Energy 313 (2024) 134050

Contents lists available at ScienceDirect

Energy

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

Reinforcement learning-based particle swarm optimization for wind farm layout problems

Zihang Zhang [a], Jiayi Li [a], Zhenyu Lei [a], Qianyu Zhu [a], Jiujun Cheng [b], Shangce Gao [a,*]

[a] Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan
[b] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, 200092, China

ARTICLE INFO

Keywords:
Wind farm layout optimization
Proximal policy optimization
Particle swarm optimizer
Wake effect

ABSTRACT

Optimizing wind farm layouts is critical to maximizing wind power generation. The wake effect significantly impacts turbines located downwind, making farm layout a key determinant of power generation efficiency. Traditional algorithms often overlook the value of leveraging historical information, which can lead to entrapment in local optima. Our survey reveals that previous studies on wind farm layout optimization (WFLO) have not adequately integrated the historical data of particle swarm optimization (PSO) with reinforcement learning's empirical pool, resulting in the loss of valuable information. Here, we present a novel approach that enhances algorithm development and exploration by utilizing historical data and integrating proximal policy optimization from reinforcement learning with an experience pool. This method markedly outperforms the conventional genetic PSO in terms of performance. Extensive numerical experiments across wind farms of various sizes and four distinct wind scenarios demonstrate the superior efficacy of our reinforcement learning-based particle swarm optimization (RPSO) algorithm compared to 12 state-of-the-art methods. Under four wind scenarios, the average power conversion efficiencies of RPSO for the three turbine scales reach 98.68%, 98.14%, and 97.33%, respectively, underscoring the high competitiveness of the proposed RPSO for WFLO in diverse wind conditions.

1. Introduction

To address the global warming crisis, the development and utilization of renewable energy are imperative. Among these, wind energy stands out as a pivotal alternative to fossil fuels due to its affordability, widespread availability, and environmental sustainability [1–3]. In 2022, global wind power generation witnessed a remarkable increase of 17%, solidifying its position as a key energy source for the future as recognized by an expanding array of nations.

The efficiency of wind turbine power generation hinges on various factors. Previous studies have predominantly focused on maximizing power generation efficiency through wind speed prediction, optimization of wind turbine models, modeling wake effects, and optimizing wind turbine layouts. The downstream wake effect phenomenon significantly curtails the output power of downstream turbines due to interference from upstream turbines. Optimizing turbine layout offers a viable strategy to mitigate the impact of wake effects and thereby maximize array power generation efficiency, rendering it a pivotal area of investigation. Despite the widespread adoption of advanced intelligent algorithms for layout optimization, traditional approaches still grapple with suboptimal solutions, susceptibility to local optima, and instability [4–7].

To address these challenges, this work proposes a Genetic Particle Swarm Optimization (PSO) based on Proximal Policy Optimization (PPO) reinforcement learning (RL) algorithm [8,9]. PPO is a new type of Policy Gradient algorithm. Traditional Policy Gradient algorithms rely heavily on the selection of the step size, but it is also difficult to determine the appropriate step size, which leads to too large a difference between the old and new policies during training, and is not conducive to learning [10]. PPO solves the problem of step size selection by realizing small batch updating in multiple training steps. In the Genetic PSO algorithm, r is a parameter which controls the proportion of the current optimal individual in the next generation of heredity. The selection of optimal particles and the genetic ratio can greatly affect the efficiency of the PSO algorithm, and in previous studies, this ratio r is chosen randomly and sum to 1, or at a constant value from past studies, which may not be an appropriate approach for different problems [11]. Therefore, we introduce the PPO algorithm to utilize the empirical pool data to help the PSO algorithm determine a better r for higher power generation efficiency.

This work compares the proposed algorithm with other state-of-the-art (SOTA) algorithms applied to wind turbine layout optimization

* Corresponding author.
E-mail address: gaosc@eng.u-toyama.ac.jp (S. Gao).

https://doi.org/10.1016/j.energy.2024.134050
Received 19 April 2024; Received in revised form 15 November 2024; Accepted 30 November 2024
Available online 5 December 2024

under the same experimental conditions. The experimental results show that the proposed algorithm outperforms the other algorithms for three sizes of wind farms (30, 35, and 40 turbines), four wind scenarios. Also, the proposed algorithm is competitive in the ablation experiments, which demonstrate the significant improvement of RL on the original algorithm.

The main highlights and contributions of this work can be summarized as follows:

(1) In this work, the proposed method of combining RL with traditional intelligent algorithms is applied to wind turbine layout optimization. Extensive numerical experiments show that the proposed method outperforms all the twelve SOTA models compared in this problem.
(2) RL utilizes historical search information to break out of otherwise constrained conditions and help guide the PSO to converge in a better direction.
(3) Instead of the previous approach of random selection in PSO, we employ the PPO strategy. Furthermore, we introduce two independent parameters to replace the original constraint of their summation equaling 1. Experimental results unequivocally demonstrate that this novel methodology significantly enhances solution quality.

The rest of the paper is structured as follows: Section 2 summarizes the work related to wind turbines. Section 3 presents the numerical model for the wind turbine layout optimization problem. Section 4 describes the proposed algorithm in detail. Experimental results are presented in Section 5. Section 6 summarizes the work in this paper.

2. Related work

To maximize wind turbine output efficiency, researchers typically examine various factors including wind farm siting, wind speed prediction, wind turbine design, wake flow modeling, and wind farm layout optimization (WFLO). As energy technology continues to advance, optimization of wind turbine design and wake modeling has become increasingly efficient and stable. Consequently, WFLO has emerged as a crucial aspect in enhancing the power generation capacity of wind farms [12,13]. WFLO involves optimizing the positioning of each wind turbine within the array to minimize wake effects between turbines and maximize overall power generation capacity. As the number of turbines increases, optimizing wake effects becomes increasingly challenging [14].

In the past, extensive efforts have been dedicated to improving WFLO through the optimization of intelligent algorithms. Generally, WFLO problems can be approached through two main methods: discrete modeling and continuous modeling. Discrete modeling involves placing wind turbines at the centers of pre-defined grids of equal size. The finer the division of the wind field, the more intricate the WFLO problem that can be addressed. Conversely, continuous modeling permits turbines to be placed anywhere within the grid, not necessarily at the center. However, this approach typically entails higher computational complexity and is highly sensitive to initialization. In [15], Mosetti et al. used genetic algorithms (GA) for the first time in the Jensen wake model to solve the discrete WFLO problem. They divide the wind farm into equal-sized grids of a certain size, and each grid is used to place a turbine. Subsequently, Marmidis et al. [16] further divided the wind farm gridding, subdivided into 100 squares with higher accuracy, and solved the WFLO by Monte Carlo simulation. Cheng et al. [17] used three optimization algorithms, GA, PSO and simulated annealing, and a back-propagation neural network-based machine learning approach to optimize a novel U-type Darrieus Wind Turbine (UDWT). The results show that the UDWT can improve aerodynamic and structural performance compared with conventional two- and four-blade wind turbines. Recently, Neshat et al. [18] proposed a novel, robust and efficient multi-objective swarm optimization approach (DMOGWA) to achieve an optimal compromise between maximizing the power output of a wave energy converter and minimizing the impact on the nacelle acceleration of a wind turbine. This approach is applied to hybrid offshore renewable energy platforms to optimize power production and reduce the levelised cost of energy by integrating or sharing several renewable energy technologies. Experimental results show that DMOGWA has advantages over many well-known multi-objective swarm intelligence methods and mainstream evolutionary multi-objective algorithms. In GA, Grady et al. [19] improved the GA algorithm by proposing a method to improve the number of individual iteration steps to optimize the discrete WFLO problem. In [20], Emami and Noghreh propose a new objective function that better balances the cost, power, and efficiency of wind farms, and then encode the GA algorithm in a new way, so that the algorithm no longer relies on subpopulations, and can be solved with a single genetic algorithm. Chen et al. [21] used a nested genetic algorithm to optimize the layout of wind turbines with different hub heights and further demonstrated the effectiveness of the proposed approach on a large commercial wind farm. In 2015, Chen et al. [21] further used a multi-objective genetic-based algorithm to optimize wind farm layout, again experimenting under real commercial wind farms. Ju et al. [22] proposed Adaptive Genetic Algorithm (AGA) and Self-Inspired Genetic Algorithm (SIGA) to address the limitations of traditional GA. The worst turbines were randomly relocated using AGA, and then SIGA was used to re-direct the localization. Subsequently, Ju et al. [23] proposed a support vector regression-based bootstrap genetic algorithm to optimize the layout of wind farms by considering landowner participation. For the first time, Wang et al. [24] proposed a differential evolution algorithm with a new coding mechanism by treating the position of each turbine as an individual. The whole population then represents a layout, reducing the dimension of the search space and eliminating the population size parameter. Yu et al. [25] proposed a differential evolutionary family of algorithms with chaotic local search-based as well as enhanced memory storage mechanisms for WFLO.

Not limited to EAs, other types of intelligent algorithms have also demonstrated efficacy in addressing the WFLO problem. Feng et al. [26] introduced an enhanced randomized search (RS) algorithm employing a continuous formulation. Experimental results highlighted the algorithm's performance, which was attributed to both GAs and previous RS algorithms. In [27], the authors presented an enhanced ACO algorithm tailored for the WFLO problem, specifically addressing wake effects. Their work yielded notable progress in this domain. Bai et al. [28] further improved the utilization of the adaptive genetic algorithm proposed by the previous authors by transforming the relocation of multiple wind turbines into a single-player reinforcement learning problem, which was further solved by Monte Carlo tree search embedded in an evolutionary algorithm. Experimental results show that the proposed method brings significant improvements over previous methods. Hou et al. [29] employed the PSO algorithm to optimize a mathematical model they proposed, which incorporates variations in wind direction and wake losses. Their objective was to maximize energy output while minimizing total production costs. Pookpunt and Ongsakul [30] introduced a binary PSO method with time-varying acceleration coefficients. They demonstrated the effectiveness of their approach in addressing the WFLO problem through experimental comparisons with numerous variants. Tao et al. [31] utilized a hybrid Gray Wolf Optimization algorithm to address the WFLO problem, integrating multiple wind turbine placement strategies. They achieved promising results, particularly when employing the Frandsen–Gaussian wake model. An adaptive genetic learning particle swarm optimization method for optimizing the WFLO problem was proposed by Lei et al. [32]. The strategy adaptively adjusts the position of the worst turbine to improve the conversion efficiency of the wind farm. Later, the same authors improved the PSO algorithm by a chaotic local search mechanism for optimizing large-scale WFLO problems with good results [33].

Table 1
The nomenclature used in this article.

Symbol	Description
π	The ratio of the circumference
r	The diameter of rotors
R	The radius of influence of wake effect
v	The number of velocity
h	The height of wind turbine towers
S	The intersecting area of wake effect
d	The distance between two turbines
N	The number of turbines
Φ	The layout
η	The conversion efficiency
σ	The wind scenario
p_i	The location of turbines
M	The population size
L_s	The length of the rows and columns
L_w	The width of each grid
D	The dimensions of WFLO
$pbest$	Individual historical best position
$gbest$	Particle global optimal position
π_θ	The policy of PPO
ϵ	The hyperparameter for control strategy updating
μ	The entrainment constant

In recent years, more and more researchers have tried to use RL to optimize various single-objective and multi-objective problems. Dong and Zhao [34] propose an automatic grouping method for wind turbines based on RL, and their proposed automatic grouping strategy enables large-scale wind farms to be divided into subgroups that are favorable for RL training. Bai et al. [28] leveraged an adaptive genetic algorithm, integrating Monte Carlo Tree Search to iteratively relocate wind turbine layouts to more optimal positions, resulting in notable performance improvements. Yu and Lu [35] introduced a Q-Learning based multi-objective differential evolutionary algorithm to address the WFLO problem. By utilizing Q-Learning, the authors fine-tuned the parameters of the differential evolution algorithm, achieving a balance between local and global searches. Yu and Zhang [36] introduced a teaching optimization algorithm based on RL. Through adjustments to the structure of the original algorithm, performance was optimized. The enhanced algorithm surpassed other algorithms in performance across two simulated wind conditions and four wind farm scenarios. In [37], the authors demonstrated a significant improvement in power generation for wind farms by utilizing sparse datasets through a novel deep RL-based wind farm control scheme. Remarkably, this approach does not necessitate wake-up modeling. In [38], the authors employed deep RL to optimize the thrust coefficient and yaw angle, leading to a significant enhancement in the total power generation of a wind farm.

3. Wind farm layout numerical model

In this section, we begin by introducing the most crucial wake model utilized in wind farm optimization. Subsequently, we outline the objective function and evaluation methods employed in this study. The nomenclature used in this article is summarized in Table 1.

3.1. Wake model

In a wind farm, the presence of a wind turbine upwind significantly impacts the performance of turbines located downwind, a phenomenon known as the wake effect, illustrated in Fig. 1. To mitigate the adverse effects of wake turbulence on overall energy extraction, it is imperative to devise a more optimized layout to enhance energy capture efficiency. In our study, we adopt the Jensen model to assess wind speeds affected by wake, as it is a widely used semi-empirical wake model in engineering. Specifically, wind turbines 1 and 2 are positioned upwind, while wind turbine 3 is situated downwind. As a result of the wake effect, the wind speed experienced by wind turbine 3 is significantly diminished.

3.1.1. Jensen single wake model

The Jensen model operates under the assumption of momentum conservation, whereby the momentum deficit undergoes linear variation. Wind turbines separated by x can be expressed as:

$$\pi r^2 v_1' + \pi(R^2 - r^2)v_1 = \pi R^2 v_3, \tag{1}$$

$$R = r + \mu x, \tag{2}$$

μ is the entrainment constant:

$$\alpha = \frac{0.5}{\ln(\frac{h}{k_0})}, \tag{3}$$

where k_0 is the roughness of wind farm ground surface, h is the height of wind turbine towers. Meanwhile, this model gives the wind velocity just passing through the turbine v_1' as approximately equal to $\frac{1}{3}v_1$ ambient wind velocity according to classical theory. The downwind turbine velocity v_3, which is effect of a single wind turbine, can be expressed as [39]:

$$v_3 = v_1\left[1 - \frac{2}{3}\left(\frac{r}{r + \frac{0.5}{\ln(\frac{h}{k_0})}x}\right)^2\right]. \tag{4}$$

3.1.2. Jensen multiple wake model

In practical wind farm applications, a wind turbine is influenced by the wake effects generated by multiple upwind turbines, as depicted in Fig. 1. Consequently, a single wake effect model is inadequate. Instead, the Jensen multiple wake effect model is employed, with the formula expressed as follows [39]:

$$v = v_1\left[1 - \sqrt{\sum_i^N \left(1 - \frac{v_i}{v_1}\right)^2 \left(\frac{S_i}{S_r}\right)^2}\right], \tag{5}$$

where v is the combined wind speed of the turbine after it has been subjected to multiple wake effects, v_i is the wind speed affected by the wake effect of the ith upwind fan. Note that here we assume that all wind turbines are subjected to an initial velocity (v_1, v_2, etc.) that is equal and equal to the ambient wind speed. S_i is the area where the wake effect of the ith turbine intersects the turbine at the downwind location. S_r is the surface area of each turbine (assuming that all turbines in the wind farm have the same configuration). The intersecting areas of the two turbines can be roughly divided into the two scenarios of Fig. 2.

In all cases, the intersecting area S_i is equal:

$$S_i = S_{\widehat{ABD}} + S_{\widehat{CBD}} - 2S_{\triangle ABC}, \tag{6}$$

by the cosine theorem:

$$\alpha = \arccos\left(\frac{R^2 + d^2 - r^2}{2Rd}\right) \tag{7}$$

$$\beta = \arccos\left(\frac{r^2 + d^2 - R^2}{2Rd}\right), \tag{8}$$

substituting radians into the sector formula,

$$S_i = \left[\arccos\left(\frac{R^2 + d^2 - r^2}{2Rd}\right) + \arccos\left(\frac{r^2 + d^2 - R^2}{2Rd}\right)\right] \cdot R^2 - Rd\sin(\alpha). \tag{9}$$

Note that the above equation omits the conversion between radian and angle.

3.2. Objective function of WFLO

The goal of WFLO is to maximize the overall output of the wind farm while minimizing the adverse effects of wake turbulence on turbine

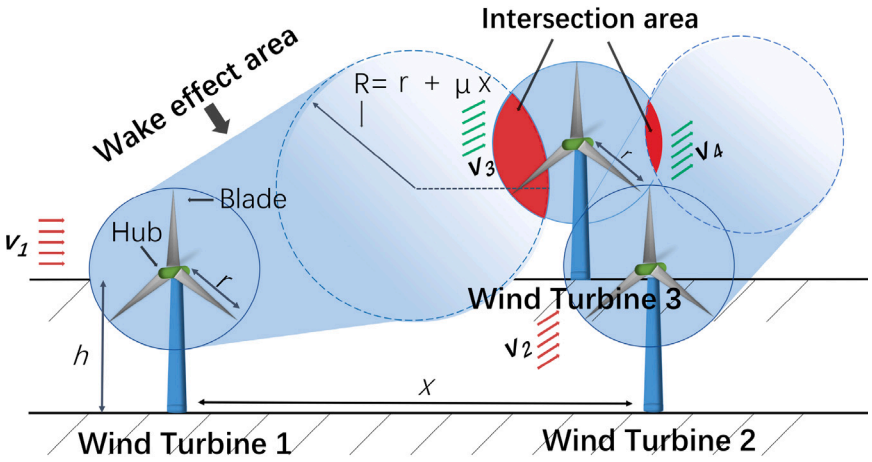

Fig. 1. The illustration of wake effect.

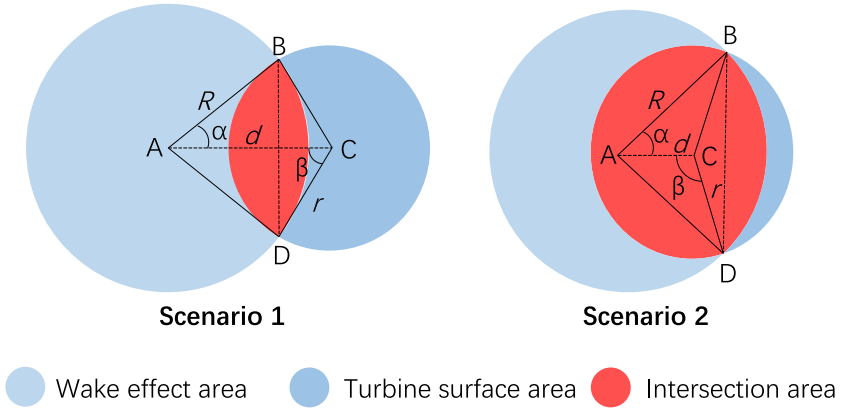

Fig. 2. The intersecting area of two turbines.

performance, all while minimizing construction costs. The objective function can be expressed as:

$$O = \min \frac{C(N)}{T(N,\sigma,\boldsymbol{\Phi})}, \tag{10}$$

where N is the number of turbines in the wind farm. In this study, N is a constant, T is the energy output of the wind field of N turbines under σ wind conditions, $\boldsymbol{\Phi}$ layout. σ is wind scenario that includes wind speed and direction, etc. $\boldsymbol{\Phi}$ is the wind farm layout. The construction cost uses the formula given in [15]:

$$C(N) = N\left(\frac{2}{3} + \frac{1}{3}e^{-0.00174N^2}\right). \tag{11}$$

Therefore, the objective function can be written to maximize the output of the wind farm [15]:

$$\begin{aligned} O &= \max T(N,\sigma,\boldsymbol{\Phi}) \\ &= \sum_{i=1}^{N}\sum_{v,\theta} p(v,\theta)T_i(v,\theta,\boldsymbol{\Phi}), \end{aligned} \tag{12}$$

where v and θ are wind speed and wind direction under wind scenario σ respectively. $p(v,\theta)$ is the probability of v and θ. In WFLO, it is common to use the conversion efficiency η to represent the performance of the whole wind farm. Therefore, we use the following equation to represent the performance of the current layout $\boldsymbol{\Phi}$:

$$\eta = \frac{T(N,\sigma,\boldsymbol{\Phi})}{N\sum_{v,\theta} p(v,\theta)T_r(v,\theta,\boldsymbol{\Phi})}, \tag{13}$$

where $T_r(v,\theta,\boldsymbol{\Phi})$ is the power rating at wind speed v, wind condition θ and layout $\boldsymbol{\Phi}$. η is closer to 1, indicating that the layout is more effective.

4. Proposed RPSO

The proposed reinforcement learning-based particle swarm optimization (RPSO) algorithm is presented in this section. The overall flow of RPSO is schematically shown in Fig. 3. Inspired by the use of historical information to optimize the PSO algorithm, we propose to use the PPO reinforcement learning approach to generate more reasonable and efficient *pbest* (individual historical optimal) and *gbest* (global optimal) values to influence the subsequent iterations.

4.1. Initialization

In this study, integer modeling will be employed to optimize the layout of wind turbines within a predefined wind farm. For instance, in a wind farm comprising 30 wind turbines, the dimension of the wind turbine layout optimization problem is 30. During initialization, a set of random integer locations for the wind turbines (ensuring mutual exclusivity) is generated within the boundaries of the wind farm. The positions and velocities of the particles are described as follows:

$$P_i = \left\{p_i^1, p_i^2, p_i^3, \ldots, p_i^D\right\}, i \in \{1,2,3,\ldots,M\} \tag{14}$$

$$V_i = \left\{v_i^1, v_i^2, v_i^3, \ldots, v_i^D\right\}, i \in \{1,2,3,\ldots,M\} \tag{15}$$

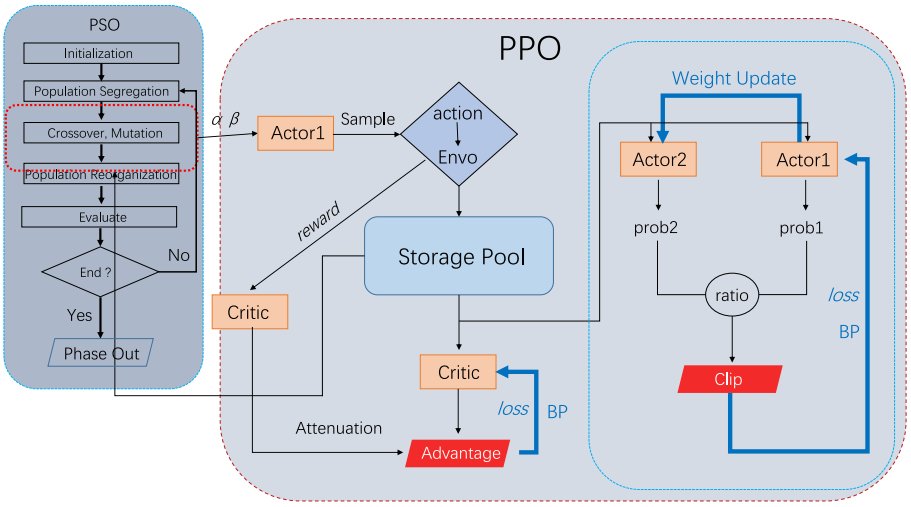

Fig. 3. The flowchart of RPSO.

where p_i^N denotes the position of the dth wind turbine in the wind farm. M is the population size and d is the number of turbines to be optimized. Subsequently, the x_i^d is converted into 2D coordinates for evaluating the generation output of the wind farm:

$$\begin{cases} p_{i,x}^d = \left(p_i^d - L_s \times \left\langle \frac{p_i^d - 1}{L_s} \right\rangle - 0.5 \right) \times L_w \\ p_{i,y}^d = \left(\frac{p_i^d - 1}{L_s} + 0.5 \right) \times L_w, \end{cases} \tag{16}$$

where L_s is the length of the rows and columns in the wind farm (in this study the rows and columns are of equal length). L_w is the width of each grid where the turbines are placed. In order to accurately calculate the power generation of each turbine under wind direction v_w, we need to bring along the wind direction information:

$$\begin{pmatrix} p_{i,x,v_w}^d \\ p_{i,y,v_w}^d \end{pmatrix} = \begin{pmatrix} \cos(v_w) & -\sin(v_w) \\ \sin(v_w) & \cos(v_w) \end{pmatrix} \begin{pmatrix} p_{i,x}^d \\ p_{i,y}^d \end{pmatrix}. \tag{17}$$

4.2. Exemplar genetic learning

In [33,40], an effective exemplar construction-based PSO algorithm (GLPSO) is proposed, which generates a learning exemplar through genetic operators to guide the updating of the PSO algorithm. In the *Crossover* section of this algorithm, the method used in the previous work can be represented as:

$$o_{i,d} = \begin{cases} r' \cdot p_{i,d} + (1 - r') g_d, & \text{if} \quad F_i > F_{k_d} \\ p_{j_d,d}, & \text{otherwise} \end{cases} \tag{18}$$

where r' is a random parameter in $[0, 1]$. $p_{i,d}$ ($pbest$) is the historical optimal position of the dth turbine (particle) of the ith individual and g_d ($gbest$) is the global optimal position of the dth turbine (particle). $i \in 1, 2, \ldots, M$, $d \in 1, 2, \ldots, D$, M and D are population size and problem dimensions (number of wind turbines), respectively. This approach makes it possible to further improve the performance of good particles by integrating information from the global best particle, so that the offspring of poor particles have more dimensions from another better particle. This arithmetic crossover utilizes the historical search experience of particles in PSO to improve gene quality.

Remark 1. The randomization factor r alone does not ensure that the algorithm consistently yields constructive outcomes with each execution. Drawing inspiration from the utilization of historical search data to enhance the PSO approach, we propose leveraging *Advantage* (defined by Eq. (23)) in PPO to optimize the algorithm. The computation of the *Advantage* function involves utilizing historical information such as past states, actions, and reward data to guide strategy improvement. This approach allows for more informed decision-making and enhances the algorithm's efficacy over time.

4.3. Historical information optimization

To capture this historical information to guide the updating of the strategy by RL, we make the first term of Eq. (18):

$$o_{i,d} = r' \cdot p_{i,d} + (1 - r') \cdot g_d, \quad \text{if} \quad F_i > F_{k_d} \tag{19}$$

rewritten as:

$$o_{i,d} = W_\alpha \cdot p_{i,d} + W_\beta \cdot g_d, \quad \text{if} \quad F_i > F_{k_d} \tag{20}$$

where W_α and W_β are two uncorrelated weight variables, but we do not want to create too much destructive perturbation to the original values, so the range of values remains at $[0, 1]$. With this setup, we are able to more accurately capture information about the optimal solution from historical searches during the search process, without having to convert the $p_{i,d}$ and g_d weight relationship to be limited to a single parameter r. This approach provides a more nuanced understanding of the search space and helps to improve the performance of the algorithm.

Under the new constraints, the objective of optimizing W_α and W_β is to find a set of optimal weights such that the combination of $p_{i,d}$ and g_d improves the global convergence of the PSO. We can define an optimization objective function to evaluate the performance under different weights:

$$\max_{W_\alpha, W_\beta} f(W_\alpha \cdot p_{i,d} + W_\beta \cdot g_d) \tag{21}$$

Remark 2. We assert that $pbest$ and $gbest$ should be independent variables, each playing a distinct role in the algorithm. By allowing them to operate independently, we can more accurately capture information about optimal particles in historical searches. This approach offers a more nuanced understanding of the search space and avoids the simplistic restriction of their relationship using a single parameter r or its complement $1 - r$.

PPO is based on policy gradient and its objective function is:

$$O_{ppo}(\theta) = \mathop{\mathbb{E}}_{s_t, a_t \sim \pi_{\theta'}} [\min(r(\theta) A_\pi(s_t, a_t), \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon) A_\pi(s_t, a_t))] \tag{22}$$

In PPO, the *Advantage* function the relative superiority of the next state action of the current strategy over the current strategy. By calculating the *Advantage* function, PPO can estimate whether the parameters chosen in different states are better than average. If the settings of

W_α and W_β perform better in certain situations, the $Advantage$ function is biased towards positive values, which enhances the likelihood that this pair of parameters will be selected. $Advantage$ function to represent the advantage of taking an action in a given state relative to the average, which can be expressed as:

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t), \tag{23}$$

where s_t and a_t denote states and actions taken at moment t. The s_t refers here specifically to the parameters W_α and W_β to be adjusted:

$$s_t = (W_\alpha, W_\beta), \tag{24}$$

the action a_t is considered to regulate the specific value of s_t, and the specific action space is given in Eq. (27). $Q_\pi(s_t, a_t)$ denotes the expected value of the cumulative reward that can be obtained after taking action a_t in state s_t, i.e., the *action*-value function. $V_\pi(s_t)$ denotes the expected value of the cumulative reward that can be obtained when the action is taken in state st, i.e., the *state*-value function. $r(\theta)$ is the ratio of the old and new strategies, denoted as:

$$r(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}, \tag{25}$$

where $\pi_{\theta(a_t|s_t)}$ denotes the probability of choosing an action a_t in given state s_t of the current policy network, $\pi_{\theta'}$ denotes the old policy network.

With the help of RL, make W_α and W_β dynamic tuning. In each iteration t, PPO is used to update the weights in response to the current optimization requirements:

$$W_\alpha^{(t+1)} = W_\alpha^{(t)} + \Delta W_\alpha, \quad W_\beta^{(t+1)} = W_\beta^{(t)} + \Delta W_\beta \tag{26}$$

where ΔW_α and ΔW_β is the increment obtained through the reinforcement learning strategy, which is used to continuously optimize the weights during the iteration process and gradually approach the optimal solution.

Remark 3. Traditional PPO experience pools focus solely on reinforcement learning history, while PSO's historical information is limited to its own iterative optimization. This study is the first to integrate both by combining PSO and PPO data in a unified experience pool, enabling PPO to leverage the state, action, and reward data from each PSO iteration for experience replay. This fusion allows PPO to harness PSO's accumulated exploration paths, enhancing learning of optimal policy parameters and significantly improving both convergence speed and global search capability.

Unlike traditional RL, PPO as deep reinforcement learning uses a deep neural network to represent the policy (π_θ), enabling end-to-end changes in RL. Instead of π_θ, we use an $Actor$ network constructed with three fully connected layers plus a Softmax as classifier.

Since it is the PPO that is used during the iteration of the algorithm, we do not want the PPO to interfere too much with the algorithm in a destructive way, so we use the $Clip$ correction term to limit $r(\theta)$ to the interval $[1-\epsilon, 1+\epsilon]$. ϵ is a magnitude hyperparameter for control strategy updating, which in this study is set to 0.2.

In order to need to select actions based on the output probabilities of the policy network, the PPO strategy is utilized to find more reasonable and precise W_α and W_β. The state dimension is the value of W_α and W_β, and the action dimension is four, which can be represented as the four actions of W_α and W_β, respectively:

$$a_t = \begin{cases} a_t^{W_\alpha} + 1\times10^{-3}, & \text{if } A_l = 0 \mid A_l \in \{AL\} \\ a_t^{W_\beta} - 1\times10^{-3}, & \text{if } A_l = 1 \mid A_l \in \{AL\} \\ a_t^{W_\alpha} - 1\times10^{-3}, & \text{if } A_l = 2 \mid A_l \in \{AL\} \\ a_t^{W_\beta} + 1\times10^{-3}, & \text{if } A_l = 3 \mid A_l \in \{AL\} \\ \vdots \end{cases} \tag{27}$$

Algorithm 1: The pseudo-code of RPSO

```
Input : WindSpeed, Winddirection, N, and M
Output: Best layout and η of array
for i = 1 to M do
    Initializing coordinate and velocity p_i ;
    Evaluating p_i by Eq. (13);
end
Calculate gbest from pbest;
Initializing W_α, W_β.;
while it ≤ Maxit do
    Incoming RL pool;
    for i = 1 to Maxepisode do
        Sampled under Eq. (29) by π_θ;
        Choose W_α, W_β by Eq. (27);
        Calculate Advantage by Eqs. (23)–(25);
        Update Actor, Critic;
        Update π_θ;
        if reward ≥ MaxReward then
            Break;
        end
    end
    Update pbest, gbest through W_α and W_β
    by Eqs. (30)–(31);
end
```

A_l denotes the action labels sampled from the distribution of the discrete action space. AL is a finite set of real numbers $\mathbb{R}$:

$$AL = \{0, 1, 2, \dots\}, \tag{28}$$

as the problem size increases, the action space dimensions become higher, and in this study, the action space was set up in four modes. $a_t^{W_\alpha}$ and $a_t^{W_\beta}$ represent actions taken on W_α and W_β at moment t. In the discrete action space we use the $Categorical$ distribution, which can be expressed as a probability mass function:

$$P(X = \mathbf{k}) = \prod_{i=1}^{K} p_i^{k_i}, \tag{29}$$

where k is a k-dimensional vector of random variables taking values, k_i denotes the number of occurrences of the ith category, and p_i denotes the probability of the ith category. The results generated by PPO will further affect the PSO algorithm, which can be expressed as follows:

$$\begin{cases} v^{it+1} = & \omega \cdot v^{it} + c_1 \cdot e_1 \cdot p' - c_2 \cdot e_2 \cdot \boldsymbol{P}^{it} \\ \boldsymbol{P}^{it+1} = & \boldsymbol{P}^{it} + v^{it+1}, \end{cases} \tag{30}$$

where c_1 and c_2 are the individual learning factor and the social learning factor, respectively. e_1 and e_2 are random numbers between $[0, 1]$. ω is the inertia constant. p' is the $pbest$ after being affected by $O_{ppo}(\theta)$:

$$\begin{cases} g_d' & = g_d \cdot O_{ppo}(\theta) \\ p' & = g_d' + p_{i,d} \cdot O_{ppo}(\theta). \end{cases} \tag{31}$$

The pseudo-code for RPSO is shown in Algorithm 1. First, the given wind farm information, the number of turbines N with the number of populations M, etc. is input. A set of turbine positions p_i is randomly initialized and the $pbest$ and $gbest$ are computed using Eq. (13). Subsequently, initialize the two scaling factors W_α and W_β that we are trying to learn. In PPO, sampling is done by Eq. (29), and new actions (here W_α and W_β) are generated according to Eq. (27). Subsequently, the dominance function is computed via Eqs. (23)–(25), and the parameters are updated via the gradient. This cycle continues until the reward value reaches a specified value or the number of RL interactions reaches a specified value to jump out of the cycle. The newly learned W_α and W_β are reorganized into $pbest$ and $gbest$ via Eqs. (30)–(31).

Table 2
A brief summary of the algorithms and strategies used in this paper.

Algorithm	Description	Parameter
AGA [22]	Self-informed genetic algorithm.	$p_c = 0.6,\ p_m = 0.1,\ p_e = 0.2$
AGPSO [32]	An adaptive strategy particle swarm optimizer.	$\omega = 0.73,\ c = 1.50,\ p_m = 0.01$
ALGSA [41]	An aggregative learning gravitational search algorithm.	$lim = 2,\ p = 0.9$
BSA [42]	Bird swarm algorithm.	$c_1 = 1.5,\ c_2 = 1.5$
CGPSO [33]	A chaotic local search-based particle swarm optimizer.	$\omega = 0.73,\ c = 1.50,\ p_m = 0.01$
CJADE [43]	Chaotic local search-based differential evolution algorithms.	$c = 0.01,\ p = 0.05,\ CR = F = 0.5$
CLPSO [44]	Comprehensive learning particle swarm optimizer.	$c_1 = 1.5,\ c_2 = 1.5$
GLPSO [40]	Genetic learning particle swarm optimization.	$\omega = 0.73,\ c = 1.50,\ p_m = 0.01$
HGSA [45]	A hierarchical gravitational search algorithm.	$R_n = 2$
IntGA [32]	An adaptive strategy-incorporated integer genetic algorithm.	$\omega = 0.73,\ c = 1.50,\ p_m = 0.01$
IWO [46]	Invasive weed optimization.	$S_{min} = 2,\ S_{max} = 5,\ \sigma = 0.5$
$LSHADE_S$ [47]	An excellent variant of LSHADE.	$L_r = 0.80$

Table 3
The setup parameters of wind turbines.

Parameter	Value
Scale of wind farm	21 × 21
Width of grids	77 × 3 m
Surface roughness	0.25 mm
Rotor diameter r	77 m
Wind turbines height	80 m

4.4. Characteristics of the proposed RPSO

The proposed RPSO algorithm leverages the advantage of experience playback in reinforcement learning, distinguishing it from traditional PSO methods. This strategy, which utilizes historical search information to direct the iteration process, has been extensively employed in EAs in the past. However, the integration of historical information across both evolutionary algorithms and reinforcement learning domains is relatively uncommon. Specifically, RL parameters are adjusted to steer the EA strategy based on historical search data. In previous approaches, EA parameters were often randomly determined. With the incorporation of RL, the algorithm replaces this randomization with a more systematic and efficient updating method, resulting in solutions that better guide the algorithm's evolution.

5. Experimental results

In this section, we will conduct a comprehensive comparison of the proposed RPSO algorithm with twelve state-of-the-art intelligent algorithms. These algorithms include CGPSO, CLPSO, GLPSO, AGPSO, CJADE, LSHADE-SPACMA, BSA, HGSA, ALGSA, AGA, IntGA, and IWO. They will be evaluated across four distinct wind scenarios. Among the algorithms under comparison, CGPSO, CLPSO, GLPSO, and AGPSO represent advanced variants of PSO developed in recent years. Similarly, CJADE and LSHADE-SPACMA are notable variants of the DE series renowned for their effectiveness. The remaining algorithms comprise variants of genetic and metaheuristic algorithms known for their robust performance in optimization tasks. The brief description and parameter settings of all the algorithms are shown in Table 2. The setup parameters of wind turbines are summarized in Table 3. In the ablation experiment, we focus on analyzing the effect of RL.

5.1. Experiment setup

All algorithms in this study optimize the WFLO problem for 30, 35, and 40 wind turbine sizes under four different complex wind scenarios. These four wind scenarios are shown in Fig. 4. Wind scenario 1 (S1) represents a scenario with three wind speeds and twelve wind directions; wind scenario 2 (S2) represents a uniformly distributed scenario with three wind speeds and twelve wind directions; wind scenario 3 (S3) represents a scenario with four wind speeds and twelve wind directions; and wind scenario 4 (S4) represents a scenario with six wind speeds and twelve wind directions. The number of iterations for all algorithms was 200, the population size was 50, and 10 independent non-repeating experiments were performed to eliminate chance. All algorithms are implemented on an experimental platform with Intel (R) Core (TM) i9-12900K 3.90 GHz CPU, 32 GB of RAM, GPU RTX 3090, and 24 GB of video memory.

5.2. Analysis of results for wind scenarios

Table 4 enumerates the conversion efficiencies of all algorithms across four wind scenarios, considering three turbine sizes positioned at 30, 35, and 40. The conversion efficiencies are determined using Eq. (13). The numbers in parentheses denote the standard deviation of the results obtained from all independent and unrepeated experiments. Additionally, we conducted the Wilcoxon signed-rank test for each algorithm, yielding a p-value less than 0.05. It is evident that the proposed RPSO algorithm consistently achieves the highest conversion efficiency across all scenarios. Additionally, the underlined entries in the figure represent the second-highest experimental results. Moreover, the figure illustrates that the enhanced variants of PSO, namely CGPSO, GLPSO, and AGPSO, perform admirably in addressing the WFLO problem. In addition, we have done a sensitivity analysis of the algorithm's performance for various parameter scenarios. As shown in Table 5, the first-order sensitivity index (S1) means the contribution of each parameter independently to the result. Total effect sensitivity index (ST) means the total contribution of each parameter and its interaction with other parameters to the results. Second-order interaction effects (S2) means the contribution to the results of the interactions between the parameters, listing only the meaningful interaction terms. From the data in the table, it can be seen that algorithm is the dominant factor affecting the results, accounting for 73.25% of the total effect, indicating that the choice of different algorithms has the most significant effect on the final results. Wind condition and turbine count have a smaller effect, but wind condition shows a smaller effect when combined with the algorithm combination showed some interaction effect, which suggests that different algorithms may have different effects under different wind conditions. Turbine count has a lower total contribution and almost no interaction effect, suggesting that it plays a smaller role in the output.

The stability of all algorithms is depicted in Fig. 5. Notably, the proposed RPSO algorithm achieves the highest conversion efficiency while maintaining results within a relatively stable range across experiments, exhibiting minimal fluctuations and variability. Fig. 7 illustrates the convergence behavior of all algorithms 200 iterations for four wind scenarios. In all scenarios, RPSO consistently outperforms other algorithms, exhibiting significant improvement from the first 20 iterations onward. However, in scenarios S2 and S4, similar convergence values are observed with GLPSO around the 20th iteration. This occurrence could be attributed to the standardized nature of S2 and the increased complexity of S4, which features 6 wind speeds and 12 wind directions, introducing higher levels of stochasticity compared to other scenarios.

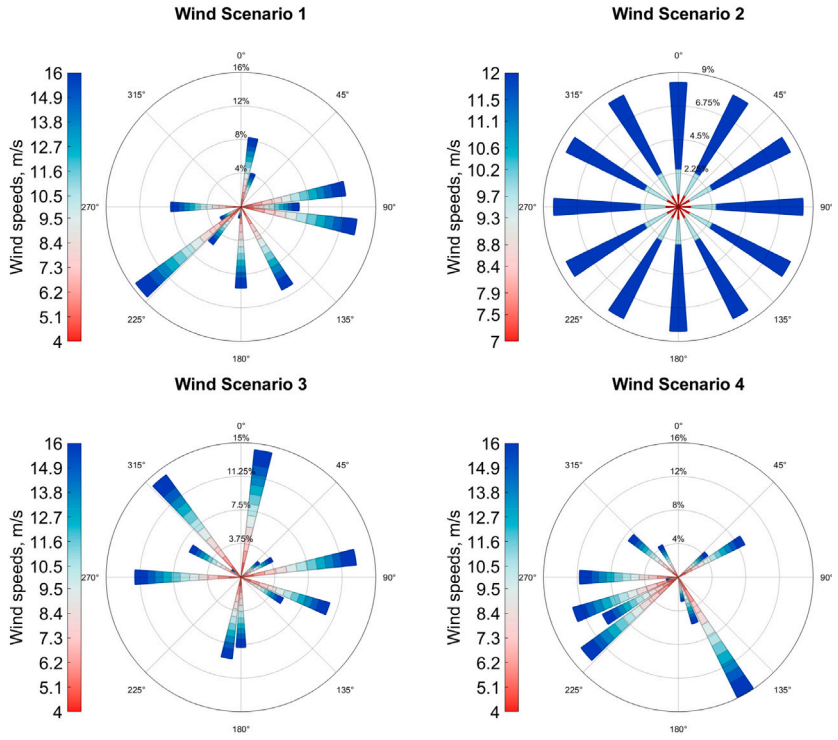

Fig. 4. Four kinds of wind scenarios rose figure.

Table 4
The Conversion efficiency (%) of the algorithm for four wind scenarios and three turbine scales.

		TN30	TN35	TN40	*p*-value			TN30	TN35	TN40	*p*-value
	AGA	95.58 (±0.16)	94.16 (±0.22)	92.87 (±0.19)	9.77E−04		AGA	95.01 (±0.26)	93.41 (±0.26)	92.01 (±0.28)	9.77E−04
	IntGA	93.87 (±0.51)	92.65 (±0.67)	91.05 (±0.31)	9.77E−04		IntGA	94.66 (±0.68)	92.76 (±0.88)	91.40 (±0.38)	9.77E−04
	HGSA	94.94 (±0.09)	93.77 (±0.21)	92.48 (±0.28)	9.77E−04		HGSA	93.55 (±0.21)	91.76 (±0.29)	89.78 (±0.24)	9.77E−04
	BSA	95.87 (±0.37)	94.39 (±0.46)	93.11 (±0.43)	9.77E−04		BSA	94.51 (±0.49)	92.71 (±0.54)	91.20 (±0.43)	9.77E−04
	ALGSA	94.57 (±0.21)	94.31 (±0.28)	92.52 (±0.29)	9.77E−04		ALGSA	94.01 (±0.42)	92.36 (±0.32)	89.72 (±0.28)	9.77E−04
	IWO	93.28 (±0.16)	91.70 (±0.21)	90.32 (±0.17)	9.77E−04		IWO	91.92 (±0.21)	90.35 (±0.34)	88.41 (±0.37)	9.77E−04
Wind scenario 1	CJADE	96.05 (±0.36)	94.65 (±0.00)	93.15 (±0.00)	9.77E−04	Wind scenario 2	CJADE	95.26 (±0.14)	93.46 (±0.26)	91.35 (±0.00)	9.77E−04
	$LSHADE_S$	97.48 (±0.24)	96.29 (±0.21)	94.78 (±0.24)	1.95E−03		$LSHADE_S$	96.81 (±0.42)	95.36 (±0.20)	94.00 (±0.36)	9.77E−04
	AGPSO	97.53 (±0.19)	96.15 (±0.19)	95.32 (±0.32)	9.77E−04		AGPSO	97.26 (±0.36)	95.85 (±0.21)	94.40 (±0.21)	9.77E−04
	CLPSO	93.61 (±0.13)	91.67 (±0.00)	90.18 (±0.03)	9.77E−04		CLPSO	91.62 (±0.36)	89.80 (±0.04)	87.63 (±0.00)	9.77E−04
	GLPSO	97.48 (±0.21)	96.43 (±0.18)	95.23 (±0.34)	9.77E−04		GLPSO	97.15 (±0.14)	95.91 (±0.27)	94.26 (±0.32)	9.77E−04
	CGPSO	97.51 (±0.26)	96.32 (±0.20)	95.13 (±0.20)	9.77E−04		CGPSO	97.19 (±0.20)	95.93 (±0.24)	94.42 (±0.22)	9.77E−04
	RPSO	**98.17 (±0.18)**	**97.21 (±0.44)**	**95.83 (±0.31)**	–		RPSO	**97.77 (±0.30)**	**96.71 (±0.34)**	**95.06 (±0.31)**	–
	AGA	96.10 (±0.11)	95.10 (±0.13)	94.00 (±0.11)	9.77E−04		AGA	96.94 (±0.15)	96.04 (±0.11)	95.05 (±0.20)	9.77E−04
	IntGA	95.10 (±0.35)	94.02 (±0.39)	92.67 (±0.43)	9.77E−04		IntGA	95.78 (±0.37)	94.46 (±0.37)	93.40 (±0.68)	9.77E−04
	HGSA	95.94 (±0.17)	94.70 (±0.14)	93.56 (±0.17)	9.77E−04		HGSA	96.40 (±0.20)	95.12 (±0.15)	94.02 (±0.24)	9.77E−04
	BSA	96.27 (±0.20)	95.37 (±0.32)	94.15 (±0.22)	9.77E−04		BSA	96.91 (±0.28)	95.65 (±0.51)	94.71 (±0.35)	9.77E−04
	ALGSA	96.20 (±0.27)	95.07 (±0.17)	93.44 (±0.26)	9.77E−04		ALGSA	96.41 (±0.19)	95.08 (±0.30)	93.89 (±0.27)	9.77E−04
	IWO	94.61 (±0.23)	93.36 (±0.14)	92.11 (±0.07)	9.77E−04		IWO	95.20 (±0.17)	94.02 (±0.17)	92.92 (±0.21)	9.77E−04
Wind scenario 3	CJADE	96.71 (±0.17)	95.47 (±0.13)	94.14 (±0.01)	9.77E−04	Wind scenario 4	CJADE	96.94 (±0.08)	96.09 (±0.11)	94.96 (±0.22)	9.77E−04
	$LSHADE_S$	97.37 (±0.08)	96.39 (±0.13)	95.38 (±0.24)	9.77E−04		$LSHADE_S$	98.28 (±0.26)	97.59 (±0.23)	96.85 (±0.26)	9.77E−04
	AGPSO	97.47 (±0.12)	96.54 (±0.13)	95.55 (±0.16)	9.77E−04		AGPSO	98.29 (±0.14)	97.55 (±0.21)	96.94 (±0.24)	9.77E−04
	CLPSO	94.53 (±0.17)	93.43 (±0.09)	91.69 (±0.13)	9.77E−04		CLPSO	95.06 (±0.17)	93.47 (±0.36)	92.46 (±0.08)	9.77E−04
	GLPSO	97.48 (±0.14)	96.58 (±0.13)	95.53 (±0.13)	9.77E−04		GLPSO	98.14 (±0.20)	97.49 (±0.16)	97.01 (±0.23)	9.77E−04
	CGPSO	97.48 (±0.12)	96.52 (±0.09)	95.64 (±0.17)	9.77E−04		CGPSO	98.33 (±0.11)	97.57 (±0.29)	96.85 (±0.20)	9.77E−04
	RPSO	**97.82 (±0.06)**	**97.07 (±0.21)**	**95.90 (±0.31)**	–		RPSO	**98.68 (±0.14)**	**98.14 (±0.17)**	**97.33 (±0.41)**	–

Table 5
Sensitivity analysis of each hyper-parameter using variance decomposition (Sobol).

Parameters	First-order sensitivity (S1)	Total-order sensitivity (ST)	Second-order interactions (S2)
Wind_C	0.1097	0.1860	Wind_C - Turbines: 0.0055
Turbines	0.1400	0.1301	Wind_C - Algorithms: 0.1005
Algorithms	0.6695	0.7325	Turbines - Algorithms: −0.0184

Fig. 6 illustrates the distribution of selected algorithms across four wind angles under scenario S4, showcasing their response to wake effects. Each black dot in the wind farm denotes a wind turbine, while the color of each region signifies the prevailing wind speed in that area. The majority of regions impacted by wake effects are situated behind the turbines, highlighting the importance of minimizing wake

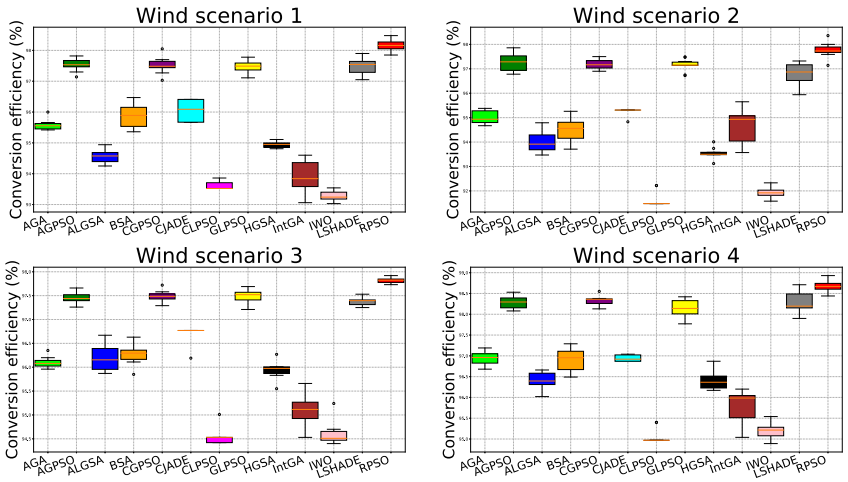

Fig. 5. Box figures of all algorithms for four wind scenarios.

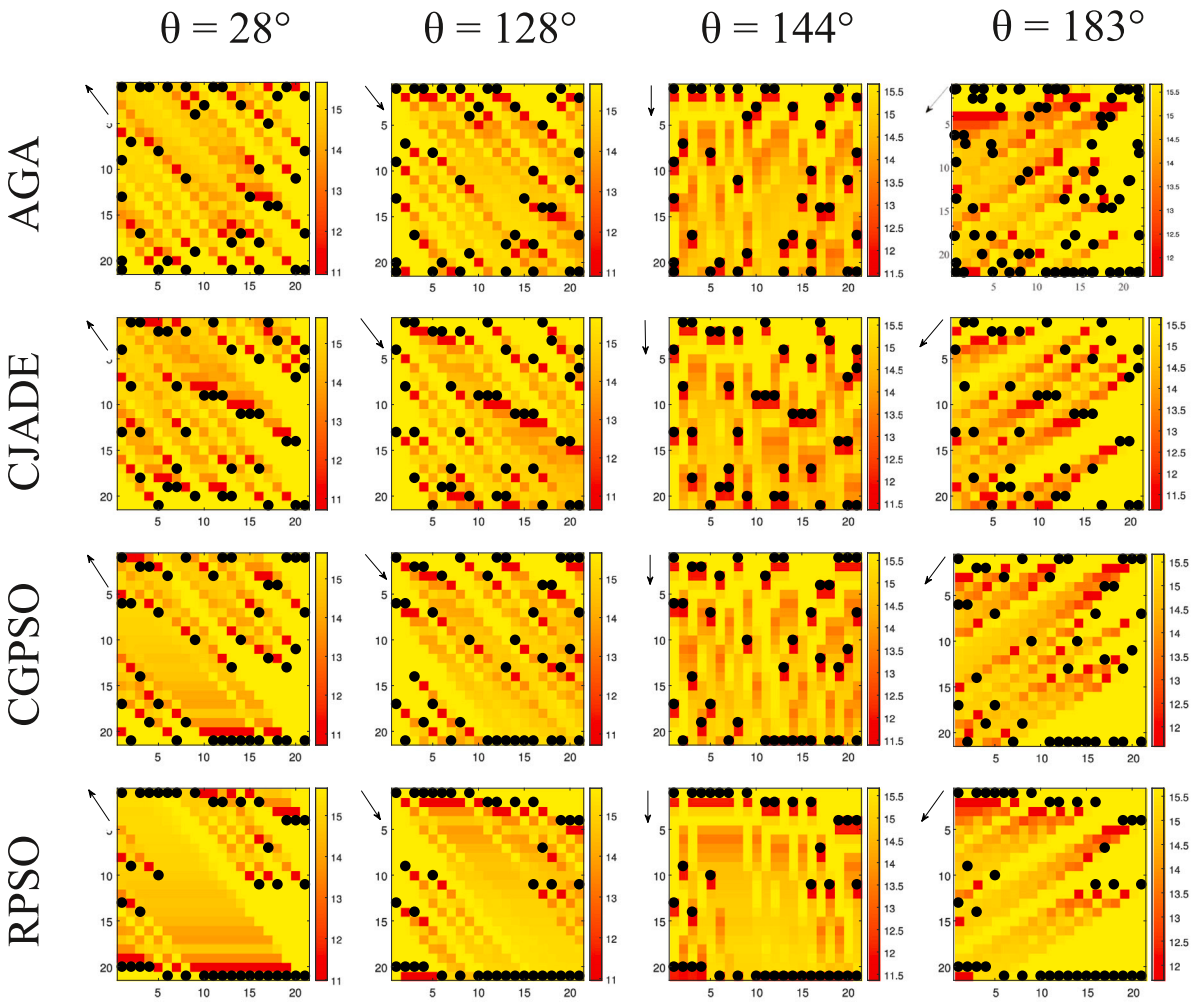

Fig. 6. The layouts of four algorithms with wake attenuation under four wind directions. The black dot represents the turbine and the red to yellow color blocks indicate low to high wind speeds at that point.

effects on downstream turbines to address this challenge effectively. Examining the distribution, it is evident that many underperforming AGAs are positioned in areas severely affected by wake effects. Conversely, RPSOs are strategically located on both sides of the wind field in a linear arrangement, potentially minimizing the impact of wake effects on downstream turbines. This strategic placement underscores the efficacy of RPSO in optimizing turbine layout to mitigate wake effects and enhance overall performance.

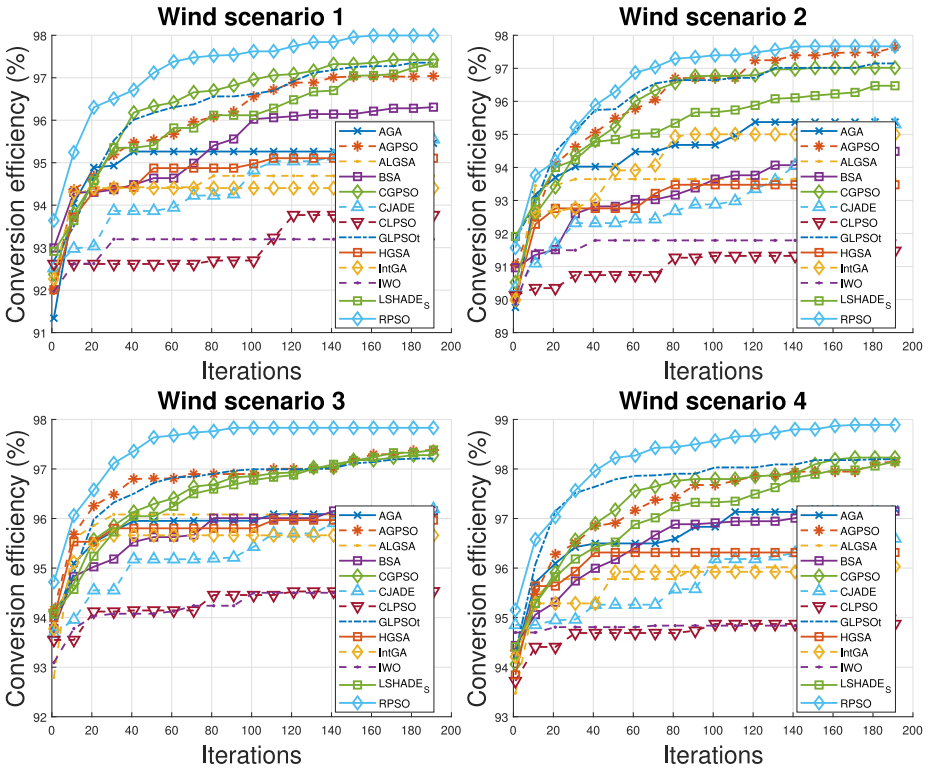

Fig. 7. Convergence figures of all algorithms for four wind scenarios.

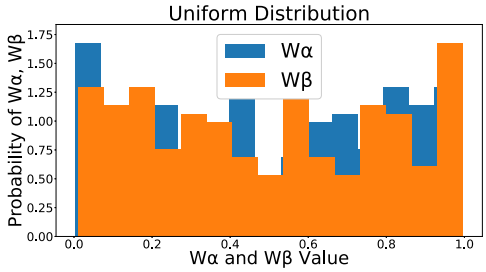

Fig. 8. Uniform distribution.

5.3. Analysis of RL

We first analyze the results produced by RPSO through the distribution and convergence plots of the new weights W_α and W_β. Subsequently, to further validate the effectiveness of RL, we compare the performance of the algorithm without RL through ablation experiments.

5.3.1. Analysis of distribution of results

In the original work, r' is obeying a uniform distribution from 0 to 1. Its probability density function can be expressed as follows:

$$prob(r') = \begin{cases} 1, & \text{if} \quad 0 \le r' \le 1 \\ 0. & \text{otherwise} \end{cases} \tag{32}$$

The probability that any value r' is taken in the interval $[0,1]$ is equal, i.e., the probability density function is constant. The distribution is shown in Fig. 8, all values are evenly distributed between 0 and 1. By learning from the empirical playback of RL, our defined W_α and W_β are no longer disordered random values, but converge in a more optimal direction by learning historical search information. The portion of the distribution generated by the RPSO is shown in Fig. 9. The blue square bar represents W_α and the orange square bar represents W_β. It can be seen that after RL learning, the value of W_α is higher than W_β in most cases, which can reflect the fact that the effect of *pbest* on the next generation may be a bit more favorable. The convergence curves of W_α and W_β with the number of iterations are shown in Fig. 10. Where the green curve represents the original r', which floats randomly between 0 and 1, the blue curve represents W_α, and the orange curve represents W_β. It can be seen that the two weight ratios converge in a specific direction from the initial value, which reflects the guiding role that the empirical information of RL plays in the algorithm.

Remark 4. In contrast to previous work, we explore the solution space for negative values. It is argued in the higher dimensional space of the PPO that the *gbest*, although acting as a globally optimal particle, does not necessarily have to be additive to the algorithm by a certain weighting, but can also be subtractive. We believe that this represents two different directions of exploration.

5.3.2. Ablation experiment

To further validate the effectiveness of our proposed RL strategy within the algorithm, we conducted ablation experiments on the PPO algorithm employed in RL. The results of these experiments are presented in Table 6. Fig. 11 shows the convergence of the effect of whether or not to use RL on the model algorithm for four wind conditions (30 turbines). As can be seen from the figure, RL enhances the model significantly and plays a constructive role in the whole phase of algorithm convergence, not only speeding up the convergence of the algorithm, but also improving the performance. No-RL denotes the algorithm after removing the PPO strategy. From the experimental results, it can be seen that the algorithm under the no-RL strategy loses to RPSO in all scenarios and for all turbine sizes. To eliminate chance factors, we conducted the Wilcoxon signed-rank test on the experimental outcomes (see Table 7), revealing that all p-values were less than 0.05. RL primarily facilitates the utilization of the experience pool playback strategy within the PSO algorithm, leveraging historical search data to inform the direction of subsequent iterations. Without

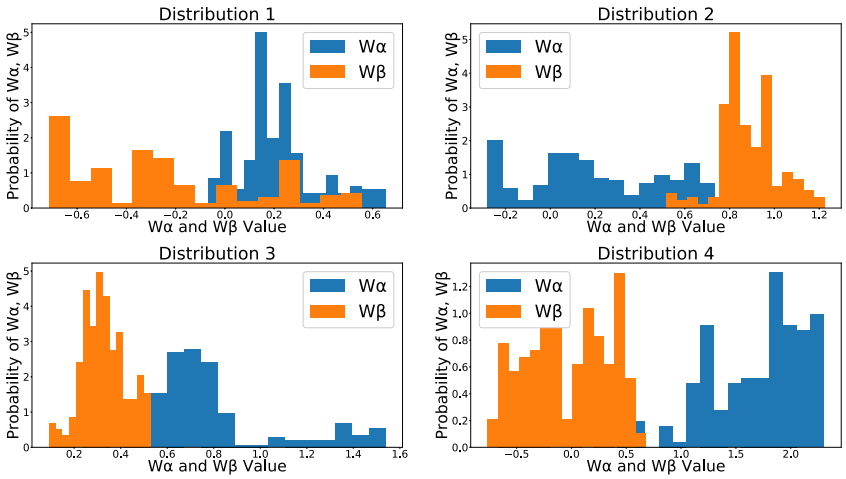

Fig. 9. Probability distributions of the 4 RPSO species W_α and W_β.

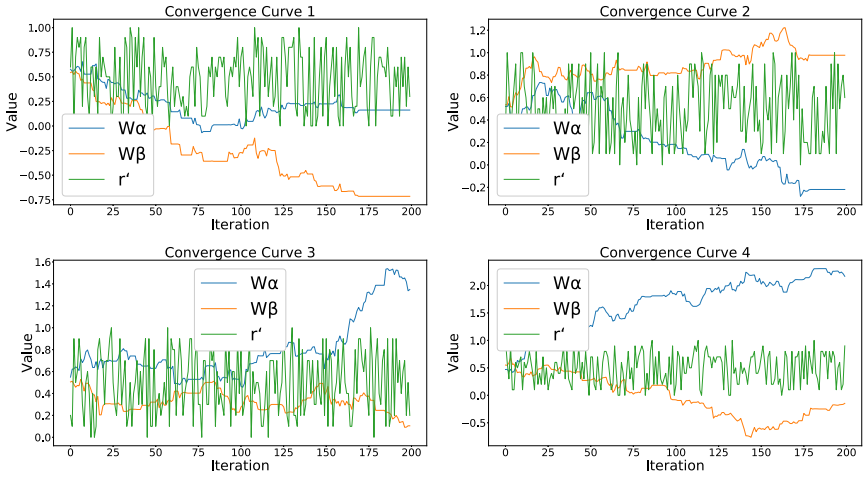

Fig. 10. Convergence curves of W_α and W_β with number of iterations.

RL, the original algorithm relies solely on randomness during iteration:

$$o_{i,d} = r' \cdot p_{i,d} + (1 - r')g_d. \tag{33}$$

There is no rigorous theory justifying this constrained relationship. We believe this has the potential to have a destructive or meaningless effect on the algorithm's search.

Under the influence of RL, W_α and W_β searches based on historical information would reasonably lead to the generation of new *gbest*:

$$g'_d = g_d \cdot O_{ppo}(\theta). \tag{34}$$

The *gbest* acts as a globally optimal particle and has a certain percentage to influence the *pbest* in the next iteration:

$$p' = g'_d + p_{i,d} \cdot O_{ppo}(\theta), \tag{35}$$

with this RL-guided learning strategy, RPSO can more effectively utilize historical search information to allow gbest and pbest to maximize their respective potentials in iterations. From the experimental results, we simultaneously expanded the solution space of the search and explored the effect of negative values with good results.

6. Conclusion

In this study, we propose an RPSO algorithm that integrates a PPO deep reinforcement learning strategy with the aim of accelerating the convergence of the PSO algorithm by combining historical information from the PPO and PSO strategies, as well as the ability to search for the global optimum, minimize the wake impact on the downstream turbines and maximize the energy output. Experimental evaluations are carried out in four different wind farm environments, considering turbine sizes of 30, 35 and 40, and comparisons with 12 SOTA algorithms for solving the WFLO problem show that our proposed RPSO is highly

Table 6
Comparison of algorithm without RL strategy with RPSO output on four wind scenarios.

	Methods	TN30	TN35	TN40
S 1	RPSO	**98.17 (±0.18)**	**97.21 (±0.44)**	**95.83 (±0.31)**
	No-RL	97.62 (±0.29)	96.42 (±0.28)	95.38 (±0.00)
S 2	RPSO	**97.77 (±0.30)**	**96.71 (±0.34)**	**95.06 (±0.31)**
	No-RL	97.06 (±0.29)	95.77 (±0.25)	94.18 (±0.25)
S 3	RPSO	**97.82 (±0.06)**	**97.07 (±0.21)**	**95.90 (±0.31)**
	No-RL	97.51 (±0.11)	96.61 (±0.21)	95.70 (±0.20)
S 4	RPSO	**98.68 (±0.14)**	**98.14 (±0.17)**	**97.33 (±0.41)**
	No-RL	98.29 (±0.15)	97.57 (±0.15)	96.92 (±0.25)

Table 7
Statistical test of *p*-value with No-RL.

–	S1	S2	S3	S4
p-value	1.95e−03	9.77e−04	9.77e−04	1.95e−03

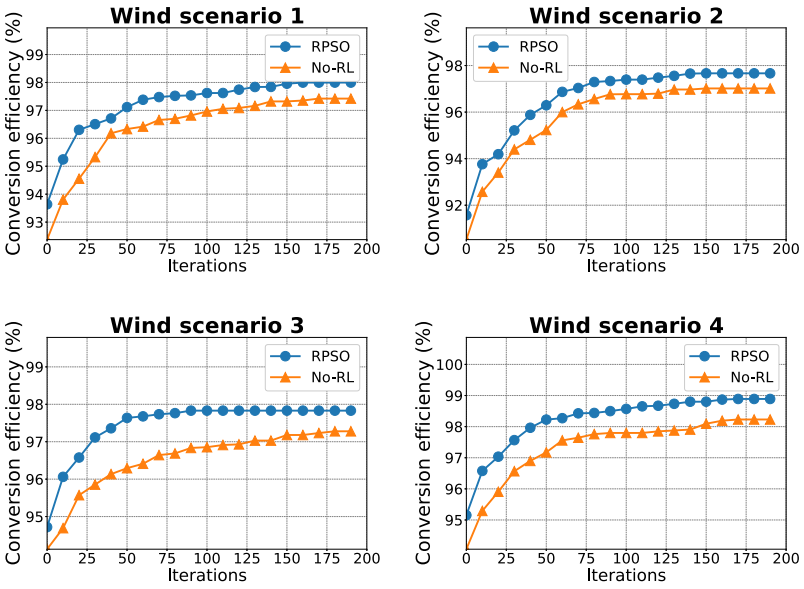

Fig. 11. Convergence comparison for four wind conditions with and without RL strategies.

competitive. The experimental results demonstrate the convergence and robustness of our algorithm, and the ablation experiments highlight the key role of RL, which guides the global best particles and the historical best particles with appropriate weights in the subsequent iterations. RPSO can significantly enhance the power generation efficiency of wind farms by mitigating wake effects and reducing deployment costs in real-world scenarios, thereby promoting the sustainable and effective use of renewable energy.

CRediT authorship contribution statement

Zihang Zhang: Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Jiayi Li:** Writing – original draft, Software, Conceptualization. **Zhenyu Lei:** Writing – review & editing, Software, Conceptualization. **Qianyu Zhu:** Writing – review & editing, Conceptualization. **Jiujun Cheng:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Shangce Gao:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgments

This research was partially supported by the Japan Society for the Promotion of Science (JSPS), Japan KAKENHI under Grant JP23K24899, and Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145.

Data availability

Data will be made available on request.

References

[1] Cheng M, Zhu Y. The state of the art of wind energy conversion systems and technologies: A review. Energy Convers Manage 2014;88:332–47.
[2] Mohammadi K, Mostafaeipour A. Using different methods for comprehensive study of wind turbine utilization in Zarrineh, Iran. Energy Convers Manage 2013;65:463–70.
[3] Kadri A, Marzougui H, Aouiti A, Bacha F. Energy management and control strategy for a DFIG wind turbine/fuel cell hybrid system with super capacitor storage system. Energy 2020;192:116518.
[4] Zhang B, Hu W, Li J, Cao D, Huang R, Huang Q, et al. Dynamic energy conversion and management strategy for an integrated electricity and natural gas system with renewable energy: Deep reinforcement learning approach. Energy Convers Manage 2020;220:113063.
[5] Zhang B, Hu W, Cao D, Li T, Zhang Z, Chen Z, et al. Soft actor-critic–based multi-objective optimized energy conversion and management strategy for integrated energy systems with renewable energy. Energy Convers Manage 2021;243:114381.
[6] Zhao X, Jiaqiang E, Zhang Z, Chen J, Liao G, Zhang F, et al. A review on heat enhancement in thermal energy conversion and management using field synergy principle. Appl Energy 2020;257:113995.
[7] Jang D, Kim K, Kim K-H, Kang S. Techno–economic analysis and Monte Carlo simulation for green hydrogen production using offshore wind power plant. Energy Convers Manage 2022;263:115695.
[8] Cai Q, Yang Z, Jin C, Wang Z. Provably efficient exploration in policy optimization. In: International conference on machine learning. PMLR; 2020, p. 1283–94.
[9] Kuznetsova E, Li Y-F, Ruiz C, Zio E, Ault G, Bell K. Reinforcement learning for microgrid energy management. Energy 2013;59:133–46.
[10] Gu Y, Cheng Y, Chen CLP, Wang X. Proximal policy optimization with policy feedback. IEEE Trans Syst Man Cybern A 2021;52:4600–10.
[11] Zhang J, Zhang Z, Han S, Lü S. Proximal policy optimization via enhanced exploration efficiency. Inform Sci 2022;609:750–65.
[12] Rezaeiha A, Montazeri H, Blocken B. A framework for preliminary large-scale urban wind energy potential assessment: Roof-mounted wind turbines. Energy Convers Manage 2020;214:112770.
[13] Chen C, Liu H. Medium-term wind power forecasting based on multi-resolution multi-learner ensemble and adaptive model selection. Energy Convers Manage 2020;206:112492.
[14] Saad AS, El-Sharkawy II, Ookawara S, Ahmed M. Performance enhancement of twisted-bladed savonius vertical axis wind turbines. Energy Convers Manage 2020;209:112673.
[15] Mosetti G, Poloni C, Diviacco B. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. J Wind Eng Ind Aerodyn 1994;51:105–16.
[16] Marmidis G, Lazarou S, Pyrgioti E. Optimal placement of wind turbines in a wind park using Monte Carlo simulation. Renew Energy 2008;33:1455–60.
[17] Cheng B, Yao Y. Design and optimization of a novel U-type vertical axis wind turbine with response surface and machine learning methodology. Energy Convers Manage 2022;273:116409.

[18] Neshat M, Sergiienko NY, Nezhad MM, da Silva LSP, Amini E, Marsooli R, et al. Enhancing the performance of hybrid wave-wind energy systems through a fast and adaptive chaotic multi-objective swarm optimisation method. Appl Energy 2024;362:122955.
[19] Grady SA, Hussaini MY, Abdullah MM. Placement of wind turbines using genetic algorithms. Renew Energy 2005;30:259–70.
[20] Emami A, Noghreh P. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. Renew Energy 2010;35:1559–64.
[21] Chen Y, Li H, Jin K, Song Q. Wind farm layout optimization using genetic algorithm with different hub height wind turbines. Energy Convers Manage 2013;70:56–65.
[22] Ju X, Liu F, Wang L, Lee W-J. Wind farm layout optimization based on support vector regression guided genetic algorithm with consideration of participation among landowners. Energy Convers Manage 2019;196:1267–81.
[23] Ju X, Liu F. Wind farm layout optimization using self-informed genetic algorithm with information guided exploitation. Appl Energy 2019;248:429–45.
[24] Wang Y, Liu H, Long H, Zhang Z, Yang S. Differential evolution with a new encoding mechanism for optimizing wind farm layout. IEEE Trans Ind Inf 2017;14:1040–54.
[25] Yu Y, Zhang T, Lei Z, Wang Y, Yang H, Gao S. A chaotic local search-based LSHADE with enhanced memory storage mechanism for wind farm layout optimization. Appl Soft Comput 2023;141:110306.
[26] Feng J, Shen WZ. Solving the wind farm layout optimization problem using random search algorithm. Renew Energy 2015;78:182–92.
[27] Eroğlu Y, Seçkiner SU. Design of wind farm layout using ant colony algorithm. Renew Energy 2012;44:53–62.
[28] Bai F, Ju X, Wang S, Zhou W, Liu F. Wind farm layout optimization using adaptive evolutionary algorithm with Monte Carlo tree search reinforcement learning. Energy Convers Manage 2022;252:115047.
[29] Hou P, Hu W, Soltani M, Chen Z. Optimized placement of wind turbines in large-scale offshore wind farm using particle swarm optimization algorithm. IEEE Trans Sustain Energy 2015;6:1272–82.
[30] Pookpunt S, Ongsakul W. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. Renew Energy 2013;55:266–76.
[31] Tao S, Kuenzel S, Xu Q, Chen Z. Optimal micro-siting of wind turbines in an offshore wind farm using Frandsen–Gaussian wake model. IEEE Trans Power Syst 2019;34:4944–54.
[32] Lei Z, Gao S, Wang Y, Yu Y, Guo L. An adaptive replacement strategy-incorporated particle swarm optimizer for wind farm layout optimization. Energy Convers Manage 2022;269:116174.
[33] Lei Z, Gao S, Zhang Z, Yang H, Li H. A chaotic local search-based particle swarm optimizer for large-scale complex wind farm layout optimization. IEEE/CAA J Autom Sin 2023;10:1168–80.
[34] Dong H, Zhao X. Reinforcement learning-based wind farm control: Towards large farm applications via automatic grouping and transfer learning. IEEE Trans Ind Inf 2023.
[35] Yu X, Lu Y. Reinforcement learning-based multi-objective differential evolution for wind farm layout optimization. Energy 2023;284:129300.
[36] Yu X, Zhang W. A teaching-learning-based optimization algorithm with reinforcement learning to address wind farm layout optimization problem. Appl Soft Comput 2023;111135.
[37] Dong H, Zhang J, Zhao X. Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations. Appl Energy 2021;292:116928.
[38] Xie J, Dong H, Zhao X, Karcanias A. Wind farm power generation control via double-network-based deep reinforcement learning. IEEE Trans Ind Inf 2021;18:2321–30.
[39] Archer CL, Vasel-Be-Hagh A, Yan C, Wu S, Pan Y, Brodie JF, et al. Review and evaluation of wake loss models for wind energy applications. Appl Energy 2018;226:1187–207.
[40] Gong Y-J, Li J-J, Zhou Y, Li Y, Chung HS-H, Shi Y-H, et al. Genetic learning particle swarm optimization. IEEE Trans Cybern 2015;46:2277–90.
[41] Lei Z, Gao S, Gupta S, Cheng J, Yang G. An aggregative learning gravitational search algorithm with self-adaptive gravitational constants. Expert Syst Appl 2020;152:113396.
[42] Meng X-B, Gao XZ, Lu L, Liu Y, Zhang H. A new bio-inspired optimisation algorithm: Bird swarm algorithm. J Exp Theor Artif Intell 2016;28:673–87.
[43] Gao S, Yu Y, Wang Y, Wang J, Cheng J, Zhou M. Chaotic local search-based differential evolution algorithms for optimization. IEEE Trans Syst Man Cybern A 2019;51:3954–67.
[44] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 2006;10:281–95.
[45] Wang Y, Yu Y, Gao S, Pan H, Yang G. A hierarchical gravitational search algorithm with an effective gravitational constant. Swarm Evol Comput 2019;46:118–39.
[46] Karimkashi S, Kishk AA. Invasive weed optimization and its features in electromagnetics. IEEE Trans Antennas and Propagation 2010;58:1269–78.
[47] Li Y, Han T, Zhou H, Tang S, Zhao H. A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem. Inform Sci 2022;606:350–67.