

Article

A Multi-Local Search-Based SHADE for Wind Farm Layout Optimization

Yifei Yang ^{1,†} , Sichen Tao ^{2,†} , Haotian Li ² , Haichuan Yang ^{3,*}  and Zheng Tang ^{2,*} ¹ Faculty of Science and Technology, Hirosaki University, Hirosaki-shi 036-8560, Japan; yyf7236@hirosaki-u.ac.jp² Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan; d2278002@ems.u-toyama.ac.jp (S.T.); d2372011@ems.u-toyama.ac.jp (H.L.)³ Graduate School of Technology, Industrial and Social Sciences, Tokushima University, Tokushima 770-8506, Japan

* Correspondence: you.kaisen@tokushima-u.ac.jp (H.Y.); ztang@eng.u-toyama.ac.jp (Z.T.)

† These authors contributed equally to this work.

Abstract: Wind farm layout optimization (WFLO) is focused on utilizing algorithms to devise a more rational turbine layout, ultimately maximizing power generation efficiency. Traditionally, genetic algorithms have been frequently employed in WFLO due to the inherently discrete nature of the problem. However, in recent years, researchers have shifted towards enhancing continuous optimization algorithms and incorporating constraints to address WFLO challenges. This approach has shown remarkable promise, outperforming traditional genetic algorithms and gaining traction among researchers. To further elevate the performance of continuous optimization algorithms in the context of WFLO, we introduce a multi-local search-based SHADE, termed MS-SHADE. MS-SHADE is designed to fine-tune the trade-off between convergence speed and algorithmic diversity, reducing the likelihood of convergence stagnation in WFLO scenarios. To assess the effectiveness of MS-SHADE, we employed a more extensive and intricate wind condition model in our experiments. In a set of 16 problems, MS-SHADE's average utilization efficiency improved by 0.14% compared to the best algorithm, while the optimal utilization efficiency increased by 0.3%. The results unequivocally demonstrate that MS-SHADE surpasses state-of-the-art WFLO algorithms by a significant margin.



Citation: Yang, Y.; Tao, S.; Li, H.; Yang, H.; Tang, Z. A Multi-Local Search-Based SHADE for Wind Farm Layout Optimization. *Electronics* **2024**, *13*, 3196. <https://doi.org/10.3390/electronics13163196>

Academic Editor: Ahmed Abu-Siada

Received: 11 July 2024

Revised: 8 August 2024

Accepted: 9 August 2024

Published: 13 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: differential evolution; green energy; wind farm layout optimization; large scale

1. Introduction

The process of global warming has accelerated significantly during this century, primarily due to the substantial carbon emissions linked to industrialization in developing countries [1]. While environmental protection is consistently highlighted as a critical United Nations issue, the Earth's greenhouse effect continues to escalate [2]. The predominant driver of current climate change undeniably remains the excessive use of fossil fuels, which are a major source of carbon emissions [3]. To effectively combat global warming, it is imperative for humanity to transition towards the widespread adoption of clean energy sources [4]. There is a wealth of safe and dependable clean energy alternatives, including wind, solar, and tidal power. However, the utilization of these energy sources in actual production lags far behind that of fossil fuels [5]. The fundamental obstacle lies in the fact that clean energy, despite its environmental benefits, often fails to generate comparable revenue for businesses [6]. Given the choice, individuals and organizations tend to opt for cheaper energy options, relegating clean energy, with its higher initial costs and lower energy production capacity compared to fossil fuels, to a secondary role in power generation [7]. Therefore, to realize the goal of reducing carbon emissions, the cost of clean energy generation should continuously be driven down and the efficiency of power production enhanced.

Wind farm layout optimization (WFLO) is an optimization problem for the design of wind turbine layout based on the calculation of the wake effect model [8]. When the wind

turbine itself is unchanged, and the wind direction and wind speed of the wind farm have a certain regularity, a better layout can bring higher power generation efficiency [9]. Jensens' wake effect model is widely used to solve the WFLO problem [10]. The model predicts the energy content of wind farms to better estimate their generating capacity. WFLO is generally optimized with discrete models, due to the fact that there must be a certain setup distance between turbines to ensure safety in use [11]. The discrete WFLO model is more simplified, but the difficulty of optimization is increased. Most of the discrete optimization problems still use the genetic algorithm (GA), particle swarm optimization (PSO), and their improved algorithms that were proposed decades ago, compared to the various algorithms that are constantly being improved in continuous optimization [12]. To a certain extent, they do achieve good results. For example, Chen et al. used a binary number encoding genetic algorithm to solve the WFLO problem [13]. Ju et al. used support vector regression to enhance the performance of GA on the WFLO problem [14]. Gao et al. adopted a mechanism for PSO that combines GA and an improvement of chaotic mapping to enhance the local search capability on such problems [15]. It is easy to realize through these studies that they are all based on a certain meta-heuristic algorithm (MHA) that is adapted for WFLO to improve and, thus, propose a solution. Since improved MHAs optimize WFLO well, better performance should be obtained based on more advanced MHAs [16,17]. Coding and the changes in the evaluation method are straightforward, but it is hard to think that they will enable MHAs originally used for continuous optimization to perform well on discrete problems. GA and PSO are extremely simple in structure and have better performance compared to methods other than algorithms. That is why these algorithms are often favored in WFLO [18].

The primary distinction in optimizing discrete versus continuous problems using MHAs lies in the range of values that individuals can take. Continuous problems endow the algorithm with infinite differentiability within a defined domain, while discrete problems involve a finite set of points [19]. In the realm of continuous optimization problems, algorithms converging to a local optimum often witness an increase in their search step size [20]. MHAs commonly integrate a mechanism for greedy strategy, empowering individuals in each generation to explore more promising directions and enhance exploitation efficiency [21]. Moreover, the retention criterion typically demands that only one individual needs to discover a superior solution in the solution space compared to the previous generation. However, this mechanism introduces a challenge in discrete optimization; local optimal solutions are considerably more entrenched [22]. When an individual is far from an optimal solution in a discrete optimization problem, it is likely to remain unchanged due to the presence of a greedy mechanism if there is no relatively better solution between the individual and optimal solution. Conversely, if the problem is continuous, the individual will always have the potential to find a relatively better solution at a point between its current position and the optimal solution, thus allowing it to move closer to the optimal solution. That is, the discrete nature of the problem complicates determining whether the current location is worth exploiting based on convergence trends. The endeavor of us to optimize MHAs for WFLO is trying to address the aforementioned challenges. This study provides a valuable paradigm for enhancing continuous optimization algorithms when dealing with discrete optimization problems—specifically, balancing the influence of the optimal individual on the population.

In continuous optimization problems, the differential evolution (DE) algorithm is highly regarded for its exceptional performance across single-objective, multi-objective, and large-scale optimization challenges, positioning it at the apex of optimization algorithms [23–25]. However, DE's computational nature hampers its effectiveness in discrete problems, where algorithms like GA tend to excel. Fundamentally, DE boasts formidable exploitation capability, facilitating swift convergence compared to most algorithms [26]. Yet, within benchmark test sets like IEEE CEC (Congress on Evolutionary Computation), the enhancements made to DE predominantly focus on fortifying its exploitation capacity to align with the challenges in these test sets [27,28]. Hence, when optimizing DE for WFLO, there

arises a necessity to bolster its exploratory functionalities while preserving its strengths in exploitation capabilities. By ensuring DE's enhanced exploitation capability, its remarkable prowess in local search can yield substantially superior performance in WFLO compared to algorithms such as PSO or GA. Since the improvements in this study will primarily target the population structure, using SHADE as a basis is advantageous. Using SHADE (success-history-based parameter adaptation for differential evolution) as an improvement target aligns well with our approach and helps to effectively implement our improvement method [29], because past studies have shown that test set improvement algorithms based on SHADE often have a varying population size, which can be detrimental to the design of our population structure [30–33].

Building upon the foundations of the sophisticated continuous optimization algorithm, with DEs as a cornerstone, this paper introduces a multi-local search-based SHADE referred to as MS-SHADE, which has demonstrated exceptional performance in WFLO. Specifically, we modify the original single operator to a triple operator with varying degrees of exploitation capability. By rationally allocating the number of populations, individuals within the population are assigned different roles, thereby balancing the exploitation and exploration abilities. The enhancements, rigorously tested under diverse and intricate wind conditions, exhibit superior performance compared to previous studies.

The main contributions of this paper are summarized as follows:

- (1) In this paper, we conduct a thorough analysis of the problem itself and enhance the model to incorporate realistic wind conditions.
- (2) MS-SHADE combines multiple operators, providing a balance between algorithm exploitation and exploration.
- (3) The proposed MS-SHADE performs well on WFLO.

In Section 2, the related methodology of MS-SHADE is introduced. Section 3 summarizes detailed data and comparison experiments. Finally, the conclusion of this paper is given in Section 4.

2. Methodology

WFLO comprises two components: the modeling of the problem and the optimization method. The modeling component includes a wind field model to simulate wind conditions and terrain, as well as a wake model to simulate turbine wakes.

2.1. Modeling

For the wake model, we employed Jensen's Wake Effect model for two primary reasons [34]. Firstly, the computational process of this model is simpler, facilitating a streamlined experimental process while allowing for the evaluation of performance differences among various algorithms [35–38]. Secondly, the model has been widely employed in numerous mainstream studies, making comparisons under identical conditions more convincing.

Figure 1 shows a single wake model of Jensen; the wind velocity V_0 and the rotor radius of wind turbine L_W are specified. The near-field wake wind speed directly behind the turbine is designated as V_1 , while the wake wind speed in the far-field is designated as V_2 . The wake radius L_X , is calculated based on the extent to which the wake propagates. By applying the law of conservation of momentum, we derive the relationship between L and V as presented in Equation (1):

$$\pi \cdot L_W^2 \cdot V_1 + \pi \cdot (L_X^2 - L_W^2) \cdot V_0 = \pi \cdot L_X^2 \cdot V_2, \quad (1)$$

Based on trigonometric functions, one can derive Equation (2). According to Betz's theory, the value of V_1 relative to V_0 is expressed as shown in Equation (3) [39]. The dimensionless scalar θ , identified in Equation (4), as the decay constant, determines the rate at which the wake expands with distance, thereby describing the growth of the wake width. The determination of θ is sensitive to factors including ambient turbulence, turbine-induced

turbulence, and atmospheric stability. z denotes the wind turbine's hub height, while z_0 represents the surface roughness of the wind farm area. The parameter is crucial for the decay coefficient calculation. The decay constant typically assumes a default value of 0.075 in most land cases. For offshore applications, a value of 0.04 is recommended [40].

$$L_X = L_W + X \cdot \tan\theta, \quad (2)$$

$$V_1 = (1 - a)V_0, \quad (3)$$

$$\theta = \frac{0.5}{\ln\left(\frac{z}{z_0}\right)}, \quad (4)$$

The desired wake wind speed can be derived to

$$V_2 = \frac{L_W^2(1 - a)V_0 + \left((L_W + X \cdot \tan\frac{0.5}{\ln(\frac{z}{z_0})})^2 - L_W^2\right)V_0}{L_W + X \cdot \tan\left(\frac{0.5}{\ln(\frac{z}{z_0})}\right)^2}. \quad (5)$$

In our wind condition modeling, we implemented significant changes compared to previous studies. Previous studies typically included one-direction, two-direction, and symmetrical wind directions. Considering the unstable nature of the tropospheric atmosphere due to monsoon changes, our wind model incorporates winds from multiple directions (four or more) at random angles. Regarding wind speeds, past studies often assumed normally distributed wind conditions, with some employing integer values for wind speeds, which clearly oversimplifies the actual variability of wind conditions. We defined the wind speed distribution using a Weibull distribution, with a mean value of 13 m/s—reflecting the average wind speed in Japan—based on related studies. We then sampled this distribution 100,000 times using a continuous function to construct a realistic wind speed sample [41]. Figure 2 illustrates the continuous curve of wind speeds under the influence of both the traditional normal distribution and the Weibull distribution that we used. In this case, the overall wind speed is more in line with the Weibull distribution and is closer to the scenario where the average wind speed is 13 m/s. Conversely, the publicly available wind speed model used in past studies did not achieve the set average wind speed during our validation. Even the modified model slightly deviates from the true wind speed by averaging the probabilities of high and low wind speeds. We are not dismissing past studies that used similar models, as their methods have demonstrated validity in the experiments conducted in this paper. However, by improving the wind field model to make it more realistic, we believe that it holds obvious value for practical applications. Furthermore, wind environments vary from region to region. This paper builds a more appropriate model based on publicly available information from the meteorological departments of the corresponding regions, thereby enhancing the effectiveness of our method.

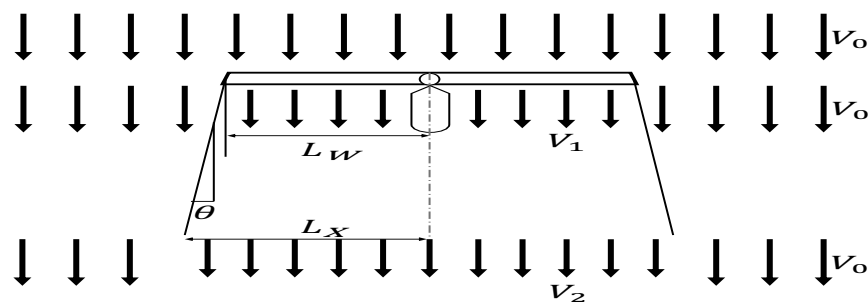


Figure 1. Jensen's single wake model.

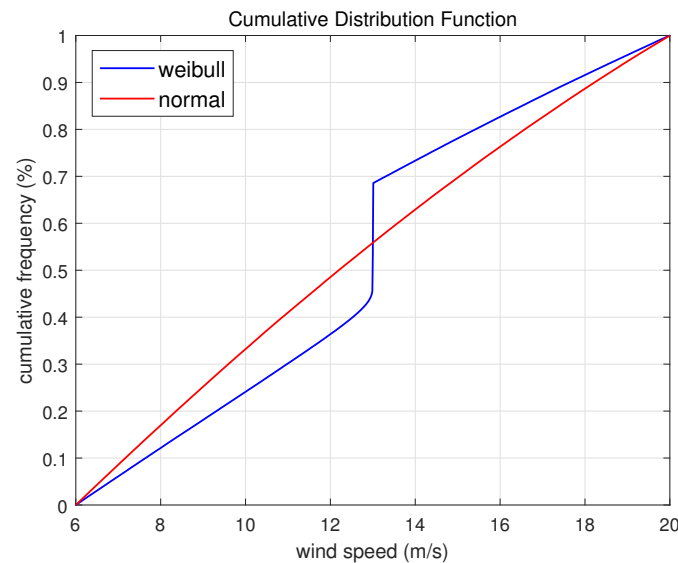


Figure 2. The wind speed in different distributions.

2.2. Differential Evolution Algorithms

DE was the pioneering algorithm that introduced the difference formula into the field of evolutionary computation [42]. All subsequent DE algorithms have relied heavily on its foundational operators and strategies. Contemporary mainstream DE algorithms employ a similar operator, which utilizes a first-order difference formula involving five distinct individual components:

$$V_{i,j} = X_{r1,j}^t + F \cdot (X_{r2,j}^t - X_{r3,j}^t + X_{r4,j}^t - X_{r5,j}^t), \quad (6)$$

$$U_{i,j} = \begin{cases} V_{i,j}, & \text{if } rand < CR \\ X_{i,j}^t, & \text{otherwise} \end{cases} \quad (7)$$

where F is the scaling factor, and CR is the variance factor, which determines which dimensional values are retained in the individual. $rand$ indicates a random number. t represents the current generation, and i, j denotes the j -th dimension in the i -th individual of the current generation. The logic of the operator is to obtain the values of all dimensions for every individual in V via this formula. New individuals are generated by incorporating a portion of the values from both the original and the fully mutated individuals, as determined by CR . The expression of DE, when employing a greedy strategy for solving minimization problems, can be stated as

$$X_{i,j}^{t+1} = \begin{cases} U_{i,j}, & \text{if } fit_{U_{i,j}} < fit_{X_{i,j}^t} \\ X_{i,j}^t, & \text{otherwise} \end{cases} \quad (8)$$

where fit denotes the solution obtained by solving the current individual through the evaluation function. Most enhanced DE algorithms implement an adaptive selection of the parameters F and CR , along with external archiving of individuals. The parameter design of the improved SHADE algorithm, which utilizes success history information, is adopted in this study.

2.3. The Proposed MS-SHADE

As SHADE was originally designed for the IEEE CEC2013 competition, the first step involves making improvements to the platform. This competition, hosted by IEEE, is for bounded unconstrained single-objective continuous optimization, and all the problems addressed are functional problems. The boundary detection component for individuals

encompasses two modules: the detection of computational boundaries and the detection of duplicate individuals. For example, in the case of a simple 4×4 wind farm equipped with four turbines, a one-dimensional vector is obtained by vectorially expanding the matrix representing the layout in Figure 3. The turbine location numbers are recorded to obtain the vector $[1, 6, 11, 16]$. The length of the vector corresponds to the number of turbines, and each value within the vector represents the position of a turbine. Using this vector as an individual for the iteration of the algorithm, the upper and lower bounds of the individual are set at the maximum value counted in the wind field vector and 1, respectively, while the number of dimension of the individual corresponds to the number of turbines.

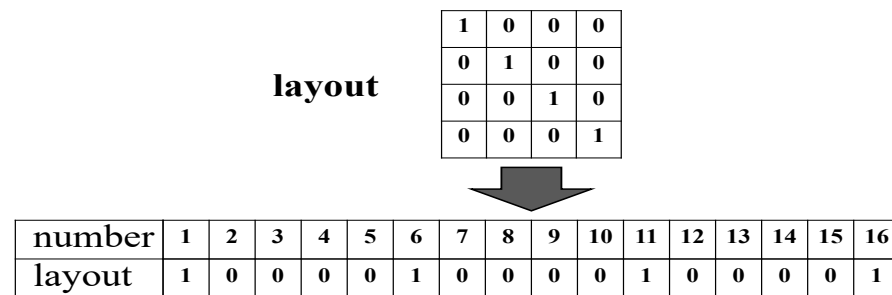


Figure 3. One-dimensional vector layout.

In MS-SHADE, duplicate individuals are detected and updated through proximity updating. Illustrated in Figure 4 is the method used in this study for converting real-number position information into integer position information in wind farm problems. Specifically, real numbers are first rounded off. Due to repeated position information resulting from rounding, when positions overlap, the turbines positioned later in the sequence are shifted one unit backward. Therefore, the final position information will be as shown in the figure, converting from positions with decimals to non-repeating integer positions.

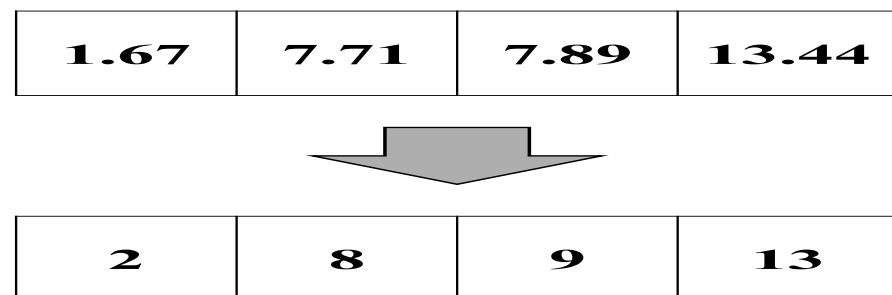


Figure 4. Detection of duplicate individuals.

MS-SHADE introduces a multi-operator combination approach, employing a roulette wheel selection mechanism based on the DE operator. We utilize three distinct sets of operators: a randomized operator with superior exploration capabilities, an operator guided by both the current and global optima, and an operator solely guided by the current optimum. These operators have been featured in various studies previously:

$$V_{i,j} = \begin{cases} X_{i,j}^t + F \cdot (X_{r1,j}^t - X_{r2,j}^t), \\ X_{i,j}^t + F \cdot (X_{pbest,j}^t - X_{i,j}^t + X_{r1,j}^t - X_{all,j}^t), \\ X_{i,j}^t + F \cdot (X_{gbest,j}^t - X_{i,j}^t + X_{pbest,j}^t - X_{i,j}^t). \end{cases} \quad (9)$$

According to the three formulas of Equation (9), the population pop is divided into three groups of pop_{rand} , pop_{pbest} , and pop_{gbest} . The cumulative distributions occupied by the three populations in the roulette wheel are C_{rand} , C_{pbest} , and C_{gbest} .

Regarding the update of parameters F and CR , the design presented in SHADE is followed. F and CR are updated by following the Cauchy and normal distributions, respectively, while also keeping track of the successful parameter history:

$$\begin{aligned} F_i &= \mathcal{C}(M_{F,r}, 0.1) \\ CR_i &= \mathcal{N}(M_{C,r}, 0.1) \end{aligned} \quad (10)$$

where \mathcal{C} and \mathcal{N} denote the Cauchy and normal distributions, respectively. $M_{F,r}$ and $M_{C,r}$ represent the memory archive of F and CR . These adaptive parameters play a significant role in balancing the exploitation and exploration capabilities of SHADE, enabling it to achieve excellent performance on the standard test set. The memory archive is updated according to Equation (10):

$$\begin{aligned} M_{F,r} &= (1 - c) \cdot M_{F,r} + c \cdot \text{mean}_L(F_1, F_2, \dots, F_N) \\ M_{C,r} &= (1 - c) \cdot M_{C,r} + c \cdot \text{mean}_A(CR_1, CR_2, \dots, CR_N) \end{aligned} \quad (11)$$

where c is a random number sampled from the interval $[0.05, 0.2]$, and mean_A represents the arithmetic mean's root. The update equation for mean_L is as follows:

$$\text{mean}_L(F_1, F_2, \dots, F_G) = \frac{\sum_F \cdot F_i^2}{\sum_F \cdot F_i} \quad (12)$$

finally, the aforementioned formulas and mechanisms are combined to form the complete MS-SHADE and shown as pseudo-code in Algorithm 1.

Algorithm 1: Pseudo-code of MS-SHADE.

```

1 /*Initialization */
2 Initialize  $pop$  and obtain  $fit$ .
3 Initialize parameter  $F$  and  $CR$ .
4 Initialize  $C_{rand}$ ,  $C_{pbest}$ , and  $C_{gbest}$ .
5 /*Main loop */
6 while  $nFES < FES$  do
7   Group population by  $C_{rand}$ ,  $C_{pbest}$ , and  $C_{gbest}$ .
8   for  $i = 1 : N$  do
9      $V_i \leftarrow$  obtain vector by Equation (9).
10     $U_i \leftarrow$  obtain new individual by Equation (7).
11     $U_i \leftarrow$  boundary detection.
12     $fit_{U_i} \leftarrow$  evaluation  $U_i$ .
13     $X_i \leftarrow$  check new individual by Equation (8).
14     $fit_{X_i} = fit_{U_i}$ .
15    $F$  and  $CR \leftarrow$  update parameters by Equations (10)–(12).
16 /*Output the optimal */

```

3. Results

This section includes a comparison of the results of MS-SHADE with the state-of-the-art single-target WFLO algorithms, as well as parameter tuning and ablation experiments. The parameters for the other algorithms in the experiments were set according to their originally determined optimal values. In MS-SHADE, C_{rand} , C_{pbest} , and C_{gbest} were set to 0.1, 0.8, and 0.1, and the size of population was set to $0.5 \times D$. D stands for dimension, which refers to the number of turbines in this paper. The wind field size was set to 12×12 with no terrain constraints. The number of turbines was considered in four scenarios: 20, 30, 40, and 50 turbines. The number of evaluations was set to 24,000, experiments were conducted on a single computer, and all algorithms were run 25 times to obtain results.

3.1. Wind Condition Setting

In the experiments, four wind conditions with random wind directions and wind speeds following the Weibull distribution were utilized. As shown in Figure 5, WS1 represents the scenario with two wind directions, WS2 represents three wind directions, WS3 represents four wind directions, and WS4 represents five wind directions. The non-uniform sampling of wind speeds in each wind direction aims to better simulate complex and realistic wind conditions, thereby making our problem more challenging compared to the uniformly distributed problems commonly addressed in previous studies.

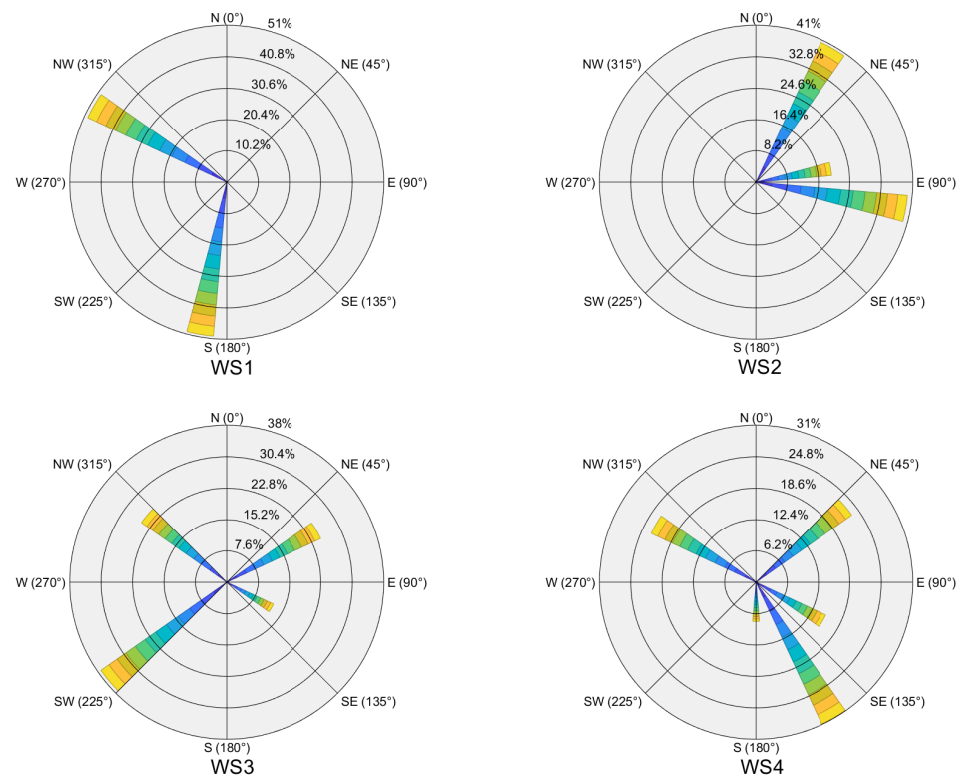


Figure 5. The wind rose of wind scenarios in experiments.

3.2. Comparison of MS-SHADE with Advanced Algorithms

“WS1tn20” refers to the layout results for 20 turbines under WS1 wind conditions, tn30 refers to the layout results for 30 turbines, and so on. “mean” denotes the average of the results from 25 runs, and “std” represents the standard deviation of the results from the 25 runs. “W/T/L” refers to the comparison results of MS-SHADE with other algorithms based on the Wilcoxon rank-sum test, presented as win/tie/loss. Bolded results indicate the best performing set of means. Among the compared algorithms, LSHADE [43] is an improved heuristic algorithm based on SHADE. AGA [14] is a binary algorithm derived from the GA, specifically enhanced for WFLO. AGPSO [44] is a highly efficient WFLO algorithm that combines the GA with PSO. CGPSO [15], currently the most effective WFLO algorithm, builds on AGPSO by incorporating improvements from chaos mapping. The data presented in the experiment represent the actual total power obtained from the model, divided by the total power ceiling, to calculate the power generation efficiency of the turbine. Regarding the consumption of computational resources, MS-SHADE took 29 min, LSHADE took 30 min, AGA took 1 h and 7 min, AGPSO took 1 h and 32 min, and CGPSO took 1 h and 35 min. All experiments were conducted on a PC with an i5-13500 processor and 16 GB of memory.

Table 1 shows the experimental results of MS-SHADE with four advanced algorithms in 16 problems. In terms of mean performance, MS-SHADE achieved the best results, with the exception of the WS4tn20. Regarding the W/T/L results, MS-SHADE had only one

draw with AGPSO, winning in all other comparisons. The experimental results demonstrate that MS-SHADE is currently one of the top-performing algorithms in WFLO.

Table 1. Experimental results of MS-SHADE with advanced algorithms.

	MS-SHADE		AGPSO		CGPSO		LSHADE		AGA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
WS1tn20	99.876%	0.266%	99.622%	0.180%	99.670%	0.165%	98.592%	0.229%	98.536%	0.182%
WS1tn30	97.599%	0.355%	97.301%	0.237%	97.363%	0.254%	94.410%	0.350%	95.775%	0.215%
WS1tn40	94.490%	0.525%	94.210%	0.316%	94.230%	0.315%	89.638%	0.304%	92.618%	0.316%
WS1tn50	91.040%	1.004%	90.982%	0.359%	90.922%	0.425%	85.512%	0.439%	89.163%	0.284%
WS2tn20	99.871%	0.109%	99.341%	0.258%	99.331%	0.261%	98.135%	0.163%	98.096%	0.201%
WS2tn30	97.229%	0.656%	96.772%	0.337%	96.862%	0.287%	93.979%	0.548%	95.198%	0.209%
WS2tn40	94.194%	0.690%	93.912%	0.290%	93.782%	0.257%	90.219%	0.433%	92.028%	0.321%
WS2tn50	90.936%	0.766%	90.776%	0.374%	90.846%	0.309%	86.767%	0.267%	88.399%	0.265%
WS3tn20	99.535%	0.271%	99.220%	0.313%	99.309%	0.319%	97.681%	0.225%	98.165%	0.241%
WS3tn30	97.335%	0.210%	96.964%	0.360%	96.832%	0.419%	93.296%	0.414%	95.512%	0.325%
WS3tn40	94.267%	0.217%	93.641%	0.600%	93.932%	0.389%	88.368%	0.387%	92.335%	0.348%
WS3tn50	90.661%	0.270%	90.347%	0.655%	90.332%	0.496%	83.950%	0.296%	88.897%	0.277%
WS4tn20	98.699%	0.727%	98.695%	0.297%	98.753%	0.241%	97.001%	0.161%	97.580%	0.303%
WS4tn30	96.395%	0.232%	96.078%	0.268%	96.028%	0.356%	92.278%	0.327%	94.143%	0.271%
WS4tn40	93.055%	0.154%	92.654%	0.493%	92.607%	0.371%	87.239%	0.300%	90.414%	0.304%
WS4tn50	89.112%	0.576%	88.705%	0.463%	88.655%	0.557%	82.731%	0.269%	86.435%	0.209%
W/T/L	-/-/-		15/1/0		16/0/0		16/0/0		16/0/0	

Figure 6 shows the average efficiency obtained in WS4. The x-axis represents the number of iterations, and the y-axis represents the average utilization efficiency. As shown in the figure, MS-SHADE demonstrates a significant advantage over all other algorithms during the middle of the iteration. The slight inferiority in mean value compared to CGPSO in the case of 20 wind turbines is due to MS-SHADE occasionally generating a large variance when it falls into a local optimum, adversely affecting the average result. In the scenarios involving the other three turbine numbers, MS-SHADE maintains its lead until the end of the iteration. AGA, being a modification of the genetic algorithm, performs poorly because the optimization efficiency of the genetic algorithm has become outdated. On the other hand, LSHADE, although designed as a heuristic algorithm to search for the global optimum, is significantly weaker in optimization compared to algorithms specifically improved for WFLO. The same applies to SHADE, the underlying algorithm of LSHADE.

Table 2 shows the best results of MS-SHADE with four advanced algorithms in 16 problems. “best” shows the best result and “rank” shows the rank of best result with all algorithms in current problem. “average rank” shows the average rank of 16 problems. It can be observed that MS-SHADE achieves the best optimal results in 10 problems. In five problems, MS-SHADE attains the second-best result, and it underperforms both AGPSO and CGPSO in only one problem. In the highest-ranked problem, MS-SHADE shows a significant improvement and also leads the other algorithms in the average rankings by a relatively large margin.

Figures 7 and 8 show the layout solution visualizations of all competitors in Wind Scenario 1. MS-SHADE ranked first in both problems, while CGPSO ranked first in WS1tn20 and second in WS1tn50. The layout diagrams reveal that the two best-performing algorithms can achieve 100% efficiency, even with different layouts in the simpler scenario with 20 turbines. However, when the number of turbines increases to 50, the layout schemes generated by each algorithm result in significant differences in efficiency. LSHADE’s layout appears more haphazard and achieves the lowest efficiency. AGA’s results are 3% higher than those of LSHADE, featuring a scheme with turbines placed on the top and bottom sides and utilizing the turbines in the middle to benefit from the wake flow. AGPSO’s layout better leverages the wake flow by interleaving the turbines in the middle and on the top and bottom sides, while CGPSO reduces the number of turbines set up on the bottom

side. MS-SHADE, the most efficient algorithm, clearly maximizes the utilization of wind on the lower side and staggers the turbines on the left and right sides. From the layout results, it is evident that MS-SHADE has made significant improvements in layout details compared to previous advanced algorithms.

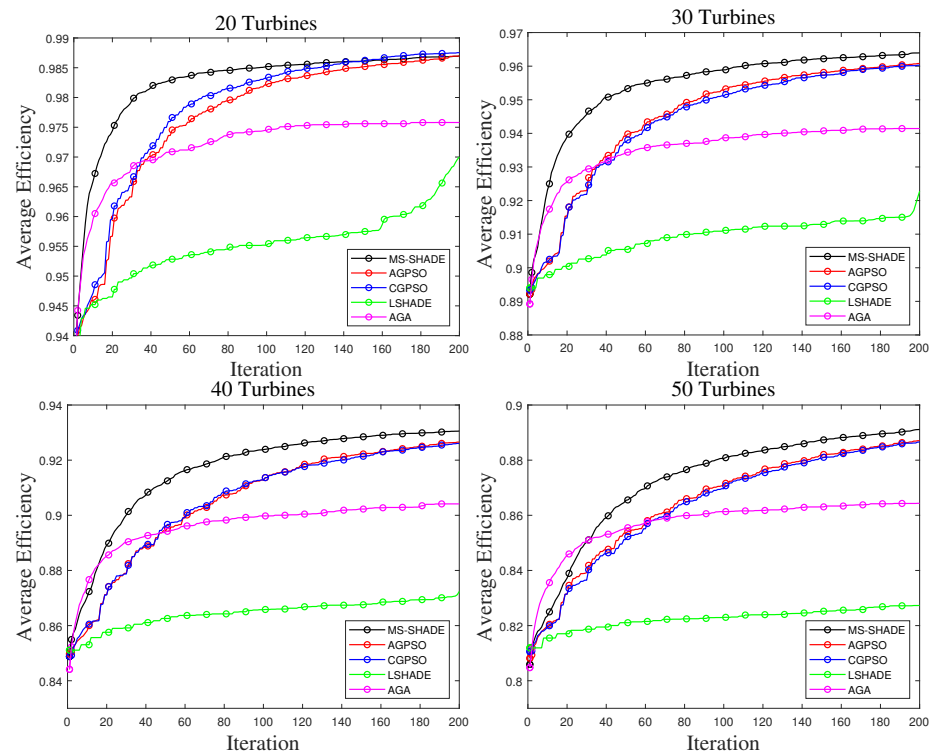


Figure 6. Average efficiency of MS-SHADE with advanced algorithms under WS4.

Table 2. Best results of MS-SHADE with advanced algorithms.

	MS-SHADE		AGPSO		CGPSO		LSHADE		AGA	
	Best	Rank	Best	Rank	Best	Rank	Best	Rank	Best	Rank
WS1tn20	100.000%	1	99.948%	3	100.000%	1	99.284%	4	98.923%	5
WS1tn30	97.981%	2	97.680%	3	98.015%	1	95.009%	5	96.430%	4
WS1tn40	94.892%	1	94.687%	3	94.722%	2	90.089%	5	93.291%	4
WS1tn50	91.788%	1	91.614%	3	91.630%	2	86.723%	5	89.786%	4
WS2tn20	100.000%	1	100.000%	1	99.831%	3	98.413%	5	98.668%	4
WS2tn30	97.669%	1	97.312%	2	97.293%	3	94.899%	5	95.651%	4
WS2tn40	94.989%	1	94.340%	2	94.300%	3	91.555%	5	92.617%	4
WS2tn50	91.463%	2	91.385%	3	91.555%	1	87.281%	5	88.895%	4
WS3tn20	99.936%	3	100.000%	1	100.000%	1	98.111%	5	98.745%	4
WS3tn30	97.864%	1	97.495%	3	97.614%	2	93.973%	5	96.192%	4
WS3tn40	94.793%	1	94.568%	3	94.694%	2	89.076%	5	93.238%	4
WS3tn50	91.059%	2	91.477%	1	91.049%	3	84.575%	5	89.577%	4
WS4tn20	99.479%	1	99.188%	3	99.306%	2	97.353%	5	98.185%	4
WS4tn30	96.949%	1	96.626%	3	96.848%	2	92.913%	5	94.795%	4
WS4tn40	93.315%	2	93.412%	1	93.195%	3	87.866%	5	91.194%	4
WS4tn50	89.702%	2	89.786%	1	89.523%	3	83.188%	5	86.782%	4
average rank		1.4		2.3		2.1		4.9		4.1

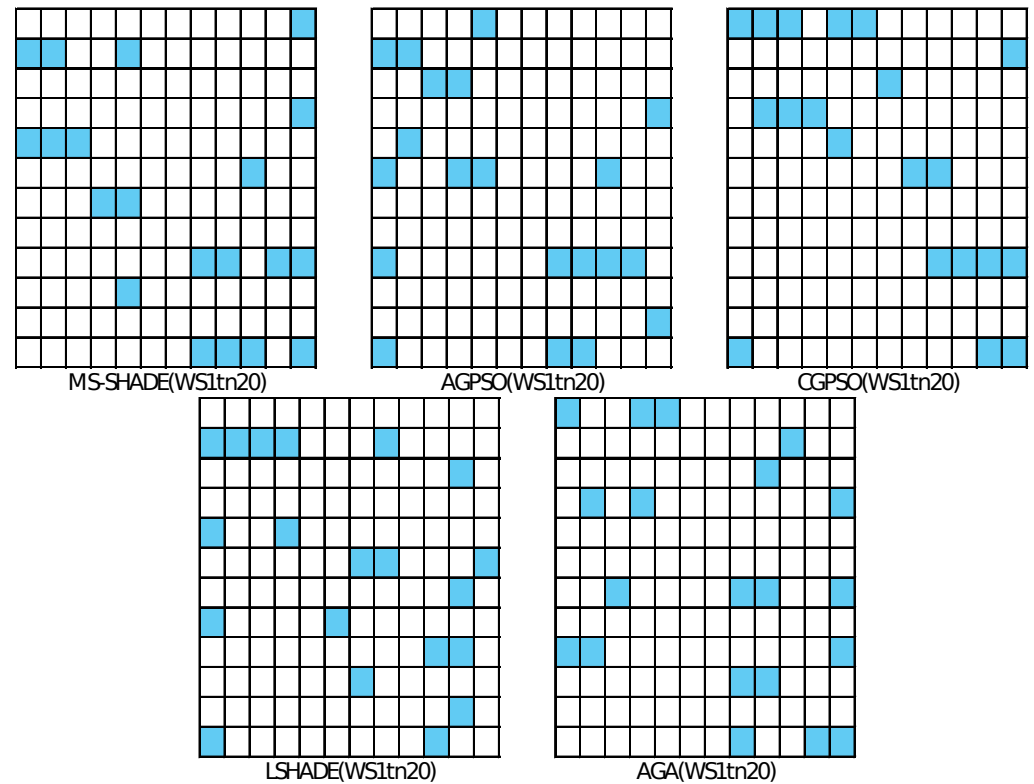


Figure 7. The layout solution visualizations of all competitors in Wind Scenario 1 with 20 turbines.

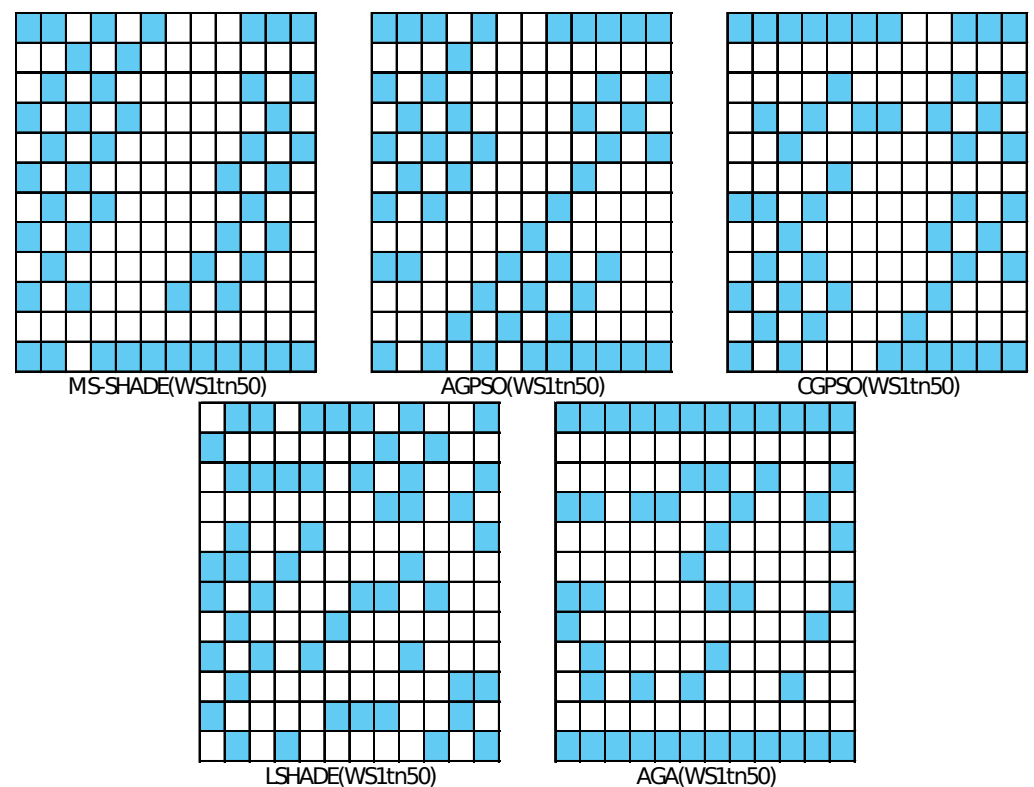


Figure 8. The layout solution visualizations of all competitors in Wind Scenario 1 with 50 turbines.

3.3. Ablation Experiments

Table 3 shows the ablation experiments results of MS-SHADE. MS-SHADE(gbest), MS-SHADE(pbest), and MS-SHADE(rand) mean MS-SHADE with only one operator shown. Among them, MS-SHADE, which utilizes all three operators, performs the best. The

combination of gbest and rand operators ensures a balance between stronger convergence and population diversity. Meanwhile, pbest, as the main operator, plays a crucial role in balancing the algorithm's exploitation and exploration capabilities. In the results, the performance achieved using only the rand operator is noteworthy because the exploration ability is more critical than the exploitation ability in discrete WFLO. Conversely, using only the gbest operator tends to lead to local optima, resulting in worse outcomes compared to pbest in pairwise comparisons. However, when compared with MS-SHADE, the gbest operator demonstrates an advantage over the pbest operator on individual problems. This indicates that each of the three operators has a unique role within the algorithm. In the overall comparison, while the stability of results from any single operator is higher than that of MS-SHADE, the average results are inferior to those achieved by MS-SHADE.

Table 3. Ablation experiments results of MS-SHADE.

	MS-SHADE		MS-SHADE(Gbest)		MS-SHADE(Pbest)		MS-SHADE(Rand)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
WS1tn20	99.876%	0.266%	99.625%	0.598%	99.869%	0.098%	99.914%	0.080%
WS1tn30	97.599%	0.355%	97.109%	1.016%	97.271%	0.646%	97.656%	0.117%
WS1tn40	94.490%	0.525%	93.678%	1.430%	94.279%	0.174%	94.509%	0.151%
WS1tn50	91.040%	1.004%	90.762%	0.986%	90.687%	0.166%	91.188%	0.191%
WS2tn20	99.871%	0.109%	98.654%	1.103%	99.785%	0.123%	99.857%	0.100%
WS2tn30	97.229%	0.656%	95.435%	1.334%	97.116%	0.184%	97.149%	0.558%
WS2tn40	94.194%	0.690%	92.123%	1.199%	93.772%	0.280%	94.365%	0.198%
WS2tn50	90.936%	0.766%	88.459%	0.606%	89.902%	0.786%	91.156%	0.148%
WS3tn20	99.535%	0.271%	99.064%	0.804%	99.518%	0.242%	99.381%	0.165%
WS3tn30	97.335%	0.210%	96.524%	1.380%	97.068%	0.205%	97.163%	0.146%
WS3tn40	94.267%	0.217%	92.959%	1.489%	93.668%	0.198%	94.016%	0.263%
WS3tn50	90.661%	0.270%	89.586%	1.570%	89.994%	0.158%	90.505%	0.317%
WS4tn20	98.699%	0.727%	98.727%	0.540%	98.826%	0.112%	98.906%	0.123%
WS4tn30	96.395%	0.232%	95.619%	1.451%	96.165%	0.275%	96.504%	0.274%
WS4tn40	93.055%	0.154%	92.300%	1.234%	92.592%	0.251%	93.023%	0.219%
WS4tn50	89.112%	0.576%	88.187%	1.700%	88.387%	0.316%	89.123%	0.212%
W/T/L	-/-/-		13/3/0		14/2/0		5/11/0	

3.4. Parameter Tuning Experiments

Based on the results obtained from the ablation experiments, only the scenario where pbest is used proportionally is demonstrated in the parametric experiments, with gbest fixed at 0.1. Therefore, when C_{pbest} is 0.7, C_{rand} is 0.2. Although the results shown in the rank-sum test are close to MS-SHADE, the overall mean performance and stability deteriorate as the percentage of the rand operator increases. This is because, when multiple operators are used in combination, the individuals produced by the operators influence each other, altering the network composition among the individuals, which significantly differs from the single-operator scenario. In summary, the experimental results clearly indicate that the parameter settings of MS-SHADE are the most suitable for this problem (Table 4).

Table 4. Parameter tuning experiments results of MS-SHADE.

	MS-SHADE		MS-SHADE($C_{Pbest} = 0.7$)		MS-SHADE($C_{Pbest} = 0.6$)		MS-SHADE($C_{Pbest} = 0.5$)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
WS1tn20	99.876%	0.266%	99.676%	0.644%	99.760%	0.465%	99.756%	0.449%
WS1tn30	97.599%	0.355%	97.602%	0.141%	97.454%	0.526%	97.594%	0.136%
WS1tn40	94.490%	0.525%	94.551%	0.192%	94.568%	0.177%	94.408%	0.556%
WS1tn50	91.040%	1.004%	91.267%	0.166%	91.258%	0.214%	91.147%	0.689%

Table 4. Cont.

	MS-SHADE		MS-SHADE(C_{Pbest} = 0.7)		MS-SHADE(C_{Pbest} = 0.6)		MS-SHADE(C_{Pbest} = 0.5)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
WS2tn20	99.871%	0.109%	99.699%	0.571%	99.774%	0.285%	99.597%	0.594%
WS2tn30	97.229%	0.656%	97.339%	0.194%	97.148%	0.684%	97.276%	0.536%
WS2tn40	94.194%	0.690%	93.607%	1.198%	93.197%	1.134%	93.028%	1.275%
WS2tn50	90.936%	0.766%	89.899%	1.259%	89.692%	0.921%	89.550%	0.968%
WS3tn20	99.535%	0.271%	99.501%	0.261%	99.495%	0.248%	99.397%	0.466%
WS3tn30	97.335%	0.210%	97.241%	0.175%	97.124%	0.392%	97.162%	0.316%
WS3tn40	94.267%	0.217%	94.146%	0.295%	94.062%	0.548%	94.049%	0.660%
WS3tn50	90.661%	0.270%	90.670%	0.221%	90.600%	0.551%	90.652%	0.440%
WS4tn20	98.699%	0.727%	98.857%	0.305%	98.819%	0.308%	98.914%	0.177%
WS4tn30	96.395%	0.232%	96.431%	0.280%	96.423%	0.283%	96.426%	0.226%
WS4tn40	93.055%	0.154%	93.077%	0.186%	92.963%	0.743%	92.985%	0.207%
WS4tn50	89.112%	0.576%	89.101%	0.515%	89.043%	0.501%	88.993%	0.780%
W/T/L	-/-/-		5/11/0		3/13/0		5/11/0	

3.5. Supplementary Experiments under the Standard Test Set

To verify the necessity of proposing MS-SHADE, as a supplementary experiment, we used the classic IEEE CEC2017 as a test set to validate the general optimization capabilities of the algorithm. Table 5 shows the results of the experiment, where rank is the average ranking result based on the Friedman test. It can be seen that MS-SHADE, as a general-purpose algorithm, also possesses considerable optimization capabilities. As a meta-heuristic algorithm, LSHADE naturally achieves better results compared to other algorithms. However, LSHADE's performance in WFLO is ranked the lowest, indicating that the direct application of meta-heuristic algorithms to practical problems is often unsatisfactory. AGPSO and CGPSO, previously considered most suitable for solving WFLO, show similar performance in the test set but are significantly outperformed by MS-SHADE. In conclusion, MS-SHADE, designed specifically for optimizing WFLO, retains a certain level of general optimization capability and is undoubtedly an effective optimization tool.

Table 5. Supplementary experiments under IEEE CEC2017.

	MS-SHADE		AGPSO		CGPSO		LSHADE		AGA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	6.05×10^{-12}	1.62×10^{-11}	1.86×10^3	2.18×10^3	2.59×10^3	3.57×10^3	0.00	0.00	1.04×10^{10}	1.09×10^9
F2	7.80×10^{-9}	2.52×10^{-8}	5.20×10^{12}	1.59×10^{13}	4.47×10^{12}	1.21×10^{13}	3.34×10^{-15}	9.25×10^{-15}	1.00×10^{30}	1.42×10^{14}
F3	1.14×10^{-7}	7.91×10^{-7}	1.95×10^4	2.32×10^4	1.65×10^4	1.60×10^4	6.69×10^{-15}	1.85×10^{-14}	3.37×10^4	2.93×10^3
F4	5.33	1.25×10^1	9.73×10^1	4.86×10^1	9.36×10^1	4.81×10^1	5.02×10^1	2.20×10^1	2.19×10^3	2.23×10^2
F5	3.76×10^1	1.24×10^1	6.87×10^1	1.60×10^1	6.45×10^1	1.50×10^1	7.86	1.63	1.95×10^2	1.79×10^1
F6	1.69×10^{-3}	7.61×10^{-3}	1.79×10^{-1}	1.12×10^{-1}	2.05×10^{-1}	1.33×10^{-1}	8.06×10^{-9}	3.26×10^{-8}	5.72×10^1	4.85
F7	6.42×10^1	1.33×10^1	1.28×10^2	2.23×10^1	1.27×10^2	1.95×10^1	3.74×10^1	1.29	3.45×10^2	3.86×10^1
F8	3.99×10^1	1.59×10^1	6.04×10^1	1.36×10^1	6.08×10^1	1.67×10^1	8.11	1.54	1.36×10^2	1.39×10^1
F9	1.73	1.44	3.74×10^2	2.31×10^2	3.49×10^2	2.14×10^2	0.00	0.00	3.10×10^3	5.36×10^2
F10	2.04×10^3	2.26×10^2	2.69×10^3	4.83×10^2	2.74×10^3	4.90×10^2	1.50×10^3	2.23×10^2	5.02×10^3	9.37×10^2
F11	9.26×10^1	2.86×10^1	1.61×10^2	1.43×10^2	1.31×10^2	1.02×10^2	1.83×10^1	2.43×10^1	6.92×10^2	7.78×10^1
F12	7.78×10^3	4.43×10^3	2.51×10^6	2.67×10^6	3.24×10^6	2.96×10^6	9.43×10^2	3.31×10^2	1.50×10^9	2.93×10^8
F13	2.34×10^2	1.25×10^3	1.77×10^4	2.32×10^4	1.17×10^4	1.35×10^4	1.57×10^1	5.36	1.84×10^8	6.96×10^7
F14	8.26×10^1	2.65×10^1	3.33×10^4	5.56×10^4	8.57×10^4	2.35×10^5	2.18×10^1	3.05	1.34×10^5	6.75×10^4
F15	3.31×10^1	2.19×10^1	4.01×10^3	5.88×10^3	6.52×10^3	7.58×10^3	2.28	1.45	1.57×10^4	8.95×10^3
F16	3.75×10^2	1.38×10^2	1.02×10^3	3.11×10^2	1.03×10^3	3.31×10^2	1.71×10^2	9.47×10^1	2.24×10^3	3.32×10^2
F17	5.84×10^1	2.78×10^1	4.43×10^2	1.78×10^2	3.84×10^2	2.04×10^2	3.24×10^1	6.32	7.55×10^2	2.02×10^2
F18	4.21×10^1	2.53×10^1	5.41×10^5	2.21×10^6	1.89×10^5	5.53×10^5	2.19×10^1	1.35	5.34×10^5	3.12×10^5
F19	2.80×10^1	1.65×10^1	6.46×10^3	7.83×10^3	8.26×10^3	8.39×10^3	4.74	1.35	1.24×10^6	8.13×10^5
F20	8.32×10^1	5.79×10^1	4.37×10^2	1.93×10^2	4.39×10^2	1.83×10^2	4.13×10^1	1.89×10^1	5.18×10^2	1.28×10^2
F21	2.32×10^2	2.28×10^1	2.68×10^2	1.89×10^1	2.61×10^2	1.48×10^1	2.07×10^2	1.31	4.41×10^2	2.51×10^1
F22	1.00×10^2	8.88×10^{-1}	6.66×10^2	1.24×10^3	9.04×10^2	1.40×10^3	1.00×10^2	1.44×10^{-14}	2.55×10^3	5.64×10^2

Table 5. Cont.

	MS-SHADE		AGPSO		CGPSO		LSHADE		AGA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F23	4.04×10^2	2.56×10^1	4.43×10^2	2.17×10^1	4.43×10^2	2.32×10^1	3.45×10^2	3.17	9.50×10^2	7.21×10^1
F24	4.44×10^2	7.27×10^1	5.37×10^2	2.82×10^1	5.38×10^2	2.99×10^1	4.21×10^2	1.58	1.08×10^3	6.60×10^1
F25	3.87×10^2	2.50	4.04×10^2	1.74×10^1	4.02×10^2	1.60×10^1	3.87×10^2	3.23×10^{-2}	6.57×10^2	1.65×10^1
F26	5.53×10^2	5.06×10^2	2.07×10^3	6.85×10^2	2.20×10^3	7.20×10^2	8.85×10^2	3.56×10^1	5.21×10^3	7.59×10^2
F27	5.30×10^2	1.21×10^1	5.59×10^2	1.71×10^1	5.65×10^2	2.24×10^1	5.06×10^2	6.85	1.18×10^3	1.08×10^2
F28	3.14×10^2	4.10×10^1	4.22×10^2	6.22×10^1	4.14×10^2	5.73×10^1	3.11×10^2	3.36×10^1	1.21×10^3	7.06×10^1
F29	4.79×10^2	3.48×10^1	8.32×10^2	2.16×10^2	8.63×10^2	2.12×10^2	4.33×10^2	6.39	2.25×10^3	3.04×10^2
F30	2.88×10^3	7.94×10^2	2.94×10^4	2.27×10^4	3.16×10^4	2.84×10^4	2.06×10^3	6.71×10^1	1.30×10^7	3.80×10^6
W/T/L	-/-/-		30/0/0		30/0/0		2/0/28		30/0/0	
rank	1.93		3.50		3.53		1.07		4.97	

4. Conclusions

MS-SHADE is not the first sequential optimization algorithm tailored for WFLO, but it is currently the best-performing one. While the use of multiple operators is common in test set problems, applying them to real-world issues is more complex. MS-SHADE relies on a multi-operator strategy, and, due to the guidance of both global and local optimal individuals, it has the same exploitation capabilities as the single-operator heavily optimal-individual-dependent PSO improvement algorithms. However, by incorporating the influence of random individuals in the third operator, its exploration capabilities far surpass those of PSOs. From the results, the improvements presented in this paper are the first to elevate the optimization effects of DEs in WFLO to the level where they outperform PSOs. Our research plays a significant role in advancing the development of sophisticated algorithms in practical applications.

In WFLO, the application of MS-SHADE can increase the actual utilization rate by 0.3%. For instance, in the case of a 100-kilowatt wind turbine, this means an average power output increase of 0.3 kilowatts per turbine, which is a considerable improvement for large-scale wind farms. Given the widespread use of wind resources, if advanced optimization algorithms like MS-SHADE can be used to design efficient wind turbine layouts, it could significantly alleviate the electrical power strain for many countries. From an economic perspective, increasing power output without additional costs could boost investor interest and accelerate the process of achieving carbon neutrality. In future studies, we plan to explore further improvements based on the most advanced DE algorithms to more effectively address large and complex WFLO problems. Additionally, we plan to incorporate simulation components to enhance the visualization of optimization results and package the optimization algorithm as a user-friendly front end [46?]. This will promote the practical application of optimization algorithms in WFLO scenarios.

Author Contributions: Conceptualization, Y.Y. and H.Y.; methodology, Y.Y. and H.Y.; software, Y.Y. and S.T.; validation, Y.Y., S.T., and H.L.; formal analysis, S.T. and H.L.; investigation, H.Y.; resources, Y.Y.; data curation, H.Y. and Z.T.; writing—original draft preparation, Y.Y.; writing—review and editing, H.Y. and S.T.; visualization, S.T.; supervision, Z.T.; project administration, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Hirosaki University Research Start Support Program, Hirosaki University, Japan, and the Tokushima University Tenure-Track Faculty Development Support System, Tokushima University, Japan.

Data Availability Statement: The data presented in this study will be openly available in <https://github.com/louiseklocky> (accessed on 1 August 2024).

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

Abbreviations

The following abbreviations are used in this manuscript:

WFLO	Wind farm layout optimization
GA	Genetic algorithm
PSO	Particle swarm optimization
MHA	Meta-heuristic algorithm
DE	Differential evolution
SHADE	Success-history based parameter adaptation for differential evolution
IEEE CEC	IEEE Congress on Evolutionary Computation
LSHADE	SHADE with iterative local search for large-scale global optimization
AGA	Wind farm layout optimization based on support vector regression-guided genetic algorithm with consideration of participation among landowners
AGPSO	An adaptive replacement strategy-incorporated particle swarm optimizer for wind farm layout optimization
CGPSO	A chaotic local search-based particle swarm optimizer for large-scale complex wind farm layout optimization

References

- Allen, M.R.; Frame, D.J.; Huntingford, C.; Jones, C.D.; Lowe, J.A.; Meinshausen, M.; Meinshausen, N. Warming caused by cumulative carbon emissions towards the trillionth tonne. *Nature* **2009**, *458*, 1163–1166. [\[CrossRef\]](#)
- O'Neill, B.C.; Dalton, M.; Fuchs, R.; Jiang, L.; Pachauri, S.; Zigova, K. Global demographic trends and future carbon emissions. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 17521–17526. [\[CrossRef\]](#)
- Liu, Z.; Guan, D.; Wei, W.; Davis, S.J.; Ciais, P.; Bai, J.; Peng, S.; Zhang, Q.; Hubacek, K.; Marland, G.; et al. Reduced carbon emission estimates from fossil fuel combustion and cement production in China. *Nature* **2015**, *524*, 335–338. [\[CrossRef\]](#)
- Hanif, I. Impact of fossil fuels energy consumption, energy policies, and urban sprawl on carbon emissions in East Asia and the Pacific: A panel investigation. *Energy Strategy Rev.* **2018**, *21*, 16–24. [\[CrossRef\]](#)
- Pao, H.T.; Li, Y.Y.; Fu, H.C. Clean energy, non-clean energy, and economic growth in the MIST countries. *Energy Policy* **2014**, *67*, 932–942. [\[CrossRef\]](#)
- Obama, B. The irreversible momentum of clean energy. *Science* **2017**, *355*, 126–129. [\[CrossRef\]](#)
- Steckel, J.C.; Jakob, M. The role of financing cost and de-risking strategies for clean energy investment. *Int. Econ.* **2018**, *155*, 19–28. [\[CrossRef\]](#)
- Samorani, M. *The Wind Farm Layout Optimization Problem*; Springer: Berlin, Germany, 2013.
- Yang, K.; Kwak, G.; Cho, K.; Huh, J. Wind farm layout optimization for wake effect uniformity. *Energy* **2019**, *183*, 983–995. [\[CrossRef\]](#)
- Shakoor, R.; Hassan, M.Y.; Raheem, A.; Wu, Y.K. Wake effect modeling: A review of wind farm layout optimization using Jensen's model. *Renew. Sustain. Energy Rev.* **2016**, *58*, 1048–1059. [\[CrossRef\]](#)
- Yang, H.; Gao, S.; Wang, R.L.; Todo, Y. A ladder spherical evolution search algorithm. *IEICE Trans. Inf. Syst.* **2021**, *104*, 461–464. [\[CrossRef\]](#)
- Tao, S.; Xu, Q.; Feijóo, A.; Zheng, G.; Zhou, J. Nonuniform wind farm layout optimization: A state-of-the-art review. *Energy* **2020**, *209*, 118339. [\[CrossRef\]](#)
- Chen, Y.; Li, H.; Jin, K.; Song, Q. Wind farm layout optimization using genetic algorithm with different hub height wind turbines. *Energy Convers. Manag.* **2013**, *70*, 56–65. [\[CrossRef\]](#)
- Ju, X.; Liu, F.; Wang, L.; Lee, W.J. Wind farm layout optimization based on support vector regression guided genetic algorithm with consideration of participation among landowners. *Energy Convers. Manag.* **2019**, *196*, 1267–1281. [\[CrossRef\]](#)
- Lei, Z.; Gao, S.; Zhang, Z.; Yang, H.; Li, H. A chaotic local search-based particle swarm optimizer for large-scale complex wind farm layout optimization. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1168–1180. [\[CrossRef\]](#)
- Bouaouda, A.; Sayouti, Y. Hybrid meta-heuristic algorithms for optimal sizing of hybrid renewable energy system: A review of the state-of-the-art. *Arch. Comput. Methods Eng.* **2022**, *29*, 4049–4083. [\[CrossRef\]](#) [\[PubMed\]](#)
- Houssein, E.H. Machine learning and meta-heuristic algorithms for renewable energy: A systematic review. In *Advanced Control and Optimization Paradigms for Wind Energy Systems*; Springer: Berlin, Germany, 2019; pp. 165–187.
- Doerr, B.; Neumann, F. A survey on recent progress in the theory of evolutionary algorithms for discrete optimization. *ACM Trans. Evol. Learn. Optim.* **2021**, *1*, 1–43. [\[CrossRef\]](#)
- Hall, R.W. Discrete models/continuous models. *Omega* **1986**, *14*, 213–220. [\[CrossRef\]](#)
- Yang, H.; Yu, Y.; Cheng, J.; Lei, Z.; Cai, Z.; Zhang, Z.; Gao, S. An intelligent metaphor-free spatial information sampling algorithm for balancing exploitation and exploration. *Knowl.-Based Syst.* **2022**, *250*, 109081. [\[CrossRef\]](#)
- Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv. (CSUR)* **2013**, *45*, 1–33. [\[CrossRef\]](#)
- Weide, B. A survey of analysis techniques for discrete algorithms. *ACM Comput. Surv. (CSUR)* **1977**, *9*, 291–313. [\[CrossRef\]](#)

23. Wang, K.; Gao, S.; Zhou, M.; Zhan, Z.H.; Cheng, J. Fractional Order Differential Evolution. *IEEE Trans. Evol. Comput.* **2024**. [[CrossRef](#)]
24. Singh, D.; Kumar, V.; Vaishali.; Kaur, M. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *Eur. J. Clin. Microbiol. Infect. Dis.* **2020**, *39*, 1379–1389. [[CrossRef](#)] [[PubMed](#)]
25. Molina, D.; LaTorre, A.; Herrera, F. SHADE with iterative local search for large-scale global optimization. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
26. Pant, M.; Zaheer, H.; Garcia-Hernandez, L.; Abraham, A. Differential Evolution: A review of more than two decades of research. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103479.
27. Yang, Y.; Tao, S.; Yang, H.; Yuan, Z.; Tang, Z. Dynamic Complex Network, Exploring Differential Evolution Algorithms from Another Perspective. *Mathematics* **2023**, *11*, 2979. [[CrossRef](#)]
28. Li, H.; Yang, Y.; Wang, Y.; Li, J.; Yang, H.; Tang, J.; Gao, S. Population interaction network in representative gravitational search algorithms: Logistic distribution leads to worse performance. *Heliyon* **2024**, *10*, 2405–8440. [[CrossRef](#)] [[PubMed](#)]
29. Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for differential evolution. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 71–78.
30. Xu, Z.; Gao, S.; Yang, H.; Lei, Z. SCJADE: Yet Another State-of-the-Art Differential Evolution Algorithm. *IEEE Trans. Electr. Electron. Eng.* **2021**, *16*, 644–646. [[CrossRef](#)]
31. Wang, X.; Li, C.; Zhu, J.; Meng, Q. L-SHADE-E: Ensemble of two differential evolution algorithms originating from L-SHADE. *Inf. Sci.* **2021**, *552*, 201–219. [[CrossRef](#)]
32. Mohamed, A.W.; Hadi, A.A.; Jambi, K.M. Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. *Swarm Evol. Comput.* **2019**, *50*, 100455. [[CrossRef](#)]
33. Li, X.; Wang, K.; Yang, H.; Tao, S.; Feng, S.; Gao, S. PAIDDE: A permutation-archive information directed differential evolution algorithm. *IEEE Access* **2022**, *10*, 50384–50402. [[CrossRef](#)]
34. Katic, I.; Højstrup, J.; Jensen, N.O. A simple model for cluster efficiency. In Proceedings of the European Wind Energy Association Conference and Exhibition, A. Raguzzi, Rome, Italy, 7–9 October 1986; Volume 1, pp. 407–410.
35. Crespo, A.; Hernandez, J.; Frandsen, S. Survey of modelling methods for wind turbine wakes and wind farms. *Wind. Energy Int. J. Prog. Appl. Wind. Power Convers. Technol.* **1999**, *2*, 1–24. [[CrossRef](#)]
36. Porté-Agel, F.; Wu, Y.T.; Chen, C.H. A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm. *Energies* **2013**, *6*, 5297–5313. [[CrossRef](#)]
37. Magnusson, M.; Smedman, A.S. Air flow behind wind turbines. *J. Wind. Eng. Ind. Aerodyn.* **1999**, *80*, 169–189. [[CrossRef](#)]
38. Barthelmie, R.J.; Hansen, K.; Frandsen, S.T.; Rathmann, O.; Schepers, J.; Schlez, W.; Phillips, J.; Rados, K.; Zervos, A.; Politis, E.; et al. Modelling and measuring flow and wind turbine wakes in large wind farms offshore. *Wind. Energy: Int. J. Prog. Appl. Wind. Power Convers. Technol.* **2009**, *12*, 431–444. [[CrossRef](#)]
39. Ammara, I.; Leclerc, C.; Masson, C. A viscous three-dimensional differential/actuator-disk method for the aerodynamic analysis of wind farms. *J. Sol. Energy Eng.* **2002**, *124*, 345–356. [[CrossRef](#)]
40. Jensen, N.O. *A Note on Wind Generator Interaction*; Risø National Laboratory: Roskilde, Denmark, 1983.
41. Tuller, S.E.; Brett, A.C. The characteristics of wind velocity that favor the fitting of a Weibull distribution in wind speed analysis. *J. Appl. Meteorol. Climatol.* **1984**, *23*, 124–134. [[CrossRef](#)]
42. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
43. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the 2014 IEEE congress on evolutionary computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.
44. Lei, Z.; Gao, S.; Wang, Y.; Yu, Y.; Guo, L. An adaptive replacement strategy-incorporated particle swarm optimizer for wind farm layout optimization. *Energy Convers. Manag.* **2022**, *269*, 116174. [[CrossRef](#)]
45. Naderi, E.; Mirzaei, L.; Trimble, J.P.; Cantrell, D.A. Multi-Objective Optimal Power Flow Incorporating Flexible Alternating Current Transmission Systems: Application of a Wavelet-Oriented Evolutionary Algorithm. *Electr. Power Components Syst.* **2024**, *52*, 766–795. [[CrossRef](#)]
46. Naderi, E.; Mirzaei, L.; Pourakbari-Kasmaei, M.; Cerna, F.V.; Lehtonen, M. Optimization of active power dispatch considering unified power flow controller: Application of evolutionary algorithms in a fuzzy framework. *Evol. Intell.* **2024**, *17*, 1357–1387. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.