Efficient Distributed Optimization of Wind Farms Using Proximal Primal-Dual Algorithms

Preprint

Jennifer Annoni,[1] Emiliano Dall'Anese,[2] Mingyi Hong,[3] and Christopher J. Bay[1]

1 National Renewable Energy Laboratory
2 University of Colorado
3 University of Minnesota

Presented at IEEE American Control Conference
Philadelphia, Pennsylvania
July 10–12, 2019

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

Conference Paper
NREL/CP-5000-73395
July 2019

Contract No. DE-AC36-08GO28308

Efficient Distributed Optimization of Wind Farms Using Proximal Primal-Dual Algorithms

Preprint

Jennifer Annoni,[1] Emiliano Dall'Anese,[2] Mingyi Hong,[3] and Christopher J. Bay[1]

1 National Renewable Energy Laboratory
2 University of Colorado
3 University of Minnesota

Suggested Citation
Annoni, Jennifer, Emiliano Dall'Anese, Mingyi Hong, Christopher J. Bay. 2019. *Efficient Distributed Optimization of Wind Farms Using Proximal Primal-Dual Algorithms: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5000-73395. https://www.nrel.gov/docs/fy19osti/73395.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5000-73395
July 2019

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Efficient Distributed Optimization of Wind Farms Using Proximal Primal-Dual Algorithms

Jennifer Annoni, Emiliano Dall'Anese, Mingyi Hong, and Christopher J. Bay

***Abstract*—This paper presents a distributed approach to performing real-time optimization of large wind farms. Wind turbines in a wind farm typically operate individually to maximize their own performance regardless of the impact of aerodynamic interactions on neighboring turbines. This paper optimizes the overall power produced by a wind farm by formulating and solving a nonconvex optimization problem where the yaw angles are optimized to allow some turbines to operate in misaligned conditions and shape the aerodynamic interactions in a favorable way. The solution of the nonconvex smooth problem is tackled using a proximal primal-dual gradient method, which provably identifies a first-order stationary solution in a global sublinear manner. By adding auxiliary optimization variables for every pair of turbines that are coupled aerodynamically, and properly adding consensus constraints into the underlying problem, a distributed algorithm with turbine-to-turbine message passing is obtained; this allows for turbines to be optimized in parallel using local information rather than information from the whole wind farm. This algorithm is computationally light, as it involves closed-form updates. This approach is demonstrated on a large wind farm with 60 turbines. The results indicate that similar performance can be achieved as with finite-difference gradient-based optimization at a fraction of the computational time and thus approaching real-time control/optimization.**

I. Introduction

Wind farm control can be used to achieve a number of objectives including increasing power production in a wind farm, improving the lifetime of turbines in a wind farm, and tracking power reference signals to improve wind integration into the energy grid. This paper focuses on increasing the power production of a wind farm by operating some wind turbines sub-optimally to improve the performance of the entire wind farm [1]. To increase power in a wind farm, one common wind plant control strategy in literature is known as wake redirection or wake steering [2]. Wake redirection typically uses yaw misalignment of the turbines with respect to the incoming wind direction to induce favorable aerodynamic interactions at downstream turbines. Various computational fluid dynamics simulations, wind tunnel experiments, and utility-scale field experiments have shown that this method can increase power without substantially increasing turbine loads [3], [4]. Currently, wind farm control approaches use a look-up table based on offline optimization results [5]. This approach breaks down when individual turbines are unavailable due to maintenance. For small wind farms, an optimization can be performed in real time and adapt to changing atmospheric and turbine conditions. However, as wind farms increase in size, computationally efficient algorithms are needed to perform real-time optimization and control.

A wind farm can be represented as a multi-agent system, with turbines being the agents. As a result, distributed optimization and control can be used in wind farm optimization and controls problems and provides a framework for efficient computation of large systems by coordinating subsystems to interact with their larger environment [6]. Distributed optimization has also been considered in previous wind farm controls literature [7], [8]. The algorithm used in [7] requires a linear model, which can be difficult to keep accurate across changing operating conditions. The two optimization methods presented in [8] offer power reference tracking and load reduction, but don't include a wake model and still require a global problem to be formulated, which becomes increasingly difficult as wind farm size grows. This is a challenging problem due to the complex aerodynamic interactions and large timescales. For example, another distributed optimization framework for wind farm controls has been presented by [9] for load reduction and power reference distribution. Yet, solving this problem becomes computationally complex as the system grows because of the number of turbines and larger flow domains.

Distributed algorithms for convex optimization problems have been developed extensively in the literature (see representative works in [10], [11] and pertinent references therein). However, the nonlinear steady-state model utilized for the wind farm leads to an underlying optimization problem utilized to maximize the output power of the wind farm that is *nonconvex*. Recently, a number of methods have been investigated for multi-agent distributed nonconvex systems [12]–[16]. In this work, the solution of the nonconvex problem is tackled using a proximal primal-dual algorithm (ProxPDA) [15], [16], and we choose ProxPDA due to its simple implementation and practical efficiency. We show in this paper that, even in a centralized setting, the proposed algorithm can provably identify a first-order stationary solution in a global sublinear manner. Further, this paper demonstrates that a wind farm can be modeled as a distributed system by

J. Annoni is with the National Renewable Energy Laboratory; email: jennifer.annoni@nrel.gov.

C. Bay is with the National Renewable Energy Laboratory; email: christopher.bay@nrel.gov.

E. Dall'Anese is with the University of Colorado Boulder; email: emiliano.dallanese@colorado.edu.

M. Hong is with the University of Minnesota; email: mhong@umn.edu.

This work was supported in part by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory. Funding for the work was provided in part by the DOE Office of Energy Efficiency and Renewable Energy, Wind Energy Technologies Office.

considering only turbines upstream of a specified turbine. By introducing pertinent optimization variables and reformulating the problem into a consensus-based version where turbines that are coupled via wakes agree on the yaw angles, this paper develops a low-complexity distributed algorithm that involves turbine-to-turbine message passing. The distributed optimization framework is tested via simulation of the Princess Amalia offshore wind farm consisting of 60 turbines, described in Section IV, and presented in previous studies [17]. The results show a significant reduction in computation time without sacrificing the overall power gain of the wind farm when comparing finite-difference gradient-based techniques, shown in Section IV-B. Finally, we conclude by discussing the implications of increased computational efficiency and propose future work in Section V.

II. Wind Farm Modeling and Control

This section briefly describes the wind turbine wake model used to model wake steering in a wind farm as well as formulates the centralized wind farm control problem.

A. Wind Turbine Wake Model

When turbines extract energy from the wind, a wake, or area of velocity deficit, forms behind the turbine. The wind turbine wake model used to characterize this velocity deficit behind a turbine in a wind farm was introduced by several recent papers including [18], [19]. In particular, it uses a Gaussian profile to model the velocity deficit behind a turbine:

$$\frac{u(x,y,z)}{U_\infty} = 1 - Ce^{-(y-\delta)^2/2\sigma_y^2}e^{-(z-z_h)^2/2\sigma_z^2} \tag{1}$$

where u is the velocity in the wake, U_∞ is the free-stream velocity, x is the streamwise direction, y is the spanwise direction, δ is the wake centerline, z is the vertical direction, z_h is the hub height, σ_y is the wake expansion in the z direction, and C is the velocity deficit at the wake center. These parameters are defined in [18].

In addition to the velocity deficit, a wake deflection model is used to describe the turbine behavior in yaw misaligned conditions, which occur when performing wake steering, and is also implemented based on [2], [18]. The wake deflection due to yaw misalignment is defined as:

$$\alpha \approx \frac{0.3\gamma}{\cos\gamma}\left(1 - \sqrt{\cos\gamma}\right) \tag{2}$$

where γ is the yaw angle of the turbine and C_T is the thrust coefficient determined by turbine operating parameters, such as blade pitch and generator torque. The initial wake deflection, δ_0, is then defined as:

$$\delta_0 = x_0 \tan\alpha \tag{3}$$

where x_0 indicates the length of the near wake, which is typically on the order of 3 rotor diameters. A full description of the wake deflection can be found in [18]

Lastly, the turbine model used in the wind turbine wake model consists of a power coefficient, C_P, and thrust coefficient, C_T, based on wind speed and constant blade pitch angle.

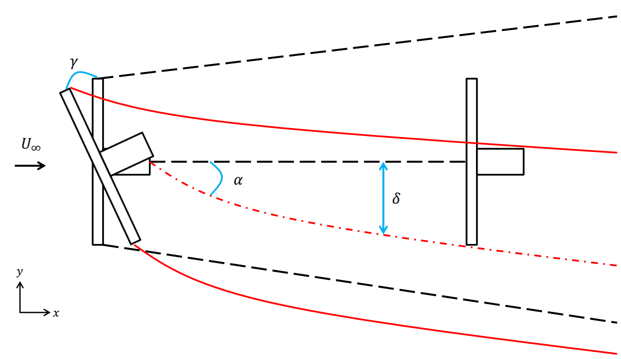

Fig. 1. Two-turbine example of wake steering control, where γ denotes the yaw angle of the upstream turbine, α denotes the deflection angle, and δ denotes the wake deflection. The black dashed lines represent the wake of the upstream turbine under non-yawed conditions and the red lines denote the wake of the upstream turbine under yawed conditions.

The coupling between C_P and C_T is critical in understanding the benefits of wind farm controls. In other words, each turbine is free to operate at its own C_P and C_T based on local conditions. In this study, the C_P and C_T curves were computed using FAST [20] and the National Renewable Energy Laboratory's (NREL's) 5 MW turbine [21].

Given these parameters, the steady-state power of each turbine under yaw misalignment conditions is given by [22]:

$$P(\gamma; u) = \frac{1}{2}\rho A C_P \left(\cos\gamma\right)^p u^3 \tag{4}$$

where ρ is the air density, A is the rotor area, $\cos\gamma^p$ is a correction factor added to account for the effects of yaw misalignment, and p is a tuneable parameter that matches the power loss caused by the yaw misalignment seen in simulations.

B. Wind Farm Control

Wake steering control uses the yaw drive of a turbine to deflect a turbine's wake away from the downstream turbine. This section describes the centralized yaw optimization problem for a two-turbine array, shown in Fig. 1. In practice, this can be extended to many turbines in a wind farm.

P_1 and P_2 denote the power from the upstream turbine and downstream turbine, respectively. The power generated by the upstream turbine depends on the local inflow wind speed, u_1, and its yaw angle, γ_1. The power generated can be expressed using (4). Therefore, the power generated by the upstream turbine can be expressed as a function of the inflow velocity and the yaw angle, $P_1(\gamma_1, u_1)$, where $u_1 = U_\infty$, i.e., freestream velocity. Because the yaw angle of the upstream turbine can be used to steer the wake into or away from the downstream turbine, the power of the second turbine is now a function of the yaw angle of the upstream turbine, γ_1. The power generated by the downstream turbine is now expressed as $P_2(\gamma_1, \gamma_2, u_2)$, where u_2 is the disturbed local incoming velocity to the downstream turbine, i.e., (1)-(4). The total power generated by the two-turbine array is given by:

$$P_{tot}(\gamma; u) = P_1(\gamma_1; u_1) + P_2(\gamma_2; u_2(\gamma_1)) \tag{5}$$

where $\gamma := [\gamma_1, \quad \gamma_2]^T$ and $u_2(\gamma_1)$ stresses the dependency of u_2 from the yaw angle of turbine 1. If multiple turbines are upstream of turbine i, the wind speeds at the downstream turbine are combined using sum-of-squares. In the following, for notational simplicity, we will drop u from the arguments of the functions modeling the power output of a turbine.

III. System-level Optimization Problem

A. Wind Farm as a Graph

To generalize the model for the wind farm operating under wake effects and facilitate distributed optimization techniques, consider modeling the wind farm as a graph $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N} := \{1, \ldots, N\}$ is the set of wind turbines and $\mathcal{E}$ is a set of directed edges; in particular, edge $(i, j) \in \mathcal{E}$ if the wind turbine i is physically coupled with the upstream wind turbine j via a wake. Let $\mathcal{W}_i \subseteq \mathcal{N} \backslash \{i\}$ be the set of upstream turbines that are coupled with the ith one via wakes. For example, in the illustrative 4-turbine system in Fig. 2(a), turbines 3 and 4 are impacted by upstream turbines, i.e., $\mathcal{W}_3 = \{1, 2\}$ and $\mathcal{W}_4 = \{2\}$, whereas the turbines 1 and 2 are not interfered by any upstream turbines. On the other hand, let $\bar{\mathcal{W}}_i := \{j | i \in \mathcal{W}_j\}$ be the set of downstream wind turbines that the turbine i interferes. For the system in Fig. 2(a), $\bar{\mathcal{W}}_1 = \{3\}$, $\bar{\mathcal{W}}_2 = \{3, 4\}$, and $\bar{\mathcal{W}}_3 = \bar{\mathcal{W}}_4 = \emptyset$.

Let $P_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i}) : \mathbb{R}^{1+|\mathcal{W}_i|} \to \mathbb{R}$ represent the power produced by a turbine i, as a function of the yaw angle γ_i and the yaw angles if the upstream turbines $j \in \mathcal{W}_i$ that might be coupled with the ith one through wakes. The function $P_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i})$ is, in general, continuous, smooth, and nonconvex for a number of existing wake models (see Section II-B). It is also assumed that $P_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i})$ has a Lipschitz-continuous gradient. For a given wind direction and speed, and based on a given wake model, the problem of maximizing the overall power output of a wind farms can therefore be stated as the following nonconvex program:

$$\min_{\{\gamma_i \in \mathbb{R}\}_{i \in \mathcal{N}}} \sum_{i \in \mathcal{N}} f_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i}) \tag{6}$$

where $f_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i}) := -P_i(\gamma_i, \{\gamma_j\}_{i \in \mathcal{W}_i}) + h_i(\gamma_i)$, with the convex function $h_i : \mathbb{R} \to \mathbb{R}$ capturing possible mechanical or electric stress associated with the yawing, or deviations from a predefined set point. The function h_i is assumed to be continuously differentiable with Lipschitz-continuous gradient.

B. Distributed Algorithmic Solution

Based on the physical coupling through wakes modeled by the graph $(\mathcal{N}, \mathcal{E})$, consider adding the auxiliary optimization variables $\{\gamma_{i,j}\}_{j \in \mathcal{W}_i}$ for each waked turbine i. The auxiliary variable $\gamma_{i,j}$ represents a *copy* of the yaw angle γ_j of turbine $j \in \mathcal{W}_i$ that is stored locally at turbine i. Upon defining the $1 + |\mathcal{W}_i| \times 1$ optimization variable $x_i := [\gamma_i, \{\gamma_{i,j}\}_{j \in \mathcal{W}_i}]^\mathsf{T}$ for each turbine $i \in \mathcal{N}$, problem (6) can be equivalently re-expressed as:

$$\min_{\{x_i \in \mathbb{R}^{1+|\mathcal{W}_i|}\}_{i \in \mathcal{N}}} \sum_{i \in \mathcal{N}} f_i(x_i) \tag{7a}$$

$$\text{subject to: } \gamma_{i,j} = \gamma_j, \ \forall j \in \mathcal{W}_i, i \in \mathcal{N} \tag{7b}$$

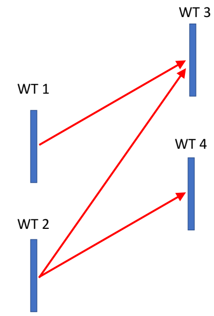

Fig. 2. Example of network of wind turbines. (a) wind farm graph, where nodes represent wind turbines and directed edges represent coupling via wakes. (b) Network of coupled variables, where nodes represent optimization variables and undirected edges represent consensus constraints.

where the $M := |\mathcal{E}|$ consensus constraint (7b) ensures that two turbines coupled through wakes agree on the yaw angle of the upstream turbine. For example, in the illustrative 4-turbine system in Fig. 2, the three consensus constraints are $\gamma_1 = \gamma_{3,1}$, $\gamma_2 = \gamma_{3,2}$, and $\gamma_2 = \gamma_{4,2}$. Notice that the total number of variables in (7) is $N + \sum_{i=1}^{N} |\mathcal{W}_i| = N + M$.

To enable the development of a distributed algorithmic solution, define the vector $x = [x_1^\mathsf{T}, \ldots, x_N^\mathsf{T}]^\mathsf{T}$, and consider constructing a "consensus" graph where:

(i) The set of nodes corresponds to the the optimization variables x

(ii) M directed edges represent the coupling among variables specified by the consensus constraints (7b).

As an example, the consensus graph for the wind farm of Fig. 2(a) is shown in Fig. 2(b); in this case, the consensus graph has four connected subgraphs.

Let $A \in \mathbb{R}^{M \times M+N}$ be the edge-node incidence matrix of the consensus graph; for the graph in Fig. 2(b), one has that:

$$A = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Using this notation, problem (7) can be rewritten in compact form as:

$$\min_{\{x_i\}_{i=1}^N} \sum_{i \in \mathcal{N}} f_i(x_i) \tag{9a}$$

$$\text{subject to } Ax = 0\,. \tag{9b}$$

The proposed algorithm hinges on the so-called augmented Lagrangian function, which is defined as:

$$L(x, \lambda) := \sum_{i \in \mathcal{N}} f_i(x_i) + \lambda^\mathsf{T} Ax + \frac{\beta}{2}\|Ax\|_2^2 \tag{10}$$

where $\lambda \in \mathbb{R}_+^M$ is the vector of dual variables associated with constraint (9b) and $\beta > 0$ is a user-defined tuning parameter. To outline the ProxPDA algorithm [15], [16], consider the following additional quantities:

- $B := |A|$, where the absolute value is taken entry-wise
- d_i: degree of node i in the consensus graph, and $D := \mathsf{diag}([d_1, \ldots, d_7])$. For example, in the graph in Fig. 2, one has $D = \mathsf{diag}([1, 2, 0, 1, 1, 0, 1])$

- $L^- := A^\mathsf{T} A$
- $L^+ := 2D - A^\mathsf{T} A$
- $G := B^\mathsf{T} B + \epsilon I$, where $\epsilon > 0$.

Notice that $L^+ := 2D - A^\mathsf{T} A = B^\mathsf{T} B$. Further, a suitable choice for ϵ is $\epsilon = 1$.

Then, based on (10), the ProxPDA algorithm involves the sequential execution of the following steps until convergence where k denotes the iteration index:

$$x^{k+1} = \arg \min_{x_i \in \mathbb{R}^{1+|\mathcal{W}_i|}} \sum_{i=1}^{N} (\nabla_{x_i} f_i(x_i^k))^\mathsf{T} (x_i - x_i^{k+1}) + (\lambda^k)^\mathsf{T} A x + \frac{\beta}{2}\|Ax\|_2^2 + \frac{\beta}{2}\|x - x^k\|_G^2 \quad (11a)$$

$$\lambda^{k+1} = \lambda^k + \beta A x^{k+1} \quad (11b)$$

where $\nabla_x f$ denotes the gradient of f with respect of x.

The primal iteration minimizes the augmented Lagrangian plus a proximal term $\frac{\beta}{2}\|x - x^k\|_G^2$; the proximal term plays a key role, as it facilitates the convergence and optimality analysis [15], [23]. In fact, if G is chosen in a way that $A^T A + G^T G$ is full rank, then the objective function of (11a) is strongly convex; at the same time, based on the structure of B, the update (11a) will be shown to be decomposable across turbines.

Leveraging the definitions above, the steps (11) can be further rewritten as:

$$x^{k+1} = \arg \min_{x_i \in \mathbb{R}^{1+|\mathcal{W}_i|}} \sum_{i=1}^{N} (\nabla_{x_i} f_i(x_i^k))^\mathsf{T} (x_i - x_i^{k+1}) + (\lambda^k)^\mathsf{T} A x + \beta x^\mathsf{T} (D + \epsilon I) x - \beta x^\mathsf{T} L^+ x^k \quad (12a)$$

$$\lambda^{k+1} = \lambda^k + \beta A x^{k+1} \quad (12b)$$

where the matrix $D + \epsilon I$ is full rank and the update of the primal variables x^{k+1} affords the following closed-form solution:

$$x^{k+1} = \frac{1}{2\beta}(D + \epsilon I)^{-1} \left(\beta L^+ x^k - \nabla_x f(x^k) - A^\mathsf{T} \lambda^k \right). \quad (13)$$

The resultant algorithm is tabulated as Algorithm 1, and it inherits the convergence results derived in [15], [16], which are adapted to the problem at hand next.

Assumption 1. *The function $f(x) := \sum_{i=1}^N f_i(x_i)$ is differentiable and has Lipschitz continuous gradient; that is, $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$ for all $x, y \in \mathbb{R}^{N+M}$.*

Assumption 2. *There exists a constant $\delta > 0$ such that:*

$$\exists\, g > -\infty, \text{ s.t. } f(x) + \frac{\delta}{2}\|Ax\|_2^2 \geq g\,, \forall x \in \mathbb{R}^{N+M}. \quad (14)$$

Assumption 2 can be readily satisfied by setting $g = 0$ since $f(x) \geq 0$. Assumption 1 leads one to select a wake model so that the resultant function $f(x)$ is strongly smooth. Next, let σ_m be the smallest nonzero eigenvalue of the matrix $A^T A$, and let c be a constant so that

$$c \geq \max\left\{ \frac{\delta}{L}, \frac{4\|G^T G\|_F}{\sigma_m} \right\}. \quad (15)$$

Then, the following result, adapted from [15], holds.

Theorem 1. *Suppose that Assumptions 1-2 hold, and suppose that β is selected such that:*

$$\beta > \frac{L}{2}\left[2c + 1 + \left((2c+1)^2 + \frac{16L^2}{\sigma_m} \right)^{\frac{1}{2}} \right]. \quad (16)$$

Then, every limit point of the iterates $\{x^k, \lambda^k\}$ generated by Algorithm 11 converges to a Karush-Kuh-Tucker (KKT) point of problem 9. Further, consensus is achieved in the sense that:

$$\lim_{k\to\infty} A x^k \to 0. \quad (17)$$

Algorithm 1 can be implemented centrally in a wind farm controller; relative to existing model-based optimization approaches, Algorithm 1 affords a low-complexity implementation and provably converge to a KKT point.

Algorithm 1 Centralized solver

Initialization: Set x^0 based on a prior guess, or the latest yaw angles.
Algorithm: for $k = 0, 1, 2, \cdots$, until $\|x^{k+1} - x^k\|_2 \leq \epsilon$:
[S1] Update x^{k+1} via (13).
[S2] Update λ^{k+1} via (12b).

Algorithm 2 Distributed algorithm

Initialization: Set x^0 based on a prior guess, or the latest yaw angles.
Algorithm: for $k = 0, 1, 2, \cdots$, until $\|x_i^{k+1} - x_i^k\|_2 \leq \epsilon$, perform at each turbine i:
[S1] Update γ_i^{k+1} and $\{\gamma_{i,j}^{k+1}\}_{j \in \mathcal{W}_i}$ via (18) and (19).
[S2] Transmit γ_i^{k+1} to turbines $j \in \bar{\mathcal{W}}_i$ and receive $\gamma_{j,i}^{k+1}$ from turbines $j \in \bar{\mathcal{W}}_i$.
[S3] Transmit $\gamma_{i,n}^{k+1}$ to turbine $n \in \mathcal{W}_i$ and receive γ_n^{k+1} from turbines $n \in \mathcal{W}_i$.
[S4] Update dual variable $\lambda_{i,n}^{k+1}$ and transmit it to turbine $n \in \mathcal{W}_i$.
[S5] Receive $\lambda_{j,i}^{k+1}$ from turbine $j \in \bar{\mathcal{W}}_i$.
Go to [S1].

Notice, however, that the computation of the dual update (12b) and the primal update (13) naturally decompose across turbines. For example, the update of γ_i^{k+1} and $\gamma_{i,j}^{k+1}$ at turbine i boil down to:

$$\gamma_i^{k+1} = \frac{1}{2\beta(d_i + \epsilon)} \left[\beta \left(d_i \gamma_i^k + \sum_{j \in \bar{\mathcal{W}}_i} \gamma_{j,i}^k \right) - \partial_{\gamma_i} f_i(\gamma_i^k) + \sum_{j \in \bar{\mathcal{W}}_i} \lambda_{j,i}^k \right] \quad (18)$$

$$\gamma_{i,j}^{k+1} = \frac{1}{2\beta(1 + \epsilon)} \left[\beta \left(\gamma_{i,j}^k + \gamma_j^k \right) - \partial_{\gamma_{i,j}} f_i(\gamma_{i,j}^k) - \lambda_{i,j}^k \right] \;\; \forall j \in \mathcal{W}_i \quad (19)$$

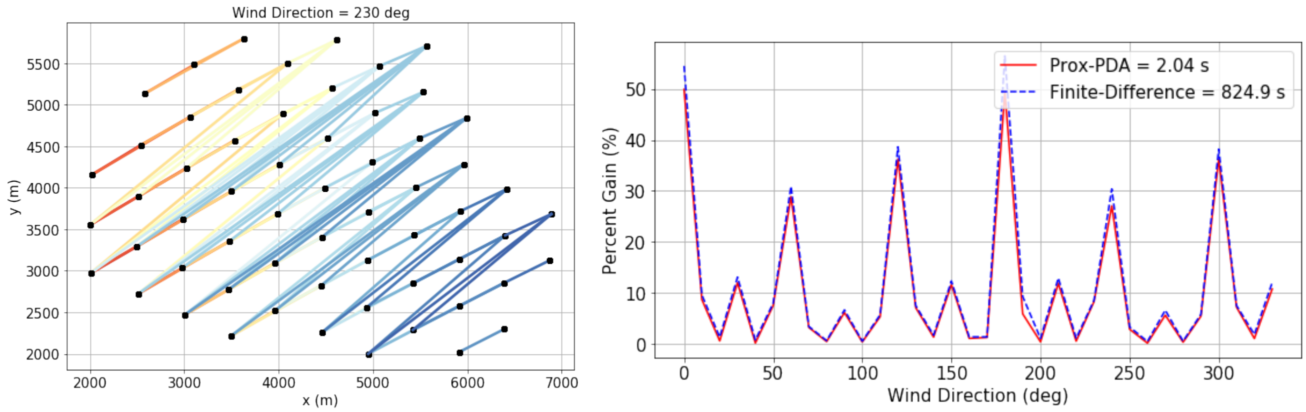

Fig. 3. (Left) The corresponding graph structure for a wind direction of 230° used to solve the optimization with Prox-PDA. (Right) Comparison between finite-difference (blue) and ProxPDA (red). The legends indicate the run-time for each algorithm.

where d_i is the degree of the node associated with γ_i in the network of couples variables [cf. Fig. 2], and the dual variable $\lambda_{i,j}$ corresponds to the constraints $\gamma_{j,i} = \gamma_i$. Assuming that turbine i computes and stores the dual variables $\lambda_{i,j}$, it turns out that turbine i can compute locally γ_i^{k+1} and $\gamma_{i,j}^{k+1}$ upon receiving $\{\gamma_{j,i}^k, \lambda_{j,i}^k\}$ for the (downstream) neighboring turbines $j \in \bar{W}_i$ and γ_n^k from the (upstream) neighboring turbines $n \in W_i$. The resultant distributed algorithm is tabulated as Algorithm 2.

IV. Simulation and Results

To demonstrate the distributed optimization framework described above, we use the Princess Amalia wind farm [17]. This wind farm has 60 turbines that are simulated as the NREL's 5 MW turbine [21] encountering a wind speed of $U_\infty = 8$ m/s with 10% turbulence intensity. We demonstrate the algorithm on 36 wind directions from 0-350 at every 10°. Turbines were not constrained in terms of allowable yaw misalignment. Future implementations will include box constraints on the yaw angles. In these simulations, the function $h_i(\gamma_i)$ is set to 0 and, therefore, the objective is to maximize the overall power output of the wind farm.

A. Graph Structure

The graph structure, A from (9), was defined for each different wind direction by considering all turbines upstream of a turbine and within a spanwise distance of 3 rotor diameters, as described in Section III-A. Alternative graphs can be considered including grouping by nearest neighbors, data-driven approaches, etc. Changes to the graph structure may improve the results of ProxPDA. Finally, it is important to note that the graph changes with changing wind direction, i.e., different turbines communicate based on the wind direction. Currently, data are is collected at a central computer in wind farms. Integrating all the data at each time step is prohibitively expensive. However, integrating data based on the defined graph structure reduces the computational complexity, i.e., each turbine only needs to integrate data from turbines it is connected to. In this paper, it is assumed that the wind direction and speed do not change over the optimization horizon. Future work will consider time-varying graph structures, $A(t)$. An example of the wind direction from 230° is shown in Fig. 3 (left).

B. Results

The power was optimized across the Princess Amalia wind farm using finite-difference gradient-based optimization and the ProxPDA algorithm described in this paper. The optimizations were run every 10° from 0-350°. The results are shown in Fig. 3. This figure indicates the ProxPDA method closely follows the centralized finite-difference method with an average difference of 0.5%. In addition, the ProxPDA results are computed significantly faster than the finite-difference results. Because the wind farm is modeled as a fully distributed system, it allows for computations to be run in parallel, thus reducing the computation time of the optimization significantly such that this can run in real time. The finite-difference solution took, on average, 824.9 s to complete and the ProxPDA solution took, on average, 2.04 s. The flow field for a wind direction of 230° and the corresponding optimized yaw angles are shown in Fig. 4. The finite-difference method computes a 8.5% potential power gain from wake steering and ProxPDA computes a 8.3% power gain.

It is important to note that some of the power gains obtained in Fig, 3 (right) are infeasible in the real world due to yaw constraints, which were not enforced in this analysis. Rather, this paper demonstrates that by modeling the wind farm as a distributed system you can achieve similar performance at a fraction of the cost. This is significant because the wind speed and direction can change on the order of minutes and this wind farm optimization algorithm presented in this paper could be able to accommodate those changes in real time.

V. Conclusions

This paper presents a distributed approach to solving the nonconvex objective function in a wind farm to maximize power. The results were compared with a centralized approach on a 60-turbine wind farm. The results indicate that the distributed approach produces comparable results with a significant speed up in computation time while guaranteeing

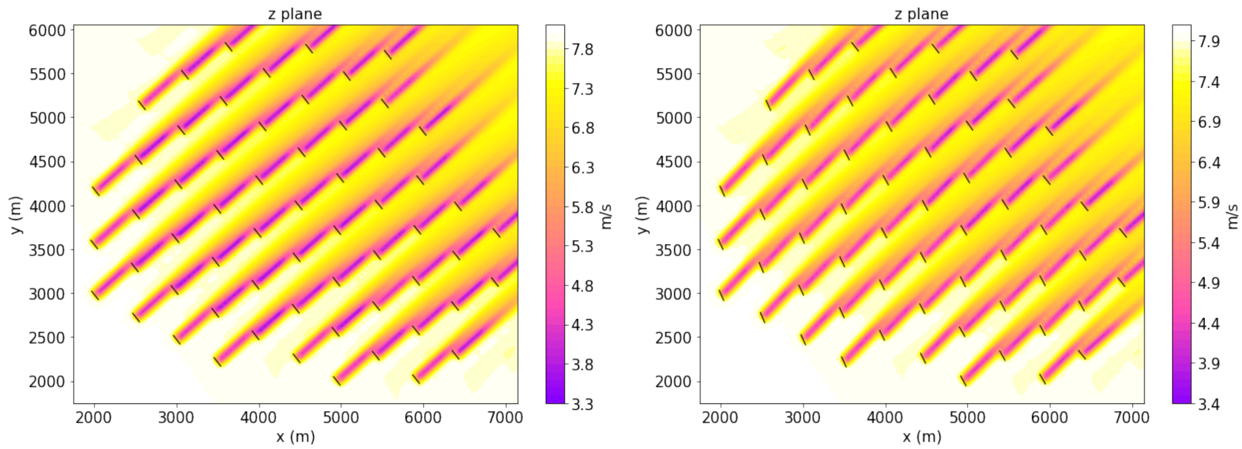

Fig. 4. (Left) Shows the flow field of the wind farm at 230° and (right) the optimized flow field with the optimized yaw angles.

global sublinear convergence to a KKT point. Future work will include alternative definitions of the graph structure, which can have a significant impact on the results. The graph structure can be defined by nearest neighbors, through data-driven techniques, etc. In addition, future work will include the feasibility of the optimization solution given the dynamics of the yaw controller of the turbines in the wind farm. This will allow for more realistic solutions over a finite horizon and move towards providing realistic data on the overall improvement in the wind farm performance.

REFERENCES

[1] S. Boersma, B. Doekemeijer, P. Gebraad, P. Fleming, J. Annoni, A. Scholbrock, J. Frederik, and J. van Wingerden, "A tutorial on control-oriented modeling and control of wind farms," in *American Control Conference*. IEEE, 2017, pp. 1–18.

[2] P. A. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Evaluating techniques for redirecting turbine wakes using SOWFA," *Renewable Energy*, vol. 70, pp. 211–218, 2014.

[3] P. Gebraad, F. Teeuwisse, J. Wingerden, P. A. Fleming, S. Ruben, J. Marden, and L. Pao, "Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study," *Wind Energy*, vol. 19, no. 1, pp. 95–114, 2016.

[4] R. Damiani, S. Dana, J. Annoni, P. Fleming, J. Roadman, J. v. Dam, and K. Dykes, "Assessment of wind turbine component loads under yaw-offset conditions," *Wind Energy Science*, vol. 3, no. 1, pp. 173–189, 2018.

[5] P. Fleming, J. Annoni, J. J. Shah, L. Wang, S. Ananthan, Z. Zhang, K. Hutchings, P. Wang, W. Chen, and L. Chen, "Field test of wake steering at an offshore wind farm," *Wind Energy Science*, vol. 2, no. 1, pp. 229–239, 2017.

[6] S. Ferrari, G. Foderaro, P. Zhu, and T. A. Wettergren, "Distributed optimal control of multiscale dynamical systems: a tutorial," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 102–116, 2016.

[7] C. Bay, J. Annoni, T. Taylor, L. Pao, and K. Johnson, "Active power control for wind farms using distributed model predictive control and nearest neighbor communication," in *American Control Conference (ACC), 2018*. IEEE, 2017, p. Submitted.

[8] V. Spudić, C. Conte, M. Baotić, and M. Morari, "Cooperative distributed model predictive control for wind farms," *Optimal Control Applications and Methods*, vol. 36, no. 3, pp. 333–352, 2015.

[9] M. Soleimanzadeh, R. Wisniewski, and K. Johnson, "A distributed optimization framework for wind farms," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 123, pp. 88–98, 2013.

[10] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, p. 1291, 2011.

[11] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[12] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[13] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[14] M. Zhu and S. Martínez, "An approximate dual subgradient algorithm for multi-agent non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1534–1539, 2013.

[15] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, Aug 2017, pp. 1529–1538.

[16] M. Hong, "Decomposing linearly constrained nonconvex problems by a proximal primal dual approach: Algorithms, convergence, and applications," 2016, [Online] arXiv preprint arXiv:1604.00543.

[17] P. A. Fleming, A. Ning, P. M. Gebraad, and K. Dykes, "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, vol. 19, no. 2, pp. 329–344, 2016.

[18] M. Bastankhah and F. Porté-Agel, "Experimental and theoretical study of wind turbine wakes in yawed conditions," *Journal of Fluid Mechanics*, vol. 806, pp. 506–541, 2016.

[19] A. Niayifar and F. Porté-Agel, "A new analytical model for wind farm power prediction," in *Journal of Physics: Conference Series*, vol. 625, no. 1. IOP Publishing, 2015, p. 012039.

[20] J. Jonkman, "NWTC design codes (FAST)," *NWTC Design Codes (FAST), NREL, Boulder, CO*, 2010.

[21] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-mw reference wind turbine for offshore system development," *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060*, 2009.

[22] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, *Wind energy handbook*. John Wiley & Sons, 2001.

[23] S. Zlobec, "On the liu–floudas convexification of smooth programs," *Journal of Global Optimization*, vol. 32, no. 3, pp. 401–407, Jul 2005.