# Genetic Generalized Additive Models

Kaaustaaub Shankar[1]

College of Engineering and Applied Science, University of Cincinnati, Cincinnati, OH 45219, USA
shankaks@mail.uc.edu

**Abstract.** Generalized Additive Models (GAMs) balance predictive accuracy and interpretability, but manually configuring their structure is challenging. We propose using the multi-objective genetic algorithm NSGA-II to automatically optimize GAMs, jointly minimizing prediction error (RMSE) and a Complexity Penalty that captures sparsity, smoothness, and uncertainty. Experiments on the California Housing dataset show that NSGA-II discovers GAMs that outperform baseline PyGAMs in accuracy or match performance with substantially lower complexity. The resulting models are simpler, smoother, and exhibit narrower confidence intervals, enhancing interpretability. This framework provides a general approach for automated optimization of transparent, high-performing models.

**Keywords:** Generalized Additive Models · Regularization · Optimization · Genetic Algorithms

## 1 Introduction

As machine learning systems increasingly find their way into high-stakes domains, from housing valuation to autonomous weapons, the need for models that are not just accurate but inherently interpretable has never been more urgent. Researchers such as Missy Cummings have argued that AI systems should never be entrusted with control over life-and-death decisions unless their decision-making logic can be tested for wrong outputs[1] and interpretable models can help resolve this. This perspective regarding safety and in general how we integrate machine learning into society has been explored by many like Cynthia Rudin who believes that the best models are models that are inherently interpretable rather than explained by post-hoc methods that are unreliable since they are post-hoc[2].

As regulations around the world (an example being the EU AI Act[3]) urge AI developers and services built on these models to think more about how to make their product safer, the demand for transparent models has expanded far beyond defense and encompasses much of our day-to-day life. However, designing these systems is a significant challenge; their structure and hyperparameters have to be specified ahead of time and these choices determine their approximation capabilities, how well they generalize, and how interpretable they are. This is true for models like generalized additive models[4] and fuzzy inference systems (Takagi-Sugeno-Kang[5] and Mamdani[6]) where the structure of these models is the structure of the explanation. Selecting an appropriate structure is therefore not just about finding the best structure to reduce error but a principled way of controlling the complexity, inductive biases, and human comprehensibility of the learned representation.

Furthermore, to keep models human-interpretable, we need to introduce sparsity constraints that limit the complexity of the model. By controlling which components are active and how flexible each component can be, we reduce cognitive load and make the model easier to understand and audit. In addition, sparsity helps ensure that each part of the model contributes meaningful information, improving both the clarity and reliability of the explanations without relying solely on accuracy metrics.

Hence, this work explores the use of a genetic algorithm, specifically NSGA-II[7], to automatically optimize the structural hyperparameters of GAMs, enabling models that are both accurate and meaningfully interpretable.

## 2 Related Work

### 2.1 Generalized Additive Models (GAMs)

Generalized Additive Models (GAMs)[4] are a class of interpretable models that decompose a prediction into a sum of univariate or independent effects. This can be represented as the following:

$$y = \beta_0 + \sum_{j=1}^{p} f_j(x_j)$$

where $f_j$ is a smooth function applied to a single feature. Through this assumption, GAMs are highly interpretable since we can visualize the contribution of each feature independently. However, defining $f_j$ remains complicated, as real-world relationships may require flexible or nonlinear functions. To accommodate this, each $f_j$ can be specified as either a simple linear term or as a spline-based function.

Linear terms provide fully transparent, globally monotonic behavior but lack the expressiveness needed when the data exhibits nonlinear structure. Spline terms, in contrast, allow $f_j$ to adapt smoothly to nonlinear relationships while maintaining interpretability. These spline functions are built from a set of basis functions, and their flexibility is controlled by two key hyperparameters: the number of spline basis functions and the regularization constant $\lambda$.

The number of basis functions determines how much curvature the model can express. The regularization constant $\lambda$ further controls smoothness: a large $\lambda$ forces $f_j$ to be nearly linear, whereas a small $\lambda$ allows for high variability. These hyperparameters therefore play a structural role, directly influencing the expressiveness and interpretability of the resulting model.

Despite their simple nature, they have proven to be effective and a competitive choice against traditional models like neural networks for tabular data in a variety of fields [8].

### 2.2 Genetic Algorithms

Genetic algorithms [9] are a class of evolutionary algorithms that optimize a population of candidate solutions using biologically inspired operators such as crossover and mutation. By evolving an entire population rather than a single point estimate, GAs perform a broad exploration of the search space and help avoid the local minima that often trap greedy or gradient-based methods. This makes them particularly effective for multi-objective optimization, where different objectives may conflict. Approaches such as CMA-ES [10] and NSGA-II [7] demonstrate how evolutionary methods can jointly optimize multiple criteria and approximate a Pareto front which is the set of solutions that are non-dominated with respect to all objectives. In the context of interpretable modeling, this provides a natural mechanism for balancing predictive accuracy against model simplicity: the GA can search over a structured model space and return a spectrum of trade-offs rather than a single solution. From this spectrum, we can identify models that best align with regulatory requirements such as models that maintain strong predictive accuracy while ensuring structural simplicity and interpretability.

### 2.3 Pareto Front

A central concept in multi-objective optimization is the *Pareto front*, the set of solutions for which no objective can be improved without degrading another. Instead of producing a single optimal model, evolutionary algorithms such as NSGA-II identify a frontier of non-dominated solutions that represent different trade-offs between competing goals. Prior work has shown that Pareto-based search is effective in balancing predictive accuracy with sparsity, interpretability, or computational cost, making it well suited for model-selection settings where multiple criteria must be jointly optimized[11][12].

## 3 Methodology

We employ NSGA-II to search over GAM structures and hyperparameters. The GA will evolve these models using mutation and crossover, and the models are evaluated using predictive performance (RMSE) and a complexity penalty promoting sparsity and interpretability.

These models are trained on the California Housing dataset and are compared to baseline PyGAMs and Decision Trees over multiple seeds.

### 3.1 NSGA-II

Each candidate model is encoded as a chromosome, where genes represent the type of term for each feature (none, linear, or spline), the number of splines, the smoothing parameter $\lambda$, and whether to scale the feature. Chromsomes were initialized by the following:

**Algorithm 1** Smart Initialization of GAM Chromosome

```
Input: Number of features $n$, maximum splines $K_{\max} = 20$
Output: Chromosome genes
genes $\leftarrow$ []
for $i = 1$ to $n$ do
    Sample component type type $\in$ {none, linear, spline} with probabilities $[0.2, 0.3, 0.5]$
    if type = spline then
        knots $\leftarrow$ random integer in $[8, K_{\max}]$
        $\lambda \leftarrow$ random float in $[0.1, 10.0]$
    else
        knots $\leftarrow$ None, $\lambda \leftarrow$ None
    scale $\leftarrow$ random choice {True, False}
    Append {type, knots, $\lambda$, scale} to genes
return genes
```

Uniform crossover and perturbation mutation were employed in the genetic algorithm. Crossover was applied with a probability of 0.3, while mutation used an adaptive rate starting at 0.15 and gradually decreasing over 100 generations. The population size was set to 80 individuals, and the algorithm was run for 50 generations.

The fitness of a chromosome is evaluated using two objectives: a combination of predictive accuracy and model complexity. For each candidate GAM, we perform cross-validation on the combined training and validation data to obtain a robust estimate of performance. The predictive error is measured using root mean squared error (RMSE), while model complexity is a joint metric based off of confidence intervals and sparsity. The reason I picked this is because sparsity is directly related to model complexity while the size of the confidence interval relates to the variability; by minimizing both, we should be able to achieve a more interpretable model. Overally, this scoring system encourages sparse, interpretable models by penalizing overly complex GAMs, while still rewarding accurate predictions. The final fitness is averaged across all cross-validation folds ($k = 5$). We set the weight for RMSE to be 5 while the complexity is 2, allowing for us to prefer RMSE while optimizing for complexity as well.

To capture the range of achievable trade-offs, we extract three representative Pareto-optimal models: the knee solution, the RMSE-optimal model, and the model with the minimal complexity penalty.

### 3.2 California Housing Dataset

The California Housing dataset is a standard benchmark for regression tasks. It contains information collected from the 1990 U.S. Census, aggregated by block group. The dataset consists of 8 features and 1 continuous target variable (median house value in hundreds of thousands of dollars). Table 1 summarizes the features.

| **Feature** | **Description** |
|---|---|
| MedInc | Median income in block group (in tens of thousands of dollars) |
| HouseAge | Median age of houses in the block group |
| AveRooms | Average number of rooms per household |
| AveBedrms | Average number of bedrooms per household |
| Population | Total population of the block group |
| AveOccup | Average number of occupants per household |
| Latitude | Block group latitude |
| Longitude | Block group longitude |
| **Target** | Median house value (in hundreds of thousands of dollars) |

Table 1: Features and target of the California Housing dataset.

### 3.3 Baseline Models

The results of our GA-trained models will be compared to a LinearGam where all features will be using spline functions and will have a max of 25 splines each whereas the decision tree will have no maximum depth.

## 4 Results

| Seed | GAM (rsme) | GAM (knee) | GAM (penalty) | Decision Tree | PyGAM |
|---|---|---|---|---|---|
| 42 | **0.6821** | 0.7125 | 0.7400 | 0.6925 | 0.7216 |
| 7 | **0.6554** | 0.6745 | 0.7246 | 0.7159 | 0.6674 |
| 123 | **0.6523** | 0.6663 | 0.7130 | 0.6961 | 0.6697 |
| 225 | **0.6228** | 0.6702 | 0.7824 | 0.6770 | 0.6536 |
| 729 | **0.6496** | 0.6890 | 0.7108 | 0.6801 | 0.6520 |

Table 2: Test RMSE scores for five different models across various random seeds. The GA models are optimized for: lowest RMSE, the "knee" (best trade-off), and lowest complexity penalty. Bold values indicate the overall best performing model for each seed.

| Seed | Baseline | GAM (knee) | GAM (best_by_rmse) | GAM (best_by_penalty) |
|---|---|---|---|---|
| | Penalty | Penalty | Penalty | Penalty |
| 42 | 0.4096 | 0.1020 | 0.2226 | **0.0495** |
| 7 | 0.4384 | 0.1434 | 0.2390 | **0.0512** |
| 123 | 0.4562 | 0.1450 | 0.3986 | **0.0497** |
| 225 | 0.4353 | 0.1083 | 0.2905 | **0.0549** |
| 729 | 0.4608 | 0.0689 | 0.2508 | **0.0491** |

Table 3: Complexity penalty scores for different models across various random seeds. Bold values indicate the lowest penalty score for each seed.

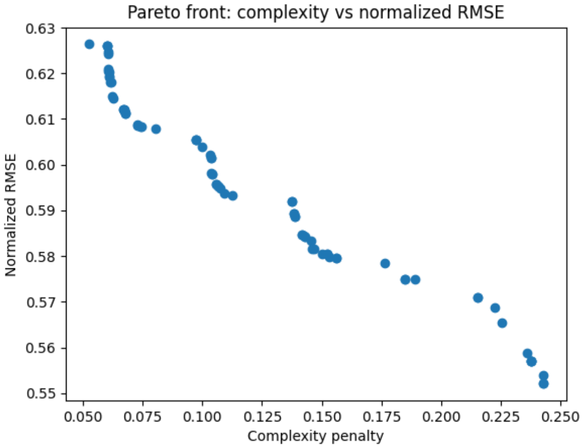

Fig. 1: Pareto Front for seed 7

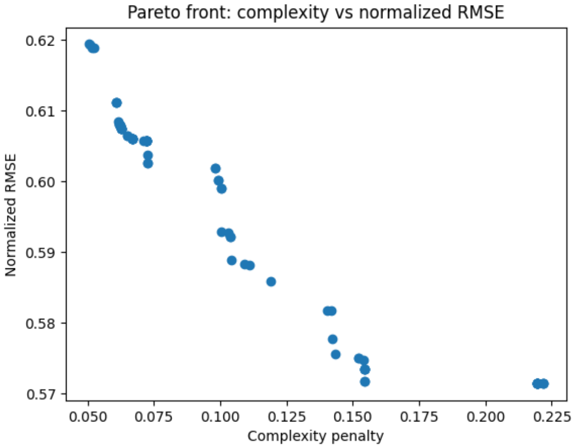

Fig. 2: Pareto Front for seed 42

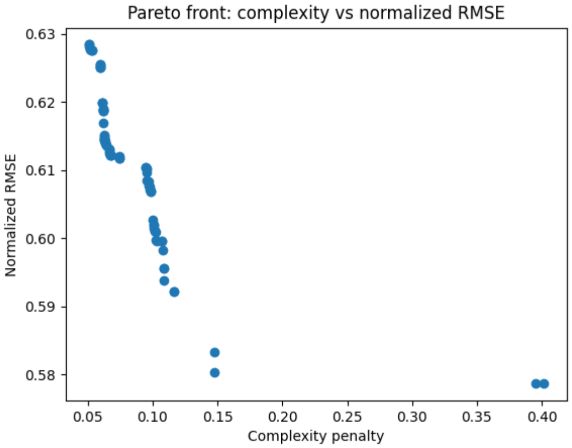

Fig. 3: Pareto Front for seed 123

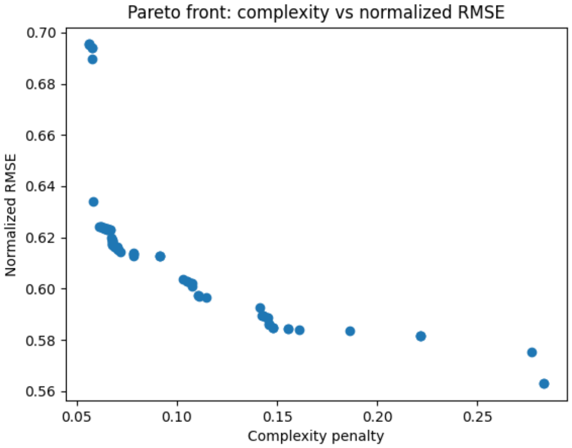

Fig. 4: Pareto Front for seed 225

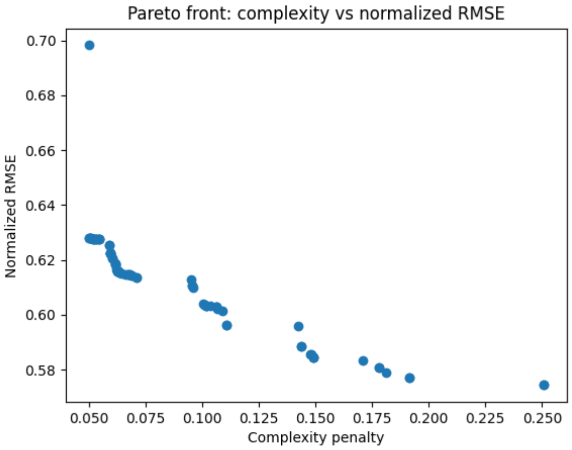

Fig. 5: Pareto Front for seed 729

## 5 Discussion

The results in Table 2 illustrate the characteristic behavior of a two-objective NSGA-II optimization applied to GAM structure search. As expected, the GA model selected for best RSME consistently achieves the lowest test error across all seeds, outperforming both the Decision Tree and the baseline PyGAM. This reflects the RMSE-dominant region of the Pareto front, where the algorithm prioritizes predictive accuracy at the expense of increased complexity and uncertainty. In contrast, the GA model selected for minimal penalty exhibits the highest RMSE among the GA solutions yet remains competitive against the baselines. The knee-point model, which represents the best global trade-off between prediction error and the penalty term, consistently occupies an intermediate position: it is less accurate than the RMSE-optimal model but typically comparable to or better than PyGAM while maintaining substantially lower complexity. These results showcase how NSGA is able to find solutions for both of these objectives.

In conjunction with Table 3, the results demonstrate that NSGA-II is able to discover models that achieve competitive predictive performance while maintaining substantially lower complexity. Notably, the GA model picked for the lowest RMSE is consistently simpler than the baseline PyGAM, despite outperforming it in prediction accuracy across all seeds. This indicates that within the space of high-performing GAMs, NSGA-II effectively identifies solutions that reside on a favorable portion of the Pareto front—models that provide strong predictive accuracy without incurring the unnecessary structural complexity typically produced by fully unconstrained spline fits. Consequently, the results reinforce the idea that meaningful interpretability gains can be achieved without sacrificing predictive performance, provided that the model search is guided by a multi-objective framework like NSGA-II.

Finally, the Pareto fronts for all seeds (Figures 1, 2, 3, 4, 5) illustrate that each run yields a diverse set of high-quality solutions. Rather than identifying a single "best" model, NSGA-II produces a spectrum of models that balance accuracy and complexity in different ways. This is particularly important in practice, where the preferred solution often depends on external requirements—such as interpretability, stability, or regulatory constraints. The results emphasize that model selection cannot rely on accuracy alone; instead, multiple objectives must be jointly considered to ensure that the chosen model is not only predictive but also appropriate for the operational and regulatory context.

## 6 Future Work

This work shows that NSGA-II can successfully optimize GAM structures by balancing accuracy with interpretability-oriented objectives. A natural next step is to extend this multi-objective approach to fuzzy and cascading rule-based systems, which pursue similar goals but involve different design choices. Fuzzy systems depend on decisions such as membership function shapes, the number and structure of rules, and the layout of cascading stages. Each of these can be treated as a structural hyperparameter, similar to knots or smoothing levels in GAMs. Future research could explore how the same types of objectives used here, such as accuracy, confidence interval width, sparsity, and rule simplicity, can guide the automated design of fuzzy models using evolutionary search. Applying NSGA-II to the design of layered or cascading fuzzy architectures may help create systems that remain interpretable while adapting their structure to the data in a principled way.

Another practical direction is to reduce the computational cost of evaluating candidate models. Fitting a GAM for every individual in every generation can be expensive, and this cost grows as we explore richer model spaces. Future work can introduce caching or memoization strategies that store and reuse the results of previously evaluated structures. This would avoid repeated model fits for identical or near-identical chromosomes and would make large-scale multi-objective optimization more feasible. Together, these extensions would broaden the impact of NSGA-II for interpretable model design across multiple modeling paradigms.

## References

1. Mary Cummings. Prohibiting generative AI in any form of weapon control. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*, 2025.
2. Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019.

3. European Union. Recital 72 of regulation (eu) 2024/1689. `https://ai-act-service-desk.ec.europa.eu/en/ai-act/recital-72`, 2024. Accessed: 2025-10-22.
4. Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986.
5. Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, 1985.
6. E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.
7. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
8. Jessica Doohan, Lucas Kook, and Kevin Burke. Comparison of generalised additive models and neural networks in applications: A systematic review, 2025.
9. John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
10. Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
11. Lennart Schneider, Bernd Bischl, and Janek Thomas. Multi-objective optimization of performance and interpretability of tabular supervised machine learning models, 2023.
12. William Blanzeisky and Pádraig Cunningham. Using pareto simulated annealing to address algorithmic bias in machine learning. *The Knowledge Engineering Review*, 37:e5, 2022.