Research Article

OPTIMIZATION OF WIND FARM LAYOUT FOR MAXIMUM ENERGY PRODUCTION BY STOCHASTIC FRACTAL SEARCH

Khoa Dang NGUYEN[1,2], *Tinh Trung TRAN*[2], *Dieu Ngoc VO*[1,3,*]

[1]Department of Power Systems, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
[2]College of Engineering, Can Tho University, Can Tho City, Vietnam
[3]Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

dangkhoa@ctu.edu.vn, tttinh@ctu.edu.vn, vndieu@hcmut.edu.vn

[*]Corresponding author: Dieu Ngoc VO; vndieu@hcmut.edu.vn

DOI: 10.15598/aeee.v23i1.240404

Article history: Received Apr 13, 2024; Revised May 26, 2024; Accepted Jul 26, 2024; Published Mar 31, 2025.

Abstract. *The wind power plant designs are different from the design of other conventional power plants such as hydropower plants, thermal power plants, and nuclear power plants because the input fuel of these types of power plants is controllable. Wind power plants depend on the speed of wind energy. Therefore, the problem of optimizing the location of turbines in a wind farm to achieve maximum annual energy output (AEP) is of great interest. In this paper, the Stochastic Fractal Search (SFS) algorithm is proposed to optimize the arrangement of turbines in the wind farm to minimize the wake effect so that the wind farm achieves the maximum generating capacity and the highest power factor (CF). SFS represents a significant advancement in optimization techniques, offering robust, adaptable, and efficient solutions to complex problems like wind farm layout optimization. Its innovative use of fractional dynamics and stochastic processes distinguishes it from traditional methods, providing superior performance in many scenarios. The proposed method was tested on a standard case with three types of turbines with different capacities of 850kW, 1000kW, and 1500kW to confirm the suitability of the algorithm and select the most appropriate turbine type. The results of AEP and wake loss calculated by the SFS algorithm were superior compared to those obtained by the PSO algorithm for these three turbine types. The turbine with the highest CF will be selected for application in the wind farm. Therefore, the proposed SFS algorithm can be a potential method to deal with the problem of optimization of wind farm layout.*

Keywords

Stochastic Fractal Search Algorithm, Wake effect, WAsP software, Wind farm layout optimization, windPRO software.

1. Introduction

The primary fuel sources for power plants such as coal, oil, and gas are gradually depleted. To ensure a stable power source and minimize the impact on the environment, countries are interested in developing sustainable energy. Using renewable energy sources such as wind energy with great potential in many countries. In 2021, wind power capacity rose by 93.6 GW, and the total global wind power capacity improved 837 GW, an improvement of 12% over the previous year. NZE2050's goal in the next 5 years, the world needs more than 86 GW of wind power annually and the total global wind power capacity is about 469 GW. By 2050, the world is expected to install 2TW of wind power and offshore wind will reach 19% by 2024 [1]. To design a wind farm, there are a number of issues worth consider-

ing and most of them have been extensively studied in distinctly individual ways such as turbine placement, research on wind characteristics, analyze the interaction between wind turbines (wake effect), design ancillary items (turbine transport routes, electrical cable systems, turbine foundations), reliability, economic issues, environmental impact assessment [2]. The wake effect is a complex and interesting problem in solving the problem of wind turbine layout to achieve maximum power [3]. he most popular models built by N.O. Jensen [4] and improved by Katic [5]. Jensen treated the influence area behind the turbines as a wind disturbance and ignores the eddy effect [6], which affected only the region near the turbines. Two types of wake effect models have been presented: Computational fluid dynamics (CFD) model [7, 8] and analytical model [9, 10, 11, 12]. Recently, several methods have been proposed to calculate the wake effect such as the binary matrix method based on the Jensen model [13], the wake effect model combined with the multi-turbine effect has been proposed for energy loss analysis [14].

Many works have presented different methods to solve problems related to the optimal position of wind turbines such as Changshui et al. [15] have proposed the "lazy greed" algorithm used to optimize the wind turbines. Zhang Changshui et al. [16] presented a "submodule" nature for turbine placement in wind farms based on the Jensen wake model, Serrano et al. [17] applied by iterative method to increase the distance between turbines in offshore wind farm in order to decline the wake effect. In addition, there are a number of studies on determining the optimal location to install power systems for wind farms such as installing substations and cable systems [18, 19]. The study of wind speed reduction through the turbine is also a complex process involving the determination of the turbine location, wind conditions, and wind turbine control methods [20]. The initial data for calculating the turbine power is the measured wind speed and it is statically represented by the Weibull distribution [21, 22, 23, 24].

Currently, there are commercial software for wind energy efficiency assessment and wind farm design, the most popular one being WAsP [25]. The main function of this software is the assessment of wind resources after analyzing the measured wind data set. WAsP analyzes wind resources by analyzing wind flows using a CFD model. In addition, the WAsP software provides various tools to design the wind farm, such as an assessment of wind power production considering the wake effect, analyzing wind speed, wind distortion, and wind turbulence. The windPRO software involves optimizing the wind farm's turbine layout for maximum power [26]. On the other hand, this software also has tools for environmental impact assessment and layout of turbines to respond to noise. Recently, metaheuristic optimization algorithms have been increasingly applied to engineering problems such as Evolutionary Strategy, Genetic Algorithms, Dolphin Echolocation, Cuckoo Optimization Algorithm, Artificial Bee Colony, Ray Optimization, Gray Wolf Optimizer, Colliding Bodies Optimization, and Chaotic Swarming of Particles [27]. These algorithms have proven themselves to be very competitive compared to modern hyper-simulation algorithms as well as other conventional methods.

In this paper, an optimal search algorithm is proposed, which is based on a random fractal search to solve the problem of optimizing wind farms to achieve maximum power energy. The mathematical model of the problem includes a fitness function with the goal of obtaining maximum energy and the constraints of the turbine. To check the feasibility and efficiency of the proposed algorithm, the results calculated by SFS will be compared with the results calculated by PSO and simulation results by windPRO software.

2. Problem and Formulation

2.1. Assumptions

To develop a general model for the problem of optimization of wind farm layout, a set of assumptions is considered in this paper.

1. The turbines have the same characteristics.

2. The same number of turbines for the case studies 3. The turbines are arranged onshore in two-dimensional (x,y).

4. Wind speed (v) follows Weibull distribution [20, 23, 24, 28].

The Weibull distribution is commonly used in wind energy analysis to model wind speed data because it can provide a good fit for the wind speed probability distribution. The Weibull distribution is characterized by two parameters: shape (k) and scale (c). These parameters determine the shape and scale of the distribution, respectively. The probability density function of the Weibull distribution is given by:

$$f(v,k,c) = \frac{k}{c}\left(\frac{v}{c}\right)^{k-1} e^{-\left(\frac{v}{c}\right)^{k}} \quad (1)$$

where, v is the wind speed; k is the shape parameter; c is the scale parameter.

2.2. Wake effect model

The wind speed changes after passing through the upstream wind turbines, which affects the downstream

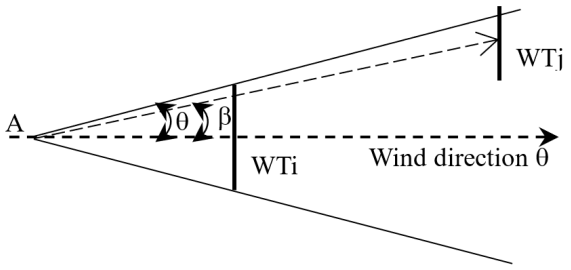


Fig. 1: A wind turbine inside the cone of another turbine [29].

wind turbines due to reduced wind speed and increased turbulence and is called the wake effect. This effect will affect the operation and energy production of wind turbines in the turbulence zone. Therefore, modeling the wake effect plays an important role in determining the location of turbines in a wind farm. The wake effect becomes more significant when the wind farm has multiple turbines. A considered turbine may be affected by wake effects from many other wind turbines [3].

The wake effect model is a crucial component in wind farm layout optimization. It simulates the interaction between wind turbines in a wind farm, accounting for the reduction in wind speed and turbulence caused by the wake of upstream turbines. A fairly simple wake effect model with extensive linear assumptions and a decaying wind speed that depends only on the distance behind the turbine was developed by N.O. Jensen [3]. Jensen treats the post-turbine influence as a wind disturbance and ignores the eddy effect, which affects only the region near the turbine.

The angle $\beta_{ij}, (0 \leq \beta \leq \pi)$, between the vector originating from the top of the hypothetical cone to the i^{th} turbine and the j^{th} turbine, is calculated as [29]:

$$\beta_{i,j} = \cos^{-1} \left\{ \frac{(x_i - x_j)\cos\theta + (y_i - y_j)\sin\theta + R/\kappa}{\sqrt{\left(x_i - x_j + \frac{R}{\kappa}\cos\theta\right)^2 + \left(y_i - y_j + \frac{R}{\kappa}\sin\theta\right)^2}} \right\} \quad (2)$$

The wind turbine j^{th} is inside the wake of turbine i^{th}, if turbin j^{th} is inside the cone. The distance between turbine i^{th} and j^{th} projected on the wind direction θ, d_{ij}, is expressed as follows [29]:

$$d_{i,j} = |(x_i - x_j)\cos\theta + (y_i - y_j)\sin\theta| \quad (3)$$

The fall in wind speed at a certain location d is [29]:

$$V_{def} = 1 - \frac{V_{down}}{V_{up}} = \frac{1 - \sqrt{1 - C_t}}{\left(1 + \frac{\kappa d_{i,j}}{R}\right)^2} \quad (4)$$

where, C_t is the thrust coefficient of turbine; d is the distance between turbine i and turbine j as a projection along with wind direction; k is the entrainment constant (decay coefficient) [30], which is empirically calculated as:

$$k = \frac{0.5}{\ln\left(\frac{H}{z_0}\right)} \quad (5)$$

where, H is the hub height, and z_0 represents the surface roughness of the terrain. k is 0.075 for land areas and 0.04 for offshore areas [31]; $d_{i,j}$ is the distance behind the turbine considering wind direction θ.

Due to the wake effect (when a turbine is affected by multiple turbines in front) the wind speed is reduced [29]:

$$V_{defi} = \sqrt{\sum_{j=1, j\neq i, \beta_{i,j}<\alpha}^{N} \left[\frac{1 - \sqrt{1 - C_t}}{(1 + \kappa d_{i,j}/R)^2}\right]} \quad (6)$$

It is easy to observe that V_{defi} is a function of wind direction (θ) and all turbine positions. It is shown that only the scaling parameter **c** of the Weibull distribution will be affected by the wake loss [6]. The wake effect is statistically described as follows [32]:

$$c'(\theta) = c(\theta).(1 - V_{defi}) \quad (7)$$

2.3. Wind turbine characteristics

The exact pattern of turbine characteristics is very important in guessing wind power energy. There have been many approaches to the introduction of wind turbines, including approximate polynomials [33]. In this article, the 9th degree polynomial model is applied to calculate the turbine characteristic model because this is the most suitable observed model.

$$f(v) = p_0 + p_1 v + p_2 v^2 + p_3 v^3 + p_4 v^4 + p_5 v^5 + \\ + p_6 v^6 + p_7 v^7 + p_8 v^8 + p_9 v^9 \quad (8)$$

The standard case is applied to the problem with turbine capacities of 850kW, 1000kW and 1500kW as shown in Table 1.

Figures 2, 3, and 4 show the comparison between the polynomial model and the actual wind turbine characteristics proposed in this paper. The characteristics of the model are observed to be very similar to those of the actual turbine.

The wind turbine characteristics are restated as follows:

$$f(\nu) = \begin{cases} 0, & v_i < v_{cut\,in},\ v_i > v_{cut\,out} \\ f(x)\ in\ Eq.(8), & v_{cut\,in} \leq v_i \leq v_{cut\,out} \\ P_{rated}, & v_{rated} \leq v_i \leq v_{cut\,out} \end{cases} \quad (9)$$

Tab. 1: Types of turbines proposed for wind farms.

Type of Turbine.	GAMESA G52/850	NORDEX N-54/1000	VESTAS V63/1500
General data			
Model:	G52/850	N54/1000	V63/1500
Rated power	850 kW	1,000	1,500
Rotor diameter	52 m	54	63.6
Swept area	2,124 m^2	2,291	3,177
Specific area	2.5 m^2/kW	2.3	2.12
Number of blades:	3	3	3
Rotor			
Minimum rotor speed	19,44	14 rd/min	-
Maximum rotor speed	30,8 rd/min	21,5 rd/min	22,9 rd/min
Cut-in wind speed	4 m/s	3,5 m/s	4 m/s
Rated wind speed	16 m/s	15,5 m/s	16 m/s
Cut-off wind speed	25 m/s	25 m/s	25 m/s
Generator			
Type	ASYNC	ASYNC	ASYNC
Number	1	1	1
Maximum speed	1900 rd/min	1513 rd/min	1650 rd/min
Voltage	690 V	690 V	690 V

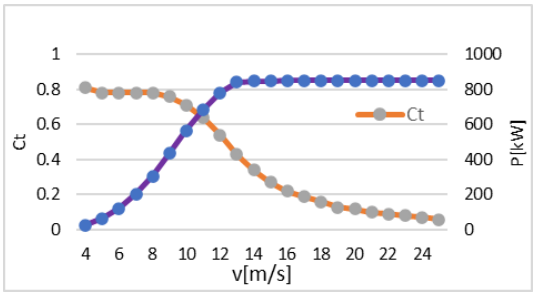


Fig. 2: GAMESA G52/850 turbine characteristics.

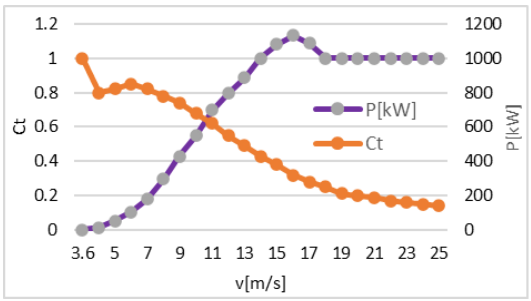


Fig. 3: NORDEX N-54/1000 turbine characteristics.

2.4. Wind power model

1) Wind Model

Wind model becomes very important in estimating wind power production. In wind pattern, wind speed and wind direction are two parameters that need to be carefully considered because they affect the power output of the wind turbine. The wind speed is usually described by the Weibull distribution and the wind direction is expressed by the probability of each sector of the wind rose [34].

This study proposes to use a 12-sector wind rose because it is widely used for wind farm design.

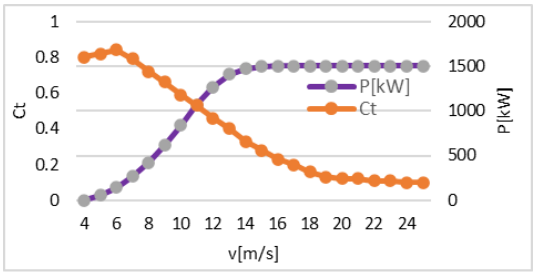


Fig. 4: VESTAS V63/1500 turbine characteristics.

2) Wind Power Output

The energy production of the turbine is shown in (10) [35]:

$$E(P,\theta) = \int_0^\infty f(v)p(v,c(\theta),k(\theta))dv \tag{10}$$

where $p(v,c(\theta),k(\theta))$ is the Weibull probability density function of wind speed.

Calculation of the energy produced by a turbine for wind direction from 0° to 360° is presented as follows [29]:

$$E(P) = \int_0^{360} p(\theta)d\theta \int_0^\infty f(v)p(v,c(\theta),k(\theta))dv \tag{11}$$

The numerical integration method will be applied to calculate the wind power output of the wind farm. The wind power output in each wind direction θ is combined as follows [29]:

$$E(P) = \sum_{i=1}^{h} f_i(\theta) \int_0^\infty f(v) \frac{k_i(\theta)}{c'_i(\theta)} \left(\frac{v}{c'_i(\theta)}\right)^{(k_i(\theta)-1)} e^{-\left(\frac{v}{c'_i(\theta)}\right)^{k_i(\theta)}} dv \tag{12}$$

3) Objective Functions

This article presents a method for optimizing the arrangement of turbines in a wind farm to maximize annual energy output:

$$Obj = \max\left[\sum E\left(P\right)\right] \tag{13}$$

3. Methodology

The SFS algorithm is a metaheuristic optimization algorithm inspired by the principles of fractal geometry and randomness. It's designed for solving complex optimization problems and is particularly useful for global optimization. SFS combines random sampling with self-similarity, creating a rich search landscape for finding the global optimum, This is based on the simulation of a dielectric breakdown process, so it becomes a suitable search engine for solving optimization problems at a general level. The procedure of the algorithm is divided into two processes as diffusive and update [36].

In the first phase, to increase search chances, each point will diffuse around the current location in response to growth characteristics. In the second phase, the simulation algorithm will work for an individual to update its location based on the location of other individuals, the second phase uses some random method such as update processes.

An outline of the Stochastic Fractal Search Algorithm works:

Step 1. Initialization: Initialize the search space and create an initial solution point within the defined bounds.

Step 2. Self-Similarity: The algorithm generates new solution points by perturbing the current solution using a random vector with a specific structure based on a fractal pattern.

Step 3. Evaluation: Evaluate the objective function for each generated solution point.

Step 4. Selection: Choose the best solution point among the current one and the newly generated ones based on the objective function values. The selected solution becomes the current one.

Step 5. Termination: The algorithm repeats steps 2-4 for a specified number of iterations or until convergence criteria are met.

Step 6. Global Optimum: The algorithm aims to converge to the global optimum by exploring the entire search space through its self-similarity and randomness.

3.1. Fractals

Some common methods are used such as an iterative functional system, L system, finite division principle, and random cracking to generate fractal shapes. These meta-heuristic algorithms are based on fractal features as a search algorithm that achieves good results both in terms of accuracy and convergence time [36].

1) Random fractals

Random processes such as Gaussian walk, fractal structure, Levy flight, osmotic cluster, Brown tree and Brownian motion trajectories are used to reduce the number of iterations of the algorithm and generate the stochastic fractal. For simplicity, consider forming a sequence with the initial term located at a random position. Then random particles are formed around the original particle and cause diffusion. The random walk algorithm is applied to simulate the diffuser. The particles produced by diffusion stick to the particles that make it up and form a group of particles. During the formation, the probability of particles being pulled to the edge is greater than that of particles entering the middle. Because of this property, it leads to a branched cluster as shown in Figure 6 [36].

2) Dielectric breakdown

Research on dielectric breakdown characteristics found that complex models can be applied to simulate the branching tendency of dielectric puncture. Examples are flashover and lightning. Niemeyer et al. [32] introduced dielectric breakdown using a random model and showed that branch discharge patterns follow fractal characteristics. This model is relatively similar to the new Diffusion Limited Aggregation (DLA) model.

3.2. Fractal search

1) Methodology of SFS

The core methodology of the SFS algorithm revolves around two main components: fractional calculus and stochastic search.

2) Fractional Calculus

- Fractional Order: Utilizes non-integer orders of differentiation and integration, providing a flexible framework to capture system dynamics with memory effects.

- Memory Effect: Helps in retaining historical search information, which guides the current search process more effectively.

- Fractal search applies the following three hypotheses:

- Each point will have an electric potential energy.

- Each point will diffuse and randomly generate a number of points.

- In each generation only keep some of the best points.

Assume P $(1 \leq P \leq 20)$ is the number of particles examined. Initially, each P_i particle is randomly placed in the search area with the same energy E_i as follows:

$$E_i = \frac{E}{P} \tag{14}$$

where, the maximum energy is E.

To optimize the fission, each particle will be diffused in each generation and generate a limit by the Levy flight. In this model, use Levy flight for the DLA development model. The Lévy flight is described by (13) as follows:

$$L(x) = \frac{1}{\pi} \int_0^{\infty} \left(e^{\left(-\alpha q^{-\beta}\right)} \cos(qx) \right) dx \tag{15}$$

where, α is the distribution coefficient; β is the distribution index, $0 < \beta \leq 2$.

Figure 7 shows the diffusion process that creates new particles around the original particle with random positions.

As a result of the diffusion process, a number of particles are produced q $(1 \leq q \leq$ the maximum diffusion number (MDN)). To generate each of these particles, both the Levy flight and Gaussian are applied as formulas (12) and (13):

$$x_i^q = x_i + \alpha_i^q \otimes Levy(\lambda) \tag{16}$$

$$x_i^q = x_i + \beta \times G \tag{17}$$

where $G = Gaussian(P_i, |BP| - (\gamma \times BP - \gamma' \times P_i)$; $\beta = \frac{\log(g_e)}{g_e}$; g_e is the number of generations; *BP(Best Point)* is the best score; γ and $\gamma' \in [0, 1]$.

To get a good score for both the Lévy flight method and the Gaussian distribution, the fractal search method randomly uses both methods. Because the Lévy distribution gives a fast convergence algorithm, and the Gaussian distribution gives better results.

Since the approaches all depend on stochastic processes, fast absorption cannot be guaranteed. Therefore, α is an important parameter for fast convergence. Two formulas are considered for α, one for a broader search, and the other for a higher precision search:

$$\alpha_i = \frac{U - L}{(g_e \times \log(E_i))^{\varepsilon}} \tag{18}$$

where, E_i is the energy of the P_i; U is upper bound and L is lower bound of the search area; ε is usually taken as $3/2$.

After the diffusion has determined the position of the particles, the energy distribution between the particles is created. Particles with better target values will have a higher energy distribution. Each diffuse particle has a target value F_i where $i = 1, 2, \ldots, q$. The energy is distributed to the points as follows:

$$E_i^j = \left[\left(\frac{F_i}{F_i + \sum_{k=1}^{q} F_k} \right) \right] \times E_i \tag{19}$$

where, F_i is the point before diffusion.

Because of the complex diffusion, only some of the best particles will be selected for the next. The energy of the discarded particles will be distributed to the selected particles and new particles will be created.

The total energy of the removed points is Φ; μ is the ratio of the energy distribution between the selected points and the newly created points. The energy distributed to the remaining points is as follows:

$$E^{t+1} = E^t \left(\left(F_t / \sum_{k=1}^{\xi} F_k \right) \times \Phi \right) \times \mu \tag{20}$$

where E^{t+1} and E^t is the energy of the t^{th} point after and before the energy distribution, ξ is the total of number points in the iteration. For each diffuse point, the number of newly formed and randomly positioned points in the search space is calculated as follows:

$$\upsilon = \frac{\log(N_e)}{\log(\text{MDN})} \tag{21}$$

where, N_e is the number of eliminated points.

The energy distribution for each produced point is equal:

$$E'_c = \frac{\Phi(1 - \mu)}{\upsilon}, c = 1, 2, ..., \upsilon \tag{22}$$

3.3. Stochastic Fractal Search algorithm

SFS is a modern optimization algorithm designed to address complex, multi-dimensional optimization problems that traditional methods often struggle to solve effectively. These problems are prevalent in various fields, including engineering, finance, and artificial intelligence, where finding the global optimum in a highly nonlinear and multimodal landscape is crucial. SFS is inspired by the natural process of fractal growth, characterized by self-similarity and recursive pattern generation. The methodology involves two main phases: diffusion and update [36].

Generate candidate solutions by creating random walks influenced by the fractal dimension. Because Fractal Search does not exchange information between particles. Therefore, SFS adds an update process to balance the accuracy and convergence time of the algorithm.

Lévy flight and Gaussian walking method have been applied to generate new particles by diffusion process [27]. The Gaussian steps involved in the diffusion phase are calculated as follows:

$$Gaussian\_walk_1 = G(\mu_{BP}, \sigma) + (\varepsilon \times BP - \varepsilon' \times P_i) \tag{23}$$

$$Gaussian\_walk_2 = G(\mu_P, \sigma) \tag{24}$$

where, ε and $\varepsilon^{'}$ are randomly distributed; μ_{BP} and σ are Gaussian parameters; μ_P and σ are the second two Gaussian parameters.

The standard deviation is as follows:

$$\sigma = |\beta \times (P_i - BP)| \tag{25}$$

The single search method is applied to reduce the size of the Gaussian walk $log(g_e)/g_e$, so the convergence time is faster.

At initialization, points are randomly initialized based on upper and lower bound constraints. Initialize the j^{th} point, P_j as follows:

$$P_j = L + \varepsilon(U - L) \tag{26}$$

where, ε is a random distribution and is limited to the interval [0,1]. The points will probe around the current position to consider the search space in the diffusion process. Besides, two statistical processes are performed for a better spatial search. The first statistic is performed on each vector, and then the next statistic is applied on all points.

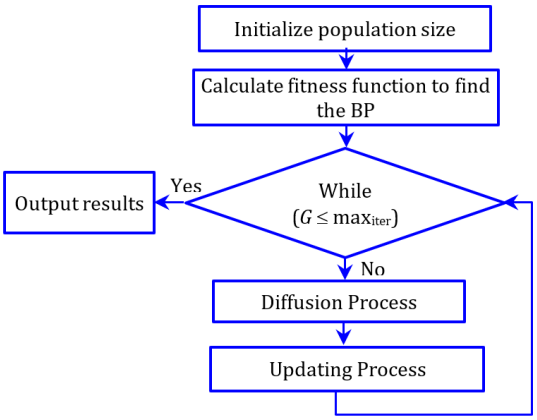
The first statistical process, rank all points by (27), N is the total number of points in group. Then each i^{th} point is assigned a probability value as follows:

$$Pa_i = \frac{rank(P_i)}{N} \tag{27}$$

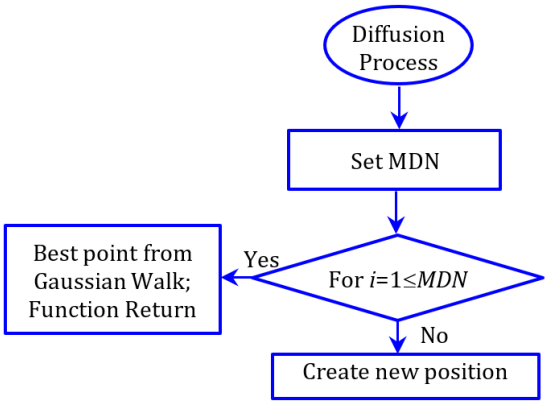
Equation (27) means that the better the score, the greater the probability of that score. It helps the bad points increase the chance to change position. The chances of finding a better solution will improve in the next generation.

$$P_i'(j) = P_r(j) - \varepsilon \times (P_t(j) - P_i(j)) \tag{28}$$

where, $P_i^{'}$ is the new point; P_r and P_t are randomly chosen points from the group.



(a) The flowchart of the SFS algorithm.



(b) The diffusion process of the SFS algorithm.

Fig. 5: The SFS algorithm flowchart and Diffusion process algorithm flowchart [36].

This property is intended to better explore and satisfy the diverse nature of the algorithm based on two statistical processes [36]. All the points obtained from the previous process will be re-ranked according to (27) before the second process is performed. Similar to the first process, if the P_i' satisfies the condition $Pa_i < \varepsilon$, the position is changed according to (29).

$$\begin{aligned} P_i'' &= P_i' - \hat{\varepsilon}(P_t' - BP) \mid \varepsilon' \leq 0.5 \\ P_i'' &= P_i' - \hat{\varepsilon}(P_t' - P_r') \mid \varepsilon' > 0.5 \end{aligned} \tag{29}$$

where, P_r' and P_t' are randomly chosen points from the first process. The $P"_i$ is new point if its objective function value is better than P_i'.

Step 1: Initialize population size: N, MDN, $\max_{\text{iter}}$, and G.

Step 2: Find the best score (BP) by calculating the fitness function.

Step 3: Check condition:

- If G $\leq \max_{\text{iter}}$: Output results.

- If G $> \max_{\text{iter}}$: Call Diffusion Process.

Step 4: Diffusion Process

- Set MDN.

- Generate new particles, if MDN is reached.

- The new point will replace the current point.

- Identify the BP for this phase.

Step 5: Updating Process

The First Updating Process:

Rank point using Eq. (24).

- Generate a new point P_i' (if $Pa_i < \varepsilon$), using Eq. (28).

- The new point (P_i') will replace the current point (P_i). - Identify the BP for this process.

The Second Updating Process:

Rank particles using Eq. (27).

- Generate a new point P_i'' (if $Pa_i < \varepsilon$), using Eq. (29). - The new point (P_i'') will replace the current point (P_i'). - Identify the BP.

Step 6: Check process stop condition:

The BP is the optimal solution if the maxiter is reached.

Otherwise, go to *Step 3:*.

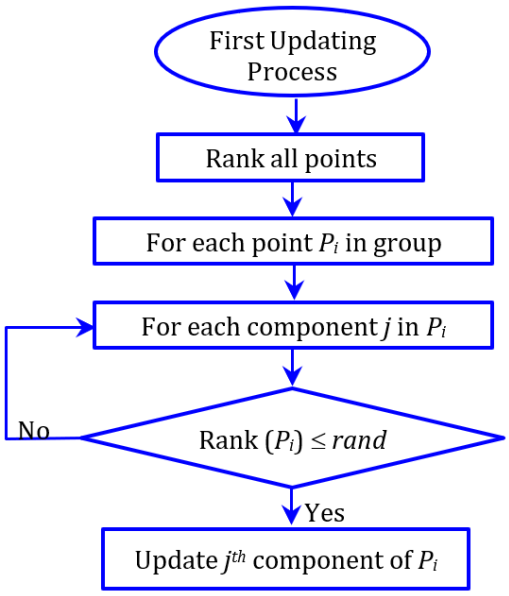
Update Phase: Improve candidate solutions by employing random walks that allow for both local and global search capabilities.

Selection: Evaluate the fitness of each candidate solution and select the best ones to form the new population.

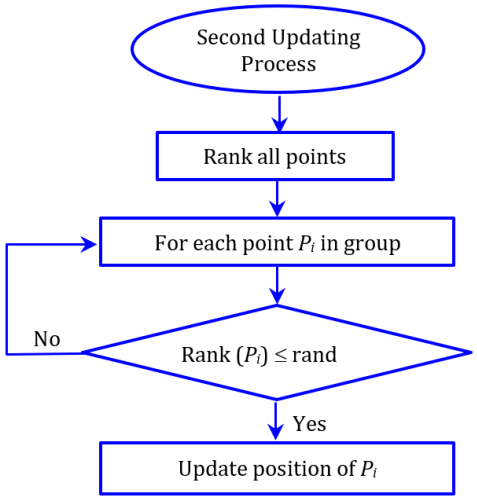
Convergence Check: Repeat the diffusion and update phases until convergence criteria, such as a predefined number of iterations or an acceptable fitness level, are met.

Exploration and Exploitation Balance: SFS effectively balances exploration (global search) and exploitation (local search) through its dual-phase methodology, enhancing its ability to find the global optimum. SFS can handle large-scale optimization problems due to its inherent scalability and adaptability to different problem sizes.

The iterative nature and extensive exploration mechanisms, like Levy flights, can be computationally expensive, especially for very large problems. The performance of SFS can be sensitive to its parameter settings, such as the step size in Levy flights and the fractal dimension, requiring careful tuning. Implementing SFS can be more complex compared to traditional optimization algorithms, necessitating a deeper understanding of fractal mathematics and stochastic processes.



(a) The first update process.



(b) The second update process.

Fig. 6: The update processes of the SFS algorithm [36].

4. Numerical Results

Wind farm data is very important to the problem of determining the optimal location of turbines in a wind farm to reduce wake-up effects and achieve maximum generating capacity. The wind farm proposed in this paper is an onshore wind farm, referenced from the WAsP workspace sample, filename Version8Windfarm.wh [25]. The wind farm is a complex terrain with elevations ranging from 146.7m to 350m. The average wind speed of the project is 7.25m/s and the average wind energy density is 388 W/m^2. The goal of the problem is the optimization of wind farm layout for each proposed turbine type so that the energy power output and the power factor of the wind

farm is the best, thereby proposing the selection of the appropriate turbine.

Tab. 2: Input parameters.

Input parameters	
Roughness (Z_0)	0.083
Wind velocity in free flow (V_0)	7.25m/s
Hub height (h)	50m
Wind Farm dimension	3200m x 4200m
Thrust coefficient (C_t)	0.75
Wind density	388W/m2

This paper proposes to select 3 types of wind turbines arranged on the same wind farm to determine the optimal layout and select the appropriate type of turbine. Case studies are summarized in the following Table 3.

Among these parameters, SFS is sensitive to MDN. The research results show that the diffusion number can affect the performance of the problem, and depends on the optimization problem. Some research results show that some functions of the problem are significantly improved when increasing MDN. However, increasing MDN will affect the convergence time and convergence speed of the problem, so it needs to be consider [36].

S.Walk is a diffuse walk and an optional parameter. (S.Walk = 0 for the first Gaussian walk and simple problems. S.Walk = 1 for the second Gaussian walk and hard problems).

The control parameters of the mentioned algorithm used in case studies are given according to Table 3.

The energy produced by the wind farm is greatly affected by wind speed and wind direction. Specify the wind speed level to shut down the turbine as below 3m/s or above 25m/s [37]. The direction of the turbine must be perpendicular to the wind direction to receive maximum wind energy [38]. The wind atlas in the project area is calculated based on long-term corrected wind measurements taking into account the effects of obstacles, roughness, and terrain elevation. Below is the wind data of the project area referenced from the WAsP software [25] as shown in Figure 7.

Figure 7 shows the prevailing wind direction of the project area with an angle from 2700 to 3000. A simplified wind rose is used, which is divided into 2 sectors (sectors 10 and 11). The highest probability occurs at a wind speed of 7.25m/s, accounting for about 12%.

The case studies using the same data source from the WAsP software library. The terrain is complex and the altitude varies from 146.7m to 350 m.

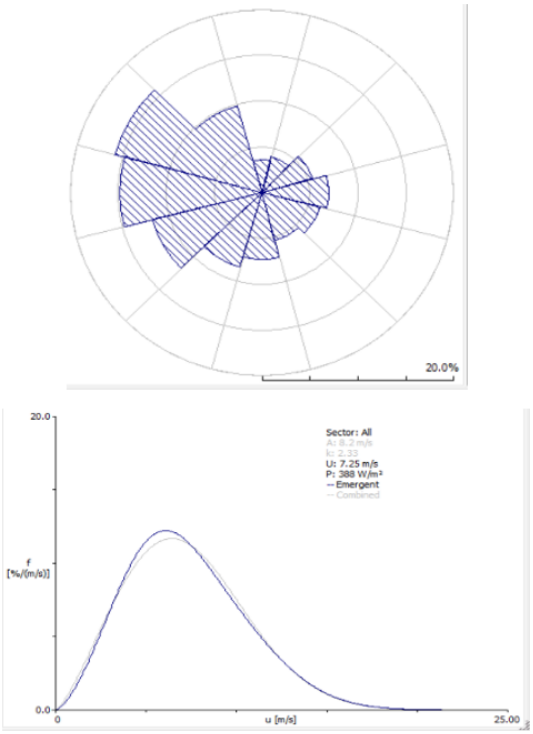


Fig. 7: The wind rose and wind speed distribution.

4.1. Case study 1

Case study 1 considers the optimal arrangement for 11 wind turbines, turbine capacity is 0.85MW. The results of the optimal arrangement of turbines in the wind farm by the SFS algorithm will be compared with the results calculated by the PSO algorithm and the simulation results by the windPRO software.

A comparison of the convergence curves of the best fitness values obtained from the SFS and PSO algorithm is shown in Figure 8. This graph gives information the convergence curve of SFS and PSO algorithms. It is clear that the SFS algorithm has reached stability

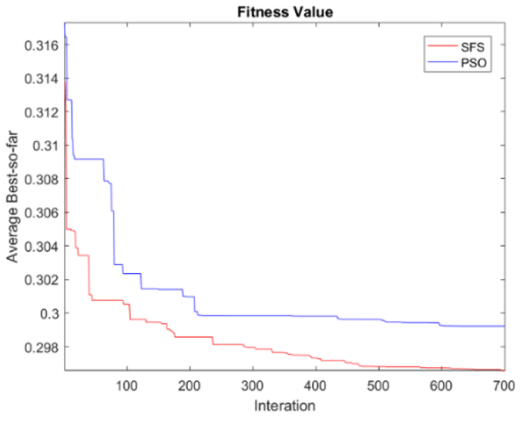


Fig. 8: The convergence curves of SFS and PSO.

Tab. 3: Case studies of optimization of wind farm layout

Case Studies	Case Study 1	Case Study 2	Case Study 3
Wind farm capacity	9.35 MW	11 MW	16.5 MW
Wind turbine	11x0.85 MW	11x1 MW	11x1.5 MW
Shape of wind farm	Fixed	Fixed	Fixed
Vcut in	4 m/s	4 m/s	4 m/s
Vcut out	25 m/s	25 m/s	25 m/s
Data Source	WASP Software	WASP Software	WASP Software
Start Point	11	11	11
Maximum_Generation	100	100	100
Maximum_Diffusion	2	2	2
S.Walk	0.5	0.5	0.5

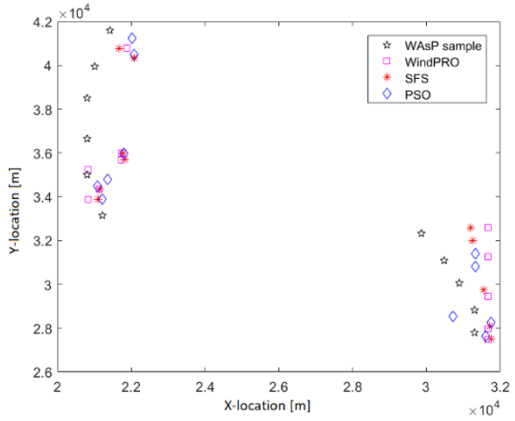


Fig. 9: Result of all algorithms.

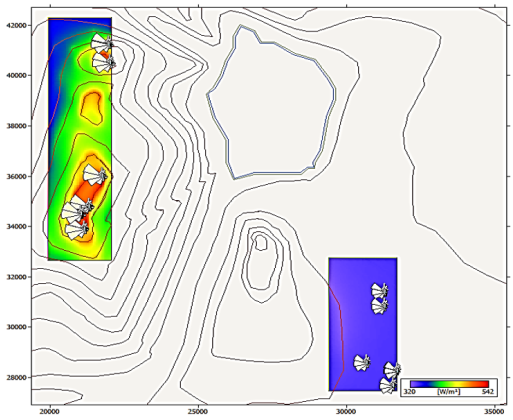


Fig. 10: Result of the arrangement of turbines by PSO.

around the 500th generation. Furthermore, the assembly speed of the SFS algorithm is faster than that of the PSO algorithm.

Figure 9 shows a comparison of the wind farm layouts of the methods. The positions of the turbines are different for the methods, annotated with different shapes and colors as shown in this figure. The turbines are installed in the wind regions with the highest wind energy density and the arrangement of the turbines ensures avoidance wake effect.

The wind farm layout calculated by SFS, PSO, and windPRO algorithms were recalculated by WAsP, and the results are presented in Table 4. The comparison of the AEP nets of the algorithms in Table 4 shows that optimizing the wind farm layout by SFS is the best. The net AEP calculated by SFS is 29.577 GWh, which is 0.282 GWh higher than windPRO, and 0.302 GWh higher than PSO. Besides, the wake loss calculated by the SFS algorithm is 0.72%. This value is lower than the wake loss calculated by windPRO and PSO, they are 1.82% and 0.76%. On the other hand, the wind farm efficiency achieved by SFS is 36.11% higher than other methods. It is 0.34% higher than windPRO software and 0.37% PSO. The capacity factor by windPRO and PSO specifically is 35.77% and 35.74%. The comparison results show that the SFS algorithm gives approximate calculation results with commercial software.

Tab. 4: Result of wind farm layout optimization based on WASP for Case study 1.

Methods	WASP	windPRO	PSO	SFS
Gross AEP [GWh]	27.754	29.839	29.500	29.790
Net AEP [GWh]	27.538	29.295	29.275	29.577
Wake loss [%]	0.778	1.820	0.760	0.720
Capacity factor [%]	33.62	35.77	35.74	36.11

Figures 10 and 11 show the arrangement of turbines in wind farm by PSO algorithm and SFS algorithm. The proposed algorithm gives results on wind farm layout similar to the results of commercial software. The turbines are located where the wind density is highest and the wake effect is minimized to ensure maximum energy production.

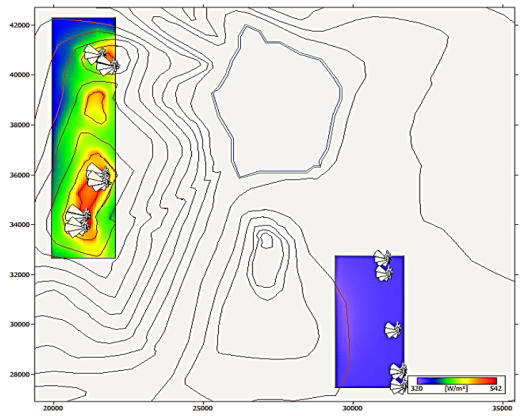


Fig. 11: Result of the arrangement of turbines by SFS.

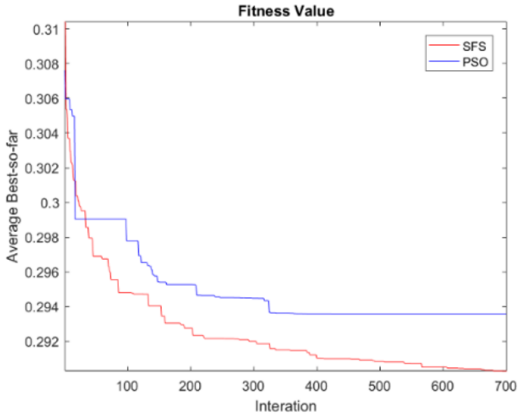


Fig. 12: Convergence curve.

4.2. Case study 2

Similar to Case Study 1, Case Study 2 presents an optimal wind farm layout for 11 turbines, turbine capacity is 1 MW. The results of the optimal arrangement of turbines in the wind farm by the SFS algorithm will be compared with the results calculated by the PSO algorithm and the simulation results by the WindPRO software.

Figure 12 provides information about the comparison of the convergence curves of the best fitness values obtained from the SFS and PSO algorithm. Similar to case study 2, the SFS algorithm reaches stability around the 600th generation and the convergence speed is faster than that of the PSO algorithm.

Figure 13 shows the arrangement of turbines of the methods on the same project area. The location of the turbines is different for the methods, which are annotated with different shapes and colors.

Table 5 gives information about the optimization results of wind farm layout by SFS, PSO, windPRO algorithms and have been recalculated by WAsP. It is clear that optimizing wind farm layout by SFS is the best.

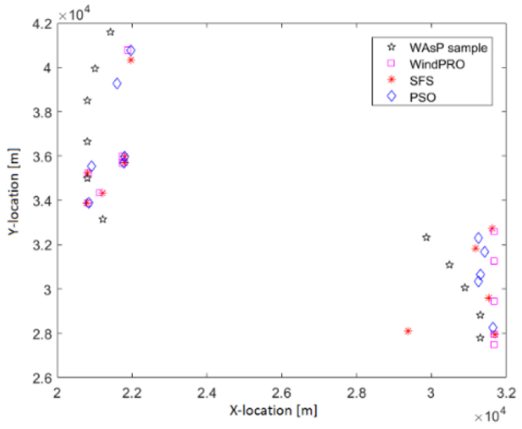


Fig. 13: Result of all algorithms.

It is superior in all parameters such as Gross AEP, net AEP, performance, and wake loss.

Tab. 5: Result of wind farm layout optimization based on WASP for Case study 2

Methods	WASP	windPRO	PSO	SFS
Gross AEP [GWh]	27.898	30.308	30.102	30.322
Net AEP [GWh]	27.802	29.910	29.937	30.166
Wake loss [%]	0.345	1.316	0.550	0.510
Capacity factor [%]	28.85	31.04	31.07	31.31

The Gross AEP of SFS is higher than that of PSO and windPRO algorithms at 0.22G Wh and 0.014 GWh. The net AEP given by SFS is 30.0166G Wh, which is 0.256 GWh higher than windPRO and 0.229 GW higher than PSO. In addition, the final loss calculated by SFS is 0.51%, which is lower than the other methods as shown in Table 5. Furthermore, the capacity factor applied by SFS is 31.31%, which is higher than the other two methods. Wind farm efficiency given by SFS is 0.27% higher than windPRO software and 0.24% higher than PSO. The analysis results show that the SFS algorithm gives the same result as commercial software.

Figures 14, and 15 provide information about the wind farm layout results using PSO and SFS. The turbines are installed where the wind density is highest, and the turbine placement ensures minimal wake effects. The result of turbine layout according to the SFS algorithm is similar to commercial softwares as shown in Figure 13.

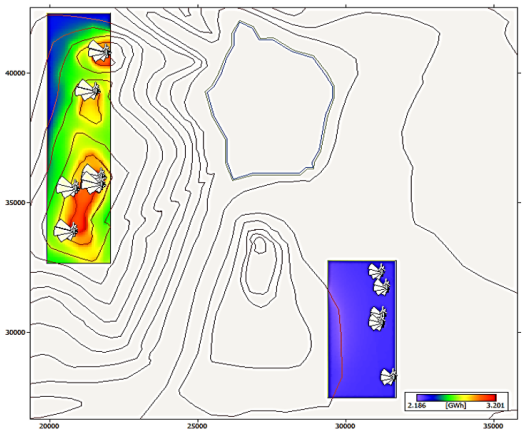


Fig. 14: Result of the arrangement of turbines by PSO.

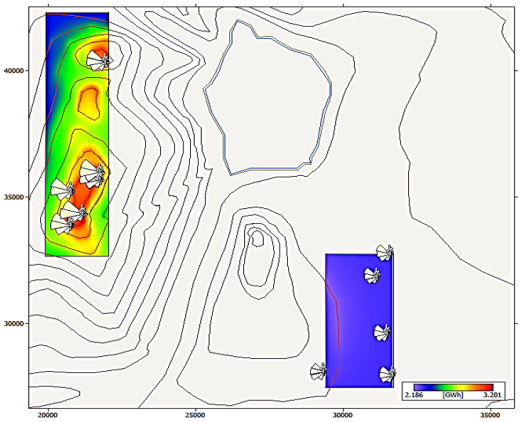


Fig. 15: Result of the arrangement of turbines by SFS.

Tab. 6: Result of wind farm layout optimization based on WASP for Case study 3

Methods	WASP	windPRO	PSO	SFS
Gross AEP [GWh]	41.110	44.66	43.76	**44.657**
Net AEP [GWh]	40.901	44.28	43.52	**44.416**
Wake loss [%]	0.508	0.850	0.550	**0.540**
Capacity factor [%]	28.30	30.64	30.11	**30.73**

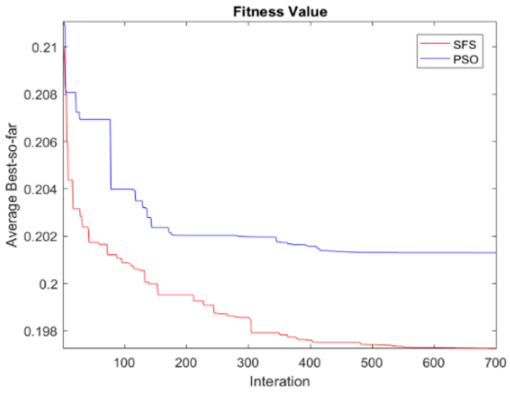


Fig. 16: Convergence curve.

4.3. Case study 3

In this case, it is proposed to use 11 wind turbines, each with a capacity of 1.5 MW as shown in Table 3. The calculation methods are similar to those in Case Studies 1 and 2. The results are presented in Table 6.

Figure 16 shows the convergence characteristics of the best fit values of the SFS and PSO algorithms. This figure shows that the convergence curve of the SFS algorithm has reached its stability around the 500th generation and is better than that of PSO software. Moreover, the convergence speed of the SFS algorithm is much faster than that of the PSO algorithm.

Figure 17 gives information about the comparision the wind farm layout of the proposed methods. The results of the turbine layout of the methods are different, which are annotated with different shapes and colors on the diagram. The wind turbines are located in the wind regions with the highest wind energy density and the arrangement of the turbines ensures avoidance wake effect.

Table 6 is a comparison of parameters such as Gross AEP, net AEP, wake loss, and performance of the methods. Similar to Case study 2, this case also shows that optimizing wind farm layout by SFS is the best compared to windPRO and PSO. Table 6 shows that the parameters performed by SFS such as Gross AEP, net AEP, performance, and wake loss are all better than the other methods. The total AEP, net AEP, and capacity factor obtained by SFS are 44,657GWh, and 30.73%, all of which are higher than windPRO and PSO. On the other hand, wake loss is lower by 0.54%. The comparison results show that the proposed algorithm gives the same results as commercial software.

Figures 18, and 19 are the results of the wind farm layout by PSO, and SFS algorithms. The wind turbines are located where the wind density is highest, and the turbine placement ensures minimal wake effects. The results of turbine layout according to the SFS algorithm are similar to commercial software and are shown in Figure 17.

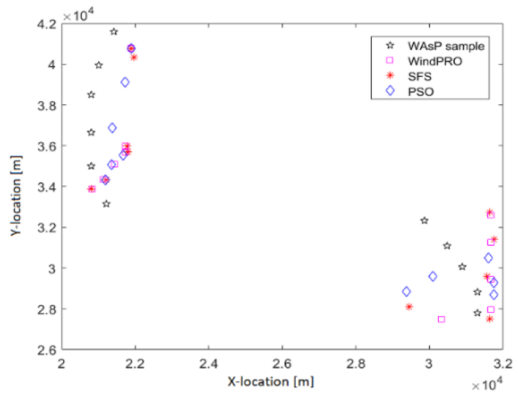


Fig. 17: Result of all algorithms.

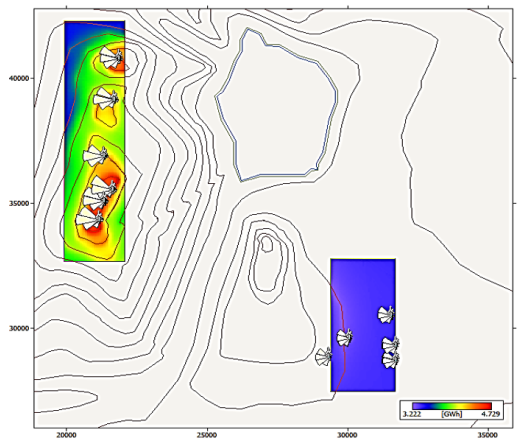


Fig. 18: Result of the arrangement of turbines by PSO.

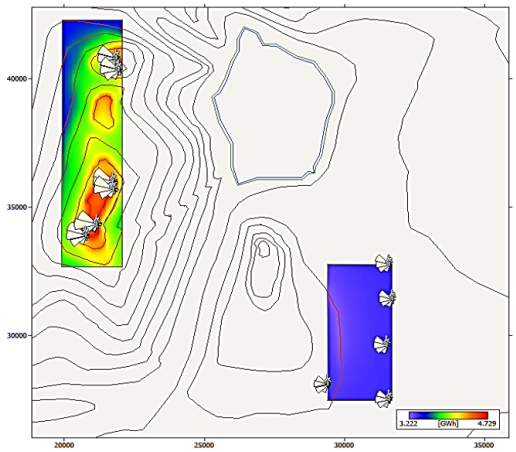


Fig. 19: Result of the arrangement of turbines by SFS.

4.4. Discussion

Table 6 shows the average and standard deviation of the methods for the proposed case studies. This result shows the superiority of convergence time and stability of the proposed algorithm to find the optimal results. This shows that the SFS algorithm is an appropriate algorithm to solve wind farm layout optimization problems.

According to case studies analyzed above, it is shown that optimally using the SFS algorithm gives the best results and is best suited to the wind farm model. Therefore, the SFS algorithm is suitable to apply to the optimization problem of wind farm layout and the criterion for selecting the appropriate type of turbine for the wind farm is a high capacity factor.

To determine the type of turbine for the wind farm, the capacity factor (CF) of the wind farm is considered. From the analyzed cases above and Table 8, it is shown that the capacity factor of the wind farm using 850 kW turbine is the highest (36.239%) compared to 1000 kW turbine (CF = 31.19%) and 1500 kW turbine (CF = 30.72%). Therefore, using 850 kW turbines is the most optimal for this wind farm layout.

The simple explanation for this result is that the rated capacity of the turbines is achieved at wind speeds of 15 m/s or more, while the highest probability distribution of wind speed is from 3 to 10 m/s. In the range of 3-10 m/s, the capacity of all 3 types of turbines is quite different, the capacity is much different only at wind speeds of about 12 m/s or more. Therefore, the CF of the 850 kW turbine will be higher than that of the other cases.

The Stochastic Fractal Search algorithm has many advantages for wind farm layout optimization problems such as: SFS has the ability to find the global optimum for complex, multi-modal optimization problems. SFS's search space is wide using self-similarity and randomness, making it less likely to get stuck in local optima. SFS is a flexible algorithm that can be applied to many different types of optimization problems, including continuous optimization, discrete optimization, and combinatorial optimization. The stochastic and self-similar aspects of SFS make it a powerful algorithm that can handle noisy or uncertain objective functions. However, SFS also has some limitations that need to be improved such as: SFS can be limited for extremely multi-dimensional or complex problems, it may require a significant number of iterations to reach an acceptable solution. Like many meta-heuristic algorithms, SFS may require tuning of specific parameters such as fractal patterns, and finding the right parameters for a particular problem can be time-consuming and SFS performance can be sensitive to the initial solution. To address this drawback, a combination of Lévy flight and Gaussian walk is proposed. Unlike traditional optimization methods, SFS's use of fractional order dynamics provides a superior balance between exploration (global search) and exploitation (local search), leading to more efficient and effective op-

Tab. 7: Result of the average and standard deviation from case studies

Methods \ Case studies		Case Study 1	Case Study 2	Case Study 3
windPRO	Avg.	29.295	29.858	44.282
	Std.	0	0	0
	Min value	29.295	29.858	44.282
	Max value	29.295	29.858	44.282
SFS	Avg.	29.448	30.056	44.316
	Std.	0.058	0.086	0.104
	Min value	29.352	29.858	44.039
	Max value	29.577	30.166	44.416
PSO	Avg.	29.057	29.518	43.275
	Std.	0.143	0.250	0.239
	Min value	28.768	29.114	42.908
	Max value	29.275	29.937	43.520

timization. The results of 3 case studies show that the convergence of SFS is better than the PSO algorithm.

The PSO is relatively easy to understand and implement. It requires fewer parameters to adjust compared to other meta-heuristic algorithms. However, PSO can suffer from premature convergence, where particles converge to a local optimum rather than the global optimum, especially in complex, high-dimensional search spaces. The performance of PSO heavily depends on the proper tuning of its parameters, such as inertia weight, cognitive coefficient, and social coefficient. PSO might not scale well with very large problems, as the computational cost increases with the number of particles and iterations required for convergence [40].

The windPRO is widely recognized and accepted in the industry for wind farm design and optimization. It offers a wide range of modules for different aspects of wind farm planning, including resource assessment, layout optimization, and environmental impact analysis [26]. On the other hand, windPRO is a commercial software that can be expensive, limiting its accessibility to smaller developers or research institutions with limited budgets. The accuracy of windPRO's outputs is highly dependent on the quality and resolution of the input data. Inaccurate wind resource data or terrain models can lead to suboptimal designs.

5. Conclusion and Future Work

In this paper, the SFS algorithm is introduced to address the optimization problem of turbine placement in a wind farm. In the considered problem, the annual energy output calculation model is used to select the type of turbine suitable for terrain conditions, wind data, and wake effects. A mathematical model is introduced to calculate the wake effect for the problem of wind turbine energy production, and proven for accurate and efficiency. The results are compared with the widely used algorithm (PSO) and the commercial software (windPRO), which shows that the SFS algorithm is suitable for the optimization of the wind farm layout. The optimal layout of the wind farm using SFS is close to that of windPRO software. Three case studies with different types of turbines are proposed to further verify the performance and reliability of the algorithms.

The Stochastic Fractal Search algorithm offers a promising optimization technique with several advantages, making it suitable for various optimization tasks across different domains. It explores the search space efficiently by randomly exploring regions and exploiting promising areas through local searches, leading to a more thorough exploration of the solution space. Due to its stochastic nature, SFS is robust against getting stuck in local optima. It can escape from local optima by employing random perturbations and diversification strategies, enabling it to continue searching for better solutions. On the other hand, SFS exhibits favorable convergence properties, often converging to near-optimal solutions within a reasonable number of iterations. In future research, we aim to investigate the correlation between the arrangement of wind turbines using a multi-objective function and the associated costs of this optimization. We will also focus on introducing further improvements to the SFS algorithm, utilizing approaches suggested in references [41, 42, 43, 44].

Acknowledgment

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Author Contributions

Khoa Dang Nguyen: Conceptualization, Methodology, Software, Writing- Original draft preparation; Tinh Trung Tran: Data curation, Visualization, Investigation; Dieu Ngoc Vo: Resources, Validation, Writing - Review & Editing, Supervision.

References

[1] Global Offshore Wind Report 2021 – GWEC. Available at: https://gwec.net/global-offshore-wind-report-2021/.

[2] GONZÁLEZ, J. S., M. B. PAYÁN, J. M. R. SANTOS, F. GONZÁLEZ-LONGATT. A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable and Sustainable Energy Reviews*. 2014, vol. 30, pp. 133-–144. DOI: 10.1016/j.rser. 2013.09.027.

[3] SHAKOOR, R., M. Y. HASSAN, A. RAHEEM, Y.-K. WU. Wake effect modeling: A review of wind farm layout optimization using Jensen's model. *Renewable and Sustainable Energy Reviews*. 2016, vol. 58, pp. 1048-–1059. DOI: 10.1016/j.rser.2015.12.229.

[4] JENSEN, N. O.. A note on wind generator interaction. *Risø National Laboratory*. 1983.

[5] KATIC´ I, J . HØJSTRUP, N. O. JENSEN. A Simple Model for Cluster Efficiency. *In: Proceedings of the European wind energy association conference and exhibition. Rome, Italy*. 1986.

[6] BASTANKHAH, M., F. PORTÉ-AGEL. A new analytical model for wind-turbine wakes. *Renewable Energy*. 2014, vol. 70, pp. 116–123. DOI: 10.1016/j.renene.2014.01.002.

[7] ANDERSEN S. J., J. N. SØRENSEN, S. Ivanell, R. F. Mikkels. Comparison of engineering wake models with CFD simulations. *Journal of Physics: Conference Series*. 2014, vol. 524. DOI: 10.1088/1742-6596/524/1/012161.

[8] CARRARO M, et al. CFD Modeling of Wind Turbine Blades with Eroded Leading Edge. *Fluids*. 2022, vol.9, no. 9. DOI: 10.3390/fluids7090302.

[9] HU W. Q. YANG, J. ZHANG, J. HU. Coupled On-Site Measurement/CFD Based Approach for Wind Resource Assessment and Wind Farm Micro-Siting over Complex Terrain. *In: IOP Conference Series: Earth and Environmental Science*. 2020, vol. 455. DOI: 10.1088/1755-1315/455/1/012037.

[10] THØRGERSEN M., T. SØRENSEN, P. NIELSEN, A. GRÖTZNER, S. CHUN. WindPRO/PARK: Introduction to Wind Turbine Wake Modelling and Wake Generated Turbulence. *EMD International A/S: Aalborg, Denmark*. 2005. DOI: 10.1088/1755-1315/455/1/012037.

[11] AINSLIE J. F.. Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*. 1988, vol. 27, iss. 1, pp. 213—224. DOI: 10.1016/0167-6105(88)90037-2.

[12] AINSLIE J. F.. Wind fields in wakes. *In: European Union wind energy conference. H.S. Stephens and Associates*. 1996, pp. 764-768.

[13] HOU, P., W. HU, M. SOLTANI, Z. CHEN. A novel energy yields calculation method for irregular wind farm layout. *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan*. 2015, pp. 380–385. DOI: 10.1109/IECON.2015.7392129.

[14] HOU, P., W. HU, M. SOLTANI, C. CHEN, B. ZHANG, Z. CHEN. Offshore Wind Farm Layout Design Considering Optimized Power Dispatch Strategy. *IEEE Transactions on Sustainable Energy*. 2017, vol. 8, no. 2, pp. 638-647. DOI: 10.1109/ TSTE.2016.26142669.

[15] CHANGSHUI Z., H. GUANGDONG, W. JUN. A fast algorithm based on the submodular property for optimization of wind turbine positioning. *Renewable Energy*. 2011, vol. 36, iss. 11, pp. 2951–2958. DOI: 10.1016/j.renene.2011.03.045.

[16] WAGNER M., J. DAY, F. NEUMANN. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*. 2013, vol. 51, pp. 64—70. DOI: 10.1016/j.renene.2012.09.008.

[17] GONZÁLEZ, J. S., M. B. PAYÁN, J. R. SANTOS. A New and Efficient Method for Optimal Design of Large Offshore Wind Power Plants. *IEEE Transactions on Power Systems*. 2013, vol. 28, no. 3, pp. 3075–3084. DOI: 10.1109/TPWRS.2013. 2251014.

[18] LUMBRERAS S., A. RAMOS. Optimal Design of the Electrical Layout of an Offshore Wind Farm Applying Decomposition Strategies. *IEEE Transactions on Power Systems*. 2013, vol. 28, no. 2, pp. 1434–1441. DOI: 10.1109/TPWRS.2012.22049064.

[19] HOU P., W. HU, Z. CHEN. Optimisation for offshore wind farm cable connection layout using adaptive particle swarm optimisation minimum

spanning tree method. *IET Renewable Power Generation*. 2016, vol. 10. iss. 5, pp. 694—702. DOI: 10.1049/iet-rpg.2015.0340.

[20] SINGH, K. A., M. G. M. KHAN, M. R. AHMED. Wind Energy Resource Assessment for Cook Islands With Accurate Estimation of Weibull Parameters Using Frequentist and Bayesian Methods. *IEEE Access*. 2022, vol. 10, pp. 25935-–25953. DOI: 10.1109/ACCESS.2022. 3156933.

[21] DHAKAL R., A. SEDAI, S. POL, S. PARAMESWARAN, A. NEJAT, H. MOUSSA. A Novel Hybrid Method for Short-Term Wind Speed Prediction Based on Wind Probability Distribution Function and Machine Learning Models. *Applied Sciences* . 2022, vol. 12, no. 18. DOI: 10.3390/app12189038.

[22] ALSAMAMRA H. R., S. SALAH, J. A. H. SHOQEIR, A. J. MANASRA. A comparative study of five numerical methods for the estimation of Weibull parameters for wind energy evaluation at Eastern Jerusalem, Palestine. *Energy Reports*. 2022, vol. 8, pp. 4801-–4810. DOI: 10.1016/j.egyr.2022.03.180.

[23] OKAKWU I., D. AKINYELE, O. OLABODE, T. AJEWOLE, E. OLUWASOGO, A. OYEDEJI. Comparative Assessment of Numerical Techniques for Weibull Parameters' Estimation and the Performance of Wind Energy Conversion Systems in Nigeria. *IIUM Engineering Journal*. 2023, vol. 24, no. 1, pp. 138—157. DOI: 10.31436/iiumej.v24i1.2611.

[24] ROSALIND A., N. GARY, D. STUART, W. HAMISH. Wind Turbine Interference in a Wind Farm Layout Optimization Mixed Integer Linear Programming Model. *Wind Engineering*. 2011, vol. 35, iss. 2, pp. 165–175. DOI: 10.1260/0309-524X.35.2.165.

[25] WAsP software. Accessed January 2024, accessed January 2024. Available at: http://www.wasp.dk.

[26] windPRO software. Accessed January 2024, accessed January 2024. Available at: http://www.emd.dk/windpro.

[27] LE T. T., D. N. VO. Optimal Layout for Offshore Wind Farms Using Metaheuristic Search Algorithms. *GMSARN International Journal*. 2017, vol. 11, pp. 1–15.

[28] Gary L. J.. Wind Energy Systems. *Manhattan, Kansas State University*. 2006.

[29] KUSIAK A., Z. SONG. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*. 2010, vol. 35, iss. 3, pp. 685-–694. DOI: 10.1260/0309-524X.35.2.165.

[30] BASTANKHAH M., F. PORTÉ-AGEL. A new analytical model for wind-turbine wakes. *Renewable Energy*. 2014, vol. 70, pp. 116-123. DOI: 10.1016/j.renene.2014.01.002.

[31] STEFAN L.. Configuration study of larger wind park. *Dissertation, Chalmers University of Technology*. 2003.

[32] NIEMEYER L., L. PIETRONERO, H. J. WIESMAN. Fractal Dimension of Dielectric Breakdown. *Physical Review Letters*. 2010, 1984, vol. 52, pp. 1033-1037. DOI: 10.1103/PhysRevLett.52.1033.

[33] RAJ, M. S. M., M. ALEXANDER, M. LYDIA. Modeling of wind turbine power curve. *ISGT2011-India, Kollam, India*. 2011, pp. 144–148. DOI: 10.1109/ISET-India.2011.6145371.

[34] POBO˘CÍKOVÁ I., M. MICHALKOVÁ, Z. SEDLIA˘CKOVÁ, D. JURÁŠOVÁ. Modelling the Wind Speed Using Exponentiated Weibull Distribution: Case Study of Poprad-Tatry, Slovakia. *Applied Sciences*. 2003, vol. 13, no. 6. DOI: 10.3390/app13064031.

[35] HUSSAIN I., A. HAIDER, Z. ULLAH, M. RUSSO, G. M. CASOLINO, B. AZEEM. Comparative Analysis of Eight Numerical Methods Using Weibull Distribution to Estimate Wind Power Density for Coastal Areas in Pakistan. *Energies*. 2023, vol. 16, no. 3. DOI: 10.3390/en16031515.

[36] HAMIT S.. Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*. 2015, vol. 75, pp. 1–18. DOI: 10.1016/j.knosys.2014. 07.025.

[37] TRAN, T. T., K. H. TRUONG, D. N. VO. Stochastic fractal search algorithm for reconfiguration of distribution networks with distributed generations. *Ain Shams Engineering Journal*. 2020, vol. 11, iss. 2, pp. 389–407. DOI: 10.1016/j.asej.2019.08.015.

[38] SHARMA K., M. R. AHMED. Wind energy resource assessment for the Fiji Islands: Kadavu Island and Suva Peninsula. *Renewable Energy*. 2016, vol. 89, pp. 168—180. DOI: 10.1016/j.renene.2015.12.014.

[39] RAMADAN H. S.. Wind energy farm sizing and resource assessment for optimal energy yield in Sinai Peninsula, Egypt. *Journal of Cleaner Production*. 2017, vol. 161, pp. 1283-–1293. DOI: 10.1016/j.jclepro.2017.01.120.

[40] DU, W.-B., Y. GAO, C. LIU, Z. ZHENG, Z. WANG. Adequate is better: particle swarm optimization with limited-information. *Applied Mathematics and Computation*. 2015, vol. 268, pp. 832–838. DOI: 10.1016/j.amc.2015.06.062.

[41] AYALA, H. V. H., et al. Design of heat exchangers using a novel multiobjective free search differential evolution paradigm. *Applied Thermal Engineering*. 2016, vol. 94, pp. 170–177. DOI: 10.1016/j.applthermaleng.2015.10.066.

[42] COELHO, L. D. S., V. C. MARIANI, F. A. GUERRA, M. V. F. DA LUZ, J. V. LEITE. Multiobjective Optimization of Transformer Design Using a Chaotic Evolutionary Approach. *IEEE Transactions on Magnetics*. 2014, vol. 50, no. 2, pp. 669–672. DOI: 10.1109/TMAG.2013.2285704.

[43] NETO, J. X. V., et al. Wind turbine blade geometry design based on multi-objective optimization using metaheuristics. *Energy*. 2018, vol. 621, pp.645–658. DOI: 10.1016/j. energy.2018.07.186.

[44] SILVA, M. A. C., C. E. KLEIN, V. C. MARIANI, L. S. COELHO. Multiobjective scatter search approach with new combination scheme applied to solve environmental/economic dispatch problem. *Energy*. 2013, vol. 531, pp. 14–21. DOI: 10.1016/j.energy.2013.02.045.