Abstract— One of the main issues of the wind plant power generation nowadays is that the current stand alone controller of each turbine in the wind plant is not able to cope with chaotic nature of wake aerodynamic effect. Therefore, it is necessary to re-tune the controller of each turbine in the wind plant such that the total power generation is improved. This article presents an investigation of a data driven approach using moth-flame optimization algorithm (MFO) to the problem of improving wind plants power generation. The MFO based technique is applied to search the turbine’s optimum controller such that the aggregation power generation of a wind plant is maximized. The MFO is a population based optimization method that mimics the behavior of moths that navigate on specific angle with respect to the moon location. Here, it is expected that the MFO can solve the control accuracy problem in the existing algorithms for maximizing wind plant. A row of wind turbines plant with wake aerodynamic effect among turbines is adopted to demonstrate the effectiveness of the MFO based technique. The model of the wind plant is derived based on the real Horns Rev wind plant in Denmark. The performance of the proposed MFO algorithm is analyzed in terms of the statistical analysis of the total power generation. Numerical results show that the MFO based approach generates better total wind power generation than spiral dynamic algorithm (SDA) based approach and safe experimentation dynamics (SED) based approach.

Keywords— Moth-flame Optimization (MFO); data driven; wind plant optimization; power generation; alternative energy.
method in improving the total power generation of the existing wind plant.

This article investigates the effectiveness of moth-flame optimization (MFO) as a data-driven approach for improving the power generation of a row of wind turbines plant. Next, the assessment in terms of, best, worst and standard deviation of wind plant power generation is presented. Moreover, the results are also compared with SDA and SED [1] based approaches. The combination of the number of agents or populations and iterations is also investigated since this combination may influence the performance of the algorithm in producing the optimum results.

The structure of this article is organized as follows. Section II explains material and method, which consists of the formulation of the wind plant optimization problem, the MFO algorithm and the procedure to adopt it in a data-driven approach. In Section III, the data-driven based MFO is verified to a row of wind turbines plant. Then, a performance comparison between the MFO, Spiral Dynamic Algorithm, and the Safe Experimentation Dynamics based approaches are also elaborated in Section III. In the final section, we conclude our findings.

II. MATERIAL AND METHOD

A. Problem Formulation

For this study, consider a wind plant with a \( N \) number of wind turbines with either random or deterministic formation of wind turbine position. The controller of the turbine \( k \) is denoted as \( s_k(k = 1,2,\ldots, N) \), which is a generic symbol of the turbine regulators, like pitch the angle of the blade and speed of turbine the e motor [20]. The power generation of turbine \( k \) is expressed by \( f_k(s_1, s_2,\ldots, s_N) \). The time-varying magnitude of wind speed with different direction is considered in this investigation. Hence, the controller \( s_1, s_2,\ldots, s_k, s_{k+1},\ldots, s_N \), which are not included turbine \( k \), might also affect the power generation of turbine \( k \), i.e., \( f_k \). This is due to the aerodynamic interaction among turbines. In the same way, any changes of the controller \( s_k \) not only change the power generation \( f_k \) but also the power generation of other turbines, i.e., \( f_1, f_2,\ldots, f_{k-1}, f_{k+1},\ldots, f_N \). Therefore, we can state that the power generation of turbine \( k \) is highly affected by the controller \( s_k \) and is weakly affected by other controllers, i.e., \( s_1, s_2,\ldots, s_{k-1}, s_{k+1},\ldots, s_N \).

The relation between power generation \( f_k \) and controllers \( s_1, s_2,\ldots, s_N \) is assumed to be unknown since the turbulence behavior between turbines are very complex in reality and it is hard to get a precise wind plant’s dynamic model.

Nevertheless, it is assumed that the aggregate power generation of the wind plant is observable where it is expressed by:

\[
\bar{f}(s_1, s_2,\ldots, s_N) = \sum_{k=1}^{N} f_k(s_1, s_2,\ldots, s_N)
\]  

Finally, the wind plant control problem is stated by:

**Problem 1.** Consider the wind plant aggregate power generation \( \bar{f}(s_1, s_2,\ldots, s_N) \) is given in (1) and let functions \( f_k(k = 1,2,\ldots,N) \) are unknown with respect to its controller \( s_k(k = 1,2,\ldots,N) \). Next, find controller \( s_k \) such that \( \bar{f}(s_1, s_2,\ldots, s_N) \) is maximized.

B. Moth-Flame Optimization Algorithm

Firstly, define \( f: R^m \rightarrow R \) as a loss function and \( P_i \in R^m(i = 1,2,\ldots,n) \) is the tuning parameter for \( n \) size of populations. Next, for \( i = 1,2,\ldots,n \), a maximization problem is given by

\[
\max_{P_1,P_2,\ldots,P_n} f(P_i).
\]  

The Moth-Flame Optimization algorithm updates \( P_i(i = 1,2,\ldots,n) \) using

\[
P_i(t + 1) = \begin{cases} 
G_i e^{b_0} \cos(2 \pi t) + C_j(t), & \text{if } i \leq \text{flame no} \\
G_i e^{b_0} \cos(2 \pi t) + C_{\text{flame no}}(t), & \text{if } i > \text{flame no},
\end{cases}
\]  

where \( i = 1,2,\ldots,n \) for \( t = 0,1,\ldots \). This equation is referred as the logarithmic spiral equation that used to update the next position for each moth. Here, \( P_i \) indicates the moth while \( C_j(= 1,2,\ldots, \text{flame no}) \) represents the flame. In particular, if \( i \leq \text{flame no} \), then \( P_i(t + 1) \) is updated according to \( C_j \), where \( C_j = C_j \in R^m, i = 1,2,\ldots,n \). Meanwhile, for \( i > \text{flame no} \), \( P_i(t + 1) \) is updated according to \( C_{\text{flame no}} \). The symbol \( G_i \) represents the displacement between the \( i^{th} \) moth and the \( j^{th} \) flame, which is calculated as follows:

\[
G_i = \left| C_j(t) - P_i(t) \right|, 
\]  

In equation (3), \( b \) is a constant for describing the shape of the logarithmic spiral, while \( o \) is a generated random number between \( y \) to 1, where \( y \) is linearly decreasing gain from \(-1\) to \(-2\) iteratively, as shown in (5)

\[
y = -1 + t \times \left( \frac{-1}{t_{\text{max}}} \right).
\]  

Here, \( t \) is the current number of iteration and \( t_{\text{max}} \) is the maximum number of iterations. In order to get high exploitation of the promising solutions, the number of flames with respect to the iteration number is proposed as below:

\[
\text{flame no} = \text{round} \left( \frac{C_{\text{max}} - t \times \frac{C_{\text{max}} - t}{t_{\text{max}}}}{t_{\text{max}}} \right).
\]  

where \( C_{\text{max}} \) is maximum number of flames, \( t \) is the current number of iteration and \( t_{\text{max}} \) is the maximum number of iteration. Then, the procedure of the MFO algorithm is given by:

**Step 1:** Determine the size of populations \( n \), maximum number of iterations \( t_{\text{max}} \) and the constant \( b \). Let algorithm start with \( t = 0 \).
Step 2: Determine the first tuning parameter $P_i(0) \in R^m$, $i = 1, 2, ..., n$ arbitrarily in a searching space. Compute the flame number equation in (6). Then, sort $P_i(0)$ in descending order where from higher value of objective function to lower value of objective function and find the best solution $P^*$. Here, $P^* = P_i(0)$ for $i_c = \arg \max_i f(P_i(0)), i = 1, 2, ..., n$. Next, store the result at $C_i(0) \in R^m$, $i = 1, 2, ..., n$. Proceed with step 4.

Step 3: Compute the flame no at (4). Next, merge $P_i(t - 1)$ and followed by $C_i(t - 1)$ as follows

$$\text{merged population} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix}$$ (7)

then sort merged population in descending order where from higher value of objective function to lower value of objective function. Select the best $n$ positions from merged population as the flame and store the result at $C_i(t) \in R^m$, $i = 1, 2, ..., n$. Find the best solution, $P^*$ in $P^* = P_i(t)$, where $i_c = \arg \max_i f(P_i(t)), i = 1, 2, ..., n$.

Step 4: Execute the MFO algorithm in (3).

Step 5: Update $t = t + 1$. Proceed Step 3 if a termination criterion $t_{max}$ is not achieved. In another way, the procedure terminates with the optimal tuning parameter $P^*$.

In Step 5, the termination of the procedure is selected from the maximal iterations, where the procedure terminates once a pre-specified $t_{max}$ is achieved. In this case, a preliminary trials is performed to observe its convergence curve in order to decide the maximum iteration.

C. Data Driven Design

Note that the presented MFO algorithm in Section II-B is generic algorithm that can be applied for many engineering optimization problems. Therefore, another procedure is required to apply the MFO algorithm for wind plant optimization problem. By applying the MFO technique in the previous section, the data driven MFO based technique for finding the optimal controller of wind plant power generation is given by:

**Procedure I:** Determine the number of iterations $t_{max}$.

**Procedure II:** Apply the MFO procedure in Section II-B by denoting $f = f$ ands $P_i$.

**Procedure III:** This data-driven procedure terminates after $t_{max}$. The optimal controller $s^* = P^*$ and the corresponding aggregate power generation $\bar{f}$ is observed.

III. RESULTS AND DISCUSSION

Now, we validate the data-driven based MFO algorithm in improving wind plant power generation. Initially, a wind plant model that represents an actual wind plant is adopted to assess the data-driven method. Here, the model of the wind plant, which is take from [20] is explained. Next, the MFO algorithm technique is used to a row wind plant model.

A. Dynamic Model of Wind Plant

Let $x = 1, 2, ..., n$, be the set of $n$ turbines in the wind plant, the approaching wind speed is defined by $v_a$, the diameter of turbine motor is denoted by $D_k$, the region of motor swept of turbine $l$ is defined by $A_l$. The roughness coefficient that depicts the gradient of wake propagation is denoted by the symbol $G$, the overlay area between the turbine $kwake$ and turbine $lmotor$ swing area is defined by $A_k^{cw,l}$. The notation $(z, r_m)$ is represented as a center point in the wake of the turbine with the length to the turbine motor circle plane and $r_m$ is the length to the center of the turbine motor axis. Next, the resultant wind speed is given by:

$$\bar{v}_l = v_a \left[ 1 - 2 \sum_{k \in x, l \in x} q_k \frac{D_k}{D_k + 2G(z_l - r_m)} A_k^{cw,l} A_l \right]$$ (8)

where $z_k$ is the length to the turbine $k$ motor circle plane, while $z_l$ is the length to the turbine $l$ motor circle plane. Figure 1 depicts the demonstration of wake interaction between the two turbines. For $l \in x$, the wind speed $\bar{v}_l$ is computed based on the wind speed resultant deficit generated by each front turbine. We assume that the wake grows proportionally to the length $z$ and its diameter has a round cross-section. Note that, in reality, we may not expect an ideal proportionality of wake with round cross-section. Furthermore, the individual turbine power is expressed by:

$$J_l = 2\rho A_l s_l (1 - s_l) \bar{v}_l^3$$ (9)

where $\rho$ is the air density.

**Remark 1.** Note that, in this study, our proposed data-driven MFO only use the measurable total power generation without even know the detailed model of wind plant in (8) and (9). In order to represent this dynamic model of the wind plant, the algorithm will capture the data of total power aggregation after the incoming wind has pass through all the turbines from the first row until the final row. In that case, our proposed method has a good potential to be applied in actual wind plant system since the data-driven MFO only capture the total power data without even know the complex aerodynamic interactions amongst turbines.
B. Numerical Evaluation

The performance of the MFO based algorithm is demonstrated using a ten turbines row of wind plant, which is illustrated in Figure 2. The diameter of each wind turbine is 80 m. The length between each turbine is equivalent to the total diameter of seven turbines, which is 560 m. Other coefficients of wind plant, such as air density $\rho$, roughness coefficient $\phi$ is taken from [7]. In this study, the performance of the MFO, is benchmarked with the SDA based method and SED based method.

Firstly, the wind speed is assumed to be constant at $V_m = 8$ m/s. Then, MFO coefficients are set after run for several preliminary experiments, where $b$ is set between 0.75 to 1.0 with 0.05 increment. The coefficients of SDA based approach is denoted by $yr = 0.97$ and $\alpha = \pi/4$. Meanwhile, the coefficients of SED based approach with updated step size $K_z = 0.03$ and the probability to update the tuning parameter $E = 0.3$ are used. Note that the MFO based method only requires one pre-defined parameter $b$; while the SDA and SED based methods require two pre-defined parameters. Hence, the MFO based method requires less effort to fine-tune the pre-defined parameter. The initial controller value of each turbine is set between 0 and 0.3333. In order to observe the stochastic behavior in the proposed approaches, 100 trials are performed to MFO, SED and SDA based approaches.

Table 1 tabulates the statistical evaluation of the aggregate power generation for MFO based approach after 10000 evaluations. The optimal value of $b$ is selected based on the maximum value for the best, mean and worst of total power generation and minimum value of standard deviation. Notice that there are four data that reach higher best total power generation (4.7648415724 MW) which are $b = 0.95, 0.80, 0.75$, and $0.70$. Moreover, $b = 0.95$ produces slightly lower standard deviation value than other $b$ values. This shows that the best value of $b$ for defining the shape of logarithmic spiral is $b = 0.95$. Table 2 tabulates the total power generation analysis of MFO based approach compared to SDA and SED based approaches. It is clearly shown that the MFO based method yields higher best total power generation (4.7648415724 MW) than the SDA (4.7648415723 MW) and SED (4.7644075485 MW) based approaches. The average and lowest values of the total power generation also shows the same pattern. MFO algorithm also reaches lowest standard deviation than the SDA and SED based approaches. This result verifies that MFO algorithm can achieve maximum total power generation consistently.
Fig. 3 Graph Response Total Power Generation of MFO algorithm

Fig. 4 Graph Response Total Power Generation of SDA algorithm

Fig. 5 Graph Response Total Power Generation of SED algorithm
TABLE I
Performance Comparison of Total Power Generation for different Value of b

<table>
<thead>
<tr>
<th>b</th>
<th>Mean (MW)</th>
<th>Best (MW)</th>
<th>Worst (MW)</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>4.7641691656</td>
<td>4.7648415724</td>
<td>4.6976008998</td>
<td>6.7241e+03</td>
</tr>
<tr>
<td>0.95</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>5.6161e-10</td>
</tr>
<tr>
<td>0.90</td>
<td>4.7630743588</td>
<td>4.7648415724</td>
<td>4.6764808948</td>
<td>1.2433e+04</td>
</tr>
<tr>
<td>0.85</td>
<td>4.7641691656</td>
<td>4.7648415724</td>
<td>4.6976008999</td>
<td>6.7241e+03</td>
</tr>
<tr>
<td>0.80</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>4.3903e-10</td>
</tr>
<tr>
<td>0.75</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>3.7441e-10</td>
</tr>
<tr>
<td>0.70</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>4.7648415724</td>
<td>3.7441e-10</td>
</tr>
</tbody>
</table>

TABLE II
The Comparison Performance of Total Power Generation (MW) between MFO, SDA and SED based approach

<table>
<thead>
<tr>
<th>Statistical Evaluation</th>
<th>MFO</th>
<th>SDA</th>
<th>SED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>4.7648415724</td>
<td>4.7648415723</td>
<td>4.7644075485</td>
</tr>
<tr>
<td>Highest</td>
<td>4.7648415724</td>
<td>4.7648415723</td>
<td>4.7648415242</td>
</tr>
<tr>
<td>Lowest</td>
<td>4.7648415724</td>
<td>4.7648415723</td>
<td>4.7627457259</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>5.6161x10^{-10}</td>
<td>1.1039824x10^{-2}</td>
<td>4.513106x10^{02}</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

This paper presents a data-driven method based on moth-flame optimization (MFO) algorithm has been investigated. This study aims to propose a MFO for a power generation of wind plant and compare the findings with SDA and SED based approaches. In this simulation results, the MFO based method exhibits a slightly higher total power generation than the SDA and SED based approaches. This proves the potential of MFO based approach for data driven method of wind plant control. In the future, it is necessary to improve the convergence speed of MFO since it will take longer time if the size of wind plant is large. In this case, one might consider a multi-resolution version of MFO to increase the convergence speed.

ACKNOWLEDGMENT

Universiti Malaysia Pahang Internal Research Grant PGR1703105 under the Research and Innovation Department, University of Malaysia Pahang (UMP) assisted our study for this project.

REFERENCES