Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

RESEARCH ARTICLE

Wake Expansion Continuation: Multi-Modality Reduction in the Wind Farm Layout Optimization Problem

Jared J. Thomas* | Spencer McOmber | Andrew Ning

Department of Mechanical Engineering, Brigham Young University, Utah, United States of America

Correspondence
*Jared J. Thomas. Email: jaredthomas@byu.net

Abstract

In this paper we present a continuation optimization method for reducing multi-modality in the wind farm layout optimization problem that we call Wake Expansion Continuation (WEC). We achieve the reduction in multi-modality by starting with an increased wake diameter while maintaining normal velocity deficits at the center of the wakes, and then reducing the wake diameter for each of a series of optimization runs until the accurate wake diameter is used. We applied and demonstrated the effectiveness of WEC with two different wake models. We tested WEC on four optimization case studies with a gradient-based optimization method and a gradient-free optimization method. We found a significant improvement in the mean, standard deviation, and minimum wake loss for optimization with WEC compared to optimization without WEC for all test cases. We found the gradient-free optimization algorithm resulted in less optimal layouts on average for all cases than the gradient-based algorithm with WEC. We also applied WEC to the gradient-free algorithm for one case study with significantly improved results, but there was more improvement when we applied WEC to a gradient-based algorithm. WEC enables gradient-based algorithms to search the wind farm layout optimization space more globally, and provides more optimal results more consistently than optimization without WEC.

KEYWORDS:
Optimization, wind farm, gradient-based

1 | INTRODUCTION

The difficulty of solving the wind farm layout optimization (WFLO) problem is primarily due to the large number of variables and constraints required for realistic problems and the multi-modal nature of the problem's design space. Gradient-free optimization methods are the most common methods used to solve the WFLO problem. However, the performance of gradient-free methods is reduced in high dimensional problems[1]. The WFLO problem scales quickly to high dimensions as the number of turbines is increased. Gradient-based optimization methods are well suited for high dimensional problems, particularly if numerically exact derivatives are provided. Gradient-based methods are not widely used for WFLO problems because they are highly susceptible to local optima[2]. However, due to their relatively low computational cost and their ability to efficiently handle many variables and constraints, gradient-based methods are gaining interest in the WFLO community have been shown to find good solutions to WFLO problems[3–7].

Many techniques have been presented to make the WFLO problem more tractable, including discretization, multi-start, re-parameterization, and hybrid approaches. Discretization techniques, used with gradient-free methods, attempt to simplify the problem by reducing the number of possible solutions[8,9]. Through discretization, the number of possible turbine locations within a wind farm can be reduced from infinite to something on the order of hundreds of locations. However, discretization disregards any locations that are not pre-selected, and can thus preclude this approach from finding even a local optimum. It is also possible that constraints on variables other than position may render the discretized optimization problem intractable. Multi-start approaches involve running many optimizations of one problem with different starting points[7,10–12]. This approach reduces the sensitivity of gradient-based optimization methods to local optima. Re-parameterization approaches seek to reduce the complexity of the problem by defining the wind farm with just a few variables, such as row spacing, column spacing, grid rotation, etc. and can be very effective if the problem of interest can be properly parameterized[13]. Hybrid approaches combine gradient-based and gradient-free algorithms iteratively[14–16], and, depending on the problem size, can yield comparable results to multi-start approaches[14]. While each of these techniques yield improved results, there is still need for further improvement because current methods have a wide spread in the quality of results, are highly dependent on starting locations, artificially limit the design space, and/or cannot be applied in realistic wind farm optimization scenarios. The limitations of the existing methods indicate a need for better methods that avoid local optima, whether real or imposed by the approach itself.

The fluctuations of wind speed as turbines move in and out of the wakes during optimization are primarily responsible for the multi-modal nature of the WFLO problem. In this paper we propose a method designed to overcome the problem of local optima caused by the wakes. We reduce the impact of local optima by widening and combining wake regions without altering individual wake-center deficits. We then run a series of optimizations with increasingly realistic wake distributions until we get back to the original model. We will refer to the new method as Wake Expansion Continuation, or WEC.

In the following sections, we will: present an overview of WEC (section 2), introduce the simulation models used to study WEC (section 3), demonstrate how to apply WEC to existing models (section 4), provide a series of wind farm layout optimization case studies for comparing optimization methods (section 5), give some details on the computational environment used in our studies and discuss how we tuned the various methods for our case studies (section 6), present and discuss the results of the case studies (section 7), and provide concluding comments (section 8).

2 | INTRODUCTION TO THE WAKE EXPANSION CONTINUATION

In Gaussian continuation optimization, the design space is approximated using a series of radial basis functions[17]. When optimization is performed, the standard deviations of the radial basis functions starts at a relatively high value and slowly decreases until the original standard deviation is reached. Increasing the standard deviation of the basis functions has the effect of causing the various basis functions to blend in to each other, effectively removing the local optima and providing a relatively unimodal design space to the optimization algorithm. Slowly returning the standard deviation to the original value allows the optimization algorithm to adjust for any shift in the global optimum due to the blending of the basis functions and avoid local optima.

WEC works in a manner similar to Gaussian continuation optimization, except that because wind turbine wakes are roughly Gaussian-shaped, the Gaussian basis functions are built in to the model directly rather than used to approximate the model. Other key differences are that the basis functions in WEC are not radial, and that the Gaussian functions in the wind farm model change location during the WFLO because they are tied to the turbine locations. Because of these last two differences, we cannot guarantee that WEC will converge to the global optimum. However, we will show that use of the WEC method does significantly improve WFLO results. We have also presented a preliminary study showing significant benefit of the WEC method[12] and a study validating WEC optimization results with large eddy simulations (LES)[7].

The WEC method can be explained in three basic steps. The first two steps are preparatory, and need only be performed once for each wake model:

1) Determine how the wake diameter and wake deficit are controlled for the selected wake model.

2) If necessary, introduce a factor to the model such that the wake diameter can be directly controlled without significantly altering the wake deficit in the center of the wake. If the model already provides such control then this step is not needed. Wake center is analogous to the mean of a Gaussian distribution. Wake deficit is analogous to the leading coefficient of a Gaussian distribution. Wake diameter is analogous to the standard deviation of a Gaussian distribution. Keeping the wake deficit in the center of the wake constant, while changing the wake diameter, mimics the behavior of increasing the standard deviation without changing the height of a Gaussian distribution as shown by Mobahi and Fisher[17].

3) Run a series of optimizations such that the wake diameter is larger than normal for the first optimization, and then reduces with each subsequent optimization until the wake diameter is no longer altered from the original model. Each optimization after the first should be hot-started using the result of the previous optimization.

To apply the wake expansion technique to the wind farm layout optimization problem, we need to determine the best way to expand the basis functions, or wake diameter in this case. We have investigated three ways of expanding the wake: (1) increasing the wake spreading angle (WEC-A), (2) multiplying the initial wake diameter (WEC-D), and (3) a hybrid of WEC-A and WEC-D that uses WEC-A in the near wake and WEC-D in the far wake (WEC-H). The impact on the wake shape of each of these three methods of expanding the wake are shown in fig. 1.

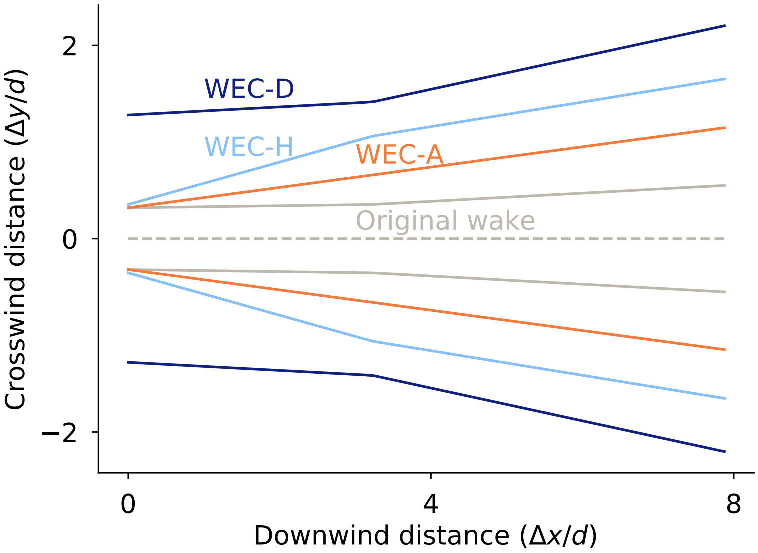


FIGURE 1 The impact on wake shape of expanding the wake by increasing the spreading angle (WEC-A), the diameter (WEC-D), or a downstream diameter that also changes the near wake spreading angle (WEC-H). The relative amount of expansion in the figure is for convenience in comparing the WEC methods. The actual amount of expansion is variable for all methods.

To test the wake expansion approaches discussed, we implemented each of them in the Bastankhah and Porté-Agel wake model and compared them in detail. We found that of the wake expansion methods we investigated, the WEC-D method is the best as it consistently had the lowest minimum, mean, and standard deviation of wake loss. The better performance of WEC-D over WEC-A and WEC-H can be explained by the fact that both WEC-A and WEC-H alter the wake shape, while WEC-D only multiplies the standard deviation of the basis functions in a manner consistent with Gaussian continuation optimization theory (see Mobahi and Fisher[17]). In the balance of this paper we will refer to WEC-D as WEC because the other variants are not relevant in the remainder of this discussion. For further details on the other wake expansion methods you can refer to (J. J. Thomas, Dissertation, Brigham Young University, 2021).

3 | SIMULATION MODELS

Because of the general nature of the proposed method, it can be applied to any wake model where the wake center deficit and wake diameter can be controlled independently. In this study we have applied WEC to the 2016 version of the Bastankhah and Porté-Agel wake model[18], which we will refer to as the Bastankhah model, and the cosine version of the Jensen wake model[19], which we will refer to as the Jensen cosine model. Following our discussion of the wake models we will describe the turbine and other models that we used for this work.

3.1 | The Bastankhah Wake Model

In the Bastankhah wake model, two primary characteristics of the wakes, wake deficit and wake diameter, are particularly easy to isolate. This, along with the smoothness and differentiability of the model, make it a good example for demonstrating WEC. We used the Bastankhah wake model, as defined in eq. (1)[18], along with the Niayifar and Porté-Agel wind farm model[20].

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left[1 - \sqrt{1 - \frac{C_T \cos(\gamma)}{8\sigma_y \sigma_z / d^2}}\right] \exp\left(-0.5\left[\frac{\Delta y - \delta}{\sigma_y}\right]^2\right) \exp\left(-0.5\left[\frac{z - z_h}{\sigma_z}\right]^2\right), \tag{1}$$

where $\Delta\bar{u}/\bar{u}_\infty$ is the normalized wake velocity deficit, C_T is the thrust coefficient, γ is the upstream turbine's yaw angle with respect to the inflow direction, $\Delta y - \delta$ and $z - z_h$ are the distances of the point of interest from the wake center perpendicular to the wind direction in the horizontal and vertical directions respectively. The standard deviation of the wake deficit in the horizontal direction is defined as

$$\sigma_y = k_y[\Delta x - x_0] + \frac{d\cos(\gamma)}{\sqrt{8}}, \tag{2}$$

and the standard deviation of the wake deficit in the vertical directions is defined as

$$\sigma_z = k_z[\Delta x - x_0] + \frac{d}{\sqrt{8}}, \tag{3}$$

where Δx is the downstream distance from the turbine generating the wake to the point of interest, x_0 is the length of the wake potential core, d is the diameter of the turbine generating the wake, and k_y and k_z are determined as a function of turbulence intensity (I) as defined in eq. (4)[20].

$$k^* = 0.3837I + 0.003678 \tag{4}$$

While the Niayifar and Porté-Agel wind farm model calculates k^* based on local turbulence intensity at each turbine, the local turbulence intensity calculations introduce more local optima and discontinuities. For this reason, we chose to ignore local turbulence intensity while using WEC and re-introduce a smooth version of the local turbulence intensity in a final optimization step following all WEC steps in studies performed using WEC with the Bastankhah wake model and gradient-based optimization methods. While local turbulence intensity does impact the accuracy of the power predictions, it does not alter the general trends within the design space, and most simple wake models ignore local turbulence intensity.

The Gaussian shape of the Bastankhah wake model is well suited for gradient-based optimization because it is smooth, continuous, and has no flat regions. However, in the near wake, the model can either be flat, which can cause premature convergence, or be undefined, which can cause optimizations to fail. Because no turbines will be placed in this region of the wake in the final optimized layout, the accuracy of the model in the near wake is second in importance to wake shape and continuity.

We define our near wake model using the location where the original model first begins to be defined, x_d, derived by Thomas et al.[7] and reproduced in eq. (5).

$$x_d = x_0 + d\left[\frac{k_y + k_z\cos(\gamma) - \sqrt{[k_y + k_z\cos(\gamma)]^2 - 4k_y k_z[C_T - 1]\cos(\gamma)}}{2k_y k_z\sqrt{8}}\right]. \tag{5}$$

We find the standard deviation of the wake at the point x_d as shown in eq. (6).

$$\sigma_{yd} = k_y[x_d - x_0] + d\frac{\cos(\gamma)}{\sqrt{8}}. \tag{6}$$

We then use the definition of the wake diameter at the point of discontinuity to provide an estimate for the wake diameter and velocity deficit at the rotor hub. With the assumption that σ_{yd} is the value of the wake diameter at the rotor hub, we can define the slope of the near wake, $k_{y_{\mathrm{NEAR}}}$, as shown in eq. (7).

$$k_{y_{\mathrm{NEAR}}} = \frac{\sigma_{yo} - \sigma_{yd}}{x_0}. \tag{7}$$

For more details on near wake approximation of the Bastankhah wake model used in this study, please see[7].

3.2 | The Jensen Cosine Wake Model

The Jensen cosine wake model is a variant of the popular "top hat" model often referred to as the Jensen model. The cosine version simply multiplies the top hat shape with a factor that changes the wake deficit shape to follow a cosine curve (see fig. 2b for a comparison of the top hat and cosine variants of the Jensen model). We used the Jensen cosine wake model as defined in eq. (8), along with the Katic wake combination model[21]

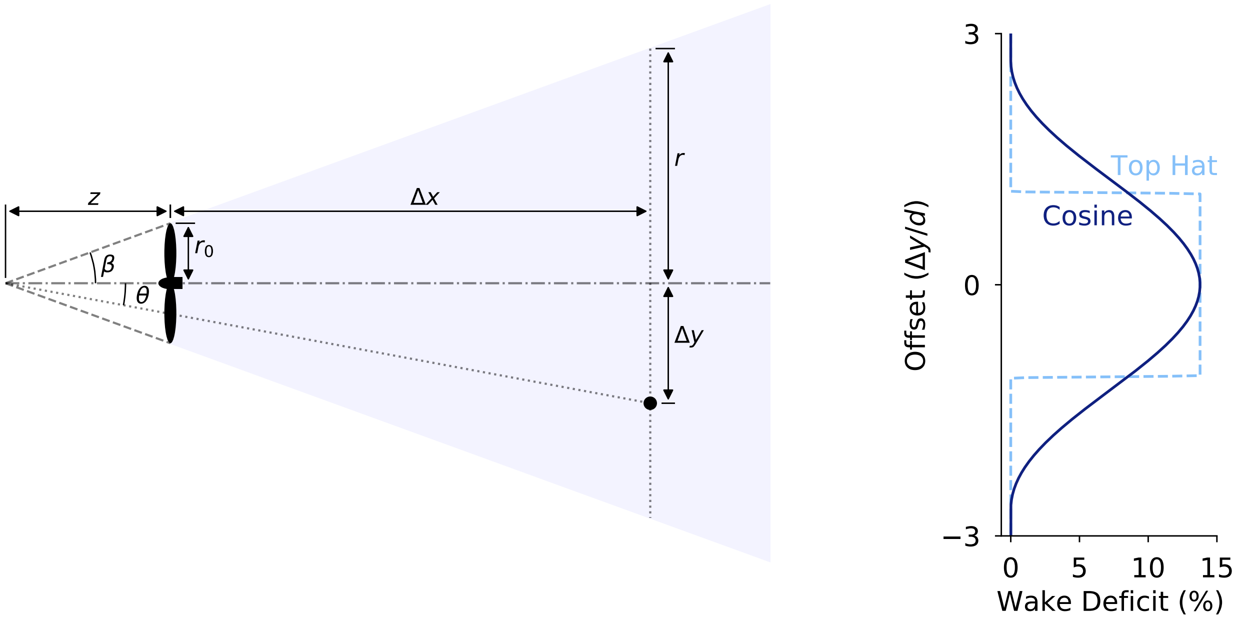
$$\frac{\bar{u}}{\bar{u}_\infty} = 1 - 2a\left[\frac{r_0}{r_0 + \alpha \Delta x}\right]^2 f_\theta, \tag{8}$$

where $\bar{u}$ represents the wind velocity at a distance Δx downstream of the waking turbine, $\bar{u}_\infty$ represents the free-stream wind velocity, r_0 represents the radius of the wind turbine, α represents the wake entrainment constant[19], and a represents the axial induction (for which we assumed an ideal value of $a = 1/3$). The cosine factor, f_θ is defined as

$$f_\theta = \frac{1 + \cos(n\theta)}{2}, \tag{9}$$

where θ is the angle from the wake center line to the point of interest measured from the wake vertex (a distance z upstream of the wind turbine as shown in fig. 2a), and n is a factor derived from the wake spreading angle, β. The value of n can be calculated as $n = \pi/\beta$.

We used the same wake spreading angle as Jensen, $\beta = 20°$ [19]. A comparison of the velocity deficit profiles for the Jensen top hat and Jensen cosine wake models is provided in fig. 2b.



(a) Horizontal geometry of the Jensen cosine wake model. The wind is blowing to the right. The dashed line down the middle represents the center line of the wake. The large black dot represents any given point of interest.

(b) Wake deficit profiles at 6 rotor diameters down wind for the Jensen top hat and Jensen cosine wake models.

FIGURE 2 The Jensen wake models[19].

3.3 | Turbine Model

We based our turbines on the Vestas V-80 2MW wind turbine for all studies in this paper. The values of C_P and C_T were based on a linear interpolation of the power and thrust coefficient curves presented by Niayifar and Porté-Agel[20] and shown in figs. 3 and 4. The other turbine specifications are provide in table 1. The C_T curve was only used with the Bastankhah wake model. We used a constant axial induction (a) of 1/3 and set α to 0.1 when using the Jensen cosine wake model.

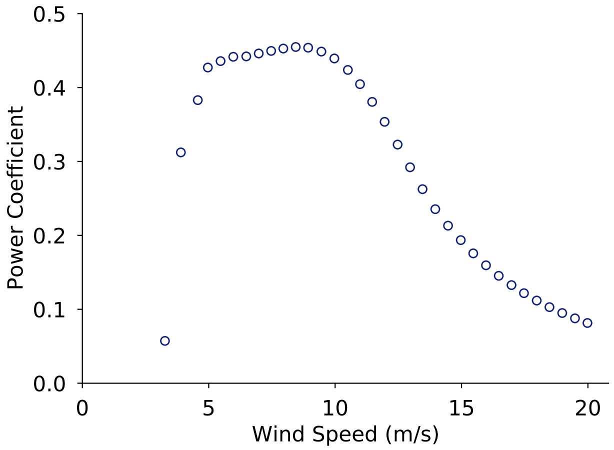


FIGURE 3 C_P curve for the Vestas V80 2MW wind turbine[20] for a given rotation speed schedule.

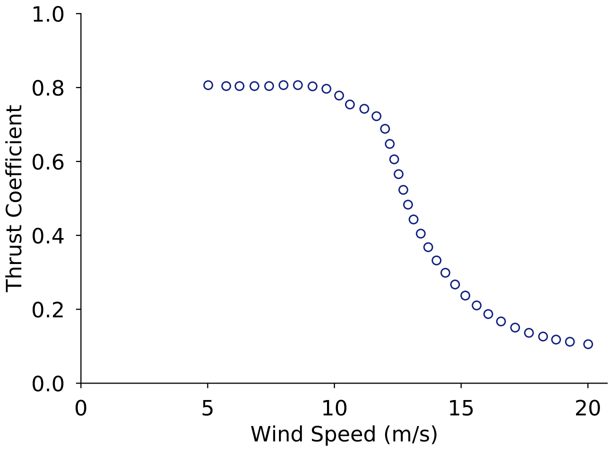


FIGURE 4 C_T curve for the Vestas V80 2MW wind turbine[20] for a given rotation speed schedule.

TABLE 1 Turbine Specifications

Rotor Diameter	80.0 m
Hub Height	70.0 m
Cut-in Speed	4.0 m/s
Cut-out Speed	25.0 m/s
Rated Speed	16.0 m/s
Rated Power	2.0 MW
Generator Efficiency	0.944

3.4 | Other Models

We combined the wake deficits using a linear combination method[20] with the Bastankhah wake model, and a sum of squares method[21] with the Jensen wake model. We used a reference height of 80 m for all wind speed measurements and adjusted to different heights with a power law using a wind shear exponent of 0.31 as shown in eq. (10).

$$u = u_r \left[\frac{z}{z_r} \right]^{\psi}, \tag{10}$$

where u_r is the reference wind speed, z is the height of interest, z_r is the height at which u_r was measured, and ψ is the shear exponent.

To save computation time, the inflow wind speed at each turbine was approximated using a single point at the wind turbine hub location. Individual turbine inflow wind velocities, U_i, were solved consecutively from upstream to downstream for each wind state (direction and speed combination) for the Bastankhah wake model, but in no particular order for the Jensen cosine model. The power output of each turbine was then calculated as

$$P_i = \frac{1}{2} \rho A_{ri} C_P U_i^3, \tag{11}$$

where ρ is the air density, $A_{r,i}$ is the rotor-swept area of turbine i, and C_P is the power coefficient. We calculated annual energy production (AEP) as

$$AEP = [24][365] \sum_{j=1}^{n_s} \sum_{i=1}^{n_t} p_j P_{ij}, \tag{12}$$

where n_s is the number of wind states (direction and speed combination), p_j is the probability of a given wind state, n_t is the number of wind turbines, and P_{ij} is the power produced by turbine i given wind state j.

We maximized AEP as the objective of the optimizations in this paper, but present the results using wake loss because it provides a convenient baseline for comparison. We calculated wake loss (energy lost due to wake effects) across all the optimization results as

$$L = 100\left[1 - \frac{AEP_0}{AEP_t}\right], \tag{13}$$

where AEP_0 represents the optimized AEP found from a given starting layout and AEP_t is the theoretical maximum AEP calculated as shown in eq. (14).

$$AEP_t = [24][365] \sum_{j=1}^{n_s} p_j P_j, \tag{14}$$

where P_j is the power of a single un-waked wind turbine at wind state j calculated using eq. (11).

4 | APPLYING WEC TO THE WAKE MODELS

In this section we apply WEC, step by step as discussed in section 2, to the Bastankhah wake model and the Jensen cosine wake model. We present how to apply the first two steps of the WEC method to each wake model. The final step of the WEC method is the same for all models, with the exception that different WEC factors and number of WEC steps may be optimal for different wake models.

4.1 | Applying WEC to the Bastankhah Wake Model

4.1.1 | Bastankhah WEC Step (1)

The first parenthetical term of eq. (1) defines the magnitude of the velocity deficit. The exponential terms determine the wake diameters in the horizontal and vertical directions. The wake diameters and velocity deficit are coupled through σ_y and σ_z. However, because the diameters and deficit are expressed in separate terms, it is possible to adjust the diameters without impacting the deficit by adjusting the σ_y and σ_z values in only the diameter part of the equation.

4.1.2 | Bastankhah WEC Step (2)

Independent control of the wake diameter is obtained by introducing a factor, ξ, to σ_y and σ_z in the exponential terms of eq. (1), as shown in eq. (15). The only change from eq. (1) to eq. (15) is the addition of the WEC factor, ξ.

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left[1 - \sqrt{1 - \frac{C_T \cos(\gamma)}{8\sigma_y\sigma_z/d^2}}\right] \exp\left(-0.5\left[\frac{\Delta y - \delta}{\xi\sigma_y}\right]^2\right) \exp\left(-0.5\left[\frac{z - z_h}{\xi\sigma_z}\right]^2\right) \tag{15}$$

Increasing ξ widens the wakes without changing the wake center velocity deficit. Widened wakes mix and smooth out the local optima, as shown in fig. 5.

4.2 | Applying WEC to the Jensen Cosine Wake Model

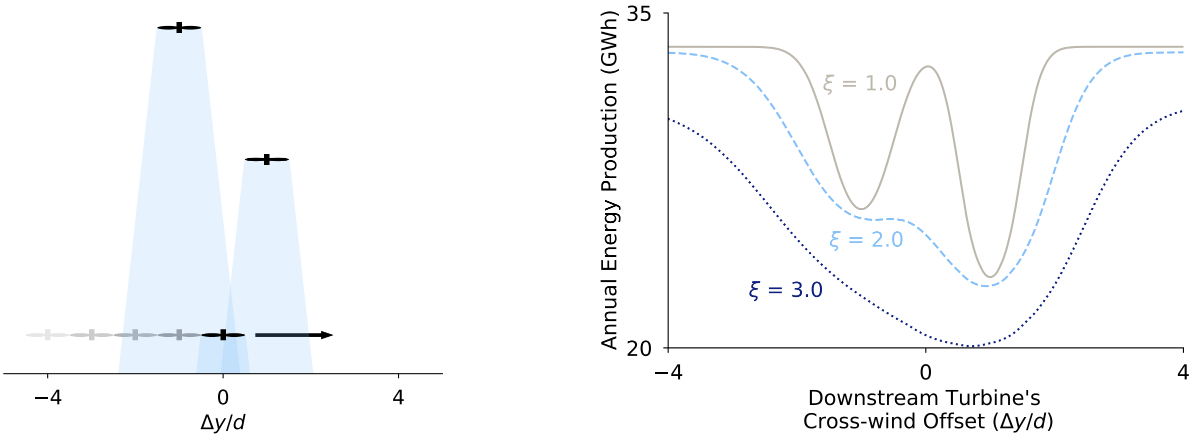
In this section we apply WEC to the Jensen cosine wake model. A diagram displaying the Jensen cosine wake model from an overhead perspective is shown in fig. 2a. The diagram will give context to the variables and parameters discussed in this section.

4.2.1 | Jensen Cosine WEC Step (1)

The cosine factor, f_θ, in eq. (8) controls the wake diameter for the Jensen cosine wake model. By breaking down the cosine factor into its constituent parameters, it is possible to adjust the wake diameter without impacting the velocity deficit in the center of the wake.

As can be seen in eq. (9), f_θ is a function of θ, which represents the angle between the wake's center line and the downwind turbine's location as measured from the wake vertex (a distance z along the wake center line upstream of the wind turbine). This angle θ can be calculated as

$$\theta = \arctan\left(\frac{\Delta y}{\Delta x + z}\right), \tag{16}$$



(a)

(b)

FIGURE 5 The impact of the WEC factor, ξ, for WEC using the Bastankhah wake model. A simple wind farm with one movable turbine and one wind direction is shown from above in fig. 5a. Wind is from the top. Increasing the WEC factor removes the local optima between the wakes of the two front turbines, as demonstrated in fig. 5b.

where Δx and Δy represent the spacing in the wind direction and perpendicular to the wind direction, respectively, between the upwind turbine and the location of interest. The variable z represents the distance between the wake's vertex and the upwind turbine. By adjusting the value of z, we are able to increase or decrease the wake's spread. The expression for z based on the initial wake diameter is

$$z = \frac{r_0}{\tan(\beta)}. \tag{17}$$

4.2.2 | Jensen Cosine WEC Step (2)

We can directly adjust the wake diameter without impacting the magnitude of the deficit in the center of the wake by applying a factor, ξ, to the rotor radius, r_0, in eq. (17), as seen below in eq. (18).

$$z = \frac{\xi r_0}{\tan(\beta)} \tag{18}$$

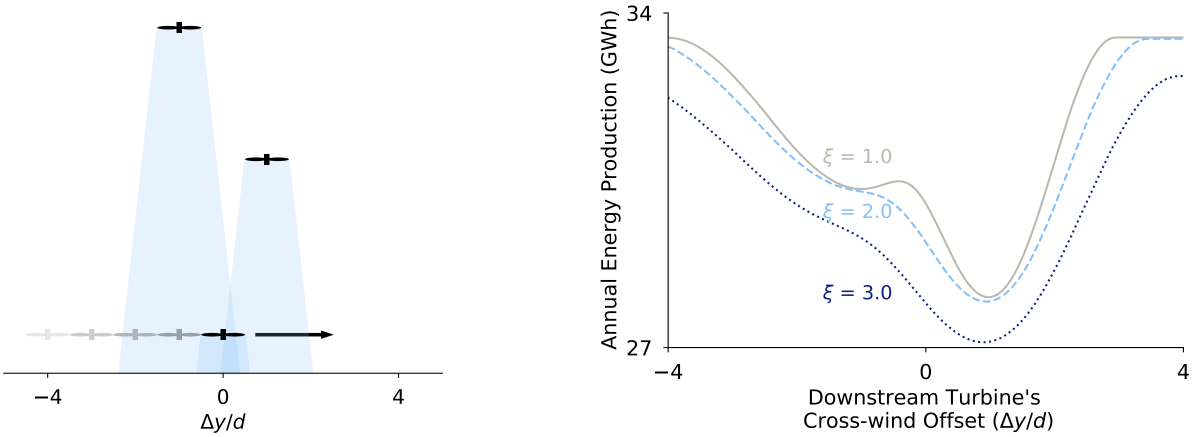
Through a series of substitutions we can combine the new vertex distance, eq. (18), with the Jensen cosine wake model to obtain

$$\frac{\bar{u}}{\bar{u}_\infty} = 1 - a\left[\frac{r_0}{r_0 + \alpha x}\right]^2 \left[1 + \cos\left(\frac{\pi}{\beta}\arctan\left(\frac{\Delta y}{\Delta x + \frac{\xi r_0}{\tan(\beta)}}\right)\right)\right]. \tag{19}$$

Because we have inserted the WEC factor, ξ, into eq. (19), we may adjust the wake diameter without changing the velocity deficit in the center of the wake. Note that this is true because when $\Delta y = 0$, the term inside the inverse tangent in eq. (19) goes to zero, regardless of the value of ξ. Because this is the only place where the WEC factor is found in eq. (19), the velocity deficit will be constant with respect to ξ at the wake center. This behavior is shown in fig. 6, which also shows that local optima can be smoothed out through WEC for the Jensen cosine model.

5 | CASE STUDIES

To demonstrate the effectiveness of WEC on a range of problems, we present three wind farm optimization case studies. The cases were chosen to represent a range in size, complexity, and difficulty of the optimization problem.



(a)

(b)

FIGURE 6 The impact of the WEC factor, ξ, on the Jensen cosine wake model. A simple wind farm with one movable turbine and one wind direction is shown from above in fig. 6a. Wind is from the top. Increasing the WEC factor removes the local optima between the wakes of the two front turbines, as demonstrated in fig. 6b.

5.1 | Case 1: 16 Turbines and 20 Wind Directions

We selected case 1 to provide a meaningful problem that would be tractable for nearly any optimization method. We defined a wind farm with 16 wind turbines and a square boundary with enough space for four rows and columns of wind turbines with a five rotor diameter spacing between rows and columns (see fig. 7). We used a simple wind rose composed of a double Gaussian distribution binned into 20 directions and a constant wind speed of 10 m/s in all directions (see fig. 8).

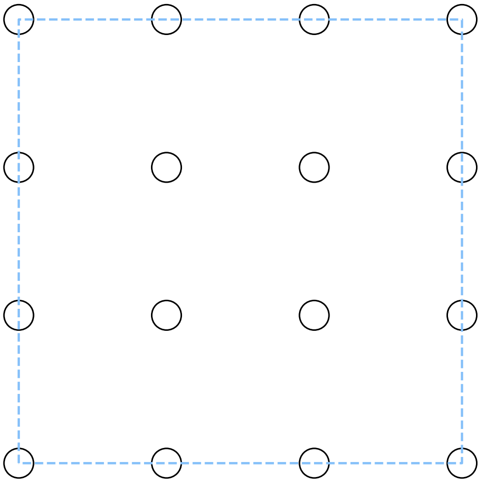


FIGURE 7 Baseline wind farm layout for case 1. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

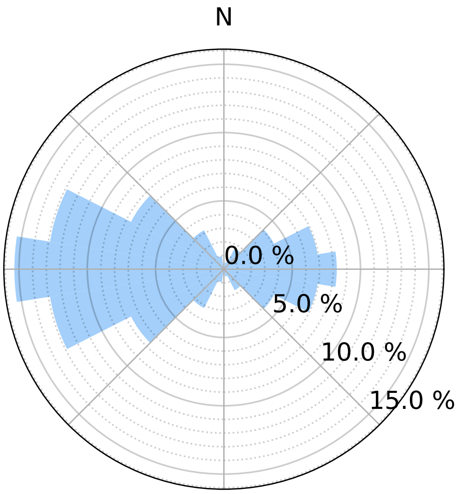


FIGURE 8 Direction probability wind rose for case 1. This wind rose is composed of a double Gaussian distribution binned into 20 directions.

For case 1, the optimization problem was formulated as

$$\begin{aligned}\underset{x_i,y_i}{\text{maximize}} \quad & AEP(x_i, y_i) \;\; i = 1...16,\\ \text{subject to} \quad & s_{ij} \geq 2d \;\; i,j = 1...16, \;\; i \neq j,\\ & x_{min} \leq x_i \leq x_{max} \;\; i = 1...16,\\ & y_{min} \leq y_i \leq y_{max} \;\; i = 1...16,\end{aligned} \tag{20}$$

where (x_i, y_i) is the position of each turbine i, $s_{i,j}$ represents the separation distance between each pair of turbines i and j, and $x_{max/min}$ and $y_{max/min}$ represent the boundaries of the wind farm. This case has a total of 32 variables and 120 constraints.

5.2 | Case 2: 38 Turbines and 12 Wind Directions

We created case 2 to be significantly more challenging than case 1 while remaining simple enough to be reasonably simulated using LES[7]. We defined a wind farm with 38 turbines and a circular boundary as shown in fig. 9. The circular boundary was originally chosen to simplify LES simulation. The boundary size allowed for a five-diameter minimum spacing between turbines. We used the Nantucket wind rose binned into 12 directions with the wind speed for each direction set to 8 m/s as shown in fig. 10.

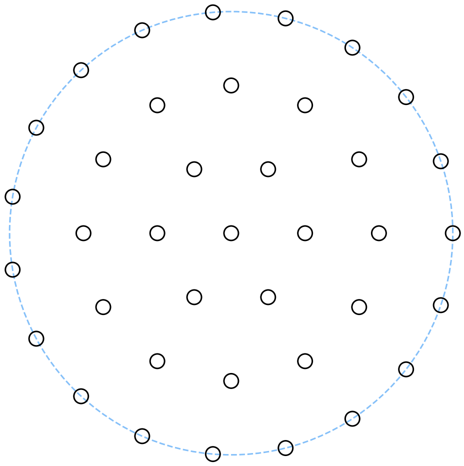


FIGURE 9 Baseline wind farm layout for cases 2 and 3. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

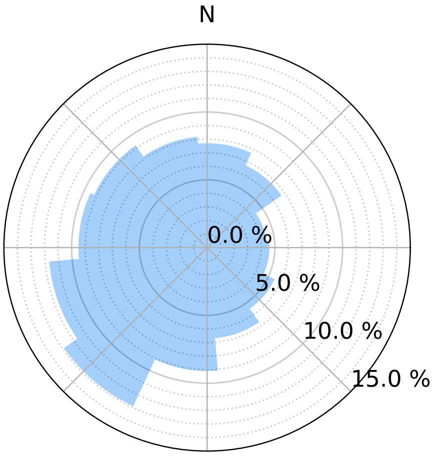


FIGURE 10 Directional probability wind rose for case 2. This is the Nantucket wind rose binned into 12 directions[22].

The optimization problem for case 2 was formulated as

$$\begin{aligned}\underset{x_i,y_i}{\text{maximize}} \quad & AEP(x_i, y_i) \;\; i = 1...38,\\ \text{subject to} \quad & s_{ij} \geq 2d \;\; i,j = 1...38 \;\; i \neq j,\\ & [x_c - x_i]^2 + [y_c - y_i]^2 \leq r_b^2 \;\; i = 1...38,\end{aligned} \tag{21}$$

where (x_c, y_c) is the location of the center of the wind farm, and r_b is the radius of the wind farm boundary. Case 2 has a total of 76 variables and 741 constraints.

5.3 | Case 3: 38 Turbines and 36 Wind Directions

Case 3 is the same as case 2 but with more wind directions and different wind speeds in each direction. We used the Nantucket wind rose binned into 36 directions with the wind speed for each direction determined as the average of all samples in that sector as shown in figs. 11 and 12. The optimization problem for case 3 was formulated as shown in eq. (21) with the same number of variables and constraints.

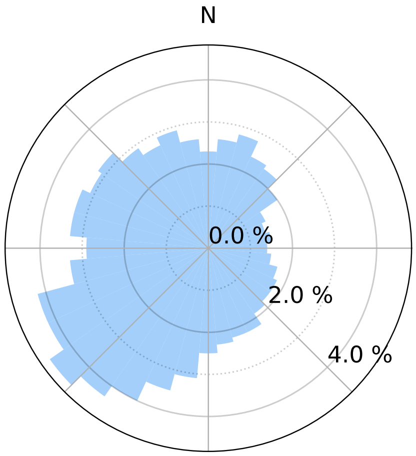


FIGURE 11 Directional probability wind rose for case 2. This is the Nantucket wind rose binned into 36 directions[22].

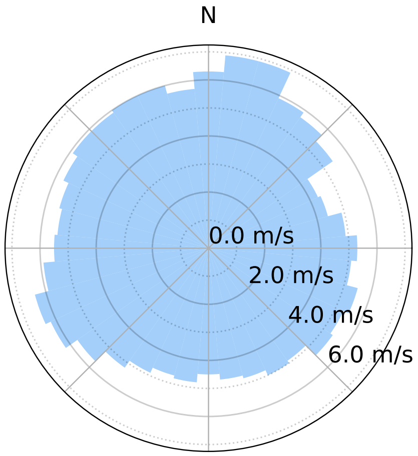


FIGURE 12 Average speed wind rose for case 2. This is the Nantucket wind rose binned into 36 directions[22]

5.4 | Case 4: 60 Turbines and 72 Wind Directions

We selected case 4, based on the Princess Amalia Wind Park, to provide a larger and somewhat more realistic problem. The Amalia wind farm has 60 wind turbines. We used the convex hull of the existing turbine locations to create the boundary fig. 13. We used wind data binned into 72 wind directions with the wind speed for each direction determined as the average of all samples in that sector as shown in figs. 14 and 15.

For case 4, the optimization problem was formulated as

$$\begin{aligned} \underset{x_i,y_i}{\text{maximize}} \quad & AEP(x_i, y_i) \;\; i = 1...60, \\ \text{subject to} \quad & s_{ij} \geq 2d \;\; i, j = 1...60, \;\; i \neq j, \\ & b_{ik} \geq 0 \;\; i = 1...60, k = 1...14, \end{aligned} \tag{22}$$

where $b_{i,k}$ represents the distance of each turbine i from each boundary k. Case 4 has a total of 120 variables and 2610 constraints.

6 | IMPLEMENTATION AND OPTIMIZATION METHODS

We implemented the code for each wake model in Fortran and wrapped it with Python. We set up and solved the optimization problems in OpenMDAO[24] using two optimization algorithms, the gradient-based Sparse Nonlinear OPTimizer (SNOPT)[25] and the gradient-free Augmented Lagrangian Particle Swarm Optimizer (ALPSO)[26]. We used both algorithms were through pyOptSparse[27]. We obtained exact gradients for both wake models using algorithmic differentiation provided by Tapenade[28]. Other gradients were obtained by hand.

To statistically compare results between the optimization methods, we created 199 pseudo-random starting wind farm layouts and one planned layout for each case, for a total of 200 different starting layouts for each case (basically a multi-start approach). Each of the starting layouts had all the turbines inside the wind farm boundary and did not have any turbines spaced less than one rotor diameter apart. The same sets of starting layouts were used for all optimization methods.

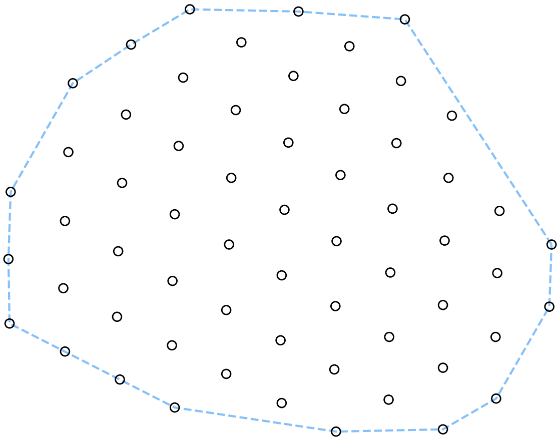


FIGURE 13 Baseline wind farm layout for case 4 with 60 wind turbines. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

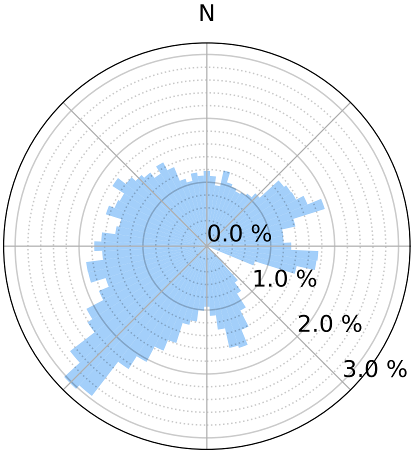


FIGURE 14 Direction probability wind rose for case 4. The wind rose is comprised of data from Noorzeewind[23]. Measurements were taken from July 1, 2005, to June 30, 2006.

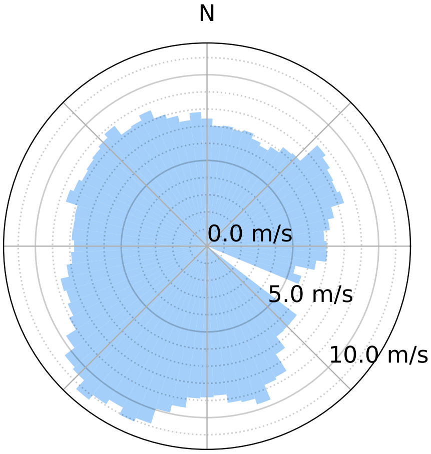


FIGURE 15 Average speed wind rose for case 4. The wind rose is comprised of data from Noorzeewind[23]. Measurements were taken from July 1, 2005, to June 30, 2006.

In the following subsections we present how we tuned and used SNOPT, ALPSO, and WEC to enable fair comparisons between the methods.

6.1 | SNOPT

We used two optimizations for each run with SNOPT when using the Bastankhah model. The first run did not use local turbulence intensity; the second one did. We ran SNOPT with different convergence tolerances for each case. We determined tolerances based on the tolerance required to achieve the majority of the improvement in several test runs without unnecessarily delaying termination. Convergence tolerances used in each case are presented in table 2. We scaled the objective and constraint derivatives for all cases to be between ±1. We formulated the objective function in kWh and then scaled it by 1×10^{-4} for optimization with SNOPT.

TABLE 2 SNOPT Scaling for Each Case Study

	Bastankhah		Jensen Cosine
	Without Local TI	With Local TI	
Case 1	1×10^{-2}	1×10^{-3}	1×10^{-3}
Case 2	9×10^{-3}	1×10^{-3}	
Case 3	9×10^{-3}	1×10^{-3}	
Case 4	1×10^{-3}	1×10^{-4}	

6.2 | ALPSO

We ran ALPSO using the default parameters provided in pyOptSparse[29] with a few exceptions. We set the craziness velocity to be 1×10^{-2} as this value resulted in increased optimized AEP compared to the default value across all the cases we tested. We tried adjusting the initial particle velocity, but found no difference in results. ALPSO introduces a new parameter, inner iterations, that is not found in a typical particle swarm method. The inner iteration parameter controls how many times the unconstrained problem formulation is run before the Lagrange multipliers and penalty factors (used to enforce the constraints) are updated. The outer iteration count is then used to represent how many sets of inner iterations are run. As demonstrated in by Jansen and Perez[26], we tested a series of values for inner iteration number and found that the inner iteration count had a large impact on the end results and convergence rates for all cases tested. We used different inner iteration counts for each case study, but held the number of function calls relatively constant at about 2×10^{4} as all cases appeared converged after this many function calls regardless of the number of inner iterations. We used a constant population seed of 1.0 while testing inner iteration counts and a random seed for the final case study results. The number of outer iterations was based on the number of inner iterations and desired function call cap for each case. We scaled the objective and design variables to be between ±1. All optimization runs using ALPSO had a population size of 30. The adjusted meta-parameters we used for the various ALPSO runs are shown in table 3.

TABLE 3 Varied ALPSO Meta-Parameters

	Inner Iterations	Outer Iterations	Function Calls
Case 1	5	134	20130
Case 2	25	28	21030
Case 3	15	45	20280
Case 4	10	68	20430

While it may be noted that each optimization run with ALPSO is similar in population size to running 30 optimizations using a gradient-based algorithm, we decided to run a full set of 200 optimizations for each test, just as was done for SNOPT. Using the same number of optimizations for ALPSO as for SNOPT is a benefit to ALPSO in terms of comparison because ALPSO is already designed to search the space broadly and should not need as many different starts to find a good result. This decision was to reflect how optimizations are often performed in practice, with many optimizations being run regardless of the algorithm being used, as well as to simplify comparison. While ALPSO and other population-based algorithms do carry many samples of the design space concurrently, they are used to inform one another and drive to a single final solution, playing a similar role as the gradients do in gradient-based optimization. We considered a direct comparison between full optimizations, rather than population members, to be more informative in how results would look in practice using each algorithm. This approach also enabled a direct comparison of function calls and the optimization objective.

6.3 | SNOPT+WEC

There are two parameters that need to be tuned for WEC: the WEC factor, and the number of steps. For tuning WEC, we used the following approach. When the maximum WEC factor value was varied, the number of steps was held at six. When the number of steps was varied, the maximum WEC factor was held at $\xi = 3$. We tested all 200 starting layouts with each parameter set. We did not adjust the k^* value in eq. (4) to the local turbulence intensity during optimization, but we did adjusted it for calculating the results shown. We used Case 2 (discussed in section 5.2) for tuning.

The mean wake loss results for tuning WEC are shown in figs. 16 and 17. Here we see that, on average, WEC results in less wake loss with a WEC factor of 3 and using 5 or more steps. The lowest standard deviations also occurred when the WEC factor was 3. We found it best to use at least 4 or 5 steps to achieve a small standard deviation (meaning more consistent results). Based on these results, we decided to use 6 steps and a maximum ξ value of three for the case studies.

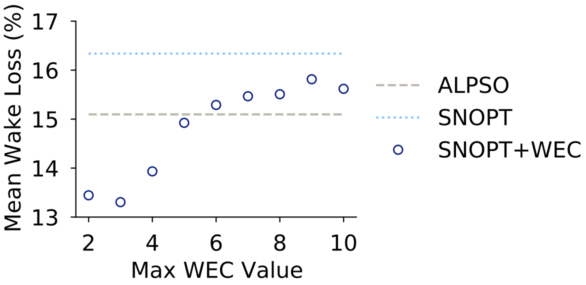


FIGURE 16 Mean wake loss results for varying the maximum WEC expansion factor while holding the number of steps constant at six. Each data point represents 200 separate optimizations with different starting points.

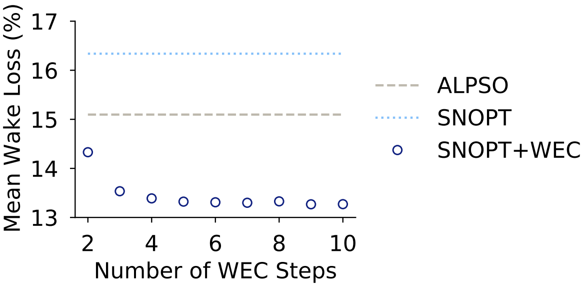


FIGURE 17 Mean wake loss results for varying the number of WEC steps while holding the expansion factor constant at three. Each data point represents 200 separate optimizations with different starting points.

When optimizing with WEC in the case studies, we adjusted the convergence tolerances because the WEC steps are primarily for exploring the design space and escaping local optima. The corresponding convergence tolerances and WEC factors used at each step with SNOPT+WEC are shown in table 4.

TABLE 4 WEC Parameters and Convergence Tolerances

WEC step	1	2	3	4	5	6	Final
WEC Factor (ξ)	3.0	2.6	2.2	1.8	1.4	1.0	1.0
Local TI	No	No	No	No	No	No	Yes
Case 1	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-3}
Cases 2-4	9×10^{-3}	9×10^{-3}	9×10^{-3}	9×10^{-3}	9×10^{-3}	9×10^{-3}	1×10^{-3}

6.4 | ALPSO+WEC

Because WEC alters the design space, it can be applied with any optimization algorithm. To see if WEC may be helpful for gradient-free algorithms, we tested the impact of WEC with ALPSO in case 2 (see section 5.2). For this test we used a maximum

WEC factor of 3 with 6 steps, the same maximum ξ value and number of steps that we found to be best for using WEC with SNOPT. The objective and constraints were all scaled to be between ±1. We chose to use 25 inner iterations, the number of inner iterations that was best for ALPSO without WEC for this case. The WEC steps were applied by dividing the outer iterations by the number of WEC steps plus 1 and rounding up, so each WEC step was run with 5 outer iterations. The WEC steps were run with no local TI, just as for WEC with SNOPT, and a final 5 outer iterations was run with local TI.

It is likely that there are more optimal settings for using WEC with ALPSO, that may be found by tuning the number of inner and outer iterations for the changing design spaces at each WEC step. When using WEC with SNOPT, most of the turbine movement tends to occur during the first WEC step. It seems reasonable then that using more outer iterations in the first WEC step with ALPSO may lead to better results because the early part of the optimization is when most of the exploration takes place and the highest WEC factor provides the most reduction in local optima. However, this case was only performed as a basic test to see if there was any significant benefit for applying WEC while optimizing with ALPSO and so parameter values were based on what we learned from the preceding tuning studies.

6.5 | WEC with the Jensen Cosine Model

As a proof of concept in applying WEC to other wake models, we implemented WEC with the Jensen cosine model using the same maximum value of ξ (3) and the same number of WEC steps (6) as found best for use with WEC as applied to the Bastankhah model in section 6.3. We tested WEC with the Jensen cosine model only on case 2 (38 turbines and 12 directions. See section 5.2). Further improvements with WEC are likely available if WEC were tuned to this model and case combination.

7 | RESULTS AND DISCUSSION

In the WFLO problem, we want low wake loss values, tight distributions, and relatively few function calls. The benefit of low wake loss is increased efficiency, leading to a reduced cost of energy. Tight distributions indicate that results are more consistent and less dependent on the starting layout, so we should need fewer optimizations to be confident of having a good wind farm design. A low number of function calls shows that wind farm design optimization can be done more quickly, more variables could be tested, and planning costs could be reduced. WEC generally reduces the variance and value of wake loss while keeping function calls well below the number required for gradient-free optimization. Starting and final wake loss distributions for the case studies are shown in the box plots of fig. 18.

We can see in fig. 18 that all the optimization methods applied were successful at finding fairly good results, but it is clear that WEC has a significant impact, reducing the mean, minimum, and standard deviation of the wake loss distributions for all cases as compared with SNOPT or ALPSO alone, with the exception that ALPSO found the best overall result for case 4. However, SNOPT+WEC provided the best results on average for all cases. For case 2, the SNOPT+WEC distribution does not even overlap with the SNOPT distribution.

While most of the distributions are fairly normally distributed, the WEC results for case 1 did have some high outliers. It is also clear that WEC improved the results of ALPSO on case 2 on average and in finding a better overall solution. The impact of WEC with ALPSO was less than the impact of WEC with SNOPT. The low outliers in the starting location wake loss distributions are the designed layouts depicted in figs. 7, 9 and 13.

The convergence history for SNOPT, SNOPT+WEC, and ALPSO for each case study are shown in figs. 19 to 23. The ALPSO runs all terminated with the same number of function calls because function calls was controlled by the number of inner and outer iterations (see table 3). The straight lines in the ALPSO histories (from 10^0 to about 1.5×10^2 in fig. 19) represent the first outer iteration because we have only plotted ALPSO and ALPSO+WEC points at the completion of each outer iteration. This pattern is seen in the ALPSO and ALPSO+WEC convergence histories for all of the case studies. We calculated wake loss values using the wake models without WEC and with local TI, if applicable, in all the convergence history figures. This way we can see what is happening in the design space we are interested in, rather than the altered design space used to inform the optimization algorithms. The y-axes in figs. 19 to 23 correspond with the y-axis in fig. 18, but with bounds adjusted as appropriate for visualizing the convergence histories of each case study. We chose to use a log scale for function calls because of the large difference in the number of function calls used for gradient-based methods compared to the number of function calls used for the gradient-free methods. The median number of function calls differed by one to two orders of magnitude (see tables 5 to 9).

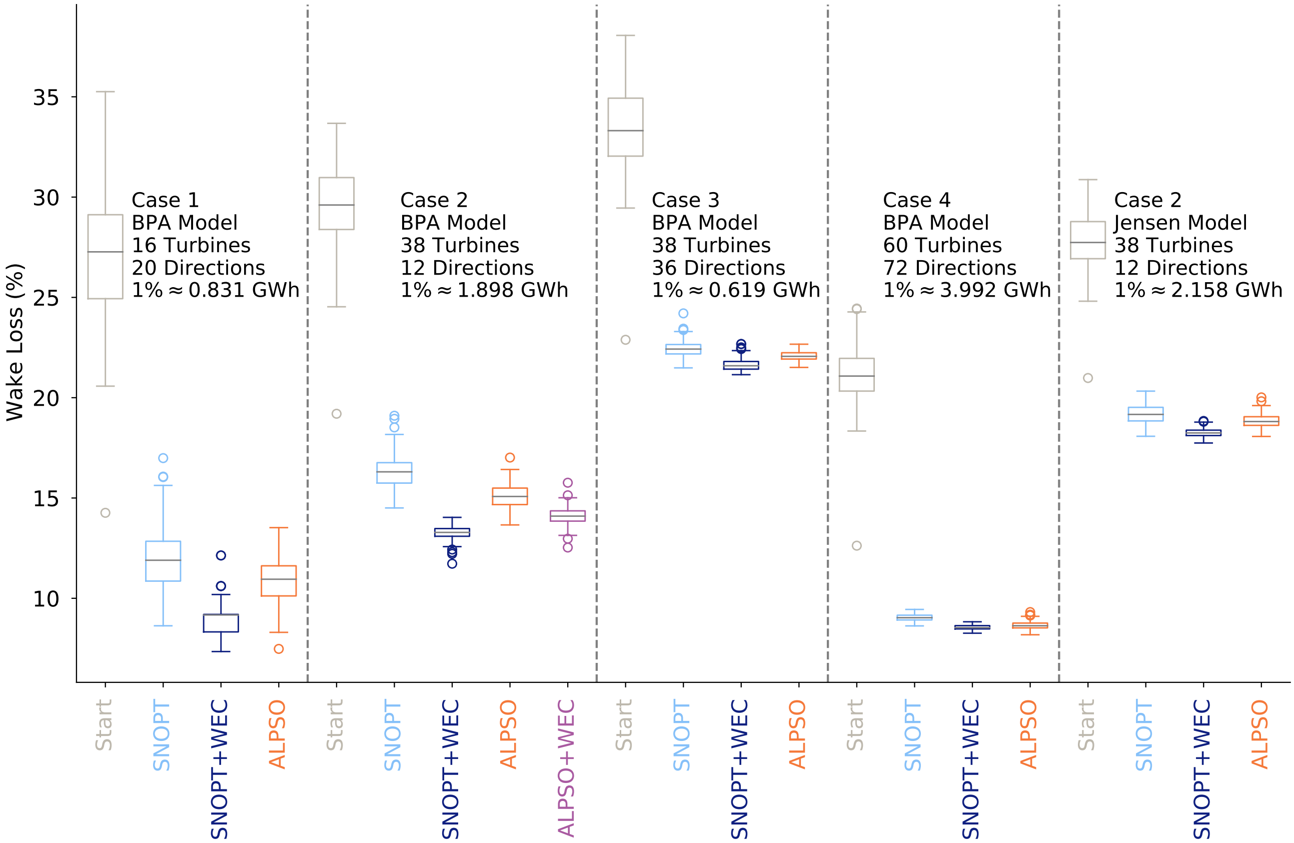


FIGURE 18 Results distributions of all case studies. BPA refers to the Bastankhah Model. Jensen refers to the Jensen cosine model.

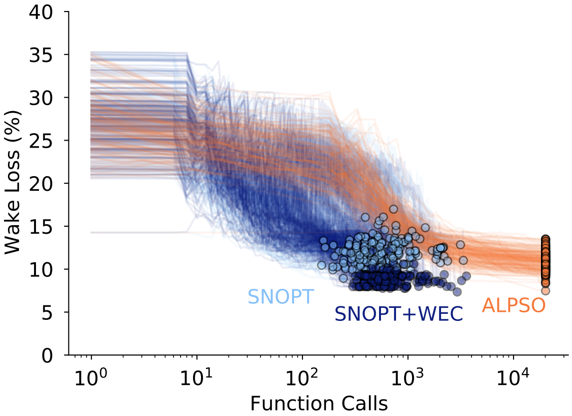


FIGURE 19 Convergence histories of optimizations for case 1 (16 turbines and 20 wind directions) using the Bastankhah model. Markers indicate optimized values.

The convergence histories for case 1, shown in fig. 19, illustrate the significant drop in wake loss due to using WEC with only a small extra cost in function calls compared to SNOPT alone. The same effect can be seen in the other cases as shown in figs. 20 to 23. While SNOPT+WEC did terminate in fewer function calls than SNOPT in case 4 (fig. 23), SNOPT+WEC required more function calls on average than SNOPT alone. There is not a clear pattern for convergence rate between SNOPT alone and SNOPT+WEC across the case studies. The rate of convergence was much higher for SNOPT than for ALPSO, with or without WEC, with SNOPT alone nearing convergence within the same number of function calls required for ALPSO to complete a single outer iteration. The convergence rate is the most noticeable difference between the Bastankhah and Jensen model results. As can be seen in comparing fig. 20 and fig. 21, SNOPT, with or without WEC, generally converges faster when using the Jensen model than when using the Bastankhah model. However, the convergence rate of ALPSO appears quite similar for both the Bastankhah and Jensen models. The difference in convergence rate for the gradient-based algorithm is likely due to the simpler and smoother nature of the Jensen model compared with the Bastankhah model.

Because WEC allows transitions through, and out of, local optima, the wake loss for cases with WEC applied often remain at higher levels longer in the convergence histories than without WEC and then drop sharply just before convergence. This is a feature of the method and demonstrates the effectiveness of WEC at allowing optimization algorithms to escape local optima by moving through locations that would be relatively poor in the non-WEC design space. The different characteristics in the convergence paths with and without WEC, for both SNOPT and ALPSO, are very apparent in fig. 20, where both algorithms with WEC stay higher longer and then drop rapidly below the results of the algorithms without WEC.

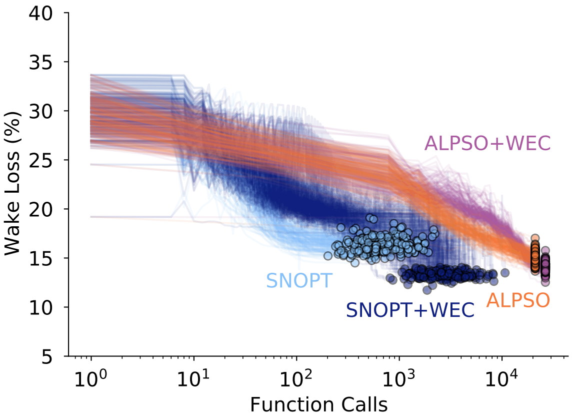


FIGURE 20 Convergence histories of optimizations for case 2 (38 turbines and 12 wind directions) using the Bastankhah model. Markers indicate optimized values.

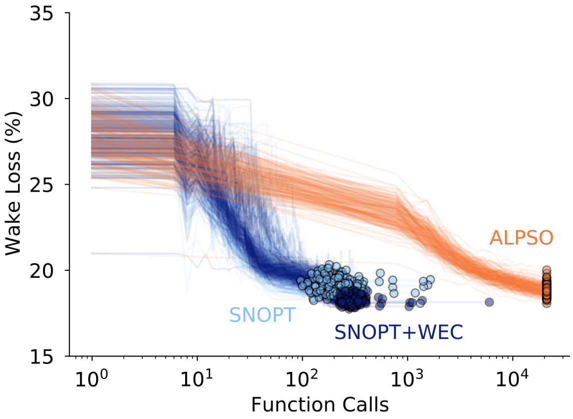


FIGURE 21 Optimization convergence histories for case 2 (38 turbines and 12 wind directions) using the Jensen model. Markers indicate optimized values.

Cases 3 and 4 (figs. 22 and 23) showed similar trends as cases 1 and 2 (figs. 19 to 21), but with smaller gains in wake loss for using WEC over not using WEC. Cases 3 and 4 also had wider spreads in the number of function calls required for SNOPT to converge. There are two likely contributors to this difference. The first is an increase in the number of wind directions and different wind speeds in each wind direction for cases 3 and 4. The second is that case 4 provides more space for the turbines in general, resulting in a flatter design space with more similar local optima. The increased wind resource complexity and flatter design space could both increase the number of function calls required. More benefit may also be available from WEC on these cases if WEC were tuned specifically to them. Because we tuned only to case 2, we may have missed some potential improvements available through WEC for cases 1, 3, and 4. However, because wake loss is relative to the energy available, the actual energy gains for using WEC in case 4 are greater than in any of the other cases except case 2 with Bastankhah. Even without tuning WEC to these cases specifically, the SNOPT with WEC results are significant, and only slightly overlapped, as compared to the SNOPT results without WEC (see figs. 18, 22 and 23). Using SNOPT+WEC also resulted in the best average for case 4 and the best overall and average for case 3, with SNOPT+WEC giving a result nearly equal to the best found by ALPSO on case 4.

The function call counts for most WEC optimizations were weighted towards the first and last optimizations. We found the first optimization to typically be responsible for around 20%–30% of the function calls, which makes sense because much of the turbine movement occurs in the first optimization. The seventh step includes both turbulence intensity (if applicable) and

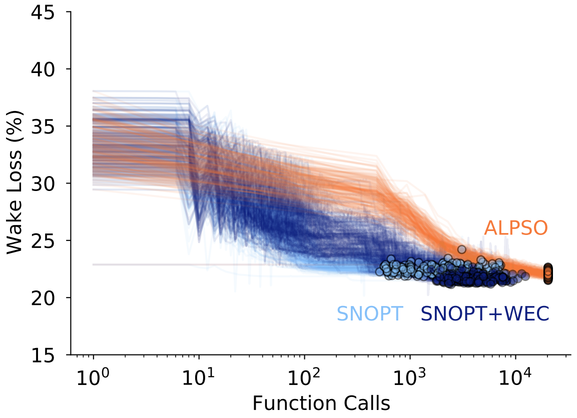


FIGURE 22 Convergence histories of optimizations for case 3 (38 turbines and 36 wind directions) using the Bastankhah model. Markers indicate optimized values.

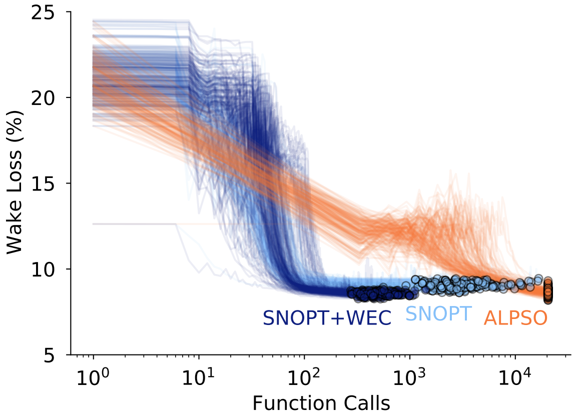


FIGURE 23 Convergence histories of optimizations for case 4 (60 turbines and 72 wind direct ions) using the Bastankhah model. Markers indicate optimized values.

a tighter convergence tolerance, so we saw a significant number of function calls for that step as well (about 20%–50%). The individual WEC step optimizations used relatively few function calls compared to the individual steps for SNOPT alone due to warm starting.

We compared wake loss and the number of function calls required for each of the cases discussed previously. We used function calls as a surrogate for time. We did not report wall time because we did not maintain enough consistency in the computational resources used for each optimization run (cores, processor types, computational isolation, etc). We also performed a Welch's t-test between the SNOPT and SNOPT+WEC wake loss percentage results, as well as between the ALPSO and ALPSO+WEC wake loss percentage results. The Welch's t-tests showed $p < 0.001$ for all cases, indicating a high confidence that the results are not a product of random chance but rather demonstrate actual improvement in the wake loss due to using WEC. The final statistical results of optimizations from the 200 starting points for each optimization method on the case studies presented in section 5 are shown in tables 5 to 9.

While mean and median are both reported for the final wake loss in each case, they were nearly equal for all cases. SNOPT+WEC resulted in the lowest average wake loss across all cases. The largest difference was for case 2, where SNOPT+WEC resulted in a reduction in wake loss percentage of 3.058 percentage points compared with the results from SNOPT alone.

The low average values of wake loss and lower standard deviations of wake loss for SNOPT with WEC, indicate that SNOPT with WEC is more accurate and reliable than SNOPT or ALPSO alone. This improvement in performance does come at the cost of more function calls than SNOPT without WEC. The increase in function calls when using WEC is expected because WEC runs seven optimizations to convergence (one for each value of ξ, and a final optimization to account for local turbulence intensity) while SNOPT alone runs only two (with and without local turbulence intensity). Because WEC results also have smaller standard deviations than SNOPT alone for all cases and ALPSO for all but one case, and the individual WEC step optimizations use

TABLE 5 Case 1 Results for Bastankhah Model: 16 Turbines, 20 Directions

	Function Calls			Wake Loss (%)[*]					
	Median	Low	High	Median	Mean	SD	Low	High	p
SNOPT	546	152	3348	11.898	11.882	1.470	8.630	16.988	
SNOPT+WEC	618	300	3580	9.169	8.860	0.698	7.346	12.137	< 0.001
ALPSO	20130	20130	20130	10.951	10.940	1.094	7.479	13.523	

[*] For case 1 with Bastankhah, 1% wake loss represents approximately 0.83 GWh

TABLE 6 Case 2 Results for Bastankhah Model: 38 Turbines, 12 Directions

	Function Calls			Wake Loss (%)*					
	Median	Low	High	Median	Mean	SD	Low	High	p
SNOPT	591	202	2220	16.304	16.338	0.790	14.505	19.102	
SNOPT+WEC	2679	814	10696	13.285	13.280	0.341	11.725	14.035	< 0.001
ALPSO	21030	21030	21030	15.076	15.097	0.555	13.658	17.016	
ALPSO+WEC	26460	26460	26460	14.097	14.096	0.425	12.532	15.762	< 0.001

* For case 2 with Bastankhah, 1% wake loss represents approximately 1.898 GWh

TABLE 7 Case 2 Results for Jensen Model: 38 Turbines, 12 Directions

	Function Calls			Wake Loss (%)*					
	Median	Low	High	Median	Mean	SD	Low	High	p
SNOPT	197	96	1646	19.167	19.184	0.442	18.077	20.328	
SNOPT+WEC	288	216	5988	18.244	18.256	0.216	17.742	18.840	< 0.001
ALPSO	21030	21030	21030	18.811	18.848	0.291	18.069	20.016	

* For case 2 with Jensen, 1% wake loss represents approximately 2.158 GWh

TABLE 8 Case 3 Results for Bastankhah Model: 38 Turbines, 36 Directions

	Function Calls			Wake Loss (%)*					
	Median	Low	High	Median	Mean	SD	Low	High	p
SNOPT	2004	514	7764	22.423	22.443	0.362	21.487	24.203	
SNOPT+WEC	3762	1584	12368	21.592	21.646	0.287	21.148	22.680	< 0.001
ALPSO	20280	20280	20280	22.062	22.075	0.231	21.510	22.667	

* For case 3 with Bastankhah, 1% wake loss represents approximately 0.619 GWh

relatively few function calls due to warm starting, fewer runs would be needed to gain the same level of confidence in the results. The reduction in overall runs could lead to large overall reductions in the number of function evaluations for design studies performed with WEC as compared to SNOPT or ALPSO alone, while simultaneously achieving more efficient final wind farm layouts.

The relatively wide spread and moderate results from SNOPT on most cases demonstrate one of the weaknesses of gradient-based algorithms. While gradient-based algorithms tend to need fewer function calls, and thus typically less time, they are highly

TABLE 9 Case 4 Results for Bastankhah Model: 60 Turbines, 72 Directions

	Function Calls			Wake Loss (%)*					
	Median	Low	High	Median	Mean	SD	Low	High	p
SNOPT	2714	560	16640	9.033	9.043	0.180	8.625	9.445	
SNOPT+WEC	471	276	1410	8.539	8.551	0.118	8.260	8.830	< 0.001
ALPSO	20430	20430	20430	8.632	8.647	0.185	8.183	9.314	

* For case 4 with Bastankhah, 1% wake loss represents approximately 3.992 GWh

susceptible to local optima. WEC was designed to help alleviate the problem of local optima in the WFLO problem. That WEC wake loss results, on average, were lower and less spread for all test cases, as compared with SNOPT and ALPSO, seems to indicate that WEC is at least partially overcoming the problem of local optima and providing significant benefits over the other optimization methods.

8 | CONCLUSION

The wake expansion continuation (WEC) method we proposed in this paper uses inherent characteristics of typical wake models to reduce the multi-modal nature of the wind farm layout optimization (WFLO) design space. WEC can help gradient-based optimization algorithms solve the global WFLO problem and also improve the results of gradient-free algorithms. We tested WEC with two wake models, two optimization algorithms (one gradient-based and one gradient-free) and four WFLO problems, one with 16 turbines and 20 wind directions, one with 38 turbines and 12 wind directions, one with 38 turbines and 36 wind directions, and one with 60 turbines and 72 wind directions. We found a statistically significant reduction ($p < 0.001$) in optimized wake loss using WEC compared to optimization without WEC for all test cases. We found that WEC reduced the average, minimum, and standard deviation of the wake loss for all cases. We found more improvement using WEC with the gradient-based optimization algorithm than with the gradient-free algorithm. While we still recommend using multiple optimization runs when using WEC, because of the lower standard deviation fewer runs should be needed to get results at least as good as those without WEC. Alternatively, because of the improved average, you can expect better results if you use the same number of runs.

Future work should investigate potential improvements and best practices for WEC to reduce the number of function calls required, provide a more complete comparison to gradient-free wind farm layout optimization including discrete parameterization. It may also be advantageous to consider using WEC with multiple wake models in series because of the rapid convergence of WEC with the simple Jensen cosine model. In such a study WEC with a gradient-based optimization algorithm would take the place of the gradient-free algorithm in previously studied hybrid gradient-free then gradient-based optimization studies. Studies investigating larger and more complex cases, with both general and specific WEC tuning along with more wake models, would also be informative. While similar methods have been applied on other applications through the use of surrogate models composed of Gaussian-basis functions, future work should also consider various other applications where the WEC approach of directly altering the underlying equations could be applied.

ACKNOWLEDGMENTS

This work was supported in part by a Utah NASA Space Grant Consortium Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Utah NASA Space Grant Consortium.

References

1. Rios Luis Miguel, Sahinidis Nikolaos V.. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization.* 2013;56(3):1247–1293.

2. Herbert-Acero Jf, Probst O., Rethore Pe, Larsen Gc, Castillo-Villar Kk. A Review of Methodological Approaches for the Design and Optimization of Wind Farms. *Energies.* 2014;7(11).

3. Fleming P., Ning A., Gebraad P., Dykes K.. Wind Plant System Engineering through Optimization of Layout and Yaw Control. *Wind Energy.* 2015;.

4. Guirguis David, Romero David A., Amon Cristina H.. Toward efficient optimization of wind farm layouts: Utilizing exact gradient information. *Applied Energy.* 2016;179(Supplement C):110 - 123.

5. Gebraad Pieter, Thomas Jared J., Ning Andrew, Fleming Paul, Dykes Katherine. Maximization of the Annual Energy Production of Wind Power Plants by Optimization of Layout and Yaw-Based Wake Control. *Wind Energy.* 2017;20(1):97–107.

6. Baker Nicholas F., Stanley Andrew P. J., Thomas Jared J., Ning Andrew, Dykes Katherine. Best Practices for Wake Model and Optimization Algorithm Selection in Wind Farm Layout Optimization. In: ; 2019; San Diego, CA.

7. Thomas Jared J., Annoni Jennifer, Fleming Paul, Ning Andrew. Comparison of Wind Farm Layout Optimization Results Using a Simple Wake Model and Gradient-Based Optimization to Large-Eddy Simulations. In: ; 2019; San Diego, CA.

8. Mosetti G., Poloni C., Diviacco B.. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics.* 1994;51(1):105 - 116.

9. Grady S.A., Hussaini M.Y., Abdullah M.M.. Placement of wind turbines using genetic algorithms. *Renewable Energy.* 2005;30(2):259 - 270.

10. Martínez Julián Eduardo González, Zaayer Michiel. Layout optimisation of offshore wind farms with realistic constraints and options. Master's thesisDelft University of Technology2014.

11. Stanley Andrew P. J., Ning Andrew, Dykes Katherine. Optimization of Turbine Design in Wind Farms with Multiple Hub Heights, Using Exact Analytic Gradients and Structural Constraints. *Wind Energy.* 2019;22(5):605–619.

12. Thomas Jared J., Ning Andrew. *A Method for Reducing Multi-Modality in the Wind Farm Layout Optimization Problem.* 2018.

13. Stanley Andrew P. J., Ning Andrew. Massive Simplification of the Wind Farm Layout Optimization Problem. *Wind Energy Science.* 2019;4(4):663–676.

14. Réthoré Pierre-Elouan, Fuglsang Peter, Larsen Gunner Chr., Buhl Thomas, Larsen Torben J., Aagaard Madsen Helge. TOPFARM: Multi-fidelity optimization of wind farms. *Wind Energy.* 2014;17(12):1797-1816.

15. Graf Peter, Dykes Katherine, Scott George, et al. Wind Farm Turbine Type and Placement Optimization. *Journal of Physics: Conference Series.* 2016;753(6):062004.

16. Mittal Prateek, Mitra Kishalay. Decomposition based multi-objective optimization to simultaneously determine the number and the optimum locations of wind turbines in a wind farm. *IFAC-PapersOnLine.* 2017;50(1):159 - 164. 20th IFAC World Congress.

17. Mobahi Hossein, III John Fisher. A Theoretical Analysis of Optimization by Gaussian Continuation. In: ; 2015.

18. Bastankhah Majid, Porté-Agel Fernando. Experimental and theoretical study of wind turbine wakes in yawed conditions. *Journal of Fluid Mechanics.* 2016;806:506-541.

19. Jensen N. O.. *A Note on Wind Generator Interaction.* : Risø National LaboratoryDK-4000 Roskilde, Denmark; 1983.

20. Niayifar Amin, Porté-Agel Fernando. Analytical Modeling of Wind Farms: A New Approach for Power Prediction. *Energies.* 2016;9(9):1-13.

21. Katic I, Højstrup J, Jensen N.O.. A Simple Model for Cluster Efficiency. In: European Wind Energy Association; 1986; Rome - Italy.

22. Western Regional Climate Center . Station Wind Rose: Nantucket https://wrcc.dri.edu/cgi-bin/wea_windrose.pl?laKACK2012.

23. NoordzeeWind . Reports: Environment (Meteorological Data) https://www.noordzeewind.nl/en_nl/knowledge/weather-data.html2006.

24. Gray Justin, Moore Kenneth T., Naylor Bret A.. OpenMDAO: An Open Source Framework for Multidisciplinary Analysis and Optimization. *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference.* 2010;.

25. Gill P., Murray W., Saudners M.. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Review.* 2005;47:99-131.

26. Jansen P.W., Perez R.E.. Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach. *Computers & Structures.* 2011;89(13):1352 - 1366.

27. Perez Ruben E., Jansen Peter W., Martins Joaquim R. R. A.. pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization. *Structural and Multidisciplinary Optimization.* 2012;45(1):101–118.

28. Hascoët L., Pascual V.. The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. *ACM Transactions On Mathematical Software.* 2013;39(3).

29. Wu Neil, Kenway Gaetan, Mader Charles A., Jasa John, Martins Joaquim R. R. A.. pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software.* 2020;5(54):2564.

□