in decreasing order. The first individual is then generated by selecting the feature with the highest information gain value, i.e., the first feature in the rank list. The selection of the first two features in the rank list constitutes the second individual. The third individual has three features selected: the first three features in the rank list. This process continues until either all features are selected or the population size is met. The algorithm of this method is provided in Algorithm 2.

c) Segmented Initialization (SI): This method was proposed by Xu et al. [27]. It provides a simple, yet, powerful modification to Rnd in terms of exploration capability. Rather than providing equal chances for selecting each feature for each individual, it splits the population into three similar-sized groups. They obtain a segmented initial population by varying the probabilities of selecting a feature in each group to 25%, 50%, and 75%, respectively, as presented in Algorithm 3.

III. PROPOSED METHOD

In our proposed method, Algorithm 4, we boost random initialization with information gain and population segmentation. First, we partition the features into four bins, B^1 to B^4 (Algorithm 5). For this purpose, we calculate the first, second, and third quartiles of information gain values of the features. The bin B^1 consists of the least important features, i.e., the ones having information gain values less than the first quartile. In contrast, the bin B^4 includes the most important features, i.e., the ones having information gain values greater than the third quartile. The contents of the remaining two bins, B^2 and B^3, are determined similarly. Next, analogous to the SI method, we create M similar-sized sub-groups to increase diversity. Each sub-group populates its individuals

Algorithm 5: FIND BINS

Input: total feature size D; information gain values of features V
Output: bins B

Function `FindBins(D, V)`:
- $thr_1 \leftarrow FirstQuartile(V)$
- $thr_2 \leftarrow Median(V)$
- $thr_3 \leftarrow ThirdQuartile(V)$
- **for** *i=1,...,D* **do**
  - **if** $V_i < thr_1$ **then**
    - $B^1 \leftarrow B^1 \cup i$
  - **else if** $V_i < thr_2$ **then**
    - $B^2 \leftarrow B^2 \cup i$
  - **else if** $V_i < thr_3$ **then**
    - $B^3 \leftarrow B^3 \cup i$
  - **else**
    - $B^4 \leftarrow B^4 \cup i$
- **return** B

Algorithm 6: CREATE SUB-POPULATION

Input: total feature size D; population size N; bins B; feature ratio F; deviation size of feature ratio δ
Output: sub population SubPop

Function `CreateSub(D, N, B, F, δ)`:
- $x^0 \leftarrow \langle 0, 0, \ldots, 0 \rangle$
- **for** *i=1,...,N* **do**
  - $x^i \leftarrow x^0$
  - $size \leftarrow \lceil (D/100) \times ((F - \delta) + (rand \times 2\delta)) \rceil$
  - $size \leftarrow min(D, max(1, size))$ *// range check*
  - **while** $\sum_j x^i_j < size$ **do**
    - **if** $rand < 0.4$ **then**
      - $k \leftarrow RandomlySelectItem(B^4)$
    - **else if** $rand < 0.7$ **then**
      - $k \leftarrow RandomlySelectItem(B^3)$
    - **else if** $rand < 0.9$ **then**
      - $k \leftarrow RandomlySelectItem(B^2)$
    - **else**
      - $k \leftarrow RandomlySelectItem(B^1)$
    - $x^i_k \leftarrow 1$
  - $SubPop(i) \leftarrow x^i$
- **return** $SubPop$

independently, within their predetermined range of feature ratios. In this study, we define feature ratio as given below:

$$f_{ratio} = \frac{selected\ \#\ of\ features}{total\ \#\ of\ features} \tag{6}$$

In the feature selection domain, it may be wise to eliminate a proportion of the features in the initial population. Therefore, we introduce the maximum ratio of selected features, K, as a parameter to the proposed method. This parameter enforces an upper limit to f_{ratio}, and shrinks the search space. Then, this search space is split into M equal portions to determine the feature ratio of each sub-group. Accordingly, each sub-group is provided with a baseline f_{ratio} value, β, as multiples of K/M, up to K. In particular, $\beta_1 = K/M$, $\beta_2 = 2K/M$, and $\beta_M = KM/M = K$ are the baseline values of first, second, and M-th sub-groups, respectively. Finally, to boost the diversity within the sub-groups, we introduce another parameter, δ, which indicates the deviation size of the feature ratio. This parameter allows the f_{ratio} value to span the range of $[\beta_i - \delta, \beta_i + \delta]$.

Individual generation of sub-groups is given in Algorithm 6. In a sub-group, each individual determines its number of selected features within its f_{ratio} range. Then, until this number of features is selected, it selects features one-by-one, as follows. It spins a roulette wheel having 40%, 30%, 20%, and 10% probabilities for the bins B^4, B^3, B^2, and B^1, respectively. According to the roulette wheel selection, a random feature is selected from its respective bin to be included in the current individual. This process is iterated for the population size of the sub-group. Finally, all the sub-groups are merged to generate the initial population.

To emphasize the novelty of our proposed algorithm, we state its differences with the IGR and SI methods as follows. The IGR method ranks the features based on their information gain scores. Then, the first N of them are selected, where N is the population size. There are two shortcomings of this approach. The first one is the static assignment of the number of selected features. In IGR, the number of selected features can only increase as high as the population size. Therefore, the f_{ratio} value will approximate 0 when the dataset size is enormous compared to the population size. The second shortcoming of IGR is the selection of the identical features for every individual in the population as the first N ranks are included iteratively. On the contrary to IGR, the SI method does not utilize the information carried within the dataset but executes in a completely random fashion. Briefly, it sets the selection probability of all features and randomly decides whether a feature is selected or not independent from the selection of others. As a result of this approach, less informative features may be selected while discriminative features are filtered out. In addition, all or no features may be selected in extreme cases. Even though our proposed algorithm utilizes the information gain scores and population segmentation at its core, both ideas are implemented differently. First, our proposed algorithm prohibits the selection of individuals having more than $(K + \delta)\%$ of the total feature size into our initial population since one of our objectives is to minimize the number of features. Then, the sub-groups are partitioned with their guaranteed f_{ratio} ranges. Moreover, these f_{ratio} ranges are determined dynamically with respect to the dataset size. Finally, the feature selection in the initial population process is boosted with information gain scores, i.e., more informative features are given higher probabilities to be selected than the

TABLE I
DATASET SPECIFICATIONS

dataset	# of features	# of instances	# of classes
Pima Indian Diabetes (PM)	8	768	2
Breast Cancer (BC)	9	699	2
Mushrooms (MR)	22	8124	2
Wisconsin Breast Cancer (WBC)	30	569	2
Ionosphere (IO)	34	351	2
Waveform (WF)	40	5000	3
Connect-4 Opening (C4)	42	67,557	3
Covertype (CT)	54	581,012	7
Spambase (SB)	57	4601	2
Sonar (SON)	60	208	2
Musk (MU)	168	6598	2
Stanford Sentiment Treebank (SST)	18,296	10,242	3

less informative ones.

IV. Experimental Setup

A. Datasets

We held the experiments on twelve datasets, eleven of which are benchmark datasets commonly used in feature selection studies and retrieved from the UCI Machine Learning Repository [28]. Table I presents the key characteristics of the datasets. In these datasets, the number of features varies between 8 and 18,296, and the number of instances ranges up to 600,000. Some datasets have more than two classes. In the experiments, we utilized the two classes, which have more instances than the other classes. Other than that, for the datasets having more than 10,000 instances, we sampled the dataset by selecting 10,000 instances proportional to the classes' original distribution. Apart from the benchmark datasets, we applied our proposed model on a well-known, real-world dataset, i.e., Stanford Sentiment Treebank (SST). SST was introduced in 2013 by Socher et al. [29]. It contains movie reviews that are sentence-based labelled for their sentiments as positive, negative, or neutral. There exist more than 10,000 sentences split into train and test sets. Before performing any experiments on this dataset, we applied four preprocessing operations: conversion to lowercase, punctuation removal, tokenization, and stop words removal, in respective order. Then, we extracted features using Bag-of-Words. We obtained 18,296 features with this approach, which gave us the chance to verify the efficiency of our proposed model on datasets with an extreme amount of features.

B. Performance Evaluation

We utilized Logistic Regression (LR) to evaluate the performance of each feature subset. LR calculates the likelihood of an event to occur by building a probabilistic model. We used the scikit-learn library[1] for the implementation of LR.

We used the hypervolume indicator to compare the overall performance of the methods. Hypervolume indicator has been a favored metric in multiobjective optimization problems as it measures the convergence and the diversity of the solutions [30]. It calculates the area between a reference point and the set of solutions. For this reason, it captures the distance of the solutions to the ideal point and the distribution of the solutions over the objectives. A higher hypervolume indicates better results. In the hypervolume calculation, we utilized both objective values. One of the objectives, accuracy, has a value range of [0,1]. On the other hand, the other objective, selected feature size, has a wider value range and may increase up to the original feature size of the dataset. To equalize their contributions to the hypervolume, we represented the feature size objective value with f_{ratio}, which also has a value range of [0,1]. Finally, in the hypervolume calculation, we selected the reference point as (1,0), i.e., all the features are selected, and the accuracy is 0. Apart from hypervolume comparisons, we also utilized accuracy to compare all methods in terms of the achieved maximum accuracy value.

Finally, we applied a one-way Anova test to measure whether the differences between the groups are significant, followed by pairwise post hoc two independent scores t-tests if found significant. The significance levels of these tests were set to 0.05.

C. Parameter Settings

In this subsection, we share the parameter settings used in the experimentation part of the study. The population size and the number of generations were set to 40 and 60, respectively. Moreover, the crossover rate was set to 1, and the mutation rate was set to 0.02. To obtain statistically significant results, we ran each algorithm 30 times independently for each dataset [27]. The only exception is the SST dataset, with 15 independent runs, as its execution time is enormous compared to others. We set the parameters of LR as follows: *l2* for penalty, *1e5* for c, and *liblinear* for solver. Finally, we applied 5-fold cross-validation when testing our proposed method on the benchmark datasets.

The parameters of our proposed algorithm were selected as follows. We set the number of sub-groups, M, as 6 in this study. Moreover, we set the maximum ratio of selected features, K, and deviation size of feature ratio δ as 75% and 2.5%, respectively. Therefore, in our experiments, there are 6 sub-groups, with the baseline f_{ratio} values as 12.5%, 25%, 37.5%, 50%, 62.5%, and 75%. Considering the 2.5% deviation size, individuals in the first sub-group can select 10% to 15% of the features, while individuals in the sixth sub-group can select 72.5% to 77.5%. Moreover, none of the individuals may consist of more than 77.5% of the features selected in our initial population. We note that these parameters are highly problem/domain-dependent. In this study, we do not optimize these parameters per dataset; rather, we execute them with the same values to observe their reaction to the change of the initialization method.

V. Results and Discussion

In this section, we share the experiment results along with analysis and discussions. The mean and standard deviation

[1]https://scikit-learn.org/

TABLE II
HYPERVOLUME COMPARISON OF INITIAL POPULATION GENERATION METHODS

Dataset (ANOVA)	Rnd	IGR	SI	Proposed Method
PM $(+)$	$0.769 \pm 0.001(\uparrow)$	$0.770 \pm 0.000(\uparrow)$	$0.770 \pm 0.000(\uparrow)$	0.769 ± 0.001
BC $(+)$	$0.964 \pm 0.000(\uparrow)$	$0.963 \pm 0.001(\downarrow)$	$0.964 \pm 0.001(\circ)$	0.964 ± 0.001
MR $(-)$	0.939 ± 0.003	0.939 ± 0.002	0.939 ± 0.002	0.938 ± 0.003
WBC $(-)$	0.978 ± 0.001	0.978 ± 0.001	0.978 ± 0.001	0.978 ± 0.001
IO $(-)$	0.901 ± 0.003	0.900 ± 0.003	0.901 ± 0.002	0.901 ± 0.002
WF $(+)$	$0.917 \pm 0.001(\downarrow)$	$0.920 \pm 0.001(\uparrow)$	$0.917 \pm 0.001(\downarrow)$	0.919 ± 0.001
C4 $(+)$	$0.791 \pm 0.009(\downarrow)$	$0.809 \pm 0.000(\uparrow)$	$0.801 \pm 0.004(\downarrow)$	0.805 ± 0.001
CT $(+)$	$0.769 \pm 0.006(\downarrow)$	$0.775 \pm 0.001(\circ)$	$0.772 \pm 0.006(\downarrow)$	0.775 ± 0.001
SB $(+)$	$0.892 \pm 0.009(\downarrow)$	$0.912 \pm 0.001(\uparrow)$	$0.899 \pm 0.009(\downarrow)$	0.906 ± 0.008
SO $(+)$	$0.812 \pm 0.014(\downarrow)$	$0.834 \pm 0.007(\uparrow)$	$0.823 \pm 0.013(\circ)$	0.826 ± 0.010
MU $(+)$	$0.863 \pm 0.014(\downarrow)$	$0.925 \pm 0.002(\uparrow)$	$0.883 \pm 0.011(\downarrow)$	0.904 ± 0.006
SST $(+)$	$0.380 \pm 0.004(\downarrow)$	$0.694 \pm 0.010(\uparrow)$	$0.543 \pm 0.008(\downarrow)$	0.670 ± 0.007

TABLE III
MAXIMUM ACCURACY COMPARISON OF INITIAL POPULATION GENERATION METHODS

Dataset (ANOVA)	Rnd	IGR	SI	Proposed Method
PM $(+)$	$0.776 \pm 0.001(\uparrow)$	$0.777 \pm 0.000(\uparrow)$	$0.777 \pm 0.001(\uparrow)$	0.775 ± 0.002
BC $(+)$	$0.971 \pm 0.001(\circ)$	$0.970 \pm 0.001(\downarrow)$	$0.971 \pm 0.001(\circ)$	0.970 ± 0.001
MR $(+)$	$0.953 \pm 0.003(\uparrow)$	$0.954 \pm 0.003(\uparrow)$	$0.954 \pm 0.004(\uparrow)$	0.951 ± 0.003
WBC $(-)$	0.981 ± 0.001	0.981 ± 0.001	0.981 ± 0.001	0.981 ± 0.001
IO $(-)$	0.906 ± 0.004	0.904 ± 0.003	0.905 ± 0.003	0.905 ± 0.003
WF $(+)$	$0.924 \pm 0.002(\downarrow)$	$0.927 \pm 0.001(\circ)$	$0.925 \pm 0.001(\downarrow)$	0.927 ± 0.001
C4 $(+)$	$0.807 \pm 0.007(\downarrow)$	$0.831 \pm 0.000(\uparrow)$	$0.822 \pm 0.006(\circ)$	0.823 ± 0.003
CT $(+)$	$0.778 \pm 0.001(\circ)$	$0.777 \pm 0.001(\downarrow)$	$0.777 \pm 0.001(\circ)$	0.777 ± 0.001
SB $(+)$	$0.918 \pm 0.002(\downarrow)$	$0.923 \pm 0.000(\circ)$	$0.923 \pm 0.002(\circ)$	0.923 ± 0.002
SO $(-)$	0.842 ± 0.010	0.838 ± 0.007	0.838 ± 0.011	0.837 ± 0.008
MU $(+)$	$0.941 \pm 0.001(\downarrow)$	$0.927 \pm 0.002(\downarrow)$	$0.948 \pm 0.001(\circ)$	0.947 ± 0.001
SST $(+)$	$0.724 \pm 0.008(\downarrow)$	$0.697 \pm 0.011(\downarrow)$	$0.715 \pm 0.011(\downarrow)$	0.752 ± 0.007

TABLE IV
EXECUTION TIME COMPARISON OF INITIAL POPULATION GENERATION METHODS

Dataset (ANOVA)	Rnd	IGR	SI	Proposed Method
PM $(+)$	$0.7 \pm 0.2(\downarrow)$	$0.7 \pm 0.1(\circ)$	$0.7 \pm 0.1(\downarrow)$	0.6 ± 0.1
BC $(+)$	$1.0 \pm 0.1(\downarrow)$	$0.9 \pm 0.1(\circ)$	$1.0 \pm 0.1(\downarrow)$	0.9 ± 0.1
MR $(+)$	$28.1 \pm 4.6(\downarrow)$	$26.4 \pm 4.8(\downarrow)$	$29.9 \pm 6.8(\downarrow)$	21.8 ± 4.1
WBC $(+)$	$5.6 \pm 0.4(\downarrow)$	$5.0 \pm 0.3(\circ)$	$5.0 \pm 0.5(\circ)$	4.9 ± 0.5
IO $(+)$	$4.8 \pm 0.6(\downarrow)$	$3.6 \pm 0.3(\uparrow)$	$4.1 \pm 0.5(\circ)$	4.0 ± 0.6
WF $(+)$	$25.1 \pm 3.0(\uparrow)$	$27.2 \pm 1.9(\uparrow)$	$27.3 \pm 3.0(\uparrow)$	28.8 ± 2.3
C4 $(+)$	$79.4 \pm 9.0(\uparrow)$	$112.6 \pm 4.8(\downarrow)$	$98.1 \pm 8.6(\downarrow)$	88.8 ± 5.3
CT $(+)$	$111 \pm 16.6(\downarrow)$	$57.2 \pm 3.6(\uparrow)$	$92.7 \pm 9.6(\downarrow)$	78.0 ± 11.4
SB $(+)$	$156.6 \pm 22.2(\uparrow)$	$156.9 \pm 14.7(\uparrow)$	$195.8 \pm 30.6(\downarrow)$	169.3 ± 18.5
SO $(+)$	$5.7 \pm 0.5(\downarrow)$	$4.4 \pm 0.4(\uparrow)$	$5.4 \pm 0.4(\downarrow)$	4.9 ± 0.4
MU $(+)$	$1221.1 \pm 127.1(\downarrow)$	$126.5 \pm 14.9(\uparrow)$	$1209.0 \pm 141.1(\downarrow)$	934.3 ± 93.8
SST $(+)$	$10627.5 \pm 313.8(\downarrow)$	$929.7 \pm 98.8(\uparrow)$	$5239.8 \pm 270.0(\downarrow)$	2878.4 ± 112.4

of the experiment results achieved by four initial population generation methods are presented in Tables II, III, and IV. To provide a clear presentation, we sorted the datasets in ascending order with respect to their feature sizes in the tables. Moreover, we presented the statistical test results in these tables. At the left-most column, next to the dataset name, a $(-)$ symbol means that the one-way ANOVA test indicates no significant differences between the initial population generation methods; hence, post hoc pairwise analysis is not necessary. On the contrary, a $(+)$ symbol indicates that the differences are significant. In such cases, post hoc pairwise test results were presented with the $(\uparrow)$, $(\downarrow)$, and $(\circ)$ symbols next to the results of Rnd, IGR, and SI methods, indicating whether they perform significantly better than, worse than, or not different than our proposed method, respectively. A result is considered better if it is higher for hypervolume and accuracy and lower for execution time.

In Table II, we present the hypervolume scores achieved by each initial population generation method. For the datasets having less than 40 features, all the methods achieve a similar hypervolume. The ANOVA test indicates no significant differences for MR, WBC, and IO datasets. It can be observed that the IGR method is superior in terms of the hypervolume metric as the feature size increases. It significantly achieves

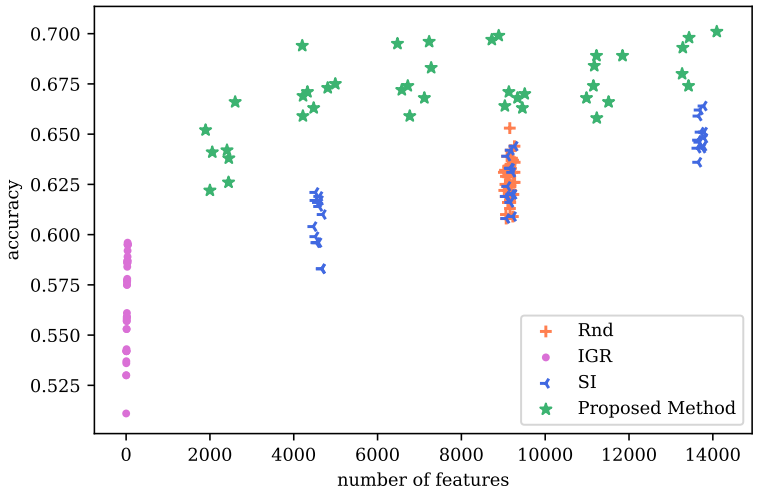

Fig. 1. Distribution of initial populations for all initial population generation methods in the large-scale SST dataset.

the maximum hypervolume score for all the datasets having more than 40 features. In the table, IGR is followed by our proposed method. Similarly, our proposed method significantly outperforms the other two methods, i.e., Rnd and SI, for all the datasets having more than 40 features.

The reason behind this outcome is the inclusion of individuals with minimal f_{ratio} values in the IGR method. As stated before, the hypervolume indicator values the diversity of the solutions, as well as their convergence to the ideal point. IGR method begins with an individual having only one feature selected. The second individual consists of two features, and so on. These individuals with small f_{ratio}s remain in the population with respect to the employed evolutionary algorithm's elitism property. Moreover, these diverse solutions provide a non-negligible classification performance as they have the highest information gain values. Hence, they contribute largely to the hypervolume score. Other methods may struggle to explore such a diverse space closer to the ideal point.

However, higher hypervolume scores do not solely mean that IGR is the best initial population generation method. In Table III, we present the maximum accuracy values achieved by each method. We observed that IGR could not achieve the maximum accuracy value when the dataset's total feature size is larger than the population size (40 in this study). This failure gets even more remarkable as the dataset size increases. Especially for the MU and SST datasets, even the Rnd method performs better than IGR. For the datasets with more than 40 features, our proposed method consistently achieves the highest accuracy values, except for the SO dataset, where the ANOVA test indicates that the differences are insignificant. Apart from the SST dataset, the SI method can also achieve high maximum accuracy values that are indifferent to our proposed method.

Finally, in Table IV, we compare the methods in terms of their execution times as the ANOVA test indicates significant differences for every dataset, regardless of the size. Our proposed method executes significantly faster than Rnd (the only exception is the SB dataset) and SI (exceptions are WBC, IO, and WF datasets). IGR is the fastest initial population generation method for the datasets having more than 40 features. The difference exists regarding the search spaces of the methods. Training a machine learning model with a higher number of features requires more time. IGR has the smallest mean f_{ratio}, followed by our proposed method, SI, and Rnd, in respective order. The employed evolutionary algorithm builds new generations upon the initial populations; hence, the initial population has a high effect on the execution time. Therefore, the obtained results are consistent with theory.

To sum up, all the methods achieve a similar result for both hypervolume and maximum accuracy when the dataset is small. As the dataset grows larger, the IGR method executes fast and achieves high hypervolume scores. However, it tends to get stuck in a local optimum in terms of maximizing accuracy. SI and our proposed method can handle this task well, i.e., as the dataset grows larger, they both find the maximum accuracy values with significant differences. However, the SI method cannot produce diverse solutions; hence, its hypervolume scores remain low. Moreover, its execution time is significantly larger. Our proposed method consistently achieves high accuracy values while keeping the hypervolume score high, with an acceptable execution time.

To provide a better understanding of the initial population generation methods, we visualize the initial populations generated by the four methods for the SST dataset in Figure 1. In this plot, the x-axis represents the number of features, and the y-axis represents the accuracy values. The distributions in the figure support our findings above. In high-dimensional datasets, the difference between the methods becomes clear. In Rnd method (orange markers), all individuals cluster around the center ($f_{ratio} \approx 0.5$). IGR method (pink markers) generates individuals with significantly fewer features. However, like the Rnd method, it clusters around one point and does not provide an exploration area to the search algorithm. In the SI method (blue markers), a more diverse population

is obtained when compared to Rnd and IGR methods. It provides three cluster points that increase the searching ability of the algorithm. Nevertheless, our proposed method (green markers) generates the most diverse individuals among all methods. It provides a large space for the search algorithm for exploration and exploitation. Moreover, it achieves the highest accuracy values among all methods as it leverages the information obtained from the information gain metric. As a result, when finding the most promising features, the evolutionary algorithm is challenged as the number of features increases. Therefore, a diverse initial population boosted with intrinsic knowledge can enhance the searching ability of the algorithm by broadening its exploration space.

VI. Conclusion

Feature selection is one of the key parts of a data analysis process, as a machine learning model's performance depends on the quality of data. Contributions of feature selection are two-fold: reducing the data amount by identifying irrelevant or redundant features and enhancing the model's learning performance. Recently, population-based metaheuristic algorithms have been commonly proposed for the feature selection task. These algorithms begin their search for the optimal feature subset with an initial population, and the initial population has a high impact on the final solution.

In this paper, we introduced a novel initial population generation mechanism for evolutionary algorithms in the feature selection domain. Specifically, our proposed method aims to generate a diverse initial population that can provide a wiser and larger search space to the optimization algorithm. To verify our proposed method's efficiency, we designed various experiments using 12 well-known datasets. Moreover, we compared our method with three off-the-shelf initial population generation methods. The experimental results showed that the proposed method significantly outperforms other methods, especially when the dataset is high-dimensional.

In future work, we plan to further analyze the impact of different methods on generating the initial population. Moreover, we aim to improve the initial population by hybridizing different methods.

References

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[2] X. Qin, X. Liu, and S. Boumaraf, "A new feature selection method based on monarch butterfly optimization and fisher criterion," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[3] K. Demir, B. H. Nguyen, B. Xue, and M. Zhang, "A decomposition based multi-objective evolutionary algorithm with relieff based local search and solution repair mechanism for feature selection," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[4] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media, 2012, vol. 454.

[5] L. Perino, S. Cascianelli, and M. Masseroli, "Hybrid evolutionary framework for selection of genes predicting breast cancer relapse," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[6] A. Deniz, M. Angin, and P. Angin, "Evolutionary multiobjective feature selection for sentiment analysis," *IEEE Access*, vol. 9, pp. 142 982–142 996, 2021.

[7] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2015.

[8] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, 2021.

[9] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Transactions on Cybernetics*, 2020.

[10] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Archives of Computational Methods in Engineering*, pp. 1–25, 2020.

[11] Y. Xue, W. Jia, and A. X. Liu, "A particle swarm optimization with filter-based population initialization for feature selection," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1572–1579.

[12] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[13] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125 076–125 096, 2020.

[14] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective evolutionary algorithms and applications*. Springer Science & Business Media, 2006.

[15] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature extraction, construction and selection*. Springer, 1998, pp. 117–136.

[16] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, p. 100663, 2020.

[17] H. Zhang and G. Sun, "Feature selection using tabu search method," *Pattern recognition*, vol. 35, no. 3, pp. 701–711, 2002.

[18] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Information Sciences*, vol. 422, pp. 462–479, 2018.

[19] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[20] F. Zou, D. Chen, and Q. Xu, "A survey of teaching–learning-based optimization," *Neurocomputing*, vol. 335, pp. 366–383, 2019.

[21] A. K. Shukla, P. Singh, and M. Vardhan, "An adaptive inertia weight teaching-learning-based optimization algorithm and its applications," *Applied Mathematical Modelling*, vol. 77, pp. 309–326, 2020.

[22] H. E. Kiziloz, A. Deniz, T. Dokeroglu, and A. Cosar, "Novel multiobjective tlbo algorithms for the feature subset selection problem," *Neurocomputing*, vol. 306, pp. 94–107, 2018.

[23] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," *Neural Networks*, pp. 2309–2312, 2004.

[24] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping," *Computers & Structures*, vol. 86, no. 11-12, pp. 1204–1218, 2008.

[25] S. Elsayed, R. Sarker, and C. A. C. Coello, "Sequence-based deterministic initialization for evolutionary algorithms," *IEEE transactions on cybernetics*, vol. 47, no. 9, pp. 2911–2923, 2016.

[26] A. Deniz and H. E. Kiziloz, "On initial population generation in feature subset selection," *Expert Systems with Applications*, vol. 137, pp. 11–21, 2019.

[27] H. Xu, B. Xue, and M. Zhang, "Segmented initialization and offspring modification in evolutionary algorithms for bi-objective feature selection," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 444–452.

[28] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[29] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[30] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE transactions on evolutionary computation*, vol. 10, no. 1, pp. 29–38, 2006.