Effectiveness of Nature-Inspired Algorithms using ANFIS for Blade Design Optimization and Wind Turbine Efficiency

Md. Rasel Sarkar 1, Sabariah Julai 1,* , Chong Wen Tong 1, and Siti Fauziah Toha 2

1 Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia; raselbdeee@gmail.com (M.R.S.); chong_wentong@um.edu.my (C.W.T.)
2 Department of Mechatronics Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, 53100 Gombak, Selangor, Malaysia; tsfauziah@iium.edu.my

* Correspondence: sabsz@um.edu.my; Tel.: +603-79674461

Received: 8 February 2019; Accepted: 26 March 2019; Published: 1 April 2019

Abstract: Blade design of the horizontal axis wind turbine (HAWT) is an important parameter that determines the reliability and efficiency of a wind turbine. It is important to optimize the capture of the energy in the wind that can be correlated to the power coefficient ($C_p$) of HAWT system. In this paper, nature-inspired algorithms, e.g., ant colony optimization (ACO), artificial bee colony (ABC), and particle swarm optimization (PSO) are used to search for the blade parameters that can give the maximum value of $C_p$ for HAWT. The parameters are tip speed ratio, blade radius, lift to drag ratio, solidity ratio, and chord length. The performance of these three algorithms in obtaining the optimal blade design based on the $C_p$ are investigated and compared. In addition, an adaptive neuro-fuzzy interface (ANFIS) approach is implemented to predict the $C_p$ of wind turbine blades for investigation of algorithm performance based on the coefficient determination ($R^2$) and root mean square error (RMSE). The optimized blade design parameters are validated with experimental results from the National Renewable Energy Laboratory (NREL). It was found that the optimized blade design parameters were obtained using an ABC algorithm with the maximum value power coefficient higher than ACO and PSO. The predicted $C_p$ using ANFIS-ABC also outperformed the ANFIS-ACO and ANFIS-PSO. The difference between optimized and predicted is very small which implies the effectiveness of nature-inspired algorithms in this application. In addition, the value of RMSE and $R^2$ of the ABC-ANFIS algorithm were lower (indicating that the result obtained is more accurate) than the ACO and PSO algorithms.

Keywords: optimization; blade design parameters; coefficient of performance; ant colony optimization; particle swarm optimization; artificial bee colony; ANFIS

1. Introduction

Renewable energy, such as wind energy, biomass energy, hydro energy, and solar energy have been extensively harnessed and exploited during the last decade because of environmental concerns. The most promising source of renewable energy is wind energy due to the factor of low cost in comparison with other sources of renewable energy, such as solar energy and biomass energy, etc. [1,2]. Wind energy is harnessed via the use of a wind turbine system that converts mechanical energy into electrical energy by using a generator. It is classified into various types of systems, such as constant and variable speed systems, power controlling systems, and off grid or on grid [3] systems. Based on the rotation of the axis, there are two types of wind turbines such as the vertical axis wind turbine (VAWT) and the horizontal axis wind turbine (HAWT). In this paper, the HAWT blades are the main concern.
The power coefficient of the blades, which can be defined as the capture capability of efficiency, is the most basic index of wind energy [4]. Design parameters selections are critical for optimization of wind turbine performance. Various parameters influence the energy production of wind turbines, such as, rotor rotational velocity, wind speed, and blade pitch angle [5,6]. For the power coefficient optimization of wind turbine blades, the influence of lift to drag ratio, blade radius, tip-speed ratio, solidity ratio, and the chord length of the blade has been widely investigated. Besides, nature-inspired algorithms (ACO, ABC, and PSO) have also been used in various wind turbine applications.

Eroglu and Ulusam [7] determined the wind farm layout using ACO. They found maximum energy output that considered wake loss, wind turbine location, and wind direction. Fuchs and Gjengedal [8] applied ACO for the necessary time step resolution in a transmission expansion and wind power integration in the Nordic area. They determined the average and peak values for power production from wind technology. Jovanovi et al. [9] focused on maximum segregating of computed graphs of supply and demand. For the optimization, they used ACO and found that the error was less than 5% in comparison with the optimal solutions. Abd-Allah et al. [10] investigated the lightning point in wind turbine farm as lightning is harmful to wind turbine farm. They used ACO to search for the sensitive points in wind farm. Mustafar et al. [11] studied the loss of reduction of the transformer tap setting to control reactive power using ACO technique.

In the last few years, artificial bee colony (ABC) has massively been studied for a complex problem, modeling, and optimization. The ABC algorithm is not only used in optimization but also for prediction and control. Delgarm et al. [12] determined the performance of energy building using the multi-objective artificial bee colony optimization approach. In addition, Song et al. [13] used an improved ABC algorithm to optimize the combustion of a boiler to consider NOx emission. Ebrahim et al. [14] used the ABC algorithm for proportional integral derivative (PID) controller parameters optimization for speed control of an AC driver. Derakhshan et al. [15] optimized the wind turbine blade shape to determine the aerodynamic performance of blades. Habbi [16] proposed a fuzzy logic system that was self-generated by the ABC algorithm. Based on the above reviews, the ABC algorithm is vastly used for complex problem solving due to its excellent search capability.

Many researchers have also been using the PSO algorithm for location optimization. Safaei et al. [17] proposed the new two-step PSO algorithm for the placement of wind turbine generators for maximum allowable capacity and minimizing power loss in wind turbines. Wind turbine placement at a wind farm was optimized by Wan et al. [18]. Pookpunt and Ongsaku [19] used the binary PSO for wind turbine optimum placement at a wind farm. The size optimization of the hybrid system was determined by PSO [20].

ANFIS is one kind of neural network that shows better learning and estimation capabilities [21]. In the power coefficient prediction or estimation, there are various forms of artificial intelligence. Artificial neural networks (ANN) are user-friendly tools which can learn the mathematical mapping between input and output variables of a nonlinear system. Dalibor and Zarko [22] applied adaptive neuro-fuzzy for the power coefficient estimation. Tip-speed ratio and pitch angle were considered for the power coefficient estimation. They found that the optimal power coefficient was around 0.36. Rajakumar and Ravindran [5] determined the power coefficient and lift and drag coefficient for various airfoil (NASA) using computational fluid dynamics. They showed that the power coefficient level was higher when the drag coefficient was set to zero and the correction of tip loss factor was set to be a constant. Shahaboddin and Dalibor [23] used support vector regression (SVR) based on kernel function for estimating power coefficient and minimizing the generalization error bound. They found similarities in the results of ANFIS and ANN. Sedaghat and Mirhosseini [24] implemented blade element momentum theory (BEM) for a power coefficient of 300 kW in HAWT technology in the province of Semnan in Iran. They obtained the maximum power coefficient of 0.51 when the tip-speed ratio was up to its optimum value of 10.

In summary, there is no particular study which focuses on the optimization of the power coefficient wind turbine blades by using three algorithms. The study aims to find the most appropriate blades
parameters that provide the maximum power coefficient using ACO, ABC, and PSO. An ANFIS is used to investigate the effectiveness of these nature-inspired algorithms by comparing the results predicted from ANFIS with the results obtained from the nature-inspired algorithm [25]. An endeavor is prepared for retrieving the correlation between \( C_p \) and the best combination of optimized blade parameters such as lift and drag ratio, blade radius, tip speed ratio, solidity ratio, and chord length of the blade of HAWTs.

2. Power Coefficient of Horizontal Axis Wind Turbine Blade

The progress of the present commercial wind power generator has been continuously moving forward to the latest megawatt (MW) turbine. For HAWTs, parameter selection is challenging. The production of wind turbine power is influenced by various fixed parameters, such as wind velocity, the chord length of blades, rotor diameter, and lift to drag ratio, etc. [4]. There are two goals of the design of a HAWT, i.e., optimizing and estimating the power coefficient [26]. Recently, attention has been placed on a rotor of wind turbine design for maximum aerodynamic performance [27–29]. The mechanical power equation for a HAWT can be written as

\[
P_{\text{mechanical}} = 0.5 \rho C_p A_s V_w^3
\]

where air density is expressed by \( \rho \) in \((\text{kg/m}^3)\), wind velocity in \( V_w \)(\text{ms}^{-1}), and \( C_p \) is known as the rotor efficiency or the power coefficient \((C_p)\). Wind energy conversion is directly dependant on the \( C_p \) of the aerodynamic system which is converted from wind energy to electrical power. The power coefficient can be written as [24,30,31]

\[
C_{\text{optimization}}(\lambda, \frac{C_L}{C_D}, \sigma, r, c) = \left( \frac{16}{27} \right) \left[ \lambda + \left( \frac{1.32 + \left( \frac{\lambda - 8}{20} \right)^2}{\left( \sigma A_s \right)^{\frac{3}{2}}} \right)^{-1} - \frac{(0.57)\lambda^2}{C_L C_D} \right]^{\frac{3}{2}}
\]

where the swept area of the turbine rotor is \( A_s \) in \((\text{m}^2)\) and \( \lambda \) is the tip-speed ratio. \( R, C_D \) and \( C_L \) are the blades radius, drag, and lift coefficient blade airfoil, respectively. The wind turbine coefficient strongly depends on the rotor blade performance and airfoil section. The blade is an important part of the HAWT. The blade design is an important part of the HAWT design process. There are two types of blade designs: aerodynamic and structural design. Both designs are important for HAWT performance [32]. The aerodynamic efficiency, annual energy production (AEP), and power performance are those aspects accounted for in aerodynamic design. On the other hand, the structural design is concerned by material, mass, fatigue load, stability, etc. [33]. The theoretical maximum power coefficient is \( C_{\text{max}} = 0.59 \). The power coefficient of modern wind turbine reaches up to 0.51 which is close to the Betz limit [31]. The power coefficient directly depends on the lift to drag ratio of HAWT blades. The power coefficient is varying with tip speed ratio as well as lift to drag ratio [34]. For each aerodynamic airfoil, \( C_L \) and \( C_D \), depends on the attack angle and Reynolds number. Solidity ratio of blades can be defined as [5,35]

\[
\sigma = \text{solidity ratio} = \frac{\text{Number of blades} \times \text{area of each blades}}{A_s}
\]

Lift and drag coefficients are dimensionless numbers that are used for measurement of aerodynamic lift and drag forces. It can be defined as,

\[
C_L = \frac{L}{0.5 p V_w^2 A_s}
\]

\[
C_D = \frac{D}{0.5 p V_w^2 A_s}
\]
Therefore, the lift to drag ratio can be defined as. The lift to drag ratio is also called a sliding ratio.

\[ \varepsilon = \frac{C_L}{C_D} \]  

(6)

The tip speed ratio of a WT can be expressed as [36],

\[ \lambda = \frac{R \omega_m}{V_w} \]  

(7)

where \( \omega_m \) is the rotational speed of the rotor in (rad/s) and \( V_w (m/s) \) is the wind velocity. The radius of the blade is \( R \). For the optimization and prediction, the range of the tip-speed ratio is 3 to 10.


Population-based optimization algorithms, which are developed and influenced heavily by nature, find near-optimal solutions to difficult optimization problems. The population consisting of possible solutions to the problem is modified by applying some operators on the solutions depending on the information of their fitness. This step is a common feature in all population-based algorithms. As a result, in the area of search space, the population is moved towards a better solution. Evolutionary algorithms and swarm intelligence-based algorithms are the two important classes of population-based optimization algorithms. Nonetheless, any collection of interacting agents or individuals is considered to be a swarm method [25]. Nature-inspired algorithms are those algorithms inspired by nature. In this paper, ACO, ABC, and PSO algorithms are utilized for optimization processes.

3.1. Ant Colony Algorithm

ACO was first expressed by Dorigo and Gambardella [37] and was later modified and presented as an optimization technique by Shen et al. [38] and Dorigo et al. [39]. If the ants have found a food source, they will carry out some evaluation about the size of the source and carry a percentage of total food to the nest of ant, while sending off some pheromones on the way back, familiar as a pheromone trail. This pheromone trail provides the chance to the other ants of the same nest to find the food source. By the same way, other ants follow the same nest to reach the food source. The total amount of pheromone collected on the ground is directly proportional to the quantity and quality of the base source that the ants discover [40]. Since the pheromone is like a vaporizable substance, the quantity of pheromone will decrease over time [41]. Therefore, the indication path of the ants for food collection and pheromone trails stay on the path. Based on the methodology, the shorter path is the priority of the pheromone trail. Indeed, the ants are collecting their food by the shortest path. The optimization method is based on updating the pheromone path of the better solution. The problem of optimization can be solved by the support of an artificial ant colony by using information through pheromones deposited on graph edges.

Assume the vector \( X = [x_1, x_2, \ldots, x_n] \) are the parameters of optimization, where the total number of parameters in \( n \), the lower and upper bounds are to be \( x_i \in D(x_i) = [x_{i,low}, x_{i,up}] \) with \( i = 1, 2, \ldots, n \). The definition field \( D(x_i) \) is divided into \( M \) subspaces, and the middle of each subspace defines a node. A single artificial ant \( k (k = 1, 2, \ldots N_{ant}) \), where \( N_{ant} \) is the maximum number of ants, would choose to move from one node to the other, in the total of \( P \) nodes in each \( D(x_i) \). The length of each sub-space \( h_i \) an be expressed by

\[ h_i = \frac{x_{i,up} - x_{i,low}}{M} \]  

(8)

For each level which has \( P \) nodes on it, there are \( M \times n \) nodes in total. \( k \) is the state vector of ant in the entire tour, as shown in Figure 1 with travel index \([i_8, i_7, i_6, \ldots, i_4]\). The travel index is directly dependent on the cumulative probability \( (CP) \) from the probability \( P_{ij} \) of the ant \( k \) to move the \( i \)th node on the \( j \)th level. For example, if \( M = 10 \), \( CP = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \), and the generated random number lies between 0.8 and 0.9, the first travel index, \( i_8 \), is chosen as 8 (eighth
column of the CP). These processes will be continued until all travel index is found. The values of the parameters $X$, held by ant, are as follows

$$[x_1, x_2, \ldots, x_n] = [x_{1\text{low}} + i_8 \times h_8, x_{2\text{low}} + i_7 \times h_7, x_{3\text{low}} + h_6 \times i_6, \ldots, x_{n\text{low}} + h_n \times i_n]$$

(9)

Figure 1. State space graph of ACO.

The rule of state transition of the ant $k$ is defined as

$$P_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^{n} \tau_{kj}}$$

(10)

where the ant probability $P_{ij}$ moves to the $i$th node on the $j$th level. The pheromone at the node is $\tau_{ij}$.

The pheromone is updated by using the Equation (11), when all ants finished their tours

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \frac{Q}{f_{\text{best}}}$$

(11)

where the pheromone decay parameter range is $0 < \rho < 1$, $Q$ is the quantity of pheromone laid by an ant per cycle, $\tau_0$ is a constant for the initial value of $\tau_{ij}$ (for initialization $\tau_{ij}$ on the right-hand-side is set to be $\tau_0$), and $f_{\text{best}}$ is function of objective. From the objective function, the best value is given by each ant searching period [42].

3.2. Artificial Bee Colony

ABC is based on the swarm intelligence optimization algorithm which is proposed and developed by Karaboga [43, 44]. In this algorithm, there are three types of bees, such as employed, onlooker, and scouts. Half of the colony bees are onlookers and the other half of bees are employed. The number of employed bees is equal to the amount of food source [45]. The employed bees pass information to onlooker bees about the food sources. According to the obtained information from employed bees, the onlooker bees select the food sources and find new food sources near to the selected food sources [46]. When the selected food sources have been mined out, the employed bees become scout bees. Those scout bees randomly find out new food sources to replace the one which has been mined out [45, 47, 48]

$$X_{ij} = X_{ij}^{\text{min}} + \text{rand}(0, 1)\left(X_{ij}^{\text{max}} - X_{ij}^{\text{min}}\right)$$

(12)

where $i = 1, 2, \ldots, SN, j = 1, 2, \ldots, D, SN = CS/2$ is the number of food sources is equal to the number of employed bees [48]. In addition, employed bees are equal to the onlooker bees. $X_{ij}^{\text{min}}$ and $X_{ij}^{\text{max}}$ are the lower and upper boundary of parameters of optimization problem. $D$ is the number of problem parameter $\text{rand}(0, 1)$ is the random number distribution within [0,1].
After initialization, the employed bees try to detect new food sources with the nearest areas. They saved in memory until creating new food sources to following the equation

\[ v_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \]  (13)

where \( j = 1, 2, \ldots, D \) and \( k = 1, 2, \ldots SN \) are the random selected indices. \( X_{ij} \) give randomly different solution form to \( X_{kj} \). In addition, a random integer number represented by \( \phi_{ij} \) within \([-1, 1]\). The parameters value can be acceptable if the value produced by operation exceeds its predetermined boundaries. So, the boundaries are set as: if \( X_i > X_i^{max} \) then \( X_i = X_i^{max} \); if \( X_i > X_i^{min} \) then \( X_i = X_i^{min} \).

After producing \( v_i \) within the boundaries, a fitness value for a maximization problem can be assigned to the solution of following equation

\[ fitness_i = \begin{cases} 
\frac{1}{1 + f_i} & \text{if } f_i \geq 0 \\
1 + \text{abs}(f_i) & \text{if } f_i < 0 
\end{cases} \]  (14)

where \( f_i \) is the objective value of the solution \( v_i \). For maximization problems, the objective function can be directly used as a fitness function. If a greedy selection is applied between \( X_i \) and \( v_i \), then the better one is selected depending on fitness values presenting the nectar amount of the food sources at \( X_i \) and \( v_i \). If the source at \( v_i \) is superior to that of \( X_i \) in terms of profitability, the employed bee memorizes the new position and forgets the old one. Otherwise the previous position is kept in her memory. If \( X_i \) cannot be improved, the counter holding the number of trails is incremented by 1, otherwise the counter resets to 0.

In the end, employed bees finish their investigation. Their information has been shared with onlooker bees on the dance area about the amount of nectar as well as the positions of their sources. This is a feature for the multiple interactions of artificial bees of ABC. An onlooker bee investigates the nectar information which is taken from all employed bees. The nectar information picks a site of food source with a probability value linked to its nectar amount (fitness) \( p_i \)

\[ p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \]  (15)

The probabilistic choice relies on the fitness values of the results in the population. In this probabilistic selection scheme, the number of observers visiting them arise because of the increase in the amount of the nectar food source (the fitness of solutions). This is one of the advantages and positive feedback aspects of ABC.

A random number (real) \( r_i \) is within the range of 0 and 1 and generated for every source. If the probability value \( p_i \) in Equation (15) associated with that source is greater than this random number \( r_i(p_i > r_i) \), then the onlooker bee produces a modification on the position of this food source site by using Equation (13) as in the case of the employed bee. After the evolution of the source, the greedy selection scheme is used and the onlooker bee either remembers the new position by forgetting the old one or maintains the old one. If the solution for \( X_i \) is not improved, the counter holding trials can be incremented by 1. Or else, the counter is reset to 0. This procedure is repeated until all onlookers can be distributed onto the sites of the food source.

In a cycle, after the entire employed bees and onlooker bees finish their searches, the algorithm verifies if there is an exhausted source to be abandoned. In order to settle on if a source is to be abandoned, the updated counters are used. These counters have been updated during the search. For greater counter value than the control parameter of the ABC algorithm (known as the limit), the source related to the counter is supposed to be exhausted and also abandoned.

The food source exhausted by its bee is restored with another food source revealed by the scout, which signifies the mechanism for negative feedback and unsteadiness property in self-organization of ABC. This can be simulated by creating a site position at random and reinstating it with the abandoned
and exhausted one. It is supposed that \( X_i \) is the abandoned source. At random, the scout discovers a fresh and new food source to replace the \( X_i \). This operation is expressed as the ABC first equation. In each cycle of the fundamental of ABC, it has been considered that only one source has been abandoned or exhausted. On the other hand, only one employed bee is a scout. It is also observed that if more than one counter goes over the “limit” value, one of the maximum can be programmatically selected.

### 3.3. Particle Swarm Optimization

PSO is the meta-heuristics approach. Meta-heuristics are used in a problem where the optimization problem is unable to solve it. As PSO is the meta-heuristics type, it can, therefore, solve more complex problems [49,50]. Firstly, the PSO algorithm was established by Kennedy and Eberhard in the year of 1995 [51,52]. In the complex mathematical problem, binary or discrete problems and continuous problems are solved by the PSO. Particle swarm optimization is the multi-parallel searching technique to obtain optimum results. PSO has been inspired by the natural “bird flocking” or fish schooling. In this algorithm, the set of swarms or particle have been described by their positions, and velocity vectors fly throughout the search space.

The particle motion in PSO is defined by the velocity vector in each direction. The best solution achieved by all particles is called the best global particle. All swarm position and velocity are updated by the best global particle and the obtained experience. The experience sharing between particle and swarm is the vital reason behind PSO success [53].

At starting period, the random population of swarms is generated with random position vectors and velocity of vectors \( X_i = (x_{i1}, x_{i2}, \ldots x_{id}) \) and \( V_i = (v_{i1}, v_{i2}, \ldots v_{id}) \) respectively. The best position of each particle is \( P_i = (p_{i1}, p_{i2}, \ldots p_{id}) \) according to the best fitness value obtained by the particle time \( t \). After iteration, the new positions and velocities to the particle for the next fitness evolution are calculated by the following equations [54]

\[
\begin{align*}
  v_{id}(t+1) &= \omega \times v_{id}(t) + c_1 \times \varphi_1 \times (p_{id} - x_{id}(t)) + c_2 \times \varphi_2 \times (p_{gd} - x_{id}(t)) \\
  x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1)
\end{align*}
\]  
(16)

where \( \omega \) is the inertia weight that plays a vital role to ensure the convergent behaviour of \( \varphi_1 \) and \( \varphi_2 \) which are uniform and random numbers with interval \((0, 1)\). \( c_1 \) and \( c_2 \) are constants of acceleration, \( p_{ibest} \) is the personal best particle, and \( p_{gbest} \) is the best position of particle. The position of each parameter is illustrated in Figure 2.

![Particle swarm optimization algorithm.](image-url)

The best position of the particle is the population so far [55]. The previous velocity can be represented by the first part of Equation (16). This equation provides the required momentum for the particles to roam across the search space. The second part of the Equation (16) is known as the “cognitive” part or component and it stands for the personal thinking of each particle. With the help of the cognitive component, the particles are encouraged to move toward their own best positions that
are found. The third part of the Equation (16) is known as the “social” component to find the optimal global solution. The social component represents the effect of the collaboration of the particles. This part constantly drags the particles in the direction of the global best particle found so far.

A standard ACO has three control parameters, i.e., (i) the information heuristic factor, (ii) the expectation heuristic factor, and (iii) pheromone evaporation factor. In standard PSO, it also has three control parameters, i.e., (i) cognitive factor, (ii) social factors, and (iii) inertia weight. On the other hand, a standard ABC algorithm has only one control parameter, i.e., limit. In this paper, the expression for determining the value of limit is only depends on population (colony size) and the dimension of the problem (i.e. maximum cycle number). Apparently, these two common control parameters also exist in ACO and PSO. Therefore, ABC is the simplest algorithm among all.

There are several weaknesses of using meta-heuristic method such as ACO, PSO, and ABC. They tend to be trapped into local optima while doing the search. The process also requires long computational time with slow convergence speed. Although this method can find many parameters, it is difficult to encode the scheme. Despite these disadvantages, this method is capable of solving deterministic combinatorial problems, also combinatorial problems characterized by uncertainty [56,57].

4. Adaptive Neuro-Fuzzy Interface System

ANFIS system can be categorized to discrete control system by on-line identification, an approximation of a highly nonlinear function and it can predict a chaotic time series. The fuzzy “IF-THEN” rules, and predetermined inputs and outputs are used to construct for ANFIS system. The member functions (MFs) of ANFIS are created by input and output parameters. ANFIS obtains the FIS through the input, output, and algorithm of back propagation. FIS is executed by the combination of a database, a rule base, as well as the reasoning mechanism. Firstly, fuzzy logic finds the rule base. The database allocates the MFs that works in the rules of fuzzy logic. Finally, the reasoning mechanism reduces from the input data and rules which come to the feasible outcome. It can be adjusted to better performance for changing the environment. These systems have similar-human intelligence within a specific domain [58].

The ANFIS (training and evaluation) systems have been employed in the MATLAB. In ANFIS system, there are five input parameters to influence the power coefficient of the HAWT blade as shown in Table 1. The ‘IF-THEN’ rules of Takgi and Sugeno’s class of fuzzy logic with two inputs for the first order Sugeno are employed for the purpose of this study [59,60].

<table>
<thead>
<tr>
<th>Input/output Parameters</th>
<th>Input Variables</th>
<th>ABC</th>
<th>ACO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip Speed Ratio</td>
<td>3 to 10 [24]</td>
<td>6.4479</td>
<td>6.0032</td>
<td>6.00</td>
</tr>
<tr>
<td>Blade Radius</td>
<td>1 to 5 [61]</td>
<td>4.016</td>
<td>4.3089</td>
<td>4.50</td>
</tr>
<tr>
<td>Lift to drag ratio</td>
<td>1 to 110 [61]</td>
<td>109.4848</td>
<td>109.9935</td>
<td>110.00</td>
</tr>
<tr>
<td>Solidity ratio</td>
<td>0.01 to 0.45 [35]</td>
<td>0.3885</td>
<td>0.35</td>
<td>0.45</td>
</tr>
<tr>
<td>Chord length</td>
<td>0.01 to 0.45 [62]</td>
<td>0.193</td>
<td>0.232</td>
<td>0.40</td>
</tr>
<tr>
<td>Power coefficient</td>
<td></td>
<td>0.529</td>
<td>0.52</td>
<td>0.52</td>
</tr>
</tbody>
</table>

If \( x \) is \( P \) and \( y \) is \( R \), then

\[
 f_1 = p_1 x + q_1 y + r_1
\]

The input parameters of MFs that are made up of the 1st layer are providing the input values to the following layer. Each node is considered as an adaptive node having a node function \( 0 = \mu_{PQ}(x) \)
and $0 = \mu_{RS}(x)$, where $\mu_{PQ}(x)$ and $\mu_{RS}(x)$ are MFs. The maximum value of Triangle MFS is (1.0) and (0.0) is the minimum value. Like as,

$$
\mu(x) = \text{triangle}(x; p_i, q_i, r_i) = \begin{cases} 
0, & x < p_i \\
\frac{x - p_i}{q_i - p_i}, & p_i \leq x \leq q_i \\
\frac{r_i - x}{r_i - q_i}, & q_i \leq x \leq r_i \\
0, & r_i \leq x 
\end{cases} 
$$

(19)

where the set of parameters are $\{p_i, q_i, r_i\}$ and $x$ and $y$ are the inputs to nodes and it signifies to tip speed ratio, blade radius, lift and drag ratio, solidity, chord length and the power coefficient.

In ANFIS system, the second layer of ANFIS is called a membership function. It looks for the weights of each function of membership. The first layer sends a signal to the membership function. The nodes of MFs are referred to as non-adaptive. The MFs layer refers to as a multiplier to receive to signal and send out the outcome in $w_i = \mu_{PQ}(x) \times \mu_{PQ}(y)$ form. Output nodes are represented the firing strength of a rule.

The third layer is called a rule layer. In this layer, all the neurons perform as the pre-condition matching fuzzy rules. Every rule’s activation level is calculated by the several fuzzy rules which are equal to the number of layers. This layer’s node calculates the weights of normalized and it is considered as non-adaptive. Each node calculates the value of the rule’s firing strength over the sum of the rules firing strengths in the form of $w_i = \frac{w_i}{w_1 + w_2}, i = 1, 2$. The results are mentioned to as the normalized firing strengths.

The 4th layer is known as the defuzzification layer. It is countable to provide the outcome values as the result of the inference of rules. Every node of the layer is an adaptive node having the following node function, $\theta_i^j = w_i^j x f = w_i^j (p_i x + q_i y + r_i)$. Here, $\{p_i, q_i, r_i\}$ are the set of variable. This set variable is nominated as the consequent parameters.

$$
\theta_i^j = \sum_i w_i^j x f = \frac{\sum_i w_i f}{\sum_i w_i} 
$$

(20)

The final layer is called an output layer of ANFIS. The final layer obtains the inputs from the previous layer. After that, this layer is converted the fuzzy classification outputs into a binary (crisp) but this is not a non-adaptive layer. The structure of the ANFIS is shown in Figure 3. The total output is calculated by summing all signal received from the nodes.

![Figure 3. The structure of ANFIS.](image)

5. Results and Discussion

5.1. Convergence Graph Power Coefficient and Computational Time

The convergence rate which is one of the key factors determines the reliability of an algorithm. It is usually considered as the higher the rate of convergence, the better the algorithm if only the outcomes converge and fall within the acceptable range. As discussed before, the nature-inspired
algorithm produces the highest maximized objective function ($C_p$) values from Equation (2) for every case, and thus, is directly ascertaining its accuracy. In this section, the manner of how fast the algorithm converges and attains its highest maximized value in the sequence of the number of iterations are investigated and discussed.

Figure 4a shows the convergence graph for all algorithms with 20 populations using ABC, ACO, and PSO algorithms. The maximum iteration has been varied from 100, 200, 300, 400, 500, and 600 iterations. It can be observed that the PSO has achieved a speedy convergence in the initial stage (achieved in less than 20 iterations). However, the searching for this algorithm tends to become stagnant after less than 10 iterations. When the search result comes at stagnant, all particles tend to assemble together around the global minimum, and thus, the global best result is not improved for time being. For ACO, it can be observed the search started at the lower value of power coefficient in comparison to ABC and PSO. However, the ants kept on searching the optimum value in less than around 350 iterations, which performed much better than the PSO, but became stagnant, or had no significant changes towards the end of the maximum iteration. For the ABC, the searching pattern shows the acceleration in convergence speed. For all cases, the convergence speed accelerated can be seen in less than around 200 iterations but became stagnant towards the end of the maximum iteration. It can be concluded that in terms of $C_p$ value, ABC has performed well in obtaining the highest value for all cases.

Figure 4b refers to the time taken for the algorithm to finish from the first until the maximum number of iteration. The computation time of all algorithms is all in second (s). From Figure 4b, it can be observed that the PSO algorithm has taken ±1 s to finish the simulation for all cases. For the ABC algorithm, it needed around 2 s to finish the simulations. ACO needed more time than the ABC and PSO to complete the simulation, i.e., around 2.7 s.

From Figure 5a, the convergence graphs are obtained with 50 populations for all algorithms with the iteration 100, 200, 300, 400, 500 and 600. PSO has achieved a speedy convergence in the initial stage but became stagnant from 10 iterations and onwards. For ACO, it can be noticed that the convergence graph of ACO is speedy after 50 iterations. After 50 iterations, ACO is not really improved until the maximum iteration. For ABC algorithms, the population kept on searching the optimum value in less than 250 iterations. With 50 populations, the $C_p$ obtained is almost similar to the result with 20 populations. The ABC algorithm has achieved the maximum power coefficient in comparison to PSO and ACO.

Figure 5b shows the computational time for 50 population’s size for all algorithms. The algorithm of ACO took the longest time, i.e., 3.6 s as compared to ABC and PSO. On the other hand,
Computational time in Figure 4b refers to the time taken for the algorithm to finish from the first until the maximum number of iteration. The computation time of all algorithms is all in second (s). From Figure 4b, it can be observed that the PSO algorithm has taken ±1 s to finish the simulation for all cases. For the ABC algorithm, it needed around 2 s to finish the simulations. ACO needed more time than the ABC and PSO to complete the simulation, i.e., around 2.7 s.

From Figure 5a, the convergence graphs are obtained with 50 populations for all algorithms with the iteration 100, 200, 300, 400, 500 and 600. PSO has achieved a speedy convergence in the initial stage but became stagnant from 10 iterations and onwards. For ACO, it can be noticed that the convergence graph of ACO is speedy after 50 iterations. After 50 iterations, ACO is not really improved until the maximum iteration. For ABC algorithms, the population kept on searching the optimum value in less than 250 iterations. With 50 populations, the \( C_p \) obtained is almost similar to the result with 20 populations. The ABC algorithm has achieved the maximum power coefficient in comparison to PSO and ACO.

![Convergence graph](image1)

![Computational time](image2)

**Figure 5.** (a) Convergence graph; (b) computational time of ABC, ACO, and PSO with 50 populations.

Figure 5b shows the computational time for 50 population’s size for all algorithms. The algorithm of ACO took the longest time, i.e., 3.6 s as compared to ABC and PSO. On the other hand, the PSO algorithm needed a much shorter time as compared to the ABC and ACO algorithms, which took around 0.6 s to complete the simulations.
Figure 6a shows the convergence progress for all algorithms with 100 populations. The maximum iteration varied from 100, 200, 300, 400, 500, and 600 iterations. It can be observed that the PSO achieved a speedy convergence in the initial stage (achieved in less than 50 iterations). For the ACO algorithm, it kept on searching for the optimum value within 100 iterations, and the result became stagnant after 250 iterations onwards. The final objective for PSO and ACO achieved an almost similar value. For ABC, the searching pattern shows an acceleration in the convergence speed. The acceleration in the convergence speed can be seen in less than 250 iterations. For ABC, it can be observed that the search started at the lower value of $C_p$ in comparison with ACO and PSO. In terms of $C_p$ value, the consistency of the ABC algorithm has been observed in comparison with ACO and PSO. It signifies that the ABC has outperformed PSO and ACO, with the highest maximize value of 0.5295 among all the compared algorithms ($ACO = 0.52$ and $PSO = 0.52$).

![Convergence Graph](image1)

**Figure 6.** (a) Convergence graph; (b) computational time of ABC, ACO, and PSO with 100 populations.

From Figure 6b, PSO has completed the maximum iteration number in a shorter computational time, i.e., around 2.1 s in comparison to the ACO and ABC. The ABC algorithm has taken more time to complete the simulation compared to the ACO algorithm. It can be concluded that the computational time of all algorithms has taken within 6 s and it is acceptable and manageable for simulation purposes.
5.2. Prediction of Power Coefficient Using ANFIS

The best input combination of ABC, ACO, and PSO algorithms from Table 1 are used in ANFIS for the prediction of the power coefficient. To achieve the goal, 60% of the data is employed for training and 40% data is used for testing.

The scatter of Figure 7 showed the prediction of the power coefficient values via the ANFIS model against measured values of ABC optimization by Equation (2). Figure 7a provides the results for the training data set where it has been observed that most of the points drop along with the diagonal line. Figure 7b shows the testing data set where most of the points are very close to the diagonal line. From here, the predicted values via the ANFIS are very close to the measured values.

The predicted power coefficient training and testing values using ANFIS model for a wind turbine blade are presented in Figure 7c,d, respectively. These figures illustrate the training and testing values using the ANFIS model for a power coefficient and measured values of ACO optimization data. All values of data set for training and testing are very close to the diagonal line. Figure 7e,f show the training and testing data set of ANFIS, respectively. These data set of training and testing are measured from the best input combination of optimized values of PSO.

![Figure 7](image-url)
Most of the training and testing data are sometimes below the diagonal line and sometimes above the diagonal line. As seen in Table 2, the \( R^2 \) correlation coefficients are near to 1. As a result, the proposed ANFIS model used for this investigation has a very good correlation with the power coefficient measured for training and testing data sets of three algorithms ABC, ACO, and PSO. Note that the \( R^2 \) is higher for the ABC-ANFIS power coefficient as shown in Table 2 to compare ACO-ANFIS and PSO-ANFIS, which translates to a better correlation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th></th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE(W/m²)</td>
<td>( R^2 )</td>
<td>RMSE(W/m²)</td>
</tr>
<tr>
<td>ABC-ANFIS</td>
<td>0.00654</td>
<td>0.999</td>
<td>0.363</td>
</tr>
<tr>
<td>ACO-ANFIS</td>
<td>0.00544</td>
<td>0.9989</td>
<td>0.814</td>
</tr>
<tr>
<td>PSO-ANFIS</td>
<td>0.235</td>
<td>0.9711</td>
<td>1.911</td>
</tr>
</tbody>
</table>

An ANFIS method is very helpful for the faster prediction of power coefficient with blade parameters. Figure 8 shows the wind power coefficient comparison between predicted and measured values by best input combination of ABC algorithm optimized parameters using ANFIS technique. From Figure 8, the measured (optimized) value \( (C_p = 0.5295) \) is very close to the predicted value, i.e., \( C_p = 0.5215 \) with small variation.

Figure 9 illustrates the predicted and optimized values of the ACO algorithm using ANFIS model for a power coefficient of the wind turbine blade. It can be observed that the predicted and measured values are almost similar whereas the predicted and optimized value is 0.5175 and 0.52, respectively.
An ANFIS method is very helpful for the faster prediction of power coefficient with blade parameters. Figure 8 shows the wind power coefficient comparison between predicted and measured values by the best input combination of ABC algorithm optimized parameters using ANFIS technique. From Figure 8, the measured (optimized) value ($C_p = 0.5295$) is very close to the predicted value, i.e., $C_p = 0.5215$ with small variation.

Figure 8. Predicted (ANFIS) vs. optimized (ABC) of the power coefficient ($C_p$).

Figure 9 illustrates the predicted and optimized values of the ACO algorithm using ANFIS model for a power coefficient of the wind turbine blade. It can be observed that the predicted and measured values are almost similar whereas the predicted and optimized value is 0.5175 and 0.52, respectively.

Figure 9. Predicted (ANFIS) vs. optimized (ACO) of the power coefficient ($C_p$).

Figure 10 shows the predicted values 0.5135 and the measured values 0.52 of PSO algorithm for the power coefficient of wind turbine blades. Figure 11 shows the predicted power coefficient comparison between three algorithms (ABC, ACO, and PSO) values. The prediction of the power coefficient of the best input combination for getting the value of the ABC algorithm is higher compared to the ACO and PSO algorithms. Thus, the proposed ANFIS model has obtained high accuracy for predicting the wind turbine power coefficient of the three algorithms.

Table 2 approaches the results of the statistical analysis for the proposed ANFIS model for both training and testing values for ABC, ACO, and PSO. Several statistical indicators have been used to evaluate the proposed model performance, i.e., RMSE and $R^2$ as shown in Table 2. It should be noted that the units of RMSE and $R^2$ are dimensionless, respectively. Based on the results, the difference in RMSE and $R^2$ values for training and testing are relatively small. Therefore, it delivers additional evaluation and verification on the suitability of the proposed ANFIS model to estimate the power coefficient. The investigation results recommended that the proposed ANFIS model can effectively be implemented for prediction for all cases of ABC, ACO, and PSO algorithms optimization. Based on the ANFIS results, the ABC algorithm shows better accuracy in both optimization and prediction.
coefficient of the best input combination for getting the value of the ABC algorithm is higher compared to the ACO and PSO algorithms. Thus, the proposed ANFIS model has obtained high accuracy for predicting the wind turbine power coefficient of the three algorithms.

**Figure 10.** Predicted (ANFIS) vs. optimized (PSO) of the power coefficient ($\eta$).

**Figure 11.** Prediction of ABC, ACO, and PSO.

Table 2 approaches the results of the statistical analysis for the proposed ANFIS model for both training and testing values for ABC, ACO, and PSO. Several statistical indicators have been used to evaluate the proposed model performance, i.e., RMSE and $R^2$ as shown in Table 2. It should be noted that the units of RMSE and $R^2$ are dimensionless, respectively. Based on the results, the difference in RMSE and $R^2$ values for training and testing are relatively small. Therefore, it delivers additional evaluation and verification on the suitability of the proposed ANFIS model to estimate the power coefficient. The investigation results recommended that the proposed ANFIS model can effectively be implemented for prediction for all cases of ABC, ACO and PSO algorithms optimization. Based on the ANFIS results, the ABC algorithm shows better accuracy in both optimization and prediction as compared to ACO and PSO algorithm results. The ANFIS model has a very small RMSE (0.363) during testing and the value is slightly higher (0.00654) in training of the ABC-ANFIS model for the power coefficient. It has been observed that the ANFIS model showed constantly good correlation throughout the testing and training. It has been outperformed by ANFIS in terms of the effect on the wind turbine power coefficient. Therefore, the study results suggested that the proposed ANFIS model can effectively be embedded for the prediction of all cases of ABC, ACO, and PSO algorithm optimizations. Based on the ANFIS results, the ABC algorithms show better accuracy for both optimization and prediction compared to the ACO and PSO algorithms results.

### 5.3. Validation

Table 3 shows that the validation of this present investigation with related literatures. Sedaghat et al. [24] carried out the investigation of 300 kW horizontal axis wind turbine for aerodynamic design in the province of Semnan. They used the blade element momentum theory (BEM) for the blade “Airfoil is RISO-A1-18”. The power coefficient was found to be 0.51 which was almost similar to our present investigation. Rajakumar et al. [5] optimized the power coefficient of the wind turbine rotor using the blade “Airfoil NACA 4410” and “NACA 2415” by CFD analysis. The power coefficients of the blade “Airfoil NACA 4410” and “NACA 2415” were found to be 0.48 and 0.45, respectively. The obtained power coefficients of both airfoils were slightly lower than the present investigation. Based on the supervisory control and data acquisition (SCADA) data, Dia et al. [63] investigated the power coefficient of wind turbines. The discovered power coefficient was 0.508, which was almost similar to our present investigation. Table 3 shows that the power coefficients of the present investigation are...
consistent and reliable with the findings of the other researchers. Therefore, it can be concluded that the power coefficient obtained from the literature supports our present study.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Theory</th>
<th>Blade Model</th>
<th>Maximum CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Investigation</td>
<td>ABC algorithms</td>
<td>Airfoil S822</td>
<td>0.529</td>
</tr>
<tr>
<td></td>
<td>ACO algorithms</td>
<td></td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>PSO algorithms</td>
<td></td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>ABC-ANFIS</td>
<td></td>
<td>0.5215</td>
</tr>
<tr>
<td></td>
<td>ACO-ANFIS</td>
<td></td>
<td>0.5175</td>
</tr>
<tr>
<td></td>
<td>PSO-ANFIS</td>
<td></td>
<td>0.5135</td>
</tr>
<tr>
<td>Validation</td>
<td>Blade element momentum theory (BEM) [24]</td>
<td>Airfoil RISO-A1-18</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>CFD analysis [5]</td>
<td>NACA 4410</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>Supervisory control and data acquisition (SCADA) system [63]</td>
<td>NACA 2415</td>
<td>0.45</td>
</tr>
</tbody>
</table>

6. Conclusions

In conclusion, the blade design parameters have been obtained through optimization by nature-inspired algorithms (ABC, ACO, and PSO) with the objective function to maximize the $C_p$. The prediction of power coefficient has also been carried out by using an ANFIS approach to satisfy the optimized results. The optimization process is employed on a non-linear maximization problem. ACO, PSO, and ABC have successfully been implemented to find the power coefficient for a wind turbine. The results have shown that ACO, PSO, and ABC have the ability to find the best combination of the five inputs of the HAWT blade parameters to get the optimum value of the power coefficient 0.520, 0.520, and 0.5295, respectively. It can be concluded that the ABC algorithm has performed well to find the maximum value of the power coefficient compared to ACO and PSO.

It can be observed that the predicted values of the power coefficient for ABC = 0.5215, ACO = 0.5175 and PSO = 0.5135 are very close to the objective value ($C_p$), i.e., ABC = 0.5295, ACO = 0.52, and PSO = 0.52. Therefore, it can be concluded that the presented prediction (ABC-ANFIS, ACO-ANFIS and PSO-ANFIS) are acceptable based on the RMSE and $R^2$. The values of $R^2$ are near to value 1 which are suitable for the better prediction results. Based on the ANFIS results, the effectiveness of ABC algorithm is more accurate for both optimization and prediction in comparison to ACO and PSO algorithms. By using ANFIS, it can be used for more complex problem solutions, such as accuracy, identification, and prediction of the power coefficient of the HAWT blades. The results have been obtained by optimization and prediction results and can be used for further design and prototype horizontal axis wind turbine modeling. The comparison with similar literature for the power coefficient also shows the consistent results, which supports our present investigation.

Author Contributions: All authors discussed and agreed on the idea and contribution. M.R.S. contributed to the writing manuscript, mathematical modeling, simulation, and prediction. S.J. contributed to optimizing the model and simulation. S.J., C.W.T., and S.F.T. contributed to the revisions and discussion of the results.

Funding: This research was funded by Ministry of Higher Education of Malaysia and University Malaya (ERGS No. ER0142013A and High Impact Research Grant (HIR-D000006-16001)).

Acknowledgments: The authors would like to express their gratitude to the Ministry of Higher Education of Malaysia and University Malaya (ERGS No. ER0142013A and High Impact Research Grant (HIR-D000006-16001)) for funding and providing facilities to conduct the research.

Conflicts of Interest: The authors declare no conflict of interest.

References


