CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Multi-Strategy Synergy-Based Differential Evolution for Large Wind Farm Layout Optimization

Haotian Li[1] | Yifei Yang[2] | Qiong Fu[1] | Zhenyu Lei[1] | Zhe Xu[3] | Shangce Gao[1]

[1]Faculty of Engineering, University of Toyama, Toyama, Japan | [2]Faculty of Science and Technology, Hirosaki University, Hirosaki-shi, Japan | [3]School of Computer Information & Engineering, Changzhou Institute of Technology, Changzhou, China

Correspondence: Shangce Gao (gaosc@eng.u-toyama.ac.jp)

Received: 27 March 2026 | **Revised:** 3 May 2026 | **Accepted:** 12 May 2026

Keywords: differential evolution | multi-strategy | population structure | wind farm layout optimization

ABSTRACT

Exploring clean energy alternatives to fossil fuels has become a major research focus worldwide. Wind energy is regarded as one of the most promising renewable energy sources due to its clean and sustainable nature. However, wake interactions among turbines significantly reduce power conversion efficiency, making wind farm layout optimization (WFLO) crucial for maximising energy output. As the number of turbines increases, wake interactions become more pronounced, further deteriorating overall efficiency. Metaheuristic algorithms have been widely adopted to address the complex constraints and design objectives of WFLO. Nevertheless, traditional heuristic methods often suffer from poor solution quality and premature convergence when dealing with large-scale WFLO problems under complex wind conditions. To overcome these limitations, this study proposes a multi-strategy synergy-based differential evolution algorithm (LSDE) for large-scale WFLO under complex wind scenarios. The proposed method is evaluated against nine representative WFLO algorithms under four complex wind scenarios (4, 5, 6 and 7 wind directions) and three turbine scales (30, 50 and 100 turbines). Experimental results demonstrate that LSDE achieves superior performance, stability and robustness. Specifically, LSDE improves power conversion efficiency by 3.40%, 3.51%, 3.10% and 3.36% under the four wind scenarios, respectively, compared with other mainstream algorithms.

1 | Introduction

Driven by continuous technological and societal advancements, electricity has become a fundamental component of modern life [1]. However, global power generation still relies heavily on fossil fuels, which aggravate environmental degradation and accelerate climate change [2]. As a result, the increasing frequency of natural disasters has caused severe ecological damage and substantial economic losses [3]. To address these challenges, many countries have introduced policies to reduce carbon emissions and promote renewable energy development [4]. Among various renewable energy sources, including hydropower, solar, tidal, geothermal and wind energy, wind power has attracted significant attention due to its high energy density, low transmission loss and relatively limited environmental impact. According to the International Energy Agency [5], wind power is expected to contribute at least 18% of global electricity generation by 2050, highlighting its importance in future energy systems.

In recent years, considerable efforts have been devoted to improving wind energy utilization, particularly by enhancing the efficiency of wind energy conversion systems. Existing studies have explored multiple aspects, including site selection [6], turbine modelling [7], wake modelling [8], wind speed forecasting [9] and wind farm layout optimization (WFLO) [10]. Among these, WFLO plays a critical role in maximising energy output. In

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

practice, the actual power generation of a wind farm is often significantly lower than its rated capacity. This is mainly due to the wake effect [11], which reduces the inflow wind speed of downstream turbines and decreases their power conversion efficiency. As wind farms scale up, the number of turbines increases substantially, leading to intensified wake interactions. These interactions introduce strong spatial coupling among turbines and result in a highly nonlinear and high-dimensional search landscape. Under complex multidirectional wind conditions, turbine placement decisions become strongly interdependent, further increasing the optimization difficulty.

Traditional exact methods, such as linear programming and mixed integer programming [12], often struggle with large-scale WFLO problems due to the exponential growth of combinatorial complexity. Consequently, intelligent optimization approaches have been widely adopted [13]. Metaheuristic algorithms (MHAs), including genetic algorithm (GA) [14], particle swarm optimization (PSO) [15], gravitational search algorithm (GSA) [16] and differential evolution (DE) [17], have demonstrated strong performance in solving complex optimization problems. However, these methods are mainly effective for small- and medium-scale wind farms under simpler wind conditions.

Despite these advances, research on large-scale WFLO under complex multidirectional wind scenarios remains limited. As the problem scale increases, wake-induced coupling among turbines significantly complicates the search landscape. Under such conditions, conventional MHAs often suffer from poor population coordination, diversity loss and premature convergence. In particular, classical DE relies on fixed mutation strategies and static population control mechanisms. These are insufficient to adapt to dynamically changing search environments caused by multidirectional wake interactions. Consequently, achieving an effective balance between global exploration and local exploitation becomes increasingly difficult. Therefore, developing structurally adaptive and high-performance optimization algorithms for large-scale WFLO under complex wind conditions remains a critical and open problem.

The main contributions are summarised as follows:

1. A multi-strategy synergy-based differential evolution algorithm (LSDE) is developed for large-scale WFLO under complex wind conditions.
2. A hybrid framework integrating hierarchical and distributed population structures with an adaptive switching mechanism is designed to improve population coordination and dynamically balance exploration and exploitation.
3. A nonlinear population size reduction strategy is incorporated to enhance adaptability in dynamic search environments and improve the algorithm's robustness in large-scale scenarios.
4. Extensive experiments under four complex wind scenarios demonstrate the superior performance and scalability of the proposed LSDE compared with nine representative algorithms.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 introduces the WFLO model and the Jensen wake model. Section 4 presents the proposed algorithm and its key strategies. Section 5 reports the experimental settings and results. Finally, Section 6 concludes the paper and discusses future research directions.

2 | Related Work

Wind farm layout optimization (WFLO) is a high-dimensional and computationally intensive problem defined over a complex search space. The presence of wake interactions among turbines significantly increases both the evaluation cost and the nonlinearity of the optimization landscape. This challenge becomes particularly severe in large-scale wind farms, where the number of turbines grows substantially, leading to a rapid increase in the computational burden. In this context, MHAs have demonstrated strong capability in handling complex and NP-hard optimization problems. For instance, directionally aligned differential evolution has been successfully applied to photovoltaic parameter estimation [18], while hybrid local search strategies have been employed to improve WEC layout performance [19]. In addition, many-objective evolutionary algorithms and swarm intelligence frameworks have achieved competitive results in domains such as protein structure prediction and UAV path planning [20, 21]. These studies collectively indicate the effectiveness of MHAs in complex real-world applications and highlight their potential for solving WFLO problems under limited evaluation budgets.

Existing WFLO research can be broadly categorised into discrete and continuous modelling approaches. In discrete formulations, turbines are restricted to predefined grid locations, where increasing grid resolution leads to higher solution accuracy at the cost of increased computational complexity [22]. In contrast, continuous formulations allow turbines to be placed anywhere within the feasible region, not only offering greater flexibility but also introducing higher sensitivity to initialisation and increased computational cost. Early work by Mosetti et al. [23] demonstrated the feasibility of applying GAs to WFLO using a grid-based representation. Since then, GAs have been extensively adopted due to their robustness and adaptability. Subsequent studies have focused on improving encoding strategies, objective formulations and wake modelling techniques. For example, matrix-based representations and improved objective functions have been proposed to enhance optimization performance [24], while more accurate wake models, such as Gaussian-based formulations and neural network-based models, have been developed to improve prediction accuracy [25, 26]. Despite these advances, more sophisticated wake models often incur significantly higher computational costs. In contrast, the Jensen wake model remains widely used due to its simplicity and computational efficiency, making it suitable for large-scale WFLO evaluation.

Beyond GA-based approaches, a variety of MHAs have been developed for WFLO. PSO and its variants have been enhanced with adaptive learning and local search mechanisms to improve convergence behaviour [27]. Reinforcement learning has also been introduced to improve parameter identification and enhance robustness in dynamic environments [28]. DE, known for its simplicity and strong global search capability, has attracted considerable attention. Several improved DE variants

have been proposed by incorporating memory mechanisms, adaptive parameter control and hybrid local search strategies [29, 30]. These approaches have demonstrated promising results in WFLO. However, their performance often deteriorates as problem scale and environmental complexity increase.

Despite the progress achieved, most existing studies primarily focus on small- to medium-scale wind farms under simplified wind conditions. The impact of large-scale deployment and complex, multidirectional wind scenarios remains insufficiently explored. As wind farm size increases, wake-induced interactions among turbines result in highly coupled and nonlinear search spaces, posing significant challenges to conventional optimization algorithms. Moreover, WFLO is inherently NP-hard and nonconvex, further increasing the difficulty of obtaining high-quality solutions. Therefore, developing scalable and adaptive optimization algorithms for large-scale WFLO under complex wind environments remains a critical research challenge.

To address these limitations, this study proposes a multi-strategy synergy-based differential evolution algorithm (LSDE). The proposed method integrates a hierarchical distributed population framework with an adaptive switching mechanism and a nonlinear population size reduction strategy, aiming to improve optimization performance and robustness in large-scale WFLO problems under complex wind conditions.

3 | Wind Farm Layout Optimization Problem

In this section, we present a detailed formulation of the WFLO problem, including the wake model, wind condition model, objective function and turbine configuration.

3.1 | Wake Effect Model

Although a variety of wake models have been developed in recent years, the Jensen wake model is adopted in this study due to its favourable balance between modelling fidelity and computational efficiency [31]. On the one hand, the Jensen model has been extensively utilised in the literature, enabling fair and consistent comparisons across different optimization algorithms. On the other hand, its simpler mathematical formulation significantly reduces computational overhead, making it particularly suitable for large-scale WFLO problems [32].

The Jensen model characterises the reduction of wind speed caused by upstream turbines and its impact on downstream turbines, thereby introducing a nonlinear dependency between turbine layout and overall power output. As depicted in Figure 1, the wake generated by Turbine 1 propagates downstream and alters the inflow conditions of Turbine 2. The resulting wake interaction is governed by the relative turbine positions, local wind speeds and wake expansion behaviour.

Based on the conservation of momentum, the wind speed distribution within the wake can be expressed as:

$$\pi r^2 v_1 + \pi\left(R^2 - r^2\right)v_0 = \pi R^2 v_1', \tag{1}$$

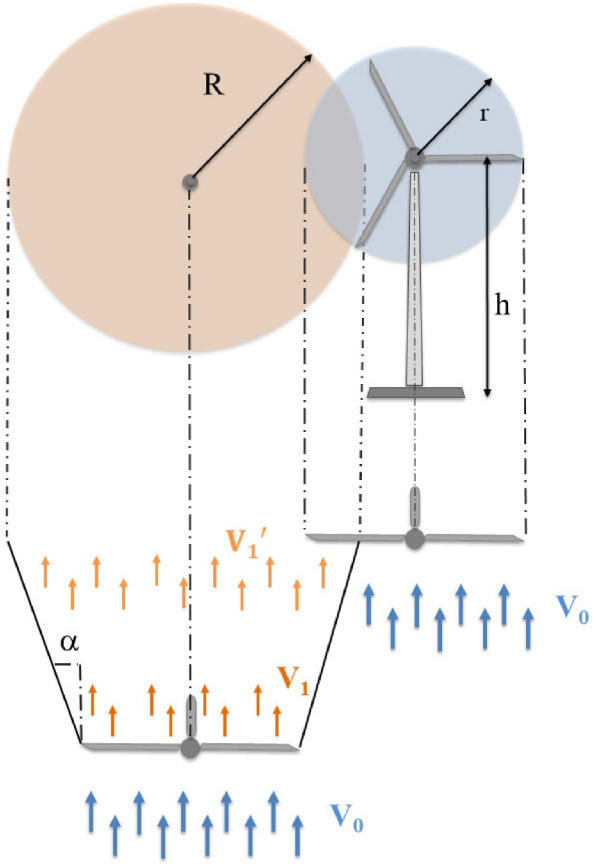


FIGURE 1 | Schematic illustration of the wake effect model, showing velocity deficit and wake expansion behind a wind turbine.

where r denotes the rotor radius, R represents the wake radius, v_0 is the free-stream wind speed, v_1 is the wind speed immediately downstream of the turbine and v_1' is the wind speed at the wake boundary. Following standard assumptions in the literature, v_1 is approximated as $\frac{1}{3}v_0$, leading to

$$v_1' = v_0\left[1 - \frac{2}{3}\left(\frac{r}{R}\right)^2\right]. \tag{2}$$

In the Jensen framework, the wake expands linearly in a conical manner. Accordingly, the wake radius R increases with the downstream distance and is given by

$$R = r + \alpha d, \tag{3}$$

where d is the distance between turbines and α is the wake expansion coefficient. The coefficient α depends on atmospheric conditions and can be calculated as:

$$\alpha = \frac{1}{2}\left[\ln\left(\frac{h}{z}\right)\right]^{-1}, \tag{4}$$

where h denotes the hub height and z represents the surface roughness parameter.

In practical wind farms, a turbine is typically influenced by multiple upstream wakes. The combined wake effect on the j-th turbine is modelled using a superposition approach, where the effective wind speed is expressed as:

$$v_j = v_0\left[1 - \sqrt{\sum_i^N \left(1 - \frac{v_{ij}}{v_0}\right)^2 \left(\frac{S_{i\to j}}{S_j}\right)^2}\right]. \tag{5}$$

Here, S_i denotes the cross-sectional area of the wake generated by turbine i, S_j is the rotor swept area of turbine j, and $S_{i\to j}$ represents the overlapping area between the wake region of turbine i and the rotor area of turbine j.

The overlap area $S_{i\to j}$ depends on the relative distance between turbines and can be categorised into three cases, as illustrated in Figure 2:

$$S_{i\to j} = \begin{cases} \pi r^2, & d_{i\to j} \le R - r, \\ 0, & d_{i\to j} \ge R + r, \end{cases} \tag{6}$$

where $d_{i\to j}$ denotes the distance between the centres of the wake and rotor regions.

For the partial overlap case $(|R - r| < d_{i\to j} < R + r)$, the intersection area is computed geometrically as

$$S_{i\to j} = S_{\overset{\frown}{AO_1B}} + S_{\overset{\frown}{AO_2B}} - S_{\overline{AO_1O_2B}}, \tag{7}$$

where $S_{\overset{\frown}{AO_1B}}$ and $S_{\overset{\frown}{AO_2B}}$ denote the areas of two circular sectors, and $S_{\overline{AO_1O_2B}}$ corresponds to the area of the overlapping quadrilateral.

The sector areas are given by

$$\begin{aligned} S_{\overset{\frown}{AO_1B}} &= R^2 \arccos\left(\frac{d^2 + R^2 - r^2}{2Rd}\right), \\ S_{\overset{\frown}{AO_2B}} &= r^2 \arccos\left(\frac{d^2 + r^2 - R^2}{2rd}\right), \end{aligned} \quad d = d_{i\to j}. \tag{8}$$

The quadrilateral AO_1O_2B can be decomposed into two congruent triangles. By applying Heron's formula, its area can be expressed as:

$$S_{\overline{AO_1O_2B}} = \frac{1}{2}\sqrt{\left((R + r)^2 - d_{i\to j}^2\right)\left(d_{i\to j}^2 - (R - r)^2\right)}, \quad |R - r| < d_{i\to j} < R + r. \tag{9}$$

3.2 | Wind Condition Model

Accurate modelling of wind conditions is essential for evaluating the practical effectiveness and robustness of WFLO algorithms. To better approximate real-world environments, this study refines wind condition modelling by incorporating more realistic representations of both wind direction and wind speed. In terms of wind direction, conventional studies often assume simplified patterns such as unidirectional, bidirectional or symmetric distributions. However, real wind fields are influenced by seasonal variations and terrain effects, resulting in irregular and highly nonuniform directional characteristics. To capture these properties, the proposed model considers multiple wind direction sectors (e.g., 4–7 sectors) with asymmetric and biased probability distributions, enabling the constructed scenarios to reflect diverse atmospheric conditions more realistically.

For wind speed modelling, the Weibull distribution is employed due to its capability to characterise the skewed and heavy-tailed nature of wind speed data. Based on empirical studies and field observations [33], the wind speed range is defined within 6–20 m/s, and the distribution parameters are calibrated such that the expected wind speed is approximately 13 m/s. It should be noted that this value represents a statistical expectation rather than a fixed input. To generate representative wind data, 100,000 samples are drawn from the Weibull distribution, forming a dataset that captures both natural variability and realistic statistical characteristics. Compared with models assuming constant wind speed or normal distributions, this approach more accurately reflects the stochastic behaviour of wind.

To further enhance realism, publicly available meteorological data are incorporated to adjust both wind direction and speed distributions, ensuring that their joint distribution is consistent with regional characteristics. Based on this strategy, four representative wind scenarios (WS1–WS4) are constructed, corresponding to 4, 5, 6 and 7 nonuniformly distributed wind directions, respectively. As illustrated in Figure 3, each wind rose diagram represents the probabilistic structure of wind conditions, where the radial length indicates directional occurrence probability and the colour encodes wind speed magnitude. It can be observed that all scenarios exhibit clear directional bias, with certain prevailing directions dominating the distribution. In addition, wind speeds within each sector are generated through

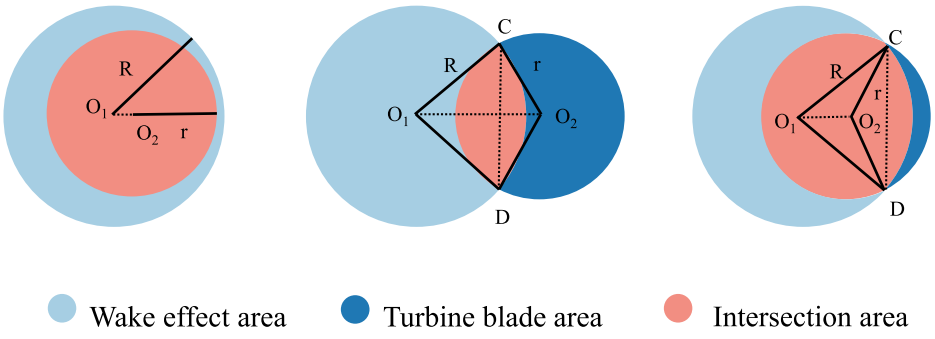
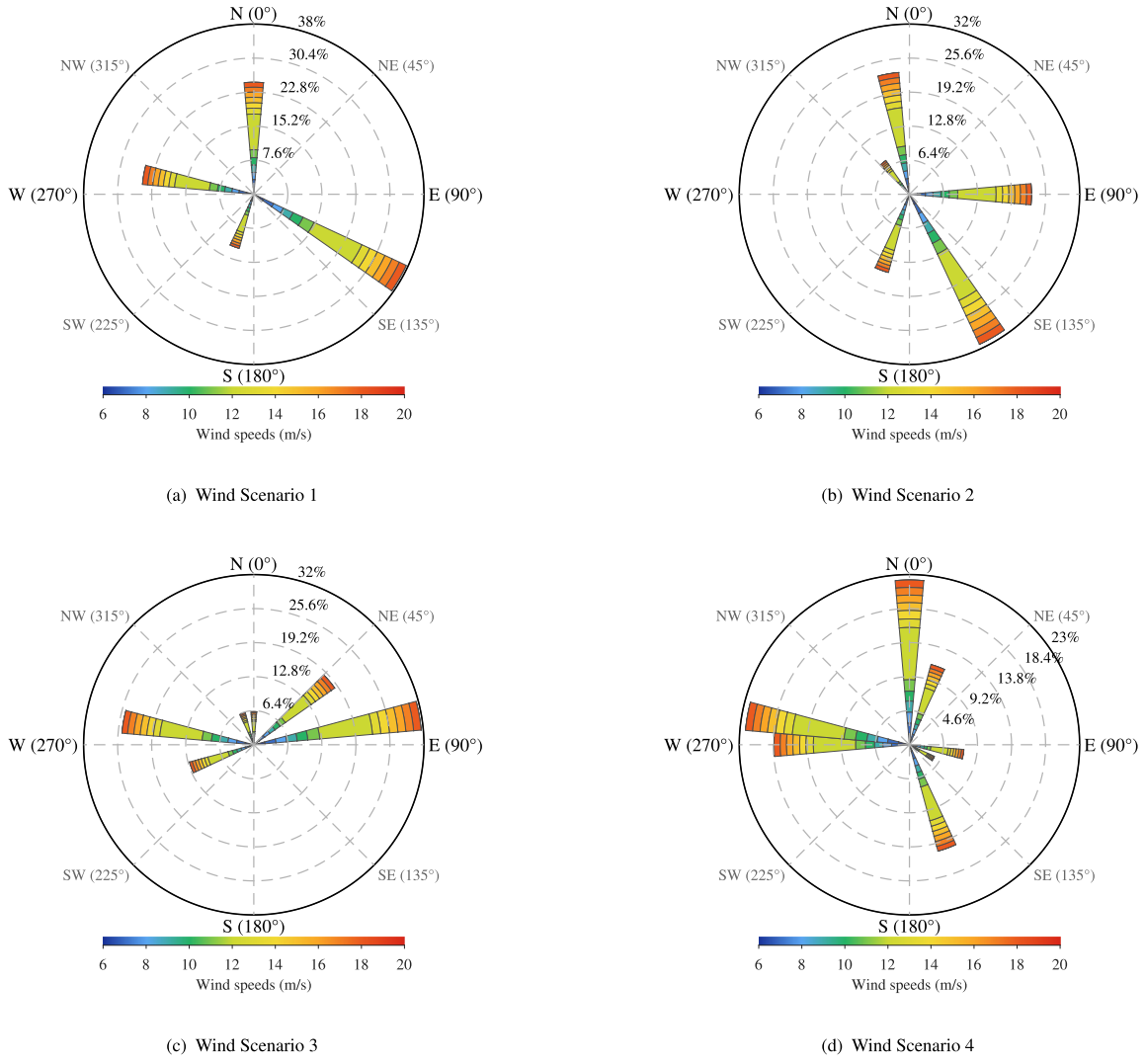


FIGURE 2 | Schematic illustration of the overlap between the wake effect region and the turbine rotor area for two turbines.



(a) Wind Scenario 1

(b) Wind Scenario 2

(c) Wind Scenario 3

(d) Wind Scenario 4

FIGURE 3 | Wind rose diagrams illustrating wind direction frequency and wind speed distributions for the four wind scenarios (WS1–WS4).

nonuniform sampling governed by the Weibull distribution rather than uniform assignment. This modelling strategy avoids the oversimplified assumptions commonly adopted in previous studies and results in a more complex and realistic optimization landscape, thereby providing a more rigorous basis for evaluating WFLO algorithms under practical wind conditions.

3.3 | Optimization Objective

The WFLO problem aims to identify an optimal spatial configuration of wind turbines that enhances overall power generation while mitigating wake-induced energy losses under given wind conditions. Although some studies formulate WFLO as a multi-objective problem by incorporating construction costs [34], this study focuses on energy performance and assumes a fixed number of turbines with comparable installation costs. Accordingly, the WFLO problem is formulated as a single-objective optimization task [22], where the objective function is defined as:

$$O = \max P(N, \Lambda, \varphi) = \sum_{i=1}^{N} \sum_{v} \sum_{\varphi} p(v, \varphi) P_i(v, \varphi, \Lambda), \tag{10}$$

where N denotes the number of wind turbines and Λ represents the layout configuration of the wind farm. The variables v and φ correspond to wind speed and wind direction, respectively. The term $p(v, \varphi)$ is the joint probability distribution of wind speed and direction, while $P_i(v, \varphi, \Lambda)$ denotes the power output of the i-th turbine under layout Λ for a given wind condition (v, φ).

To further evaluate the quality of a layout, the conversion efficiency of the wind farm is introduced as a normalised performance metric, defined as:

$$\eta = \frac{P(N, \Lambda, \varphi)}{N \sum_{v} \sum_{\varphi} p(v, \varphi) P_r(v, \varphi)}, \tag{11}$$

where η denotes the conversion efficiency, and $P_r(v, \varphi)$ represents the ideal power output of a turbine under wind condition (v, φ) without considering wake interactions. This metric reflects the proportion of energy effectively captured by the wind farm relative to its theoretical maximum, where a higher value of η indicates a more efficient layout with reduced wake losses.

3.4 | Turbine Configuration and the Power Output Model

The number of turbines deployed in a wind farm plays a crucial role in determining overall energy generation efficiency. To evaluate the performance of the proposed optimization algorithm under different layout densities, three representative turbine scales are considered, that is, $N = 30$, 50 and 100.

The power output of each turbine is modelled based on the GE1.5sle wind turbine power curve [35]. This model characterises the relationship between wind speed and turbine output through three key operating regions defined by the cut-in, rated, and cut-out wind speeds, which are set to 2 m/s, 12.8 m/s and 18 m/s, respectively. Specifically, no power is generated when the wind speed is below the cut-in threshold or exceeds the cut-out limit. Within the intermediate region, the output power increases with wind speed following a cubic relationship and reaches a constant rated value once the wind speed exceeds the rated threshold. This piecewise formulation provides a realistic approximation of turbine operating behaviour while maintaining computational efficiency.

Accordingly, the turbine power output $P(v)$ is defined as:

$$P(v) = \begin{cases} 0 \text{ kW}, & \text{if } v < 2\,\text{m/s or } v > 18\,\text{m/s}, \\ 0.3v^3 \text{ kW}, & \text{if } 2\,\text{m/s} \le v < 12.8\,\text{m/s}, \\ 629.1 \text{ kW}, & \text{if } 12.8\,\text{m/s} \le v \le 18\,\text{m/s}. \end{cases} \tag{12}$$

4 | Optimization Algorithm for WFLO

4.1 | Motivation

Since the development of evolutionary algorithms, the interplay between exploration and exploitation has been recognized as a fundamental factor governing the search behaviour of optimization processes [36]. Exploitation focuses on refining candidate solutions in the vicinity of promising regions to improve solution quality, whereas exploration aims to investigate unexplored areas of the search space to maintain population diversity and avoid local optima [37]. Achieving an effective balance between these two components remains challenging: Excessive exploitation may result in premature convergence, while excessive exploration can slow convergence and reduce optimization efficiency.

This issue is particularly critical for WFLO. In WFLO, turbine placement decisions are strongly coupled through wake interactions, meaning that the contribution of one turbine depends not only on its own position but also on the spatial configuration of other turbines. As the number of turbines increases, the dimensionality and combinatorial complexity of the search space grow rapidly. Moreover, under complex wind scenarios with multiple nonuniform wind directions and distinct wind speed distributions, the fitness landscape becomes highly nonlinear, rugged and multimodal. These characteristics make WFLO prone to local optima and require an optimiser to maintain sufficient diversity while still exploiting promising layout patterns.

Recent studies indicate that DE tends to favour exploitation due to its inherent design [38]. As reported in [39], DE variants that overly emphasise exploitation often fail to deliver competitive performance in complex optimization tasks. Although advanced DE variants, such as SHADE, L-SHADE and jSO, have significantly improved parameter adaptation and convergence performance, they largely inherit the core mutation and selection mechanisms of classical DE. As a result, their search behaviour still relies on population-level stochastic differences, without explicitly modelling structured information exchange among individuals. For large-scale WFLO problems, this limitation becomes more pronounced. The strong spatial coupling among turbines and the high-dimensional decision space require coordinated search across multiple regions. However, most existing DE variants adopt a single-population structure with implicit interaction patterns, which restricts their ability to maintain diversity and coordinate global exploration under complex wind conditions. In addition, DE typically employs a greedy selection strategy (e.g., one-to-one selection), which gradually reduces population diversity as evolution progresses [37]. Although this mechanism accelerates convergence in simple problems, it may lead to diversity loss in complex landscapes. As a result, individuals tend to repeatedly exploit a limited set of elite solutions, increasing the risk of premature convergence and stagnation. Once a large portion of the population becomes trapped in local optima, the effectiveness of information exchange is significantly weakened, which limits the algorithm's global search capability, especially in large-scale WFLO scenarios with strong wake interactions.

Therefore, improving DE for WFLO requires not only adaptive parameter control but also a search structure capable of regulating how information is exchanged among individuals and subpopulations. Motivated by this observation, this study proposes a layered and distributed DE framework, termed LSDE, to enhance the coordination between exploration and exploitation. The distributed population structure is introduced to maintain multiple search regions and reduce the risk of population-wide stagnation, which is particularly important for handling the multimodal characteristics of WFLO. The hierarchical interaction mechanism is designed to guide inferior individuals using information from better performing groups while avoiding excessive convergence towards a single elite region, thereby improving search coordination under strong wake coupling. Furthermore, the nonlinear population reduction strategy gradually adjusts the search scale during evolution, enabling broad exploration in the early stage and refined exploitation in the later stage. Through these designs, LSDE aims to address the structural challenges of large-scale WFLO, including strong wake-induced coupling, high-dimensional layout decisions, rugged fitness landscapes and diversity loss under complex wind conditions. Rather than simply combining existing operators, the proposed framework restructures the information exchange process of DE to better match the optimization characteristics of WFLO.

4.2 | Success-History Based Parameter Adaptation for Differential Evolution (SHADE)

SHADE [40] is a widely recognized differential evolution algorithm that achieved outstanding performance on the IEEE CEC2013 benchmark suite and has been regarded as a state-of-the-art method. It incorporates a success history–based

parameter adaptation mechanism to dynamically adjust the control parameters F and CR. In addition, SHADE introduces the concept of the *pbest* individual into the mutation strategy. This mechanism guides the search towards high-quality solutions. At the same time, it maintains a balance between exploration and exploitation. These design principles have significantly influenced subsequent DE variants, such as L-SHADE [41], jSO [42] and LSHADE-EpSin. Despite their improved performance, many of these algorithms retain the core parameter adaptation and mutation framework of SHADE. Therefore, SHADE is adopted as the baseline algorithm for further enhancement. A brief description of SHADE is provided as follows.

SHADE generates offspring through a sequence of evolutionary operations, including initialisation, mutation, crossover and selection. The initial population is defined as:

$$x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,D}\}, \quad i \in \{1, 2, ..., N\}. \tag{13}$$

The mutation operation follows the current-to-pbest/1 strategy:

$$v_i = x_i + F \cdot \left[\left(x_{pbest} - x_i\right) + \left(x_{R1} - x_{R2}\right)\right], \tag{14}$$

where x_{R1} and x_{R2} are randomly selected individuals from the current population, and x_{pbest} is randomly chosen from the top $p\%$ individuals according to fitness values,

$$p_i = \text{rand}\left(p_{\min},\ 0.2\right). \tag{15}$$

Here, p controls the size of the elite subset, with its upper bound set to 0.2 and lower bound defined as $2/N$.

Subsequently, binomial crossover is applied between the mutant vector v_i and the target vector x_i to generate the trial vector u_i:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if rand}(0,1) < CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}, & \text{otherwise}, \end{cases} \tag{16}$$

where $j \in [1, D]$ denotes the j-th dimension and j_{rand} is a randomly selected index.

Finally, SHADE employs a greedy selection mechanism to update the population.

$$x_i^{t+1} = \begin{cases} u_i, & \text{if } fit(u_i) < fit\left(x_i^t\right), \\ x_i^t, & \text{otherwise}, \end{cases} \tag{17}$$

where $fit(\cdot)$ denotes the objective function value.

4.3 | Proposed LSDE Framework

Inspired by SHADE, this study proposes a differential evolution algorithm that integrates an adaptive switching population structure with a nonlinear decay mechanism, aiming to address large-scale WFLO problems under complex wind conditions.

4.3.1 | *Population Structure for Adaptive Switching*

In this study, an adaptive switching population structure is developed to improve the balance between exploration and exploitation in SHADE. Conventional DE algorithms tend to emphasise exploitation, which often leads to premature convergence and limited ability to escape local optima. To overcome this limitation, the proposed framework introduces a dynamically adjustable population structure that adapts to the current search state. This design preserves the strong exploitation capability of SHADE while enhancing global exploration. The overall architecture is illustrated in Figure 4. To better illustrate the proposed framework, Figure 4 presents its overall structure. It consists of three components: (a) solution space decomposition, (b) hierarchical structure and (c) distributed subpopulation interaction.

As shown in Figure 4a, the population is initialised in the solution space and partitioned into multiple subpopulations using a distance-based clustering strategy. Specifically, individuals are grouped according to their spatial proximity in the decision space, resulting in multiple approximately equal-sized subpopulations with distinct spatial distributions. The mathematical definition of this population decomposition is given as follows. Formally, the population at generation t is defined as:

$$\mathcal{P}^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ..., x_N^{(t)}\}. \tag{18}$$

A distance-based clustering method is employed to partition the population into K disjoint subsets [43] as follows:

$$\{\mathcal{P}_1^{(t)}, \mathcal{P}_2^{(t)}, ..., \mathcal{P}_K^{(t)}\} = \mathcal{C}\left(\mathcal{P}^{(t)}, K\right), \tag{19}$$

subject to

$$\mathcal{P}_i^{(t)} \cap \mathcal{P}_j^{(t)} = \varnothing, \quad i \neq j, \qquad \bigcup_{i=1}^{K} \mathcal{P}_i^{(t)} = \mathcal{P}^{(t)}, \tag{20}$$

where individuals are assigned to subpopulations based on their spatial proximity in the decision space, and each cluster is directly treated as one subpopulation. In this study, the number of subpopulations is fixed to three.

This structure enables parallel exploration of multiple regions in the search space without enforcing rigid boundaries. Each subpopulation performs local search independently, while inter-subpopulation interactions are introduced to maintain diversity and support global exploration.

As illustrated in Figure 4b, each subpopulation adopts a three-layer hierarchical structure to enhance exploitation capability. This design allows individuals with different fitness levels to play distinct roles during the search process. The details are as follows:

- **Top layer:** This layer contains the top 25% of individuals with the best fitness values, that is, $\lfloor \frac{N}{4} \rfloor$ individuals. These elite individuals guide the search direction of lower layers, thereby accelerating convergence towards promising regions as follows:

$$x_t^{new} = x_t + F \cdot \left[\left(x_{pbest} - x_t\right) + \left(x_{Mr1} - x_{Mr2}\right)\right], \tag{21}$$

where x_t denotes a Top-layer individual, and x_{Mr1}, x_{Mr2} are randomly selected from the Middle layer.

- **Middle layer:** This layer includes individuals ranked within the 26%–50% percentile. These promising individuals guide the Bottom layer and enhance convergence as follows:

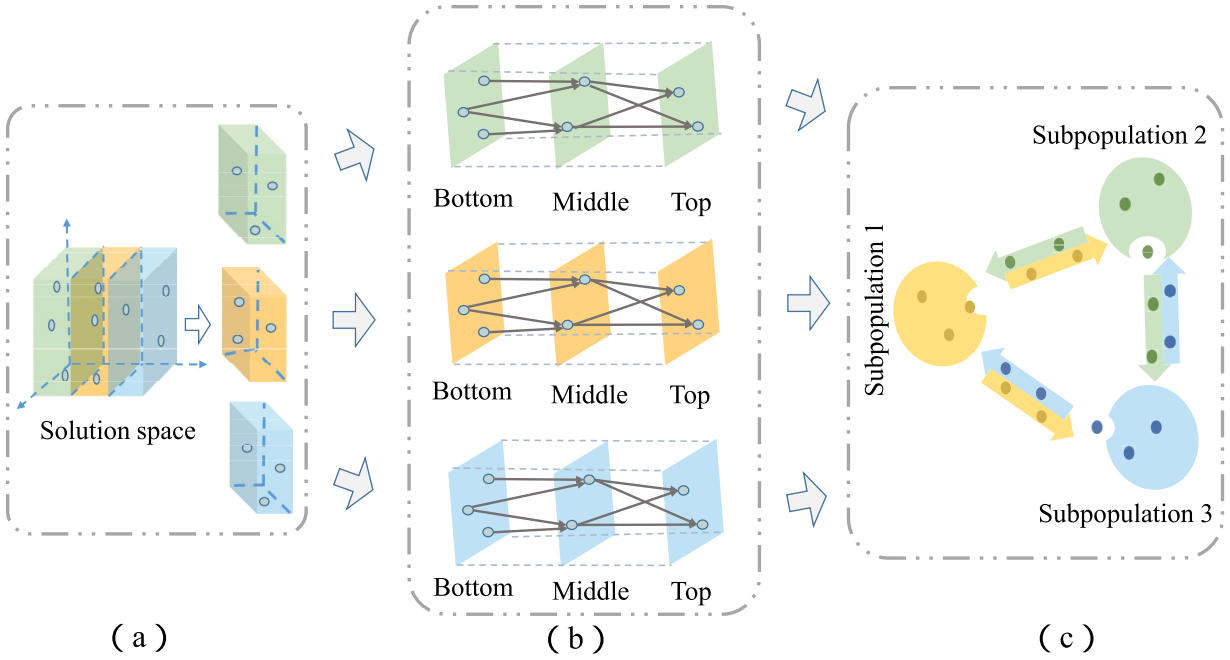


FIGURE 4 | Overview of the proposed adaptive switching population framework: (a) solution space representation and population decomposition, (b) three-layer hierarchical structure and (c) distributed subpopulation interaction mechanism.

$$x_m^{new} = x_m + F \cdot \left[\left(x_{pbest} - x_m \right) + \left(x_{Br1} - x_{Br2} \right) \right], \tag{22}$$

where x_{Br1}, x_{Br2} are selected from the Bottom layer to enhance diversity.

- **Bottom layer:** This layer contains the remaining 50% of individuals and primarily maintains diversity. These individuals are guided by the Middle layer and gradually converge towards promising regions.

As shown in Figure 4c, the framework adopts a distributed structure with K subpopulations $\{\mathcal{P}_1, ..., \mathcal{P}_K\}$. Each subpopulation performs semi-independent search while interacting with others. To avoid stagnation, a restart mechanism is introduced. When two subpopulations $\mathcal{P}_i$ and $\mathcal{P}_j$ become stagnant, they are partially reinitialised. The new individuals are sampled from another subpopulation $\mathcal{P}_k$, where $k \in \{1, ..., K\}\{i, j\}$. This mechanism restores diversity and enables exploration of new regions. The stagnation indicator is defined as:

$$\delta_i^{(t)} = \begin{cases} 1, & \text{if no individual in } \mathcal{P}_i \text{ is updated at generation } t, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

When $\delta_i^{(t)} = \delta_j^{(t)} = 1$, the restart mechanism is triggered as follows:

$$\begin{aligned} \mathcal{P}_i^{(t)} &= \left(\mathcal{P}_i^{(t)} \backslash \text{Worst}_{10\%}\left(\mathcal{P}_i^{(t)} \right) \right) \cup \text{Sample}_{10\%}\left(\mathcal{P}_k^{(t)} \right), \\ \mathcal{P}_j^{(t)} &= \left(\mathcal{P}_j^{(t)} \backslash \text{Worst}_{10\%}\left(\mathcal{P}_j^{(t)} \right) \right) \cup \text{Sample}_{10\%}\left(\mathcal{P}_k^{(t)} \right). \end{aligned} \tag{24}$$

This process maintains population size. At the same time, it introduces new genetic material. The replacement ratio of 10% provides a balance between diversity and convergence stability. In summary, the proposed framework combines a hierarchical structure with distributed subpopulations. The restart mechanism enhances information exchange, while the layered design improves exploitation. Together, these components achieve an effective balance between exploration and exploitation.

4.3.2 | *Nonlinear Population Reduction Strategy*

Although the proposed population structure improves search efficiency, its performance can be further enhanced by adaptive population size control. Linear population reduction has been widely used to balance exploration and exploitation by gradually decreasing population size during evolution [41, 42].

However, linear reduction may not be sufficiently flexible for complex, high-dimensional problems [44]. To address this limitation, a nonlinear population reduction strategy is introduced, in which a cosine annealing scheme is employed to control the population size dynamically as follows:

$$N_{\text{Popu}}(n) = N_{\min} + \frac{N_{\max} - N_{\min}}{2} \left(1 + \cos\left(\frac{\pi \, nFES}{FES_{\max}} \right) \right), \tag{25}$$

where $N_{\max}$ and $N_{\min}$ denote the maximum and minimum population sizes, respectively. In this study, $N_{\min} = 0.3D$ to preserve sufficient diversity in later stages. The variable $nFES$ represents the current number of function evaluations, and $FES_{\max}$ is the maximum evaluation budget. This nonlinear scheduling provides a smooth transition from global exploration to local exploitation, thereby improving adaptability and robustness in high-dimensional WFLO problems (Algorithm 1).

4.4 | Framework of LSDE

The overall workflow of LSDE is outlined in Algorithm 1. The framework integrates multiple coordinated components to enhance optimization performance under complex wind conditions. During initialisation, wind information, including wind speed and direction, is first loaded, followed by the generation and evaluation of the initial population using the objective function. In the subsequent hierarchical evolution stage, individuals are ranked according to their fitness values and partitioned into three layers, namely the Top, Middle, and Bottom layers. The Middle and Bottom layers are updated under the guidance of superior layers, enabling structured information propagation and improving search efficiency.

ALGORITHM 1 | Pseudo-code of LSDE

Input: Wind speed, wind direction, Number of turbines T, Population size N, Number of subpopulations K, and Maximum function evaluations $FES_{\max}$
Output: Optimal layout $\mathbf{x}_{best}$ and its corresponding conversion efficiency
Initialise population $\mathbf{x}_i \in \mathbb{R}^T$ $(i = 1, 2, ..., N)$
Divide the population into K subpopulations $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_K\}$
Evaluate initial fitness using Equation (10)
Compute the global best solution $\mathbf{x}_{Pbest}$
while $FES < FES_{\max}$ **do**
    /* Hierarchical structuring */
    Rank individuals by fitness and assign them into Top (25%), Middle (26%–50%), and Bottom (50%) layers
    /* Layer-guided evolution */
    Update the Middle layer using guidance from the Top layer according to Equation (21)
    Update the Bottom layer using guidance from the Middle layer according to Equation (22)
    /* SHADE-based evolutionary update */
    Generate adaptive parameters F_i and CR_i
    Perform mutation according to Equation (14) to obtain the mutant vector $\mathbf{v}_i$
    Apply crossover using CR_i to generate the trial vector $\mathbf{u}_i$
    Update individual $\mathbf{x}_i$ based on the selection rule in Equation (17)
    /* Distributed structure and subpopulation communication */
    Compute the stagnation indicator $\delta_k^{(t)}$ (0 or 1) for each subpopulation $\mathcal{P}_k$ using Equation (23)
    if there exist $p \neq q$ such that $\delta_p^{(t)} = \delta_q^{(t)} = 1$ **then**
    Select a source subpopulation $\mathcal{P}_k$, where $k \in \{1, ..., K\}\{p, q\}$
    Update $\mathcal{P}_p$ and $\mathcal{P}_q$ by removing the worst 10% individuals and injecting 10% samples from $\mathcal{P}_k$
    end if
    /* Nonlinear population size reduction */
    Compute the target population size $N_{\text{Popu}}(n)$ using Equation (25)
    Adjust the population to $N_{\text{Popu}}(n)$ by removing the worst individuals
end while
return best layout and conversion efficiency

To enhance robustness, a restart mechanism is incorporated. To detect stagnation, a binary stagnation indicator $\delta_k^{(t)}$ is calculated for each subpopulation $\mathcal{P}_k$ according to Equation (23), where $\delta_k^{(t)} = 1$ indicates that the k-th subpopulation fails to improve at iteration t, and $\delta_k^{(t)} = 0$ otherwise. If two subpopulations $\mathcal{P}_p$ and $\mathcal{P}_q$ satisfy $\delta_p^{(t)} = \delta_q^{(t)} = 1$, the restart mechanism is triggered. Specifically, the worst performing 10% individuals in the stagnant subpopulations are removed and replaced with 10% samples drawn from another nonstagnant subpopulation $\mathcal{P}_k$, where $k \in \{1, ..., K\}\{p, q\}$. This process maintains the population size while introducing new genetic material. It effectively alleviates stagnation, maintains diversity and reduces the likelihood of convergence to local optima.

Furthermore, a nonlinear population reduction strategy is employed. Guided by a cosine scheduling scheme, the population size is dynamically adjusted throughout the search process, progressively eliminating low-quality individuals. This strategy supports a smooth transition from global exploration in the early stage to refined local exploitation in the later stage. These components, including hierarchical updating, stagnation-triggered restart and nonlinear population reduction, are iteratively executed until the maximum number of function evaluations is reached.

4.5 | Overall Implementation

To provide a clear overview of the LSDE algorithm, its complete workflow is illustrated in Figure 5. The process begins with loading wind data, followed by population initialisation and fitness evaluation based on Equation (10). According to their fitness values, individuals are ranked and assigned to three hierarchical layers (Top, Middle and Bottom), which form the basis of the layer-guided evolutionary process. Individuals in lower layers are updated under the guidance of upper layers using Equations (21) and (22), while the SHADE mechanism further refines the population through adaptive mutation and crossover operations defined in Equations (14) and (17). This hybrid design allows LSDE to effectively combine hierarchical structural guidance with adaptive parameter control.

To enhance robustness and alleviate stagnation, LSDE adopts a distributed structure to enable information exchange among subpopulations, as described in Equation (23). When two subpopulations fail to produce improvements simultaneously, low-quality individuals are replaced with samples from an active subpopulation, thereby restoring diversity and reducing the risk of premature convergence. Furthermore, a nonlinear population size reduction strategy, defined in Equation (25), is employed to dynamically adjust the population size, promoting extensive exploration in the early stages and progressively refined exploitation as the search evolves.

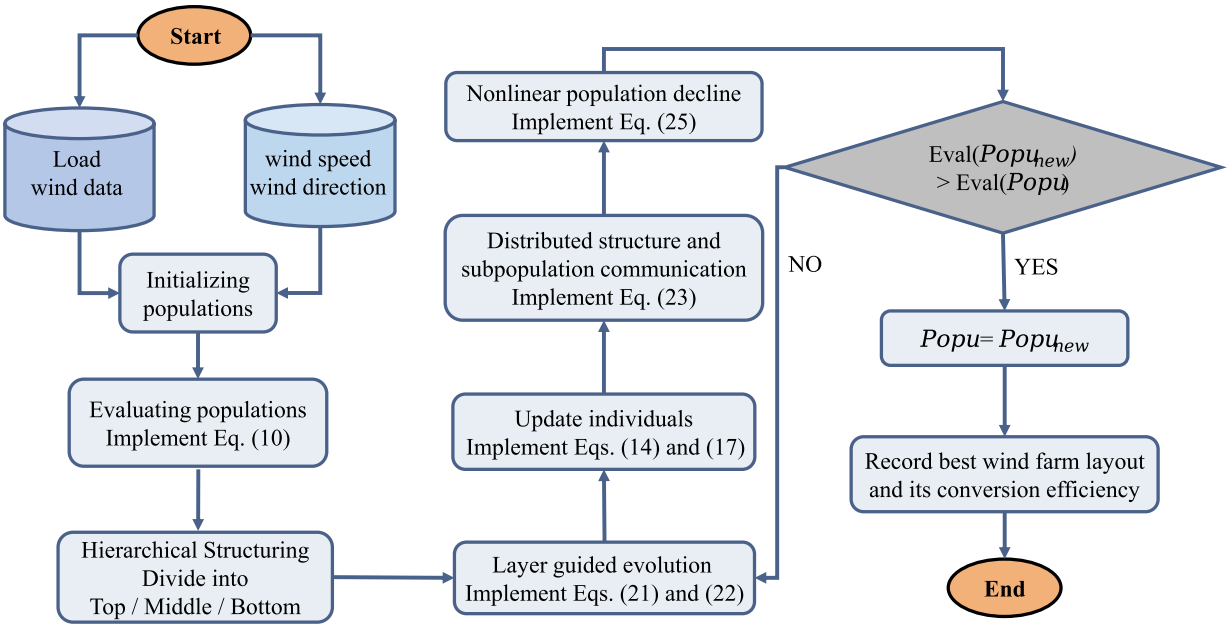


FIGURE 5 | Flowchart of the proposed LSDE algorithm for wind farm layout optimization, including hierarchical structuring, distributed subpopulation interaction and adaptive population control.

At each iteration, the newly generated population is re-evaluated and compared with the current population. If improved solutions are identified, both the population and the corresponding wind farm layout, together with the conversion efficiency, are updated. This iterative procedure continues until the maximum number of function evaluations is reached, ultimately yielding the optimised wind farm configuration.

4.6 | Computational Complexity Analysis

To quantitatively assess the computational efficiency of the proposed LSDE algorithm for the WFLO problem, its time complexity is analysed. Let N denote the population size, D the problem dimension (i.e., the number of turbines), W the number of wind scenarios (combinations of wind direction and wind speed), G the maximum number of iterations and K the number of subpopulations (with $K = 3$ in this study). The computational cost per generation and over the entire evolution process is detailed as follows:

1. Initialisation and encoding: Generating N individuals in a D dimensional space requires $O(ND)$. Because this step is performed only once, its cost is negligible compared with the iterative evaluation process.
2. Hierarchical partitioning and sorting: Each subpopulation of size N/K is sorted according to fitness and divided into three-layers. The sorting cost is

$$K \cdot O\left(\frac{N}{K}\log\frac{N}{K}\right) = O(N\log(N/K)), \tag{26}$$

while the layer assignment requires $O(N)$.

3. Mutation, crossover and selection: These vector-based operations scale linearly with the population size and problem dimension, resulting in a cost of $O(ND)$ per generation.
4. Inter-subpopulation information exchange: The exchange of individuals among subpopulations introduces a linear cost of $O(N)$. Any subsequent reordering is already included in Step (2).
5. Nonlinear population reduction: The cosine-based scheduling function used to update the population size incurs a constant cost of $O(1)$ per generation.
6. Fitness evaluation (wake and power computation): For a layout with D turbines, wake interaction evaluation requires $O(D^2)$. Considering N individuals evaluated over W wind scenarios, the total cost per generation is $O(NWD^2)$, which dominates the overall computational expense.

In summary, the computational complexity of LSDE is dominated by the fitness evaluation stage, resulting in a per-generation complexity of $O(NWD^2)$ and an overall complexity of $O(GNWD^2)$ over G iterations. Other operations, including hierarchical partitioning, inter-subpopulation communication and nonlinear population reduction, introduce only minor linear or logarithmic overhead. Based on this analysis, the computational cost of LSDE is primarily determined by the wake and power evaluation, and its scalability is governed by the problem dimension and the number of wind scenarios.

5 | Numerical Experimental Results

This section presents the experimental set-up and corresponding results. The proposed LSDE algorithm is evaluated against nine representative state-of-the-art algorithms, including AGA [45], SUGGA [46], AGPSO [22], CGPSO [27], ISE [39], CJADE [17], SHADE [40], BDE [47] and CLSHADE [29]. The evaluation is conducted under four complex wind scenarios (4, 5, 6, and 7 wind directions) and three turbine-scale settings (30, 50 and 100 turbines).

The compared methods cover several major categories of metaheuristic algorithms. AGA and SUGGA are classical genetic algorithm-based approaches specifically designed for WFLO

problems. AGPSO and CGPSO represent particle swarm optimization–based methods with enhanced learning mechanisms. CLSHADE and BDE are advanced differential evolution variants tailored for WFLO, while ESE is a representative spherical evolutionary algorithm. In addition, two high-performing DE-based algorithms from benchmark studies, namely SHADE and CJADE, are included to ensure a comprehensive comparison.

A summary of the compared algorithms, including their publication sources and key characteristics, is provided in Table 1. To ensure a fair and consistent evaluation, all algorithms are tested under identical wind conditions and the same experimental settings. In addition, the detailed parameter configurations of all compared algorithms are presented in Table 2 to ensure reproducibility.

5.1 | Experimental Setup

To evaluate the performance of the proposed algorithm on the WFLO problem, four complex wind scenarios are considered: WS1–WS4, corresponding to 4, 5, 6 and 7 nonuniform wind directions, respectively, each associated with distinct wind speed distributions. The corresponding wind rose diagrams are shown in Figure 3.

Because the problem complexity increases with the number of turbines, three turbine-scale settings are adopted, namely 30, 50 and 100 turbines, to assess the robustness and scalability of the proposed method. All experiments are conducted on a standardized 15 × 15 grid-based wind farm layout. Each grid cell has a width of 231 m, resulting in a total area of approximately 3.47 km × 3.47 km. To ensure a fair comparison, the maximum number of fitness evaluations (FES) is uniformly set to 20,000 for all algorithms. Each experiment is independently repeated 51 times to reduce stochastic variability. All simulations are performed under identical computational conditions using MATLAB R2024a on Windows 11, executed on an Intel Core i7-11700K processor (2.50 GHz, 8 cores and 16 threads) with 32 GB RAM.

The performance of each algorithm is evaluated using power conversion efficiency, defined as the ratio between the actual total power output and the corresponding ideal output without wake interactions. In addition, four statistical indicators (MAX, MIN, MEAN and STD) are reported to reflect performance stability. Statistical significance is further evaluated using the Wilcoxon signed-rank test at a significance level of 0.05. The comparison results are summarised using the $W/T/L$ metric, representing the numbers of wins, ties and losses of the proposed algorithm against each competitor.

5.2 | Statistical Results Under the WS1

Under WS1, layout optimization is performed for wind farms with 30, 50 and 100 turbines. For each configuration, the maximum, minimum, mean and standard deviation of conversion efficiency are obtained from 51 independent runs, as summarised in Table 3, where the best results are highlighted in bold and the second best results are indicated by underlining. The mean value reflects the average performance under identical wind conditions and turbine scales. Among the nine comparative algorithms, seven are specifically tailored for WFLO problems, while the remaining two are general-purpose optimisers with strong benchmark performance.

As reported in Table 3, LSDE consistently achieves the highest mean conversion efficiency across all three turbine scales. For the 30-turbine case, the average efficiency of the competing algorithms is 95.82%, whereas LSDE attains 98.15%, corresponding to an improvement of 2.33%. When the number of turbines increases to 50, the average efficiency decreases to 89.36%. LSDE achieves 92.88%, corresponding to a 3.52% improvement. For the 100-turbine case, the average efficiency further declines to 74.12%. LSDE maintains 78.47%, resulting in a 4.35% improvement. These results indicate that although the WFLO problem

TABLE 1 | Overview of benchmark algorithms used in comparative experiments for WFLO, including their sources, principle descriptions and publication years.

Name	Publishing house	Principle description	Year
SHADE	IEEE CEC2013	A success-based differential evolution parameter adaptive algorithm [40]	2013
AGA	Applied energy	An information-guided utilization-based self-information genetic algorithm [45]	2019
SUGGA	Energy conversion and management	A support vector regression–guided genetic algorithm [46]	2019
CJADE	IEEE T SYST MAN CY-S	A differential evolution algorithm based on chaotic local search [17]	2021
AGPSO	Energy conversion and management	A particle swarm optimization algorithm incorporating adaptive replacement strategy [22]	2022
CGPSO	IEEE/CAA JAS	A particle swarm optimization algorithm based on chaotic local search [27]	2023
ISE	Engineering applications of artificial intelligence	A spherical evolution algorithm for enhancing exploration capabilities [39]	2023
CLSHADE	Applied soft computing	An LSHADE algorithm based on chaotic local search with an enhanced memory storage mechanism [29]	2023
BDE	Energy	A discrete two-population differential evolution algorithm [47]	2025

TABLE 2 | Detailed parameter settings of the compared algorithms.

Algorithms	Parameters
LSDE	$N = 120$, $SF = 0.5$, $CR = 0.5$, $Nmin = 0.3D$, $M_s = 5$, $A = 1.4$
SHADE	$N = 120$, $SF = 0.5$, $CR = 0.5$, $M_s = 5$, $A = 1.4$
AGA	$N = 120$, $P_e = 0.2$, $P_c = 0.6$, $P_m = 0.1$, $P_r = 0.5$
SUGGA	$N = 120$, $P_e = 0.2$, $P_c = 0.6$, $P_m = 0.1$, $P_r = 0.5$
CJADE	$N = 120$, $SF = 0.5$, $CR = 0.5$, $M_s = 5$, $C = 0.1$, $P = 0.05$
AGPSO	$N = 120$, $C = 1.49618$, $S = 7$, $P_m = 0.01$
CGPSO	$N = 120$, $C = 1.49618$, $\omega = 0.7298$, $P_m = 0.01$, $L = 25$
ISE	$N = 5$, $F = 10$, $D_c = \frac{1}{3}D$
CLSHADE	$N = 18D$, $P = 0.11$, $H = 5$, $L = 50$
BDE	$N = 120$, $F = 0.5$, $n = 0.05$, $CR = 0.1$

TABLE 3 | Statistical comparison of conversion efficiency under WS1 for three turbine scales (30, 50 and 100), including max, min, mean and standard deviation over 51 independent runs.

Turbine		LSDE	SHADE	CLSHADE	BDE	CJADE	ISE	AGPSO	CGPSO	AGA	SUGGA
$T = 30$	MAX	**98.76**	96.13	96.05	97.56	95.98	95.69	97.63	97.78	95.74	96.12
	MIN	**97.54**	94.41	94.98	96.79	95.00	94.83	97.01	97.07	95.03	94.92
	STD	0.42	0.29	0.27	0.19	0.21	0.16	0.15	**0.14**	0.16	0.19
	MEAN	**98.15**	95.52	95.57	97.14	95.54	95.27	97.31	97.33	95.39	95.46
$T = 50$	MAX	**93.57**	89.49	89.31	91.93	89.37	88.92	92.38	92.36	90.16	90.00
	MIN	**92.18**	86.87	87.00	90.50	86.83	87.68	91.35	91.36	89.10	88.72
	STD	0.59	0.58	0.52	0.24>	0.40	0.26	0.25	**0.23**	0.25	0.28
	MEAN	**92.88**	88.59	88.53	91.30	88.55	88.25	91.89	91.91	89.51	89.33
$T = 100$	MAX	**79.01**	73.86	73.85	76.16	73.51	73.30	77.45	77.47	75.56	75.24
	MIN	**77.67**	72.20	72.24	75.02	72.32	72.35	76.09	76.47	74.42	74.52
	STD	**0.15**	0.42	0.36	0.36	0.31	0.21	0.25	0.21	0.23	0.17
	MEAN	**78.47**	73.04	72.89	75.68	72.89	72.79	76.94	76.99	74.95	74.84
$W/T/L$		—	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0

Note: For MAX, MIN and MEAN, higher values indicate better performance, whereas for STD, lower values indicate better stability.

becomes increasingly challenging with higher dimensionality and stronger wake interactions, LSDE maintains superior performance, demonstrating strong scalability and robustness. Furthermore, the Wilcoxon signed-rank test results, reflected by the $W/T/L$ statistics in Table 3, show that LSDE achieves a 3/0/0 record against all competing algorithms under WS1, confirming that the observed improvements are statistically significant at the 0.05 level.

Figure 6 presents the boxplots and convergence curves of conversion efficiency under WS1 for the three turbine scales. From the boxplots, LSDE consistently achieves the highest median and mean values across all cases. Notably, its minimum values remain competitive with, and in some cases, exceed the upper range of other algorithms, indicating strong robustness. As the problem scale increases, the distribution of LSDE becomes more concentrated, indicating improved solution stability. The convergence curves further reveal the search dynamics of LSDE. Although it may not dominate in the initial iterations, it exhibits accelerated improvement in the middle stage, particularly when several competing algorithms begin to stagnate. In the later stage, LSDE continues to refine solutions while others approach convergence saturation, demonstrating sustained search capability and effective exploitation of promising regions.

Figure 7 compares the conversion efficiency and efficiency drop rates of all algorithms under WS1. Although all methods exhibit decreasing efficiency as the turbine scale increases, LSDE consistently maintains the highest performance. Although several algorithms, such as SHADE, BDE and AGPSO, exhibit similar declining trends, LSDE achieves higher absolute efficiency with a relatively moderate performance drop. This indicates that LSDE is more resilient to increasing problem dimensionality and intensified wake interactions. The lower degradation rate further highlights its robustness under complex wind conditions.

Figure 8 illustrates the layout evolution process of LSDE. For the 30-turbine case, turbines are initially concentrated near the central region at $FES = 500$. As the search progresses, the layout gradually expands towards peripheral regions, forming a more balanced configuration. At $FES = 20000$, a stable and structured pattern emerges, and the conversion efficiency increases from 91.86% to 97.16%. For the 50-turbine case, the

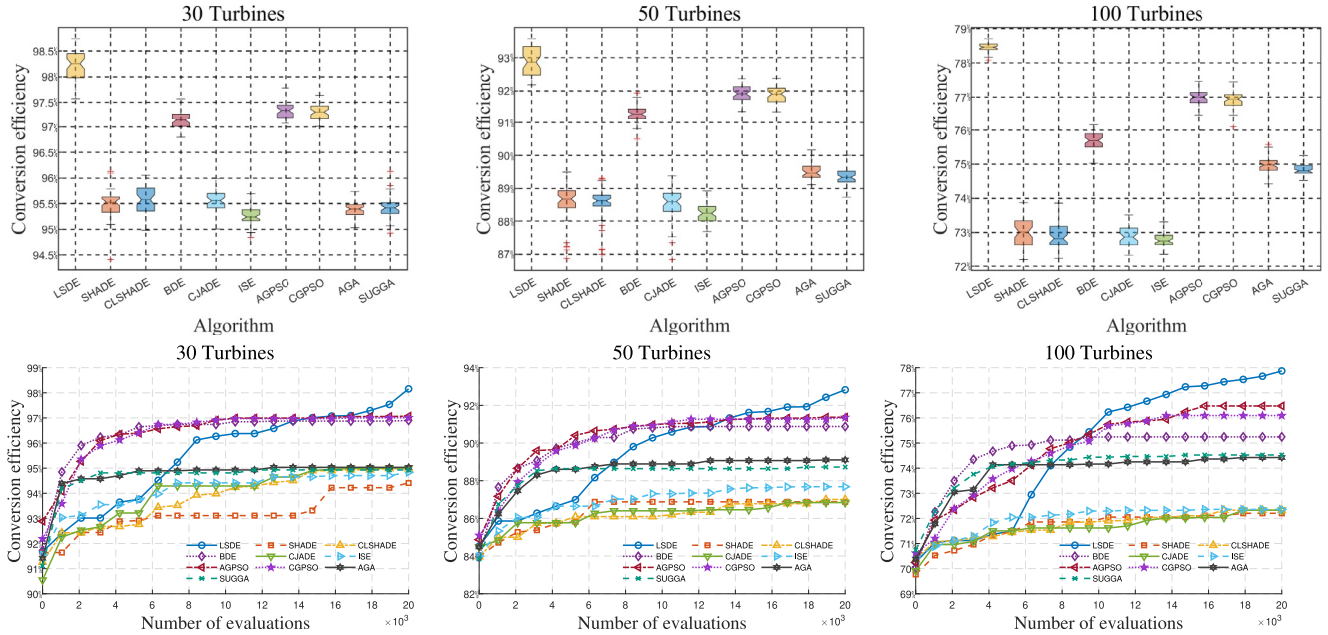


FIGURE 6 | Boxplots and convergence curves of conversion efficiency under WS1 for three turbine scales (30, 50 and 100), showing statistical performance and convergence behaviour of different algorithms.

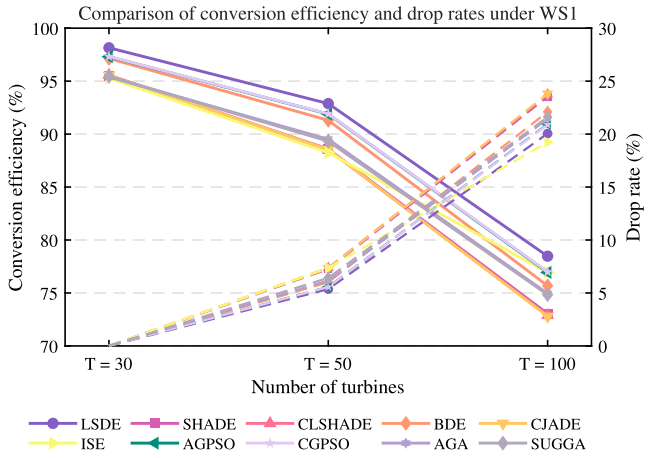


FIGURE 7 | Comparison of conversion efficiency and efficiency drop rates of different algorithms under WS1 across three turbine scales (30, 50 and 100).

initial layout is biased towards one side. As evolution proceeds, it gradually aligns with the dominant wind direction, resulting in a more uniform distribution and reduced wake interference. The efficiency improves from 85.63% to 91.56%. For the 100-turbine case, strong overlap and high-density clustering are observed in early stages. As evolution proceeds, the layout becomes more organised and evenly distributed, leading to improved spatial utilization and an efficiency increase from 70.41% to 77.07%.

Overall, despite the increasing optimization difficulty with larger turbine scales, LSDE maintains stable convergence behaviour and consistent performance gains. The improvement in conversion efficiency increases from 5.3% in the 30-turbine case to 6.7% in the 100-turbine case, further demonstrating its effectiveness and adaptability for large-scale WFLO problems.

5.3 | Statistical Results Under the WS2

Under WS2, wind farm layout optimization is conducted for three turbine scales (30, 50 and 100 turbines). Each configuration is independently executed 51 times, and the maximum, minimum, mean, and standard deviation of conversion efficiency are reported in Table 4, where the best results are highlighted in bold and the second best results are indicated by underlining. The mean value reflects the average performance under identical wind conditions and turbine scales.

As shown in Table 4, LSDE consistently achieves the highest mean conversion efficiency across all cases. For the 30-turbine scenario, the average efficiency of the comparative algorithms is 95.23%, while LSDE attains 98.02%, corresponding to an improvement of 2.79%. When the number of turbines increases to 50, the average efficiency decreases to 88.62%. LSDE achieves 92.59%, corresponding to a 3.97% improvement. For the 100-turbine case, the average efficiency further declines to 73.58%. LSDE maintains 77.34%, resulting in a 3.76% improvement. Compared with WS1, WS2 introduces more complex wind conditions due to increased directional diversity and stronger nonuniformity, leading to performance degradation across all algorithms. Nevertheless, LSDE exhibits the smallest degradation and maintains a consistent performance advantage. The cumulative improvement across all turbine scales increases slightly from 10.20% under WS1 to 10.52% under WS2, further demonstrating its robustness under more challenging conditions. In addition, the Wilcoxon signed-rank test results, summarised by the $W/T/L$ statistics in Table 4, show that LSDE achieves a 3/0/0 record against all competing algorithms, confirming that the performance improvements are statistically significant at the 0.05 level.

Figure 9 illustrates the boxplots and convergence curves of conversion efficiency under WS2. From the boxplots, LSDE

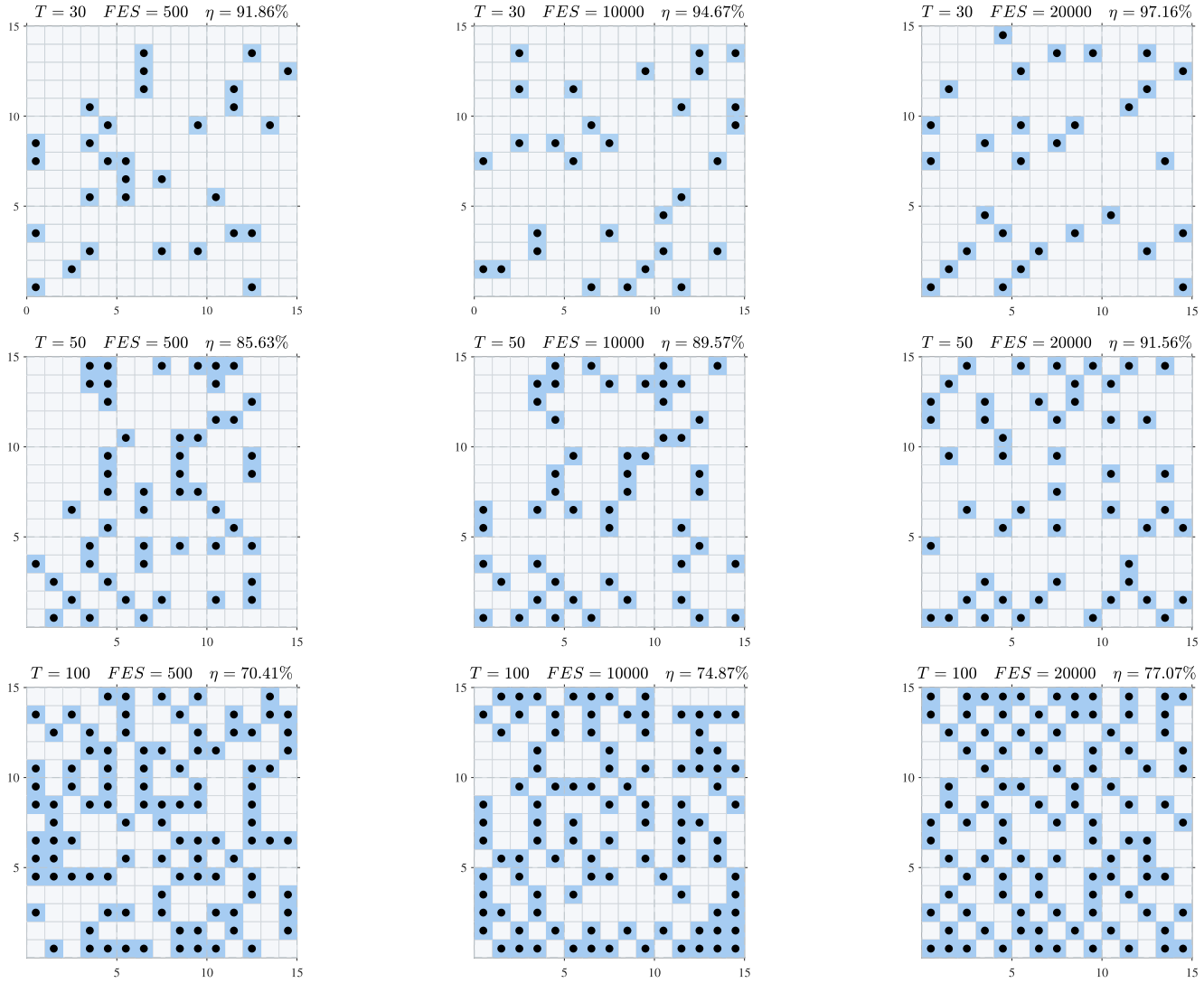


FIGURE 8 | Visualisation of the layout evolution process of the LSDE algorithm under WS1 for three turbine scales (30, 50 and 100) at different stages of FES.

TABLE 4 | Statistical comparison of conversion efficiency under WS2 for three turbine scales (30, 50 and 100), including max, min, mean and standard deviation over 51 independent runs.

Turbine		LSDE	SHADE	CLSHADE	BDE	CJADE	ISE	AGPSO	CGPSO	AGA	SUGGA
$T = 30$	MAX	**98.70**	95.58	95.90	97.15	95.59	95.10	97.13	97.21	95.22	95.43
	MIN	**97.34**	94.09	94.02	95.98	94.16	94.42	96.36	96.34	94.40	94.24
	STD	0.42	0.38	0.40	0.22	0.33	0.16	0.17	0.19	0.17	**0.15**
	MEAN	**98.02**	94.91	95.00	96.59	94.98	94.73	96.73	96.80	94.77	94.64
$T = 50$	MAX	**93.87**	88.90	89.05	90.92	88.88	88.46	91.36	91.16	88.99	88.97
	MIN	**91.32**	87.25	87.08	89.93	87.32	87.45	90.30	90.15	87.97	87.84
	STD	0.58	0.38	0.39	0.24	0.32	0.23	**0.18**	0.19	0.21	0.20
	MEAN	**92.59**	87.97	88.04	90.35	87.96	87.85	90.77	90.79	88.40	88.23
$T = 100$	MAX	**78.21**	73.39	73.32	75.33	73.66	73.10	76.29	76.26	74.21	74.27
	MIN	**76.48**	72.09	72.16	74.50	72.00	72.51	75.39	73.93	73.32	73.46
	STD	**0.12**	0.34	0.32	0.20	0.43	0.13	0.19	0.32	0.17	0.14
	MEAN	**77.34**	72.63	72.71	74.89	72.72	72.72	75.86	75.79	73.78	73.80
W/T/L		—	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0

Note: For MAX, MIN and MEAN, higher values indicate better performance, whereas for STD, lower values indicate better stability.

maintains superior performance and stable distributions across all turbine scales. In the 30-turbine case, although LSDE exhibits a wider spread, indicating moderate variability, its overall performance remains clearly superior to that of the competing algorithms. As the turbine scale increases to 50 and 100, most algorithms exhibit degraded solution quality, reflected by reduced medians and wider dispersion. In contrast, LSDE maintains higher median values while its distribution becomes more concentrated, indicating improved stability in higher dimensional search spaces. Moreover, several competing algorithms exhibit noticeable outliers at larger scales, whereas LSDE maintains compact distributions with limited outliers, further demonstrating its robustness.

The convergence curves reveal the dynamic search behaviour of LSDE. For the 30-turbine case, LSDE shows steady improvement and surpasses other algorithms in later iterations. In the 50-turbine scenario, it achieves superior performance in the mid-to-late stage and maintains a clear advantage thereafter. For the 100-turbine case, LSDE demonstrates consistent convergence from the middle stage onward, highlighting its capability to sustain effective search under increased dimensionality. Overall, LSDE not only achieves superior final performance but also maintains stable convergence behaviour across different scales.

Figure 10 compares the conversion efficiency and efficiency drop rates of 10 algorithms under WS2 across three turbine scales. As the number of turbines increases, the conversion efficiency of all algorithms declines. However, LSDE consistently maintains the highest absolute efficiency across all scales. Although all algorithms exhibit downward trends as turbine numbers increase, LSDE shows the lowest efficiency drop rate among them. This indicates stronger resilience to complexity-induced perturbations and enhanced robustness under increasingly challenging wind conditions.

Figure 11 illustrates the layout evolution of LSDE under WS2. For the 30-turbine case, when $FES = 500$, turbines are relatively scattered, and several empty flow channels appear within the site. At $FES = 10000$, turbines are redistributed more evenly along diagonal and boundary directions, gradually reducing vacant areas. At $FES = 20000$, the layout becomes more structured with alternating sparse and dense regions, and conversion efficiency increases from 92.11% to 96.75%. For the 50-turbine case, when $FES = 500$, local clustering is observed with irregular empty zones near the centre. As evolution proceeds, these clusters are gradually dispersed into multiple ventilation corridors aligned with the prevailing wind direction. At $FES = 20000$, the configuration evolves into a stable block–strip hybrid pattern, and conversion efficiency improves from 85.15% to 90.84%. For the 100-turbine case, when $FES = 500$, turbine distribution is dense and uneven, with strong local wake interactions. At

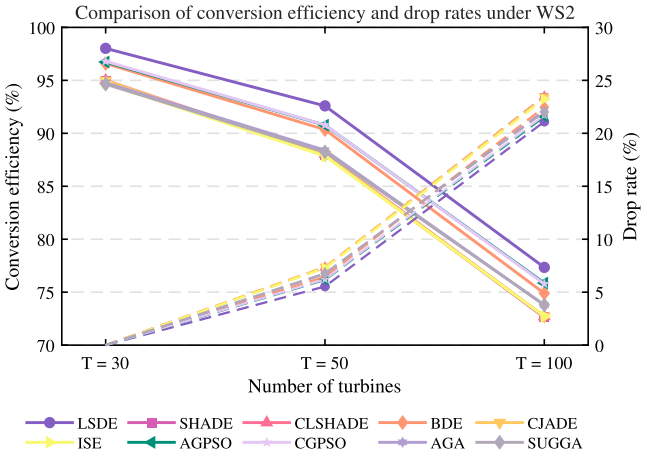


FIGURE 10 | Comparison of conversion efficiency and efficiency drop rates of different algorithms under WS2 across three turbine scales (30, 50 and 100).

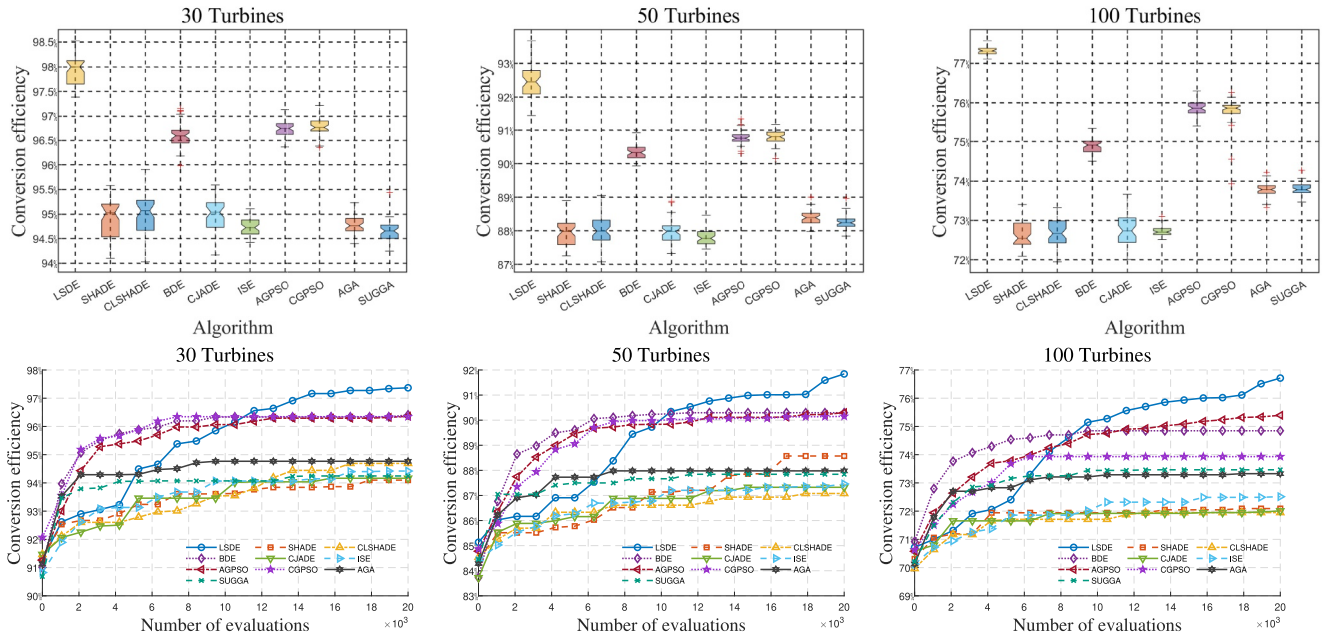


FIGURE 9 | Boxplots and convergence curves of conversion efficiency under WS2 for three turbine scales (30, 50 and 100), showing statistical performance and convergence behaviour of different algorithms.

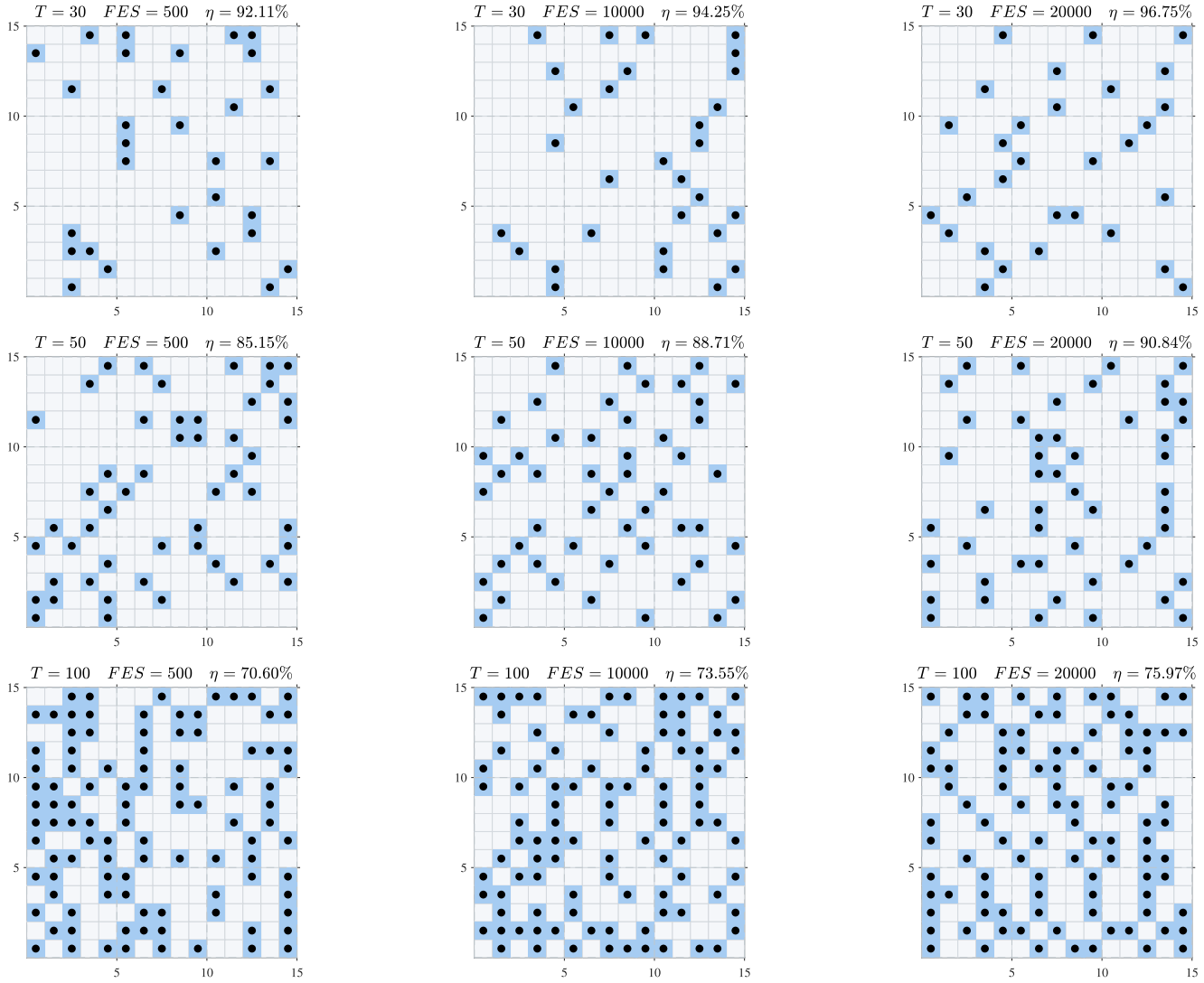


FIGURE 11 | Visualisation of the layout evolution process of the LSDE algorithm under WS2 for three turbine scales (30, 50 and 100) at different stages of FES.

$FES = 10000$, the algorithm alleviates high-density regions by forming orderly flow channels. At $FES = 20000$, turbines form an approximately chessboard-like configuration with improved spacing and spatial utilization. Conversion efficiency increases from 70.60% to 75.97%.

Overall, under WS2, LSDE progressively transforms clustered initial layouts into structured wake mitigation patterns. Although increasing turbine numbers raise problem complexity, efficiency improvements of 4.64%, 5.69% and 5.37% are achieved for the three turbine scales, respectively, demonstrating consistent adaptability.

5.4 | Statistical Results Under the WS3

Under WS3, wind farm layout optimization is performed for three turbine scales (30, 50 and 100 turbines). Each configuration is independently executed 51 times, and the maximum, minimum, mean and standard deviation of conversion efficiency are summarised in Table 5, where the best results are highlighted in bold and the second best results are indicated by underlining. The mean value reflects the average performance under identical wind conditions and turbine scales.

As reported in Table 5, LSDE consistently achieves the highest mean conversion efficiency across all configurations. For the 30-turbine case, the average efficiency of the comparative algorithms is 96.30%, while LSDE attains 98.55%, corresponding to an improvement of 2.25%. When the number of turbines increases to 50, the average efficiency decreases to 91.28%. LSDE achieves 94.60%, corresponding to a 3.32% improvement. For the 100-turbine case, the average efficiency further declines to 79.59%. LSDE maintains 83.33%, resulting in a 3.74% improvement. These results indicate that although the WFLO problem becomes more challenging with increasing dimensionality and intensified wake interactions, LSDE not only preserves superior performance but also exhibits an increasing relative advantage. The Wilcoxon signed-rank test results, summarised by the $W/T/L$ statistics in Table 5, show that LSDE achieves a 3/0/0 record against all competing algorithms, confirming that the performance improvements are statistically significant at the 0.05 level.

Figure 12 presents the boxplots and convergence curves of conversion efficiency obtained by 10 algorithms under WS3 across three turbine scales. From the boxplots, LSDE maintains competitive performance across all problem scales. In the 30-turbine scenario, LSDE exhibits a narrow box width, indicating highly concentrated solutions with low variance. Its efficiency distribution remains higher than the maximum values obtained by the competing algorithms, reflecting strong solution quality and stable convergence behaviour. In the 50-turbine scenario, the box width of LSDE increases slightly but remains narrower than that of most competitors. Both its mean and median values continue to rank highest. In the 100-turbine scenario, variance increases moderately as expected under higher complexity. However, LSDE maintains higher median and upper bound values than other algorithms. Although several comparative methods generate noticeable outliers in the 50 and 100 turbine cases, LSDE produces almost none, further supporting its stability under complex conditions.

The convergence curves further corroborate these findings. In the 30-turbine case, many algorithms stagnate early, whereas LSDE achieves additional improvement in later iterations. In the 50-turbine case, LSDE escapes local optima earlier and maintains steady improvement. For the 100-turbine case, most algorithms plateau in the mid-to-late stages, while LSDE continues gradual progress. These observations indicate LSDE's strong global search ability and resistance to premature convergence under increasing dimensionality.

TABLE 5 | Statistical comparison of conversion efficiency under WS3 for three turbine scales (30, 50 and 100), including max, min, mean and standard deviation over 51 independent runs.

Turbine		LSDE	SHADE	CLSHADE	BDE	CJADE	ISE	AGPSO	CGPSO	AGA	SUGGA
$T = 30$	MAX	**98.96**	96.32	96.47	98.02	96.42	96.05	98.14	98.21	96.70	96.45
	MIN	**98.15**	94.83	94.94	96.88	94.96	95.44	97.07	97.12	95.90	95.77
	STD	**0.13**	0.22	0.31	0.24	0.24	0.15	0.25	0.24	0.18	0.16
	MEAN	**98.55**	95.92	95.88	97.45	95.95	95.71	97.62	97.65	96.22	96.09
$T = 50$	MAX	**95.15**	91.04	91.11	93.56	91.42	90.89	94.00	94.06	92.36	91.91
	MIN	**94.05**	89.83	89.86	92.47	89.89	90.02	92.58	92.89	91.28	90.96
	STD	0.27	0.31	0.24	0.23	0.36	**0.19**	0.32	0.24	0.22	0.20
	MEAN	**94.60**	90.36	90.27	92.89	90.43	90.41	93.38	93.46	91.72	91.48
$T = 100$	MAX	**84.76**	79.71	79.59	80.81	79.42	78.15	81.80	81.73	80.41	80.27
	MIN	**81.91**	78.77	78.80	79.75	78.44	77.42	79.92	80.72	79.62	79.38
	STD	0.59	0.17	0.16	0.27	0.17	0.17	0.32	0.23	0.18	**0.15**
	MEAN	**83.33**	79.05	79.08	80.23	79.10	77.75	81.31	81.38	79.98	79.86
W/T/L		—	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0

Note: For MAX, MIN and MEAN, higher values indicate better performance, while for STD, lower values indicate better stability.

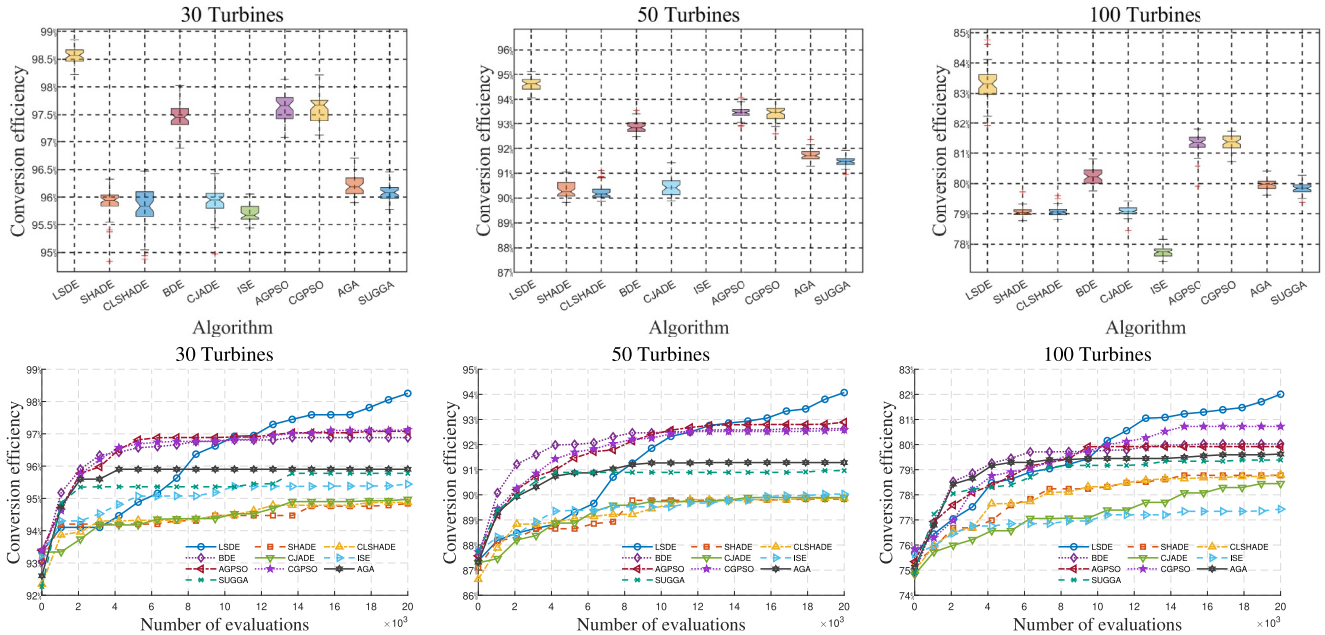


FIGURE 12 | Boxplots and convergence curves of conversion efficiency under WS3 for three turbine scales (30, 50 and 100), showing statistical performance and convergence behaviour of different algorithms.

Figure 13 compares the conversion efficiency and efficiency drop rates of different algorithms under WS3 across three turbine scales. As turbine numbers increase, the efficiency of all algorithms declines. However, LSDE consistently maintains the highest absolute efficiency across all scales. From the solid-line portion, the efficiency reduction between 30 and 50 turbines appears similar among algorithms. When the scale increases to 100 turbines, the slopes of BDE, AGPSO, AGA, SUGGA and CGPSO become noticeably steeper than that of LSDE, indicating reduced adaptability to large-scale configurations. In contrast, LSDE exhibits a smoother decline trend, suggesting stronger resilience to increasing wake interactions and system complexity. From the dashed-line portion, LSDE demonstrates the lowest relative efficiency drop among the 10 algorithms, highlighting its robustness under more complex wind environments.

Figure 14 illustrates the layout evolution of LSDE under WS3. For the 30-turbine case, when $FES = 500$, turbines are relatively concentrated in the central region. At $FES = 10000$, the distribution expands along the prevailing wind direction, extending towards site boundaries. At $FES = 20000$, the layout becomes more uniform, approaching a structured grid-like pattern, and conversion efficiency increases from 93.87% to 97.49%. For the 50-turbine case, when $FES = 500$, the layout is denser with local overlaps. As evolution proceeds, clustered regions are gradually dispersed into a more balanced configuration. At $FES = 20000$, turbines align more clearly with the main wind direction and wake interference is reduced. Conversion efficiency improves from 88.22% to 92.92%. For the 100-turbine case, when $FES = 500$, the distribution is compact and uneven due to strong interactions. At $FES = 10000$, local adjustments increase turbine spacing, forming a more regular matrix-like structure. At $FES = 20000$, the configuration stabilises with improved spatial coverage, and efficiency increases from 75.71% to 79.30%.

Overall, as FES increases, LSDE progressively transforms initially centre-clustered layouts into balanced spatial configurations with improved wake mitigation and spatial utilization.

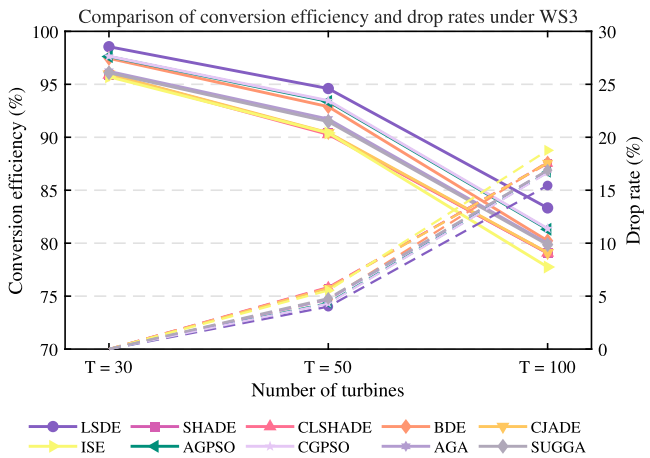


FIGURE 13 | Comparison of conversion efficiency and efficiency drop rates of different algorithms under WS3 across three turbine scales (30, 50 and 100).

5.5 | Statistical Results Under the WS4

Under WS4, layout optimization is conducted for wind farms with 30, 50 and 100 turbines. Each configuration is independently executed 51 times, and the maximum, minimum, mean and standard deviation of conversion efficiency are summarised in Table 6, where the best results are highlighted in bold and the second best results are indicated by underlining. The mean value reflects the average performance under identical wind conditions and turbine scales.

As reported in Table 6, LSDE achieves the highest mean conversion efficiency across all turbine scales. For the 30-turbine case, the average efficiency of the comparative algorithms is 93.92%, while LSDE attains 97.01%, corresponding to a 3.09% improvement. When the number of turbines increases to 50, the average efficiency decreases to 86.28%. LSDE achieves 90.04%, corresponding to a 3.78% improvement. For the 100-turbine case, the average efficiency further declines to 69.38%. LSDE maintains 72.59%, resulting in a 3.21% improvement.

Compared with WS1–WS3, WS4 introduces stronger wind direction variability and more pronounced multi-scale wind speed fluctuations, making it the most challenging scenario. Despite this increased complexity, LSDE maintains consistent relative improvements across all scales. Notably, the aggregated improvements across the four wind scenarios (WS1–WS4) remain close to 10% (10.20%, 10.52%, 9.31% and 10.08%), indicating that LSDE is largely insensitive to increasing environmental complexity. This further demonstrates its strong robustness. In addition, the Wilcoxon signed-rank test results, summarised by the $W/T/L$ statistics in Table 6, show that LSDE achieves a 3/0/0 record against all competing algorithms, confirming statistically significant improvements at the 0.05 level.

Figure 15 presents the boxplots and convergence curves under WS4. From the boxplots, LSDE consistently maintains superior median and upper quartile values across all turbine scales. In the 30-turbine case, the interquartile range is narrower with no visible outliers, indicating concentrated solutions and stable convergence. As the scale increases to 50 and 100 turbines, the dispersion moderately increases due to higher dimensionality. However, LSDE still preserves a compact distribution with limited outliers, reflecting strong robustness under complex conditions.

The convergence curves further reveal the dynamic search behaviour. In the 30-turbine case, several algorithms converge rapidly and stagnate early, whereas LSDE continues to improve in the middle and later stages. For the 50-turbine case, LSDE maintains steady convergence throughout the search process. Although AGPSO and CGPSO temporarily outperform LSDE in intermediate iterations, LSDE ultimately regains the lead. In the 100-turbine scenario, all algorithms exhibit slower convergence due to increased complexity. However, LSDE sustains gradual improvement and achieves higher final efficiency, indicating strong global search capability and resistance to premature convergence. These results collectively indicate stable convergence behaviour and scalability under WS4 as the turbine scale increases.

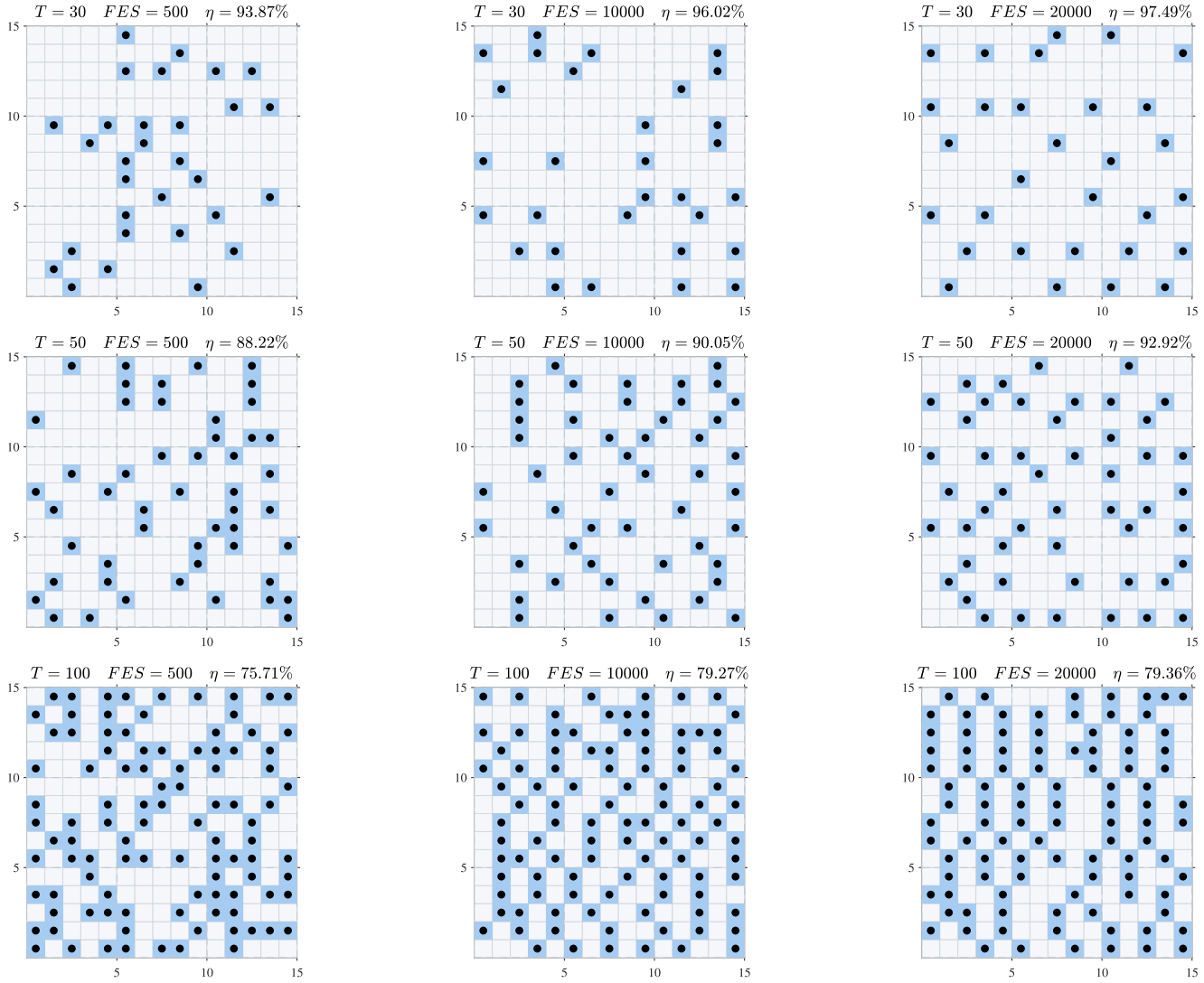


FIGURE 14 | Visualisation of the layout evolution process of the LSDE algorithm under WS3 for three turbine scales (30, 50 and 100) at different stages of FES.

TABLE 6 | Statistical comparison of conversion efficiency under WS4 for three turbine scales (30, 50 and 100), including max, min, mean and standard deviation over 51 independent runs.

Turbine		LSDE	SHADE	CLSHADE	BDE	CJADE	ISE	AGPSO	CGPSO	AGA	SUGGA
$T = 30$	MAX	**97.85**	94.45	94.35	95.83	94.44	94.07	96.01	96.16	93.83	93.71
	MIN	**96.16**	92.82	92.79	94.68	92.89	93.00	95.02	94.94	92.98	92.87
	STD	0.49	0.37	0.41	0.24	0.27	0.22	0.26	0.26	0.18	**0.17**
	MEAN	**97.01**	93.65	93.55	95.27	93.63	93.42	95.50	95.49	93.41	93.30
$T = 50$	MAX	**91.43**	86.34	86.24	88.07	86.31	85.66	88.66	88.64	86.36	86.30
	MIN	**88.65**	85.40	85.03	87.08	85.41	84.65	87.86	87.63	85.46	85.37
	STD	0.90	0.19	0.24	0.23	0.17	0.22	**0.16**	0.18	0.21	0.20
	MEAN	**90.04**	85.71	85.66	87.58	85.77	85.17	88.22	88.24	85.95	85.79
$T = 100$	MAX	**73.54**	69.38	69.32	70.58	69.32	68.21	71.74	71.60	69.84	69.94
	MIN	**71.65**	68.57	68.32	69.54	68.70	67.53	70.79	70.77	69.00	68.84
	STD	0.75	0.17	0.17	0.22	**0.13**	0.17	0.19	0.20	0.21	0.16
	MEAN	**72.59**	68.93	68.93	70.09	68.97	67.86	71.24	71.23	69.31	69.28
W/T/L		—	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0

Note: For MAX, MIN and MEAN, higher values indicate better performance, while for STD, lower values indicate better stability.

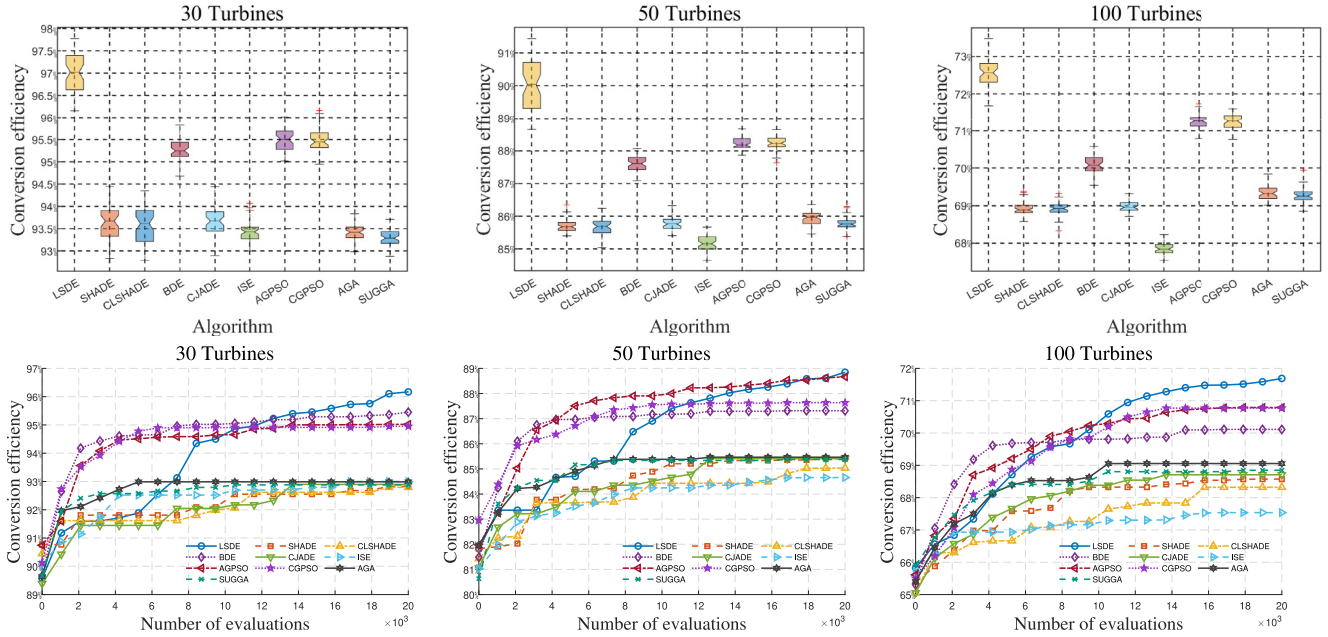


FIGURE 15 | Boxplots and convergence curves of conversion efficiency under WS4 for three turbine scales (30, 50 and 100), showing statistical performance and convergence behaviour of different algorithms.

Figure 16 compares conversion efficiency and efficiency drop rates under WS4 across three turbine scales. As turbine numbers increase, efficiency declines for all algorithms. However, LSDE consistently maintains the highest absolute efficiency across all scales. From the solid lines, the efficiency reduction from 30 to 100 turbines follows a similar overall trend for most algorithms, whereas LSDE exhibits a slightly gentler slope, indicating slower performance degradation. In contrast, several competing algorithms, including BDE, CGPSO and SHADE, show steeper declines in large-scale configurations, reflecting greater sensitivity to increased wake interactions. From the dashed-line portion, LSDE demonstrates the lowest relative efficiency drop among the 10 algorithms, particularly in the 100-turbine case. This suggests strong resilience to increased dimensionality and environmental variability.

Figure 17 illustrates the layout evolution of LSDE under WS4. For the 30-turbine case, when $FES = 500$, turbines are relatively scattered, with a slight distribution bias towards the lower region of the site. At $FES = 10000$, turbines gradually redistribute along the diagonal direction, forming a more balanced configuration. At $FES = 20000$, the layout stabilises into a clearer structural pattern, and conversion efficiency increases from 90.68% to 95.66%. For the 50-turbine case, the initial distribution at $FES = 500$ appears dense and partially overlapped. As evolution proceeds, the algorithm progressively reorganises turbines along the prevailing wind direction, forming a more regular band-like structure. At $FES = 20000$, wake interference is reduced and efficiency improves from 82.83% to 86.22%. For the 100-turbine case, the initial configuration is compact with noticeable clustering. At $FES = 10000$, local adjustments increase spacing and improve ordering. At $FES = 20000$, the turbines form a balanced grid-like pattern, and conversion efficiency rises from 66.02% to 70.36%.

Overall, as FES increases, LSDE progressively transforms irregular initial layouts into more structured configurations

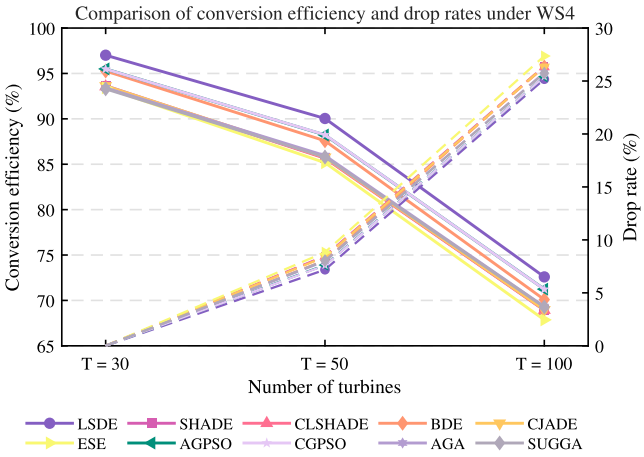


FIGURE 16 | Comparison of conversion efficiency and efficiency drop rates of different algorithms under WS4 across three turbine scales (30, 50 and 100).

with improved spatial utilization. The efficiency improvements of approximately 5.0%, 3.4% and 4.3% for the three turbine scales further demonstrate stable convergence under WS4.

5.6 | Ablation Analysis and Mechanisms Discussion

5.6.1 | *Component-Wise Ablation Study*

To further investigate the effectiveness and necessity of each core component in the proposed LSDE framework, a component-wise ablation study is conducted under the WS1 scenario with 50 turbines. Four representative variants are constructed by removing or modifying one component at a time while keeping all other components unchanged, allowing the independent contribution of each module to be evaluated. The experimental results are summarised in Table 7.

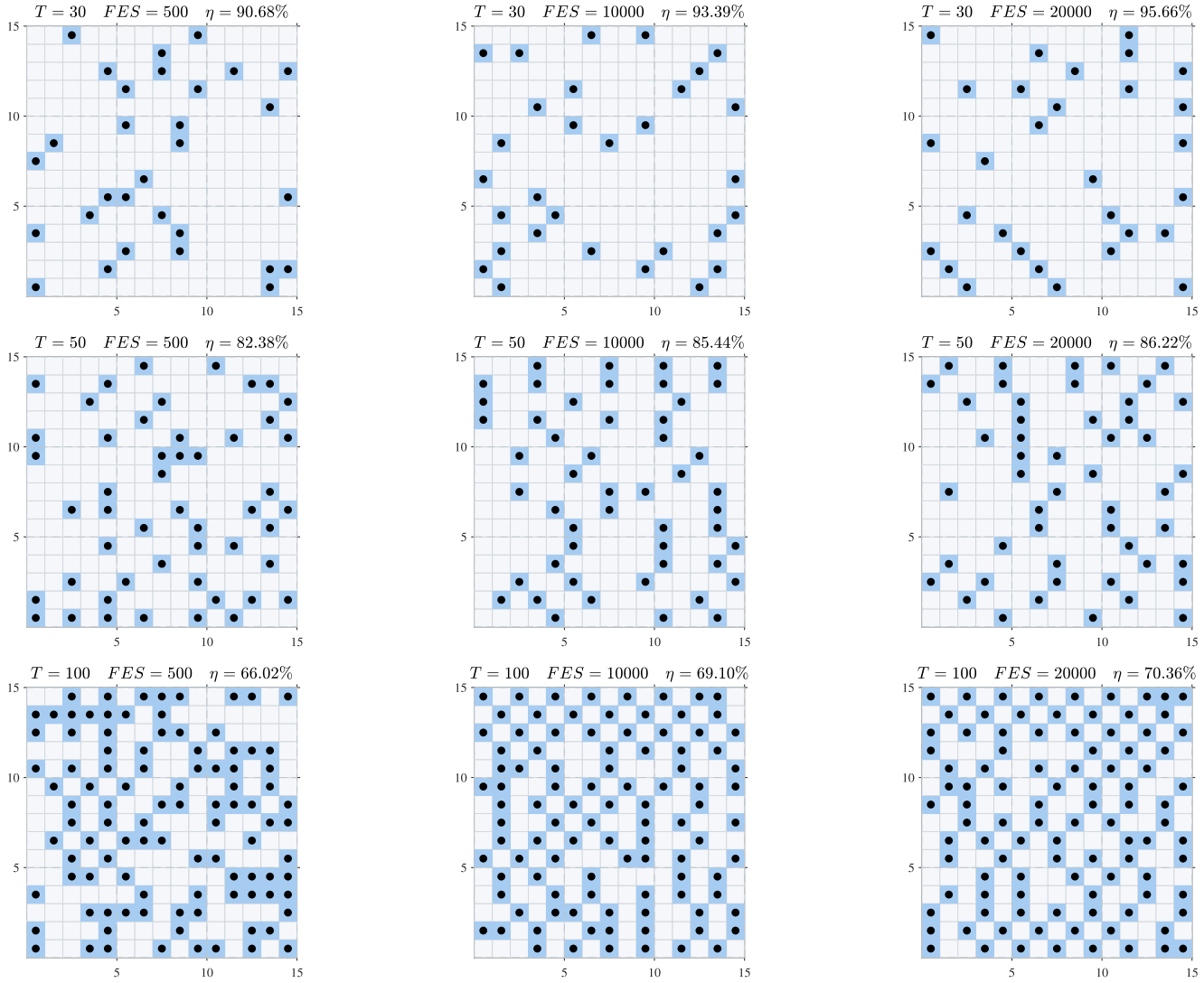


FIGURE 17 | Visualisation of the layout evolution process of the LSDE algorithm under WS4 for three turbine scales (30, 50 and 100) at different stages of FES.

TABLE 7 | Component-wise ablation study of LSDE for mechanism analysis under WS1 with 50 turbines.

Variant	Hierarchy	Subpopulation	Restart	Population reduction	Conversion efficiency (%)
LSDE	✓	✓ ($K = 3$)	✓	Cosine NLPR	**93.32**
LSDE-w/o-H	×	✓ ($K = 3$)	✓	Cosine NLPR	90.45
LSDE-w/o-R	✓	✓ ($K = 3$)	×	Cosine NLPR	91.41
LSDE-LPR	✓	✓ ($K = 3$)	✓	Linear PR	92.08
LSDE-w/o-D	✓	× ($K = 1$)	×	Cosine NLPR	88.72

As shown in Table 7, the complete LSDE achieves the best performance with a conversion efficiency of 93.32%, whereas all ablated variants exhibit noticeable performance degradation, indicating that each component contributes positively to the overall framework. Specifically, removing the hierarchical structure (LSDE-w/o-H) results in the most significant drop 90.45%. This demonstrates its critical role in structuring information flow and improving convergence efficiency. Eliminating the restart mechanism (LSDE-w/o-R) leads to a decrease to 91.41%. This suggests that the stagnation-triggered restart strategy effectively maintains diversity and prevents premature convergence. Replacing the nonlinear population reduction with a linear scheme (LSDE-LPR) reduces the performance to 92.08%. This indicates that the cosine-based strategy provides a more effective balance between exploration and exploitation. Moreover, the single-population variant (LSDE-w/o-D) yields the worst performance 88.72%. This highlights the importance of the distributed multi-subpopulation structure in enhancing diversity and global search capability under complex WFLO landscapes.

Overall, these results demonstrate that each component plays a distinct and indispensable role in the LSDE framework. More

importantly, their combination produces a clear synergistic effect, leading to superior performance to all ablated variants. These findings further indicate that the proposed LSDE is not a simple combination of existing strategies but a structured framework in which each component is carefully designed to address the characteristics of WFLO problems.

5.6.2 | *Effect of the Subpopulation Number and Nonlinear Population Reduction*

To further analyse the contribution of the proposed hierarchical design, an ablation study is conducted under WS1 with 50 turbines. Two key factors are examined: (1) the number of subpopulations and (2) whether the nonlinear population reduction strategy (NLPR) is enabled (w NLPR) or disabled (w/o NLPR). In total, 10 configurations are evaluated, as summarised in Table 8, and the corresponding results are visualised using a heat map in Figure 18.

The results reveal that the number of subpopulations has a pronounced impact on the optimization behaviour. When the number is small (e.g., 1 or 2), the algorithm tends to converge prematurely, which can be attributed to limited population diversity and insufficient structural decomposition. As the number of subpopulations increases, the balance between global exploration and local exploitation is progressively improved. In particular, the configuration with three subpopulations achieves the best conversion efficiency, indicating that it provides an effective trade-off between convergence speed and exploration capability. This superiority of $K = 3$ is consistently observed across all NLPR settings. However, further increasing the number of subpopulations (e.g., 4 or 5) leads to excessive partitioning, which weakens the search depth within each subgroup and results in a slight performance degradation.

In addition, NLPR consistently improves performance across all configurations. Under the w NLPR setting, conversion efficiency ranges from 88.54% to 93.32%, whereas under the w/o NLPR setting, it ranges from 86.87% to 92.04%. This improvement indicates that NLPR effectively reallocates computational resources during the evolutionary process. As a result, convergence stability and adaptive search capability are enhanced. From a structural perspective, the results suggest that an appropriate level of population decomposition facilitates efficient information exchange. At the same time, it maintains sufficient search depth within each subgroup. Overall, the findings demonstrate that the number of subpopulations plays a critical role in regulating the exploration–exploitation balance, with $K = 3$ providing the most favourable configuration, while NLPR further strengthens this balance and improves both efficiency and robustness under realistic wind conditions.

5.6.3 | *Replacement Ratio Analysis*

To further examine the effect of the replacement ratio (RR), an additional hyperparameter analysis is conducted under WS1 for wind farms with 30, 50 and 100 turbines. In this experiment, the number of subpopulations is fixed to $K = 3$, whereas RR is varied across four settings, that is, 5%, 10%, 15% and 20%. All other algorithmic parameters remain unchanged. The corresponding results are presented in Figure 19.

As shown in Figure 19, the performance of the proposed framework exhibits a clear dependence on the RR across all turbine scales. When RR is set to 5%, the perturbation introduced by the restart mechanism is weaker, limiting the ability of stagnant subpopulations to escape local optima. This results in lower conversion efficiency, particularly in large-scale scenarios (e.g., 100 turbines), where the search landscape is more complex and requires stronger diversity injection. As RR increases to 10%, the best performance is consistently achieved across all turbine scales. This indicates that a moderate replacement ratio effectively balances diversity introduction and solution preservation. Specifically, the restart mechanism introduces sufficient new individuals to explore underexploited regions. At the same time, it preserves high-quality solutions accumulated during previous iterations. When RR is further increased to 15%, the performance

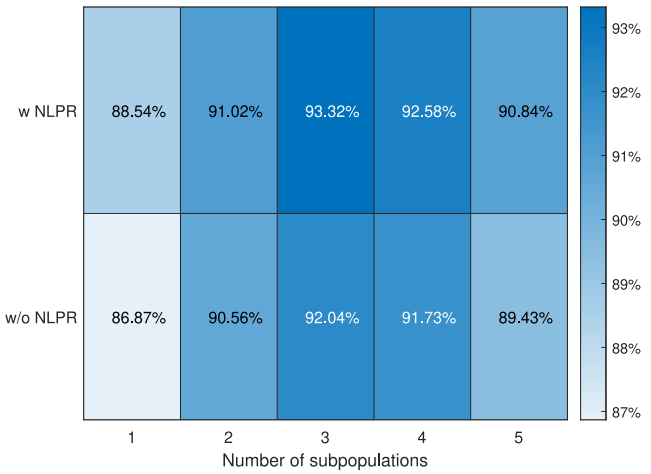


FIGURE 18 | Heat map visualisation of conversion efficiency under different subpopulation numbers with and without NLPR.

TABLE 8 | Ablation study of the subpopulation number and nonlinear population reduction (NLPR) on conversion efficiency, comparing results with and without NLPR.

w/NLPR		w/o NLPR	
# of subpopulations	**Conversion efficiency**	**# of subpopulations**	**Conversion efficiency**
1	88.54	1	86.87
2	91.02	2	90.56
3	93.32	3	92.04
4	92.58	4	91.73
5	90.84	5	89.43

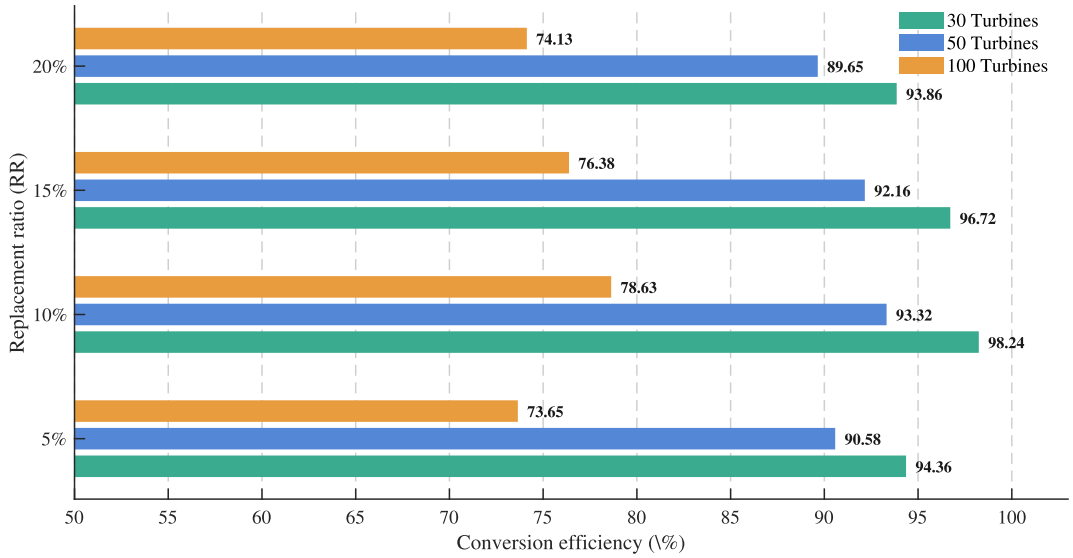


FIGURE 19 | Analysis of the effect of the replacement ratio on conversion efficiency under WS1 across three turbine scales (30, 50 and 100).

remains competitive but shows a slight decline compared to the 10% setting, suggesting that excessive perturbation begins to interfere with the exploitation process. This effect becomes more pronounced at 20%. A clear performance degradation is observed across all configurations, indicating that overly aggressive replacement disrupts convergence and weakens local refinement capability.

Furthermore, a consistent pattern is observed across different turbine scales, where the superiority of the 10% setting remains stable from 30 to 100 turbines, demonstrating strong robustness under varying problem sizes. This observation supports the theoretical expectation that an appropriate level of structural perturbation is essential for maintaining a balance between exploration and exploitation. Therefore, considering both performance and robustness, 10% is selected as the default RR in this study.

6 | Conclusion and Future Works

This study presents a multi-strategy synergy-based differential evolution algorithm (LSDE) to address convergence instability and premature stagnation in WFLO under complex wind conditions. The proposed framework integrates an adaptive switching population structure to regulate the balance between global exploration and local exploitation. The hierarchical layered design enhances convergence efficiency through elite guidance, while the distributed subpopulation structure enables structured information exchange and improves the ability to escape local optima. Furthermore, a nonlinear population reduction strategy is incorporated to reinforce the adaptive switching mechanism, allowing LSDE to preserve population diversity in early stages and achieve stable convergence in later stages.

Comprehensive experiments conducted under four complex wind scenarios (WS1–WS4) and three turbine scales (30, 50 and 100 turbines) confirm the effectiveness and robustness of the proposed approach. Compared with nine state-of-the-art algorithms, LSDE consistently achieves superior performance, with average improvements in power conversion efficiency of 3.40%, 3.51%, 3.10% and 3.36% under WS1–WS4, respectively. These results indicate that the coordinated integration of hierarchical structuring, adaptive switching and nonlinear population control effectively alleviates the limitations of conventional differential evolution methods, while maintaining strong scalability and robustness in large-scale WFLO problems.

Future work will be pursued along three application-oriented directions. First, learning-assisted strategies, such as deep neural networks, will be introduced to enable adaptive parameter control and predictive search guidance. This direction is particularly promising for real-time or dynamic wind farm layout optimization, where wind conditions vary over time and fast decision-making is required. Second, more advanced wake interaction models that incorporate three-dimensional flow dynamics and atmospheric stability effects will be considered. Such extensions are especially relevant for large-scale offshore wind farms, where complex wake interactions and environmental uncertainties significantly influence energy production. Third, economic and operational factors, including installation cost, maintenance scheduling and grid integration constraints, will be incorporated to formulate a multi-objective optimization framework. This direction is expected to enhance the applicability of LSDE in practical engineering scenarios, such as integrated wind farm planning and long-term investment decision-making. These directions aim to bridge the gap between algorithmic development and real-world deployment, thereby further improving the practical impact of the proposed method.

Acknowledgements

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant Nos. JP25K21298 and JP25K03179 and Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant No. JPMJSP2145.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data are available upon request.

References

1. H. M. A. Farid, A. Razzaq, M. Riaz, T. Senapati, and S. Moslem, "Optimising Wave Energy Plant Location Through Neutrosophic Multi-Criteria Group Decision-Making," *CAAI Transactions on Intelligence Technology* 11, no. 1 (2026): 167–189, https://doi.org/10.1049/cit2.70058.

2. H. Li, Z. Zhang, Q. Fu, et al., "Optimizing Wave Energy Converter Layouts in Real Ocean Conditions With a Multi-Layered Spherical Evolutionary Algorithm," *Applied Soft Computing* 187 (2026): 114289, https://doi.org/10.1016/j.asoc.2025.114289.

3. Y. Lan, X. Xu, J. Liu, X. Zhang, Y. Lu, and L. Cheng, "A Survey on Reinforcement Learning for Optimal Decision-Making and Control of Intelligent Vehicles," *CAAI Transactions on Intelligence Technology* 10, no. 6 (2025): 1593–1615, https://doi.org/10.1049/cit2.70073.

4. S. Carley and D. M. Konisky, "The Justice and Equity Implications of the Clean Energy Transition," *Nature Energy* 5, no. 8 (2020): 569–577, https://doi.org/10.1038/s41560-020-0641-6.

5. S. A. Vargas, G. R. T. Esteves, P. M. Maçaira, B. Q. Bastos, F. L. C. Oliveira, and R. C. Souza, "Wind Power Generation: A Review and a Research Agenda," *Journal of Cleaner Production* 218 (2019): 850–870, https://doi.org/10.1016/j.jclepro.2019.02.015.

6. T. R. Ayodele, A. Ogunjuyigbe, O. Odigie, and J. L. Munda, "A Multi-Criteria GIS Based Model for Wind Farm Site Selection Using Interval Type-2 Fuzzy Analytic Hierarchy Process: The Case Study of Nigeria," *Applied Energy* 228 (2018): 1853–1869, https://doi.org/10.1016/j.apenergy.2018.07.051.

7. B. Hand, G. Kelly, and A. Cashman, "Aerodynamic Design and Performance Parameters of a Lift-Type Vertical Axis Wind Turbine: A Comprehensive Review," *Renewable and Sustainable Energy Reviews* 139 (2021): 110699, https://doi.org/10.1016/j.rser.2020.110699.

8. R. Nash, R. Nouri, and A. Vasel-Be-Hagh, "Wind Turbine Wake Control Strategies: A Review and Concept Proposal," *Energy Conversion and Management* 245 (2021): 114581, https://doi.org/10.1016/j.enconman.2021.114581.

9. X. Liu, Z. Lin, and Z. Feng, "Short-Term Offshore Wind Speed Forecast by Seasonal ARIMA-A Comparison Against GRU and LSTM," *Energy* 227 (2021): 120492, https://doi.org/10.1016/j.energy.2021.120492.

10. S. R. Reddy, "Wind Farm Layout Optimization WindFLO: An Advanced Framework for Fast Wind Farm Analysis and Optimization," *Applied Energy* 269 (2020): 115090, https://doi.org/10.1016/j.apenergy.2020.115090.

11. F. González-Longatt, P. Wall, and V. Terzija, "Wake Effect in Wind Farm Performance: Steady-State and Dynamic Behavior," *Renewable Energy* 39, no. 1 (2012): 329–338, https://doi.org/10.1016/j.renene.2011.08.053.

12. A. H. Schrotenboer and E. Ursavas, "Vis IF. Mixed Integer Programming Models for Planning Maintenance at Offshore Wind Farms Under Uncertainty," *Transportation Research Part C: Emerging Technologies* 112 (2020): 180–202, https://doi.org/10.1016/j.trc.2019.12.014.

13. Y. Wang, Y. Yu, S. Cao, X. Zhang, and S. Gao, "A Review of Applications of Artificial Intelligent Algorithms in Wind Farms," *Artificial Intelligence Review* 53, no. 5 (2020): 3447–3500, https://doi.org/10.1007/s10462-019-09768-7.

14. B. Alhijawi and A. Awajan, "Genetic Algorithms: Theory, Genetic Operators, Solutions, and Applications," *Evolutionary Intelligence* 17, no. 3 (2024): 1245–1256, https://doi.org/10.1007/s12065-023-00822-6.

15. E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application," *Swarm and Evolutionary Computation* 63 (2021): 100868, https://doi.org/10.1016/j.swevo.2021.100868.

16. E. Rashedi, E. Rashedi, and H. Nezamabadi-Pour, "A Comprehensive Survey on Gravitational Search Algorithm," *Swarm and Evolutionary Computation* 41 (2018): 141–158, https://doi.org/10.1016/j.swevo.2018.02.018.

17. S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic Local Search-Based Differential Evolution Algorithms for Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, no. 6 (2021): 3954–3967, https://doi.org/10.1109/tsmc.2019.2956121.

18. S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A State-of-the-art Differential Evolution Algorithm for Parameter Estimation of Solar Photovoltaic Models," *Energy Conversion and Management* 230 (2021): 113784, https://doi.org/10.1016/j.enconman.2020.113784.

19. M. Neshat, B. Alexander, N. Y. Sergiienko, and M. Wagner, "New Insights into Position Optimisation of Wave Energy Converters Using Hybrid Local Search," *Swarm and Evolutionary Computation* 59 (2020): 100744, https://doi.org/10.1016/j.swevo.2020.100744.

20. Z. Lei, S. Gao, Z. Zhang, M. Zhou, and J. Cheng, "MO4: A Many-Objective Evolutionary Algorithm for Protein Structure Prediction," *IEEE Transactions on Evolutionary Computation* 26, no. 3 (2022): 417–430, https://doi.org/10.1109/tevc.2021.3095481.

21. Z. Mao, Z. Yang, D. Luo, et al., "A Multi-Strategy Enhanced Dung Beetle Algorithm for Solving Real-World Engineering Problems," *Artificial Intelligence Review* 58, no. 8 (2025): 1–79, https://doi.org/10.1007/s10462-025-11235-5.

22. Z. Lei, S. Gao, Y. Wang, Y. Yu, and L. Guo, "An Adaptive Replacement Strategy-Incorporated Particle Swarm Optimizer for Wind Farm Layout Optimization," *Energy Conversion and Management* 269 (2022): 116174, https://doi.org/10.1016/j.enconman.2022.116174.

23. G. Mosetti, C. Poloni, and B. Diviacco, "Optimization of Wind Turbine Positioning in Large Windfarms by Means of a Genetic Algorithm," *Journal of Wind Engineering and Industrial Aerodynamics* 51, no. 1 (1994): 105–116, https://doi.org/10.1016/0167-6105(94)90080-9.

24. A. Emami and P. Noghreh, "New Approach on Optimization in Placement of Wind Turbines Within Wind Farm by Genetic Algorithms," *Renewable Energy* 35, no. 7 (2010): 1559–1564, https://doi.org/10.1016/j.renene.2009.11.026.

25. M. Bastankhah and F. Porté-Agel, "A New Analytical Model for Wind-Turbine Wakes," *Renewable Energy* 70 (2014): 116–123, https://doi.org/10.1016/j.renene.2014.01.002.

26. H. Sun and H. Yang, "Wind Farm Layout and Hub Height Optimization With a Novel Wake Model," *Applied Energy* 348 (2023): 121554, https://doi.org/10.1016/j.apenergy.2023.121554.

27. Z. Lei, S. Gao, Z. Zhang, H. Yang, and H. Li, "A Chaotic Local Search-Based Particle Swarm Optimizer for Large-Scale Complex Wind Farm Layout Optimization," *IEEE/CAA Journal of Automatica Sinica* 10, no. 5 (2023): 1168–1180, https://doi.org/10.1109/jas.2023.123387.

28. Z. Zhang, J. Li, Z. Lei, Q. Zhu, J. Cheng, and S. Gao, "Reinforcement Learning-Based Particle Swarm Optimization for Wind Farm Layout Problems," *Energy* 313 (2024): 134050, https://doi.org/10.1016/j.energy.2024.134050.

29. Y. Yu, T. Zhang, Z. Lei, Y. Wang, H. Yang, and S. Gao, "A Chaotic Local Search-Based LSHADE With Enhanced Memory Storage Mechanism for Wind Farm Layout Optimization," *Applied Soft Computing* 141 (2023): 110306, https://doi.org/10.1016/j.asoc.2023.110306.

30. H. Long, P. Li, and W. Gu, "A Data-Driven Evolutionary Algorithm for Wind Farm Layout Optimization," *Energy* 208 (2020): 118310, https://doi.org/10.1016/j.energy.2020.118310.

31. I. Katic, J. Højstrup, and N. O. Jensen, "A Simple Model for Cluster Efficiency," in *European Wind Energy Association Conference And Exhibition* (1986), 407–410.

32. A. Crespo, J. Hernandez, and S. Frandsen, "Survey of Modelling Methods for Wind Turbine Wakes and Wind Farms," *Wind Energy* 2, no. 1 (1999): 1–24, https://doi.org/10.1002/(sici)1099-1824(199901/03)2:1<1::aid-we16>3.0.co;2-7.

33. S. E. Tuller and A. C. Brett, "The Characteristics of Wind Velocity That Favor the Fitting of a Weibull Distribution in Wind Speed Analysis," *Journal of Applied Meteorology and Climatology* 23, no. 1 (1984): 124–134, https://doi.org/10.1175/1520-0450(1984)023<0124:tcowvt>2.0.co;2.

34. R. J. Stevens, B. F. Hobbs, A. Ramos, and C. Meneveau, "Combining Economic and Fluid Dynamic Models to Determine the Optimal Spacing in Very Large Wind Farms," *Wind Energy* 20, no. 3 (2017): 465–477, https://doi.org/10.1002/we.2016.

35. F. Bai, X. Ju, S. Wang, et al., "Wind Farm Layout Optimization Using Adaptive Evolutionary Algorithm With Monte Carlo Tree Search Reinforcement Learning," *Energy Conversion and Management* 252 (2022): 115047, https://doi.org/10.1016/j.enconman.2021.115047.

36. M. Črepinšek, S. H. Liu, and M. Mernik, "Exploration and Exploitation in Evolutionary Algorithms: A Survey," *ACM Computing Surveys* 45, no. 3 (2013): 1–33, https://doi.org/10.1145/2480741.2480752.

37. Z. Cai, X. Yang, M. Zhou, Z. H. Zhan, and S. Gao, "Toward Explicit Control Between Exploration and Exploitation in Evolutionary Algorithms: A Case Study of Differential Evolution," *Information Sciences* 649 (2023): 119656, https://doi.org/10.1016/j.ins.2023.119656.

38. H. Li, Y. Yang, Y. Wang, et al., "Population Interaction Network in Representative Gravitational Search Algorithms: Logistic Distribution Leads to Worse Performance," *Heliyon* 10, no. 11 (2024): e31631, https://doi.org/10.1016/j.heliyon.2024.e31631.

39. H. Yang, S. Gao, Z. Lei, J. Li, Y. Yu, and Y. Wang, "An Improved Spherical Evolution With Enhanced Exploration Capabilities to Address Wind Farm Layout Optimization Problem," *Engineering Applications of Artificial Intelligence* 123 (2023): 106198, https://doi.org/10.1016/j.engappai.2023.106198.

40. R. Tanabe and A. Fukunaga, "Success-History Based Parameter Adaptation for Differential Evolution," in *2013 IEEE Congress on Evolutionary Computation* (2013), 71–78.

41. R. Tanabe and A. S. Fukunaga, "Improving the Search Performance of SHADE Using Linear Population Size Reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)* (2014), 1658–1665.

42. J. Brest, M. S. Maučec, and B. Bošković, "Single Objective Real-Parameter Optimization: Algorithm JSO," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (2017), 1311–1318.

43. A. K. Jain, "Data Clustering: 50 Years Beyond K-means," *Pattern Recognition Letters* 31, no. 8 (2010): 651–666, https://doi.org/10.1016/j.patrec.2009.09.011.

44. Y. Yang, S. Tao, H. Li, H. Yang, and Z. Tang, "A Multi-Local Search-Based SHADE for Wind Farm Layout Optimization," *Electronics* 13, no. 16 (2024): 3196, https://doi.org/10.3390/electronics13163196.

45. X. Ju and F. Liu, "Wind Farm Layout Optimization Using Self-Informed Genetic Algorithm With Information Guided Exploitation," *Applied Energy* 248 (2019): 429–445, https://doi.org/10.1016/j.apenergy.2019.04.084.

46. X. Ju, F. Liu, L. Wang, and W. J. Lee, "Wind Farm Layout Optimization Based on Support Vector Regression Guided Genetic Algorithm With Consideration of Participation Among Landowners," *Energy Conversion and Management* 196 (2019): 1267–1281, https://doi.org/10.1016/j.enconman.2019.06.082.

47. J. Li, Z. Zhang, T. Zheng, J. Tang, Z. Lei, and S. Gao, "Discrete Bi-Population Differential Evolution for Optimizing Complex Wind Farm Layouts in Diverse Terrains," *Energy* 335 (2025): 137885, https://doi.org/10.1016/j.energy.2025.137885.