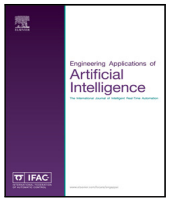
Engineering Applications of Artificial Intelligence 123 (2023) 106210

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Investigating the best automatic programming method in predicting the aerodynamic characteristics of wind turbine blade

Sibel Arslan [a,*], Kemal Koca [b]

[a] Department of Software Engineering, Sivas Cumhuriyet University, Sivas, 58140, Turkiye
[b] Department of Mechanical Engineering, Abdullah Gül University, Kayseri, 38080, Turkiye

ARTICLE INFO

Keywords:
Automatic programming
Genetic programming
Artificial bee colony programming
Aerodynamic coefficients
Power efficiency
Wind turbine blade

ABSTRACT

Automatic programming (AP) is a subfield of artificial intelligence (AI) that can automatically generate computer programs and solve complex engineering problems. This paper presents the accuracy of four different AP methods in predicting the aerodynamic coefficients and power efficiency of the AH 93-W-145 wind turbine blade at different Reynolds numbers and angles of attack. For the first time in the literature, Genetic Programming (GP) and Artificial Bee Colony Programming (ABCP) methods are used for such predictions. In addition, Airfoil Tools and JavaFoil are utilized for airfoil selection and dataset generation. The Reynolds number and angle of attack of the wind turbine airfoil are input parameters, while the coefficients C_L, C_D and power efficiency are output parameters. The results show that while all four methods tested in the study accurately predict the aerodynamic coefficients, Multi Gene GP (MGGP) method achieves the highest accuracy for R^2_{Train} and R^2_{Test} (R^2 values in C_D Train: 0.997-Test: 0.994, in C_L Train: 0.991-Test: 0.990, in PE Train: 0.990-Test: 0.970). By providing the most precise model for properly predicting the aerodynamic performance of higher cambered wind turbine airfoils, this innovative and comprehensive study will close a research gap. This will make a significant contribution to the field of AI and aerodynamics research without experimental cost, labor, and additional time.

1. Introduction

Over the past decade, severe constraints on pollution and global climate change have been observed due to restrictions on fossil fuel usage. These constraints have led to increased interest and abrupt growth in the use of energy associated with renewable energy sources such as hydropower, onshore and offshore wind, geothermal, solar, wave and tidal energy (Abdmouleh et al., 2015). Among renewable energy sources, wind power enjoys a better reputation than other renewable energy sources because it offers advantages in many respects, including energy cost and technological maturity. As a result, many countries have set a goal to generate electricity from wind energy. In this context, the wind energy sector plays a major role in conjunction with its innovative design methods such as the increasing size of wind turbines and the optimization of wind turbine blades (Huang et al., 2017). On the other hand, aerodynamic performance, which takes into account the aerodynamic design of wind turbine blades, has become a favorite topic in the industry.

The lift coefficient (C_L) and drag coefficient (C_D) are the most important parameters in determining the aerodynamic performance of a wind turbine blade (Papadimitriou and Papadimitriou, 2016; Genc et al., 2019). These aerodynamic coefficients can vary depending on model geometry (Karasu et al., 2020), different flow characteristics (Belamadi et al., 2016), and angle of attack (Elsakka et al., 2019). Therefore, researchers have conducted numerous studies to identify them. Kawazoe and Morita pointed out that C_L and C_D increase as the distance between the ground and the pressure surface of the blade profile decreases (Kawazoe and Morita, 2004). They also investigated flexible blades to increase the C_L coefficient, which led to an increase in wind power performance (MacPhee and Beyene, 2019; Descoteaux and Olivier, 2021). In addition, the effects of local flexible membrane material mounted over suction surfaces of airfoil have been experimentally studied to increase wind turbine blade performance and reduce noise and vibration (Genç et al., 2020; Koca et al., 2022c,d,b; Genç et al., 2022a; Koca et al., 2022a). Some researchers have also studied flow phenomena such as the formation of a laminar separation bubble (LSB) at the boundary layer separation, and experimentally investigated how the C_L and C_D coefficients are affected by these parameters (Koca et al., 2022b; Genç et al., 2022a; Koca et al., 2022a; Wahidi and Bridges, 2009; Koca et al., 2018; Genç et al., 2022b). Because these studies so far have conducted using experimental techniques such as pressure, velocity, and aerodynamic force measurements. The experimental cost, labor, and time required for the studies are high. For this reason,

* Corresponding author.
E-mail addresses: sibelarslan@cumhuriyet.edu.tr (S. Arslan), kemal.koca@agu.edu.tr (K. Koca).

https://doi.org/10.1016/j.engappai.2023.106210
Received 23 November 2022; Received in revised form 28 February 2023; Accepted 22 March 2023
Available online 29 March 2023

aerodynamic researchers have begun to take interest in numerical or theoretical simulations. The use of artificial intelligence (AI) methods in these applications has gradually gained acceptance. One such study, which plays an important role, predicted the aerodynamic coefficients of a delta wing using an artificial neural network (ANN) (Tumse et al., 2022). Another study, using ANN, estimated the aerodynamic performance of an airfoil with flow control mechanism (Akbıyık and Yavuz, 2021).

As outlined above, a number of studies have investigated the prediction of aerodynamic coefficients for flight vehicles; however, there are still considerable gaps in the literature, particularly when it comes to using and testing advanced AI methods. These methods outperformed state of art algorithms in solving various engineering problems in many studies (Xing et al., 2022a,b). In addition, automatic programming (AP) methods such as Genetic Programming (GP) and Artificial Bee Colony Programming (ABCP), which can automatically generate computer programs among the methods, have begun to attract more and more attention from AI researchers. Because of their flexible and comprehensible forms, GP and ABCP are used in a variety of applications, such as image segmentation (Yuan et al., 2022), parameter tuning (Stanovov et al., 2022), project scheduling (Luo et al., 2022), estimation (Emeksiz, 2022; Moradkhani et al., 2021), prediction (Sattar et al., 2022; Arslan and Öztürk, 2019; Hosseini et al., 2021; Bouaziz et al., 2016), symbolic regression (Karaboga et al., 2012; Arslan and Ozturk, 2019b; Gorkemli and Karaboga, 2019), modeling (Kütük and Arslan, 2022) and feature selection (Arslan and Ozturk, 2019a).

The main motivations for preferring the AP methods in this study are as follows:

- The prediction problem of aerodynamic coefficients, which we aim to solve, has limited information. In the literature, AP methods based on random processes have been shown to propose successful models for different real-world problems with limited information (Stanovov et al., 2022; Sattar et al., 2022; Arslan and Ozturk, 2019b).
- AP methods can describe the relationship between input and output parameters in problems generating nonlinear mathematical formulas. These formulas allow us to determine the effectiveness of the inputs to the problem and the quality of the relationship. Many state of art algorithms, such as metaheuristics and ANN, do not have this property.
- The methods give faster and more efficient results and saves high experimental cost, labor and time for predicting the coefficients.
- In the literature, AP methods have been shown to have higher performance in prediction problems compared to conventional algorithms such as ANN (Desai and Shaikh, 2012; Salmasi et al., 2013).
- To the best of our knowledge, there are no studies and researches in the literature for predicting the aerodynamic coefficients of a more cambered wind turbine airfoil using AP methods.

A further aim of the present study is to investigate the best AP method for predicting the aerodynamic coefficients of AH 93-W-145 wind turbine blade. First, we used GP, ABCP and their Multi Gene GP (MGGP) and Multi Hive ABCP (MHABCP) versions to propose new models and then compare the methods based on different performance criteria. The present study is also a pioneer in comparing the success of MGGP and MHABCP. Briefly, there is a research gap of utilizing AI techniques in engineering application especially at aerodynamic research. As mentioned above, this pioneering and detailed study will fill research gap by proposing the most accurate model in terms of correctly prediction of aerodynamic performance of more cambered wind turbine airfoil. This will lead major contribution to the researchers who interested in aerodynamics and AI.

The remainder of this paper is organized as follows: Section 2 explains the methods, and Section 3 presents performance evolution criteria, parameters, and datasets. Simulation results are discussed in Section 4. Section 5 gives a general evaluation of the study and mentions the limitations of the methods. The concluding part of the study is ensured in Section 6.

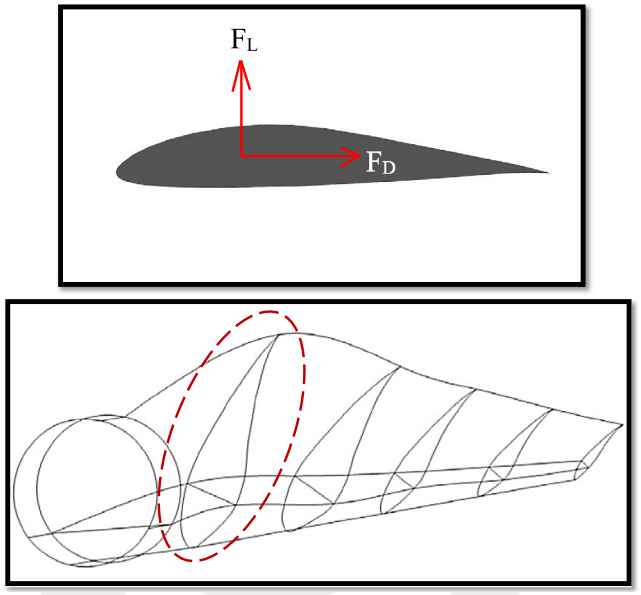


Fig. 1. The sketch of aerodynamic forces on AH 93-W-145 wind turbine airfoil.

2. Methods and methodology

2.1. Dataset collection

This section explains data collection process, lift (F_L) and drag (F_D) forces of the AH 93-W-145 wind turbine blades, which are used for the root part of commercial wind turbine blades, were collected and are presented in Fig. 1.

Airfoil Tools[1] and JavaFoil[2] were used to select the airfoils and acquire the dataset for the AP methods. A panel approach is used by the free program JavaFoil to resolve the velocity and pressure environment surrounding aerodynamic airfoils. This program effectively predicts the aerodynamic behavior of an optimized airfoil geometry at low angles of attack by combining the potential flow approach with a boundary layer analysis (Meana-Fernández et al., 2020). The following steps should be followed to get the airfoil polars: Airfoil and its coordinates can be determined thanks to Airfoil Tools. The coordinated of selected airfoils is then introduced in JavaFoil.

The computations are initiated once the analysis's airfoil Reynolds number and range of angles of attack are given. With those inputs, the program immediately returns the C_L and C_D values for each angle of attack. Each dataset is distinguished by a Reynolds number, which is based on the airfoil chord and identified as follows:

$$Re = \frac{\rho U_\infty c}{\mu} \tag{1}$$

which ρ is air density, U_∞ is the flow velocity, c is the chord length of selected airfoil, and μ is the dynamic viscosity. Furthermore, after determination of selected wind turbine airfoil thanks to Airfoil Tools, the lift (C_L) and the drag (C_D) coefficients as well as power efficiency (Bleischwitz et al., 2015) are calculated by means of JavaFoil with the following equations, respectively:

$$C_L = \frac{F_L}{0.5\rho U_\infty^2 S} \tag{2}$$

[1] http://airfoiltools.com/

[2] https://www.mh-aerotools.de/airfoils/javafoil.htm

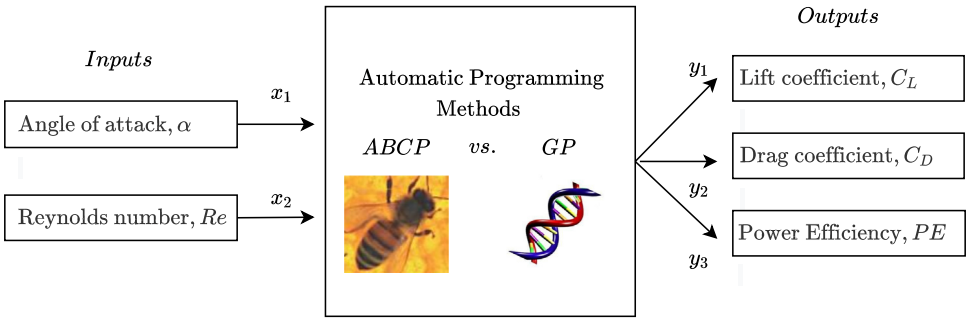


Fig. 2. The structure of proposed AP methods.

$$C_D = \frac{F_D}{0.5\rho U_\infty^2 S} \tag{3}$$

$$\text{Power Efficiency } (PE) = \frac{C_L^{3/2}}{C_D} \tag{4}$$

where S is the surface area of selected airfoil. Related to the dimensions of airfoil, its chord length and span length were 200 mm and 300 mm, respectively. Related to the input data limitations, the angle of attack was limited as $0° \leq a \leq 20°$. It was varied with 2° intervals. The flow velocities U_∞ were 6.5 m/s, 9.6 m/s and 13.2 m/s which corresponding to Reynolds number of 0.7×10^5, 1.05×10^5 and 1.4×10^5, respectively. For the output parameters, C_L was varied from 0.05 to 1.3, C_D was varied from 0.13 to 0.425 and power efficiency was differed from 0.175 to 4.7.

2.2. Overview of AP methods

As mentioned earlier, we employed four AP methods to predict the performance of the AH 93-W-145 wind turbine blade. Two of these methods, Multi Gene GP (MGGP) and Multi Hive ABCP (MHABCP), are advanced versions of GP and ABCP respectively. To provide a clear understanding of the results, we first present the basic principles of GP and ABCP, and then delve into the specific features of MGGP and MHABCP.

2.2.1. GP and ABCP

In Fig. 2, the structure of the AP methods is illustrated, showing that the input parameters used are the angle of attack and Reynolds number, and the output parameters predicted aerodynamic properties such as the lift coefficient, drag coefficient, and power efficiency.

Cramer (1985) applied a tree approach similar to the first modern application to Markov decision processes. Koza (1994) showed that the relationship between dependent and independent parameters of systems can be determined by using GP without the need for an initial model. GP is inspired by GA, and the main difference between these two algorithms is the representation of chromosomes. Another trend in GP-like method is ABCP, which is based on swarm intelligence and simulates the foraging behavior of honeybees. It uses similar algorithm steps as Artificial Bee Colony (ABC) (Karaboga et al., 2012). Many recent studies have shown that ABCP is robust and practical compared to other AI methods (Karaboga et al., 2012; Arslan and Ozturk, 2019b; Gorkemli and Karaboga, 2019; Arslan and Ozturk, 2019; Nekoei et al., 2021). Similar to the two methods in GP and ABCP, the solutions are represented by computer programs in tree structure. These computer programs consist of various combinations of terminals and functions specific to the problem (Poli et al., 2008). A tree structure that could represent a solution in both methods is shown in Fig. 3.

As shown in Fig. 3, the tree structure of the solution includes terminals such as constants, parameters, and functions such as arithmetic, mathematical and logical functions. The specific terminals and functions used in the solution may vary depending on the problem type.

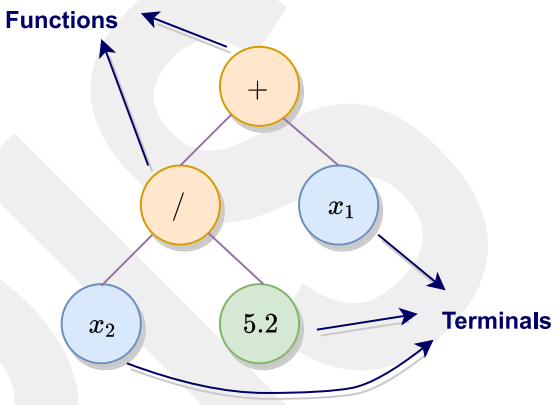


Fig. 3. Example of tree solution in GP and ABCP.

The mathematical relationship of the solution is represented in Eq. (5). In this equation, the independent parameters are (x_1, x_2), the constant is (5.2), and the dependent parameter is $f(x_1, x_2)$.

$$\mathrm{f}(x_1, x_2) = \frac{x_2}{5.2} + x_1 \tag{5}$$

In addition to representing the solutions of individuals in the tree structure, GP and ABCP also improve the solutions with similar operators. Karaboga et al. (2012) adapted the crossover operator in GP to ABCP, known as an "information sharing mechanism". An example of a crossover/information sharing mechanism operator can be seen in Fig. 4. The operator replaces a randomly selected subtree from the candidate solution with a randomly selected subtree from the neighboring solution. In ABCP, the neighborhood solution is different from the candidate solution. The generated candidate solution is expected to be a better solution according to the mechanism.

Besides the crossover operator, operators such as mutation and reproduction are also used in GP to improve the solutions. The flowcharts of GP and ABCP are shown in Fig. 5 and Fig. 6 respectively. The first step in the flowchart of GP is to generate the initial population. The individuals in the population can be generated by different methods, e.g., Full, Grow, or Ramp Half and Half (Nekoei et al., 2021). The selection, crossover and mutation operators in GA are adapted to GP to improve existing solutions and search for new ones. The selection operator is also used to evaluate the fitness of each individual in the population based on the objective function value of the problem. Individuals with good fitness are more likely to be found in the next generation, but individuals with low fitness values are also more likely to be found. Methods such as the selection operator, the roulette wheel, and the tournament method are used to improve the population.

Depending on the fitness value of the crossover operator, two parents are selected from the population. Generally, the crossover points for both parents are determined randomly. As shown in Fig. 4, the subtrees of the two crossover points in the individuals are replaced and

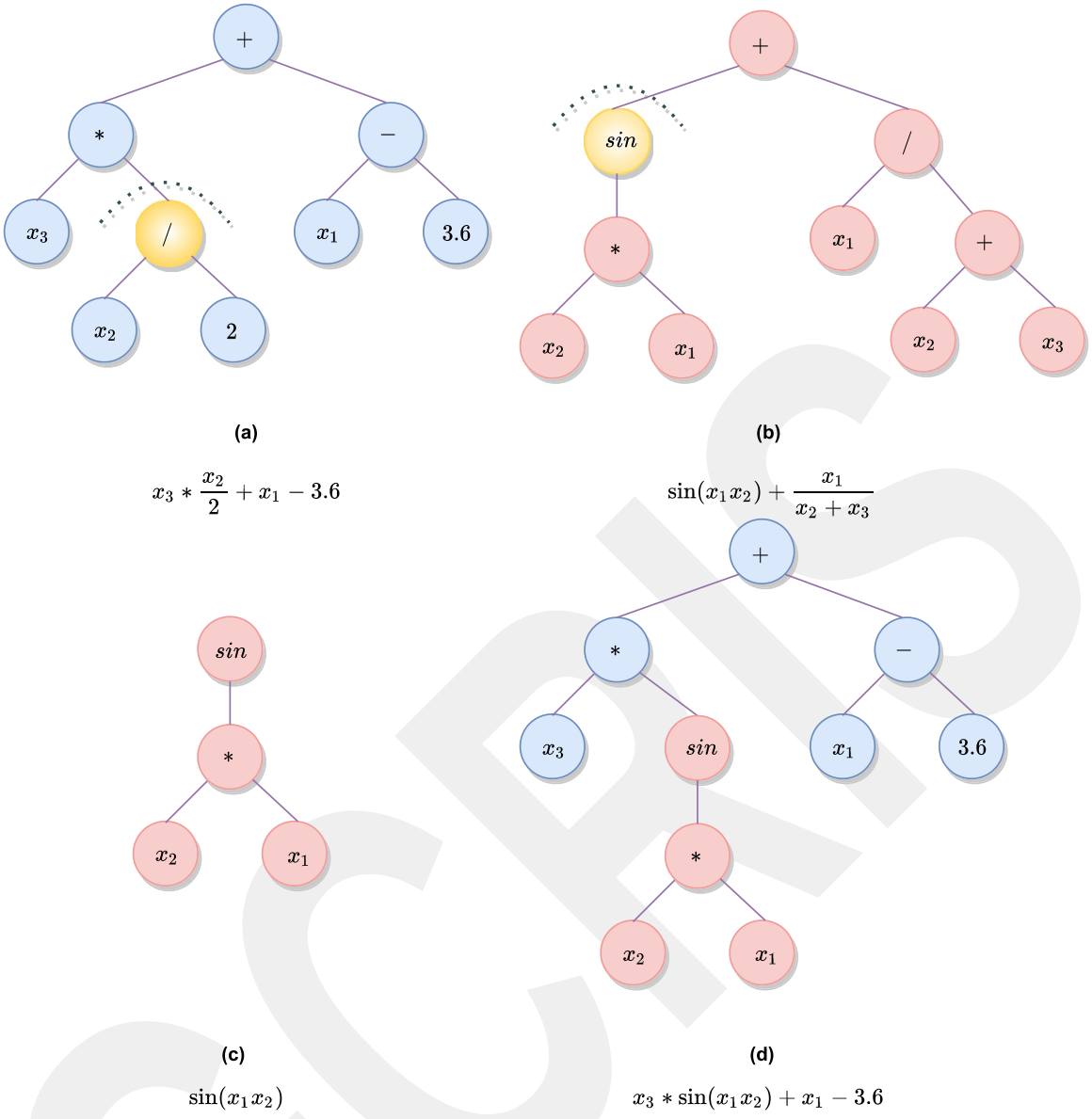


Fig. 4. Example of crossover in GP/information sharing mechanism in ABCP. (a) candidate solution (b) neighborhood solution (c) received subtree from the neighborhood solution (d) generated candidate solution.

new individuals are generated. The mutation point or node is randomly determined by the individual in the mutation operator. A randomly generated tree or node is placed at the location of the node selected from the tree. The operator makes it possible to find new, unprecedented and unexplored solutions (Poli et al., 2008). The best solutions of the previous generation are passed on to the current generation in elitism. The program is then terminated according to predefined termination criteria when the solutions have reached a certain fitness value and the program has reached the maximum number of iterations.

The most important change made in adapting the ABC algorithm to ABCP is the neighborhood mechanism in the generation of candidate solutions, called the information sharing mechanism, shown in Fig. 4. At the same time, it has the same algorithm steps as the ABC. In the ABCP, there are three different types of bees: (*i*) employed bees, (*ii*) onlooker bees, and (*ii*) scout bees. Employed bees leave the hive and have a specific food source in their memory (Karaboğa, 2014). When they return to the hive, they share the information about the food source with the onlooker bees. The onlooker bees decide which source they will go to depending on how much nectar is available in the source shared by the employed bees. During the scout bee phase, they check to see if all food sources are exhausted. If the food source is exhausted, the source is abandoned. The bees that officially worked with an abandoned source become scout bees and randomly search for a new source. Unlike other bees, scout bees find new food sources without passing on information. The exhaustion of food resources is controlled by a parameter called *limit*. For each source the number of improvement attempts is recorded and in each cycle, it is checked if the number of attempts exceeds the *limit*.

2.2.2. MGGP and MHABCP

MGGP is a widely used version of GP and represents solutions with multi-tree structures of low depth. MHABCP, on the other hand, is an ABCP based high-level AP method that aims to build reliable models for high-dimensional symbolic regression problems. Similar to MGGP, the solutions in MHABCP also have a multi-tree structure. In GP, each tree in the solution is referred to as a gene, and in MHABCP, it is called a hive. The mathematical formula of the solution results from the linear combination of the subtrees. An example of a solution with subtrees of low depth that can be generated by both algorithms is shown in Fig. 7. The formula for the solution is given in Eq. (6). In Eq. (6), a coefficient

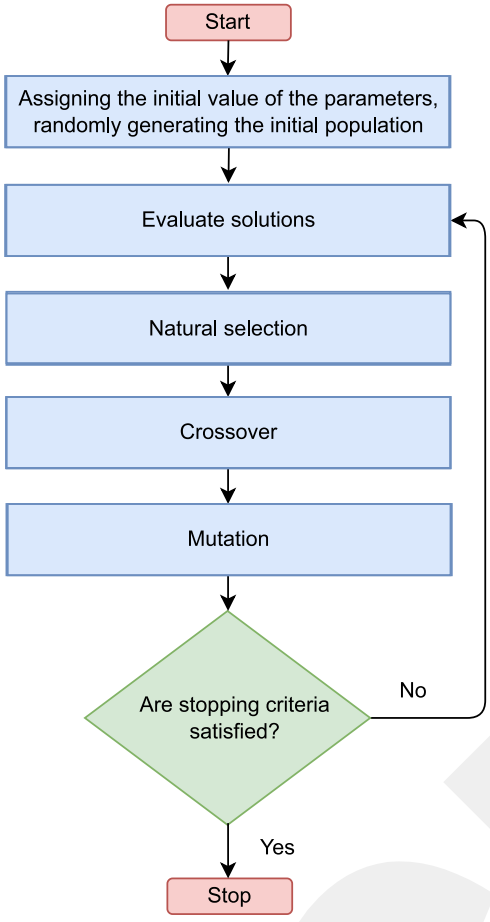


Fig. 5. The flow chart of GP algorithm.

is added to the partial equation of each tree. Additionally, w_0 is a bias. The maximum number of trees that can be found in a solution is expressed by the maximum number of genes (G_{max}) in MGGP and the maximum number of hives (H_{max}) in MHABCP.

$$\mathrm{f}\left(x_1, x_2\right) = w_0 + w_1\left(\left(\sin\left(x_1\right) * x_2\right) - \left(x_1 + 5.2\right)\right) + w_2\left(\frac{\cos\left(x_2\right) + 6.4}{\log\left(x_1\right)}\right) \tag{6}$$

In addition to the crossover/information sharing mechanism in these algorithms, there is a new operator that allows individual tree structures to change places within themselves. This operator is called high-level crossover in MGGP, while in MHABCP it is called the hive exchange mechanism. The operator performs an evolution, as shown in Fig. 8. It should be noted that the trees are exchanged and pruned according to the predefined maximum number of trees. The mathematical formula for a solution in both algorithms is calculated by Eq. (7).

$$x_i = w_{i0} + \sum_{j=1}^{k}\left(w_{ij} * T_{ij}\right) \tag{7}$$

where x_i is the ith solution in the population, w_{i0} is the initial weight (bias), w_{ij} is the weight of the jth tree for solution i, T_{ij} is the jth tree defined for the ith solution.

3. Experimental design

In this section, the datasets of AH 93-W-145 wind turbine blade are explained and the experimental setup, such as the fitness functions and parameters, are examined.

Table 1
Parameters.

Parameters	GP	ABCP	MGGP	MHABCP
Population Size	250	–	250	–
Colony Size	–	250	–	250
Generation	250	250	250	250
Maximum Tree Depth	10	10	5	5
Crossover Rate	0.84	–	0.84	–
Mutation Rate	0.14	–	0.14	–
Direct Reproduction Rate	0.02	–	0.02	–
Tournament size	6	–	6	–
Limit	–	50	–	50
Maximum Number of Hive	–	–	4	4
Information Sharing Mechanism Rate	–	1	0.8	0.8
Hive Exchange Mechanism Rate	–	–	0.2	0.2
Functions	+, −, * ,/ (protected), *square*, *cube*			

3.1. Performance evolution criteria

The AP methods are evaluated by using six different performance evolution criteria. These functions are as follows:

- *Sum Squared Error (SSE)*: It is illustrated in Eq. (8).

$$SSE = \sum_{i=1}^{N}\left(f\left(x_i\right) - Y_i\right)^2 \tag{8}$$

- *Root Mean Square Error (RMSE)*: It is given in Eq. (9).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f\left(x_i\right) - Y_i\right)^2} \tag{9}$$

- *Mean Absolute Error (MAE)*: It is shown in Eq. (10).

$$MAE = \frac{\sum_{i=1}^{N}\left|f\left(x_i\right) - Y_i\right|}{N} \tag{10}$$

- *Mean Square Error (MSE)*: It is presented in Eq. (11).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(f\left(x_i\right) - Y_i\right)^2 \tag{11}$$

- *Mean Absolute Percentage Error (MAPE)*: It is given in Eq. (12).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f\left(x_i\right) - Y_i}{Y_i}\right| \tag{12}$$

- *Symmetric Mean Absolute Percentage Error (SMAPE)*: It is given in (13).

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|f\left(x_i\right) - Y_i\right|}{\left(\left|Y_i\right| + \left|f\left(x_i\right)\right|\right)/2} \tag{13}$$

where N is the number of samples, $f\left(x_i\right)$ denotes the predicted output of models generated by the methods, Y_i is the actual output parameter. The complexity of the solution trees is calculated as a function of the nodes' number and the depth of the trees by where C is the complexity of the solution, d is the depth of the solution, and n is the total number of nodes in the corresponding depth of the solution.

3.2. Parameters

The parameters used by the methods in all runs are listed in Table 1. To ensure a fair comparison, the same settings were used for the parameter values of both GP and MGGP, as well as ABCP and MHABCP. The initial solutions were generated using the ramped half and half method for all methods. To improve the solutions, crossover, mutation and direct reproduction operators were used in GP, while an information

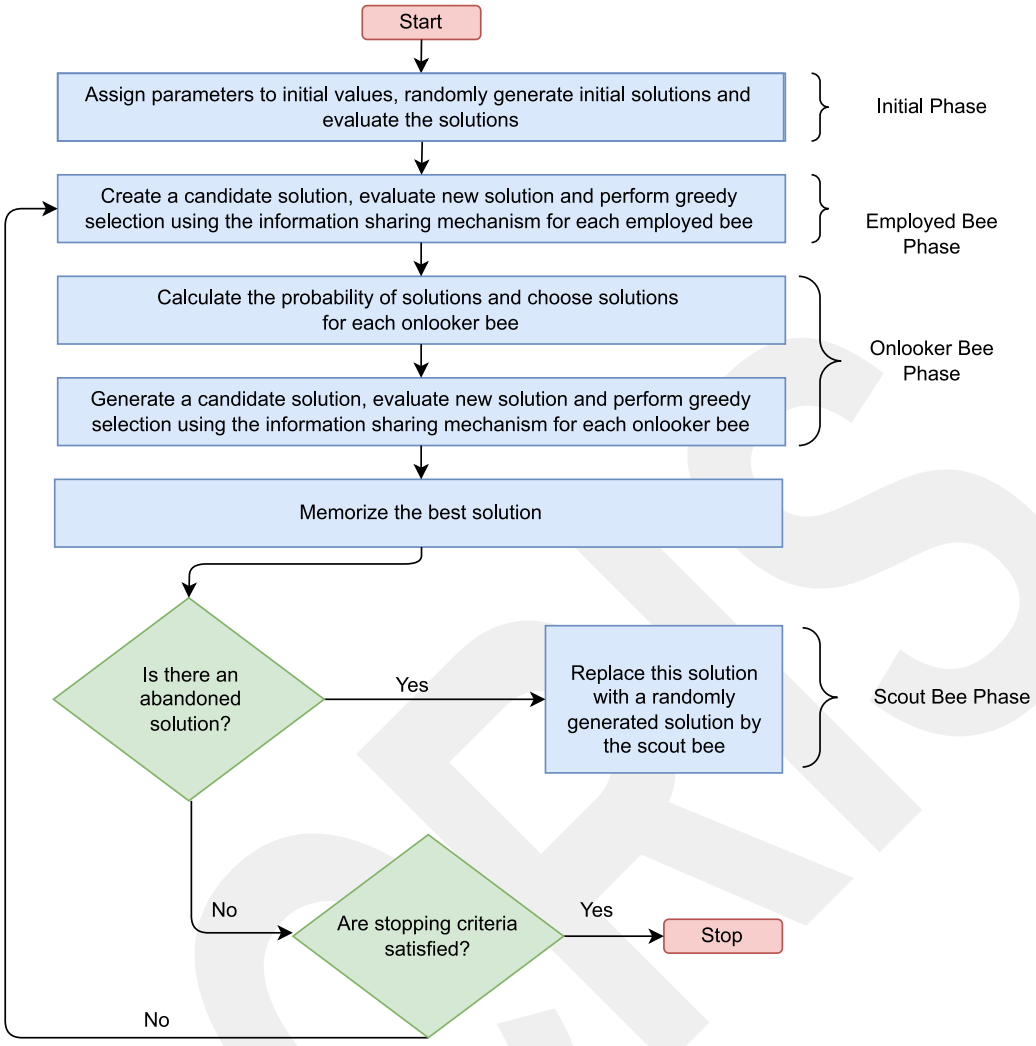


Fig. 6. The flow chart of ABCP algorithm.

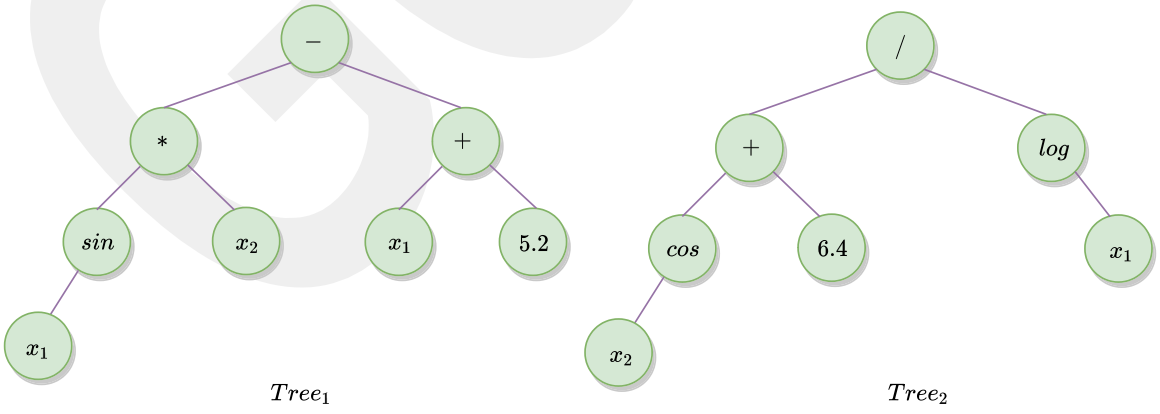


Fig. 7. Example of multi-tree structure in MGGP and MHABCP.

sharing mechanism was used in ABCP. Additionally, GP and MHABCP also employed a high-level crossover/hive exchange mechanism at a rate of 0.2.

The function of *square* in Table 1 is used to represent the square of a number, while the function of *cube* is used to represent the cube of a number. In the MGGP, individuals are evolved by crossover at a rate of 0.84, and a decision is made to perform a high-level crossover with a randomly generated value. If the value is less than 0.2, the high-level crossover is not applied, otherwise the standard crossover operator is used.

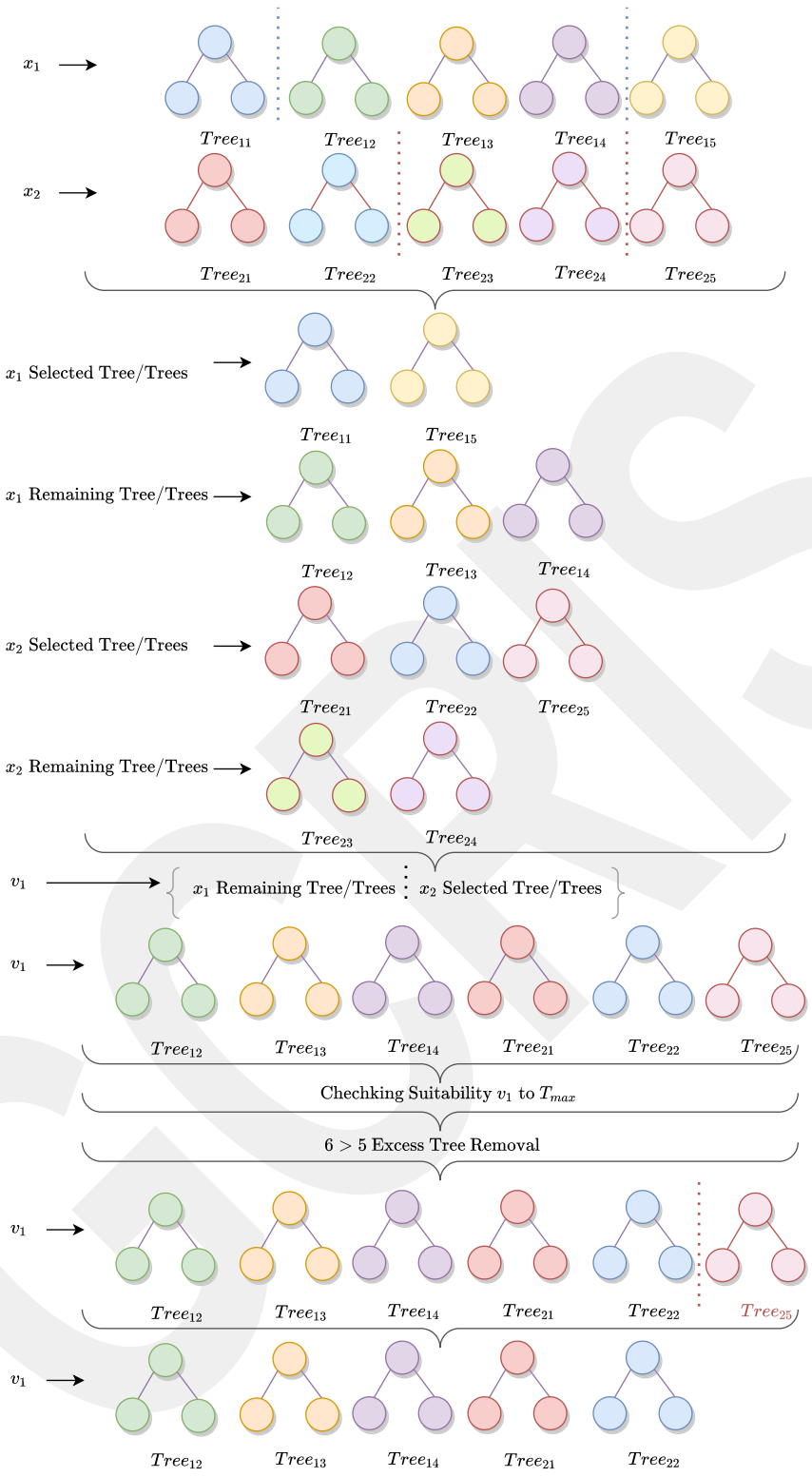


Fig. 8. Example of high-level crossover in MGGP/hive exchange mechanism in MHABCP.

3.3. Datasets

Three different datasets were used, each for a different coefficient, with a 70% training samples to 30% test samples split. There were a total of 33 samples in each dataset, with 23 randomly chosen for training and the remaining 10 for testing. Each sample had two characteristics: the angle of attack (AoA) and the Reynolds number (Re). It is worth noting that the non-normalized samples were found to be more successful than the normalized samples in the datasets and thus the experimental results were based on non-normalized samples. The datasets were randomized in order to track the success of the methods on the test samples.

The experiments that were conducted following the information in this section are listed below:

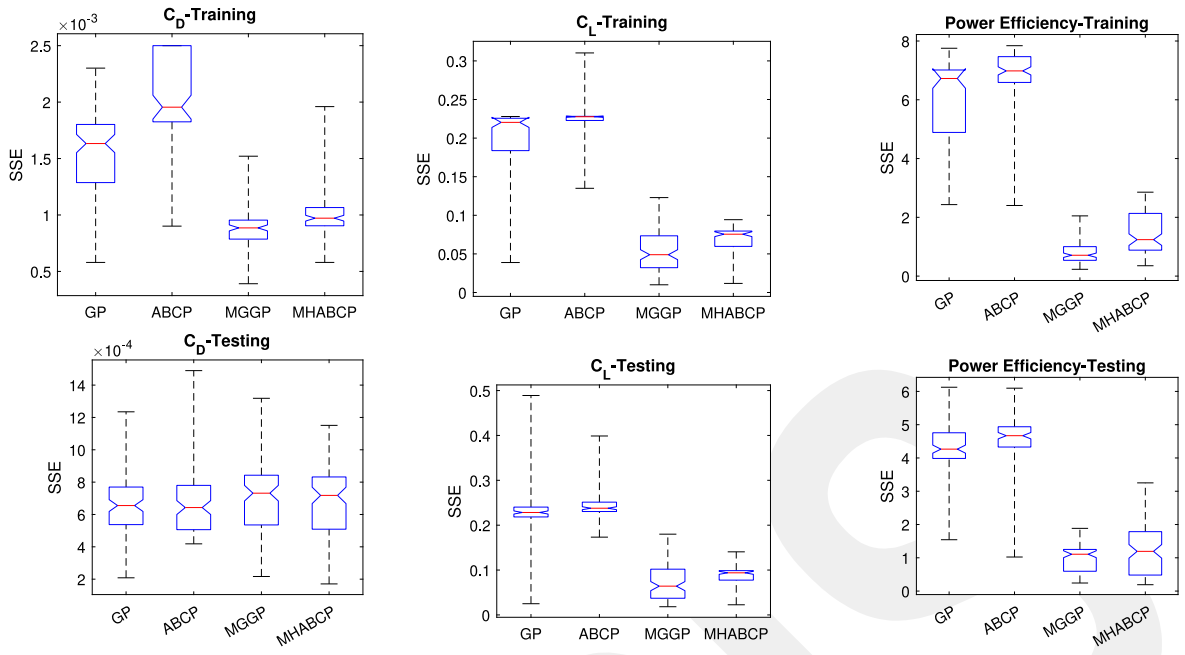


Fig. 9. Distribution of SSEs of the best-of-run individuals.

- Datasets were generated from programs (Airfoil Tools and JavaFoil) for each aerodynamic coefficient. These 3 datasets were randomly split for experiments.
- Parameters of each algorithm were adjusted in such a way that fair comparisons could be made according to the operators mentioned in Section 2.2.
- Datasets were tested with each of the four methods in 100 independent runs. Each run is performed in 250 iterations. The algorithms were run according to the flow diagrams.
- After the experiments were completed, the results were analyzed according to the 6 different performance criteria in Section 3.1.

4. Simulation results

The experimental results of the methods are presented and discussed in this section. First, the success of training and test data is obtained with six different metrics for each coefficient. Then, the statistical significance of the differences between the methods and the results are evaluated using t-tests. Information on the R^2 of these methods is also provided. Finally, some additional analyses are performed to compare the performance of the methods and to investigate the best models resulting from the runs.

4.1. Results on the training/test sets

The performance evolution criteria of the training and test distribution values are given. The error bars in the boxes indicate the minimum and maximum of the metrics and the circles show the median of the metrics. Because the median is thought to be more robust to outliers, the median is used along with the mean for the analysis (Gonçalves et al., 2015).

Fig. 9 shows SSE distributions for all coefficients. After comparing the four methods in the SSE training, MGGP has the lowest median value, making it the most successful one. The method that comes closest to MGGP's success is MHABCP. GP and ABCP have worse results than these versions. When evaluating the lowest and highest values, the intervals of MGGP for C_D and PE training, as well as MHABCP for C_L training, are lower, indicating that their standard deviations are low. ABCP is the worst method for C_D and C_L training. In addition, the interval between the first and third quartile of the ABCP for C_L training is quite narrow. Although MGGP and MHABCP have similar error values in training and testing PE, the range of MHABCP is wider than MGGP. In C_D test, the median values of GP and ABCP are similar and MHABCP has a lower minimum value.

RMSE and MAE distributions in Figs. 10 and 11 show that similar results can be noted for the SSE distributions. In C_D test, the methods have close median values in both the RMSE and MAE distributions. For these three distributions in the C_L tests, GP and ABCP have a narrow interval and the standard deviation of GP is quite high. The standard deviation of the versions is generally lower than for the GP and ABCP.

Fig. 12 shows the MSE distributions. With the exception of the C_D test, MGGP is the most successful method. In addition, the standard deviation for all test datasets is generally high. In particular, the range of the first fourth quartile of MHABCP in the PE test is high. In the C_L testing, the values for the second and third quartiles of the methods GP and ABCP are very close. In C_L training, the range of ABCP is quite narrow compared to the other methods.

Fig. 13 illustrates the MAPE distributions. ABCP has low performance on all training datasets. Unlike the other methods, the standard deviation of ABCP in the C_D test is quite high. It can be noted that the deviation of GP is high in the C_L test.

Fig. 14 points out the SMAPE distributions. The training sets are similar compared to the MAPE distributions. The ranges are generally wider in the C_D test than in the MAPE distribution. When all plots are evaluated together, MGGP and MHABCP are more successful than the standard methods. MGGP shows the highest performance, while ABCP shows the lowest performance.

MGGP has the lowest error values among all methods, and one-sided two sample t-tests were used to see if these differences were statistically significant (significant level α: 0.05). The t-test results for C_D, C_L, and power efficiency (PE) are provided in Tables 2, 3, and 4. Additionally, Table 5 shows the results of the t-test investigating the difference between GP and ABCP.

When the p-value in the t-test is small, it means that the error values of the first method are higher than those of the second method, and this test rejects the null hypothesis at the default significance level of 5%, in favor of the alternative hypothesis ($h = 1$). In this context, the other methods have higher error values than MGGP for the training sets of all coefficients in Tables 2 and 4, but for the C_D testing, it cannot be

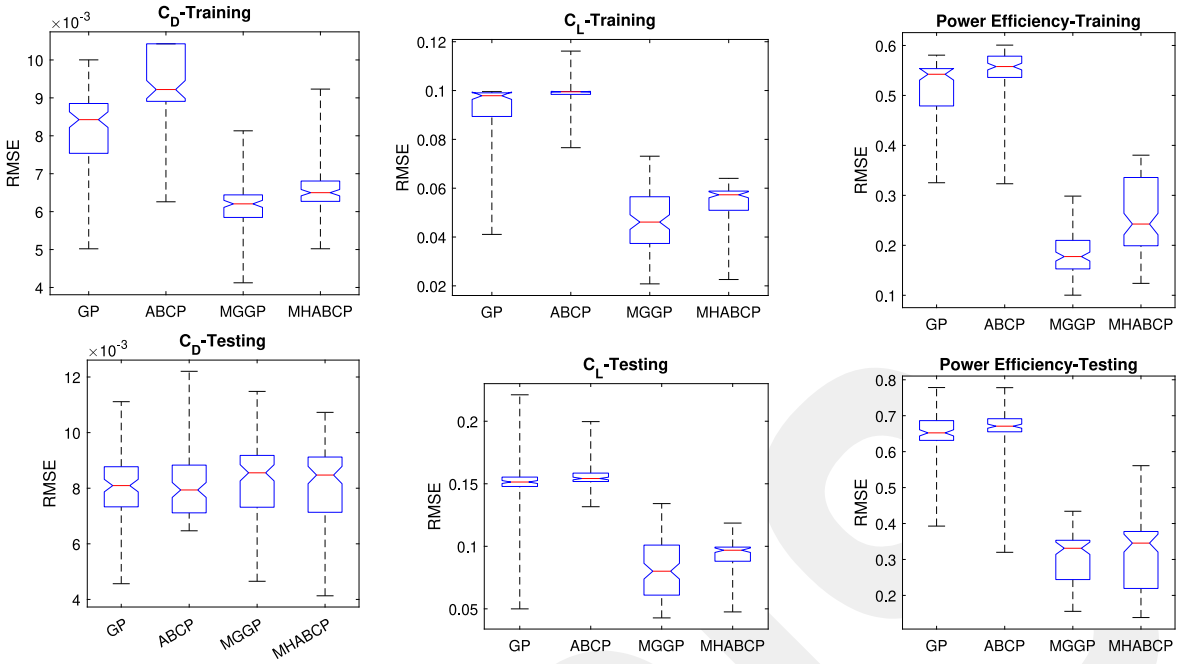


Fig. 10. Distribution of RMSEs of the best-of-run individuals.

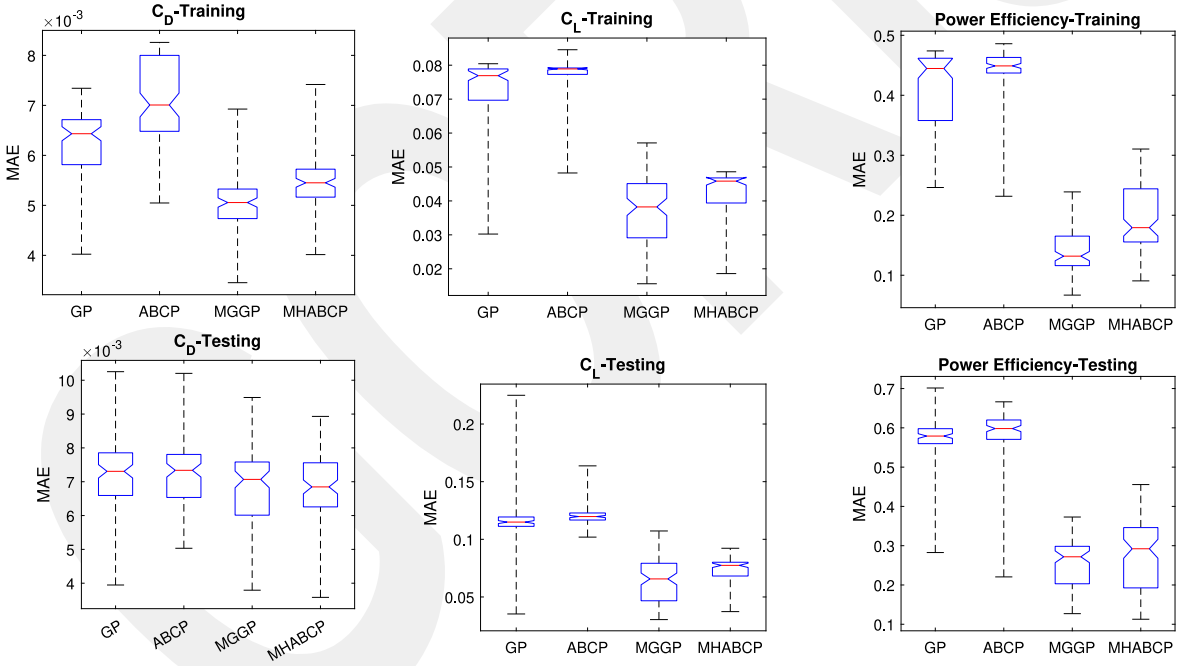


Fig. 11. Distribution of MAEs of the best-of-run individuals.

Table 2
P-values between MGGP and other automatic programming methods for C_D.

Metric	GP-MGGP		ABCP-MGGP		MHABCP-MGGP	
	C_D Training	C_D Testing	C_D Training	C_D Testing	C_D Training	C_D Testing
SSE	0.00	0.77	0.00	0.84	0.00	0.71
RMSE	0.00	0.70	0.00	0.71	0.00	0.67
MAE	0.00	0.00	0.00	0.00	0.00	0.40
MSE	0.00	0.77	0.00	0.17	0.00	0.71
MAPE	0.00	0.23	0.00	0.02	0.00	0.68
SMAPE	0.00	0.29	0.00	0.05	0.00	0.69

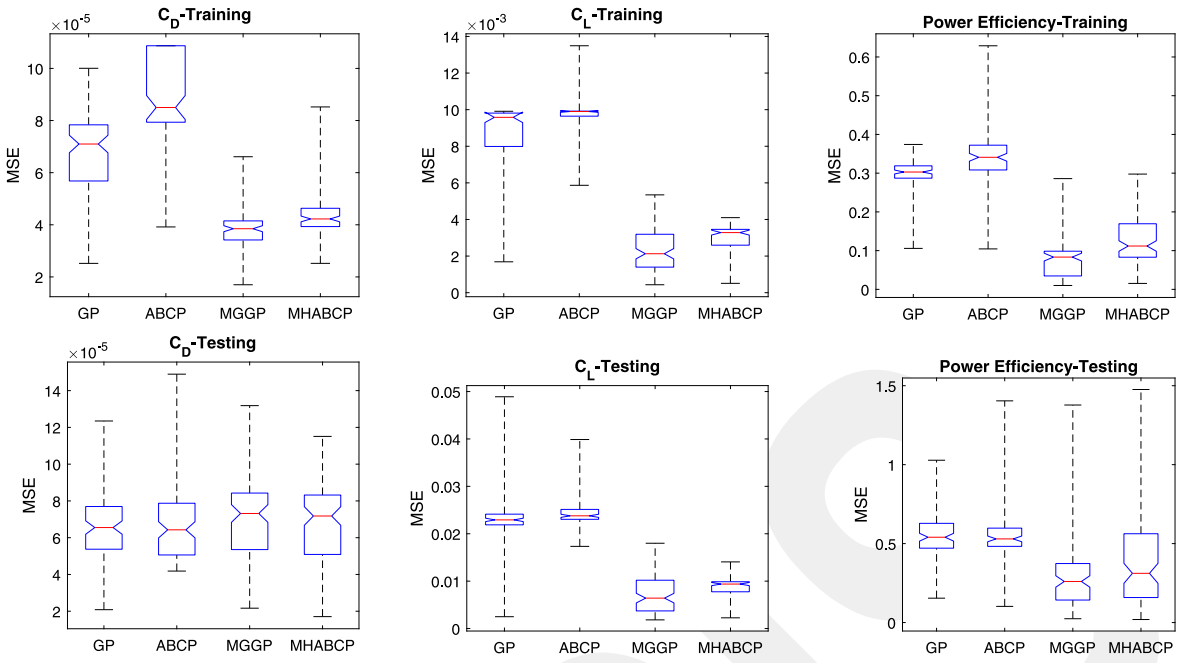


Fig. 12. Distribution of MSEs of the best-of-run individuals.

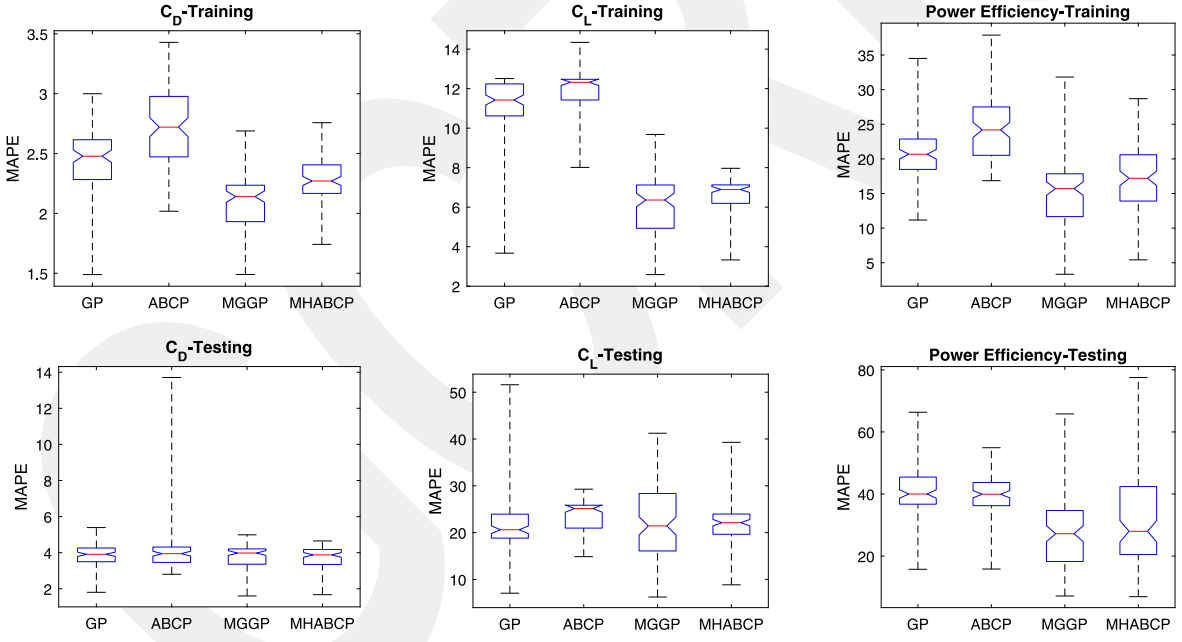


Fig. 13. Distribution of MAPEs of the best-of-run individuals.

Table 3
P-values between MGGP and other automatic programming methods for C_L.

Metric	GP-MGGP		ABCP-MGGP		MHABCP-MGGP	
	C_L Training	C_L Testing	C_L Training	C_L Testing	C_L Training	C_L Testing
SSE	0.00	0.00	0.00	0.00	0.00	0.00
RMSE	0.00	0.00	0.00	0.00	0.00	0.00
MAE	0.00	0.00	0.00	0.00	0.00	0.00
MSE	0.00	0.14	0.00	0.00	0.00	0.00
MAPE	0.00	0.24	0.00	0.03	0.06	0.52
SMAPE	0.00	0.00	0.00	0.00	0.04	0.07

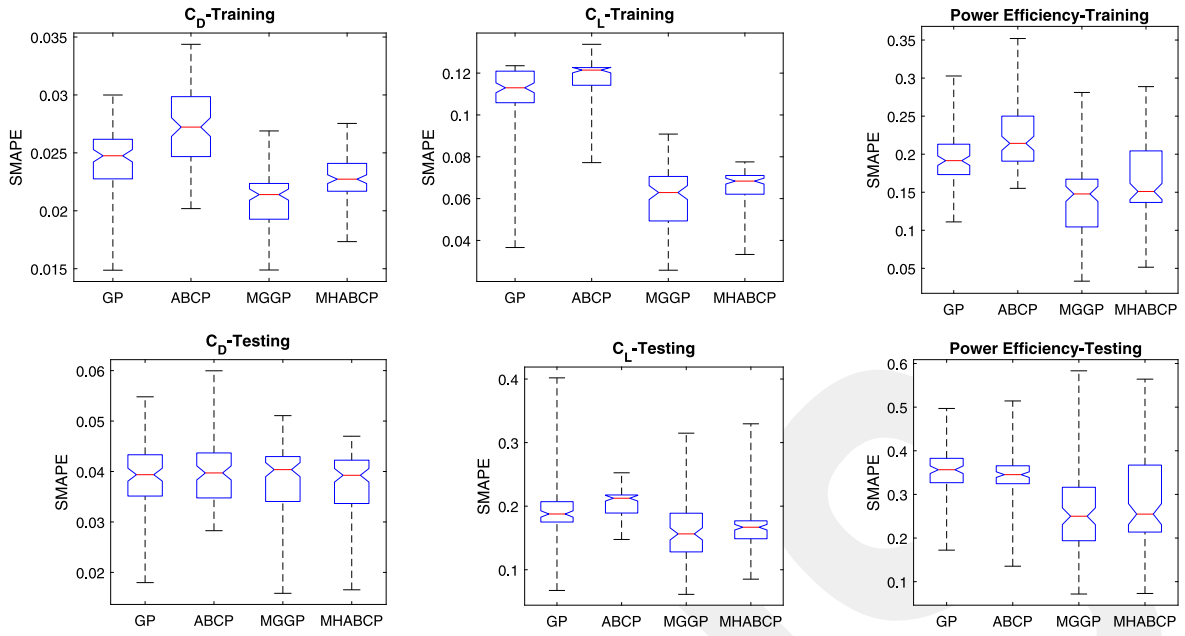


Fig. 14. Distribution of SMAPEs of the best-of-run individuals.

Table 4
P-values between MGGP and other automatic programming methods for PE.

Metric	GP-MGGP		ABCP-MGGP		MHABCP-MGGP	
	PE Training	PE Testing	PE Training	PE Testing	PE Training	PE Testing
SSE	0.00	0.00	0.00	0.00	0.00	0.00
RMSE	0.00	0.00	0.00	0.00	0.00	0.00
MAE	0.00	0.00	0.00	0.00	0.00	0.00
MSE	0.00	0.00	0.00	0.00	0.00	0.01
MAPE	0.00	0.00	0.00	0.00	0.00	0.02
SMAPE	0.00	0.00	0.00	0.00	0.00	0.04

Table 5
P-values between ABCP and GP for all coefficients.

Metric	ABCP-GP					
	C_D Training	C_D Testing	C_L Training	C_LTesting	PE Training	PE Testing
SSE	0.00	0.57	0.00	0.05	0.00	0.00
RMSE	0.00	0.49	0.00	0.02	0.00	0.00
MAE	0.00	0.24	0.00	0.02	0.00	0.00
MSE	0.00	0.16	0.00	0.84	0.00	0.16
MAPE	0.00	0.04	0.00	0.52	0.00	0.77
SMAPE	0.00	0.07	0.00	0.02	0.00	0.90

Table 6
R^2 values for the C_D coefficient.

Metric	GP		ABCP		MGGP		MHABCP	
	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}
Mean	0.99	0.98	0.99	0.95	1.00	0.98	0.99	0.98
Best	1.00	0.99	1.00	0.99	1.00	0.99	1.00	1.00
Worst	0.99	0.96	0.99	0.96	0.99	0.96	0.99	0.97

Table 7
R^2 values for the C_L coefficient.

Metric	GP		ABCP		MGGP		MHABCP	
	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}
Mean	0.93	0.87	0.92	0.86	0.98	0.96	0.98	0.95
Best	0.99	0.99	0.97	0.90	1.00	0.99	1.00	0.99
Worst	0.92	0.73	0.89	0.78	0.96	0.90	0.97	0.92

Table 8
R^2 values for the PE coefficient.

Metric	GP		ABCP		MGGP		MHABCP	
	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}	R^2_{Train}	R^2_{Test}
Mean	0.90	0.49	0.88	0.45	0.99	0.87	0.98	0.86
Best	0.96	0.81	0.96	0.87	1.00	0.97	0.99	0.98
Worst	0.88	0.40	0.87	0.38	0.98	0.78	0.96	0.78

said that there is a significant difference, except for the MAE values of GP and ABCP. For the C_L training (Table 3), it can be noted that the methods have higher values than MGGP, except for the MAPE and SMAPE values of MHABCP. In general, this is also true for C_L and PE testing, except for a few performance criteria.

Table 5 shows that ABCP has higher error values in training compared to GP. However, with the exception of the PE testing, the error values are generally close to those of GP. R^2 values how much the coefficients of the models obtained by the methods can explain the independent parameters (AoA and Re) in Tables 6, 7, and 8.

The R^2 values for the C_D coefficient in Table 6 show that almost all models are in good fit with both training and test data as they

Table 9
Mean/Std time of AP methods.

Problems	GP	ABCP	MGGP	MHABCP
C_D	251.26/69.89	254.42/164.47	63.92/8.59	117.86/30.65
C_L	266.04/55.43	380.27/234.47	72.42/12.42	224.52/27.49
Power efficiency	266.06/63.50	449.05/197.64	71.52/13.73	108.47/16.50

Table 10
Comparative results of the best models for the C_D coefficient.

Criteria		GP	ABCP	MGGP	MHABCP
SSE_{Train}		0.001	0.001	0.001	0.001
SSE_{Test}		0.000	0.000	0.000	0.000
$RMSE_{\text{Train}}$		0.006	0.007	0.005	0.006
$RMSE_{\text{Test}}$		0.005	0.006	0.005	0.004
MAE_{Train}		0.004	0.005	0.004	0.006
MAE_{Test}		0.004	0.005	0.004	0.004
MSE_{Train}		0.000	0.000	0.000	0.000
MSE_{Test}		0.000	0.000	0.000	0.000
$MAPE_{\text{Train}}$		1.489	2.018	1.490	2.423
$MAPE_{\text{Test}}$		1.804	2.804	2.243	1.895
$SMAPE_{\text{Train}}$		0.015	0.020	0.015	0.024
$SMAPE_{\text{Test}}$		0.018	0.028	0.022	0.019
R^2_{Train}		0.995	0.994	0.997	0.995
R^2_{Test}		0.994	0.988	0.994	0.995
Best Model	Nodes	190	191	80	47
	Depth	10	10	5	5
	Complexity	1441	1512	303	162

are close to 1. As even the worst R^2_{Test} values are 0.96 for ABCP and MGGP methods. We can safely conclude that all methods predict the C_D coefficient with great success. By examining the R^2 values for C_L, it is clear that both the mean and worst R^2_{Test} values of GP and ABCP have decreased. However, from Table 7, the R^2 values of the best of all methods for the training and test data are quite close to 1. In Table 8, the mean success of GP and ABCP in the R^2_{Test} values extracted for performance is less than 50%, indicating that the predictive abilities of the models have decreased, but they are assigned an unjustifiably high R^2_{Train} value. However, no overfitting is observed for MGGP and MHABCP. Both methods achieve high success in the training and test data. When the R^2 tables are evaluated along with the boxplots, MHABCP sets very successful models with higher fitness values, while MGGP has lower fitness values. This suggests that the fitness values, while necessary to compare the methods, are not sufficient and that the results should be supported by the R^2 values.

4.2. CPU times of AP methods

Many studies in the literature have compared the AP methods by evaluating the time required by each algorithm. These methods may show an increase in the time required depending on the complexity of the improvement mechanisms within the algorithms (Karaboga et al., 2012; Gorkemli and Karaboga, 2019; Moghaddas et al., 2022). Koza et al. stated that the complexity of the individuals generated in the execution of GP can significantly affect the execution time, leading to uncertainty in the length of the study (Koza et al., 2006). In our study, for a fair comparison in determining the operational complexity, we took the same number of evaluations and presented the different CPU execution times of the algorithms in Table 9. This table shows the mean times and standard deviation (in milliseconds) of 100 runs for each problem. These experiments were performed using Windows 10, Intel Core i7-8565U CPU (1.8 GHz) with 8 GB of RAM.

It is clear from Table 9 that the quickest method among all the methods is MGGP. MGGP is notably faster than the other methods, with a minimum of 1.5 times faster and its variations are also quite minimal.

Table 11
Comparative results of best models for the C_L coefficient.

Criteria		GP	ABCP	MGGP	MHABCP
SSE_{Train}		0.039	0.151	0.024	0.018
SSE_{Test}		0.025	0.173	0.018	0.141
$RMSE_{\text{Train}}$		0.041	0.081	0.032	0.028
$RMSE_{\text{Test}}$		0.050	0.132	0.043	0.119
MAE_{Train}		0.031	0.063	0.027	0.022
MAE_{Test}		0.035	0.103	0.030	0.092
MSE_{Train}		0.002	0.007	0.001	0.001
MSE_{Test}		0.002	0.017	0.002	0.014
$MAPE_{\text{Train}}$		6.526	9.178	4.016	3.717
$MAPE_{\text{Test}}$		7.050	24.787	10.519	15.226
$SMAPE_{\text{Train}}$		0.065	0.092	0.040	0.037
$SMAPE_{\text{Test}}$		0.067	0.194	0.086	0.152
R^2_{Train}		0.986	0.946	0.991	0.993
R^2_{Test}		0.986	0.903	0.990	0.921
Best Model	Nodes	188	192	65	66
	Depth	10	10	5	5
	Complexity	1543	1481	239	243

Table 12
Comparative results of best models for the PE coefficient.

Criteria		GP	ABCP	MGGP	MHABCP
SSE_{Train}		2.971	6.994	0.546	2.776
SSE_{Test}		6.058	4.149	0.588	1.768
$RMSE_{\text{Train}}$		0.479	0.579	0.154	0.379
$RMSE_{\text{Test}}$		0.650	0.627	0.242	0.378
MAE_{Train}		0.246	0.473	0.102	0.217
MAE_{Test}		0.701	0.556	0.218	0.430
MSE_{Train}		0.135	0.104	0.023	0.041
MSE_{Test}		0.154	0.102	0.024	0.019
$MAPE_{\text{Train}}$		13.977	26.075	5.882	10.878
$MAPE_{\text{Test}}$		15.747	15.832	7.145	7.536
$SMAPE_{\text{Train}}$		0.138	0.214	0.058	0.102
$SMAPE_{\text{Test}}$		0.172	0.135	0.072	0.076
R^2_{Train}		0.944	0.957	0.990	0.983
R^2_{Test}		0.808	0.872	0.970	0.976
Best Model	Nodes	81	134	83	67
	Depth	10	10	5	5
	Complexity	560	959	311	246

The next fastest method after MGGP is MHABCP, which demonstrates that these versions are faster than the standard algorithms. In contrast, ABCP is the slowest method. When evaluated by problem, the methods are most efficient in solving the C_D problem. After C_D, the problems are solved in the order of power efficiency and C_L.

4.3. Analysis of the best models

Tables 10, 11, and 12 show the comparative results for all methods to determine the best models with the lowest error and highest R^2 values after 100 runs for C_D, C_L, and PE, respectively.

Analyzing the best models for the C_D coefficient in Table 10, it should be noted that MGGP has the lowest error values, except for 3 criteria ($MAPE_{\text{Train}}$, $MAPE_{\text{Test}}$, $SMAPE_{\text{Test}}$). For R^2 values, all models have high and close fit values. However, MGGP for R^2_{Train} and ABCP for R^2_{Test} are higher than the others. The methods have similar values for almost all performance criteria. Unlike their standard versions, MGGP and MHABCP generate models with very low complexity. The model generated by MGGP is about 4.75 times more complex than that generated by GP, while the model generated by MHABCP is about 9.33 times less complex than that generated by ABCP.

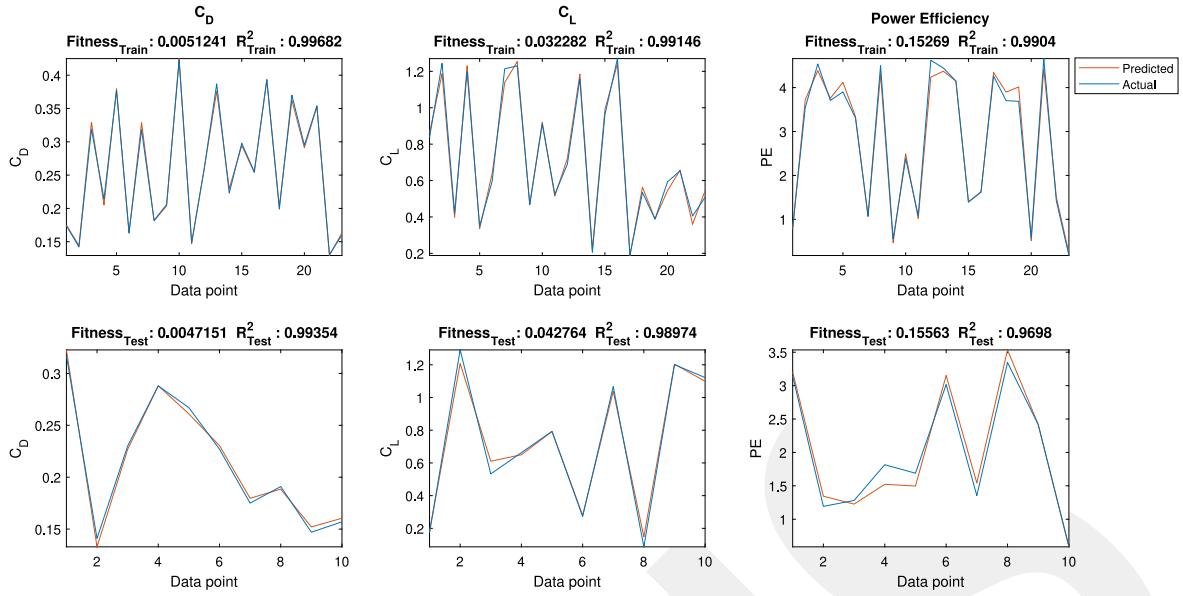


Fig. 15. The relationship between the predicted and actual data points of the best models.

For the coefficient C_L (Table 11), the MGGP model produced lower error values for almost all performance evaluation criteria. However, MGGP has about 1.5 times higher error value for $SMAPE_{\text{Test}}$ than GP. For R^2_{Train} and R^2_{Test} values, the other models, except ABCP, achieved very similar values. The highest R^2_{Test} values belong to MGGP. When we evaluate the number of nodes and the complexity of the trees generated by the models, we find that the model GP is very complex. The MGGP and MHABCP models have complex values that are very close to each other. The MGGP model is about 6.5 times more complex than the GP model ; the MHABCP model is about 6.1 times less complex than ABCP.

Similar to the model analysis of the other coefficients for the PE coefficient (Table 12), MGGP produced a very low error value compared to other methods for SSE_{Train}. MHABCP, which produces the closest error value to MGGP, is about 5 times higher than MGGP. The same is true for SSE_{Test}. For R^2_{Test} , GP and ABCP have lower performance than the versions. The worst method for the coefficient PE is GP.

The simplified formulas of the best models of the MGGP are also given in Table 13. AoA meant the angle of attack, whereas Re stand for Reynolds number in the table. The comparison between the predicted values of these models and the actual values is shown below, as in Fig. 15 which shows that the prediction performance of the models is well-fitted to the training data. At the same time, MGGP produces models with lower error values with a smaller number of training data, which proves that it can predict the coefficients accurately.

Fig. 15, in Fig. 16 scatter plots are drawn to show values for typically two parameters in each dataset. Scatter plots are used to show values for typically two variables for a dataset. In Fig. 16, the data are plotted as blue points, where in each case the value of the actual values determines the position on the horizontal axis and the value of the predicted values determines the position on the vertical axis. The closer the blue points shown in the figure were to the lines, the stronger the correlation between them. When evaluating all the points in Fig. 16, it is clear that the predicted values are systematically increased by the actual values since the blue points cluster near the lines. There is also a strong positive correlation between the actual and predicted values for both the training and test data for all coefficients.

Table 13
Best formulas obtained by MGGP.

Simplified Formulas
$C_D = 4.449e-5(\text{AoA}-5.876)\left(\text{AoA}^2+2.0\text{AoA}-9.241\right)+\frac{1.613e+14(\text{AoA}^3-854.5)(851.1\text{AoA}-7865.0)}{5.765e+17\text{AoA}^3+1.153e+18\text{Re}-9.178e+22}+\frac{1.906e+15(\text{AoA}+59.94)(\text{AoA}^2+\text{Re})}{2.882e+17\text{AoA}^3+2.882e+17\text{Re}+1.654e+18}-\frac{9.967e+14(\text{AoA}+\text{Re})(\text{AoA}^3-846.8)(\text{AoA}-9.66)}{2.951e+20\text{AoA}^3+5.903e+20\text{Re}-4.699e+25}-0.2495$
$C_L=\frac{\frac{3.294e+15\text{AoA}(\text{AoA}-1.0\text{Re})}{\text{Re}}-2.251e+15}{2.679e+12\text{Re}-1.419e+17}-\frac{1.0(3.02e+14\text{AoARe}-2.644e+17)}{2.815e+14\text{AoA}-\frac{2.815e+14\text{Re}}{\text{AoA}}+9.3e+15}-\frac{1.141\text{AoAReé}^2}{(\text{AoA}^2-1.0\text{Re})^2}-\frac{5.731e+14\text{AoA}(\text{AoA}-1.0\text{Re})(\text{AoA}+2.032)}{9.007e+15\text{AoA}+9.007e+15\text{Re}+1.81e+16}+0.1977$
$PE=\frac{1.027e+15\text{AoA}+1.027e+15\text{Re}}{3.436e+10\text{AoA}+3.436e+10\text{AoA}^2-3.436e+10\text{Re}}-\frac{1.0(2.448e+15\text{AoA}^2+2.448e+15\text{AoA})}{3.508e+10\text{Re}-1.037e+15}-\frac{1.0(1.295e+15\text{AoA}-1.295e+15\text{AoA}^2)}{1.754e+10\text{Re}-5.24e+14}-\frac{49650.0}{\text{Re}}+\frac{5.222e+15\text{AoA}(2.0\text{AoA}+7.837)}{2.749e+11\text{AoA}-2.749e+11\text{Re}}+\frac{1.027e+15\text{AoA}(\text{AoA}+7.837)}{1.718e+10\text{AoA}+1.718e+10\text{Re}}+29890$

5. Discussion

In this study, we have proposed mathematical models using AP methods to predict aerodynamic coefficients. One of the main strengths of these methods is that they can formulate the problem without requiring a prior model, unlike classical algorithms. Despite the many studies that have been conducted, as mentioned in Section 1, there is still a research gap in using AP methods and verifying their success in aerodynamic research. In addition to their good performance, the methods have also proposed models for aerodynamic coefficients that depend on dependent parameters.

From the boxplots and the tables of statistical test results, it can be noted that MGGP method predicts the most accurate models. It has also been shown that this method solves the problems faster than the other three methods in Section 4.

In the following, we present the limitations of all methods and some proposed solutions from some studies to overcome them:

- *Scalability*: AP methods can be very computationally intensive and have difficulty solving large or complex problems. Parallel computing can be used to solve large and complex problems. However, since we had limited data, we did not use parallel computing in our study.
- *Early convergence*: AP methods can converge to local solutions before reaching the global optimum. However, convergence rates can also be slow. Therefore, versions of the methods with other optimization methods or other improvement operators are recommended. For example, AP versions used in our study (MGGP and

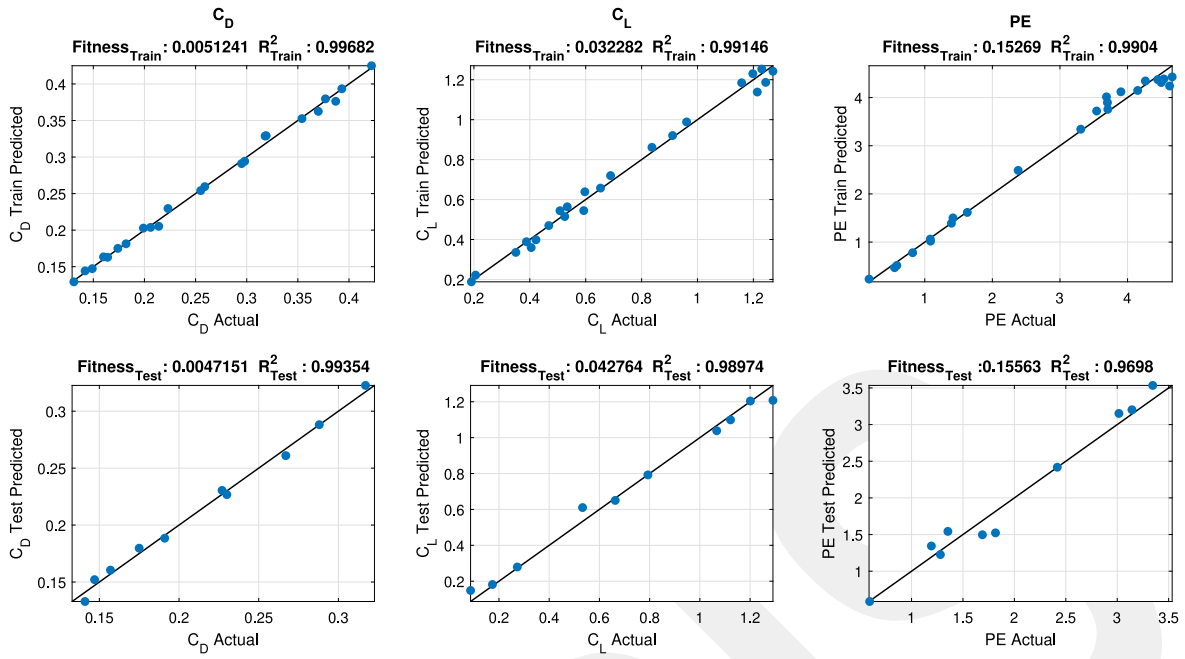


Fig. 16. Best models prediction scatterplots.

MHABCP) produced faster and more successful models than the standard versions (GP and ABCP) by overcoming this limitation. In addition, studies have been conducted in the literature to increase the convergence rate of AP methods (Nekoei et al., 2021; Boudardara and Gorkemli, 2020).

- *Dependence on initial solutions*: The methods of AP can optimize randomly generated initial solutions. Therefore, parameter tuning can be used to increase the chances of finding the global optimum. In our study, we try different parameters (population size, functions, etc.) to generate initial solutions for all methods.

6. Conclusions

The goal of this study is to investigate the best AP method for predicting the aerodynamic coefficients and power efficiency of wind turbine blade at low Reynolds numbers and different angles of attack. For this purpose, two AP pioneer methods, including GP and ABCP, are used in conjunction with their MGGP and MHABCP versions.

The results clearly show that MGGP has the best performance in predicting the aerodynamic coefficients and power efficiency compared to GP, ABCP and MHABCP (MGGP R^2 values in C_D Train: 0.997-Test: 0.994, in C_L Train: 0.991-Test: 0.990, in PE Train: 0.990-Test: 0.970). Thus, this study clearly shows that the prediction of aerodynamic coefficients and power efficiency can be correctly guaranteed by MGGP without the need for numerous experimental tests, which are time-consuming and costly.

CRediT authorship contribution statement

Sibel Arslan: Methodology, Software, Validation, Writing – reviewing, Supervision. **Kemal Koca:** Methodology, Investigation, Data analysis, Conceptualization, Writing – reviewing & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Abdmouleh, Z., Alammari, R.A., Gastli, A., 2015. Review of policies encouraging renewable energy integration & best practices. Renew. Sustain. Energy Rev. 45, 249–262.

Akbıyık, H., Yavuz, H., 2021. Artificial neural network application for aerodynamics of an airfoil equipped with plasma actuators. J. Appl. Fluid Mech. 14 (4), 1165–1181.

Arslan, S., Ozturk, C., 2019. Artificial bee colony programming descriptor for multi-class texture classification. Appl. Sci. 9 (9), 1930.

Arslan, S., Öztürk, C., 2019. A comparative study of automatic programming techniques. Informatica 43 (2).

Arslan, S., Ozturk, C., 2019a. Feature selection for classification with artificial bee colony programming. In: Ser, J.D., Villar, E., Osaba, E. (Eds.), Swarm Intelligence. IntechOpen, Rijeka, http://dx.doi.org/10.5772/intechopen.85219.

Arslan, S., Ozturk, C., 2019b. Multi hive artificial bee colony programming for high dimensional symbolic regression with feature selection. Appl. Soft Comput. 78, 515–527.

Belamadi, R., Djemili, A., Ilinca, A., Mdouki, R., 2016. Aerodynamic performance analysis of slotted airfoils for application to wind turbine blades. J. Wind Eng. Ind. Aerodyn. 151, 79–99.

Bleischwitz, R., de Kat, R., Ganapathisubramani, B., 2015. Aspect-ratio effects on aeromechanics of membrane wings at moderate Reynolds numbers. AIAA J. 53 (3), 780–788.

Bouaziz, S., Dhahri, H., Alimi, A.M., Abraham, A., 2016. Evolving flexible beta basis function neural tree using extended genetic programming & hybrid artificial bee colony. Appl. Soft Comput. 47, 653–668.

Boudardara, F., Gorkemli, B., 2020. Solving artificial ant problem using two artificial bee colony programming versions. Appl. Intell. 50 (11), 3695–3717.

Cramer, N.L., 1985. A representation for the adaptive generation of simple sequential programs. In: Proceedings of an International Conference on Genetic Algorithms and the Applications. pp. 183–187.

Desai, C.K., Shaikh, A., 2012. Prediction of depth of cut for single-pass laser micro-milling process using semi-analytical, ANN and GP approaches. Int. J. Adv. Manuf. Technol. 60 (9), 865–882.

Descoteaux, P.-O., Olivier, M., 2021. Performances of vertical-axis hydrokinetic turbines with chordwise-flexible blades. J. Fluids Struct. 102, 103235.

Elsakka, M.M., Ingham, D.B., Ma, L., Pourkashanian, M., 2019. CFD analysis of the angle of attack for a vertical axis wind turbine blade. Energy Convers. Manage. 182, 154–165.

Emeksiz, C., 2022. Multi-gen genetic programming based improved innovative model for extrapolation of wind data at high altitudes, case study: Turkey. Comput. Electr. Eng. 100, 107966.

Genç, M.S., Açıkel, H.H., Koca, K., 2020. Effect of partial flexibility over both upper and lower surfaces to flow over wind turbine airfoil. Energy Convers. Manage. 219, 113042.

Genç, M.S., Açıkel, H.H., Koca, K., 2022a. Experimental investigation on effect of partial flexibility at low aspect ratio airfoil–Part I: Installation on suction surface. In: EPJ Web of Conferences, Vol. 269. EDP Sciences, p. 01017.

Genc, M.S., Kemal, K., Acikel, H.H., 2019. Investigation of pre-stall flow control on wind turbine blade airfoil using roughness element. Energy 176, 320–334.

Genç, G., Koca, K., Genç, M.S., 2022b. Unsteady aerodynamics over surface of a chambered airfoil at stall angle and low Reynolds number. In: EPJ Web of Conferences, Vol. 269. EDP Sciences, p. 01016.

Gonçalves, I., Silva, S., Fonseca, C.M., 2015. On the generalization ability of geometric semantic genetic programming. In: European Conference on Genetic Programming. Springer, pp. 41–52.

Gorkemli, B., Karaboga, D., 2019. A quick semantic artificial bee colony programming (qsABCP) for symbolic regression. Inform. Sci. 502, 346–362.

Hosseini, A.S., Hajikarimi, P., Gandomi, M., Nejad, F.M., Gandomi, A.H., 2021. Genetic programming to formulate viscoelastic behavior of modified asphalt binder. Constr. Build. Mater. 286, 122954.

Huang, Y.-F., Gan, X.-J., Chiueh, P.-T., 2017. Life cycle assessment and net energy analysis of offshore wind power systems. Renew. Energy 102, 98–106.

Karaboğa, D., 2014. Yapay zeka optimizasyon algoritmalari. Nobel Akademik YayıNcıLıK 245.

Karaboga, D., Ozturk, C., Karaboga, N., Gorkemli, B., 2012. Artificial bee colony programming for symbolic regression. Inform. Sci. 209, 1–15.

Karasu, İ., Açikel, H.H., Kemal, K., Genç, M.S., 2020. Effects of thickness and camber ratio on flow characteristics over airfoils. J. Therm. Eng. 6 (3), 242–252.

Kawazoe, H., Morita, S., 2004. Ground effect on the dynamic characteristics of a wing-rock delta wing. In: 34th AIAA Fluid Dynamics Conference and Exhibit. p. 2352.

Koca, K., Genç, M.S., Açıkel, H.H., 2022a. Experimental investigation on effect of partial flexibility at low aspect ratio airfoil-Part II: Installation both on suction and pressure surface. In: EPJ Web of Conferences, Vol. 269. EDP Sciences, p. 01028.

Koca, K., Genç, M.S., Açıkel, H.H., Çağdaş, M., Bodur, T.M., 2018. Identification of flow phenomena over NACA 4412 wind turbine airfoil at low Reynolds numbers and role of laminar separation bubble on flow evolution. Energy 144, 750–764.

Koca, K., Genç, M.S., Bayır, E., Soğuksu, F.K., 2022b. Experimental study of the wind turbine airfoil with the local flexibility at different locations for more energy output. Energy 239, 121887.

Koca, K., Genç, M.S., Ertürk, S., 2022c. Impact of local flexible membrane on power efficiency stability at wind turbine blade. Renew. Energy 197, 1163–1173.

Koca, K., Genc, M.S., Veerasamy, D., Özden, M., 2022d. Experimental flow control investigation over suction surface of turbine blade with local surface passive oscillation. Ocean Eng. 266, 113024.

Koza, J.R., 1994. Genetic programming as a means for programming computers by natural selection. Stat. Comput. 4 (2), 87–112.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G., 2006. Genetic Programming IV: Routine Human-Competitive Machine Intelligence, Vol. 5. Springer Science & Business Media.

Kütük, N., Arslan, S., 2022. Biosorption of methyl orange from aqueous solution with hemp waste, investigation of isotherm, kinetic and thermodynamic studies and modeling using multigene genetic programming. Chem. Pap. 76 (12), 7357–7372.

Luo, J., Vanhoucke, M., Coelho, J., Guo, W., 2022. An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem. Expert Syst. Appl. 198, 116753.

MacPhee, D.W., Beyene, A., 2019. Performance analysis of a small wind turbine equipped with flexible blades. Renew. Energy 132, 497–508.

Meana-Fernández, A., Díaz-Artos, L., Fernández Oro, J.M., Velarde-Suárez, S., 2020. An optimized airfoil geometry for vertical-axis wind turbine applications. Int. J. Green Energy 17 (3), 181–195.

Moghaddas, S.A., Nekoei, M., Golafshani, E.M., Behnood, A., Arashpour, M., 2022. Application of artificial bee colony programming techniques for predicting the compressive strength of recycled aggregate concrete. Appl. Soft Comput. 130, 109641.

Moradkhani, M., Hosseini, S., Olazar, M., Altzibar, H., Valizadeh, M., 2021. Estimation of the minimum spouting velocity and pressure drop in open-sided draft tube spouted beds using genetic programming. Powder Technol. 387, 363–372.

Nekoei, M., Moghaddas, S.A., Golafshani, E.M., Gandomi, A.H., 2021. Introduction of ABCEP as an automatic programming method. Inform. Sci. 545, 575–594.

Papadimitriou, D.I., Papadimitriou, C., 2016. Aerodynamic shape optimization for minimum robust drag and lift reliability constraint. Aerosp. Sci. Technol. 55, 24–33.

Poli, R., Langdon, W., Mcphee, N., 2008. A Field Guide to Genetic Programming. ISBN: 978-1-4092-0073-4.

Salmasi, F., Yıldırım, G., Masoodi, A., Parsamehr, P., 2013. Predicting discharge coefficient of compound broad-crested weir by using genetic programming (GP) and artificial neural network (ANN) techniques. Arab. J. Geosci. 6 (7), 2709–2717.

Sattar, M., Majid, A., Kausar, N., Bilal, M., Kashif, M., 2022. Lung cancer prediction using multi-gene genetic programming by selecting automatic features from amino acid sequences. Comput. Biol. Chem. 98, 107638.

Stanovov, V., Akhmedova, S., Semenkin, E., 2022. The automatic design of parameter adaptation techniques for differential evolution with genetic programming. Knowl.-Based Syst. 239, 108070.

Tumse, S., Bilgili, M., Sahin, B., 2022. Estimation of aerodynamic coefficients of a non-slender delta wing under ground effect using artificial intelligence techniques. Neural Comput. Appl. 1–22.

Wahidi, R., Bridges, D., 2009. Experimental investigation of the boundary layer and pressure measurements on airfoils with laminar separation bubbles. In: 39th AIAA Fluid Dynamics Conference. p. 4278.

Xing, H., Xiao, Z., Qu, R., Zhu, Z., Zhao, B., 2022a. An efficient federated distillation learning system for multitask time series classification. IEEE Trans. Instrum. Meas. 71, 1–12.

Xing, H., Xiao, Z., Zhan, D., Luo, S., Dai, P., Li, K., 2022b. SelfMatch: Robust semisupervised time-series classification with self-distillation. Int. J. Intell. Syst. 37 (11), 8583–8610.

Yuan, D., Zhang, D., Yang, Y., Yang, S., 2022. Automatic construction of filter tree by genetic programming for ultrasound guidance image segmentation. Biomed. Signal Process. Control 76, 103641. http://dx.doi.org/10.1016/j.bspc.2022.103641, URL https://www.sciencedirect.com/science/article/pii/S174680942200163X.