

Research Article

Optimization Design of Wind Turbine Blade Based on an Improved Particle Swarm Optimization Algorithm Combined with Non-Gaussian Distribution

Fangjin Sun ^{1,2,3}, Zhonghao Xu ⁴, and Daming Zhang ^{5,6}

¹Guangxi Key Laboratory of Geomechanics and Geotechnical Engineering, Guilin University of Technology, Guilin 541004, China

²Guangxi Key Laboratory of New Energy and Building Energy Saving, Guilin 541004, China

³College of Civil Engineering and Architecture, Guilin University of Technology, Guilin 541004, China

⁴College of Civil Engineering, Liaoning Technical University, Fuxin 123000, China

⁵Guangxi Key Laboratory of Embedded Technology and Intelligence, Guilin 541006, China

⁶College of Information Science and Engineering, Guilin University of Technology, Guilin 541006, China

Correspondence should be addressed to Daming Zhang; e_horse@163.com

Received 3 January 2021; Revised 29 September 2021; Accepted 6 October 2021; Published 23 October 2021

Academic Editor: Valeria Vignali

Copyright © 2021 Fangjin Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To overcome the problem of particle swarm optimization (PSO) being trapped in local minima, a particle swarm optimization algorithm combined with non-Gaussian stochastic distribution is presented for optimization design of wind turbine blade. Before updating the particle velocity, a limited test was performed for every particle to search for the global best solution. Taking the maximum wind turbine annual power generation as the final objective, a 1.3 MW wind turbine blade was optimized. The results were compared with those of the original wind turbine blades and traditional particle swarm optimization. Compared with the original output power, the year output power increased by 5.3% with non-Gaussian stochastic distribution combined with PSO, whereas the computation time was 65% of the traditional PSO. As time step increased, residuals of non-Gaussian stochastic distribution combined with PSO greatly diminished, with improved computation efficiency. It is shown that the non-Gaussian distribution combined with PSO ensures the global best solution. The non-Gaussian distribution combined with PSO provides a more reliable theoretical basis for the design of wind turbine blades.

1. Introduction

Wind turbine is a kind of power machinery which converts wind energy into mechanical energy. Blade is one of the most important parts of wind turbine. The aerodynamic shape of blade determines the conversion efficiency of wind energy. Therefore, the optimization design technology of aerodynamic shape of wind turbine blade plays an essential role in the design and manufacture of wind turbine. The optimal design of wind turbine blades involves complex aerodynamic performance calculation and searching optimization process, in which the optimization algorithm is one of the vital indicators that directly determine the quality of blade design [1, 2].

For the optimization algorithm research of blade optimal design, genetic algorithm (GA) and particle swarm optimization algorithm (PSO) are commonly adopted all around the world. The PSO algorithm is a new and efficient group intelligent optimization algorithm developed in recent years [3]. Because the PSO algorithm searches for the optimal goal by individual experience and group experience, there is no mutation link, and it is easy to implement. In recent years, it has been favored by scientific researchers and has been widely developed. PSO algorithm has also been applied to the optimal design of different wind turbine blades [4–6]. Liao et al. [7] adopted an improved PSO algorithm to optimize the blade extreme load. Based on multidisciplinary optimization theory, the PSO algorithm was applied by

Ceruti [8] to optimize the geometry of small wind turbine blades to obtain the optimal chord distribution and airfoil. Ma et al. established a time-domain coupled calculation model of floating wind turbines based on PSO and FAST programs and proposed a combined wind turbine blade optimization design method [9]. Posteljnik et al. used PSO to optimize the design of the maximum vertical turbine power and the minimum blade mass of the small vertical axis turbines and added the target of the minimum wind speed, but it led to the increase of Pareto solutions [10].

However, the biggest shortcoming of the PSO algorithm is that in the iterative process, all solutions are optimized, which will lead to premature convergence, and the global search cannot be performed well, thus falling into the situation of local solution convergence [11]. At present, the particle swarm optimization algorithm is mainly improved by introducing the ideas of other intelligent algorithms, such as the shuffled frog-leaping algorithm [12], hybrid genetic coding algorithm [13], and hybrid differential evolution algorithm [14]. Scholars have proposed new type of PSO algorithms, such as PSO-DLTA [15], DMPSO [16], MPSEOG [17], and WI-PSO [18]. Zhao and Li [19] proposed a new two-stage multiswarm particle swarm optimizer (TMPSO), which employs a multiswarm method to adopt two-stage different search strategies in the entire iterative process. Lim et al. [20] proposed a PSO based on self-adaptive topology connection (SATC), in which the SATC module adaptively modifies its topology connectivity in different search stages to promote each particle to perform search with different exploration and development intensity. Nevertheless, only a few scholars have applied the improved PSO algorithm to wind turbine research. Based on the improved particle swarm optimization algorithm, Cao [21] established an optimal control model to maximize the overall output power of the wind farm and employed the particle swarm optimization algorithm to optimize the axial inductance of each unit to determine the rotation speed and the pitch angle when the power of the wind farm is maximized. Chen et al. [22] combined an improved particle swarm optimization algorithm and an airfoil parameterized expression method based on general integration theory to replace and optimize the existing commonly used thin airfoil NACA64618, which improved the aerodynamic and static aeroelastic performance of the new airfoil. Compared with the original algorithm, the search ability of the improved algorithm has been improved to a certain extent, but there are still deficiencies in the global search ability. Lévy flight is a random search process following the Lévy distribution, which is a non-Gaussian random process. The local and global search capabilities of Lévy flight are more prominent, which is suitable for applications in the field of optimization. Liu et al. [23] adopted dynamic penalty function and vertical flight to further improve the artificial bee colony algorithm. In the search stage of the cuckoo algorithm, He et al. [24] adopted the reverse learning of the solution of Lévy flight to effectively improve the search efficiency of the optimal solution.

In order to overcome the above shortcomings of the traditional particle swarm optimization algorithm, this paper solves the problem that the particle swarm optimization algorithm easily falls into the local optimum. By combining

the particle swarm optimization algorithm with non-Gaussian random Lévy distribution, the jumping ability of the particles is increased, so that the particles have the ability to jump out of the local optimum when they fall into the local optimum, so as to seek the global optimum. This paper combines non-Gaussian random distribution with particle swarm optimization to optimize the design of wind turbine blades. Section 2 introduces the related theories of traditional particle swarm optimization algorithm and particle swarm optimization algorithm combined with non-Gaussian distribution. In Section 3, based on the theory of blade element momentum (BEM), an optimized design program was developed using the particle swarm optimization algorithm combined with non-Gaussian random distribution (hereinafter referred to as “non-Gaussian PSO”). In Section 4, with the goal of the maximum annual power generation of the wind turbine, the optimal design of a 1.3 MW wind turbine blade is carried out, and the results of non-Gaussian PSO are compared with the optimization results of the original wind turbine blades and traditional PSO. Finally, Section 5 provides a general conclusion.

2. Methodology

2.1. Particle Swarm Optimization Algorithm. Particle swarm optimization (PSO) is an optimization algorithm proposed by Kennedy and Eberhart in 1995 [25]. In the PSO algorithm, the potential solutions to the considered optimization problem are called particles and fly through the solution domain. Particle swarm optimization is a stochastic optimization technique derived from swarm intelligence, compared with the genetic algorithm [26].

Suppose that the search space of the problem to be solved is D dimension and the total number of particles is n . Then, the position of the i -th particle can be expressed as $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{iD})$, where D is D -dimensional space. The best, most suitable and advantageous position of the i -th particle in the history of “flight” (that is, the solution optimal corresponding to the position) is expressed as $\mathbf{P}_i = (\mathbf{P}_{i1}, \mathbf{P}_{i2}, \dots, \mathbf{P}_{iD})$; here, it is assumed that \mathbf{P}_g is the optimal position found in the search for “flight” for all particles of \mathbf{P}_i ($i = 1, \dots, n$), that is, the optimal solution. The position change rate of the i -th particle (that is, the speed at which the i -th individual bird flies in search of food) is vector $\mathbf{V}_i = (\mathbf{V}_{i1}, \mathbf{V}_{i2}, \dots, \mathbf{V}_{iD})$. Through the analysis and summary, the position of each particle (that is, where each individual bird is in the food-seeking space) is changed according to the following formula (“flight”) [26]:

$$\mathbf{V}_{id}^{k+1} = \omega \mathbf{V}_{id}^k + c_1 r_1 (\mathbf{P}_{id}^k - \mathbf{X}_{id}^k) + c_2 r_2 (\mathbf{P}_{gd}^k - \mathbf{X}_{gd}^k), \quad (1)$$

$$\mathbf{X}_{id}^{k+1} = \mathbf{X}_{id}^k + \mathbf{V}_{id}^{k+1}, \quad (2)$$

$$1 \leq i \leq n, 1 \leq d \leq D,$$

where c_1 and c_2 are nonnegative constants, called acceleration factors and also known as a learning factors. r_1 and r_2 are random numbers within $[0,1]$, providing infinite possibilities for finding the optimal result. k is the current iteration number. ω is called the inertia factor. The range of

position variation of dimension d is $[-X_{Maxd}, X_{Maxd}]$, which is the range of the solution space. The velocity range of each particle in the PSO algorithm is $[-V_{Maxd}, V_{Maxd}]$, which is the range of the position change of each bird in the bird flock. If the position and velocity of the particle exceed the boundary range in the iterative process, then the boundary value is taken. The basic algorithm steps of PSO are as follows [27]:

- (1) Initialize the particle swarm according to the formula, randomly set the initial position \mathbf{X}_i and initial velocity \mathbf{V}_i of each particle, and provide the particle search algorithm with the possibility of searching for the best results.
- (2) Calculate the fitness value of each particle based on the basic principle of the algorithm.
- (3) On this basis, update the best position \mathbf{P}_i of individual particles. The update criterion is for each particle comparing its fitness value with the fitness value of the best position \mathbf{P}_i it has experienced and selecting the best one as the current fitness value.
- (4) On this basis, update the best position \mathbf{P}_g of individual particles. For each particle, compare its current fitness value with the fitness value corresponding to the global best position. If the current fitness value is higher, update the best position of the particle swarm with the current position.
- (5) Adjust the velocity \mathbf{V}_i and position \mathbf{X}_i of the particle according to (1) and (2).
- (6) Finally, if the end conditions are met, the search process will be terminated; if the end condition is not met, go to step (2) and continue the algorithm until the condition is met. Output the search results; the whole algorithm proposed to solve the problem process is completed.

2.2. Non-Gaussian PSO. Although the traditional PSO is a global optimization search algorithm, it performs optimal search on all solutions in the iterative process, which will lead to premature convergence, and cannot perform global search well, falling into local solution convergence [11]. In order to overcome this shortcoming of the traditional PSO algorithm, this paper uses the corresponding content of the optimal design of the rotor blades of the wind turbine to improve the adaptability of the particle swarm optimization algorithm: combining non-Gaussian random distribution with PSO. Non-Gaussian random distribution can improve search efficiency. The non-Gaussian random distribution here adopts Lévy distribution, because Lévy distribution can realize the multistep hopping of particles in the search space to achieve the purpose of improving search efficiency. A detailed introduction to Lévy's non-Gaussian random distribution can be found in [28].

The Lévy distribution was proposed by the French mathematician Lévy in the 1930s [29]. Lévy flight is a random search process that follows the Lévy distribution

and is a non-Gaussian random process. This is a short-distance local search and occasionally long-distance global search, whose step size and time follow Lévy distribution. Lévy flight is roughly similar to the random foraging walks of many animals in nature. Some scholars have proved that the foraging trajectories of some animals and insects in nature conform to the Lévy distribution. Because Lévy flight's local and global search capabilities are more prominent, it is more suitable for application in the field of optimization. For example, Lévy flight is applied for location update in the cuckoo algorithm. Broadly speaking, Lévy flight is a random walk with a step length obeying Lévy distribution, usually expressed by a simple power rate formula: $L(s) \sim |s|^{-1-\beta}$, where the index $0 < \beta \leq 2$. Mathematically speaking, a simple Lévy distribution can be defined as

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{3/2}}, & 0 < \mu < s < \infty, \\ 0, & \text{others.} \end{cases} \quad (3)$$

where $\mu > 0$ is the minimum step size and γ is the scale parameter. Lévy flight is a non-Gaussian random process, which is a random walk method extracted from Lévy's stable distribution, ensuring that the PSO algorithm can perform a global search more efficiently without falling into local stagnation.

In non-Gaussian PSO, the new state of each particle can be written as

$$X^{t+1} = X^t + \gamma \otimes Levy(\omega), \quad (4)$$

where γ is the step size, which is related to the scale of the solved problem; here, we take any number of particle diameters in the interval $(0, 2)$; that is, $\gamma = \text{Ran}(D)$; D is the particle diameter; $Levy(\omega)$ represents Lévy distribution.

Then, the n samples generated can be written as [28, 30]

$$n = \text{Ran}(D) \otimes Levy(\omega) \sim 0.01 \frac{\varepsilon}{|\eta|^{1/\lambda}} (x_j^t - gbest^t), \quad (5)$$

where ρ and η are derived from the normal distribution $\rho \sim N(0, \sigma_\rho^2)$, $\eta \sim N(0, \sigma_\eta^2)$. The equation is as follows:

$$\sigma_\rho = \left\{ \frac{\Gamma(1+\omega) \sin(\pi\omega/2)}{\Gamma[(1+\omega/2)] \omega 2^{(\lambda-1)/2}} \right\}^{1/\lambda}, \sigma_\eta = 1, \quad (6)$$

where Γ is the standard Gamma function. At this time, the n value obtained by (5) is taken to update the position of the particle, and the fitness value of the new position of the particle is estimated. If the calculated new particle fitness value is better than the original individual extreme value (abbreviated as $pbest$), then update the individual extreme value and set the test limit to 0; otherwise, increase the test limit by 1 to continue to estimate the value of the new particle position. Repeat the above steps until the new fitness value is better than the original individual extremum.

3. Optimization Objective and Constraints

3.1. Objective Function. In order to find a global optimal solution using non-Gaussian PSO, the annual average generating capacity of the wind turbine is adopted as the objective function of the optimal design. Here, we take the Weibull distribution density function to fit the wind conditions. By solving the wind power utilization coefficient C_p of the wind turbine at different wind speeds, the average power of the annual power generation is obtained by integration, and then the average annual power generation of the wind turbine is obtained. The calculation formula is as follows:

$$P = F(b, \varphi) = \int_{V_1}^{V_2} \frac{1}{2} \rho v^3 \pi R^2 C_p(v) W(v) dv, \quad (7)$$

where P is the annual average power, V_1 is the cut-in wind speed, V_2 is the cut-out wind speed, v is the value of the inflow wind speed at infinity, and $W(v)$ is the Weibull distribution density function.

3.2. Optimized Design Variables and Constraints. In the optimization design of the aerodynamic shape of the wind turbine blade, the main optimization objectives are the airfoil section, chord length b , and twist angle φ . In this paper, the selected airfoil series will be designed without constant attack angle [31]. For the same type of airfoil, within a certain range of attack angle α at a certain Reynolds number, the lift coefficient and drag coefficient are expressed in the form of explicit attack angle as

$$\begin{aligned} C_L &= X_1(\alpha), \\ C_D &= X_2(\alpha), \end{aligned} \quad (8)$$

where $X_1(\alpha)$ and $X_2(\alpha)$ are Newton interpolation polynomials of the attack angle α , respectively; then, the relationship of angle of attack is as follows:

$$X(\alpha, V_\infty, \omega, R, b, \varphi) = 0. \quad (9)$$

Then, the expression of the power coefficient without explicit attack angle can be written as

$$X_p = X_p(V_\infty, \omega, R, b, \varphi), \quad (10)$$

where V_∞ is the incoming flow speed, ω is the rotor speed, and R is the rotor radius, all of which are known quantities. In this way, the optimal design variables can be determined as the blade chord length b and twist angle φ .

The chord length b and twist angle φ are distributed according to the Bezier curve, and the constraints can be written as

$$\left. \begin{aligned} b_{\min} &\leq b_r \leq b_{\max} \\ b_{\min} &\leq b_t \leq b_{\max} \\ r_{\min} &\leq r_{1c} \leq r_{2c} \leq r_{\max} \\ \varphi_{\min} &\leq \varphi_r \leq \varphi_{\max} \\ \varphi_{\min} &\leq \varphi_t \leq \varphi_{\max} \\ \varphi_{\min} &\leq \varphi_{1c} \leq \varphi_{2c} \leq \varphi_{\max} \end{aligned} \right\}, \quad (11)$$

where b_r and φ_r are the chord length and torsion angle at the root of the blade; b_t and φ_t are the chord length and twist angle at the blade tip; and r_{\min} , r_{\max} , φ_{1c} , and φ_{2c} are the coordinates of Bezier's control points.

Here, we apply the theory of blade element momentum (BEM) to establish the aerodynamic model of wind turbines. In the process of building an aerodynamic model, it is necessary to consider the effect of blade tip loss. Due to space limitations, no specific formula is given here, and the specific formula can be found in [32].

3.3. Optimized Design Procedures and Processes. Combined with non-Gaussian distributed particle swarm optimization algorithm, the basic flow of blade aerodynamic optimization program is presented in Figure 1. The optimization search of non-Gaussian PSO is different from the traditional PSO. Before the speed update of the non-Gaussian PSO, the limit test is performed on each particle. If the current particle does not exceed the limit, the speed is updated normally, and its position is also updated. If the particle velocity exceeds the limit, then set the position beyond the boundary of the search space to make its value equal to the boundary value. Therefore, the key steps of the non-Gaussian PSO blade aerodynamic optimization program proposed in this paper are as follows:

- (1) Set a fitness (limit) value for each particle, and the particle group exceeding this limit will be redistributed in the search space using a non-Gaussian random distribution. In non-Gaussian PSO, the initial particle swarms are randomly distributed in the search space, and the fitness values of these particle swarms need to be calculated. In addition, these fitness values are used to calculate the individual extreme value ($pbest$) of the particle swarm, that is, the current optimal solution found by a single particle and the global extreme value ($gbest$), which is the optimal solution currently found by the entire population.
- (2) Calculate the fitness value of the newly determined particles. If the calculated new fitness value of the particles is better than the original $pbest$, then they are used as $pbest$ and as a limit; if the new fitness value is not improved compared with the original $pbest$, then the particle limit is increased by 1 on the basis of the original value. Similarly, if the calculated new fitness value is smaller than $gbest$, then it is used as the particle's $gbest$ value. Perform the same steps until the end of the iteration.
- (3) The choice of limit value has a significant influence on the calculation result and calculation efficiency. Therefore, when selecting the limit value, the optimization of the calculation result and the calculation time should be considered.

4. Results and Analysis

The non-Gaussian PSO is used to optimize the design of wind turbine blades. The parameters of the rotor are shown in Table 1. The airfoil of the rotor adopts NACA6412, and the

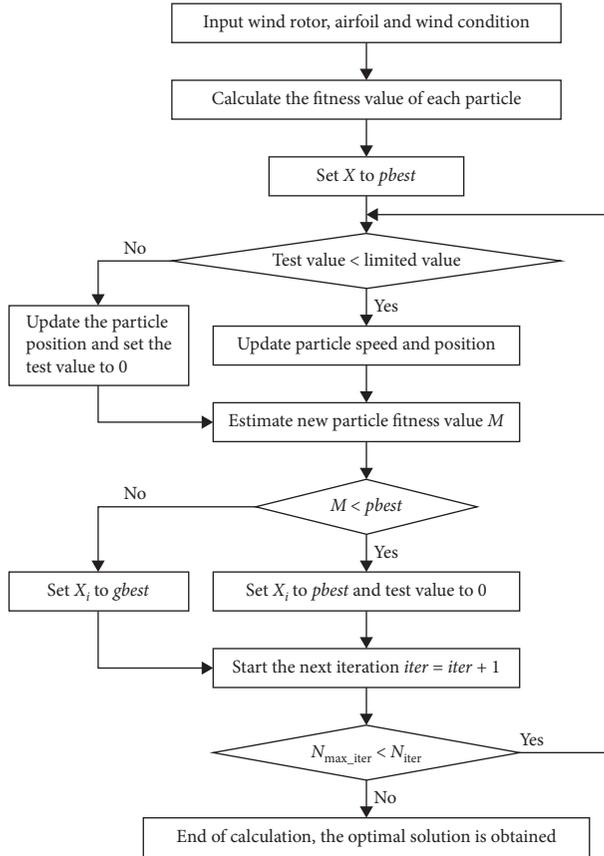


FIGURE 1: Flowchart of optimization design.

TABLE 1: Design parameters.

Parameter	Parameter value
Rotor diameter (m)	60
Cone angle (°)	0
Rated rotor speed (r/min)	19
Rated wind speed (m/s)	15
Cut-in wind speed (m/s)	4
Tip-speed ratio λ	6
Rated power (MW)	1.3
Reference height (m)	60
Hub elevation (m)	50
Cut-out wind speed (m/s)	25

blade is divided into 5 sections. Each section is taken at a distance of $0.2R$ (R is the radius of the rotor), and the chord length and twist angle of each section are calculated and compared with the original blades.

The basic parameter selection of the optimization model is as follows: population size is 40, learning factor $C_1 = C_2 = 2$, and inertia weight $w = (N_{\max_iter} - N_{iter}) / N_{\max_iter}$, where N_{\max_iter} refers to the maximum number of iterations and N_{iter} refers to the current number of iterations. The shape parameter of Weibull is 1.85, and all particles are initialized within the initial range.

In the non-Gaussian PSO algorithm, the selection of the limit M has a significant impact on the optimization design results. Therefore, the optimization design results of the blade chord length under different limits are compared here,

and compared with the original blade results and the traditional PSO results, which is shown in Figure 2.

It can be seen from Figure 2 that the non-Gaussian PSO and traditional PSO methods have optimized the chord length of the blades to be less than the original chord length of wind turbine blade, but the chord length of non-Gaussian PSO blade is smaller than that of the traditional PSO, indicating that the blades optimized by the non-Gaussian PSO algorithm have better aerodynamic characteristics than the traditional PSO. In particular, the reduction of the blade chord length is a good control of the output power and cost of the wind rotor for the concentration area of the blade mass (blade root), which proves the advanced nature of non-Gaussian PSO.

In the non-Gaussian PSO algorithm, for the limit M , it can be seen that the optimization effect is better with the increase of the limit. This is because the non-Gaussian random Lévy distribution can realize the multistep jumping of particles in the search space. The expansion of the scope makes the result more accurate. However, the calculation time is also increasing. Here, the calculation time at $M = 15$ is 1.2 times the calculation time at $M = 10$, and the calculation time at $M = 10$ is 0.7 times the calculation time at $M = 5$. Therefore, all things considered, it is reasonable to choose the limit $M = 10$ here, which can achieve a balance in improving the optimization result and the calculation cost. Therefore, in the following optimization calculations, the calculation is performed according to the limit $M = 10$.

Figure 3 shows a comparison between the blade twist angle and the original blade after using non-Gaussian PSO and traditional PSO optimization. It can be seen that the non-Gaussian PSO and traditional PSO optimized blade twist angle does not change much compared with the original wind turbine. This is because although the reduction of the twist angle can increase the output power of the wind rotor, it will also increase the axial thrust, thereby shortening the life of the blade and increasing the cost. Therefore, the change of the twist angle of the blade after using different optimization algorithms does not change much.

Figure 4 shows a comparison between the output power of the wind turbine blade and the original blade after the non-Gaussian PSO and the traditional PSO are optimized. It can be seen that after using non-Gaussian PSO optimization, the output power of the wind turbine blades is different from that of the original wind turbine blades at different wind speeds, and the output power of the blades optimized by the traditional PSO has increased, indicating that non-Gaussian PSO can better optimize the design of wind turbine blades than traditional PSO. Moreover, after non-Gaussian PSO optimization, the output power of the blade has been controlled within the rated power of 1.3MW, indicating that the non-Gaussian PSO optimized wind turbine has excellent stall control characteristics.

Further comparison found that, after using traditional PSO optimization, compared with the original blade output power, the output power of the wind rotor at a rated wind speed of 15 m/s increased by 2.1% and the annual output power increased by 5.3%. After using non-Gaussian PSO optimization, compared with the original blade output power,

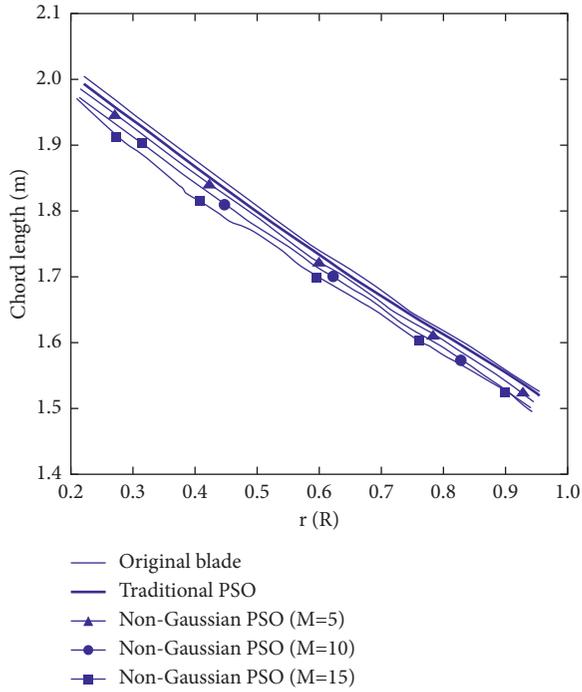


FIGURE 2: Comparison of blade chord.

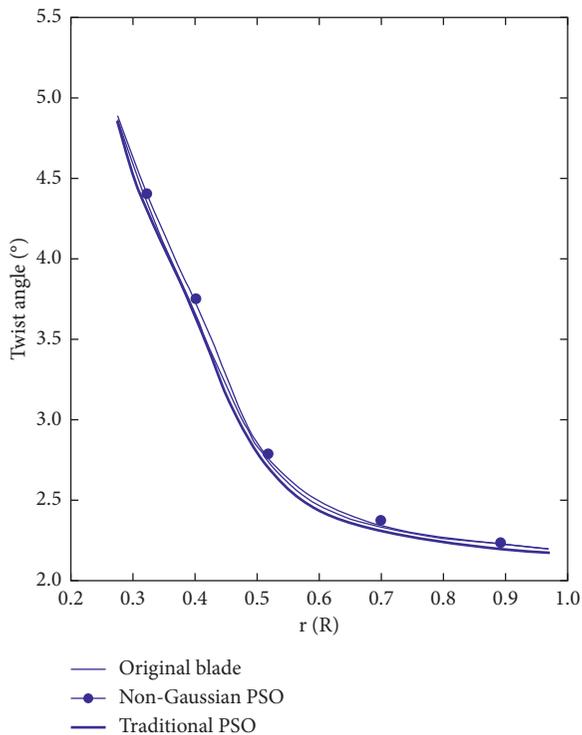


FIGURE 3: Comparison of blade twist angles.

at rated wind speed of 15 m/s, the output power of the wind turbine blade increased by 3.4%, and the annual output power increased by 7.6%. Therefore, from the perspective of computational efficiency, the calculation time of non-Gaussian PSO is about 65% of that of traditional PSO. The calculation efficiency of non-Gaussian PSO is greatly improved. This is because the non-Gaussian Lévy distribution realizes the

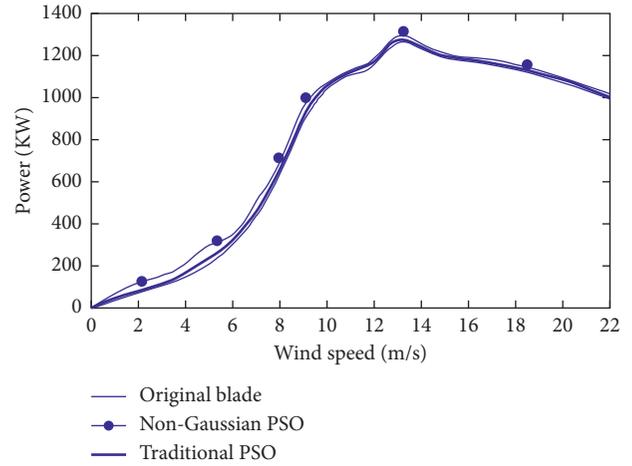


FIGURE 4: Comparison of output power before and after optimization.

TABLE 2: Comparison of computation residuals by different methods.

Residual	Time step			
	10	30	40	70
Traditional PSO	10^1	10^{-0}	10^{-1}	10^{-1}
Non-Gaussian PSO	10^{-1}	10^{-3}	10^{-5}	10^{-6}

multistep jumping of particles in the search space, which improves the search efficiency and saves the calculation time. Therefore, it can be seen that non-Gaussian PSO is a more superior and more efficient method for the optimal design of wind turbine blades than traditional PSO.

In order to illustrate the computational efficiency of non-Gaussian PSO, it is compared with the calculation convergence of the traditional PSO for optimal design. Table 2 shows the residuals calculated by different methods.

It can be seen from Table 2 that as the time step increases, the calculated residual error of the traditional PSO does not change significantly, and when the time step reaches a certain value, the calculated residual error no longer changes, indicating that the traditional PSO method has poor convergence and local convergence. Non-Gaussian PSO reduces the calculation residual rapidly with the increase of time step, and the decrease is large, indicating that the non-Gaussian PSO overcomes the disadvantages of the traditional PSO method, such as poor convergence and easiness to fall into local solution convergence, and has the characteristics of good convergence. The reason is that the p_{best} value of the particles in the non-Gaussian PSO is arbitrarily selected and, unlike the traditional PSO, it needs to be learned from the p_{best} and g_{best} values of the particles, so the non-Gaussian PSO ensures the diversity of the population and avoids the solution falling into the local minimum situation.

To illustrate the computational stability of non-Gaussian PSO, at different time steps, the errors of traditional PSO and non-Gaussian PSO were statistically analyzed, and the average value and mean-square deviation of the calculation errors of different optimization algorithms were obtained. The optimal

TABLE 3: Statistical analysis of errors by different methods.

Time step	Traditional PSO		Non-Gaussian PSO	
	Average value	Mean-square deviation	Average value	Mean-square deviation
20	$8.67e-1$	$2.5e-1$	$4.35e-3$	$3.43e-3$
40	$6.45e-1$	$3.23e-1$	$5.34e-5$	$3.76e-5$
50	$6.11e-1$	$8.23e-2$	$1.23e-5$	$6.45e-6$
60	$3.25e-1$	$5.23e-2$	$5.45e-6$	$9.45e-7$
70	$2.23e-1$	$6.34e-2$	$4.76e-7$	$1.23e-7$

TABLE 4: Comparison of the error norm using different algorithms.

Wind velocity (m/s)	Iterations	PSO	Proposed algorithm
5	25	$0.34e-3$	$2.42e-4$
	40	$5.17e-4$	$3.96e-5$
10	25	$2.51e-4$	$4.92e-5$
	40	$3.46e-4$	$7.74e-5$
15	25	$1.83e-5$	$0.47e-6$
	40	$2.72e-5$	$1.85e-6$
25	25	$3.16e-5$	$0.76e-7$
	40	$4.64e-6$	$2.75e-8$

values calculated here are all taken as 0, and when the results are less than 10^{-6} , the calculation results are considered to converge. The statistical results are shown in Table 3.

From Table 3, it can be seen that when using different time steps, the average value and mean-square deviation of the errors obtained by non-Gaussian PSO are less than those of traditional PSO, indicating that the optimization results of non-Gaussian PSO are better than those of traditional PSO and more stable results can be obtained. This is because the non-Gaussian PSO performs a limit check on each particle before performing speed update, which avoids searching in an inefficient space, thereby greatly improving computational efficiency and convergence.

In addition, in order to illustrate complexity of the proposed algorithm, comparison of the error norm between the proposed algorithm presented in this paper and the traditional PSO in computing x -direction displacement at different wind speed is provided in Table 4.

It can be seen from Table 4 that for the same number of iterations, the error norm of the proposed algorithm is much smaller than that of traditional PSO. Moreover, the proposed algorithm in this paper can achieve smaller error using fewer iterations, which suggests that the convergence speed of the method in this paper is faster and the error is smaller. However, in numerical calculation, we should pay more attention to the balance between calculation accuracy and calculation time. It is found in the calculation that when the number of iterations increases by about 13% on average, the calculation accuracy increases by about 25% and the calculation time increases by about 10% on average.

5. Conclusions

This paper proposes a particle swarm optimization algorithm combined with non-Gaussian random distribution to optimize the design of wind turbine blades. The specific conclusions are as follows:

- (1) The non-Gaussian PSO algorithm is presented and applied to optimize calculation and analysis of a 1.3 MW wind turbine blade. The results show that, compared with the traditional PSO, the non-Gaussian PSO expands the search range of particles and the optimization result increases with the increase of the search limit, making the optimization result more accurate.
- (2) Non-Gaussian PSO further improves the output power of wind turbine blades: After non-Gaussian PSO optimized blades, at rated wind speed of 15 m/s, the output power of wind turbine blades increased by 3.4%, and the annual output power increased by 7.6%. Furthermore, the calculation time of non-Gaussian PSO is about 65% of the time of traditional PSO. The reason is that the non-Gaussian Lévy distribution in the non-Gaussian PSO realizes the multistep jump of the particles in the search space, which improves the search efficiency and saves computational time.
- (3) In terms of computational efficiency and computational stability, compared with the traditional PSO, as the time step increases, the calculation residual of non-Gaussian PSO decreases rapidly, indicating that non-Gaussian PSO overcomes the shortcomings of traditional PSO, such as poor convergence and easiness to fall into local solution convergence, and non-Gaussian PSO can obtain more stable results.
- (4) Although the proposed non-Gaussian PSO is superior in many computation aspects, there are still some limitations and further improvements are needed. Since the non-Gaussian PSO performs a limit check on each particle before performing speed update, it brings some random process and takes more computation time before speed update. How to reduce the random check process and how to improve the limit check efficiency are the issues that need to be further discussed. In addition, in future work, thickness distribution and controlling surface airfoil location distribution will be considered in the optimization design to obtain better optimization results.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 52178468), Guangxi Key Laboratory of New Energy and Building Energy Saving Foundation (no. Gui Keneng 19-J-21-14), Open Foundation of Guangxi Key Laboratory of Embedded Technology and Intelligence (no. 2019-02-08), Joint Cultivation Program of National Natural Science Foundation of Guangxi (no. 2019GXNSFAA245037), Guangxi Youth Innovative Talents Research Project (no. Guike AD19245012), Guangxi Key Laboratory of Geomechanics and Geotechnical Engineering (no. GUIKENENG19-Y-21-2), and Scientific and Technology Startup Foundation of Guilin University of Technology (nos. GUTQGJJ2019042 and GUTQDJJ2019041).

References

- [1] E. Bao, X. R. Shao, and D. Y. Liu, "New methods for designing wind turbine blades," *Journal of Machine Design*, vol. 22, no. 2, pp. 24–26, 2005.
- [2] Z. H. Shen and G. L. Yu, "Influence of airfoil's camber on the performance of wind turbines," *Journal of Chinese Society of Power Engineering*, vol. 27, no. 1, pp. 136–139, 2007.
- [3] N. Nouaouria and M. Boukadoum, "Particle swarm classification for high dimensional data sets," in *Proceedings of the 22th International IEEE Conference on Tools with Artificial Intelligence*, pp. 87–93, Arras, France, 2010.
- [4] M. L. Peng, Z. C. Yang, Y. Y. Cao et al., "Turbine blade parameterization modelling based on Bezier curve and particle swarm optimization algorithm," *Proceedings of the CSEE*, vol. 32, no. 32, pp. 101–108, 2012.
- [5] X. Liu, Y. Chen, and Z. Q. Ye, "Research on HAWT Rotor blades optimization," *Journal of Shantou University (Natural Science Edition)*, vol. 21, no. 1, pp. 44–49, 2006.
- [6] C. C. Liao, X. L. Zhao, and J. Z. Xu, "Blade layers optimization of wind turbines using FAST and improved PSO Algorithm," *Renewable Energy*, vol. 42, pp. 227–233, 2012.
- [7] C. Liao, K. Shi, and X. Zhao, "Predicting the extreme loads in power production of large wind turbines using an improved PSO algorithm," *Applied Sciences*, vol. 9, no. 3, p. 521, 2019.
- [8] A. Ceruti, "Meta-heuristic multidisciplinary design optimization of wind turbine blades obtained from circular pipes," *Engineering with Computers*, vol. 35, no. 2, pp. 363–379, 2019.
- [9] Y. Ma, A. Zhang, L. Yang, C. Hu, and Y. Bai, "Investigation on optimization design of offshore wind turbine blades based on particle swarm optimization," *Energies*, vol. 12, no. 10, pp. 1–18, 2019.
- [10] Z. Posteljnik, S. Stupar, J. Svorcan, O. Peković, and T. Ivanov, "Multi-objective design optimization strategies for small-scale vertical-axis wind turbines," *Structural and Multidisciplinary Optimization*, vol. 53, pp. 277–290, 2016.
- [11] W. G. Wang, Z. S. Liu, and H. M. Ni, "Improved particle swarm optimization algorithms solving optimization problems with mixed variables and constraints," *Journal of System Simulation*, vol. 24, no. 6, pp. 1175–1179, 2012.
- [12] H. Orouji, O. Haddad, E. Fallah-Mehdipour, and M. A. Mariño, "Extraction of decision alternatives in project management: application of hybrid PSO-SFLA," *Journal of Management in Engineering*, vol. 30, no. 1, pp. 50–59, 2014.
- [13] M. Kanemasa and E. Aiyoshi, "Algorithm tuners for PSO methods and genetic programming techniques for learning tuning rules," *IEEE Trans on Electrical and Electronic Engineering*, vol. 9, no. 4, pp. 407–411, 2014.
- [14] K. Muranaka and E. Aiyoshi, "Computational properties of hybrid methods with PSO and DE," *Electronics and Communications in Japan*, vol. 97, no. 4, pp. 58–66, 2014.
- [15] W. H. Lim and N. A. Mat Isa, "Particle swarm optimization with dual-level task allocation," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 88–110, 2015.
- [16] D. Tian, X. Zhao, and Z. Shi, "DMPSO: diversity-guided multi-mutation particle swarm optimizer," *IEEE Access*, vol. 7, pp. 124008–124025, 2019.
- [17] A. A. Karim, N. A. Mat Isa, and W. H. Lim, "Modified particle swarm optimization with effective guides," *IEEE Access*, vol. 8, pp. 188699–188725, 2020.
- [18] H. T. Rauf, U. Shoaib, M. I. Lali, M. Alhaisoni, M. N. Irfan, and M. A. Khan, "Particle swarm optimization with probability sequence for global optimization," *IEEE Access*, vol. 8, pp. 110535–110549, 2020.
- [19] Q. Zhao and C. Li, "Two-stage multi-swarm particle swarm optimizer for unconstrained and constrained global optimization," *IEEE Access*, vol. 8, pp. 124905–124927, 2020.
- [20] W. H. Lim, N. A. Mat Isa, S. T. Sew et al., "A self-adaptive topologically connected-based particle swarm optimization," *IEEE Access*, vol. 6, pp. 65347–65366, 2018.
- [21] M. J. Cao, *Optimal Control of Wind Farm Output Power Considering the Effects of Wakes*, Ph.D. Dissertation, Xi'an University of Technology, Xi'an, China, 2019.
- [22] G. Chen, J. Chen, Z. Y. Sun et al., "Wind turbine airfoil design based on static aeroelastic energy," *Acta Energetica Sinica*, vol. 41, no. 6, pp. 8–15, 2020.
- [23] F. Liu, Y. Sun, G. Wang, and T. Wu, "An artificial bee colony algorithm based on dynamic penalty and Lévy flight for constrained optimization problems," *Arabian Journal for Science and Engineering*, vol. 43, pp. 7189–7208, 2018.
- [24] Q. He, M. Huang, and X. Zhou, "Deep search cuckoo algorithm based on reverse learning strategy," *Journal of Guizhou University (Natural Science Edition)*, vol. 37, no. 2, pp. 32–36, 2020.
- [25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Western Australia, December 1995.
- [26] F. J. Sun, D. H. Zhu, M. R. Liang, and D. M. Zhang, "Study on form-finding of cable-membrane structures based on particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–13, Article ID 1281982, 2020.
- [27] H. E. Sadek, X. Zhang, M. Rashad, and C. Cheng, "Improvement of interior ballistic performance utilizing particle swarm optimization," *Mathematical Problems in Engineering*, vol. 4, pp. 1–10, 2014.
- [28] M. Ghaemi, Z. Zabihinpour, and Y. Asgari, "Computer simulation study of the Lévy flight process," *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 8, pp. 1509–1514, 2009.
- [29] B. Yan, Z. Zhao, Y. Zhou et al., "A particle swarm optimization algorithm with random learning mechanism and Lévy flight for optimization of atomic clusters," *Computer Physics Communications*, vol. 219, pp. 79–86, 2017.
- [30] J. Rada-Vilela, M. Zhang, and W. Seah, "A performance study on synchronicity and neighborhood size in particle swarm optimization," *Soft Computing*, vol. 17, no. 6, pp. 1019–1030, 2013.
- [31] Z. F. Liu, Z. K. Zhou, and Z. C. Deng, "Optimization design on 20 KW constant-pitch wind turbine blade," *Transactions of*

the Chinese Society for Agricultural Machinery, vol. 5,
pp. 121–124, 2011.

- [32] D. X. He, *Wind Engineering and Industrial Aerodynamics*,
National defense industry press, Beijing, China, 2006.