

A Hierarchical Data-driven Wind Farm Power Optimization Approach Using Stochastic Projected Simplex Method

Zhiwei Xu, *Student Member, IEEE*, Hua Geng, *Senior Member, IEEE*, and Bing Chu

Abstract—In a wind farm, the interactions among the wind turbines through wakes can significantly reduce the power output of the wind farm. These together with the complex wind conditions make the power optimization problem of the wind farm very challenging. To address this problem, this paper proposes a hierarchical data-driven power optimization scheme, which does not need a wake interaction model that can be rather difficult to develop due to the complex aerodynamics between the turbines. The proposed scheme consists of two steps: firstly the power optimization problem of the wind farm is divided into several optimization sub-problems to deal with the complex wind conditions based on the wind farm power efficiencies in different wind directions. Secondly, a data-driven stochastic projected simplex algorithm is developed to solve the power optimization sub-problems. The proposed algorithm can increase the power output of the wind farm by using measurement data only and has the ability to find the optimal solutions. Finally, simulation results show that the proposed scheme can efficiently improve the power output of the wind farm in different wind conditions compared with some benchmark methods.

Index Terms—Wind farm, wake interaction, power optimization, data-driven, stochastic projected simplex method.

I. INTRODUCTION

IN recent years, wind power, as an environmental friendly renewable energy source [1], is under fast development due to the need for greener electricity system and the increasing electricity demands [2]. However in a wind farm, wakes generated by upstream wind turbines can significantly degrade the power output of the downstream wind turbines due to reduced wind speed inside the wake [3]. As a result, the greedy policy that is widely applied in practice, where each turbine is devoted to maximizing its own power output [4], often leads to the suboptimal power output of wind farm because such policy neglects the wake interactions among the turbines [5], [6]. This has been investigated experimentally [7]–[9] and the results indicate that there could be up to 33% power loss for wind farm under some worst case scenarios. To mitigate the wake interactions among the turbines and optimize the power output of wind farm, the cooperative control of wind farm has attracted the great interest of researchers [3]. The proposed

control strategies for wind farm can be classified into two categories, the model-based methods and data-driven methods.

Model-based methods usually have the following three steps. At first, the power generation model of wind farm is built, which mainly includes modeling the wake of wind turbine, wake interactions among the turbines, and power generation system of wind turbine. Then, the power optimization problem of wind farm is formulated. The control strategy of wind farm is finally developed. Traditionally, based on the analytical power generation models of wind farm, some optimization methods are applied to solve the wind farm power optimization problem, e.g. the heuristic algorithm [10] and steepest descent method [11]. However, these methods may not efficiently improve the power output of wind farm as they highly depend on the analytical models of wind farm used in optimization. These models are usually built based on simplified wake models, which could not accurately reflect the actual aerodynamics of the wake [12], especially for some wind farms sited in coteau or highland area. To overcome the limitation, the control strategies based on Computational Fluid Dynamics (CFD) models are proposed, e.g. the conjugate gradient method [13] with large eddy simulation [14]. The use of CFD simulations requires significant computational resources which is usually not available in practice even if it improves the model accuracy [15]. Consequently, the model-based methods have various difficulties in the power optimization of wind farm.

To overcome the dependence on the power generation model of wind farm, the data-driven methods have received great attentions. The methods aim to maximize the output power of wind farm with only the control inputs and some measurement data. A number of methods have been proposed, e.g. game theoretic learning algorithms including safe experimentation dynamics (SED) and payoff-based distributed learning [6], Bayesian ascent algorithm [12], simultaneous perturbation approach [16], discrete adaptive filtering algorithms [17], random search method [18], et al. Most of the above results consider simple wind conditions, such as static and slowly changing wind. However, the wind condition in a actual farm is often quite complex, which can change randomly and quickly. It means that the optimal joint control action of the wind farm likely varies quickly over time and the power optimization problem of the wind farm is a very challenging dynamic optimization problem. The above data-driven methods have significant difficulties as they need to quickly relearn the actions and even fail to achieve so.

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61722307, Grant U2066602, and Grant 52061635102. (Corresponding author: Hua Geng.)

Z. Xu and H. Geng are with the Department of Automation, Tsinghua University, and Beijing National Research Center for Information Science and Technology, Beijing 100084, China (e-mail: genghua@tsinghua.edu.cn).

B. Chu is with the School of Electronics and Computer Science, University of Southampton, Southampton SO171BJ, U.K.

This paper presents a hierarchical data-driven power optimization scheme to efficiently improve the power output of wind farm in complex wind conditions. Based on the power efficiencies of wind farm with greedy policy, the whole range of wind direction is divided into a finite number of sub-intervals, in which the power efficiencies of wind farm under different wind directions are similar and thus it can be assumed that only a wake interaction pattern exists. The power optimization problem of wind farm is then formulated to be the sum of a series of optimization sub-problems defined in the sub-intervals. To solve the sub-problems, we develop a data-driven stochastic projected simplex (SPS) power optimization algorithm. The SPS algorithm fully exploits the advantages of the Nelder-Mead method (derivative-free, fast convergence), Gradient Projection method (constraints handling) and modified adaptive random search (global optima), which can solve the constrained nonlinear optimization problem without using gradient information.

With the scheme, a finite number of power optimization algorithms are run in parallel and each of them only corresponds to one sub-problem. The advantages of this scheme are listed as follows: (1) The proposed algorithm can quickly improve the power output of wind farm when solving optimization sub-problems; (2) The proposed algorithm can find the optimal solutions of the optimization sub-problems; (3) With the change of wind direction from one sub-interval to another sub-interval, the switch of two proposed algorithms will be performed. This will utilize the already learned knowledge and thus improve the convergence speed of the algorithms, avoiding repetitiously solving the sub-problems from scratch. Precisely because of these merits, the proposed scheme can work well even in complex wind conditions.

The remainder of the paper is organized as follows. Section II describes the power generation model of wind farm. Section III formulates the power optimization problem of wind farm for complex wind conditions. The power optimization scheme is proposed in Section IV. And Section V compares the performance of different schemes for the power optimization of wind farm with simulation results. Finally, conclusion and possible directions for future research are given in Section VI.

II. WIND FARM MODEL

In this section, the power generation model of wind farm is introduced.

The wind farm with n wind turbines is considered in this study and let $N = \{1, 2, \dots, n\}$ be the set of all turbines. Suppose that the yaw control of all turbines has been designed and can guarantee their blade disk planes be perpendicular to the wind direction. The control action of wind turbine $i \in N$ is then chosen as its axial induction factor (AIF) u_i , which can be adjusted by the blade pitch and generator torque. The AIF is a measure of the wind velocity reduction over rotor plane and provides a relatively simple expression for the cooperative control of wind farm [5]. The admissible set of the u_i is given by the set $\mathcal{U}_i = \{u_i | u_{i,min} \leq u_i \leq u_{i,max}\}$, where $u_{i,min}$ and $u_{i,max}$ are respectively the lower bound and upper bound of u_i . The joint AIF of all turbines is represented by the tuple

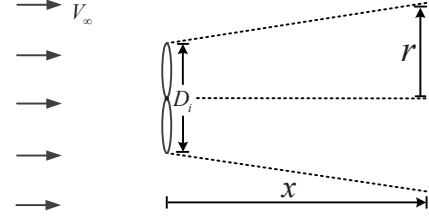


Fig. 1. Single turbine wake example [17].

$u = (u_1, \dots, u_n)$, whose admissible set is denoted as $\mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_n$, where $\times$ is the Cartesian product.

When one wind turbine extracts energy from the wind, it will cause changes of the downstream wind flow. The altered flow is called the wake of wind turbine, through which the upwind turbine will affect the wind speed and output power of downwind turbines and thus decrease the power output of wind farm. The wake has many complex characteristics, e.g. recovery, meandering, and dependence on environment parameters [5]. The recovery is that the wake velocity gradually recovers to the freestream velocity. The meandering is a large-scale stochastic phenomenon of wake that the wake structure will show horizontal and vertical oscillations over time rather than maintaining a certain fixed shape. The wake is also parameter-dependent since the external variables (such as temperature and wind condition) can affect the behavior of the wake. Therefore, the math description of the wake interactions among the turbines is one key modelling challenge in wind farm control [5]. The Park model [19] is the one of the most popular wake models and has wide application in wind farm control [16], [17]. It is also applied in this paper to resemble the interactions between the turbines.

Consider the situation in Fig. 1 for turbine i to illustrate the wake effect. In Fig. 1, V_∞ is the free stream wind speed, D_i denotes the diameter of turbine i , x is the distance from turbine i along the wind direction, and r is the distance orthogonal to the wind direction. Between the top and bottom dotted lines is the wake area generated by turbine i . Denote $V_i(x, r, u_i, V_\infty)$ as the wake velocity profile at point (x, r) generated by turbine i with the AIF u_i . Then,

$$V_i(x, r, u_i, V_\infty) = V_\infty (1 - \delta V_i(x, r, u_i)), \quad (1)$$

where $\delta V_i(x, r, u_i)$ represents the fractional deficit of the velocity at the point (x, r) . According to Park model [19], the $\delta V_i(x, r, u_i)$ is expressed as

$$\delta V_i(x, r, u_i) = \begin{cases} 2u_i \left(\frac{D_i}{D_i + 2\kappa x} \right)^2, & \text{for any } r \leq \frac{D_i + 2\kappa x}{2}, \\ 0, & \text{for any } r > \frac{D_i + 2\kappa x}{2}, \end{cases} \quad (2)$$

where κ is the roughness coefficient that measures the slope of the wake expansion.

The aggregate wind velocity $V_i(\{u_j\}_{j \in \mathcal{N}_i}, V_\infty, \theta)$ at an arbitrary wind turbine $i \in N$ can be expressed as

$$V_i(\{u_j\}_{j \in \mathcal{N}_i}, V_\infty, \theta) = V_\infty \left(1 - \delta V_i(\{u_j\}_{j \in \mathcal{N}_i}, \theta) \right) \quad (3)$$

where $\mathcal{N}_i$ is the set of upstream turbines that are coupled with the turbine i via the wakes, θ is wind direction. Based on the Park model, the aggregated velocity deficit $\delta V_i(\{u_j\}_{j \in \mathcal{N}_i}, \theta)$

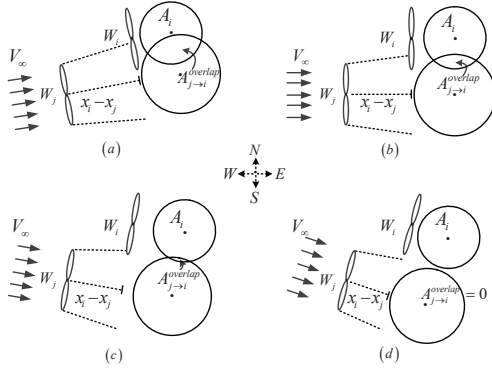


Fig. 2. Two-turbine wake interaction examples in different wind directions.

at turbine i can be formulated as

$$\delta V_i(\{u_j\}_{j \in \mathcal{N}_i}, \theta) = 2 \sqrt{\sum_{j \in \mathcal{N}_i} \left(u_j \left(\frac{D_j}{D_j + 2\kappa(x_i(\theta) - x_j(\theta))} \right)^2 \frac{A_{j \rightarrow i}^{overlap}(\theta)}{A_i} \right)^2}, \quad (4)$$

where x_i is the distance of turbine i from a common vertex along the θ , A_i is the area of the disk generated by the blade of turbine i , $A_{j \rightarrow i}^{overlap}$ is the part area of the A_i that overlaps with the wake generated by turbine j and is associated with θ . The wake interaction examples with two wind turbines are given in Fig. 2, where turbine i is denoted as W_i . From Fig. 2(a) to Fig. 2(d), the turbine distance $x_i - x_j$ and the $A_{j \rightarrow i}^{overlap}$ gradually vary with clockwise changes of the θ , which leads to different wake interaction patterns among the turbines.

The power generated by turbine $i \in N$ can be characterized by

$$P_i(u_i; \{u_j\}_{j \in \mathcal{N}_i}, V_\infty, \theta) = \frac{1}{2} \rho A_i C_{p,i}(u_i) V_i(\{u_j\}_{j \in \mathcal{N}_i}, V_\infty, \theta)^3, \quad (5)$$

where ρ is the density of air and $C_{p,i}(u_i)$ is the power coefficient defined as

$$C_{p,i}(u_i) = 4u_i(1 - u_i)^2. \quad (6)$$

The total power output of wind farm is simply the sum of the power generated by all turbines, namely

$$P(u; V_\infty, \theta) = \sum_{i=1}^n P_i(u_i; \{u_j\}_{j \in \mathcal{N}_i}, V_\infty, \theta). \quad (7)$$

Remark 1: The goals of wake model and wake interaction model are respectively to effectively identify the velocity deficit $\delta V_i(x, r, u_i)$ in (1) and the aggregated velocity deficit $\delta V_i(\{u_j\}_{j \in \mathcal{N}_i}, \theta)$ in (3). However, it is difficult to achieve the goals accurately due to the complexity of wake. In this paper, the Park model will only be used for simulating the wake, whose uncertainties do not influence the performance evaluation of the proposed scheme in the power optimization problem of wind farm because it is not used in the control design. This model is widely used in many references to evaluate the effectiveness of wind farm power optimization scheme, including [6], [16], [17], et al.

III. WIND FARM POWER OPTIMIZATION PROBLEM FOR COMPLEX WIND CONDITIONS

In this section, the power optimization problem of wind farm is formulated. It is further expressed as the sum of several

optimization sub-problems based on the power efficiencies of wind farm to deal with the complex wind conditions.

The optimization goal of wind farm in this paper is to improve its total power output (7). More specifically, the optimal joint AIF should be obtained by solving the following optimization problem without using the wake interaction model:

$$u_{opt} \in \arg \max_{u \in \mathcal{U}} P(u; V_\infty, \theta). \quad (8)$$

The above (8) is a nonlinear optimization problem with linear constraint. The wake interaction pattern varies over time due to the time-varying characteristic of wind direction. It leads to that the optimal joint control action of the (8) does not stay fixed. The developed optimization scheme is required to track the variation of wind direction to achieve the optimization goal of wind farm.

Remark 2: From (5) and (6), it can be noticed that the $u_i = 1/3$ for wind turbine $i \in N$ is the optimal control action in terms of maximizing its power output and is thus called greedy policy. However, as mentioned earlier, the greedy policy might not be optimal in maximizing the total output power of wind farm due to the wake interactions among the turbines in (4).

The power efficiency function [3] [12] of wind farm can be defined as

$$\eta(u; \theta) \triangleq (1/n) \sum_{i=1}^n \eta_i(u_i; \{u_j\}_{j \in \mathcal{N}_i}, \theta), \quad (9)$$

where $\eta_i(u_i; \{u_j\}_{j \in \mathcal{N}_i}, \theta) = P_i/P_i^*$ denotes the power efficiency of turbine i , $P_i^* = (1/2) \rho A_i C_{p,\max} V_\infty^3$ is the power output of turbine i without wake interaction, $C_{p,\max}$ is the maximum power coefficient and calculated by (6) with $u_i = 1/3$. Based on (3) and (5), it can be derived that

$$\eta_i(u_i; \{u_j\}_{j \in \mathcal{N}_i}, \theta) = \frac{C_{p,i}(u_i) \left(1 - \delta V_i(\{u_j\}_{j \in \mathcal{N}_i}, \theta)\right)^3}{C_{p,\max}}. \quad (10)$$

It is assumed that all wind turbines are identical. Then $P_i^* = P_j^*$. Let $P^* = P_j^*$. From (9),

$$\eta(u; \theta) = \frac{1}{nP^*} \sum_{i=1}^n P_i. \quad (11)$$

Remark 3: The formulas (9) and (10) indicate that the $\eta(u; \theta)$ is unrelated to the freestream wind speed V_∞ . The (11) shows that the $\eta(u; \theta)$ can be regarded as the normalization of the power output of wind farm below the rated wind speed, whose base value is the power output nP^* of wind farm without wake interactions. This means that the maximization of $\eta(u; \theta)$ can ensure the maximization of the power output of wind farm.

Although greedy policy itself does not consider the wake effect, the power generation efficiency of wind farm under greedy policy can reflect the coupling strength among the turbines as the wake effect is an inherent characteristic of wind farm. The smaller the $\eta(u; \theta)$ is, the stronger the wake interactions between the turbines are. If the changes of wind direction in an interval only lead to a small changes of the power generation efficiency of wind farm, the coupling strength among the turbines is similar and only one wake

interaction pattern needs to be considered for the interval. Then based on this, the entire interval of wind direction in this paper is divided into a certain number of sub-intervals in the following content.

According to the historical power generation data of actual wind farm with greedy policy, the values of power efficiency $\eta(u; \theta)$ under all wind directions can be calculated by (11). Based on the values of $\eta(u; \theta)$, the whole range of θ can be divided into a finite m number of sub-intervals, denoted by $\Theta_1, \Theta_2, \dots, \Theta_m$ and shown in Fig. 3(a). $\Theta_j = [\theta_{j,\min}, \theta_{j,\max}]$ is the j th sub-interval of θ , in which $\theta_{j,\min}$ and $\theta_{j,\max}$ are respectively the lower bound and upper bound of Θ_j , $j = 1, 2, \dots, m$. When $\theta_1, \theta_2 \in \Theta_j$,

$$|\eta(u; \theta_1) - \eta(u; \theta_2)| \leq \varsigma, \quad (12)$$

where $\varsigma \in [0, 1)$ is a small number. The selected ς must guarantee that the $\eta(u; \theta)$ makes minor changes that can be ignored for the fluctuations of θ in Θ_j . Thus it can be considered that there is only one wake interaction pattern for each divided sub-interval. Note that this division mainly depends on the layout and terrain of wind farm.

The power optimization sub-problem of wind farm for Θ_j is defined as

$$u_{opt,j} \in \arg \max_{u \in \mathcal{U}} P^j(u; V_\infty, \theta), \quad (13)$$

where P^j denotes the power output of wind farm under $\theta \in \Theta_j$, $j = 1, 2, \dots, m$. The sub-problem (13) can be approximately regarded as the static optimization problem, where the wake interaction pattern among the turbines is almost invariable such that the optimal joint AIF $u_{opt,j}$ under $\theta \in \Theta_j$ can be assumed as constant vector. Define

$$\alpha_j(\theta) = \begin{cases} 1, & \text{if } \theta \in \Theta_j, \\ 0, & \text{else,} \end{cases} \quad (14)$$

$\alpha(\theta) = \{\alpha_j(\theta), j = 1, 2, \dots, m\}$. Then the power optimization problem (8) of wind farm can be rewritten as

$$u_{opt} \in \arg \max_{u \in \mathcal{U}} \sum_{j=1}^m \alpha_j(\theta) P^j(u; V_\infty, \theta). \quad (15)$$

In (15), the (8) is formulated as the sum of m sub-problems associated with the sub-intervals of wind direction. Different sub-problems have different wake interaction patterns, which makes them have different optimal joint AIFs.

IV. WIND FARM POWER OPTIMIZATION SCHEME BASED ON STOCHASTIC PROJECTED SIMPLEX METHOD

In this section, the power optimization scheme of wind farm is proposed, which consists of a finite number of data-driven power optimization algorithms. Each algorithm is developed by using stochastic projected simplex method.

A. Power Optimization Scheme of Wind Farm

The power optimization scheme presented for wind farm is given in Fig. 3. As shown in Fig. 3(a), a finite m number of optimal algorithms are carried out to find the optimal joint AIFs of wind farm. Each of the algorithms only optimizes one sub-problem defined in (13), which is started when the

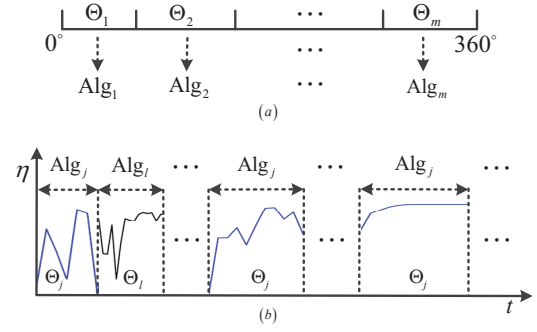


Fig. 3. power optimization scheme of wind farm.

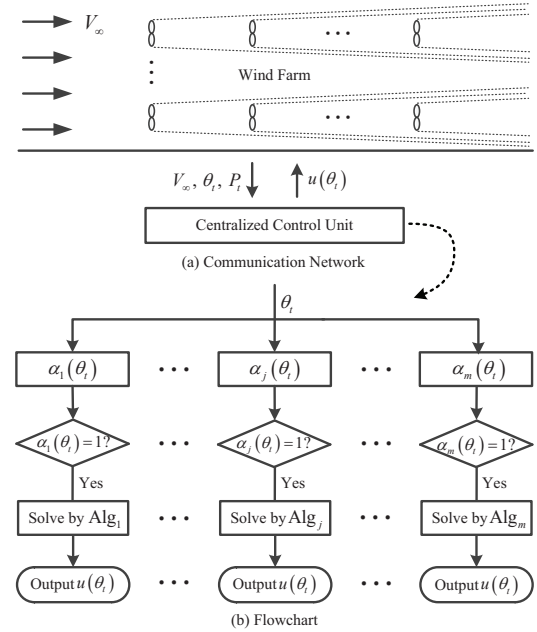


Fig. 4. Communication network and flowchart of the wind farm power optimization scheme.

corresponding sub-interval is visited by wind direction and stopped when wind direction leaves from the sub-interval. For example, in Fig. 3(b), the algorithm Alg_j is applied to obtain the $u_{opt,j}$ in $t \in \Theta_j$, where the t denotes t_{th} interaction between the control scheme and wind farm to get measurements and the θ_t is the measured wind direction in t_{th} interaction. With the switch of θ from Θ_j to Θ_l , the algorithm Alg_l is conducted. Meanwhile, the Alg_j is stopped and its operational data is saved. When the Θ_j is revisited by θ , the Alg_j is reactivated and continues searching for $u_{opt,j}$ based on the previous experience. Obviously, the proposed scheme makes full use of the learned knowledge, which can accelerate the convergence speed of algorithms. It is beneficial to quickly improve the power output of wind farm and increase the adaptability of the scheme for complex wind conditions.

The proposed power optimization framework is centralized, as shown in Fig. 4(a). The power optimization scheme is stored in centralized control unit, learning the optimal policy by many interactions with wind farm. One interaction consists of two steps. Step 1: The power optimization scheme decides an control action (joint AIF $u(\theta_t)$) according to the receiving the measurement data (V_∞, θ_t , and P_t) from actual wind farm

and sends this action to the wind farm; Step 2: The wind farm carries out the receiving control action and then feeds the corresponding response data and wind conditions back to the centralized control unit. Fig. 4(b) shows the flowchart of the power optimization scheme. When the θ_t is sent to the centralized control unit, the $\alpha(\theta_t)$ is calculated by (14). Only one unit in $\alpha(\theta_t)$ is equal to one for θ_t , which decides the optimization sub-problem to be solved. Without loss of generality, it is assumed that $\alpha_j(\theta_t)=1$ and $\alpha_l(\theta_t)=0, l \neq j$. A control action $u(\theta_t)$ is given for θ_t by the Alg_j.

B. Power Optimization Algorithm of Wind Farm

We now introduce our data-driven SPS algorithm to solve the power optimization sub-problems of wind farm.

The $q+1$ vertices of simplex are defined as the $q+1$ joint AIFs of all turbines, namely

$$U = \{u^0, u^1, \dots, u^q\} \quad (16)$$

where U is vertex set of the simplex, $u^l \in \mathcal{U}$, $l=0, 1, \dots, q$. The proposed SPS algorithm is described as follows.

Initialization: Evaluate power efficiency η at the points in the vertex set U_0 of the initial simplex. Choose constants

$$-1 < \delta^{ic} < 0 < \delta^{ro}, 0 < \delta^r < \delta^e \leq 1, 0 < \varepsilon_1, \varepsilon_2 < 1.$$

For $k = 0, 1, 2, \dots$

1) **Order:** Order the $q+1$ vertices of U_k so that

$$\eta_k^0 = \eta(u_k^0) \geq \eta_k^1 = \eta(u_k^1) \geq \dots \geq \eta_k^q = \eta(u_k^q), \quad (17)$$

where U_k is the vertex set of the simplex at iteration k , u_k^l is the l_{th} vertex of the U_k , η_k^l is the power generation efficiency of wind farm with u_k^l , $l = 0, \dots, q$. From (17), u_k^0 is the best vertex and u_k^q is the worst in U_k .

2) **Reflection:** Reflect the worst vertex u_k^q over the centroid $u^c = (1/q) \sum_{l=0}^{q-1} u_k^l$ of the remaining q vertices:

$$u^{ro} = \prod_{\mathcal{U}}(u^c + \delta^{ro}(u^c - u_k^q)), \quad (18)$$

$$u^r = u^c + \delta^r(u^{ro} - u^c), \quad (19)$$

where $\prod_{\mathcal{U}}(\bullet)$ is the Euclidean projection onto the $\mathcal{U}$, δ^{ro} and δ^r are the reflection parameters, u^r is the reflection point. Then evaluate $\eta^r = \eta(u^r)$. If $\eta_k^0 \geq \eta^r > \eta_k^q$, then replace u_k^q by the u^r and terminate the iteration:

$$U_{k+1} = \{u_k^0, u_k^1, \dots, u_k^{q-1}, u^r\}. \quad (20)$$

3) **Expansion:** If $\eta^r > \eta_k^0$, then calculate the expansion point u^e by

$$u^e = u^c + \delta^e(u^{ro} - u^c) \quad (21)$$

and evaluate $\eta^e = \eta(u^e)$, where δ^e is the expansion parameter. If $\eta^e \geq \eta^r$, replace u_k^q by the u^e , and terminate the iteration:

$$U_{k+1} = \{u_k^0, u_k^1, \dots, u_k^{q-1}, u^e\}; \quad (22)$$

Otherwise, replace u_k^q by the u^r , and terminate the iteration:

$$U_{k+1} = \{u_k^0, u_k^1, \dots, u_k^{q-1}, u^r\}. \quad (23)$$

4) **Contraction:** If $\eta^r \leq \eta_k^q$, then perform an contraction

$$u^{ic} = u^c + \delta^{ic}(u^c - u_k^q) \quad (24)$$

and evaluate $\eta^{ic} = \eta(u^{ic})$, where δ^{ic} is the contraction parameter, u^{ic} is the contraction point. If $\eta^{ic} > \eta_k^q$, replace u_k^q by the u^{ic} , and terminate the iteration:

$$U_{k+1} = \{u_k^0, u_k^1, \dots, u_k^{q-1}, u^{ic}\}; \quad (25)$$

Otherwise, perform a modified adaptive random search (MARS).

5) **MARS:**

Step 1. Decide to perform global search (GS) or local search (LS) with probability ε_1 and $1-\varepsilon_1$, respectively. Go to Step 2a if GS is selected. Otherwise run Step 2b.

Step 2a. Choose a sampled point u^s , in which the i_{th} component is randomly selected from $\mathcal{U}_i$ with probability ε_2 and is i_{th} component of the u_k^0 with probability $1-\varepsilon_2$, $i = 1, 2, \dots, n$.

Step 2b. Choose a uniformly sampled point u^s on the neighborhood $\psi(u_k^0)$ of the u_k^0 , where

$$\psi(u_k^0) = \{u \in \mathcal{U} : \|u - u_k^0\| \leq \min \{\|u_k^0 - u_k^l\|\}, l \neq 0\}. \quad (26)$$

Step 3. Evaluate $\eta^s = \eta(u^s)$.

Step 4. If $\eta^s \geq \eta_k^q$, or $u^s = u_k^0$ and $\eta^s < \eta_k^q$, replace u_k^q by the u^s and terminate the iteration:

$$U_{k+1} = \{u_k^0, u_k^1, \dots, u_k^{q-1}, u^s\}; \quad (27)$$

Otherwise, return to Step 1.

The $u \in \mathcal{U}$ in (13) is a bound constraint. Hence, the $\prod_{\mathcal{U}}(u)$ can be expressed componentwise as

$$\left[\prod_{\mathcal{U}}(u)\right]_i = \begin{cases} u_{i,min}, & \text{if } u_i \leq u_{i,min}, \\ u_i, & \text{if } u_{i,min} < u_i < u_{i,max}, \\ u_{i,max}, & \text{if } u_{i,max} \leq u_i. \end{cases} \quad (28)$$

In each iteration of the SPS algorithm, the order operation is firstly performed. The best and the worst vertices are found, which have highest and lowest power efficiency among the vertices in U_k , respectively. Next, the reflection operation will be conducted in admissible set $\mathcal{U}$ and a reflection point related with the worst vertex will be calculated through the centroid. According to the power generation efficiency of wind farm at the reflection point, the algorithm will do some operations, e.g. expansion, or contraction, or MARS to form a new simplex U_{k+1} . It can be observed that in each iteration, the worst vertex u_k^q will be replaced by the new vertex that has better or same performance. In SPS algorithm, the δ^{ro} and δ^r define how far a reflected point should be from the centroid. The δ^e decides how far to expand when the direction from the centroid to the reflected point is right. The δ^{ic} defines how far a contracted point should be from the centroid along the direction from the centroid to the worst point when the performance of the reflected point is poorer than the worst point. The MARS makes a balance between GS and LS with probability ε_1 . The GS explores the action space with probability ε_2 and learned knowledge u_k^0 . The LS finishes local exploration by exploiting u_k^0 . The flowchart of the SPS algorithm is shown in Fig. 5.

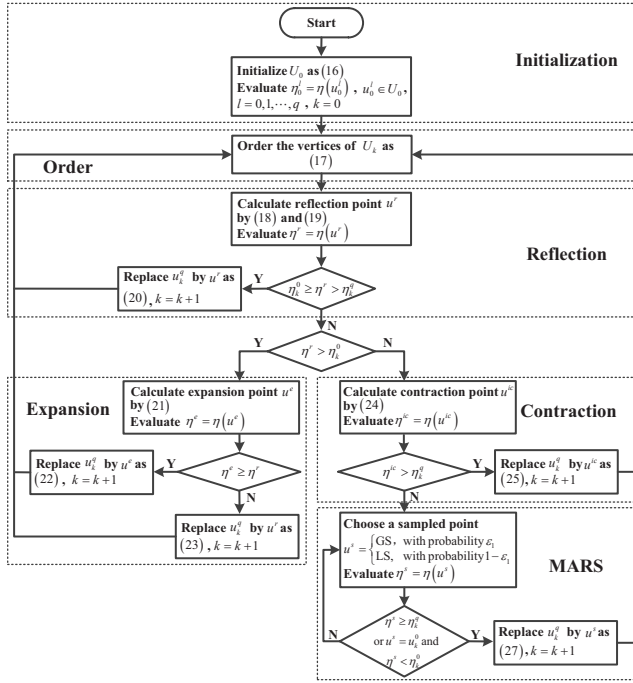


Fig. 5. Flowchart of the SPS algorithm. The power efficiency of each point will be evaluated by sending the point to wind farm and receiving corresponding power output and wind condition data.

Theorem 1. Consider the application of the SPS algorithm to any power optimization sub-problems of wind farm:

(1) All the $q + 1$ power efficiency sequences $\{\eta_k^l\}$, $l = 0, \dots, q$ are increasing and convergent, and their limits satisfy

$$\eta_*^0 \geq \dots \geq \eta_*^q.$$

(2) The optimal solution of the sub-problem can be found with probability 1 for any $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$. Meanwhile,

$$\eta_*^0 = \max_{u \in \mathcal{U}} \eta.$$

Proof. See the Appendix.

In addition to the monotone convergence and global optimality in Theorem 1, the proposed algorithm has following appealing properties:

- **Fast convergence:** The SPS algorithm has fast convergence speed. It is developed based on the Nelder-Mead method (NM) [20], which is one of the most popular derivative free nonlinear optimization methods due to its simplicity and fast convergence [21]. This property will be beneficial to improve quickly the power output of wind farm and reduce the fatigue damage of wind turbines.
- **Constraints handling:** The SPS algorithm complies with the linear constraints. It benefits from Gradient Projection (GP) method [22], projecting the iteration points violating constraints to the boundary of feasible region $\mathcal{U}$ directly such that the iteration points satisfy the control constraints of all turbines.
- **Small performance fluctuation:** The Adaptive Random Search (ARS) is developed in [23] to guarantee NM method find a global optimal solution. Its global search has a big probability to sample an action that has greatly poor performance due to random exploration in the whole

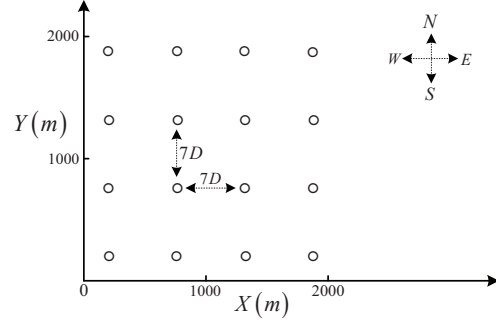


Fig. 6. Layout of 16-turbine wind farm.

action space $\mathcal{U}$. This is also very likely for the local search of the ARS since the search may be finished in the neighborhood of u_k^n (the worst vertex in U_k). In this paper, the ARS has been modified as the MARS. The global search and local search in MARS have performed exploration by fully exploiting u_k^0 (the best vertex in U_k), degrading the probability to select terrible action and guaranteeing global search capability. As a result, the MARS cannot cause a number of large amplitude oscillations of wind farm power generation performance.

To solve the power optimization problem (8), the m SPS algorithms need to be carried out in parallel by Fig. 3(a), which compose the power optimization scheme of wind farm called as SPS policy.

Remark 4: It is possible that (12) in one sub-interval is slightly violated when different control policies are used. In this case, the proposed SPS algorithm can still find the optimal solution although it may need extra iterations.

V. SIMULATION RESULTS

In this section, two simulation examples are presented to verify the performance of the SPS policy in different wind conditions.

The wind farm with 16 turbines shown in Fig. 6 is considered in this simulation. All turbines are of the same size and have a diameter of $80m$, whose spacing is $560m$. The roughness coefficient is $\kappa = 0.04$. The air density is $\rho = 1.225kg/m^3$. These parameters are the ones of Danish Horns Rev1 offshore wind farm [6]. The upstream wind speed is set as $V_\infty = 8m/s$. A common AIF set $\mathcal{U}_i = \{u_i | 0.1 \leq u_i \leq 0.33\}$ is employed to turbine $i \in N$, which is sufficient to verify the performance of different policies [17]. The power generation model based on Park model is used to describe the wind farm.

The vertices of initial simplex have an effect on the performance of the SPS algorithm. Moreover, this algorithm explores the action space by MARS to find the global optimal solution. These mean that there are differences among the results of any two optimizations (or trials) in given wind conditions due to the stochastic nature of the algorithm and thus only one optimization cannot evaluate the performance of SPS policy accurately. Then for given wind conditions, a number of wind farm power optimizations would be performed independently under SPS policy. The performance of the

TABLE I
THE AVERAGE NUMBER OF INTERACTIONS REQUIRED BY POLICIES IN ALL VISITS TO CONVERGE TO THE 98% OF THE OPTIMAL VALUE

	SED policy	ORSSRS policy	SPS policy
$\theta = 0^\circ$	99	171	18
$\theta = 45^\circ$	71	113	11

policy is finally evaluated by average performance index in these optimizations, namely

$$\eta_t' = (1/L) \sum_{l=1}^L \eta_{l,t}, \quad (29)$$

and

$$\bar{\eta}_t' = \left(\frac{1}{L} \right) \sum_{l=1}^L \left(\frac{1}{t} \sum_{\tau=1}^t \eta_{l,\tau} \right), \quad (30)$$

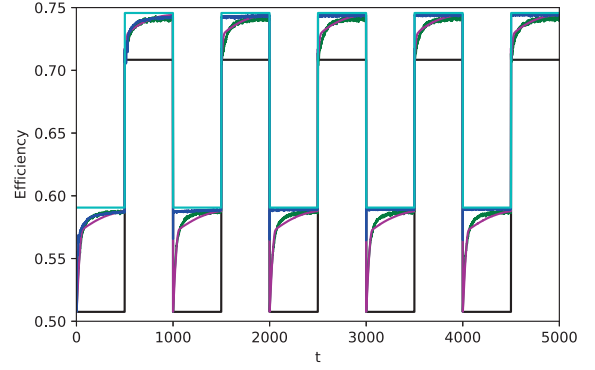
where L is the optimization numbers, l denotes the l_{th} optimization, $\eta_{l,t}$ is the power efficiency of wind farm at the t_{th} interaction of the l_{th} optimization.

The convergence speed of SPS algorithm can be influenced by its parameters. In this paper, the values of q and δ^{ic} are set as 16 and -0.5 respectively following the guideline of the original NM Method [20], ε_2 is chosen as 0.05, the same as [6] for comparison purpose. For the remaining four parameters, the 81 parameter combinations of $\delta^{ro}=1, 2, 3$, $\delta^r=0.1, 0.3, 0.5$, $\delta^e=0.6, 0.8, 1.0$, and $\varepsilon_1 = 0.3, 0.5, 0.7$ are generated based on their feasible regions. To find the best parameter choice, a Monte Carlo simulation is run for each parameter combination. The combination with the highest running average power efficiency $\bar{\eta}_t'$ is finally selected, which is $\delta^{ro}=2$, $\delta^r=0.5$, $\delta^e=0.8$, $\varepsilon_1 = 0.5$.

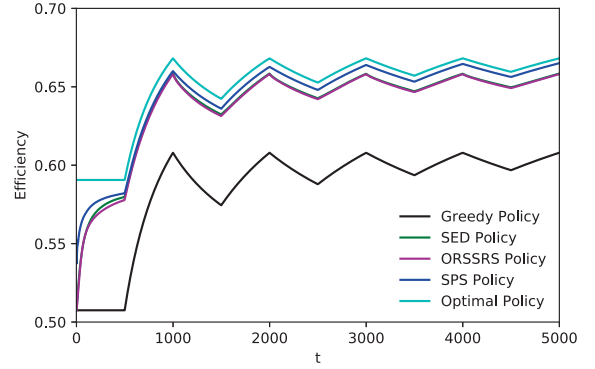
A. Simple Wind Conditions Example

This example assumes that the wind direction changes slowly with an angle set of $\{0^\circ, 45^\circ\}$. At first, wind direction points horizontally from west to east, namely $\theta=0^\circ$. Its first change happens when wind direction turns to point to southeast at time $t = 500$. Then the switch between 45 degree and 0 degree is performed for eight times.

In $\theta = 0^\circ$, the wind farm can be divided into 4 independent rows since there is no wake interactions between the rows. Thus the optimization problem boils down to searching for the optimal solution of a 4-turbine wind farm. Similarly, the wind farm can be divided into 7 independent diagonal arrays in $\theta = 45^\circ$, in which the biggest one has 4 turbines. Then in this case, the optimal policy of wind farm can be obtained by exhaustive search. The greedy policy $u = (1/3, \dots, 1/3)$ is commonly applied in practice. The popular SED policy [6] is usually used as a benchmark in the literatures to assess the performance of the different control policies of wind farm, where the new control variable is the baseline action with high probability or a random action with small probability. The baseline action is updated when new control variable generates better power performance. The Random Search (RS) methods are promising tools for wind farm power optimization [18], which can optimize the power output of the wind farm without using gradient information. The performance of multiple RS methods is investigated in [18] for wind farm power optimization problem. The results show that the Optimized Relative



(a) Trajectories of power efficiency.



(b) Trajectories of running average power efficiency.

Fig. 7. Simulation results in simple wind conditions.

Step Size Random Search (ORSSRS) policy produces higher total power production compared with other RS methods. For comparison purpose, the power optimizations based on optimal policy, greedy policy, SED policy, and ORSSRS policy are performed. Note that the original versions of SED policy in [6] and ORSSRS policy in [18] cannot work in time-varying wind conditions. To overcome this, these two policies are reinitialized with greedy policy for the change of wind conditions.

For given wind conditions, a total of 50 power optimizations using SPS policy are conducted independently. The same number of optimizations for SED policy and ORSSRS policy are run due to their stochastic nature. Fig. 7 shows the simulation results, including the trajectories of wind farm power efficiency η_t' in Fig. 7(a) and the trajectories of wind farm running average power efficiency $\bar{\eta}_t'$ in Fig. 7(b).

In Fig. 7(a), the power efficiency of wind farm with SPS policy at $\theta=0^\circ$ and $\theta=45^\circ$ nearly reaches the optimums, improving approximately 16% and 5% compared with greedy policy, respectively. Meanwhile, it can be noticed that the SPS policy selects control actions based on the learned knowledge when the same wind direction occurs again, which makes it quickly increase the power efficiency of wind farm in time-varying wind conditions. For example, the SPS policy at $t = 1000$ chooses the control action for 0 degree wind direction based on the learned knowledge at $t \in [0, 500)$. At $t = 1500$, it selects the control action for 45 degree wind direction by using learned knowledge at $t \in [500, 1000)$. The SED and ORSSRS policies have to relearn control actions for

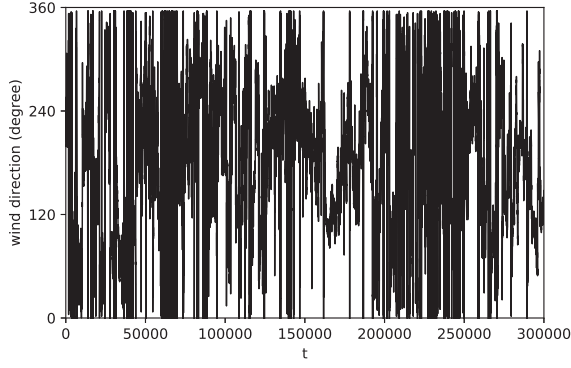


Fig. 8. Complex wind conditions.

the changes of wind conditions due to the forgetfulness of previous knowledge, which results in a much slower response and thus lower power efficiency. From Fig. 7(a), it can be observed that the 0 and 45 degree wind directions are visited for five times, respectively. To quantitatively analyze the average convergence speed of the policies, Table I gives the average number of interactions required by the policies for the wind directions in all visits to converge to the 98% of the optimal values. As can be seen, the proposed SPS policy converges much faster. Fig. 7(b) shows that the SPS policy guarantees wind farm higher running average power efficiency than greedy policy, SED policy, and ORSSRS policy, which means that more power production is generated by SPS policy. Therefore, the proposed policy can efficiently improve the power output of wind farm and increase its energy capture in simple wind conditions.

B. Complex Wind Conditions Example

In this example, the used wind conditions are shown in Fig. 8, which are a time scale of about 3.3 minutes. They are generated based on 10-minute statistics from Anholt offshore wind farm, which are published by Ørsted and can be accessible in [24]. Note that each data in [24] is copied twice to obtain the 3.3-minute wind data. The wind direction ranges from 0 degree to 360 degree. The greedy policy is applied to the wind farm in Fig. 6 and the corresponding power data can be obtained in $\theta \in [0^\circ, 360^\circ]$, which are assumed as the historical power generation data from actual wind farm including the wake interactions among the turbines. Then the power efficiencies of the wind farm can be calculated by using (11). The constant ς is set as 0.02. According to (12), the whole wind direction range is divided into the 144 sub-intervals. Without loss of generality, the divided result of $\theta \in [0^\circ, 45^\circ]$ is shown in Fig. 9. Therefore, the power optimization problem of the wind farm can be divided into 144 sub-problems and the 144 SPS algorithms would be conducted to solve the sub-problems, comprising SPS policy.

With the growth of turbine numbers in a wind farm, the search space and time of optimal solution increase exponentially. For example, the search spaces of optimal solutions for the 3-turbines, 4-turbines, and 5-turbines wind farms are as large as 24^3 , 24^4 , and 24^5 , respectively, when the discretized action set of the form $[0.1 : 0.01 : 0.33]$ is selected for each turbine. The search time of optimal solutions for

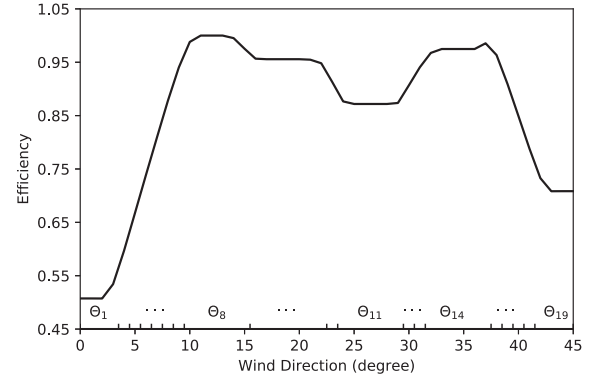


Fig. 9. Generation efficiencies of the wind farm under greedy policy and the sub-interval number of wind direction.

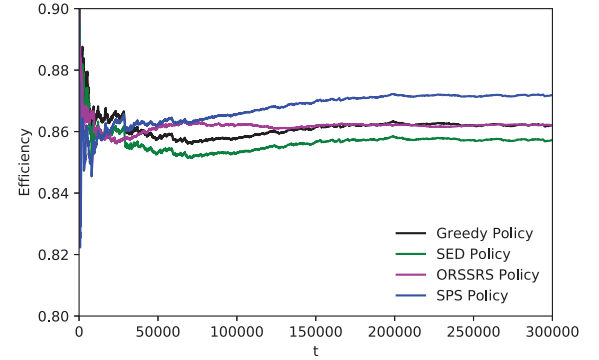


Fig. 10. Trajectories of running average power efficiency of the wind farm in complex wind conditions.

the above three wind farms by exhaustive search is about 1s, 41s, and 1320s with a PC of Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 32GB RAM, and NVIDIA GeForce RTX 2070. This means that the significant computational resources are required when the turbine numbers are more than 5. On the other hand, it would be difficult to reduce the dimension of wind farm power optimization problem except a few wind directions due to complex wake interactions among the turbines. For example, when $\theta = 6^\circ$, the dimension of the power optimization problem of the wind farm shown in Fig. 6 is as high as 16. As a result, the optimal policy of the wind farm is computationally prohibitive in complex wind conditions. Similarly, a total of 50 optimizations are respectively finished for SPS, SED and ORSSRS policies. Fig. 10 gives the trajectories of running average power efficiency of the wind farm.

From Fig. 10, the SPS policy shows superior performance than other three policies except in the early stages due to the evaluations for initial simplex vertices in all sub-problems. The ORSSRS policy and greedy policy show similar performance. Although the running average power efficiency of the wind farm with SPS policy is lower at the initial phase, its average improvement rates are respectively 1.1%, 1.6%, and 1.1% at $t \in (150000, 300000]$ compared with greedy policy, SED policy, and ORSSRS policy. It can be concluded that the proposed SPS policy can adapt to complex wind conditions and efficiently improve the power generation performance of the wind farm.

VI. CONCLUSION

In this paper, a hierarchical data-driven power optimization scheme is proposed for wind farm to mitigate the effect of wake interactions which can significantly reduce the power output of the wind farm and are challenging to model due to the complexities of wake. With this scheme, a number of SPS algorithms are carried out in parallel, which makes full use of learned knowledge and improves the convergence ability of the scheme to complex wind conditions. The SPS algorithm enjoys the advantages of NM method, GP method and MARS, needing no wake interaction model, dealing with the bound constraints of the control variables of all turbines and having ability to find globally optimal solution. Simulation tests are performed in different wind conditions to demonstrate the power generation performance of the proposed scheme. The results show that the presented scheme efficiently improves the power output of wind farm. Note that the simulation is conducted on a simplified wind farm model for illustration purpose. Future research includes considering the influence of measurement noise on the SPS algorithm performance and performing simulation test on more realistic wind farm model and experimental test on actual wind farm. In addition, the centralized control may not be suitable for very large-scale wind farm optimization due to communication, scalability and reliability issues, et al. Therefore, the distributed power optimization of large-scale wind farm in complex wind conditions is another key area of our future research.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. The real wind data from Ørsted are also greatly appreciated.

APPENDIX PROOF OF THEOREM 1

Obviously, the $\eta_k^l \leq \eta_{k+1}^l, k = 0, 1, 2, \dots$ for $l = 0, 1, \dots, q$. It means that the $\{\eta_k^l\}$ is the monotonically increasing sequence. Meanwhile, the $\eta_k^l \leq 1$ holds for $k = 0, 1, 2, \dots$ due to the wake interactions among the turbines. Therefore according to the monotone bounded theorem, the $\{\eta_k^l\}$ is convergent, $l = 0, 1, \dots, q$. Denote η_*^l as the limit of $\{\eta_k^l\}$, $l = 0, 1, \dots, q$. Since $\eta_k^0 \geq \eta_k^1 \geq \dots \geq \eta_k^q$ holds for $k = 0, 1, 2, \dots$, $\eta_*^0 \geq \dots \geq \eta_*^q$.

Without loss of generality, it is assumed that all the previous steps fail and the MARS is used to find the improved solution. The event that the global search is decided and the sampled point u^s constitutes an optimal solution of the sub-problem occurs with at least probability

$$\varepsilon_1 \left(\frac{\varepsilon_2}{|\mathcal{U}_1|} \right) \left(\frac{\varepsilon_2}{|\mathcal{U}_2|} \right) \dots \left(\frac{\varepsilon_2}{|\mathcal{U}_n|} \right),$$

where $|\mathcal{U}_i|$ denotes the cardinality of the action set of turbine i . Therefore, an optimal solution will eventually be selected with probability 1 for any $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$. Once the optimal solution of the sub-problem is selected in the K_{th} iteration, the u_{K+1}^0 is the optimal solution and $\eta_{K+1}^0 = \max_{u \in \mathcal{U}} \eta$. $\eta_k^0 = \eta_{K+1}^0$

at $k > K + 1$. It means that $\eta_k^0 \rightarrow \eta_{K+1}^0$ as $k \rightarrow +\infty$, namely $\eta_*^0 = \max_{u \in \mathcal{U}} \eta$.

REFERENCES

- [1] Q. Cao, Y. D. Song, J. M. Guerrero, and S. Tian, "Coordinated control for flywheel energy storage matrix systems for wind farm based on charging/discharging ratio consensus algorithms," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1259–1267, May. 2016.
- [2] T. Ding, R. Bo, H. Sun, F. Li, and Q. Guo, "A robust two-level coordinated static voltage security region for centrally integrated wind farms," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 460–470, Jan. 2016.
- [3] J. Park, K. H. Law, "Bayesian Ascent: A data-driven optimization scheme for real-time control with application to wind farm power maximization," *IEEE Trans. Control. Syst. Technol.*, vol. 24, no. 5, pp. 1655–1668, Sep. 2016.
- [4] P. M. O. Gebraad *et al.*, "Wind plant power optimization through yaw control using a parametric model for wake effects—a CFD simulation study," *Wind Energy*, vol. 19, pp. 95–114, Dec. 2014.
- [5] S. Boersma *et al.*, "A tutorial on control-oriented modeling and control of wind farms," in *Proc. Amer. Control Conf.*, Seattle, USA, May. 2017, pp. 1–18.
- [6] J. R. Marden, S. D. Ruben, L. Y. Pao, "A model-free approach to wind farm control using game theoretic method," *IEEE Trans. Control. Syst. Technol.*, vol. 21, no. 4, pp. 1207–1214, Jul. 2013.
- [7] M. S. Adaramola, P. A. Krogstad, "Experimental investigation of wake effects on wind turbine performance," *Renew. Energy*, vol. 36, no. 8, pp. 2078–2086, Feb. 2011.
- [8] T. Ahmad *et al.*, "Field implementation and trial of coordinated control of wind farms," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1169 – 1176, Jul. 2018.
- [9] J. A. Dahlberg, "Assessment of the Lillgrund windfarm: power performance, wake effects," Vatenfall Vindkraft AB, 6_1 LG Pilot Report, Sep. 2009.
- [10] F. Heer, P. M. Esfahani, M. Kamgarpour, and J. Lygeros, "Model based power optimisation of wind farms," in *Proc. Eur. Control Conf.*, Strasbourg, France, Jun. 2014, pp. 1145–1150.
- [11] J. Park, S. Kwon, K. H. Law, "Wind farm power maximization based on a cooperative static game approach," in *Proc. SPIE.*, vol. 8688, no. 3, pp. 108–111, Apr. 2013.
- [12] J. Park, K. H. Law, "A data-driven, cooperative wind farm control to maximize the total power production," *Applied Energy*, vol. 165, pp. 151–165, Mar. 2016.
- [13] J. P. Goit, W. Munters, J. Meyers, "Optimal coordinated control of power extraction in LES of a wind farm with entrance effects," *Energies*, vol. 9, no. 3, pp. 2–20, Jan. 2016.
- [14] Y. T. Wu, F. Porté-Agel, "Modeling turbine wakes and power losses within a wind farm using LES: An application to the Horns Rev offshore wind farm," *Renew. Energy*, vol. 75, pp. 945–955, Mar. 2015.
- [15] A. C. Kheirabadi, R. Nagamune, "A quantitative review of wind farm control with the objective of wind farm power maximization," *J. Wind Eng. Ind. Aerodynam.*, vol. 192, pp. 45–73, Sep. 2019.
- [16] J. M. Xu, Y. C. Soh, "A distributed simultaneous perturbation approach for large-scale dynamic optimization problems," *Automatica*, vol. 72, pp. 194–204, Jul. 2016.
- [17] S. Zhong, X. Wang, "Decentralized model-free wind farm control via discrete adaptive filtering methods," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 2529–2540, Jul. 2018.
- [18] M. R. Hao *et al.*, "Performance evaluation of random search based methods on model-free wind farm control," *Intelligent Manufacturing and Mechatronics*, Springer, Singapore, Apr. 2018, pp. 657–670.
- [19] I. Katic, J. Hojstrup, N. Jensen, "A simple model for cluster efficiency," in *Proc. Eur. Wind Energy Conf.*, Rome, Italy, 1986, pp. 407–410.
- [20] J. A. Nelder, R. Mead, "A Simplex Method for Function Minimization," *Computer J.*, vol. 7, no. 4, pp. 308–313, 1965.
- [21] N. Pham, A. Malinowski, T. Bartczak, "Comparative Study of Derivative Free Optimization Algorithms," *IEEE Trans. Ind. Inform.*, vol. 7, no. 4, pp. 592–600, Nov. 2011.
- [22] A. A. Goldstein, "Convex programming in Hilbert space," *B. Amer. Math. Soc.*, vol. 70, no. 5, pp. 709–710, May. 1964.
- [23] K. H. Chang, "Stochastic Nelder–Mead simplex method—A new globally convergent direct search method for simulation optimization," *Eur. J. Oper. Res.*, vol. 220, no. 3, pp. 684–694, Aug. 2012.
- [24] Ørsted, "Offshore wind data," Accessed on: Oct 31, 2020, [Online]. Available: <https://orsted.com/en/our-business/offshore-wind/wind-data>.