Discretization-based Feature Selection as a Bi-level Optimization Problem

Rihab Said, Maha Elarbi, Slim Bechikh, *Senior Member*, *IEEE*, Carlos A. Coello Coello, *Fellow*, *IEEE*, and Lamjed Ben Said

***Abstract*—Discretization-based feature selection approaches have shown interesting results when using several metaheuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), etc. However, these methods share the same shortcoming which consists in encoding the problem solution as a sequence of cut-points. From this cut-points vector, the decision of deleting or selecting any feature is induced. Indeed, the number of generated cut-points varies from one feature to another. Thus, the higher the number of cut-points, the higher the probability of selecting the considered feature and vice versa. This fact leads to the deletion of possibly important features having a single or a low number of cut-points, such as the infection rate, the glycemia level, and the blood pressure. In order to solve the issue of the dependency relation between the feature selection (or removal) event and the number of its generated potential cut-points, we propose to model the discretization-based feature selection task as a bi-level optimization problem and then solve it using an improved version of an existing co-evolutionary algorithm, named I-CEMBA. The latter ensures the variation of the number of features during the migration process in order to deal with the multimodality aspect. The resulting algorithm, termed Bi-DFS (Bi-level Discretization-based Feature Selection), performs selection at the upper level while discretization is done at the lower level. The experimental results on several high-dimensional datasets show that Bi-DFS outperforms relevant state-of-the-art methods in terms of classification accuracy, generalization ability, and feature selection bias.**

***Index Terms*—Discretization-based feature selection, features interactions, cut-points search, bi-level optimization, co-evolutionary algorithm.**

I. Introduction

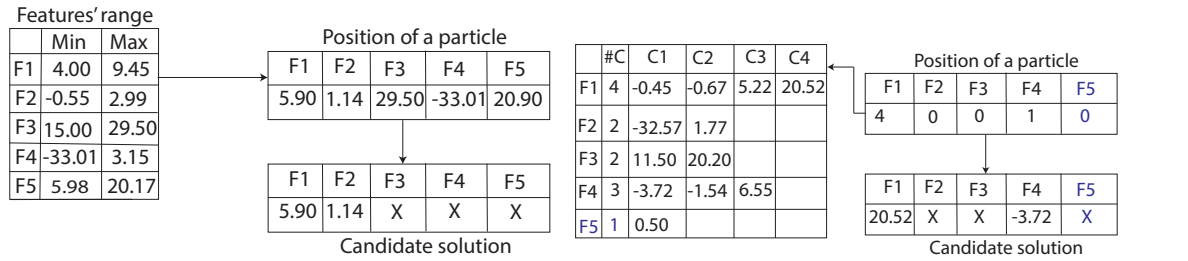
MACHINE learning applications encompass several fields such as: big data [1], predictive and data analytics [2] [3] [4], speech recognition [5], and bioinformatics [6]. For all these fields, a large amount of data is available. In fact, the adopted high-dimensional datasets in machine learning and data mining applications may have a considerable number of irrelevant and redundant features [7], [8]. Those noisy features can negatively influence the classification accuracy of any learning algorithm. Therefore, it is necessary to select only essential and relevant features in order to ensure several machine learning tasks such as classification, clustering, and regression. This pre-processing step is termed feature selection and it is used to improve the performance of the machine learning algorithm. In the literature, feature selection has proven its effectiveness when it is applied on high-dimensional datasets [9], [10]. However, the interactions between features on a large search space make feature selection still a challenging research field [11]. Recently, Evolutionary Algorithms (EAs) have been found to be successful techniques for feature selection tasks when compared to baseline methods [12], [13], [14], [15], [16].

Feature discretization represents a crucial pre-processing task for machine learning and data mining applications. Indeed, several learning algorithms are efficient on discrete data. Thus, the discretization task is used to transform the continuous values of features into their corresponding discrete ones. Indeed, it is shown that the use of discretization can ensure the effectiveness and the efficiency of learning algorithms since noise or minor fluctuations in the data can be ignored [11]. Feature selection and discretization are applied as pre-processing steps for machine learning [17]. However, these two tasks should be performed simultaneously, since the features interactions' information can be lost by performing them independently [18]. Therefore, the classification performance could be negatively affected. In spite of the high number of methods available for feature selection, most of them separate the selection process from the discretization one [18]. For any particular Feature Subset (FS), finding the optimal sequence of features cut-points (thresholds) is not a straightforward task at all. Besides, the quality of a feature subset is heavily influenced by the efficacy of its corresponding cut-points [17]. For this reason, researchers have proposed to frame the problem of feature selection and feature discretization as a *joint problem* named DBFS (*Discretization-Based Feature Selection*) [19], [11], [20], [21], [22]. Basically, the goal of this joint problem is to find an optimal sequence of cut-points from which the selected (and, consequently, the discarded) features are identified; and thus a feature subset is found along with its cut-points. As each feature requiring discretization is numerical, solving the task of feature subset search as a DBFS corresponds to a continuous optimization problem and no more to a discrete one. This fact makes the search space much larger and even infinite, as there is an infinity of real numbers between any pair of real numbers (cf. Fig. 1a).

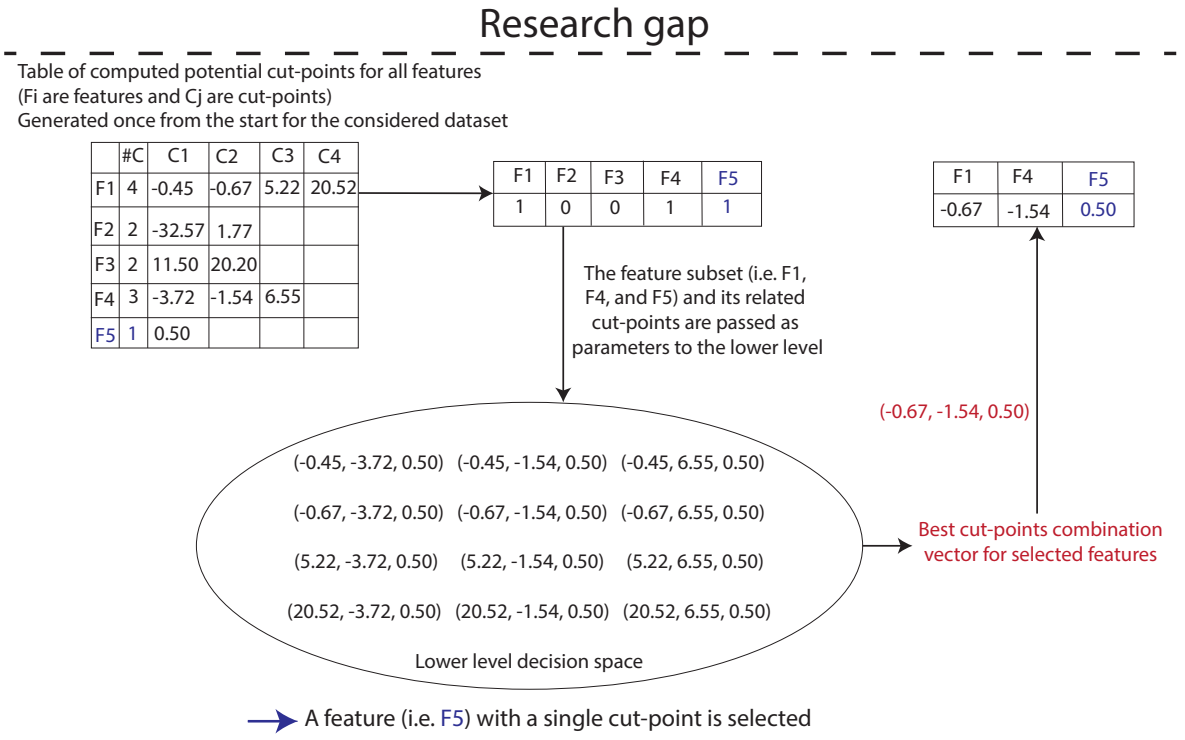
To address the issue of dealing with a continuous (infinite) search space, some researchers have suggested its discretization [11]. As not all real numbers belonging to the feature

R. Said, M. Elarbi, S. Bechikh, and L. Ben Said are with the (SMART) Strategies for Modelling and Artificial Intelligence Laboratory, ISG, University of Tunis, Tunis 2000, Tunisia (e-mail: rihabsaid.edu@gmail.com; arbi.maha@yahoo.com; slim.bechikh@fsegn.rnu.tn; lamjed.bensaid@isg.rnu.tn)

Carlos A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), México, D.F. 07300, MÉXICO and with the Basque Center for Applied Mathematics (BCAM) & Ikerbasque, SPAIN (e-mail: ccoello@cs.cinvestav.mx)



(a) Main idea of Evolve PSO [19]. (b) Main idea of Potential PSO [11].



(c) Main idea of the proposed approach (Bi-DFS).

Fig. 1: Research gap and main idea of Bi-DFS.

range allow an effective separation (splitting) of data, a table of maximum-entropy cut-points is extracted from the data set and seeded as input to the search algorithm from the start. More specifically, once the data set is available, the MDLP (Minimum Description Length Principle) is applied on each feature range to extract the set of cut-points (thresholds) with maximum entropy of the considered feature. This way, a two-dimensional table of cut-points is deduced where each row gives the list of found cut-points for each feature. Eventually, the rows do not have the same size as the number of potential cut-points differs from one feature to another (cf. Fig. 1b). Once the table of potential (maximum-entropy) cut-points is generated, the solution encoding for the search algorithm is represented as a sequence of potential cut-points indices (vector of integers). Eventually, a feature is discarded if its corresponding integer value does not belong to the feature integer range. This approach has shown interesting results in the specialized literature. Some improvements of such approaches have been also proposed using cooperative co-evolution [20] and the ReliefF filtering method [21]. In spite of the interesting results obtained and the proposal of all these improvements that mostly adopted PSO as a baseline metaheuristic optimizer, all these works share the same shortcoming which consists in ***encoding the solution as a sequence of cut-points***, from which the decision of selecting or removing any feature is induced. It is very important to note that this issue could occur in any metaheuristic algorithm (PSO (particle), GA (chromosome), ACO (ant constructed solution), etc.) as the solution would be always implemented as ***a vector of cut-points***. More specifically, *if the vector element (i.e., position in PSO or gene in GA) contains a potential cut-point, then the feature is selected; otherwise it is discarded*. This is a "dangerous" rule because the number of potential cut-points is defined by the MDLP criterion and thus it varies from one feature to another. Thus, the higher the number of cut-points is, the higher the probability of selecting the considered feature; and vice versa. This leads to raising the probability of removal of possibly important features having a single cut-point or a low number of them, such as the glycemia level, the blood pressure, and the infection rate.

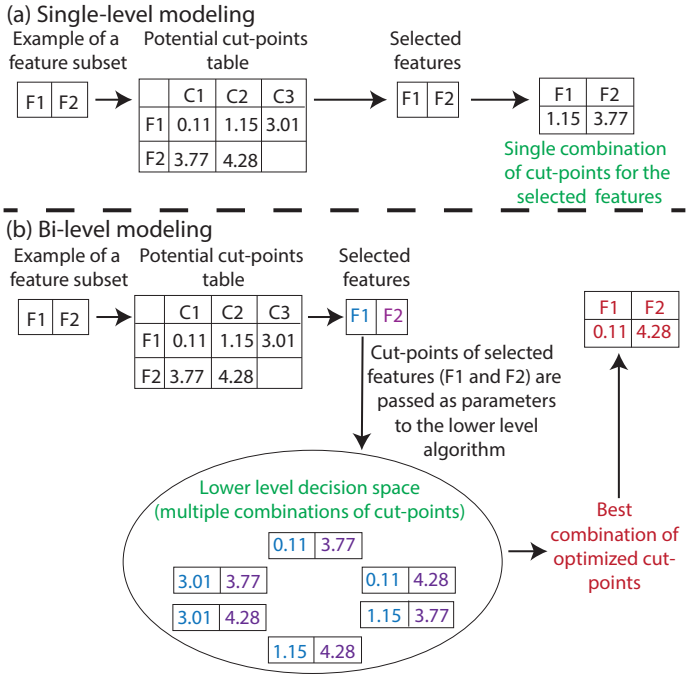


Fig. 2: The added value of the proposed bi-level model over the single-level one for the discretization-based feature selection problem.

To solve the issue of the dependency of the event of feature selection (or removal) on the number of its generated potential cut-points, we propose in this paper a bi-level modeling and resolution of the DBFS, where the upper level selects features while the lower level searches for the corresponding effective cut-points (cf. Fig. 1c). The main contributions of this paper could be summarized as follows:

1) Modeling the DBFS as a bi-level optimization problem and showing the added value of the bi-level model over the single-level one (cf. Fig. 2).
2) Solving the proposed bi-level model using an improved version of the bi-level algorithm CEMBA (Co-Evolutionary Migration-Based Algorithm) [23] which we name I-CEMBA, through the design of a new migration strategy.
3) Showing the ability of Bi-DFS in obtaining better results in terms of classification accuracy, generalization ability, and feature selection bias compared to classical and recent evolutionary approaches.

II. Background and Related Works

A. Feature selection and disretization

Feature selection and discretization represent important tasks in the machine learning field. In fact, feature selection ensures the selection ef relevant features from a dataset of multiple features [24], [25], [26], [27], [28] while feature discretization ensures the generation of a set of cut-points (thresholds or split-points) to partition feature values into their corresponding discrete values [29], [30]. A summary of feature selection and discretization approaches is provided in Section I of the Supplementary Material.

B. Discretization-based feature selection

There are few proposals that have tackled the discretization-based feature selection problem. For instance, Liu and Setiono [31] proposed one of the first discretization-based feature selection algorithms which was termed Chi2. This is a general and simple algorithm that selects and discretizes numerical attributes. Chi2 derives from the ChiMerge that ensures discretization based on the X^2 statistic. In fact, the inconsistency rate is used as a stopping criterion, while the significance value is automatically selected. However, the Chi2 algorithm has two drawbacks. On the one hand, the inaccuracy of ChiMerge is not considered in the Chi2 algorithm. On the other hand, the discretization task could be inaccurate due to the inconsistency rate of the user [32]. Thus, a Modified version of Chi2 (MChi2) was proposed by Tay and Shen [32]. Another approach for feature discretization and selection is the PrEprocessing Solution for Association Rules (PEAR) [33]. It is a supervised approach that discretizes continuous values and selects relevant features for the classification task. It consists of a ranking procedure that goes from features with a small number of cut-points to features with a large number of cut-points in order to form a final subset of top ranked features. In fact, PEAR is able to obtain good results; however, the parameters' tuning of this algorithm is difficult.

Recently, Tran et al. proposed two discretization-based feature selection methods which are: (1) EPSO [19] and (2) PPSO [11]. EPSO achieves feature discretization by the use of the Bare-Bones PSO (BBPSO). For each feature, one cut-point is evolved. The latter can be any value within the range of the feature [Minimum feature value, Maximum feature value]. Then, entropy-based cut-points are used as potential or initial cut-points in order to narrow down the search space. However, with the proposed representation of EPSO, the search space is still too large and BBPSO cannot achieve a better performance. For this reason, Tran et al. [11] proposed the PPSO approach which is based on the combination of discretization and feature selection in a single stage using BBPSO. This approach automatically chooses potentially good cut-points. Indeed, a table of potential cut-points is used and if the obtained cut-point index belongs to the interval [1, #C] (where #C represents the number of potential cut-points), the corresponding feature is selected. More recently, Zhou et al. [20] proposed a Cooperative Co-evolutionary algorithm for Discretization-based Feature Selection (CC-DFS). The idea consists of combining a genetic algorithm with a PSO method in order to search for continuous and discrete features. Furthermore, a ranking mechanism is applied to control the probability of crossover and mutation. Another recent approach was proposed by Zhou et al. [21] in which the size of features is reduced by using a pre-screening process. This approach is an Improved Discretization-based PSO for Feature Selection (IDPSO-FS) that applies the PSO method in order to search for the best combination of cut-points. More recently, Zhou et al. [22] proposed a PSO framework for multi-objective discretization-based feature selection. The proposed work is a Flexible Cut-point PSO (FCPSO) that is based on the selection of an arbitrary number of cut-points. Then, a

TABLE I: Characteristics of existing discretization-based feature selection methods and our proposal (Bi-DFS).

Reference	Approach	Solution encoding	Method type		Dependency of feature removal on the number of discretization cut-points
			Classical	Evolutionary	
[31]	**Chi2**	Interval encoding	×		YES
[32]	**MChi2**	Interval encoding	×		YES
[33]	**PEAR**	Vector encoding	×		YES
[19]	**EPSO**	PSO encoding		×	YES
[11]	**PPSO**	PSO encoding		×	YES
[20]	**CC-DFS**	PSO encoding for discrete features and binary encoding for continuous ones		×	YES
[21]	**IDPSO-FS**	PSO encoding		×	YES
	Bi-DFS	Binary vector encoding for the upper level and real vector encoding for the lower level		×	**NO**

particle update with a mutation method is applied to search for relevant features.

To conclude, we provide, in Table I, the main characteristics of existing single-objective discretization-based feature selection approaches by describing, for each approach, the solution encoding, method type (evolutionary approach or classical approach), and the dependency criterion. We mention here that the dependency criterion indicates if the selection event depends on the number of generated cut-points or not. In order to show the difference between existing discretization-based feature selection approaches and our proposed approach, we also provide the main characterstics of our proposed Bi-DFS approach. In fact, all existing approaches show a promising performance in the resolution of the discretization-based feature selection problem. However, they are based on a dependency between the number of generated cut-points and the selection event which may lead to a deletion of an important number of informative features that may affect the results.

C. Bi-level optimization

A Bi-Level Optimization Problem (BLOP) is a representation of a hierarchical structure that connects two levels: (1) the upper level (i.e., the leader) and (2) the lower level (i.e., the follower). Each level optimizes its objective functions while respecting a set of constraints. It is worth mentioning that the lower level problem belongs to the upper level constraints. The main goal in such a problem is to optimize the upper level objective while ensuring two tasks: (1) respecting the upper level constraints and (2) optimizing the lower level problem. The bi-level mathematical formulation is given as follows where F and f denote the upper and the lower objective functions, respectively:

$$Min\ F(x\ =\ (x_u, x_l)) \tag{1}$$

while respecting the following constraints:

$$\begin{aligned} x_l \in argmin_{x_l}\ f(x_u, x_l) : g_j(x_u, x_l) \leq 0,\ j = 1\ ...\ J, \\ G_k(x_u, x_l) \leq 0, k = 1\ ...\ K, \\ x_u \in X_U,\ x_l \in X_L. \end{aligned} \tag{2}$$

In the two previous equations, x_u and x_l are used for the upper level variables and the lower level variables, respectively. g_i denotes the lower level constraints set, whereas G_j describes the upper level set of constraints. Furthermore, the upper level objective function evaluation requires the optimal lower level solution x_l^*. In other words, the lower level problem uses x_u as a fixed parameter (constant) to search for the optimal solution x_l^*. After that, the upper level will be able to evaluate x_u by using x_l^*. Other details are provided in Section II of the Supplementary Material.

III. Proposed approach

A. Main idea and motivations

Feature selection and discretization should be performed in a simultaneous way. Therefore, if these two tasks are done independently, the information of features' interactions could be lost. Therefore, the classification performance may be negatively affected. In other words, if feature discretization and selection are performed independently, the feature selection process may miss relevant features [19]. Consequently, the feature selection stage may be degraded since important information about features' interaction could be lost during the process of discretization [11]. For this reason, the fact of combining discretization and feature selection tasks into a single stage may lead to a better representation for the learning task. More recently, researchers tackled the discretization-based feature selection problem by proposing evolutionary approaches. We mention here that existing approaches have a single level modeling that encodes the solution as a sequence of cut-points, from which the decision of selecting or removing any feature is induced. This rule leads to removing possibly important features having a single or a low number of cut-points, such as the glycemia level, the infection rate, and the blood pressure. To address this research gap and to solve the issue of the dependency of the event of feature selection (or removal) on the number of its generated potential cut-points, we tackle the discretization-based feature selection by proposing a bi-level model in which the two tasks of discretization and selection are modeled as a bi-level optimization problem. Indeed, features are selected at the upper level problem, while the discretization task is performed at the lower level problem. The added value of such a bi-level model is three-fold (cf. Fig. 2): (a) the feature selection event no longer depends of the number of maximum-entropy (potential) cut-points that are generated, (b) the FS quality evaluation is more precise thanks to the optimization of its cut-point sequence at the lower level, and (c) jointly performing both binary search of FSs and discretization in a synchronous manner, which allows exploiting the advantages of each search process simultaneously. As illustrated in Fig. 3,

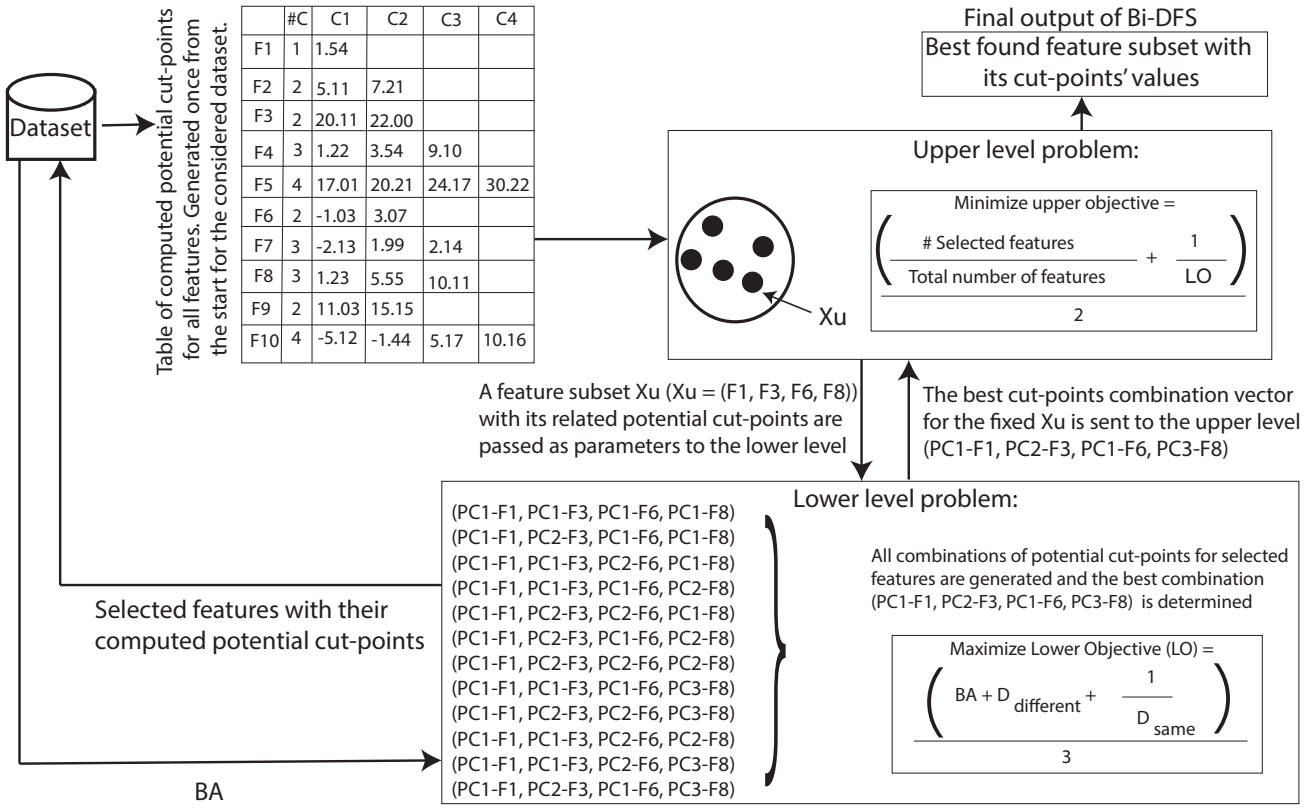


Fig. 3: Illustration of the proposed bi-level model using a simple example in which: #C represents the number of potential cut-points, C_i is the i^{th} cut-point, PC_i-Fj represents the i^{th} Potential Cut-point for the j^{th} Feature, D_{same} is the distance between same instances, $D_{different}$ is the distance between different instances, and BA is the Balanced-Accuracy. All optimized objectives' quantities are normalized in the range [0,1].

a feature subset X_u (in which some features are randomly selected and others are not) with its related potential cut-points are passed as parameters to the lower level problem. It is worth mentioning here that the potential cut-points table is obtained through the application of the entropy and the MDLP principles (cf. Appendix A of the Supplementary Material for further details). After that, a whole lower level search space exists in which all the possible combinations of potential cut-points are generated while building combination vectors of cut-points. A lower level evolutionary process is undertaken through the use of the crossover, mutation, and selection operators in order to vary potential cut-points and obtain the best combination of cut-points that will be sent to the upper level problem in order to evaluate the feature subset X_u. Fig. 3 gives a simple example in which a subset of four selected features ($F1$, $F3$, $F6$, $F8$) is sent to the lower level problem. $F1$ has one cut-point, $F3$ and $F6$ have two cut-points, and $F8$ has three cut-points. In the following, several combinations of cut-points are generated at the lower level until obtaing the best combination which is ($PC1$-$F1$, $PC2$-$F3$, $PC1$-$F6$, $PC3$-$F8$). In other words, the best combination is composed of the cut-point $PC1$-$F1$, the second Potential Cut-point of $F3$ ($PC2$-$F3$), the first Potential Cut-point of $F6$ ($PC1$-$F6$), and the third Potential Cut-point of $F8$ ($PC3$-$F8$). Finally, the upper level receives the best combination of cut-points, then it is also varied through the crossover, mutation, and selection operators. The output is the best found feature subset with its cut-points' values. By following the bi-level model:

1) We remove the dependency between the feature deletion event and the number of its generated cut-points. This way, an interesting feature with one cut-point will be selected.
2) We generate, for each feature subset X_u, several cut-points combinations in the lower level search space and we approximate its globally-optimal combination of cut-points that will be sent to the upper level problem.

In fact, in a bi-level optimization problem, each upper level solution is evaluated with the use of its corresponding lower level solutions. For this reason, a high number of evaluations is required. In other words, the approximation of optimal lower level solutions requires a high number of evaluations. To deal with this high computational cost and to solve the resulting bi-level discretization-based feature selection problem, we have designed an improved version of the CEMBA [23], called I-CEMBA, that ensures the variation of the number of features during the migration step. The main goal behind proposing I-CEMBA is to tackle the multimodality aspect caused by having several feature subsets with the same number of features. To deal with this issue, we apply a variation strategy during the CEMBA migration process in order to diversify the number of selected features among the feature subsets of the current population.

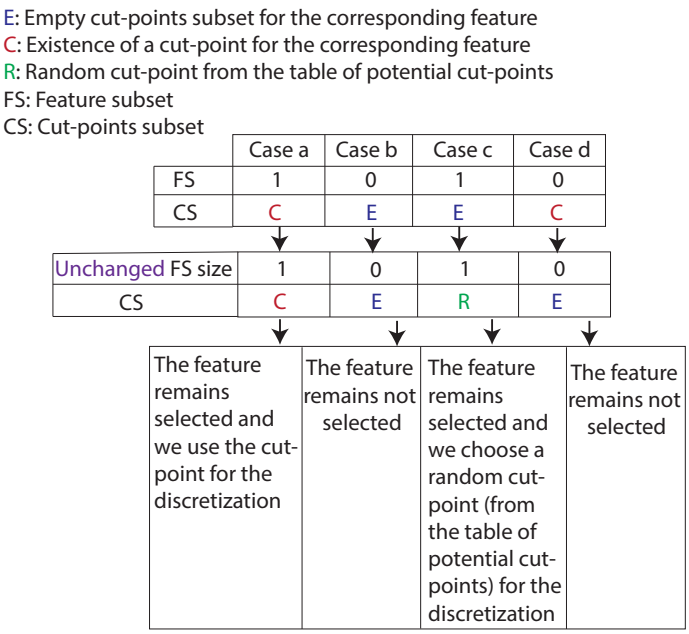


Fig. 4: Illustration of the CEMBA-B migration strategy.

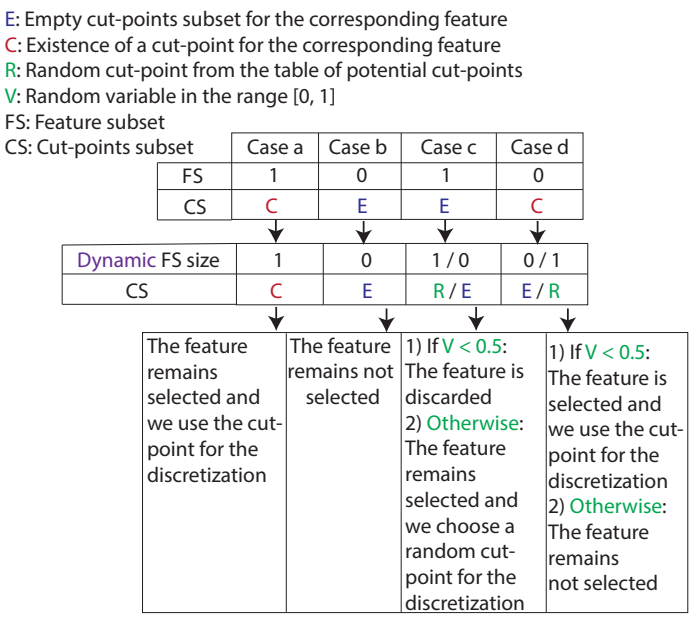


Fig. 5: Illustration of the I-CEMBA migration strategy.

B. I-CEMBA: an improved version of CEMBA

To deal with the bi-level high computational cost and to solve the proposed bi-level model, we need to use a bi-level resolution approach. Indeed, we aim to fulfill three main goals: (1) searching for the number of selected features, (2) identifying the selected features, and (3) identifying the cut-points for each selected feature. In this context, we are facing a multimodality aspect that is caused by the fact that many feature subsets having similar quality values have exactly the same number of features [21],[34]. First of all, we have used our CEMBA Baseline algorithm (CEMBA-B) [23] for the problem resolution. However, we have faced conflicting decisions between the upper level and the lower level. For instance, we can have a cut-point at the lower level problem while its corresponding feature is not selected at the upper level. Also, we can find a selected feature at the upper level while having an empty set of cut-points at the lower level. When applying CEMBA-B, we observe that the CEMBA baseline approach unchanged the feature subset size (cf. Fig. 4). In other words, CEMBA-B preserves the number of selected features. Indeed, in the two normal cases, when a feature is selected at the upper-level and its corresponding cut-point exists in the lower-level (Case a in Fig. 4), this feature remains selected; however, when a feature is discarded at the upper-level and its cut-points set is empty at the lower-level (Case b in Fig. 4), then this feature is discarded. Concerning the other cases, when a feature is discarded at the upper-level while its corresponding cut-point exists in the lower-level (Case d in Fig. 4), then this feature remains discarded; however, when a feature is selected at the upper-level but its cut-points set is empty at the lower-level (Case c in Fig. 4), then the feature is selected and we choose for it a randomly cut-point from the table of cut-points.

In order to address the multimodal issue of the feature selection problem, we proposed an improved version of CEMBA that we called I-CEMBA. The proposed approach utilizes a a repair operation of the migration strategy that is based on varying the feature subset size. It is a diversification rule that varies the number of selected features. To clarify the I-CEMBA migration strategy, we provide its details in Fig. 5. For the two normal cases (cases a and b in Fig. 5), if a feature is selected while having a cut-point at the follower, then this feature remains selected; and if the feature is discarded and it has an empty cut-points set then it remains discarded. The main difference between I-CEMBA and CEMBA-B becomes evident in cases c and d (Fig. 5). On the one hand, case c consists in having a selected feature at the upper level with an empty cut-points set at the lower level. In this situation, a random variable (i.e. V) is generated in the range [0, 1]. In the following, if $V < 0.5$, the feature will be discarded; otherwise, the feature remains selected and we choose a random cut-point from the potential cut-points table. On the other hand, case d occurs when the feature is discarded at the upper level but its corresponding cut-point exists at the follower. In this situation, if $V < 0.5$, the feature will be selected while using the existing cut-point for the discretization; otherwise, the feature remains discarded and its cut-points set will be empty.

C. Detailed description of Bi-DFS

To describe the bi-level model of discretization-based feature selection, we give details about the proposed Bi-DFS approach. We mention here that the interaction between the upper level and the lower level is given by Fig. 3. In fact, it is very rare to have a denominator (LO or D_{same}) which is equal to zero; however, in such case, we add a very small quantity called ϵ.

1) Upper level:

– Solution encoding: Upper level individuals are represented by a binary vector of n bits (features). It is a subset of features in which each bit takes 0 or 1. In other words,

if the feature is selected, then its bit takes 1; otherwise, it takes 0.

– Initialization: To generate initial upper level populations, features are randomly selected in order to constitute the initial populations.
– Evaluation: To evaluate the obtained subset of selected features, an Upper level Fitness Evaluation UFE is used. UFE includes two sub-objectives as follows: (UFE_1) the minimization of the reduction rate of the selected features number and (UFE_2) the combination of the maximization of the Balanced Accuracy (BA) and a Distance (D) measure. The combination of BA and D facilitates the search process by giving a smoother landscape of fitness and helps obtaining a good distinction between feature subsets [11]. It is important to mention here that a weight aggregation is used to ensure the combination of BA and D [35], [20]. Indeed, the distance is added to the function UFE_1 with a weighting coefficient denoted as μ [35]. μ is set to 0.8 (cf. Appendix B in the Supplementary Material) and it is used to bias fitness values towards the balanced accuracy and distance measure [36], [37]. UFE is formulated as follows:

$$UFE = UFE_1 + UFE_2 \tag{3}$$

where:

$$UFE_1 = \frac{SF}{N} \tag{4}$$

and,

$$UFE_2 = \mu \times BA + (1 - \mu) \times D \tag{5}$$

In the previous equations, SF is the number of Selected Features, N representes the total number of features and BA is the Balanced Accuracy that is defined as follows:

$$BA = \frac{1}{NC} \sum_{a=1}^{NC} \frac{NI_a}{|S_a|} \tag{6}$$

and D represents the distance which is calculated as follows:

$$D = \frac{1}{1 + exp^{-5(D_{different} - D_{same})}} \tag{7}$$

$$D_{different} = \frac{1}{|S|} \sum_{i=1}^{|S|} \min_{b|b \neq a, class(V_a) \neq class(V_b)} Dis(V_a, V_b) \tag{8}$$

$$D_{same} = \frac{1}{|S|} \sum_{i=1}^{|S|} \max_{b|b \neq a, class(V_a) = class(V_b)} Dis(V_a, V_b) \tag{9}$$

In equation 6, NC is the number of problem classes, NI defines the number of instances that are correctly identified, and S_a is the size of a sample that belongs to class a. In equation 7, we need to minimize the distance between the same class instances D_{same} and maximize the distance between the different class instances $D_{different}$. In equations 8 and 9, $Dis(V_a, V_b)$ is the calculated distance between a first vector V_a and a second one V_b. It is worth mentioning that V_a and V_b belong to [0, 1]. We mention here that UFE is normalized in the range [0, 1].

– Variation: To search for other features subsets, the two genetic oprators which are crossover and mutation are applied to the upper level populations. After that, a selection operation is performed.

2) Lower level:

– Solution encoding: A vector is used as an encoding for the lower level solution. Indeed, the lower level problem receives the feature subset with its related potential cut-points and builds vectors that represent the existing cut-points combinations. In the context of bi-level programming, each upper level feature vector (i.e., chromosome) has multiple combinations of cut-points in the lower level problem. We mention here that the best combination of cut-points is then sent to the upper level problem.
– Initialization: Entropy is used to find cut-points that are able to split intervals, while MDLP [38] is used to evaluate the generated cut-points (cf. Appendix A of the Supplementary Material for further details).
– Evaluation: The Lower Fitness Evaluation (*LFE*) is represented by the classification Balanced-Accuracy and the distance which are computed at the lower level problem and then, this computation is sent to the upper level problem.

$$LFE = \mu \times BA + (1 - \mu) \times D \tag{10}$$

– Variation: To search for the optimal combination of cut-points, the lower level search space must be varied through the use of crossover, mutation, and selection operators.

TABLE II: Time complexity of each step of a canonical GA.

Step	Time complexity
Population initialization	O(N)
Population evaluation	O(N)
Binary tournament for mating selection	O(N)
Population Variation	O(N)
Replacing parents by their children without competition	O(N)

D. Computational complexity of I-CEMBA and hardware environment

A baseline genetic algorithm starts with population initialization and then repeats the following three steps for a number of generations *G*: (1) Evaluation (fitness computation of each individual), (2) Mating selection (binary tournament is adopted in our work), and (3) Variation (crossover and mutation). Eventually, the N generated children replace their parents. Assuming that N is the population size, the time complexity of the different steps are given by Table II. We conclude from Table II that the time complexity of a canonical GA with binary tournament selection is $O(GN)$, as already studied by Goldberg and Deb in [39]. To evaluate the complexity of our algorithm, we should first study the complexity of CEMBA. CEMBA could be seen as a nested algorithm that divides the population into two populations and then co-evolves them. This is done at each level while ensuring

communication through bi-level interaction and migration. Due to the nested nature of CEMBA, its complexity could be expressed as $O(GN \times LLS)$, where LLS denotes the Lower-Level Search (i.e., the lower-level GA). As the latter also corresponds to a canonical GA, the time complexity of CEMBA is $O(GN \times GN)$, which equals to $O(G^2N^2)$. In spite of the fact that CEMBA decomposes the population to reduce the number of fitness evaluations to almost half, its complexity remains quadratic ($O(G^2N^2)$) when migration is not executed. The migration strategy time complexity is $O(N^2)$ since every lower-level solution will be evaluated with respect to every upper-level solution. As the migration operation belongs to the lower-level evolutionary process, the complexity of LLS becomes $O(G(N+N^2))$ when migration is executed in every lower-level generation. Thus, the time complexity of LLS when applying migration is $O(GN^2)$. It is important to note that the migration step is periodically applied (every 10 lower-level generations in this work). In summary, the time complexity of CEMBA (and also I-CEMBA) is:

- $O(GN \times GN) = O(G^2N^2)$ when migration is not executed; and
- $O(GN \times GN^2) = O(G^2N^3)$ when migration is executed in every generation, which is costly but is still polynomial.

We repeat again that the migration process is executed only every k generations (where $k = 10$, in this work).

Let us move now to analyze the complexity of the Bi-DFS approach using I-CEMBA. After studying the time complexity of I-CEMBA (i.e., the same complexity of CEMBA), we would like to mention that the most time- and memory-consuming operation in our approach is the fitness computation at the lower level, since it requires a communication with the data set to compute the classification performance value in terms of BA. This communication is based on applying the KNN classifier with the considered feature subset and its corresponding sequence of cut-points on the data. To deal with the significant computational cost incurred by wrapper evaluations, we adopted the GPGPU (General-Purpose computation on Graphic Processing Units) approach of Jurczuk et al. [40], which was originally developed for the evolutionary induction of decision trees. All experiments were conducted on a workstation equipped with an Intel®Xeon®Processor E5-2620 v3, 16 GB RAM and a single GPU card "GeForce RTX 2080 Ti WindForce OC-11GB" that contains 4352 CUDA (Compute Unified Device Architecture) cores. The programming language is C++ and the compiler is GCC 10.2.0. The GPU-based parallelization was implemented in CUDA-C and compiled by nvcc CUDA 7.0 (NVIDIA 2015) (single-precision arithmetic was applied). It is worth noting that the entire data set is sent to the GPU only once at the initialization phase of I-CEMBA and it remains in the GPU allocated memory space during the whole evolutionary process. This allows avoiding the important computational cost of data transfer (from CPU to GPU).

We move now to discuss an important issue of our approach which is space complexity. The main space occupied by a canonical GA is the memory space necessary to store the population's chromosomes [39]. For the case of feature selection with a binary encoding, the chromosome length (number of genes) is equal to the number of considered data set features. This may become costly in terms of memory consumption when the data set contains thousands of features (which is the case of this work). To deal with this issue, the following implementation choices and techniques are adopted in our work:

- As the upper-level chromosome is a binary vector of thousands of binary values (0 or 1), it is not a wise choice to implement it as a table of integers because the integer consumes 32 bits (4 bytes). The C++ language offers the bitset class (type) that emulates an array of Boolean elements with optimized space allocation, where each element occupies a single bit. This allowed us to reduce the number of bits required by the upper-level chromosome by 32 times. For instance, if a data set contains 10000 features, only 10000 bits are required when using a bitset array. This number of bits is less than the one required to store 157 double numbers.
- The lower-level chromosome is a vector of floats, which is memory-consuming. For this reason, allocation and deallocation operations for dynamic memory management are used along with GPU parallel computations.

To summarize, we admit that the space complexity required by our approach is important and costly and thus we looked for specific implementation choices to deal with the memory space issue. Eventually, other choices and technologies could be investigated to improve efficiency such as vector quantization and the use a multi-GPU implementation approach.

IV. Experimental environment

A. Datasets

We have used ten datasets where the number of features varies between 2308 and 12600, the number of instances varies between 50 and 203, while the number of classes is between 2 and 11. Details are provided in Section III of the Supplementary Material.

B. Baseline methods

To test the performance of our proposed Bi-DFS approach in solving the discretization-based feature selection, we conducted an experimental study in which we compared Bi-DFS with respect to Bi-DFS-C, which applies the basic CEMBA and four types of existing state-of-the-art approaches: (1) one-stage evolutionary approaches (EPSO [19], PPSO [11], and CC-DFS [20]), (2) two-stage evolutionary approaches (PSO-FS [11]), (3) traditional approaches (MChi2 [32], MDL-LFS [38] [41], MDL-CFS [38] [42], and MDL-CON [38] [43]), and (4) Genetic Algorithm (GA)-based approaches (BL-GA (Bi-Level Genetic-Algorithm) and SL-GA (Single-Level Genetic Algorithm)). Details of the adopted approaches are provided in Section IV of the Supplementary Material.

TABLE III: Default parameters settings

Specific parameters	
Bi-DFS, Bi-DFS-C	Upper Population size: UP_1 = 30, UP_2 = 30, Lower Population size: LP_1 = 30, LP_2 = 30, Upper and Lower Generations: UG = 20, LG = 20, Stopping criterion: 720000 evaluations.
BL-GA	Upper Population size: UP = 60, Lower Population size: LP = 60, Upper and Lower Generations: UG = 20, LG = 20, Stopping criterion: 720000 evaluations.
SL-GA	Population size: 60, Generation number: 20, Stopping criterion: 720000 evaluations.
Common parameters	
Crossover	Type: Uniform Crossover, Probability = 0.9.
Mutation	Type: Uniform Mutation, Probability = 0.1.

C. Experimental setup

To evaluate the performance of Bi-DFS with respect to existing approaches, we adopted a Taguchi method [26]. Table III details the adopted parameters for Bi-DFS, Bi-DFS-C, BL-GA, and SL-GA which are the size of the populations, the maximum number of generations, the stopping criterion, and the genetic operators types and probabilities. For the other algorithms, we adopted the parameters settings of their original papers. To ensure a fair comparison, we adopted the same number of evaluations (720000) as a termination criterion for all the algorithms. As we are in the case of multiple comparisons, we use the Friedman statistical test followed by a posthoc analysis based on the Holm test. The first detected if any of the algorithms is statistically different from the others, while the second adjusts p-values and defines pair-wise relationships [44]. We mention here that we used three statistical symbols: (1) "+" (better), (2) "-" (worse), and (3) "≈" (no significance). KNN (K Nearest Neighbors) is used, in this paper, as a learning algorithm and the Nested Cross-validation strategy is also used with ten folds. More details about ten-fold cross validation and structures of experiments with and without feature selection bias are provided in Section V of the Supplementary Material.

V. Analysis of Results and Discussion

A. Analysis of comparative results

In this sub-section, we provide discussions about the global observations and analysis of the Bi-DFS comparison with respect to state-of-the-art approaches. Table IV shows the classification accuracy results without feature bias of the selected and discretized features obtained by Bi-DFS, PSO-FS, EPSO, PPSO, CC-DFS, MChi2, MDL-CFS, MDL-CON, and MDL-LFS. Indeed, experiments without feature bias were performed using the test set in order to evaluate the performance of the discretized and selected features generated by each compared algorithm based on the training set [16], [45]. Obtained training accuracy results are given by Tables II and III in the Supplementary Material. We mention here that Column "NOF" refers to the Number Of Features.

Regarding the dimensionality reduction, Bi-DFS returns feature subsets that are much smaller than those returned by PSO-FS which represents the two-stage approach. Regarding the classification accuracy, the data transformed by our approach outperforms PSO-FS on all datasets. In fact, results of prostate dataset reveal the largest difference between our proposed Bi-DFS approach and PSO-FS. Indeed, PSO-FS selects 779.2 features with best classification accuracy of 90.01%, but Bi-DFS selects 50.7 features while reaching 100% as the best classification accuracy. In general, solutions produced by our proposed bi-level approach have a significantly better classification accuracy compared to the two-stage approach on all datasets. This observation is explained by the fact that PSO-FS is a two-stage approach that does not perform discretization and selection simultaneously which is not the case of Bi-DFS.

Bi-DFS is compared to two one-stage PSO approaches: (1) EPSO and (2) PPSO. As can be seen from Table IV, Bi-DFS selects a smaller number of features than those selected by EPSO on all datasets and obtains better classification accuracy compared to EPSO on all datasets. For example, Bi-DFS selects 1.9% of features with a classification accuracy of 68.54% in 9Tumor dataset; however, EPSO selects 2.4% of features with 64.20% of classification accuracy. On the other hand, it is shown in Table IV that Bi-DFS selects a smaller number of features than PPSO while obtaining similar or better classification accuracy than PPSO. For leukemia 2, Bi-DFS and PPSO reach 100% as their best classification accuracy, but, Bi-DFS selects a lower number of features than PPSO. On eight datasets, Bi-DFS achieves a better classification accuracy than PPSO while selecting a lower number of features. For the two remaining datasets, Bi-DFS achieves similar classification accuracy to PPSO with a lower number of features. The previous observation is explained by the fact that our proposed approach (Bi-DFS) gives to the algorithm the ability to select potential cut-points. The previous results are explained by the two following facts. On the one hand, the search space of EPSO is much larger and even infinite because we are manipulating an infinity of real numbers in the feature range. On the other hand, the PPSO approach is based on a relation of dependency between the number of generated cut-points and the feature selection task. As it is based on a bi-level model that solves the dependency issue between the feature selection event and its cut-points number, Bi-DFS outperforms EPSO and PPSO.

Compared with respect to CC-DFS, a similar classification accuracy is achieved on two datasets, the best classification is obtained on seven datasets, and the second best classification accuracy is achieved on one dataset. The obtained results are explained by the bi-level structure of Bi-DFS that helps to select a lower number of features while returning the best classification accuracy. It is true that CC-DFS is an improvement of existing discretization-based feature selection approaches but this approach did not solve the issue of the dependency of the event of feature selection or deletion on the number of its generated potential cut-points. For this reason, our proposed approach (i.e., Bi-DFS) outperforms CC-DFS due to the bi-level model and the diversification rule of the adopted I-CEMBA.

It is also important to test the efficiency of our proposed Bi-DFS approach compared to traditional approaches (cf.

TABLE IV: Obtained BA results without feature bias for all algorithms (using KNN as a classifer)

Dataset	Algorithm	NOF	Best	Median	Std
DLBCL	Bi-DFS	37.5	**97.95**	**94.41**	**2.21**
	PSO-FS	100.8	95.65(-)	81.06(-)	5.20(-)
	EPSO	41.9	93.97(-)	86.10(-)	4.75(-)
	PPSO	43.9	95.01(-)	87.97(-)	3.53(-)
	CC-DFS	68.1	97.06(≈)	90.18(-)	3.05(-)
	MChi2	11.0		74.10(-)	
	MDL-CFS	59.1		90.97(-)	
	MDL-CON	4.2		92.00(-)	
	MDL-LFS	4.9		73.20(-)	
SRBCT	Bi-DFS	80.0	**100.00**	99.51	**1.01**
	PSO-FS	148	96.95(-)	91.02(-)	3.01(-)
	EPSO	137.8	**100.00**(≈)	97.09(-)	1.77(-)
	PPSO	109.0	**100.00**(≈)	95.21(-)	2.06(-)
	CC-DFS	219.8	**100.00**(≈)	98.50(-)	1.20(-)
	MChi2	86.5		**100.00(+)**	
	MDL-CFS	81.9		**100.00(+)**	
	MDL-CON	5.4		85.01(-)	
	MDL-LFS	7.5		88.03(-)	
9Tumor	Bi-DFS	109.0	**68.54**	**62.01**	**2.09**
	PSO-FS	950.9	54.44(-)	46.11(-)	4.23(-)
	EPSO	139.6	64.20(-)	57.57(-)	3.31(-)
	PPSO	119.0	65.40(-)	58.13(-)	2.77(-)
	CC-DFS	279.2	62.09(-)	54.11(-)	3.97(-)
	MChi2	59.9		47.70(-)	
	MDL-CFS	39.1		52.91(-)	
	MDL-CON	8.5		27.94(-)	
	MDL-LFS	14.0		40.77(-)	
Brain Tumor 1	Bi-DFS	69.0	**88.74**	**80.22**	**3.00**
	PSO-FS	319.0	77.01(-)	70.69(-)	4.27(-)
	EPSO	152.0	78.25(-)	73.00(-)	3.90(-)
	PPSO	72.9	81.97(-)	75.10(-)	3.60(-)
	CC-DFS	189.0	82.88(-)	77.01(-)	3.40(-)
	MChi2	291.8		73.98(-)	
	MDL-CFS	117.0		78.88(-)	
	MDL-CON	6.9		55.06(-)	
	MDL-LFS	10.2		58.87(-)	
Leukemia 1	Bi-DFS	75.2	**99.88**	**97.11**	**1.22**
	PSO-FS	151.9	91.72(-)	80.06(-)	3.92(-)
	EPSO	137.0	94.95(-)	93.95(-)	1.90(-)
	PPSO	81.9	95.33(-)	94.84(-)	1.50(-)
	CC-DFS	167.7	96.07(-)	94.52(-)	1.51(-)
	MChi2	47.7		91.55(-)	
	MDL-CFS	57.0		92.99(-)	
	MDL-CON	3.7		88.93(-)	
	MDL-LFS	5.4		80.95(-)	

Dataset	Algorithm	NOF	Best	Median	Std
Leukemia 2	Bi-DFS	70.7	**100.00**	**99.00**	**1.07**
	PSO-FS	151.7	93.11(-)	85.81(-)	4.07(-)
	EPSO	140.1	93.99(-)	89.01(-)	3.01(-)
	PPSO	87.7	**100.00**(≈)	95.03(-)	2.78(-)
	CC-DFS	132.6	**100.00**(≈)	95.96(-)	2.11(-)
	MChi2	167.7		92.90(-)	
	MDL-CFS	80.0		89.12(-)	
	MDL-CON	3.3		85.16(-)	
	MDL-LFS	5.1		98.90(≈)	
Prostate	Bi-DFS	50.7	**100.00**	**99.11**	**1.44**
	PSO-FS	779.2	90.01(-)	84.80(-)	2.70(-)
	EPSO	56.1	91.03(-)	82.95(-)	3.46(-)
	PPSO	67.2	96.02(-)	92.00(-)	1.89(-)
	CC-DFS	182.0	91.87(-)	87.99(-)	2.30(-)
	MChi2	34.4		85.92(-)	
	MDL-CFS	52.3		98.97(≈)	
	MDL-CON	5.2		70.01(-)	
	MDL-LFS	5.8		72.79(-)	
Brain Tumor 2	Bi-DFS	60.0	**85.90**	**74.17**	**4.19**
	PSO-FS	419.3	81.88(-)	68.71(-)	5.40(-)
	EPSO	153.9	84.04(-)	71.06(-)	5.12(-)
	PPSO	68.0	75.44(-)	69.01(-)	5.33(-)
	CC-DFS	140.1	82.06(-)	71.07(-)	5.11(-)
	MChi2	161.1		69.90(-)	
	MDL-CFS	64.1		70.79(-)	
	MDL-CON	5.3		61.00(-)	
	MDL-LFS	6.0		52.98(-)	
11Tumor	Bi-DFS	100.2	87.70	84.70	2.50
	PSO-FS	1640.2	85.87(-)	81.92(-)	2.80(-)
	EPSO	150.9	82.87(-)	78.259(-)	3.51(-)
	PPSO	168.0	83.10(-)	77.00(-)	3.95(-)
	CC-DFS	1891.5	**87.73(+)**	**84.74(+)**	**2.49(+)**
	MChi2	2099.7		83.98(-)	
	MDL-CFS	N/A		N/A	
	MDL-CON	10.0		53.10(-)	
	MDL-LFS	14.9		61.20(-)	
Lung Cancer	Bi-DFS	145.6	**93.33**	**87.12**	**1.74**
	PSO-FS	687.7	87.93(-)	80.82(-)	2.58(-)
	EPSO	152.2	84.98(-)	80.84(-)	2.50(-)
	PPSO	204.7	83.91(-)	78.90(-)	3.32(-)
	CC-DFS	157.1	88.21(-)	80.44(-)	2.95(-)
	MChi2	N/A		N/A	
	MDL-CFS	N/A		N/A	
	MDL-CON	7.1		73.99(-)	
	MDL-LFS	12.9		80.01(-)	

Table IV). It is worth mentioning here that the MDL-CFS results are not available (N/A) for the 11Tumor and Lung Cancer datasets. This observation is explained by the fact that the CFS method is computationally expensive. The same observation is seen for the MChi2 approach on the Lung Cancer dataset because of an insufficient memory error.

Compared to MDL-LFS and MDL-CON, our proposed Bi-DFS approach obtains the best training and test accuracies on all datasets. As can seen from Table IV, Bi-DFS achieves higher median test accuracy that belongs to the range [7.11%, 26.32%] compared to MDL-LFS on all datasets. Compared to MDL-CON, Bi-DFS obtains from 2.41% to 34.07% higher median test accuracy on all the adopted datasets. It is true that MDL-CON and MDL-LFS obtain a lower number of features; however, these two methods fail to achieve the best accuracy results.

Compared to MDL-CFS, our proposed approach obtains the best or the same training accuracy on all datasets. For the test results, Bi-DFS obtains better median test accuracy on nine datasets. On the SRBCT dataset, MDL-CFS outperforms Bi-DFS in the median test accuracy; however, our proposed Bi-DFS approach has the best test accuracy on all datasets.

Compared with MChi2, Bi-DFS has the best training accuracy and the best test accuracy on nine datasets. For the SRBCT dataset, MChi2 and Bi-DFS have the same training accuracy; however, MChi2 outperforms Bi-DFS on the median test result while our proposed Bi-DFS approach has always the best test accuracy on all datasets.

All the previous results show the efficiency of Bi-DFS, which is due to the evolutionary bi-level scheme that consists in searching, for each feature subset, its best combination of cut-points in order to have as an output a set of selected features with their optimal cut-points combination. In this way, the evaluation of each feature subset quality is more precise. Moreover, the variation strategy of I-CEMBA helps Bi-DFS in dealing with the multimodality aspect of the feature selection problem.

B. Results with feature selection bias

Another experiment is performed with feature selection bias in order to confirm the performance of our proposed Bi-DFS approach on discretization-based feature selection. In fact, feature selection bias represents an important issue which occurs when the whole data set is utilized during the feature selection process. In this way, there is no unseen test data utilized for feature selection [16], [45], [46]. Table V illustrates the obtained results with feature selection bias for all algorithms

TABLE V: Obtained BA results with feature bias for all algorithms (using KNN as a classifier)

Dataset	Algorithm	NOF	Best	Median	Dataset	Algorithm	NOF	Best	Median
DLBCL	Bi-DFS	39.9	**100.00**	**100.00**	**Leukemia 2**	Bi-DFS	73.2	**100.00**	**100.00**
	PSO-FS	103.8	90.98(-)	89.21(-)		PSO-FS	157.4	90.97(-)	88.99(-)
	EPSO	45.2	98.05(-)	96.15(-)		EPSO	145.1	97.92(-)	96.18(-)
	PPSO	49.3	**100.00**(≈)	**100.00**(≈)		PPSO	90.0	**100.00**(≈)	99.12(-)
	CC-DFS	70.0	**100.00**(≈)	99.19(-)		CC-DFS	135.1	**100.00**(≈)	**100.00**(≈)
	MChi2	13.7		87.09(-)		MChi2	169.4		**100.00**(≈)
	MDL-CFS	63.5		**100.00**(≈)		MDL-CFS	82.6		**100.00**(≈)
	MDL-CON	5.1		95.70(-)		MDL-CON	4.9		97.91(-)
	MDL-LFS	5.7		98.85(-)		MDL-LFS	6.9		98.10(-)
SRBCT	Bi-DFS	83.0	**100.00**	**100.00**	**Prostate**	Bi-DFS	52.2	**100.00**	**100.00**
	PSO-FS	150.7	91.77(-)	88.10(-)		PSO-FS	782.6	93.10(-)	88.92(-)
	EPSO	140.8	92.03(-)	90.22(-)		EPSO	59.2	93.90(-)	90.85(-)
	PPSO	112.1	**100.00**(≈)	99.20(-)		PPSO	70.1	**100.00**(≈)	99.37(-)
	CC-DFS	222.0	**100.00**(≈)	99.70(-)		CC-DFS	185.8	99.11(-)	95.18(-)
	MChi2	88.2		**100.00**(≈)		MChi2	36.6		93.88(-)
	MDL-CFS	83.9		**100.00**(≈)		MDL-CFS	55.7		97.50(-)
	MDL-CON	7.6		94.69(-)		MDL-CON	7.0		**100.00**(≈)
	MDL-LFS	9.5		98.80(-)		MDL-LFS	8.2		95.62(-)
9Tumor	Bi-DFS	111.1	**99.52**	**99.09**	**Brain Tumor 2**	Bi-DFS	64.1	**100.00**	**100.00**
	PSO-FS	961.1	80.17(-)	79.59(-)		PSO-FS	425.6	92.01(-)	90.61(-)
	EPSO	144.7	90.10(-)	89.09(-)		EPSO	159.1	93.44(-)	91.69(-)
	PPSO	123.0	98.06(-)	95.20(-)		PPSO	70.0	**100.00**(≈)	99.83(-)
	CC-DFS	282.3	98.31(-)	96.17(-)		CC-DFS	145.1	**100.00**(≈)	**100.00**(≈)
	MChi2	64.0		98.99(-)		MChi2	164.3		94.05(-)
	MDL-CFS	42.0		89.15(-)		MDL-CFS	68.5		**100.00**(≈)
	MDL-CON	10.6		67.95(-)		MDL-CON	6.9		93.12(-)
	MDL-LFS	16.2		74.01(-)		MDL-LFS	8.1		**100.00**(≈)
Brain Tumor 1	Bi-DFS	71.8	**100.00**	**100.00**	**11Tumor**	Bi-DFS	102.9	**100.00**	99.00
	PSO-FS	325.6	91.29(-)	90.65(-)		PSO-FS	1650.6	85.10(-)	83.85(-)
	EPSO	157.4	95.04(-)	94.90(-)		EPSO	155.0	88.16(-)	85.41(-)
	PPSO	75.1	**100.00**(≈)	**100.00**(≈)		PPSO	170.1	99.15(-)	98.12(-)
	CC-DFS	194.2	**100.00**(≈)	99.92(-)		CC-DFS	1898.5	**100.00**(≈)	**100.00(+)**
	MChi2	299.0		80.00(-)		MChi2	2111.6		93.75(-)
	MDL-CFS		**100.00**(≈)			MDL-CFS	N/A		98.16(-)
	MDL-CON	8.7		92.77(-)		MDL-CON	12.7		72.03(-)
	MDL-LFS	12.9		83.99(-)		MDL-LFS	16.5		78.01(-)
Leukemia 1	Bi-DFS	77.0	**100.00**	**100.00**	**Lung Cancer**	Bi-DFS	148.6	**99.21**	**98.97**
	PSO-FS	155.1	89.11(-)	87.98(-)		PSO-FS	690.9	83.17(-)	80.63(-)
	EPSO	140.9	91.96(-)	90.90(-)		EPSO	155.3	84.05(-)	81.47(-)
	PPSO	85.1	**100.00**(≈)	**100.00**(≈)		PPSO	209.2	97.10(-)	96.90(-)
	CC-DFS	169.7	**100.00**(≈)	**100.00**(≈)		CC-DFS	159.1	98.14(-)	97.25(-)
	MChi2	49.0		98.72(-)		MChi2	N/A		N/A
	MDL-CFS	58.4		**100.00**(≈)		MDL-CFS	N/A		98.16(-)
	MDL-CON	5.2		98.11(-)		MDL-CON	8.8		93.00(-)
	MDL-LFS	7.9		95.90(-)		MDL-LFS	14.0		95.99(-)

using the whole dataset for the algorithm's training. According to Table V, Bi-DFS obtains 100% accuracy on eight datasets. Compared to PSO-FS, EPSO, PPSO, CC-DFS, MChi2, MDL-CFS, MDL-CON, and MDL-LFS, one can notice that Bi-DFS obtains best or similar results on all the adopted datasets.

Compared to evolutionary approaches, the proposed Bi-DFS approach has the best or similar best and median accuracies compared with PSO-FS, EPSO, PPSO, and CC-DFS on all datasets. Compared to traditional approaches, Bi-DFS outperforms MChi2, MDL-CFS, MDL-LFS, and MDL-CON on the 9Tumor, 11Tumor, and Lung Cancer datasets. Compared to MChi2, our proposed Bi-DFS approach has the best accuracy on eight datasets. However, the two approaches have the same accuracy on the two other datasets. Compared to MDL-CFS, Bi-DFS has the best accuracy on four datasets and the same accuracy as MDL-CFS on six datasets. It is seen from Table V that MDL-LFS and MDL-CON obtain a small number of features. However, on nine datasets, our proposed Bi-DFS approach outperforms these two approaches. On the other dataset, Bi-DFS has the same accuracy which is 100%.

C. Analysis of experiments with another classifier

As mentioned in the previous sections, our proposed Bi-DFS approach is able to outperform state-of-the-art approaches in experiments with and without feature selection bias when using KKT as a learning algorithm. In fact, it is interesting to know if the discretized and selected features generated by Bi-DFS can also improve the performance of classifiers other than the one used in the fitness function (i.e., KNN). For this reason, we have performed another experiment by using the Naïve Bayes (NB) algorithm. The main goal behind the experiment is to test if our obtained results (i.e., the ability of the proposed Bi-DFS approach in solving the discretization-based feature selection problem and in outperforming existing approaches) are of a general nature and if these results generalize to other learning algorithms. It is important to mention that each experiment is done independently since KNN and NB are two completely different learning approaches. Generally, it is seen from Table VI that the transformed data from our proposed Bi-DFS approach is able to improve the NB classification accuracy with and without feature selection bias.

For the obtained results without feature bias, our proposed Bi-DFS approach outperforms all algorithms on seven datasets. On two datasets, Bi-DFS has similar results as other approaches while reaching 100% accuracy. On Brain Tumor 2 dataset, CC-DFS outperforms Bi-DFS. As illustrated by Table VI, MDL-CON and MDL-LFS select a small number of features compared to our proposed Bi-DFS approach.

TABLE VI: Experiments with another classifier (NB)

Dataset	Algorithm	Without feature bias			With feature bias		
		NOF	Best	Median	NOF	Best	Median
DLBCL	Bi-DFS	35.3	**98.20**	**95.06**	38.8	**100.00**	**100.00**
	PSO-FS	105.1	94.21(-)	80.11(-)	120.2	88.14(-)	82.71(-)
	EPSO	40.7	91.36(-)	85.17(-)	43.3	98.91(-)	97.01(-)
	PPSO	42.1	93.28(-)	85.76(-)	44.5	**100.00**(≈)	**100.00**(≈)
	CC-DFS	70.0	96.35(-)	89.29(-)	72.1	99.65(-)	93.11(-)
	MChi2	12.6		72.20(-)	13.7		98.94(-)
	MDL-CFS	60.1		90.70(-)	62.2		**100.00**(≈)
	MDL-CON	5.9		91.85(-)	6.5		95.70(-)
	MDL-LFS	6.7		72.83(-)	8.0		96.91(-)
SRBCT	Bi-DFS	78.2	**100.00**	99.71	80.4	**100.00**	**100.00**
	PSO-FS	143.2	95.42(-)	92.17(-)	145.8	90.50(-)	87.02(-)
	EPSO	138.4	99.15(-)	96.78(-)	140.3	89.70(-)	89.13(-)
	PPSO	107.7	**100.00**(≈)	94.99(-)	110.9	**100.00**(≈)	**100.00**(≈)
	CC-DFS	220.1	**100.00**(≈)	99.34(-)	222.5	**100.00**(≈)	95.55(-)
	MChi2	85.3		**100.00**(≈)	87.9		**100.00**(≈)
	MDL-CFS	77.6		**100.00**(≈)	80.6		**100.00**(≈)
	MDL-CON	4.9		84.19(-)	6.5		94.69(-)
	MDL-LFS	6.3		86.95(-)	8.1		98.80(-)
9Tumor	Bi-DFS	106.6	**70.02**	**67.57**	110.7	**90.99**	**88.21**
	PSO-FS	945.8	60.12(-)	49.01(-)	953.8	70.98(-)	69.11(-)
	EPSO	140.7	66.23(-)	58.96(-)	147.6	75.92(-)	73.23(-)
	PPSO	120.1	67.35(-)	59.24(-)	123.8	80.22(-)	79.18(-)
	CC-DFS	282.6	63.33(-)	55.16(-)	286.3	87.43(-)	83.47(-)
	MChi2	55.7		48.09(-)	61.7		85.93(-)
	MDL-CFS	34.2		53.20(-)	41.8		83.90(-)
	MDL-CON	7.9		29.08(-)	9.0		67.95(-)
	MDL-LFS	13.5		41.50(-)	15.2		65.10(-)
Brain Tumor 1	Bi-DFS	65.1	**89.26**	**82.17**	70.7	**100.00**	**100.00**
	PSO-FS	315.6	75.96(-)	69.12(-)	321.4	93.00(-)	90.78(-)
	EPSO	153.2	76.11(-)	70.57(-)	155.9	96.11(-)	96.02(-)
	PPSO	70.4	80.26(-)	76.48(-)	73.6	**100.00**(≈)	99.90(-)
	CC-DFS	178.5	81.47(-)	77.19(-)	193.0	**100.00**(≈)	**100.00**(≈)
	MChi2	284.2		70.44(-)	290.1		82.00(-)
	MDL-CFS	110.6		78.67(-)	114.0		**100.00**(≈)
	MDL-CON	5.7		56.13(-)	7.5		92.77(-)
	MDL-LFS	9.4		59.82(-)	12.3		75.89(-)
Leukemia 1	Bi-DFS	77.4	**100.00**	**99.49**	81.9	**100.00**	**100.00**
	PSO-FS	159.6	92.10(-)	81.74(-)	165.3	90.32(-)	88.47(-)
	EPSO	140.5	94.11(-)	92.52(-)	150.0	93.57(-)	92.19(-)
	PPSO	83.7	95.37(-)	93.28(-)	90.5	**100.00**(≈)	**100.00**(≈)
	CC-DFS	169.7	95.93(-)	94.55(-)	177.4	**100.00**(≈)	**100.00**(≈)
	MChi2	45.5		90.73(-)	50.3		98.72(-)
	MDL-CFS	54.7		91.20(-)	60.0		**100.00**(≈)
	MDL-CON	4.2		88.32(-)	5.9		98.11(-)
	MDL-LFS	5.7		81.17(-)	6.8		**100.00**(≈)

Dataset	Algorithm	Without feature bias			With feature bias		
		NOF	Best	Median	NOF	Best	Median
Leukemia 2	Bi-DFS	75.0	**100.00**	**98.63**	77.1	**100.00**	99.93
	PSO-FS	160.2	92.62(-)	86.55(-)	164.5	94.26(-)	94.39(-)
	EPSO	149.6	93.17(-)	87.11(-)	152.7	98.11(-)	97.64(-)
	PPSO	89.3	**100.00**(≈)	94.25(-)	92.4	**100.00**(≈)	99.86(-)
	CC-DFS	137.1	**100.00**(≈)	96.68(-)	140.7	**100.00**(≈)	**100.00**(≈)
	MChi2	170.5		91.76(-)	173.7		**100.00**(≈)
	MDL-CFS	82.1		90.01(-)	85.9		**100.00**(≈)
	MDL-CON	4.7		84.67(-)		97.91(-)	
	MDL-LFS	6.9		90.11(-)		**100.00**(≈)	
Prostate	Bi-DFS	47.6	**100.00**	**99.77**	51.3	**100.00**	**100.00**
	PSO-FS	780.7	88.98(-)	83.02(-)	784.1	92.16(-)	90.36(-)
	EPSO	54.3	90.62(-)	84.19(-)	58.9	92.94(-)	91.14(-)
	PPSO	69.2	97.36(-)	91.30(-)	72.1	99.15(-)	95.17(-)
	CC-DFS	180.4	94.33(-)	89.15(-)	185.6	98.59(-)	94.11(-)
	MChi2	36.1		84.26(-)	38.0		94.72(-)
	MDL-CFS	53.2		97.09(-)	55.0		98.62(-)
	MDL-CON	5.9		71.58(-)	7.1		**100.00**(≈)
	MDL-LFS	6.1		73.88(-)	7.5		97.62(-)
Brain Tumor 2	Bi-DFS	62.1	87.98	80.19	64.3	**100.00**	**100.00**
	PSO-FS	126.3	77.52(-)	67.43(-)	130.7	93.47(-)	90.10(-)
	EPSO	155.0	79.75(-)	68.14(-)	157.9	94.17(-)	92.19(-)
	PPSO	70.1	73.02(-)	64.22(-)	73.7	**100.00**(≈)	99.13(-)
	CC-DFS	145.2	**88.20(+)**	**80.56(+)**	147.0	**100.00**(≈)	**100.00**(≈)
	MChi2	165.1		65.63(-)	169.4		92.21(-)
	MDL-CFS	64.1		67.10(-)	67.6		**100.00**(≈)
	MDL-CON	6.8		58.34(-)	8.0		93.12(-)
	MDL-LFS	7.9		50.06(-)	8.8		**100.00**(≈)
11Tumor	Bi-DFS	95.8	**90.92**	**85.69**	102.4	**100.00**	**100.00**
	PSO-FS	1610.9	88.59(-)	82.99(-)	1630.0	87.19(-)	86.05(-)
	EPSO	151.2	85.47(-)	79.62(-)	157.7	90.26(-)	89.34(-)
	PPSO	160.7	86.03(-)	79.33(-)	169.3	98.11(-)	95.16(-)
	CC-DFS	1885.3	89.67(-)	84.11(-)	1911.1	**100.00**(≈)	**100.00**(≈)
	MChi2	2102.2		82.64(-)	2110.0		95.65(-)
	MDL-CFS	N/A		N/A	N/A		**100.00**(≈)
	MDL-CON	9.9		52.18(-)	11.9		72.03(-)
	MDL-LFS	12.7		59.30(-)	15.0		77.90(-)
Lung Cancer	Bi-DFS	140.6	**95.26**	**90.93**	147.1	**100.00**	**99.90**
	PSO-FS	680.7	88.82(-)	75.97(-)	691.3	87.55(-)	75.37(-)
	EPSO	153.7	87.11(-)	75.70(-)	157.9	88.09(-)	76.69(-)
	PPSO	206.2	86.14(-)	73.17(-)	210.1	93.00(-)	84.90(-)
	CC-DFS	160.0	90.10(-)	79.86(-)	163.2	95.11(-)	91.28(-)
	MChi2	N/A		N/A	N/A		N/A
	MDL-CFS	N/A		N/A	N/A		99.02(-)
	MDL-CON	7.9		70.25(-)	9.5		93.00(-)
	MDL-LFS	13.8		75.50(-)	14.9		94.80(-)

However, Bi-DFS is able to outperform these two approaches in terms of the classification accuracy results.

Concerning the results with feature bias, Bi-DFS achieves 100% accuracy on nine datasets. On all datasets, Bi-DFS has similar or better accuracy results when compared to traditional approaches. For instance, traditional approaches obtain a lower number of features on 9Tumor, but, Bi-DFS achieves 90.99% accuracy which is the best result on this dataset. On 11Tumor, MDL-LFS selects a lower number of features than Bi-DFS. However, Bi-DFS achieves 100% median accuracy. Furthermore, the proposed Bi-DFS approach has the best accuracy on three datasets. For the other datasets, Bi-DFS has the same best accuracy which is 100%.

All the previous results and observations are explained by the added value of our proposed model that removes the dependency between the feature selection event and its number of cut-points. This fact helps Bi-DFS to ensure a more precise quality evaluation of feature subsets. Furthermore, Bi-DFS is able to solve the multimodal issue by its diversification method used in the migration process of I-CEMBA. In this way, our proposed Bi-DFS approach is able to improve the performance of several classifiers such as KNN and NB.

VI. Further discussion and threats to validity

A. Further discussion

Concerning the generalization ability assessment, the obtained training accuracy results show that our proposed Bi-DFS approach obtains the best or similar training accuracy results compared to other approaches on the majority of the adopted datasets. More details are provided in Section VI of the Supplementary Material. Furthermore, analysis of the experiment on key features selection are provided in Section IX of the Supplementary Material. Indeed, our results show the ability of Bi-DFS in selecting decisive features for lung and prostate cancer.

It is also important to investigate the effectiveness of both bi-level modeling and the improved version of CEMBA. For this reason, an experimental study was performed in order to compare Bi-DFS in which we adopt I-CEMBA, Bi-DFS-C that applies the basic CEMBA, BL-GA, and SL-GA. From the obtained classification accuracy results, we can observe that the worst performance is given by SL-GA compared to BL-GA, Bi-DFS-C, and Bi-DFS. This observation is explained by the fact that SL-GA is based on a single level model that cannot ignore the relation of dependency between the feature selection task and the number of cut-points which is not the case for BL-GA, Bi-DFS, and Bi-DFS-C. In fact, the best performance is given by Bi-DFS due to the use of the I-CEMBA approach. All details are provided in Section VII of the Supplementary Material. Another analysis was conducted in order to test the running time of Bi-DFS compared to the two best approaches (i.e., CC-DFS and PPSO). It is shown from Fig. 6 that CC-DFS and PPSO consume less CPU time than Bi-DFS because the bi-level scheme is computationally costly. The obtained results are provided in Section VIII of the Supplementary Material.

B. Threats to validity

In this section, we discuss the threats to validity that are related to our proposed approach, Bi-DFS. It is important to mention that we focus on three types of threats: (1)

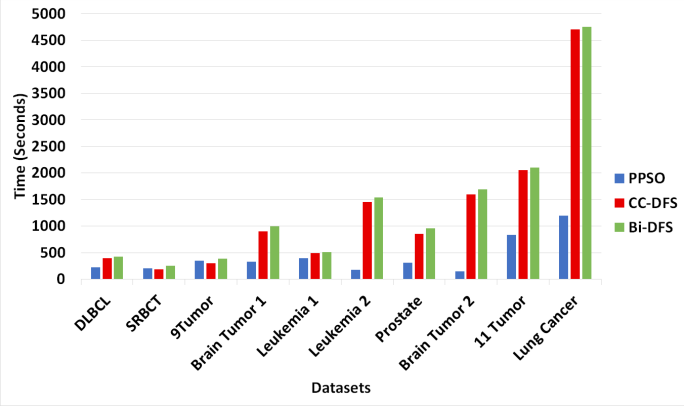


Fig. 6: Comparison of running time (in seconds) for all the adopted datasets.

internal validity, (2) construct validity, and (3) external validity. Indeed, internal validity refers to the relationship between the outcome and the treatment. Construct validity concerns the relationship between the observation and the theory. External validity concerns the generalization of the obtained results. First, the stochastic behavior of Bi-DFS concerns to internal validity. Indeed, a Taguchi method was used to tune the parameters of the proposed algorithm, obtaining encouraging results. However, there is a need for a control strategy in order to manage the adopted parameters of the proposed approach. Second, construct validity occurs because we have used the classification accuracy to test the performance of the proposed approach. In the following, it is interesting to use additional metrics in the evaluation of the proposed approach. Finally, it is important to test the generalization of the obtained results that represent the external threats to validity. In fact, our proposed approach has the best performance compared to other algorithms. However, it would be interesting to test the performance of our proposed Bi-DFS approach on other datasets.

VII. Conclusions and Future Work

The aim of this paper was to tackle the dicretization-based feature selection problem in which the two tasks are simultaneously performed. The main contributions are given as follows. On the one hand, we have proposed a bi-level model of the discretization-based feature selection that optimizes the quality of a feature subset at the upper level while performing the discretization task at the lower level. This bi-level scheme allows a more precise evaluation of any feature subset since we are approximating the optimal sequence of cut-points of each feature subset. Also, the feature selection event no longer depends on the number of its generated potential cut-points. On the other hand, we have designed an improved version of the CEMBA algorithm (I-CEMBA) in order to ensure the variation of the number of features during the migration process. The main goal was to deal with the multimodality issue in feature selection.

Compared to state-of-the-art methods, the conducted experiments on ten datasets show the ability of Bi-DFS in selecting a small number of informative and relevant features while determining their best combination of cut-points. These results are obtained due to the use of a bi-level model for the discretization-based feature selection that makes Bi-DFS able to ignore the dependency between the feature removal event and the number of its generated cut-points.

In the future, it would be interesting to tackle first the feature construction problem and the discretization task as a joint problem. The main goal is to investigate bi-level modeling in order to search for optimal constructed features. Second, it would be interesting to propose a data reduction approach by performing feature selection and instance selection simultaneously. The main goal is to consider the interaction between instances and selected features while improving the learning performance. Third, it would be interesting to tackle feature selection for multi-label data. This direction is worth investigation as it presents more challenges in the interaction between attributes when the instance belongs to several classes at the same time. Fourth, it would be interesting to investigate the causal knowledge in order to ensure a better understanding of the adopted data mechanisms.

Acknowledgements

Carlos A. Coello Coello gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (Investigación en Fronteras de la Ciencia 2016). He was also partially supported by the Basque Government through the BERC 2022-2025 program and by Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2017-0718.

References

[1] A. Lheureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, April 2017.

[2] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using iot sensor data," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS'2017)*. Beijing, China: IEEE, 24-26 November 2017, pp. 87–90, iSBN 978-1-5386-4570-3.

[3] A. Louati, H. Louati, and Z. Li, "Deep learning and case-based reasoning for predictive and adaptive traffic emergency management," *The Journal of Supercomputing*, vol. 77, no. 5, pp. 4389–4418, October 2021.

[4] H. Louati, S. Bechikh, A. Louati, C.-C. Hung, and L. B. Said, "Deep convolutional neural network architecture design as a bi-level optimization problem," *Neurocomputing*, vol. 439, pp. 44–62, June 2021.

[5] Y. H. Jung, S. K. Hong, H. S. Wang, J. H. Han, T. X. Pham, H. Park, J. Kim, S. Kang, C. D. Yoo, and K. J. Lee, "Flexible piezoelectric acoustic sensors and machine learning for speech processing," *Advanced Materials*, vol. 32, no. 35, p. 1904020, October 2019.

[6] K. A. Shastry and H. Sanjay, "Machine learning for bioinformatics," in *Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications*. Springer, January 31 2020, pp. 25–39, iSBN 978-981-15-2445-5.

[7] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, November 2013.

[8] S. Dehuri, S. Ghosh, and C. A. C. Coello, "An introduction to swarm intelligence for multi-objective problems," in *Swarm Intelligence for Multi-objective Problems in Data Mining*. Berlin, Heidelberg: Springer, 2009, pp. 1–17, iSBN 978-3-642-03625-5.

[9] M. Hammami, S. Bechikh, C.-C. Hung, and L. B. Said, "A multi-objective hybrid filter-wrapper evolutionary approach for feature selection," *Memetic Computing*, vol. 11, no. 2, pp. 193–208, July 2019.

[10] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 882–895, January 2020.

[11] B. Tran, B. Xue, and M. Zhang, "A new representation in pso for discretization-based feature selection," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1733–1746, June 2017.

[12] 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, November 2015.

[13] A. S. Abdullah, C. Ramya, V. Priyadharsini, C. Reshma, and S. Selvakumar, "A survey on evolutionary techniques for feature selection," in *2017 Conference on Emerging Devices and Smart Systems (ICEDSS'2017)*. Mallasamudram, India: IEEE, 3-4 March 2017, pp. 58–62, iSBN 9781509055562.

[14] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 473–487, September 2018.

[15] R. Said, S. Bechikh, A. Louati, A. Aldaej, and L. B. Said, "Solving combinatorial multi-objective bi-level optimization problems using multiple populations and migration schemes," *IEEE Access*, vol. 8, pp. 141 674–141 695, August 2020.

[16] B. Tran, B. Xue, M. Zhang, and S. Nguyen, "Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias," *Connection Science*, vol. 28, no. 3, pp. 270–294, May 2016.

[17] C.-F. Tsai and Y.-C. Chen, "The optimal combination of feature selection and data discretization: An empirical study," *Information Sciences*, vol. 505, pp. 282–293, December 2019.

[18] S. Sharmin, M. Shoyaib, A. A. Ali, M. A. H. Khan, and O. Chae, "Simultaneous feature selection and discretization based on mutual information," *Pattern Recognition*, vol. 91, pp. 162–174, July 2019.

[19] B. Tran, B. Xue, and M. Zhang, "Bare-bone particle swarm optimisation for simultaneously discretising and selecting features for high-dimensional classification," in *Applications of Evolutionary Computation, the 19th European Conference on the Applications of Evolutionary Computation (EvoApplications '2016)*. Porto, Portugal: Springer, March 30 – April 1 2016, pp. 701–718, iSBN 978-3-319-31204-0.

[20] Y. Zhou, J. Kang, and X. Zhang, "A cooperative coevolutionary approach to discretization-based feature selection for high-dimensional data," *Entropy*, vol. 22, no. 6, p. 613, June 2020.

[21] Y. Zhou, J. Lin, and H. Guo, "Feature subset selection via an improved discretization-based particle swarm optimization," *Applied Soft Computing*, vol. 98, p. 106794, January 2021.

[22] Y. Zhou, J. Kang, S. Kwong, X. Wang, and Q. Zhang, "An evolutionary multi-objective optimization framework of discretization-based feature selection for classification," *Swarm and Evolutionary Computation*, vol. 60, p. 100770, February 2021.

[23] R. Said, M. Elarbi, S. Bechikh, and L. B. Said, "Solving combinatorial bi-level optimization problems using multiple populations and migration schemes," *Operational Research*, pp. 1–39, January 2021.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, January 2014.

[25] A. Rosales-Pérez, J. A. Gonzalez, C. A. Coello-Coello, C. A. Reyes-Garcia, and H. J. Escalante, "Evolutionary multi-objective approach for prototype generation and feature selection," in *Iberoamerican Congress on Pattern Recognition (CIARP'2014)*. Puerto Vallarta, Mexico: Springer, Cham, 2-5 November 2014, pp. 424–431, iSBN 978-3-319-12567-1.

[26] P. J. Ross, *Taguchi techniques for quality engineering: loss function, orthogonal experiments, parameter and tolerance design*, 1996.

[27] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "Mlaco: A multi-label feature selection algorithm based on ant colony optimization," *Knowledge-Based Systems*, vol. 192, p. 105285, 2020.

[28] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling," *ieee transactions on cybernetics*, vol. 51, no. 4, pp. 1797–1811, 2020.

[29] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.

[30] H. Elhilbawi, S. Eldawlatly, and H. Mahdi, "The importance of discretization methods in machine learning applications: A case study of predicting icu mortality," in *Advanced Machine Learning Technologies and Applications (AMLTA'2021)*. Cairo, Egypt: Springer, 22-24 March 2021, pp. 214–224, iSBN 978-3-030-69717-4.

[31] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'1995)*. Herndon, VA, USA: IEEE, 5-8 November 1995, pp. 388–391, iSBN 0-8186-7312-5.

[32] F. E. Tay and L. Shen, "A modified chi2 algorithm for discretization," *IEEE Transactions on knowledge and data engineering*, vol. 14, no. 3, pp. 666–670, August 2002.

[33] L. J. Sheela and V. Shanthi, "An approach for discretization and feature selection of continuous-valued attributes in medical images for classification learning." *International Journal of Computer and Electrical Engineering*, vol. 1, no. 2, pp. 1793–8163, June 2009.

[34] P. Wang, B. Xue, J. Liang, and M. Zhang, "Multiobjective differential evolution for feature selection in classification," *IEEE Transactions on Cybernetics*, 2021.

[35] B. Tran, M. Zhang, and B. Xue, "A pso based hybrid feature selection algorithm for high-dimensional classification," in *2016 IEEE congress on evolutionary computation (CEC)*. IEEE, 2016, pp. 3801–3808.

[36] B. N. Tran, "Evolutionary computation for feature manipulation in classification on high-dimensional data," Ph.D. dissertation, Open Access Victoria University of Wellington— Te Herenga Waka, 2018.

[37] B. Tran, B. Xue, and M. Zhang, "Genetic programming for multiple-feature construction on high-dimensional classification," *Pattern Recognition*, vol. 93, pp. 404–417, 2019.

[38] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," September 1993.

[39] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 69–93.

[40] K. Jurczuk, M. Czajkowski, and M. Kretowski, "Evolutionary induction of a decision tree for large-scale data: a gpu-based approach," *Soft Computing*, vol. 21, no. 24, pp. 7363–7379, 2017.

[41] M. Gutlein, E. Frank, M. Hall, and A. Karwath, "Large-scale attribute selection using wrappers," in *2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM'2009)*. Nashville, Tennessee, USA: IEEE, 30 March-2 April 2009, pp. 332–339, iSBN 978-1-4244-2765-9.

[42] H. M. A., "Correlation-based feature selection for discrete and numeric class machine learning," in *17th International Conference on Machine Learning (ICML'2000)*. San Francisco, California, USA: Morgan Kaufmann Publishers, 29 June - 2 July 2000, pp. 359–366, iSBN 978-1-55860-707-1.

[43] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial intelligence*, vol. 151, no. 1-2, pp. 155–176, December 2003.

[44] J. Carrasco, S. García, M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm and Evolutionary Computation*, vol. 54, p. 100665, 2020.

[45] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, December 1997.

[46] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proceedings of the national academy of sciences*, vol. 99, no. 10, pp. 6562–6566, 2002.

Rihab Said received the B.Sc., M.Sc., and Ph.D. degrees in Computer Science with Business from the University of Tunis, ISG-Tunis, Tunisia, in 2015, 2018, and 2022, respectively.

Her current research interests include bi-level optimization, evolutionary machine learning, evolutionary computation, multiobjective optimization, metaheuristics, and their applications.

Dr. Said is a reviewer for several international journals such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, SWARM AND EVOLUTIONARY COMPUTATION, and IEEE ACCESS.

Maha Elarbi received the B.Sc., M.Sc., and Ph.D. degrees in Computer Science with Business from the University of Tunis, ISG-Tunis, Tunisia, in 2012, 2014, and 2019, respectively.

Her current research interests include multi- and many-objective optimization, bi-level optimization, evolutionary computation, evolutionary machine learning, and SBSE. She has received the National Presidential Prize for Scientific Research and Technology for the Best National PhD Thesis of the Year 2019.

Dr. Elarbi serves as a reviewer for several international conferences and journals such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, SWARM AND EVOLUTIONARY COMPUTATION, and IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Slim Bechikh (SM'21) received the B.Sc., M.Sc., Ph.D., and Habilitation degrees in Computer Science with Business from the University of Tunis, ISG-Tunis, Tunisia, in 2006, 2008, 2013, and 2015, respectively.

He is currently Full Professor with the University of Carthage, FSEG-Nabeul, Tunisia. He is also a Research Director within the SMART laboratory at the University of Tunis. He published over 85 papers in peer-reviewed journals and refereed conferences. His current research interests include multi-objective optimization, evolutionary machine learning, business analytics, and SBSE.

Dr. Bechikh was a recipient of the Best Paper Award of the ACM SAC-2010 in Switzerland. He supervised the Tunisian best national Doctoral thesis in ICT for the year 2019, which earned a presidential prize in scientific research and technology. He was promoted to the grade of IEEE Senior Member by August 2021. He is Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and SWARM AND EVOLUTIONARY COMPUTATION. He serves as reviewer for over 65 international journals in computational intelligence and its applications.

Carlos Artemio Coello Coello (M'98-SM'04-F'11) received a PhD in computer science from Tulane University, USA, in 1996. He is currently Professor with Distinction (CINVESTAV-3F Researcher) at the Department of Computer Science of CINVESTAV-IPN, in Mexico. He has authored and co-authored over 550 technical papers and book chapters. His publications currently report over 61,800 citations in *Google Scholar* (his h-index is 96). He has received several awards, including the 2007 *National Research Award* from the Mexican Academy of Science (in the area of exact sciences), the *2009 Medal to the Scientific Merit* from Mexico City's congress, the *Ciudad Capital: Heberto Castillo 2011 Award* for scientists under the age of 45, in *Basic Science*, the *2012 Scopus Award* (Mexico's edition) for being the most highly cited scientist in engineering in the 5 years previous to the award and the *2012 National Medal of Science* in Physics, Mathematics and Natural Sciences from Mexico's presidency (this is the most important award that a scientist can receive in Mexico). He received the *Luis Elizondo Award* from the Tecnológico de Monterrey in 2019. Additionally, he is the recipient of the prestigious *2013 IEEE Kiyo Tomiyasu Award*, "for pioneering contributions to single- and multiobjective optimization techniques using bioinspired metaheuristics," of the *2016 The World Academy of Sciences (TWAS) Award* in *Engineering Sciences*, and of the prestigious *2021 IEEE Computational Intelligence Society Evolutionary Computation Pioneer Award*. Since January 2011, he is an IEEE Fellow. He is currently the Editor-in-Chief of the *IEEE Transactions on Evolutionary Computation*.

Lamjed Ben Said received the B.Sc. degree in Computer Science with Business from the University of Tunis, ISG-Tunis, Tunisia, in 1998, the M.Sc. and Ph.D. degrees in Computer Science from the University of Paris VI, Paris, France, in 1999 and 2003, respectively, and the Habilitation degree from the University of Tunis (ISG) in 2011.

He was a Research Fellow with the France Telecom, Research and Development Department, Paris, for three years. He is currently a Full Professor with the University of Tunis (ISG), where he is also the Head of the SMART Laboratory. He published over 200 research papers in refereed international journals, conference proceedings, and book series. His current research interests include multiagent simulation, multicriteria decision making, evolutionary computation, supply chain management, and behavioral economics.

Dr. Ben Said is a reviewer for several artificial intelligence journals and conferences.