Name: 444221516 OUYANG SHIYU
Email: oyshiyu@toki.waseda.jp
Code: https://github.com/hanpier/GA-Algorithm

Feature Selection by Genetic Algorithm

Abstract

Feature selection, as a preprocessing step to machine learning, is effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. However, the recent increase of dimensionality of data poses a severe challenge to many existing feature selection methods with respect to efficiency and effectiveness.

A novel algorithm for selecting features is presented in this paper. This algorithm has the aim of optimizing the number of features required to carry out data science discoveries such as classification and regression. This algorithm is of the wrapper type.

Firstly, a suitable regression model is selected, followed by the selection of features using the genetic algorithm. After further development of the genetic algorithm, the parameters and algorithm are confirmed to be appropriate.

Keywords: Feature Selection; GA; AGA

1 Introduction

1.1 Problem Background

To solve a specific problem in data science, a model should be built accordingly. However, selecting features for modeling is a non-trivial optimization problem. Feature selection is the process of retrieving the subset of features most relevant to an item and preparing the data for further processing. The most significant reason to use feature selection techniques is that some parts of the data may be redundant or irrelevant. Consequently, evaluating the significance and relevance of each feature can improve the generalization abilities of the model by reducing overfitting and eliminating redundant features without compromising accuracy.[1] Therefore, feature selection becomes an integral part of machine learning. By selecting effectively, it is possible to not only simplify the model and make the data easier to interpret by researchers, but also to reduce data processing time in subsequent processes and avoid dimensionality.

1.2 Restatement of the Problem

Based on the background information and constraints identified in the problem statement, the following must be addressed:

- Develop a regression model that fits and evaluates the data
- Use the genetic algorithm to select features
- Enhance the algorithm and compare the results

1.3 Literature Review

There are three main methods to solve the feature selection problem: filter method, wrapper method and embedded method.

The Filter method evaluates each variable by looking for a correlation with the target variable, evaluating each variable's informative value (the ability of an individual variable to predict the target)[2]. Finally, each feature gets an assigned score, and the features are sorted according to the score and then either kept or removed from the dataset. Wrapper methods treat feature selection as a search problem and tend to find the most effective combination of variables rather than the most significant value[3]. This is because they are measured separately. Predictive models are used to evaluate feature combinations and assign scores based on model accuracy. The embedded method learns which features contribute most to the accuracy of the model as it is being created. The most common embedded feature selection method is the regularization method. There are two classes of algorithms, supervised and unsupervised. Supervised methods select features with maximum representativeness and discriminative power, while unsupervised methods use inter-feature relationships to determine the relevance of features. For example, multivariate filtering method is semi-automated, and univariate filtering method, wrapper method, and embedding method are usually supervised.

1.4 My Work

When selecting features, we need to keep the number of features as low as possible and not make them redundant. This is the main focus of my work:

- ✧ Based on the original data, select the appropriate regression model
- ✧ Genetic algorithm-based feature selection
- ✧ Enhance and compare the genetic algorithm

In the first step, the original data are used to perform regression analysis of multiple models, and the most suitable model is selected for fitting and evaluation. In the second step, features are selected using a genetic algorithm. A detailed description of the selection process of the genetic algorithm is presented in this paper, followed by the final results of feature selection in accordance with the regression model. As a final step, the genetic algorithm is further improved, and the performance of the algorithm is evaluated based on the number of iterations and the maximum score.

In summary, the whole modeling process can be shown as follows:

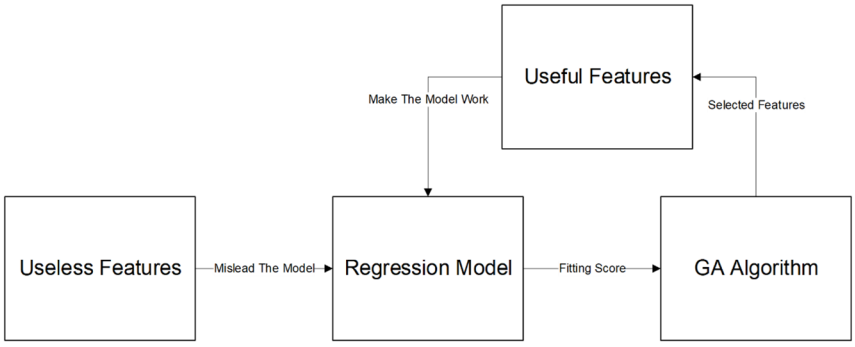

Figure 1: Model overview

2 Data Preparation

2.1 The Data

Since the amount of data is large and not intuitive, we directly visualize some of the data for display.

2.2 Data Analysis

The first step should be to analyze the data. The data set consists of fifty features, each of which has 2888 values. We do not extract useful information from practical applications because there are no missing values, but rather analyze it from the perspective of the data.

We can see from the correlation analysis as Figure 2 that several features have a strong linear relationship, and these linear relationships will have a greater impact on the regression model as a result of multicollinearity, so they should be simplified.

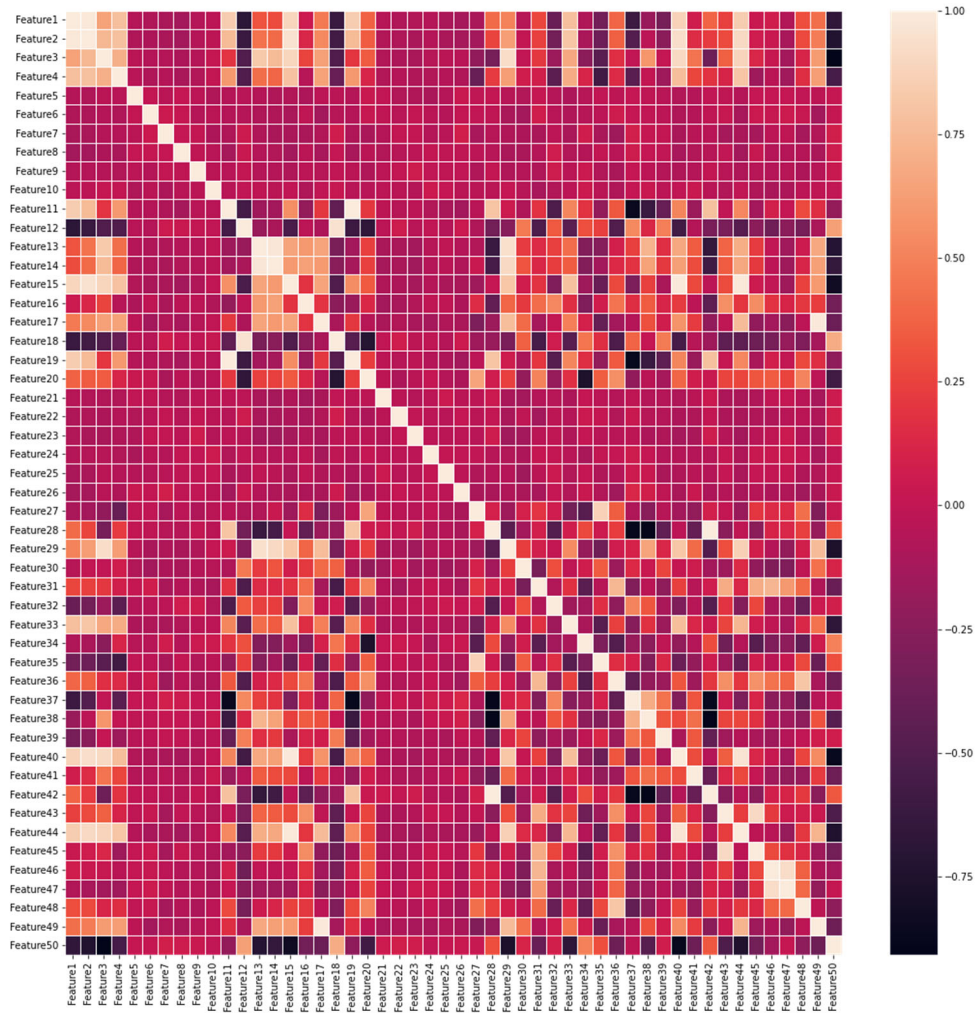

Figure 2: Correlation analysis

3 Model Establishment and Solution

3.1 Regression Model

A total of seven regression models were evaluated with R^2 as the evaluation metric. The definition of R^2 is expressed as:

$$R^2=1-\frac{SSE}{SST} \tag{1}$$

The results of different regression algorithm can be represented as Table 2. Lastly, the Linear Regression model with the highest score is selected. And then the fitness score of the genetic algorithm uses the R^2 value obtained by Linear Regression as the size of the fitness for population selection.

Table 2: Notations used in this paper

Regression	R^2
Linear Regression	0.892840
Ridge	0.881378
RandomForest Regressor	0.877568
SGD Regressor	0.871479
ElasticNet	0.854351
Lasso	0.851623
SVR	0.844713

3.2 Genetic Algorithm Model

The following outline summarizes how the genetic algorithm works[4]:

- The algorithm begins by creating a random initial population.
- The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
  - ✧ Scores each member of the current population by computing its fitness value. These values are called the raw fitness scores.
  - ✧ Scales the raw fitness scores to convert them into a more usable range of values. These scaled values are called expectation values.
  - ✧ Selects members, called parents, based on their expectation.
  - ✧ Some of the individuals in the current population that have lower fitness are chosen as elite. These elite individuals are passed to the next population.
  - ✧ Produces children from the parents. Children are produced either by making random changes to a single parent—mutation—or by combining the vector entries of a pair of parents—crossover.
  - ✧ Replaces the current population with the children to form the next generation.
- The algorithm stops when one of the stopping criteria is met. See Stopping Conditions for the Algorithm.

And the flow chart is as Figure 3.

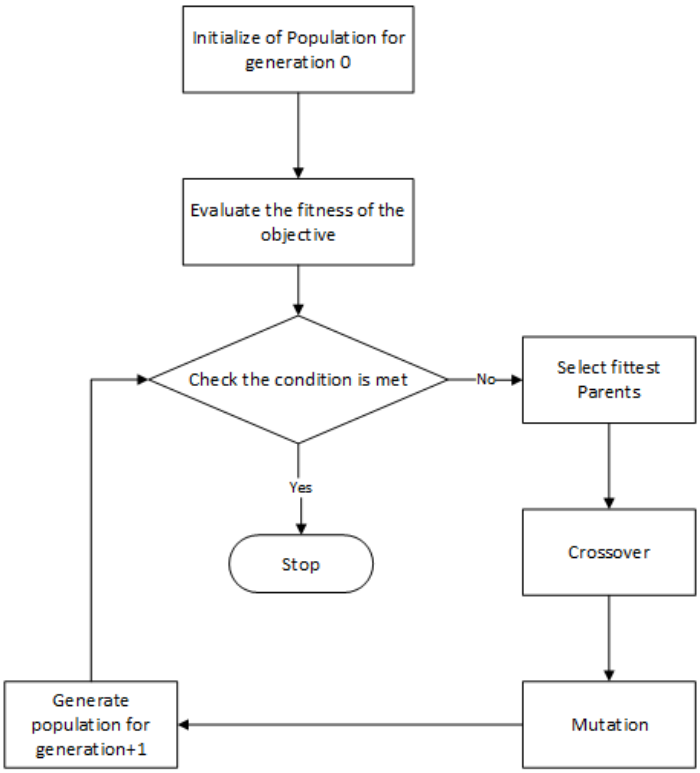

Figure 3: Flow chart of GA algorithm

3.2.1 Initial Population

Population is the group of individuals that form a generation. It is a subset of possible solutions that undergoes reproduction. A chromosome represents an individual in a population. In computational terms, the chromosome is represented by a binary string. For feature selection, the chromosome's length is taken as the number of features in the dataset. 0/1 indicates the presence/absence of the ith feature in the solution.

A large population size would cause the algorithm to slow down, while a small population will not show diversity. So it is important to choose population size wisely. Here I have used a population size of 75(~1.5x(number of features in the dataset)).

3.2.2 Creating the Next Generation

Reproduction involves forming a new generation by the mating of parents. Mating is implemented through the process of crossover. Mutation is used to add slight randomness to the individual to introduce diversity in the population.

- Crossover Children

Various crossover operations like One Point Crossover, Two Point Crossover, Uniform Crossover are used. Due to the huge amount of data, if the running time of the Two Point Crossover and Uniform Crossover programs is too long, consider abandoning their use. Here I have used One Point Crossover technique as Figure 4 which involves swapping genetic material between two points randomly chosen on the parents.

Usually, a probability is assigned to this process, indicating the chance of crossover for a

given pair. Crossover is a high probability event and is assigned an optimum probability between 0.65–0.80. Here, I have used a probability of 0.7.

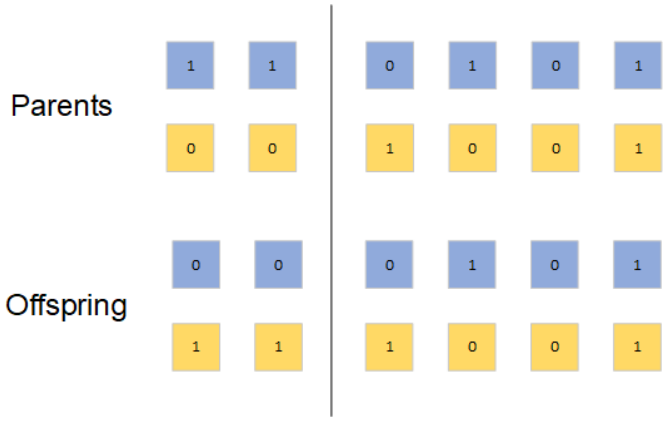

Figure 4: One point crossover

- Mutation Children

Mutation is used to introduce a slight variation in the chromosome by tweaking one of its genes. Its probability is kept very low to preserve the integrity of the population. Here, I have used a probability of 0.02.

In general, mutation is done by randomly swapping any bit of a random individual in the population. Following the conventional mutation process, it was observed that after many generations, the number of features extracted deviated a lot from N. To reduce deviation, I swapped a '0' bit with '1' bit. In this way, the deviation was reduced by a good extent.

3.2.3 Stopping Conditions for the Algorithm

After reproduction, a new generation is formed, and then the stopping criterion is checked. If the condition is satisfied, the algorithm terminates; otherwise, the process is repeated with the mutated population as the original population.

A few of the stopping criteria are listed below.

- ✧ Fixing the number of generations: This is not a very good method as we may miss out on the best generation because of an upper limit on the number of generations.
- ✧ Lack of progress in the fitness of the best individual of the population: Lack of progress does not necessarily imply convergence as evolution proceeds with punctuated equilibria that can later lead to improvement.
- ✧ Keeping an upper limit on the variance of fitness values in a population: When individuals are so similar to each other that we may not get a better individual in future generations, the algorithm would indicate convergence.

In a comprehensive analysis, I used the first and third stopping criteria. If one or three of these criteria are met, the algorithm is halted.

3.2.4 Selection

- Roulette Wheel Selection

Parent selection, which is one of the most crucial steps, is choosing individuals(parents)

from the population for reproduction, to produce the next generation.

Fitness Proportionate Parent Selection is the widely accepted criteria for parent selection. It ensures that all individuals get a chance to be selected as a parent with a probability proportionate to their fitness value. In this way, the underlying idea behind genetic algorithm would be justified.

There are various methods for parent selection like Tournament Selection, Roulette Wheel Selection, Stochastic Universal Sampling, Rank Selection, Random Selection, etc. I have used Roulette Wheel Selection here.

In Roulette Wheel Selection, a fixed point is chosen on the pie chart prepared using the fitness values. On every rotation, whichever individual comes in front of the point is selected for reproduction. This means that an individual with a greater area on the pie chart (i.e. a greater fitness value) has a high probability of being selected.

Implementation:

- ✧ Find the sum of all fitness values in a population
- ✧ Find normalized fitness values
- ✧ Find cumulative fitness values
- ✧ Generate a random number p between [0,1]
- ✧ The individual for which p is just smaller than the cumulative sum is selected. Adaptive Genetic Algorithm Model

- Elitism

Elitist strategy is used for avoiding destroying the best individual per generation. Specifically described as follows: If the next generation of groups of individual fitness value is less than the current population of individual fitness value, the best individuals in the current groups or adaptation value is greater than the value of the next generation of the best individual fitness multiple individuals directly copied to the next generation, random substitution or substitution corresponding number of the worst individuals in the next generation groups. The elitist strategy ensures that the current best individual will not be destructed by crossover and mutation operations, it is a basic protection of groups converge to the optimal solution. The top two offspring will inherit excellent genes from their parents.

3.3 Adaptive Genetic Algorithm Model

In the standard genetic algorithm, the crossover probability Pc and mutation probability Pm is an artificial set of fixed values, their value has a significant impact on the performance of the algorithm, such as optimization ability and convergence rate. Crossover probability determines the pace of new individuals generate, the larger the crossover rate is, the easier the pattern of the old individual would be destroyed, then the new individuals generate would be faster.

Excessive crossover rate may make more excellent individual mode destroyed, too small cross rate will delay the generation of new individuals, leading to prematurity and stagnation. The mutation rate is a key factor to decision algorithm to jump out of local optimal solution. The mutation rate is too small, which is not easy to generate a new mode structure, but mutation rate is so large to make GA the pure random search algorithm. Adaptive genetic algorithm is

hoping to find a more general adaptive crossover and mutation probability, to make the genetic algorithm more efficient. In the AGA, crossover and mutation probability is dynamic adjustment between the average fitness and the highest fitness of the population with the fitness of individual[5]. They satisfy the following relationship as equation (1)(2).

$$pc = \begin{cases} k_1 \dfrac{f\max - f'}{f\max - favg}, f' \geq \text{favg} \\ k_3, f' \leq favg \end{cases} \tag{2}$$

$$pm = \begin{cases} k_2 \dfrac{f\max - f'}{\text{fmax} - \text{favg}}, f' \geq \text{favg} \\ k_4, f' \leq \text{favg} \end{cases} \tag{3}$$

In the formula, f_{max} is the group's largest fitness value. f_{avg} is the value of the average fitness of the population. f' is the larger of the two individuals to cross the fitness. f is the mutation individual's fitness value. Here, just set to take the value of k_1, k_2, k_3, k_4 (in [0, 1] value), pc and pm can be adaptively adjusted. According to AGA, When the individual's fitness is lower than the contemporary population average fitness, that the individual poor performance. If the individual is selected in the selection mechanism, its crossover and mutation rate will be large. When the individual fitness is close to the maximum fitness in contemporary populations, which will be regarded as the good performance of the individual, its excellent model would be retained as far as possible. Even if the individual is selected in the selection mechanism, also its lower crossover and mutation rate.

The Adaptive Genetic Algorithm is described in Algorithm 1.

Algorithm 1: The process of AGA algorithm

Initialize the parameters
Generate a population P randomly
$generation \leftarrow 1$
while $generation \leq max\_gen$ and $variance \leq max\_tolerance$ **do**
  Clear the new population P'
  Use a fitness function to evaluate each individual in P
  while $|P'| \leq N$ **do**
    Select two parents from P
    Perform crossover with rate pc
    Perform mutation with rate pm
    Insert the offspring to P'
  endwhile
  $P \leftarrow P'$
  $generation \leftarrow generation + 1$
endwhile

4 Conclusion

4.1 Summary of Results

4.1.1 Result of GA algorithm

As a result of the calculation results from the Genetic Algorithm Model, the Figure 5 illustrates the trend of scores changing with generation. There are large fluctuations in the algorithm and it does not converge.

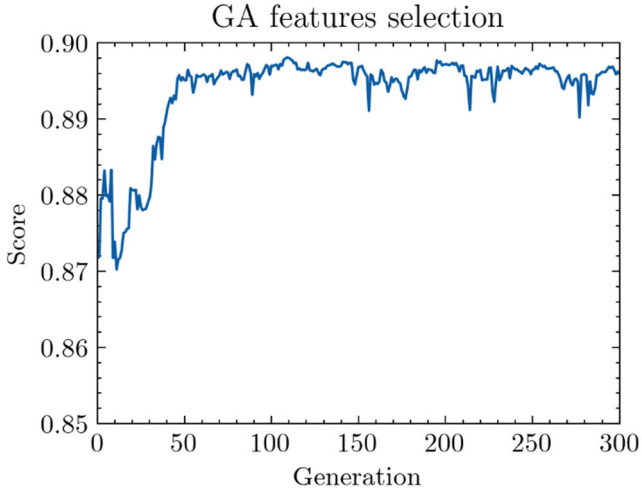

Figure 5: Result of GA feature selection

4.1.2 Result of AGA algorithm

The Figure 6 illustrates the trend of scores changing with generation based on the calculations from the adaptive genetic algorithm model. This algorithm is convergent and does not require 300 generations to achieve better results than GA.

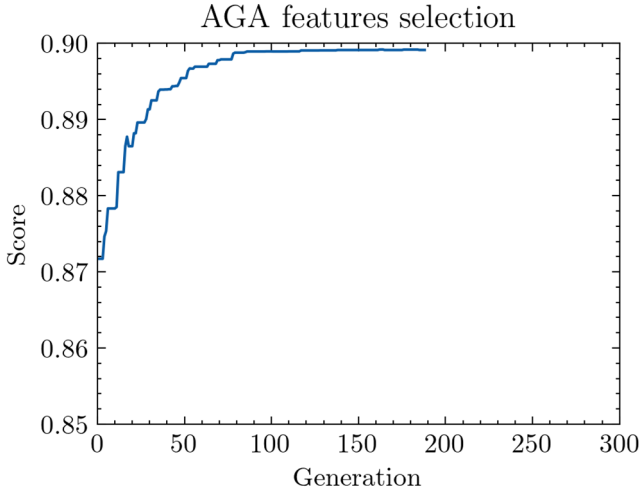

Figure 6: Result of AGA feature selection

4.1.3 Result Comparisons of Different Parameters

The different numbers of AGA represent the parameter settings from top to bottom in the figure. The final results are shown in the Table 3. The AGA algorithm appears to have a higher

score than the GA algorithm. From Figure 7, we can conclude that AGA algorithm results with differing parameters do not differ significantly. There may be a difference in this gap depending on how fitness is calculated. The representation of fitness might give the impression that these results are not very different. Therefore, we can conclude that the AGA algorithm is more effective than the GA algorithm, and when the parameters are not much different, the parameters have little influence on the final results.

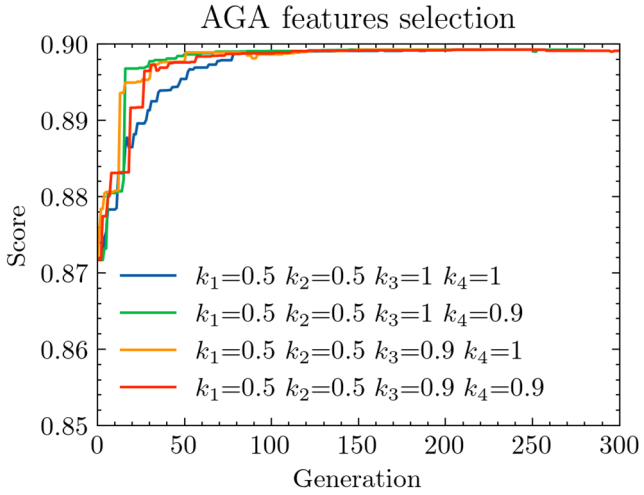

Figure 7: Result of AGA feature selection of different parameter

Table 3: Results of different algorithms and different parameters

Algorithm	Max Score	Position
GA	0.898115	109
AGA_1	0.899199	163
AGA_2	0.899288	172
AGA_3	0.899261	149
AGA_4	0.899288	240

4.2 Strengths

- Determine the most appropriate linear regression algorithm by evaluating different regression algorithms
- Improve the original genetic algorithm and test different parameters to achieve better results
- Scientific evidence has been gathered to support the credibility of the results

4.3 Possible Improvements

- Having more detailed information about parameters will enable me to analyze features more accurately if their meanings are more realistic
- There is a lack of robustness in modeling the algorithm, since parameters with the same chance will yield different results

References

[1] J. Li *et al.*, "Feature selection: A data perspective," *ACM computing surveys (CSUR),* vol. 50, no. 6, pp. 1-45, 2017.
[2] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, 2015: Ieee, pp. 1200-1205.
[3] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Information Sciences,* vol. 179, no. 13, pp. 2208-2217, 2009.
[4] S. Mirjalili, "Genetic algorithm," in *Evolutionary algorithms and neural networks*: Springer, 2019, pp. 43-55.
[5] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proceedings of an International Conference on Genetic Algorithms and their applications*, 1985, vol. 1: Hillsdale, New Jersey.