MODE-based Oblique Decision Tree Induction: Addressing the Accuracy-Complexity Trade-off through a Normalized Complexity Measure

Ulises-Ramsés Prado-Valderrábano
Artificial Intelligence Research Institute
University of Veracruz
Xalapa, Veracruz, Mexico
zS24019397@estudiantes.uv.mx

Efrén Mezura-Montes
Artificial Intelligence Research Institute
University of Veracruz
Xalapa, Veracruz, Mexico
emezura@uv.mx

Rafael Rivera-López
Department of Computer Systems
TecNM, Veracruz Institute of Technology
Veracruz, Veracruz, Mexico
rafael.rl@veracruz.tecnm.mx

Nancy Pérez-Castro
Artificial Intelligence Research Institute
University of Veracruz
Xalapa, Veracruz, Mexico
naperez@uv.mx

***Abstract*—Oblique Decision Trees improve predictive performance over univariate trees at the cost of increased computational complexity and reduced interpretability. While most approaches formulate their induction as a single-objective problem favoring accuracy, this work proposes a Multi-Objective Differential Evolution framework to jointly minimize classification error and tree size. A normalized complexity measure is introduced to capture the structural properties of the induced models and mitigate discontinuities in the Pareto Fronts. The approach is evaluated using three Multi-Objective Differential Evolution variants across twelve datasets, considering hypervolume, classification accuracy, and tree size as performance indicators. Results show that the proposed method achieves comparable predictive performance to a single-objective approach while significantly reducing model complexity, demonstrating the benefits of explicitly addressing the accuracy–complexity trade-off.**

***Index Terms*—Oblique Decision Trees, Differential Evolution, Multi-objective Optimization**

I. Introduction

Decision Trees (DTs) are widely used classification models due to their intuitive structure and high interpretability [1]. Among them, univariate DTs, also referred to as axis-parallel DTs, are the most common implementation of this model, with popular algorithms such as C4.5 [2] and CART [3]. However, these DTs evaluate a single attribute at each internal node, limiting their ability to partition complex datasets effectively. To overcome this, oblique DTs use linear combinations of attributes to represent test conditions, which define hyperplanes that split the instance space. Formally, an oblique split is defined as:

$$\sum_{i=1}^{d} \beta_i x_i \leq \theta \tag{1}$$

where $\beta_i \in \mathbb{R}$ denotes the coefficient associated with the i-th attribute x_i in a d-dimensional feature space, and θ represents the threshold value. Although oblique DTs tend to be smaller and more accurate than univariate DTs, they are harder to interpret and computationally expensive to induce [4]. In fact, determining the optimal linear combination for each internal node is an NP-complete continuous optimization problem [5]. Moreover, the induction of oblique DTs entails a trade-off between accuracy and complexity, as improving predictive performance often leads to increased model complexity [6].

In this work, oblique hyperplanes are optimized using Differential Evolution (DE), leveraging its strong performance in continuous search spaces, while the accuracy-complexity trade-off is addressed through a multi-objective formulation. This results in a Multi-Objective DE (MODE) framework for inducing oblique DTs. The main contributions are the following:

- The integration of DE with high-performance Multi-Objective Evolutionary Algorithms (MOEAs) to minimize classification error and tree size in oblique DTs.
- A normalized complexity measure that captures the structural properties of the induced models and promotes the generation of regular Pareto Fronts (PFs).

The remainder of this paper is organized as follows: Section II reviews related work, Section III describes the induction mechanism, Section IV presents the multi-objective approach, Section V reports the results, and Section VI concludes the paper.

II. Related Work

In the literature, the use of Evolutionary Algorithms (EAs) for inducing oblique DTs has mainly focused on global search strategies that optimize the tree structure and the associated

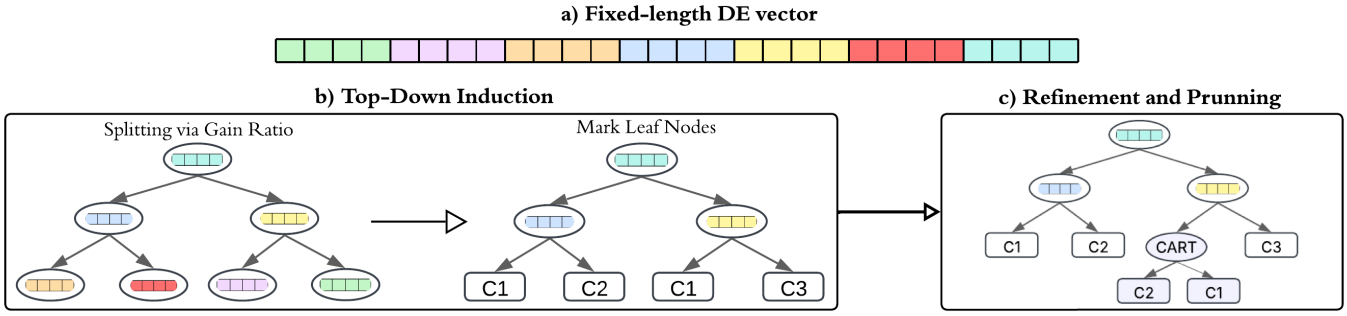


Fig. 1. Representation scheme and mapping strategy to induce oblique DTs from fixed-length DE vectors.

hyperplanes using a single fitness function. Regarding approaches using Genetic Algorithms (GAs), the GDTI method [7] uses linear chromosomes to optimize only accuracy. Conversely, the GEA-ODT approach [8] introduces a tree-based representation and a single-objective fitness function combining accuracy and DT size through a penalty term. The GDT-Mix algorithm [9] further allows both univariate and oblique test conditions under a similar single-objective formulation. Other GA-based methods include model complexity into the objective function, such as that which optimizes the Bayesian Information Criterion (BIC) [10] and the EFTI method [11]. In the context of DE, existing approaches use only a single objective, focusing on optimizing accuracy [12] [13]. In particular, self-adaptation DE versions are used to incorporate DT size as a penalization term in the fitness function [14]. Finally, a small subset of the literature addresses the problem using MOEAs. An early MOEA combined with Genetic Programming (GP) has been applied to optimize accuracy and the number of nodes in DTs [15]. More recently, the ODT-Θ-NSGA-III method, which also relies on GP, optimizes precision and recall to address binary classification tasks in imbalanced datasets [16].

In summary, the majority of approaches treat the problem as a single-objective optimization task. While a limited number of works have applied MOEAs, these are methods that do not exploit DE's strengths or focus solely on classification performance metrics. As a result, the integration of DE with high-performance MOEAs to balance predictive performance and model complexity remains unexplored.

III. DE to induce Oblique DTs

The encoding and construction of oblique DTs via the standard DE method [17] are outlined in this section, following the approach proposed in [14]. DE is employed because the main challenge in oblique DT induction lies in optimizing the real-valued coefficients in oblique splits.

A. Representation Scheme

An oblique DT is encoded as a fixed-length real-valued vector containing only the hyperplanes associated with the internal nodes of the DT (marked with different colors in Fig. 1a), allowing the direct application of DE operators. The number of internal nodes n_e for a complete binary oblique DT is estimated from the number of attributes (d) and class labels (s) of the training set as follows:

$$n_e = 2^{\max(H_i, H_l)-1} - 1 \tag{2}$$

where $H_i = \log_2(d+1)$ and $H_l = \log_2(s)$. The total vector size n is given by:

$$n = n_e(d+1) \tag{3}$$

which corresponds to the total number of real-valued coefficients required for all hyperplanes.

B. Mapping Strategy

To induce an oblique DT from each vector $\mathbf{x}_i$ in the population the following methodology is employed:

1) First, a list of candidate internal nodes $\mathbf{w}_i$ is created using $\mathbf{x}_i$, where the values $\{x_{1,i}, \ldots x_{d+1,i}\}$ are assigned to the h_1 hyperplane, the values $\{x_{d+2,i}, \ldots, x_{2d+2,i}\}$ are assigned to the h_2 hyperplane, and so on.
2) Secondly, the oblique DT is induced following a typical recursive partitioning strategy [1], as shown in Fig. 1b. At each stage, the node in $\mathbf{w}_i$ with the best *Gain Ratio* [2] is selected as:

$$w^* = \text{argmax} f(w_{j,i}, T_{train}); \quad w_{j,i} \in \mathbf{w}_i \tag{4}$$

where f is the splitting criterion and T_{train} is the set of instances associated with the internal node. Leaf nodes are assigned when there is no candidate internal node that can split the instance set.
3) Lastly, a refinement and pruning stage is applied (Fig. 1c), where leaf nodes are evaluated and, when beneficial, replaced by axis-parallel DTs induced with CART [3]. Subsequently, the error-based pruning method [2] is used to avoid overtrained DTs.

As shown, although the vector sizes in DE are fixed, the partitioning strategy and refinement mechanism allow the algorithm to induce oblique DTs of varying sizes.

C. Initialization

Since DE random initialization tends to generate irrelevant hyperplanes that increase model size and reduce accuracy [14], a dipole-based initialization method from [18] is adopted. Mixed dipoles, defined as pairs of instances from different classes $\delta = (\mathbf{v}_p, \mathbf{v}_q)$ are used. For each individual, a random δ is selected, and a hyperplane $h(\boldsymbol{\beta}, \theta)$ is constructed as:

$$\boldsymbol{\beta} = \mathbf{v}_p - \mathbf{v}_q; \qquad \theta = \frac{1}{2}(\boldsymbol{\beta}^T \mathbf{v}_p + \boldsymbol{\beta}^T \mathbf{v}_q) \tag{5}$$

where $\boldsymbol{\beta}$ contains the hyperplane coefficients and θ is the split threshold. This method ensures that the hyperplane separates at least two instances with different class labels.

IV. Multi-objective Optimization Approach

This section describes the proposed framework to induce oblique DTs using a multi-objective optimization approach, in which both classification error and tree complexity are minimized. Formally, the Multi-objective Optimization Problem (MOP) is defined as:

$$\begin{aligned} &\text{Minimize} \quad F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{6}$$

where no constraints are considered, $\mathbf{x}$ is a solution vector of n decision variables and $\Omega \subset \mathbb{R}^n$ denotes the decision space bounded by the minimum and maximum values derived from the dataset attributes. The objective functions that formalize these criteria—and guide the search for near-optimal solutions—are detailed below. Subsequently, the algorithms used to solve the MOP and the strategy for obtaining an oblique DT from the final non-dominated set are described.

A. Classification Error

The objective $f_1(\mathbf{x})$ evaluates the predictive performance of the model induced from $\mathbf{x}$ and is defined as the minimization of the training error:

$$f_1(\mathbf{x}) = \text{error}(DT, T_{train}) \tag{7}$$

where DT denotes the induced oblique DT and T_{train} the training set. By definition, $f_1(\mathbf{x}) \in [0, 1]$.

B. Normalized Complexity Measure

Unlike most existing approaches, which quantify complexity solely by the number of nodes, this work proposes a normalized measure that captures both the oblique structure and the refinement induced in the leaves. This metric is bounded in the range $[0, 1]$ and is defined as a minimization objective:

$$f_2(\mathbf{x}) = \alpha\, C_{ob} + (1 - \alpha)\, C_{ref} \tag{8}$$

where $\alpha \in [0, 1]$ controls the relative importance of the oblique and refinement complexities.

- Oblique Complexity (C_{ob}): Represents the size of the oblique structure:

$$C_{ob} = \frac{n_i}{n_e}, \tag{9}$$

where n_i is the count of oblique nodes actually used (see Fig. 1b), and n_e is the estimation given by Eq. (2).

- Refinement Complexity (C_{ref}): Quantifies the average density of the refinement trees induced at the leaves:

$$C_{ref} = \frac{n_r / n_l}{n_r^{max}}, \tag{10}$$

where n_r is the total count of nodes in all CART trees, n_l is the possible number of leaves for the oblique structure, and $n_r^{max} = 2^{\text{max\_depth}+1} - 1$ is the maximum possible number of nodes for a single CART tree given a depth limit.

C. MODE Algorithms

MOEAs are typically categorized into three groups according to their selection mechanisms [19]. This study examines one algorithm per category, emphasizing methods that allow DE integration. The main characteristics of the selected methods are summarized in Table I.

TABLE I
MODE METHODS CONSIDERED IN THIS WORK.

MOEA	Approach	Selection	Ref.
GDE3	Pareto-based	Non-dominated sorting	[20]
MOEA/D-DE	Decomposition-based	Scalarization	[21]
SMS-EMOA-DE	Indicator-based	Hypervolume (HV)	[22]

D. Preference Handling

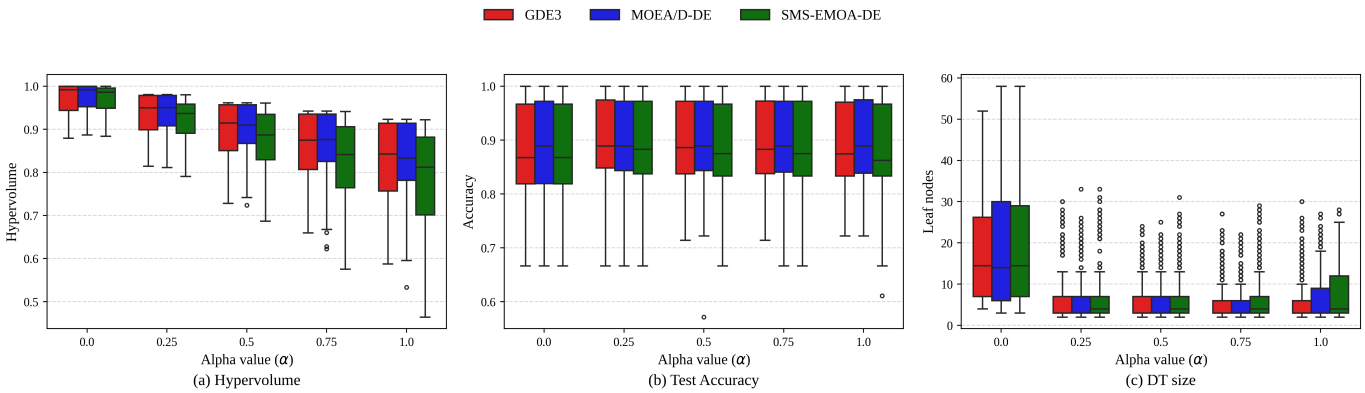
Although a MOEA produces a set of non-dominated solutions, a single oblique DT is required to evaluate its accuracy and size. To this end, the selected solution minimizes the Euclidean distance to the reference point (0,0) in the objective space, yielding a compromise between the objectives.

V. Experiments and Results

This section details the experimental setup employed to evaluate the proposed framework, followed by the results. In addition to classification accuracy and DT size, HV is used to assess the convergence and diversity of MOEAs. Statistical significance is evaluated using a Friedman test with Nemenyi post-hoc analysis. All algorithms were implemented in Python using *jMetalPy* [23] and *scikit-learn* [24].

A. Datasets Description

Twelve datasets from the UCI Machine Learning Repository [25] were selected. Table II summarizes their characteristics. An 80/20 stratified hold-out validation with 30 independent runs per MOEA was employed. This validation approach was preferred over cross-validation to offset the increased execution time of MOEAs while ensuring statistical significance and maintaining a consistent validation protocol throughout the optimization process.



(a) Hypervolume (b) Test Accuracy (c) DT size

Fig. 2. Effect of the α parameter on the performance of the MOEAs across all datasets.

TABLE II
CHARACTERISTICS OF THE DATASETS USED IN THE EXPERIMENTS.

Dataset	Instances	Attributes	Classes
australian-credit	690	14	2
banknote-auth	1372	4	2
biodeg	1055	41	2
cryotherapy	90	6	2
ecoli	336	7	8
iris	150	4	3
mammographic	830	5	2
occupancy	2665	5	2
wine	178	13	3
zoo	101	16	7
dermatology	366	34	6
ionosphere	351	33	2

B. Parameter Settings

The DE parameters were calibrated with *irace* [26], using four datasets from Table II as test instances. The HV indicator was maximized to identify effective configurations and the number of generations was fixed to $g_{max} = 100$ throughout the tuning process. The population size values were set to $NP \in \{50, 100, 150\}$, the crossover rate to $Cr \in \{0.4, 0.5, \ldots, 0.9\}$, and the scaling factor to $F \in \{0.5, 0.6, \ldots, 1.2\}$. Table III shows the resulting DE parameters for each MOEA. Additionally, the depth limit of CART trees was set to $max\_depth = 5$.

TABLE III
DE PARAMETER SETTINGS FOR EACH MOEA OBTAINED WITH *irace*.

MOEA	NP	Cr	F	g_{max}
GDE3	150	0.9	0.6	100
MOEA/D-DE	150	0.5	0.5	100
SMS-EMOA-DE	150	0.7	1.2	100

Regarding the α parameter in Eq. (8), an analysis was conducted considering $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Fig. 2 shows the impact of these values on the performance of the MOEAs. The boxplots depict the distribution of HV (Fig. 2a), accuracy (Fig. 2b), and DT size (Fig. 2c), aggregated across runs and the first ten datasets. As observed, lower α values tend to yield higher HV results. However, a value of $\alpha = 0.0$ drastically increases DT size. In terms of accuracy, the method is relatively insensitive to α, although $\alpha = 0.5$ and $\alpha = 1.0$ exhibit a less variable behavior. Furthermore, DT size is best minimized with values of 0.25, 0.5, and 0.75. Consequently, a value of $\alpha = 0.5$ was selected for the remainder of the experiments.

C. Results and Discussion

Table IV presents the average HV for each MOEA. The Friedman test indicated significant differences ($p = 9.97 \times 10^{-5}$). Consequently, the Nemenyi post-hoc test was conducted and its results can be interpreted using the Critical Difference (CD) diagram shown in Fig. 3. Therefore, it can be stated that GDE3 and MOEA/D-DE outperform SMS-EMOA-DE in terms of HV.

TABLE IV
HV OBTAINED BY THE COMPARED MOEAS.*

Dataset	GDE3		MOEA/D-DE		SMS-EMOA-DE	
	HV	Rank	HV	Rank	HV	Rank
australian-credit	**0.9082**	(1)	0.9074	(2)	0.8613	(3)
banknote-auth	0.9551	(2)	**0.9567**	(1)	0.9078	(3)
biodeg	0.8539	(2)	**0.8702**	(1)	0.8460	(3)
cryotherapy	**0.9396**	(1)	0.9375	(2)	0.9287	(3)
ecoli	0.7576	(2)	**0.7683**	(1)	0.7311	(3)
iris	**0.9134**	(1)	0.9107	(2)	0.9094	(3)
mammographic	**0.7945**	(1)	0.7937	(2)	0.7888	(3)
occupancy	**0.8788**	(1)	0.8783	(2)	0.8327	(3)
wine	**0.9611**	(1)	0.9610	(2)	0.9446	(3)
zoo	**0.9570**	(1.5)	**0.9570**	(1.5)	0.9477	(3)
dermatology	0.9340	(2)	**0.9449**	(1)	0.9145	(3)
ionosphere	0.9542	(2)	**0.9619**	(1)	0.9360	(3)
Avg. rank	–	**1.46**	–	1.54	–	3.00

*Best values are highlighted in bold.

To illustrate these results, Fig. 4 presents the aggregated PFs obtained by each MOEA across all runs for a representative dataset. As shown, the proposed complexity measure tends to generate PFs with smooth and regular shapes. This is important, as MOEA performance is known to degrade when dealing with MOPs characterized by discontinuous, degenerate, or inverted PFs [27]. In contrast, the adverse effects of using

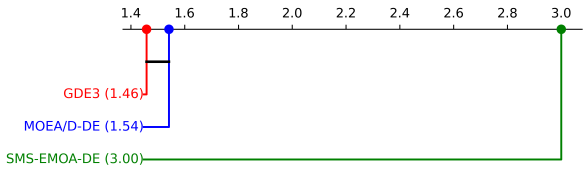


Fig. 3. CD diagram for the HV results shown in Table IV.

node count as a complexity metric are evident in Fig. 5, which induced a discontinuity effect on the PFs. The convergence and diversity capabilities of MOEA/D-DE were particularly affected under that formulation. Although the main objective of the proposed metric is to prevent these irregularities, its success could be dataset-dependent and may not completely overcome these effects in all scenarios.

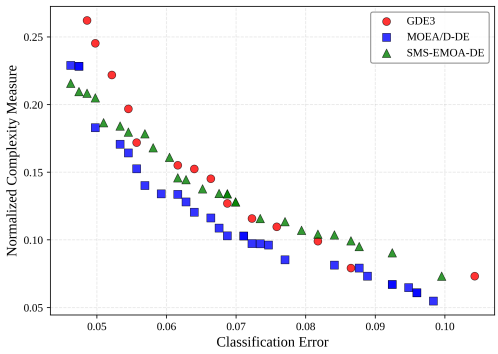


Fig. 4. Aggregated PFs obtained by MOEAs on the *biodeg* dataset using the proposed complexity metric.

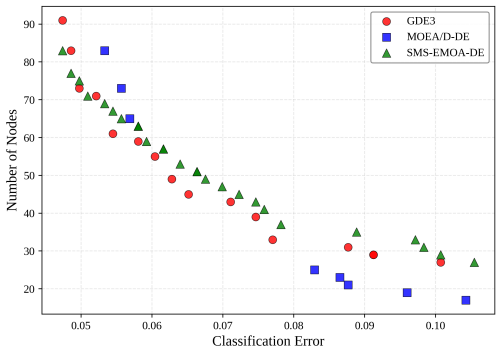


Fig. 5. Aggregated PFs obtained by MOEAs on the *biodeg* dataset using node count as complexity metric.

Table V reports the average accuracy achieved by the compared methods. For reference, the jSO method [14] is included as a single-objective baseline, where DT size is handled indirectly through a penalty term in the objective function. Although MOEA/D-DE achieves the best average rank, the Friedman test ($p = 0.1577$) indicates no statistically significant differences. Conversely, Table VI presents the results for DT size, measured by the number of leaf nodes in the induced oblique DTs. In this case, the Friedman test reveals significant differences ($p = 0.0001$). Consequently, the Nemenyi post-hoc test was applied, and the corresponding CD diagram is shown in Fig. 6. As observed, MOEA/D-DE obtained the best average ranking in terms of DT size, followed by GDE3, with no statistically significant differences between them. In contrast, SMS-EMOA-DE does not improve complexity reduction compared to jSO.

TABLE V
ACCURACY OBTAINED BY THE COMPARED METHODS ON THE TEST SET.*

Dataset	GDE3		MOEA/D-DE		SMS-EMOA-DE		jSO	
	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank
australian-credit	84.23	(4)	85.10	(3)	85.39	(2)	**86.26**	(1)
banknote-auth	98.99	(4)	99.12	(3)	99.36	(2)	**99.83**	(1)
biodeg	83.63	(2)	**84.38**	(1)	83.52	(3)	82.81	(4)
cryotherapy	84.63	(3)	88.33	(2)	83.89	(4)	**89.81**	(1)
ecoli	83.38	(2)	**83.43**	(1)	82.25	(4)	83.19	(3)
iris	95.22	(3)	95.56	(2)	93.89	(4)	**96.67**	(1)
mammographic	**84.28**	(1)	84.18	(2)	82.95	(4)	83.59	(3)
occupancy	98.11	(3)	98.32	(2)	**98.39**	(1)	98.07	(4)
wine	94.44	(2)	**95.28**	(1)	92.59	(4)	93.43	(3)
zoo	88.10	(2)	87.94	(3)	86.67	(4)	**90.16**	(1)
dermatology	**91.85**	(1)	91.71	(2)	88.29	(3)	87.48	(4)
ionosphere	**87.89**	(1)	86.90	(2)	85.40	(3)	85.21	(4)
Avg. rank	–	2.33	–	**2.00**	–	3.17	–	2.50

*Best values are highlighted in bold.

TABLE VI
DT SIZE OBTAINED BY THE COMPARED METHODS ON THE TEST SET.*

Dataset	GDE3		MOEA/D-DE		SMS-EMOA-DE		jSO	
	Size	Rank	Size	Rank	Size	Rank	Size	Rank
australian-credit	**3.80**	(1.0)	4.03	(2.0)	5.93	(4.0)	5.43	(3.0)
banknote-auth	2.07	(2.0)	**2.03**	(1.0)	3.23	(4.0)	2.93	(3.0)
biodeg	18.27	(2.0)	**16.87**	(1.0)	21.10	(3.0)	24.00	(4.0)
cryotherapy	2.10	(2.0)	**2.03**	(1.0)	2.17	(3.0)	5.40	(4.0)
ecoli	**9.37**	(1.0)	10.00	(2.0)	11.77	(3.0)	12.23	(4.0)
iris	**3.00**	(2.5)	**3.00**	(2.5)	**3.00**	(2.5)	**3.00**	(2.5)
mammographic	2.97	(2.0)	3.00	(3.0)	3.10	(4.0)	**2.67**	(1.0)
occupancy	4.33	(2.0)	**4.27**	(1.0)	6.57	(4.0)	4.40	(3.0)
wine	**3.00**	(1.5)	**3.00**	(1.5)	3.43	(3.0)	4.37	(4.0)
zoo	**7.00**	(2.0)	**7.00**	(2.0)	**7.00**	(2.0)	7.17	(4.0)
dermatology	8.27	(2.0)	**7.17**	(1.0)	10.07	(3.0)	12.60	(4.0)
ionosphere	6.77	(2.0)	**5.60**	(1.0)	8.07	(3.0)	8.90	(4.0)
Avg. Rank	-	1.83	-	**1.58**	-	3.21	-	3.38

*Best values are highlighted in bold.

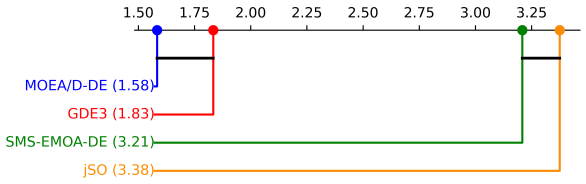


Fig. 6. CD diagram for the DT size results shown in Table VI.

In summary, the evidence presented in this section makes it possible to establish that: 1) Pareto-based and decomposition-based approaches are more effective for maximizing HV

compared to the indicator-based method; however, SMS-EMOA-DE has the drawback of producing only one solution per generation [22]; 2) the proposed normalized complexity measure facilitates the generation of smooth and regular PFs, helping to prevent performance degradation in MOEAs; 3) the MODE framework is capable of inducing oblique DTs with competitive accuracy compared to the single-objective approach; and 4) the proposed approach enables MOEA/D-DE and GDE3 to consistently induce more compact oblique DTs, leading to a significant reduction in model complexity without sacrificing accuracy.

VI. Conclusions and Future Work

This work addressed the problem of inducing oblique DTs under a multi-objective framework, using DE as the main search engine to jointly minimize classification error and DT size. Additionally, a normalized complexity metric was introduced with the aim of mitigating discontinuities in the PFs. Experimental results across datasets showed that the proposed approach yields competitive accuracy compared to the single-objective baseline. In particular, GDE3 and MOEA/D-DE proved especially effective under the proposed formulation, achieving a superior convergence–diversity performance and a significant reduction in DT size. Future work will focus on further analyzing the effects and advantages of the proposed complexity objective, as well as extending the experimental validation by considering a larger and more diverse collection of datasets. Moreover, runtime evaluation will be conducted, and mechanisms to potentially enhance interpretability will be investigated.

Acknowledgment

The first author (CVU 2055218) gratefully acknowledges the scholarship provided by the Mexican Secretariat of Science, Humanities, Technology and Innovation (SECIHTI) to pursue postgraduate studies at the University of Veracruz.

References

[1] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*, 2nd ed. USA: World Scientific Publishing Co., Inc., 2014.

[2] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[3] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[4] E. Cantú-Paz and C. Kamath, "Inducing oblique decision trees with evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 54–68, 2003.

[5] D. Heath, S. Kasif, and S. Salzberg, "Induction of oblique decision trees," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann, 1993, pp. 1002–1007.

[6] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, "Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence," *Information Fusion*, vol. 99, p. 101805, 2023.

[7] D. Dumitrescu and A. Joó, "Generalized decision trees built with evolutionary techniques," *Analele Universităţii din Timişoara. Seria Matematică-Informatică*, vol. 42, 01 2004.

[8] M. Kretowski and M. Grzes, "Global induction of oblique decision trees: An evolutionary approach," in *Intelligent Information Processing and Web Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 309–318.

[9] ——, "Mixed decision trees: An evolutionary approach," in *Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 260–269.

[10] J. B. Gray and G. Fan, "Classification tree analysis using target," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1362–1372, 2008.

[11] B. Vukobratović and R. Struharik, "Evolving full oblique decision trees," in *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, 2015, pp. 95–100.

[12] R. A. Lopes, A. R. R. Freitas, R. C. P. Silva, and F. G. Guimarães, "Differential evolution and perceptron decision trees for classification tasks," in *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, H. Yin, J. A. F. Costa, and G. Barreto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 550–557.

[13] R. Rivera-Lopez and J. Canul-Reich, "A global search approach for inducing oblique decision trees using differential evolution," in *Advances in Artificial Intelligence*, M. Mouhoub and P. Langlais, Eds. Cham: Springer International Publishing, 2017, pp. 27–38.

[14] R. Rivera-López, E. Mezura-Montes, J. Canul-Reich, and M.-A. Cruz-Chávez, "An experimental comparison of self-adaptive differential evolution algorithms to induce oblique decision trees," *Mathematical and Computational Applications*, vol. 29, no. 6, 2024. [Online]. Available: https://www.mdpi.com/2297-8747/29/6/103

[15] M. C. J. Bot, "Improving induction of linear classification trees with genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds. Las Vegas, Nevada, USA: Morgan Kaufmann, 2000, pp. 403–410.

[16] M. Chabbouh, S. Bechikh, C.-C. Hung, and L. Ben Said, "Multi-objective evolution of oblique decision trees for imbalanced data binary classification," *Swarm and Evolutionary Computation*, vol. 49, pp. 1–22, 2019.

[17] K. Price, R. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Berlin, Heidelberg: Springer-Verlag, 2005.

[18] M. Kretowski, "An evolutionary algorithm for oblique decision tree induction," in *Artificial Intelligence and Soft Computing - ICAISC 2004*, L. Rutkowski, J. H. Siekmann, R. Tadeusiewicz, and L. A. Zadeh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 432–437.

[19] S. Bandaru, A. H. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part a - survey," *Expert Systems with Applications*, vol. 70, pp. 139–159, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417416305449

[20] S. Kukkonen and J. Lampinen, "Gde3: the third evolution step of generalized differential evolution," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 443–450 Vol.1.

[21] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[22] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[23] A. Benitez-Hidalgo, A. J. Nebro, J. Garcia-Nieto, I. Oregi, and J. D. Ser, "jmetalpy: a python framework for multi-objective optimization with metaheuristics," 2019.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository," https://archive.ics.uci.edu.

[26] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, and M. Birattari, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[27] Y. Hua, Q. Liu, K. Hao, and Y. Jin, "A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 303–318, 2021.