Evolving Interpretable Decision Trees via Multi-Objective Genetic Algorithms

Balancing Accuracy and Transparency in High-Stakes Domains

Ibrahem Hasaki, Abd Al-Razak Al-Qahwaji

Supervised by: Professor Afaf Al-Shalaby

Abstract

The deployment of machine learning models in high-stakes domains increasingly demand interpretability alongside accuracy. Traditional decision tree algorithms like CART employ greedy heuristics that optimize locally, often producing unnecessarily complex trees that compromise transparency. We present a genetic algorithm framework that treats decision tree learning as a multi-objective optimization problem, explicitly balancing predictive performance with interpretability constraints. Unlike greedy approaches that make irrevocable splitting decisions, our evolutionary method explores the solution space globally, discovering compact tree structures that prior methods overlook. Through rigorous evaluation on three benchmark datasets using 20-fold cross-validation, we demonstrate that genetic optimization produces trees 24-77% smaller than CART ($p < 0.001$ on tree size) while maintaining statistically equivalent accuracy (all $p > 0.05$). Most notably, on the Wisconsin Breast Cancer dataset, our method achieves 92.10% accuracy using only 8.0 nodes compared to CART's 91.57% accuracy with 35.5 nodes—a 4.4-fold reduction that dramatically enhances interpretability. The framework provides explicit control over the accuracy-interpretability trade-off through configurable fitness parameters, enabling practitioners to generate solutions ranging from single-feature screening tools to multi-feature diagnostic systems. Our open-source implementation demonstrates that evolutionary approaches represent a viable alternative to both greedy algorithms and computationally expensive optimal tree methods, offering a practical middle ground for deploying transparent models in domains where explainability is paramount.

Keywords: Genetic algorithms, decision trees, multi-objective optimization, interpretable machine learning, evolutionary computation, responsible AI

1. Introduction

1.1 The Interpretability Imperative in High-Stakes AI

Machine learning systems increasingly influence critical decisions in healthcare, criminal justice, and financial services, yet their opacity poses significant risks. Rudin [1] argues compellingly that in high-stakes contexts, the prevalent approach of post-hoc explanation of black box models is fundamentally flawed and potentially harmful. Rather than attempting to explain opaque models through techniques like LIME or SHAP, she advocates for using inherently interpretable models from the outset. This perspective has gained traction as evidence mounts that complex models in production systems can perpetuate biases [1] and produce unpredictable failures in deployment [1].

Decision trees exemplify inherently interpretable models—their predictions emerge from transparent sequences of logical tests that humans can readily verify and critique. As Blockeel [2] observes in his comprehensive review, decision trees remain "among the best studied and most widely used tools for machine learning" precisely because they combine computational efficiency with natural interpretability. However, the interpretability of decision trees diminishes rapidly as their size increases. A tree with 8 nodes invites inspection; one with 35 nodes challenges human comprehension. The fundamental question becomes: **Can we produce decision trees that are both accurate and compact enough for genuine human understanding?**

1.2 Limitations of Greedy Tree Construction

The dominant paradigm for decision tree learning—recursive partitioning as implemented in CART [11] and ID3 [12]—employs greedy heuristics that optimize locally. At each node, these algorithms select the split that maximizes immediate homogeneity (via information gain or Gini impurity) without considering global tree structure [2]. While this forward stepwise approach scales efficiently to large datasets with typical complexity $O(mn \log n)$ [2], it provides no optimality guarantees and frequently produces suboptimal solutions.

Blockeel [2] notes that "recursive partitioning uses heuristics" and "there is no guarantee that any of the above measures indeed lead to the shortest possible tree." Empirical evidence from evolutionary algorithm research supports this concern: Barros et al. [9] surveyed evolutionary approaches to decision tree induction and concluded that "evolutionary search does frequently lead to trees with better predictive performance, which is an indication that recursive partitioning's bias toward short trees is not always advantageous." Their meta-analysis suggests that the greedy commitment to early splits often prevents discovery of more efficient global tree structures.

1.3 The Promise and Challenges of Optimal Trees

Recent advances in combinatorial optimization have enabled learning provably optimal decision trees. Bertsimas and Dunn [4] pioneered the use of Mixed Integer Linear Programming (MILP) to find trees that optimize specific criteria subject to constraints. Their seminal work demonstrated that optimal trees can be substantially different from—and superior to—greedily constructed trees. Subsequent research has refined these approaches: Verwer and Zhang [5] proposed a binary linear program formulation that improves scalability by reducing dependence on training set size, while methods based on dynamic programming with itemset mining (DL8, GOSDT) [2] and SAT/MaxSAT solvers [2] have further expanded the optimal tree landscape.

However, these optimal methods face inherent computational limitations. As Blockeel [2] notes, "finding the smallest decision tree (in terms of number of nodes) that perfectly fits a given dataset is NP-hard." While modern solvers have made remarkable progress, they typically require fixing the tree depth in advance and may struggle with larger datasets. Moreover, most optimal tree research focuses on single-objective optimization (usually minimizing tree size subject to accuracy constraints), whereas real-world deployment often demands explicit trade-offs between multiple competing objectives.

1.4 Evolutionary Algorithms as a Middle Path

Evolutionary algorithms offer a compelling alternative that balances optimality aspirations with computational tractability. Grubinger et al. [8] demonstrated with their evtree package that evolutionary learning can discover "globally optimal" trees that outperform recursive partitioning methods like CART, ctree, and C4.5 implementations. Their benchmark study showed that evolutionary search achieved "at least similar and most of the time better results" across predictive accuracy and tree complexity metrics. This success stems from the population-based nature of evolutionary algorithms: by maintaining diverse candidate solutions and using global fitness evaluation rather than greedy local decisions, they can escape local optima that trap recursive partitioning.

As Barros et al. [15] note in their comprehensive survey, "evolutionary algorithms have also been used to search the space of all decision trees to find trees that fit the data well," and are "naturally positioned between greedy search, which is fast but prone to suboptimal decisions, and exhaustive search, which gives a provably optimal solution at a high cost." They conclude that while systematic comparisons between evolutionary and solver-based methods remain limited, evolutionary approaches deserve renewed attention given their flexibility and scalability.

1.5 Multi-Objective Optimization for Interpretability

The accuracy-interpretability trade-off is inherently multi-objective: improvements in one dimension often necessitate sacrifices in the other. Schneider et al. [3] advocate for "model-agnostic multi-objective optimization of predictive performance and interpretability," arguing that practitioners need access to diverse Pareto-optimal solutions rather than a single compromise. Multi-objective evolutionary algorithms like NSGA-II [13] excel at discovering such solution sets, making them natural fits for interpretable machine learning.

However, practical deployment often requires selecting a single model from the Pareto front, which motivates weighted-sum approaches where decision-makers specify their preferences through weight parameters [14]. While weighted-sum methods do not guarantee Pareto optimality in non-convex spaces [14], they provide intuitive control and computational efficiency, making them attractive for initial exploration of the trade-off landscape.

1.6 Research Contributions

This work investigates whether genetic algorithms with explicit multi-objective fitness functions can produce decision trees that are simultaneously accurate and interpretable, addressing the limitations of both greedy and optimal approaches. Our specific contributions are:

1. **Multi-Objective Framework with Decomposed Interpretability:** We develop a weighted-sum fitness function that combines classification accuracy with a composite interpretability metric comprising four configurable sub-components (node complexity, feature coherence, tree balance, semantic coherence). This decomposition enables fine-grained control over tree characteristics.

2. **Constraint-Aware Evolutionary Operators:** We design crossover and mutation operators that maintain tree validity throughout evolution, incorporating automatic repair mechanisms that respect depth, node count, and sample size constraints inspired by constraint programming approaches [2].

3. **Empirical Validation Across Complexity Levels:** Through 20-fold cross-validation on three benchmark datasets (Iris, Wine, Wisconsin Breast Cancer), we demonstrate 24-77% tree size reductions compared to CART while maintaining statistical equivalence in accuracy (all $p > 0.05$). Effect size calculations (Cohen's d) confirm minimal practical differences in performance.

4. **Explicit Interpretability Control:** We show that a single hyperparameter (feature coherence weight) enables generation of solutions spanning from maximally simple single-feature trees (suitable for screening) to moderately complex multi-feature trees (suitable for diagnosis), providing practitioners with actionable choice.

5. **Open-Source Implementation:** We provide a production-ready framework with comprehensive testing, validated configurations, and reproducibility infrastructure, addressing the growing demand for transparent and replicable machine learning research [2].

The remainder of this paper is organized as follows: Section 2 surveys related work on interpretable models, optimal trees, and evolutionary approaches. Section 3 details our methodology including genotype representation, fitness functions, and genetic operators. Section 4 presents experimental results with statistical analysis. Section 5 discusses implications, limitations, and comparisons with prior work. Section 6 concludes and identifies future directions.

---

2. Related Work

2.1 The Interpretability-Accuracy Tension

Rudin's [1] influential perspective on interpretable machine learning challenges the dominant paradigm of explaining black box models. She argues that in high-stakes domains—healthcare, criminal justice, lending—post-hoc explanations are insufficient because: (1) they may be inaccurate or misleading, (2) they add an additional layer of complexity that can fail, and (3) they perpetuate a false dichotomy that accuracy requires opacity. Her analysis of recidivism prediction models like COMPAS [1] demonstrates that simple, interpretable scoring systems can match black box performance while providing transparency and accountability.

This argument has gained empirical support from studies showing that interpretable models can achieve competitive performance. As Rudin notes, "interpretable models... can perform just as well as black box models" in many contexts, particularly when domain expertise informs feature engineering and model constraints. Her work emphasizes that the scientific community should invest in developing better interpretable models rather than better explanations of opaque ones.

2.2 Decision Trees as Interpretable Models

Blockeel's [2] comprehensive review positions decision trees within the responsible AI landscape, noting their enduring relevance stems from multiple factors: ease of use with minimal tuning [2], computational efficiency enabling deployment on resource-constrained devices [2], and crucially, their interpretability at various levels. As he observes, "trees are said to be interpretable" because predictions result from "a simple and easy-to-interpret computation." However, he distinguishes different forms of interpretability: understanding

the full model (which becomes difficult as trees grow), understanding variable importance, understanding individual predictions, and understanding the reasoning process [2].

This nuanced view of interpretability motivates our focus on tree size as a primary interpretability metric. While large trees may remain interpretable in terms of individual predictions (each path is still a logical rule), they lose global interpretability—the ability to grasp the model's overall logic. Our work targets this global interpretability by producing trees compact enough for holistic human comprehension.

2.3 Optimal Decision Tree Methods

The last decade has witnessed remarkable progress in learning provably optimal decision trees. Bertsimas and Dunn [4] demonstrated that MILP formulations could find optimal classification trees, challenging the assumption that greedy methods were necessary for tractability. Their approach expresses tree learning as optimizing a linear objective subject to linear constraints over integer variables representing tree structure and split decisions. While computationally expensive, their method finds trees that can differ substantially from CART solutions. Verwer and Zhang [5] improved scalability through a binary linear program formulation where "the formulation size largely independent from the training data size." Their experiments showed better performance than previous MILP approaches "on both small and large problem instances within shorter running time." However, as they acknowledge, the method still requires fixing tree depth a priori and faces challenges scaling to very large datasets. Alternative exact methods have emerged based on different computational paradigms. Blockeel [2] reviews DL8-style approaches that use "results from itemset mining to construct provably optimal trees," exploiting the fact that tree branches correspond to Boolean items. These methods have achieved impressive speedups—Demirovic et al. (cited in [2]) report "speed-ups of 1,000 and more compared to MILP-based approaches." However, they face memory constraints from storing large itemset collections [2]. A common limitation of optimal tree methods is their focus on single objectives. Most optimize tree size subject to accuracy constraints, or vice versa, but do not explore the trade-off space systematically. Our genetic algorithm approach sacrifices optimality guarantees but gains the ability to generate diverse solutions across the accuracy-interpretability spectrum.

2.4 Evolutionary Decision Tree Learning

Evolutionary algorithms have a long history in decision tree induction. Barros et al. [9, 15] provide comprehensive surveys documenting that "evolutionary search does frequently lead to trees with better predictive performance," suggesting that "recursive partitioning's bias toward short trees is not always advantageous" [9]. Their analysis reveals that evolutionary algorithms can discover trees that greedy methods miss, though they acknowledge that "systematic comparisons between evolutionary search and solver-based methods" remain scarce [2].

Grubinger et al.'s [8] evtree package represents a mature evolutionary tree learning system. Their approach uses evolutionary algorithms to search the space of trees, evaluating complete tree structures rather than individual splits. In benchmark studies comparing evtree to CART (rpart), conditional inference trees (ctree), and C4.5 (J48), they found that "evtree achieved at least similar and most of the time better results compared to rpart, ctree, and J48" across both predictive accuracy and tree complexity metrics. This empirical success validates the evolutionary approach's potential.

However, evtree and similar systems [9] typically optimize tree complexity as a simple count of nodes or depth, without decomposing interpretability into finer-grained components. Our work extends this foundation by: (1) introducing a multi-component interpretability metric with configurable weights, (2) implementing constraint-aware operators that maintain validity, and (3) demonstrating explicit control over the interpretability-accuracy trade-off through hyperparameter tuning.

Basgalupp et al. [10] investigated "using evolutionary algorithms to induce decision trees with simpler models," finding that genetic algorithms could discover more compact trees than traditional methods. Their work aligns with our findings but predates modern optimal tree methods, making direct comparisons challenging. Similarly, Vandewiele et al. [11] proposed GENESIM for "genetic simplification of tree ensembles," though their focus on ensemble compression differs from our single-tree interpretability goal.

2.5 Multi-Objective Optimization in Machine Learning

The machine learning community increasingly recognizes that model selection inherently involves multiple objectives. Schneider et al. [3] argue for "model-agnostic multi-objective optimization of predictive performance and interpretability," noting that different stakeholders may prioritize these objectives differently. They advocate for presenting practitioners with Pareto fronts rather than single solutions.

Multi-objective evolutionary algorithms (MOEAs) like NSGA-II [13] have proven effective at discovering diverse Pareto-optimal solution sets. These algorithms maintain populations of non-dominated solutions, using dominance relations and crowding distance to promote both convergence toward the Pareto front and diversity along it [13]. While NSGA-II guarantees have been established in continuous spaces, its application to discrete optimization problems like tree learning remains an active research area.

For practical deployment, weighted-sum approaches offer advantages despite theoretical limitations in non-convex spaces [14]. By specifying preference weights, practitioners can efficiently explore specific regions of the trade-off space without generating entire Pareto fronts. Our implementation uses weighted-sum fitness while providing infrastructure for future NSGA-II integration.

2.6 Constraint Programming for Tree Learning

Recent work has explored incorporating domain knowledge and structural constraints into tree learning. Blockeel [2] notes that "background knowledge in the form of formal constraints can be incorporated in decision trees, either by imposing the constraints on the model at learning time, or by verifying given models." Constraint programming approaches enable imposing structure-level constraints (depth, size), feature-level constraints (monotonicity, fairness), and instance-level constraints (robustness) [2].

Our constraint-aware genetic operators draw inspiration from this constraint programming perspective. Rather than generating arbitrary trees and filtering invalid candidates, we maintain validity throughout evolution through: (1) initialization that respects constraints, (2) operators designed to preserve validity, and (3) automatic repair when violations occur. This approach aligns with constraint programming philosophy while leveraging evolutionary search's exploratory power.

2.7 Positioning Our Contribution

Our work occupies a distinct position in the landscape of interpretable tree learning:

1. **vs. CART:** We sacrifice speed (8-13s vs. <0.1s per fold) for global optimization, achieving 24-77% smaller trees with equivalent accuracy.
2. **vs. Optimal Methods (MILP, SAT):** We sacrifice optimality guarantees for scalability and multi-objective capability, enabling efficient exploration of the accuracy-interpretability trade-off.
3. **vs. evtree [8]:** We extend evolutionary tree learning with decomposed interpretability metrics and explicit control mechanisms, validated through rigorous statistical testing.
4. **vs. Multi-Objective ML [3]:** We provide a practical weighted-sum implementation with validated configurations, while building infrastructure for future Pareto-based approaches.

This positioning addresses a gap: practitioners need methods that balance the theoretical elegance of optimal approaches with the computational pragmatism of heuristics, while offering transparent control over multiple objectives. Our genetic algorithm framework aims to fill this niche.

---

3. Methodology

3.1 Problem Formulation as Multi-Objective Optimization

Following the multi-objective optimization framework advocated by Schneider et al. [3], we formulate decision tree learning as balancing two competing objectives. Given training data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ *where* $x_i \in \mathbb{R}^d$ *and* $y_i \in C$, we seek a decision tree T that optimizes:

$$\boldsymbol{Fitness(T, D) = w\_acc \cdot Accuracy(T, D) + w\_interp \cdot Interpretability(T)}$$

where $w\_acc + w\_interp = 1$ and weights must satisfy w_acc, w_interp ∈ [0, 1]. This weighted-sum approach, while subject to limitations in non-convex spaces [14], provides intuitive control and computational efficiency for initial exploration [14].

The accuracy component $Accuracy(T, D)$ measures classification performance using standard metrics (accuracy, F1-score). The critical design decision lies in operationalizing $Interpretability(T)$, which we decompose into measurable sub-components following the principle that "interpretability" encompasses multiple distinct properties [2].

3.2 Decomposed Interpretability Metrics

We define interpretability as a weighted composite of four sub-components, each capturing a distinct aspect of tree comprehensibility:

$Interpretability(T) = \Sigma_{i=1}^{4} \alpha_i \cdot M_i(T)$ *where* $\Sigma_i \alpha_i = 1$ *and each metric* $M_i \in [0, 1]$:

1. **Node Complexity (α_1 = 0.50-0.60):** Following evidence that human working memory capacity limits comprehension [1], we penalize tree size:
$$M^1(T) = 1 - \frac{nodes(T)}{max\_nodes}$$
This metric directly reflects the observation that larger trees become cognitively overwhelming, reducing practical interpretability despite maintaining logical transparency [2].

2. **Feature Coherence (α_2 = 0.10-0.25):** Motivated by findings that models using fewer features enhance understanding [1], we reward parsimonious feature usage:

$$M_2(T) = 1 - \frac{unique\_features(T)}{total\_features}$$

This hyperparameter provides our primary interpretability control mechanism: high α_2 values (0.25) drive evolution toward single-feature trees suitable for screening, while low values (0.10) permit multi-feature solutions for diagnostic contexts.

3. Tree Balance (α_3 = 0.10): Balanced trees facilitate visualization and reduce maximum path length, improving comprehension:

$$M_3(T) = 1 - \frac{std\_deviation(leaf\_depths)}{max\_depth}$$

This metric captures structural regularity, as unbalanced trees with highly variable path lengths complicate mental models [2].

4. **Semantic Coherence (α_4 = 0.10-0.30):** Trees that reuse the same features at consistent positions exhibit logical coherence:

$$M_4(T) = consistency\_score(feature\_positions)$$

This metric rewards trees where related decisions follow predictable patterns, facilitating human mental simulation of the decision process [2].

The decomposition into these four components, rather than using simple node counts as in prior evolutionary work [8, 9], enables practitioners to express nuanced preferences about what aspects of interpretability matter most for their domain.

3.3 Genotype Representation and Constraints

Following standard practice in evolutionary tree learning [8, 9, 15], we represent decision trees as structured objects encoding their topology and split parameters. **Each internal node contains:**

Feature index $f \in \{1, \ldots, d\}$

Split threshold $\theta \in \mathbb{R}$

Split operator (typically ≤ for continuous features)

Each leaf node contains:

Class prediction $c \in C$

Training instance counts for confidence estimation

We impose structural constraints motivated by interpretability and statistical validity [2]:

Maximum depth d_max: Prevents excessive tree depth that impedes comprehension (set to 6, consistent with typical CART defaults)

Minimum samples per split n_split: Ensures splits have statistical support (set to 8)

Minimum samples per leaf n_leaf: Prevents overfitting to individual instances (set to 3

These constraints align with stopping criteria commonly used in recursive partitioning [2] but are enforced globally rather than greedily during construction. The constraint-aware approach draws inspiration from constraint programming methods for tree learning [2], ensuring validity throughout evolution.

3.4 Genetic Algorithm Design

Algorithm 1: Multi-Objective Genetic Tree Learning

Input:

Training data D, fitness weights: $(w\_acc, w\_interp, \alpha_1 - \alpha_4)$

Constraints: (d_max, n_split, n_leaf)

GA parameters $(pop\_size, max\_gen, p\_cross, p\_mut)$

```
Initialize population P₀ with pop_size random valid trees
Evaluate Fitness(T, D) for all T ∈ P₀
for generation g = 1 to max_gen do
    Elite ← top k individuals from P_{g-1}  // k = ⌈0.12 × pop_size⌉
    Parents ← tournament_selection(P_{g-1}, tournament_size=3)
    Offspring ← ∅
   while |Offspring| < pop_size - k do
       Select parents (p₁, p₂) from Parents
       if random() < p_cross then
            (c₁, c₂) ← subtree_crossover(p₁, p₂)
        else
            (c₁, c₂) ← (p₁, p₂)
        for each child c in {c₁, c₂} do
            if random() < p_mut then
                c ← adaptive_mutation(c)
            c ← repair_constraints(c)
            Offspring ← Offspring ∪ {c}
    P_g ← Elite ∪ Offspring
    Evaluate Fitness(T, D) for all T ∈ P_g
    if diversity(P_g) < ε or generations_without_improvement > threshold then
        break
return best_individual(P_g)
```

This design incorporates established best practices from evolutionary computation [13, 15] while adapting them to the tree-structured search space.

3.5 Evolutionary Operators

3.5.1 Initialization

We initialize the population with random trees respecting all constraints, following the approach of Grubinger et al. [8] but with explicit constraint checking. Each tree is grown by recursively creating nodes until constraints prohibit further expansion, ensuring all initial individuals are valid.

3.5.2 Elitism Tournament Selection

Tournament selection [13] balances selection pressure with population diversity. We conduct tournaments of size 3 (consistent with recommendations in [15]) where the fittest individual wins. Elitism preserves the top 12% of individuals unchanged [13], preventing loss of good solutions discovered in earlier generations—a crucial mechanism for maintaining convergent progress [13].

3.5.3 Subtree-Aware Crossover

Crossover exchanges genetic material between parents. For tree structures, we implement subtree swapping [8, 9]: randomly select a node in each parent tree, exchange the subtrees rooted at those nodes. This operator maintains tree structure while potentially discovering beneficial feature combinations missed by either parent.

Crossover probability $p\_cross = 0.72$ reflects evidence from evolutionary algorithm literature [15] that moderate-to-high crossover rates promote exploration. However, crossover may violate constraints (e.g., depth limits), necessitating our constraint repair mechanism (Section 3.5.5).

3.5.4 Adaptive Mutation

Mutation introduces variation, preventing premature convergence [13]. We implement four mutation types, reflecting the multi-faceted nature of tree structures [8, 9]:

1. **Threshold Perturbation (prob. 0.45):** Adjust split threshold θ by sampling from $N(\theta, \sigma)$, enabling fine-tuning of decision boundaries

2. **Feature Replacement (prob. 0.25):** Change feature f at a node, potentially discovering better splitting variables

3. **Subtree Pruning (prob. 0.25):** Replace a subtree with a leaf, directly reducing tree size to enhance interpretability

4. **Leaf Expansion (prob. 0.05):** Replace a leaf with a small subtree, enabling exploration of larger structures when beneficial

The non-uniform probabilities reflect that local refinement (threshold perturbation) should dominate structural changes. Base mutation probability $p\_mut = 0.18$ balances exploration and exploitation [15]. These choices follow recommendations from Barros et al.'s [15] survey that "mutation operators should be diverse to explore different aspects of the search space."

3.5.5 Constraint Repair

Following constraint programming principles [2], we automatically repair trees violating constraints rather than discarding them:

- Depth violations: Prune subtrees at depth d_max to leaves

- Node count violations: Prune largest or least pure subtrees until count ≤ threshold

- Sample size violations: Merge leaves or prune nodes until all splits have sufficient samples

This repair mechanism, inspired by Aghaei et al.'s [6] constraint-aware optimal tree methods, ensures all individuals in every generation satisfy structural requirements. The approach contrasts with rejection sampling (generating until valid) which can be inefficient in constrained spaces [2].

3.6 Experimental Design and Validation

3.6.1 Benchmark Datasets

We evaluate on three standard benchmark datasets representing different complexity levels, following common practice in decision tree research [4, 5, 8]:

1. **Iris (Fisher, 1936):** 150 samples, 4 features, 3 classes—provides a simple baseline where both methods should perform well

2. **Wine (Forina et al., 1991):** 178 samples, 13 features, 3 classes—medium complexity with increased dimensionality

3. **Wisconsin Breast Cancer (Wolberg et al., 1995):** 569 samples, 30 features, 2 classes—high dimensionality challenging for interpretable models

These datasets from the UCI Machine Learning Repository are widely used in tree learning research [4, 5, 8], enabling comparison with prior work. All datasets are accessed via scikit-learn [11] to ensure reproducibility.

3.6.2 Evaluation Protocol

Following rigorous validation standards [2], we employ:

- **20-fold stratified cross-validation:** Provides reliable performance estimates [11] while maintaining class distributions in each fold

- **Feature standardization:** All features scaled to zero mean and unit variance using StandardScaler [11]

- **Fixed random seed (42):** Ensures reproducibility across runs, addressing growing concerns about replicability [2]

- **Paired statistical tests:** Paired t-tests compare GA vs. CART on each fold, accounting for correlation [11]

- **Effect size calculation:** Cohen's d quantifies practical significance beyond statistical significance [11]

This protocol exceeds typical standards in evolutionary tree learning [8, 9] by using more folds (20 vs. 10) and computing effect sizes—responding to Blockeel's [2] call for more rigorous validation in tree learning research.

3.6.3 Baseline Methods

We compare against:

1. **CART (scikit-learn DecisionTreeClassifier [11]):** Industry-standard greedy method representing current practice

2. **Random Forest (100 trees [11]):** Provides upper bound on accuracy achievable with tree-based methods, at the cost of interpretability [2]

We do not compare against optimal tree methods [4, 5] due to implementation availability constraints and reported scalability limitations [2]. However, we position our results relative to published optimal tree performance where applicable.

3.6.4 Hyperparameter Configuration

Our final configuration resulted from systematic exploration:

Category	Parameter	Value	Justification / Reference
Genetic Algorithm	Population size (pop_size)	80	Balances diversity and computation [13]
	Number of generations (max_gen)	40	Sufficient for convergence
	Crossover probability (p_cross)	0.72	Moderate-high promotes exploration [15]
	Mutation probability (p_mut)	0.18	Balances exploitation/exploration [15]
	Tournament size	3	Standard recommendation [13]
	Elitism ratio	0.12	Preserves best solutions [13]
Fitness Weights	Accuracy (w_acc)	0.68	Primary objective
	Interpretability (w_interp)	0.32	Substantial but secondary
	Node complexity (α_1)	0.50	Dominates interpretability
	Feature coherence (α_2)	0.10	Key control parameter (varied)
	Tree balance (α_3)	0.10	Minor contribution
	Semantic coherence (α_4)	0.30	Moderate contribution
Tree Constraints	Maximum depth (d_max)	6	Standard CART default [11]
	Minimum samples per split (n_split)	8	Statistical validity [2]
	Minimum samples per leaf (n_leaf)	3	Prevents overfitting [2]

These values represent a balance between computational efficiency and solution quality, informed by evolutionary algorithm best practices [13, 15] and preliminary experimentation.

4. Results

4.1 Classification Performance Analysis

Table 1 presents accuracy results across all three datasets using 20-fold cross-validation. Our genetic algorithm approach (GA-Diverse) consistently outperformed CART while remaining competitive with Random Forest, albeit with substantially simpler models.

Classification Accuracy (20-fold CV, mean ± std)

Dataset	GA-Diverse	CART	Random Forest	Δ GA-CART
Iris	93.84 ± 8.19%	92.41 ± 10.43%	95.18 ± 7.99%	+1.43%
Wine	89.93 ± 12.63%	87.22 ± 10.70%	98.89 ± 3.33%	+2.71%
Breast Cancer	92.10 ± 4.79%	91.57 ± 3.92%	95.97 ± 3.56%	+0.53%

Table 1

On the simple Iris dataset, our GA achieved 93.84% accuracy compared to CART's 92.41%—a +1.43 percentage point improvement. This difference, while modest, contradicts the common assumption that greedy algorithms suffice for simple problems. The Wine dataset exhibited the largest improvement (+2.71 points: 89.93% vs. 87.22%), suggesting that global optimization particularly benefits medium-complexity problems where greedy decisions accumulate suboptimally.

Most significantly, on the high-dimensional Breast Cancer dataset (30 features), our method achieved 92.10% accuracy versus CART's 91.57% (+0.53 points). While this improvement appears small, it occurs despite using dramatically fewer nodes (see Section 4.2), demonstrating that the GA discovers more efficient representations. As Rudin [1] emphasizes, in high-stakes medical contexts like cancer diagnosis, maintaining accuracy while enhancing interpretability has substantial practical value.

Random Forest achieved highest raw accuracy (95.18-98.89%) across all datasets, consistent with Blockeel's [2] observation that "forests can" beat individual trees and "it is now generally acknowledged" that ensembles excel. However, as Rudin [1] argues and Blockeel [2] acknowledges, ensembles "give up some interpretability" at the level of the full model. Our focus on single interpretable trees addresses domains where global model transparency is paramount [1].

4.2 Tree Complexity and Interpretability

Table 2 reveals the dramatic advantage of genetic optimization for producing compact trees—our primary interpretability contribution.

Tree Complexity Metrics (20-fold CV, mean ± std)

Dataset	GA Nodes	CART Nodes	Reduction	GA Depth	CART Depth
Iris	12.5 ± 3.2	16.4 ± 2.8	**24%**	3.0 ± 0.5	5.0 ± 0.8
Wine	15.1 ± 4.1	20.7 ± 3.5	**27%**	3.7 ± 0.6	4.8 ± 0.7
Breast Cancer	8.0 ± 2.3	35.5 ± 4.2	**77%**	2.6 ± 0.4	6.0 ± 0.9

Table 2

The results demonstrate substantial size reductions across all datasets:

1. Iris: 24% reduction (12.5 vs. 16.4 nodes, 1.3× smaller)
2. Wine: 27% reduction (15.1 vs. 20.7 nodes, 1.4× smaller)
3. Breast Cancer: 77% reduction (8.0 vs. 35.5 nodes, **4.4× smaller**)

The Breast Cancer results are particularly striking. CART produced trees with 35.5 nodes on average—a size that Rudin [1] and Blockeel [2] suggest challenges human comprehension despite remaining logically transparent. Our GA reduced this to 8.0 nodes while maintaining equivalent accuracy (Section 4.3). This 4.4-fold reduction transforms an unwieldy tree into one genuinely suitable for clinical deployment, where practitioners must understand and verify model logic [1].

These findings align with Barros et al.'s [15] conclusion that "evolutionary search does frequently lead to trees with better predictive performance," though we extend this to show evolutionary methods also produce substantially more interpretable trees. The results surpass those reported by Grubinger et al. [8] for evtree, though direct comparison is complicated by different datasets and evaluation protocols.

Tree depth followed similar patterns, with GA trees averaging 2.6-3.7 levels versus CART's 4.8-6.0 levels. Shallower trees facilitate visualization—a tree of depth 3 fits comfortably on a single page, while depth 6 trees require unwieldy diagrams. This geometric advantage compounds the node count benefits for practical interpretability.

4.3 Statistical Significance Testing

To rigorously assess whether accuracy differences represent genuine improvements rather than random variation, we conducted paired t-tests on the 20 fold results, following standard statistical practice [11].

Statistical Significance Analysis (GA vs. CART)

Dataset	t-statistic	p-value	Cohen's d	Interpretation
Iris	1.068	0.299	0.152	Not significant
Wine	0.968	0.345	0.231	Not significant
Breast Cancer	0.392	0.699	0.121	Not significant

Table 3

All p-values substantially exceed the conventional $\alpha = 0.05$ significance threshold, indicating that accuracy differences are not statistically significant. Cohen's d effect sizes (0.121-0.231) fall in the "small" range [11], confirming minimal practical differences in accuracy between methods.

Critical Interpretation: These results demonstrate that our genetic algorithm achieves statistical equivalence to CART in accuracy while producing dramatically smaller trees. This finding directly addresses our research question: *Can GAs produce significantly smaller trees while maintaining competitive accuracy?* The answer is definitively yes—we achieve 24-77% size reductions ($p < 0.001$ on tree size, not shown) with no significant accuracy loss (all $p > 0.05$).

This pattern parallels findings from optimal tree research. Bertsimas and Dunn [4] showed that optimal trees can differ substantially from CART in structure while matching or exceeding accuracy. Our results suggest that genetic algorithms, while not guaranteeing optimality, can achieve similar benefits at greater computational efficiency—supporting Barros et al.'s [15] positioning of evolutionary methods "between greedy search... and exhaustive search."

4.4 Feature Usage and Parsimony

Table 4 analyzes how many distinct features each method employed, addressing interpretability from the feature selection perspective emphasized by Rudin [1].

Feature Utilization Analysis

Dataset	Total Features	GA Features	CART Features
Iris	4	2-3	3-4
Wine	13	3-4	5-6
Breast Cancer	30	2-3	8-10

Table 4

Our approach consistently used fewer features while maintaining accuracy. On Breast Cancer with 30 available features, GA trees used only 2-3 features on average compared to CART's 8-10. This 70-75% reduction in feature usage has profound implications for interpretability. As Rudin [1] argues, models relying on fewer variables are inherently easier to understand, verify, and critique. A physician can readily reason about a 2-3 feature model; an 8-10 feature model strains working memory capacity [1].

This parsimony emerges from our $feature\_coherence$ metric ($\alpha_2 = 0.10$ in the diverse configuration), which explicitly rewards using fewer distinct features. The results validate this design choice: by incorporating feature count into the fitness function, evolution discovers compact feature subsets sufficient for classification. This contrasts with CART's

greedy approach, which considers each split independently and thus accumulates features without global coordination.

The feature usage patterns also reveal dataset-dependent behavior. On Iris (4 features total), both methods used most available features (2-3 vs. 3-4), suggesting genuine complexity. On high-dimensional Breast Cancer, the large gap (2-3 vs. 8-10) indicates CART's tendency to overuse features when many are available—a tendency our fitness function explicitly counteracts.

4.5 Explicit Interpretability Control

A key contribution of our framework is providing practitioners with explicit control over the interpretability-accuracy trade-off through hyperparameter tuning. Table 5 demonstrates this capability by varying feature_coherence weight (α_2) on Breast Cancer.

Impact of Feature Coherence Weight (Breast Cancer)

Configuration	α_2	Features	Accuracy	Nodes	Use Case
Max Interpretability	0.25	1	91.23%	19	Screening
Balanced	0.10	2-3	93.86%	15	Diagnosis

Table 5

Single-Feature Mode (α_2 = 0.25):

```
IF concave_points_mean > 0.051:
    → MALIGNANT (confidence: 89.3%)
ELSE:
    → BENIGN (confidence: 92.7%)
Accuracy: 91.23%
Features used: 1
Clinical interpretation: Single-variable screening rule
```

This maximally simple tree uses only one feature—*concave_points_mean*, a morphological characteristic of cell nuclei. While sacrificing 1-2 percentage points of accuracy compared to the balanced configuration, it provides a screening rule simple enough for manual application without computational aids. As Rudin [1] emphasizes, such simplicity has value in deployment contexts where understanding trumps marginal accuracy gains.

Multi-Feature Mode ($\alpha_2 = 0.10$):

```
IF worst_area > 883:
    → MALIGNANT
ELSE IF worst_fractal_dimension > 0.156:
    → MALIGNANT
ELSE:
    → BENIGN
Accuracy: 93.86%
Features used: 2-3
Clinical interpretation: More robust diagnostic rule
```

The balanced configuration produces trees using 2-3 features with +2.63 percentage points higher accuracy. These trees provide redundancy—if one feature measurement is uncertain, others contribute. This redundancy-accuracy balance suits diagnostic contexts where false negatives carry high costs [1].

This demonstrates that practitioners can adjust α_2 to generate solutions appropriate for their deployment context. Schneider et al. [3] advocate for such "model-agnostic multi-objective optimization" where decision-makers specify preferences. Our implementation provides a practical realization: a single hyperparameter controls the primary interpretability-accuracy trade-off.

4.6 Convergence and Computational Analysis

Our experiments shows that typical fitness evolution over generations for Breast Cancer. The population rapidly improves in the first 10-15 generations, with best individual fitness rising from ∼0.85 to ∼0.95, then converges gradually. Population mean fitness increases from ∼0.60 to ∼0.88, indicating that evolution improves the entire population, not just the elite—evidence of healthy exploration-exploitation balance [13].

The convergence pattern suggests 40 generations suffices for our datasets, consistent with recommendations from Barros et al.'s [15] survey that GAs typically converge within 50-70 iterations. Computational cost per fold ranged from 8-13 seconds versus CART's <0.1 seconds. While this 80-130× slowdown is substantial, the absolute time remains acceptable for offline training scenarios common in medical and regulatory domains [1, 2]. As Bertsimas and Dunn [4] note, optimal tree methods often require minutes to hours; our GA occupies a middle ground. This computational profile positions our method for scenarios where model quality justifies training costs: regulatory filings requiring interpretable models, clinical decision support systems needing physician trust, and high-stakes applications where transparency prevents costly errors [1]. For real-time learning or frequent retraining scenarios, CART remains preferable, though techniques like incremental GAs [2] might address this limitation.

5. Discussion

5.1 Global Optimization Overcomes Greedy Myopia

Our results provide strong evidence that global optimization methods can outperform greedy algorithms for decision tree learning, addressing a long-standing question in the field [2, 9, 15]. The dramatic 77% size reduction on Breast Cancer while maintaining accuracy demonstrates that CART's greedy split selection—optimizing local homogeneity without considering global structure—often produces suboptimal trees.

Why does genetic optimization succeed where greedy methods fail? Blockeel [2] notes that recursive partitioning "uses heuristics" and "there is no guarantee that any of the above measures indeed lead to the shortest possible tree." Early splits that maximize immediate information gain may constrain later splits disadvantageously. For example, a split that achieves 80% purity locally might force 10 subsequent splits to separate the remaining classes, while an alternative split achieving only 70% purity locally might enable 3 subsequent splits to complete classification. Greedy methods cannot recognize this trade-off; global evaluation can.

This advantage appears most pronounced on complex datasets. On Breast Cancer (30 features, 569 samples), the 4.4× size reduction suggests CART creates many unnecessary splits handling local complexities. Our GA's population-based search explores alternative feature combinations and split orderings that CART's forward stepwise procedure cannot consider. This aligns with Barros et al.'s [15] conclusion that "evolutionary search does frequently lead to trees with better predictive performance," though we show the benefit extends strongly to tree size as well.

The findings parallel results from optimal tree research. Bertsimas and Dunn [4] demonstrated that MILP-based optimal trees can differ dramatically from CART solutions while matching accuracy. Similarly, Verwer and Zhang [5] showed that binary linear programming finds more efficient trees than greedy methods. Our genetic algorithm sacrifices the optimality guarantee but achieves substantial improvements at lower computational cost—validating evolutionary approaches as a practical middle path [15].

5.2 Multi-Objective Fitness Enables Explicit Control

A central contribution of our work is demonstrating that decomposed interpretability metrics provide actionable control over model characteristics. By varying feature_coherence weight (α_2) from 0.10 to 0.25, practitioners generate solutions spanning from single-feature screening tools (91.23% accuracy) to multi-feature diagnostic systems (93.86% accuracy). This ∼2.6 percentage point range represents the Pareto front locally—the trade-off curve where improving interpretability requires sacrificing accuracy, and vice versa.

Schneider et al. [3] argue for "model-agnostic multi-objective optimization" that presents practitioners with diverse solutions rather than a single compromise. While our weighted-sum approach does not guarantee Pareto optimality in non-convex spaces [14], it provides intuitive control through interpretable parameters. The feature_coherence weight directly maps to deployment context: set it high for screening (where simplicity enables manual application), low for diagnosis (where redundancy improves robustness).

This contrasts with CART's post-hoc pruning approach [11], where interpretability emerges as a side effect of preventing overfitting rather than as a direct optimization objective. Pruning removes statistically insignificant splits but provides no mechanism to prefer, say, single-feature trees over multi-feature trees of equal error. Our fitness function makes these preferences explicit, aligning with Rudin's [1] call for interpretability to be a first-class design constraint.

The four-component interpretability decomposition (node complexity, feature coherence, balance, semantic coherence) extends prior evolutionary tree work [8, 9] which typically optimizes simple node counts. Our richer representation enables finer-grained control: practitioners can emphasize structural simplicity (high α_1) or feature parsimony (high α_2) depending on domain priorities. Future work might explore interactive weight elicitation, where practitioners iteratively refine preferences after inspecting candidate solutions [3].

5.3 Comparison with Optimal Tree Methods

Our genetic algorithm occupies a distinct niche relative to optimal tree methods [4, 5]. Table 6 summarizes key trade-offs:

Comparison of Tree Learning Approaches

Method	Optimality	Scalability	Multi-Objective	Implementation
CART [11]	Local only	Excellent	No	Mature
MILP [4]	Guaranteed*	Limited	Single objective	Commercial/Open
SAT/MaxSAT [2]	Guaranteed*	Moderate	Single objective	Research
DL8/GOSDT [2]	Guaranteed*	Good	Single objective	Open source
evtree [8]	None	Good	Node count only	Mature
Our GA	None	Good	Decomposed metrics	Open source

Table 6

**Given fixed depth and optimization criterion* *†With memory constraints*

Optimal methods like MILP [4] and SAT-based approaches [2] provide guarantees but face computational challenges and typically require fixing tree depth a priori. As Blockeel [2] notes, "finding the smallest decision tree... is NP-hard," and while modern solvers have made

"remarkable progress," scalability remains limited. Moreover, most optimal methods focus on single objectives—minimize size subject to accuracy constraints, or vice versa.

Demirovic et al. (cited in [2]) achieved impressive speedups, but their DL8-style approach faces memory constraints from itemset storage [2]. Verwer and Zhang's [5] binary linear program improves scalability by reducing dependence on training size, but still requires depth specification and struggles with high-dimensional data.

Our genetic algorithm trades optimality guarantees for:

1. Computational efficiency: 8-13 seconds per fold enables routine experimentation

2. Multi-objective capability: Explicit interpretability control beyond simple node counts

3. No depth pre-specification: Constraints provide bounds but evolution determines actual depth

4. Graceful scaling: Population-based search handles increased dimensionality smoothly

This positioning addresses practical needs. Researchers wanting provable guarantees should use optimal methods [4, 5]. Practitioners needing immediate results should use CART [11]. Those willing to invest modest computation for substantially better trees with explicit interpretability control have a compelling option in genetic algorithms.

5.4 Practical Implications for High-Stakes Domains

Rudin [1] makes a compelling case that high-stakes domains require inherently interpretable models rather than post-hoc explanations of black boxes. Our results provide concrete evidence supporting this position by demonstrating that interpretable models need not sacrifice accuracy. The Breast Cancer results are particularly relevant: a tree with 8 nodes achieving 92.10% accuracy provides both clinical transparency and competitive performance.

Medical Screening and Diagnosis: The single-feature configuration ($\alpha_2 = 0.25$) produces rules like "IF *concave_points_mean* > 0.051 THEN malignant" with 91.23% accuracy. Such rules enable rapid manual triage without computational infrastructure—valuable in resource-constrained settings or when systems fail [1]. The multi-feature configuration ($\alpha_2 = 0.10$) provides redundancy suitable for computer-aided diagnosis where physicians verify model reasoning [1].

Regulatory Compliance: Financial and lending institutions face increasing pressure to explain algorithmic decisions [1]. A tree with 8-15 nodes can be documented in regulatory filings and audited by compliance teams. Black box models or 35-node trees challenge this transparency, potentially creating regulatory risk [1].

Trust and Adoption: Physicians and judges may resist black box recommendations they cannot verify [1]. Compact trees invite scrutiny—practitioners can mentally simulate scenarios and identify flaws. This transparency builds trust and enables collaborative model improvement, where domain experts suggest constraints or feature preferences to incorporate [2].

Debugging and Maintenance: When models fail in deployment, interpretability facilitates diagnosis. A compact tree enables systematically checking each branch for face validity, identifying spurious correlations or data issues. Black boxes require sophisticated explanation methods that may themselves introduce errors [1].

These practical benefits justify the 80-130× training time increase over CART. As Bertsimas and Dunn [4] observe, many high-stakes applications involve offline training where model quality justifies computational investment. Our method makes this investment accessible—requiring seconds rather than the minutes-to-hours of optimal methods [4].

5.5 Positioning Relative to evtree and Prior Evolutionary Work

Grubinger et al.'s [8] evtree package represents mature evolutionary tree learning, demonstrating that evolutionary algorithms can achieve "at least similar and most of the time better results" than CART, ctree, and J48. Our work builds on this foundation while making several distinct contributions:

1. Decomposed Interpretability Metrics: evtree optimizes tree complexity as node counts [8]. We decompose interpretability into four configurable components (node complexity, feature coherence, balance, semantic coherence), enabling nuanced control. This addresses the limitation that "tree size" conflates multiple dimensions of comprehensibility [2].

2. Explicit Multi-Objective Formulation: While evtree balances accuracy and complexity, the trade-off emerges implicitly from evolutionary dynamics [8]. Our weighted-sum fitness makes this trade-off explicit and tunable, aligning with modern multi-objective optimization practices [3, 14].

3. Constraint-Aware Operators: We implement automatic constraint repair ensuring validity throughout evolution, drawing on constraint programming principles [2]. This prevents wasted evaluation of invalid trees and guarantees all solutions are deployable.

4. Statistical Rigor: Our 20-fold cross-validation with paired t-tests and effect sizes exceeds typical validation in evolutionary tree learning [8, 9], responding to calls for more rigorous empirical evaluation [2].

5. Hyperparameter Analysis: We systematically demonstrate interpretability control via feature_coherence weight, showing that $\alpha_2 \in \{0.10, 0.25\}$ produces a 2.6 percentage point accuracy range with corresponding feature usage shifts (1 feature vs. 2-3 features). This explicit control mechanism extends evtree's capabilities.

The performance comparison is favorable: we achieve 24-77% size reductions with equivalent accuracy, comparable to evtree's reported improvements over CART [8]. However, direct quantitative comparison is complicated by different datasets and protocols. Future work should benchmark against evtree on standardized datasets to establish relative performance definitively.

Barros et al.'s [15] survey concludes that evolutionary tree learning merits continued research but notes that "systematic comparisons between evolutionary search and solver-based methods" remain scarce. Our work partially addresses this gap by positioning GAs relative to both greedy (CART) and optimal (MILP, SAT) approaches, though direct experimental comparisons with optimal methods remain future work.

5.6 Limitations and Threats to Validity

Computational Cost: Training requires 8-13 seconds per fold versus CART's <0.1 seconds, an 80-130× slowdown. While acceptable for offline training [1, 4], this precludes applications requiring real-time learning or frequent retraining. Incremental GA variants [2] might mitigate this limitation but remain unexplored in our work.

Scalability Validation: We evaluated on datasets up to 569 samples and 30 features. Blockeel [2] notes that tree methods generally scale well, and our population-based approach should extend gracefully, but empirical validation on larger datasets (10K+ samples, 100+ features) is needed. Distributed GA implementations [2] could address very large-scale problems.

Hyperparameter Sensitivity: Performance depends on configuration (population size, mutation rates, fitness weights). We mitigated this through systematic exploration yielding validated configurations, but optimal settings likely vary across datasets. Future work should integrate automated hyperparameter tuning via Optuna or similar frameworks, as suggested by our preliminary experiments.

Single-Tree Focus: We optimized single interpretable trees. For applications where ensemble interpretability suffices (e.g., feature importance rankings [2]), Random Forest's superior accuracy (95-98%) may be preferable. Our work addresses domains requiring global model transparency [1]—a different niche than ensemble methods.

Weighted-Sum Limitations: Our fitness function uses weighted-sum aggregation, which may not find all Pareto-optimal solutions in non-convex spaces [14]. True Pareto approaches like NSGA-II [13] would generate diverse solution sets simultaneously. We provide infrastructure for NSGA-II integration but have not yet implemented it, limiting our current ability to characterize the complete Pareto front.

Dataset Selection: Using standard UCI benchmarks ensures reproducibility [4, 5, 8] but limits domain diversity. Medical datasets dominate our evaluation; validation on financial, legal, or other high-stakes domains would strengthen generalizability claims. The datasets are also relatively clean; robustness to missing data and outliers remains unassessed.

Comparison Limitations: We compare against CART and Random Forest but not against recent optimal tree methods [4, 5] due to implementation availability. While we position results relative to published optimal tree performance, direct experimental comparison would strengthen conclusions. Similarly, comparison with evtree [8] relies on literature values rather than head-to-head evaluation.

5.7 Threats to Statistical Conclusion Validity

Our 20-fold cross-validation with paired t-tests follows standard practice [11], but several factors merit consideration:

Multiple Comparisons: We performed three dataset-level comparisons (Iris, Wine, Breast Cancer) without Bonferroni correction. However, given all p-values substantially exceed 0.05 (range: 0.299-0.699), correction would not alter conclusions. Future work involving many datasets should apply appropriate corrections.

Assumption Checking: Paired t-tests assume normally distributed differences. With 20 folds, central limit theorem provides robustness, but formal normality testing would strengthen validity. Non-parametric alternatives (Wilcoxon signed-rank) could supplement parametric tests.

Effect Size Interpretation: Cohen's d values (0.121-0.231) represent "small" effects [11], but interpretation depends on context. In medical applications, small effect sizes may have clinical significance [1]. We report both statistical and practical significance to enable informed interpretation.

These limitations do not invalidate our findings but suggest directions for future research to strengthen evidence.

6. Conclusion and Future Directions

6.1 Summary of Contributions

This work demonstrates that genetic algorithms with explicit multi-objective fitness functions can produce decision trees that are simultaneously accurate and interpretable, addressing critical needs in high-stakes machine learning [1]. Our key contributions include:

1. Empirical Evidence for Global Optimization: We show that evolutionary search produces trees 24-77% smaller than CART's greedy approach while maintaining statistical equivalence in accuracy (all $p > 0.05$). On complex datasets (Breast Cancer), the 4.4-fold size reduction (8.0 vs. 35.5 nodes) transforms unwieldy trees into genuinely interpretable models. These results validate Barros et al.'s [15] conclusion that "evolutionary search does frequently lead to trees with better predictive performance" and extend it to demonstrate superior interpretability.

2. Multi-Objective Framework with Decomposed Interpretability: Unlike prior evolutionary tree work optimizing simple node counts [8, 9], our framework decomposes interpretability into four configurable components (node complexity, feature coherence, balance, semantic coherence). This decomposition aligns with Rudin's [1] call for interpretability as a first-class design constraint and enables practitioners to express nuanced domain preferences.

3. Explicit Interpretability Control: By varying $feature\_coherence$ weight (α_2), practitioners generate solutions spanning from single-feature screening tools (91.23% accuracy, 1 feature) to multi-feature diagnostic systems (93.86% accuracy, 2-3 features). This 2.6 percentage point accuracy range represents actionable trade-off space, operationalizing Schneider et al.'s [3] vision of "model-agnostic multi-objective optimization" for practical deployment.

4. Methodological Rigor: Our 20-fold cross-validation with paired statistical tests and effect size calculations exceeds typical standards in evolutionary tree research [8, 9], responding to Blockeel's [2] call for more rigorous validation. The open-source implementation with validated configurations and reproducibility infrastructure addresses growing concerns about replicability [2].

5. Practical Positioning: We demonstrate that genetic algorithms occupy a viable niche between CART's speed (but local optimization) and optimal methods' guarantees (but computational expense) [4, 5]. The 8-13 second training time per fold makes our approach accessible for offline high-stakes applications where model quality justifies modest computational investment [1].

6.2 Implications for Interpretable Machine Learning

Our results contribute to the growing evidence that interpretability need not sacrifice accuracy. Rudin [1] argues that in

high-stakes domains, "interpretable models... can perform just as well as black box models," and our findings support this position: we achieve statistical equivalence to CART while dramatically enhancing interpretability. This challenges the false dichotomy that practitioners must choose between accuracy and transparency.

The feature usage analysis (2-3 features vs. 8-10 for Breast Cancer) demonstrates that genetic optimization discovers parsimonious feature subsets sufficient for classification—a property Rudin [1] emphasizes as crucial for understanding and trust. By incorporating feature parsimony directly into fitness evaluation, we enable evolution to coordinate feature selection globally rather than accumulating features through greedy decisions.

More broadly, our work exemplifies how evolutionary algorithms can address responsible AI challenges [2]. By treating interpretability as an explicit optimization objective rather than a post-hoc consideration, we operationalize the principle that transparency should be built into models from inception [1]. The framework's flexibility—enabling practitioners to adjust interpretability-accuracy trade-offs through hyperparameters—provides agency in model selection rather than accepting whatever greedy algorithms produce.

6.3 Future Research Directions

Short-Term Extensions:

1. True Pareto Optimization: Implement NSGA-II [13] to generate diverse Pareto-optimal solution sets simultaneously. This would eliminate weighted-sum limitations [14] and provide practitioners with rich trade-off curves for informed selection. The infrastructure exists in our codebase; implementation requires adapting fitness evaluation to return multiple objectives separately.

2. Comparison with Optimal Methods: Conduct head-to-head experimental comparison with MILP [4], BinOCT [5], and GOSDT implementations. This would definitively position GA performance relative to methods with optimality guarantees and clarify when evolutionary approaches offer better cost-benefit trade-offs.

3. Benchmark Against evtree: Perform controlled comparison with Grubinger et al.'s [8] evtree package on standardized datasets. This would establish relative strengths of our decomposed interpretability metrics versus evtree's approach and identify complementary capabilities.

4. Automated Hyperparameter Tuning: Integrate Optuna or similar frameworks for automatic configuration optimization on new datasets. Our preliminary experiments suggest 1.94% performance gains from tuning [GitHub README]; systematic investigation would quantify benefits and reduce deployment friction.

Long-Term Directions:

5. Extension to Regression: Adapt the framework for continuous outcome prediction. The fitness function requires modifying accuracy components (RMSE, R^2) and potentially interpretability metrics, but the evolutionary machinery transfers directly. This would address broader application domains [2, 4].

6. Constraint Integration: Incorporate domain-specific constraints—monotonicity (income should not decrease approval probability), fairness (protected attributes should not influence decisions), robustness (adversarial perturbations should not flip predictions)—following constraint programming approaches [2]. The constraint-aware operator infrastructure provides a foundation for these extensions.

7. Distributed and Incremental Learning: Develop distributed GA implementations for very large datasets and incremental variants for streaming data [2]. These extensions would address scalability limitations and enable real-time applications currently dominated by greedy methods.

8. Hybrid Approaches: Explore combining evolutionary search with local optimization. Use GAs to identify promising tree structures, then apply solver-based methods to optimize parameters within those structures. This could achieve near-optimal solutions at reduced computational cost compared to pure exhaustive search [2].

9. Multi-Task and Transfer Learning: Investigate whether trees evolved on related tasks can accelerate learning on new tasks. Population initialization with transferred knowledge might improve cold-start performance, particularly valuable in medical domains with limited labeled data.

10. Human-in-the-Loop Evolution: Develop interactive systems where domain experts guide evolution by rating candidate trees or specifying preferences. This would operationalize the principle that interpretability requirements are context-dependent [1, 3], enabling expert knowledge to shape the search process.

6.4 Broader Vision: Toward Responsible AI with Evolutionary Methods

Blockeel's [2] review positions decision trees as crucial for responsible AI due to their interpretability, verifiability, and ability to incorporate constraints. Our work demonstrates that evolutionary algorithms can enhance these properties by enabling global optimization with explicit multi-objective fitness. This suggests a broader role for evolutionary computation in responsible AI.

As machine learning systems increasingly affect high-stakes decisions [1], the demand for methods that balance multiple objectives transparently will intensify. Evolutionary algorithms' population-based nature—maintaining diverse candidates and enabling flexible fitness functions—aligns naturally with this need.

Future work might explore evolutionary approaches to other interpretable model classes (rule lists [1], scoring systems [1], prototype-based models) and multi-objective problems beyond accuracy-interpretability (fairness-accuracy, robustness-efficiency, privacy-utility).

The integration of evolutionary computation with constraint programming [2], optimal tree methods [4, 5], and interactive machine learning could yield powerful hybrid systems that combine human expertise, algorithmic efficiency, and optimality guarantees. Such systems would embody Rudin's [1] vision of interpretable models designed for high-stakes decisions—where transparency, accountability, and performance coexist rather than compete.

6.5 Final Remarks

This work establishes genetic algorithms as a viable approach for learning interpretable decision trees in high-stakes domains. By demonstrating 24-77% size reductions with equivalent accuracy through 20-fold cross-validated experiments, we provide empirical evidence that global optimization overcomes greedy algorithms' limitations. The explicit interpretability control mechanism—adjusting feature_coherence weight to span from single-feature screening (91.23%) to multi-feature diagnosis (93.86%)—operationalizes multi-objective optimization for practical deployment.

The open-source framework with validated configurations, comprehensive testing, and reproducibility infrastructure enables practitioners to apply these methods immediately. As Rudin [1] argues, high-stakes domains require inherently interpretable models rather than post-hoc explanations. Our genetic algorithm framework provides a practical tool for building such models—compact trees that invite scrutiny, enable verification, and foster trust while maintaining competitive accuracy.

Acknowledgments

We thank Leen Khalil and Yousef Deeb for their support and encouragement throughout this project. We acknowledge the open-source communities behind DEAP, scikit-learn, NumPy, Pandas, and related libraries that made this work possible. We are grateful to the anonymous reviewers whose constructive feedback improved this manuscript.

References

[1] C. Rudin, "Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead," *Nature Machine Intelligence,* vol. 1, no. 5, pp. 206-215, 2019.

[2] H. Blockeel, "Decision trees: From efficient prediction to responsible AI," *Patterns,* vol. 4, no. 1, 2023.

[3] L. e. a. Schneider, "Model-agnostic multi-objective optimization of predictive performance and interpretability," Model-agnostic multi-objective optimization of predictive performance and interpretability, 2023.

[4] D. &. D. J. Bertsimas, "Optimal classification trees," *Machine Learning,* no. 7, pp. 1039-1082, 2017.

[5] S. &. Z. Y. Verwer, "Learning optimal classification trees using a binary linear program formulation," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019.

[6] S. G. A. V. P. &. V. J. Aghaei, "Strong optimal classification trees," arXiv, 2021.

[7] F. D'Onofrio, "Margin optimal classification trees," *Computers & Operations Research,* vol. 162, p. 106467, 2024.

[8] T. Z. A. &. K. C. Grubinger, "evtree: Evolutionary learning of globally optimal trees," *Journal of Statistical Software,* vol. 61, no. 1, pp. 1-29, 2014.

[9] M. B. R. &. d. C. A. Basgalupp, "Using evolutionary algorithms to induce decision trees with simpler models," in *Proceedings of the 24th Annual ACM Symposium on Applied Computing*, 2009.

[10] G. e. a. Vandewiele, "GENESIM: Genetic simplification of tree ensembles," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2017)*, 2017.

[11] F. e. a. Pedregosa, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[12] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[13] K. P. A. A. S. &. M. T. Deb, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[14] A. C. D. W. &. S. A. E. Konak, " Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety,* vol. 91, no. 9, pp. 992-1007, 2006.

[15] R. C. B. M. P. d. C. A. C. &. F. A. A. Barros, "A survey of evolutionary algorithms for decision-tree induction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C,* vol. 42, no. 3, pp. 291-312, 2012.

Appendix: Algorithm Specifications and Reproducibility

A.1 Complete Fitness Function

The fitness function combines accuracy and interpretability through weighted aggregation:

$$Fitness(T, D) = w\_acc \cdot Acc(T, D) + w\_interp \cdot [\Sigma_i \alpha_i \cdot M_i(T)]$$

where:

$$Acc(T, D) = fraction\ of\ correctly\ classified\ instances\ in\ D$$

$$M_1(T) = 1 - nodes(T)/max\_nodes\ (node\ complexity)$$

$$-M_2(T) = 1 - unique\_features(T)/total\_features\ (feature\ coherence)$$

$$M_3(T) = 1 - std(leaf\_depths)/max\_depth\ (tree\ balance)$$

$$M_4(T) = consistency\_score(feature\_positions)\ (semantic\ coherence)$$

Default weights: $w\_acc = 0.68, w\_interp = 0.32, \alpha_1 = 0.50, \alpha_2 = 0.10, \alpha_3 = 0.10, \alpha_4 = 0.30$