

Wake Expansion Continuation: Multi-modality reduction in the wind farm layout optimization problem

J J Thomas, S McOmber, and A Ning

Department of Mechanical Engineering, Brigham Young University, Provo, Utah, USA

E-mail: jaredthomas@byu.edu

Abstract. This paper presents a process using an approach related to continuation optimization methods for reducing multi-modality in the wind farm layout optimization problem, referred to as Wake Expansion Continuation (WEC). The reduction in multi-modality is achieved by starting with an increased wake spread, while maintaining normal velocity deficits at the center of the wakes, and then reducing the wake spread for each of a series of optimization runs until the standard wake spread is used. WEC was applied and shown effective with two different wake models. Four optimization case studies were tested with a gradient-based method, a gradient-free method. Results using WEC show a significant mean optimized annual energy production (AEP) improvement compared to optimization without WEC for all test cases. The gradient-free optimization algorithm was also outperformed on average by the gradient-based algorithm with WEC on all cases. WEC was also applied to the gradient-free algorithm for one case study with significantly improved results, but the improvement was more significant when WEC was applied to a gradient-based algorithm.

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, gradient-free methods have been shown to have reduced performance with 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. Gradient-based methods are not widely used for WFLO problems, but are gaining interest due to their relatively low computational cost and their ability to handle many variables and constraints. However, gradient-based methods are highly susceptible to local optima [2]. Despite this weakness, they have been shown to find good solutions to WFLO problems [3, 4, 5].

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 [6, 7]. 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 [8]. This approach reduces the sensitivity of gradient-based optimization methods to local optima. Reparameterization 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. as done in [9]. Hybrid approaches combine gradient-based and gradient-free algorithms iteratively [10, 11, 12], and, depending on the problem size, can yield result quality comparable to multi-start approaches [10]. While each of these techniques yield improved results, there is still need for 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. These limitations indicate that there is need for better optimization methods that avoid local optima, whether real or imposed by the approach itself.

Current methods seek to avoid local optima by searching the existing design space more broadly. This approach becomes intractable for large optimization problems with many variables and constraints. To address this issue, we propose a process for use with gradient-based optimization algorithms that temporarily reduces the number and magnitude of local optima using a continuation optimization approach such as the method discussed in [13]. However, while continuation optimization uses a numerical approximation of the design space, our proposed process takes advantages of the physical properties of the design space directly.

The two primary characteristics that simple wake models seek to capture are wake diameter and velocity deficit. These characteristics in turn largely determine the shape of the wind farm layout design space. 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.

During optimization, the spaces between wakes translate to locally optimal locations for turbines. However, it is often the case that there are more optimal turbine locations that are not found by the optimizer because the turbines are effectively stuck in locally optimal locations. In this paper we propose a method designed to overcome the problem of local optima by temporarily reducing the multi-modality of the design space. We will refer to the new method as Wake Expansion Continuation, or WEC.

In the following sections we will (1) present an overview of WEC, (2) introduce the simulation models used to study WEC, (3) demonstrate how to apply WEC to existing models, (4) provide a series of case studies for optimization method comparison, (5) give some details on the computational environment used in our studies, (6) discuss how we tuned the various methods for our case studies, (7) present and discuss the results of the case studies, and (8) provide concluding comments.

2. Introduction to the Wake Expansion Continuation

The WEC method is somewhat analogous to Gaussian continuation optimization. In Gaussian continuation optimization, the design space is approximated using a series of Gaussian radial basis functions. 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 [13]. 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 convex 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 any nearby local optima. We have presented a preliminary study of the WEC method showing significant benefit in [14] and a follow-on study showing that optimization results using WEC were even more significant when re-calculated using large eddy simulations (LES) in [18].

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 function in WEC are not radial, and the changing locations of the Gaussian functions during the WFLO mean that we cannot guarantee convergence to the global optimum. However, the use of the WEC method does significantly improve the optimization results.

The WEC method can be explained in three basic steps:

- 1) Determine how the wake diameter and wake deficit are controlled for the selected wake model.
- 2) If necessary, add 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. Avoiding altering the wake deficit in the center of the wake mimics the behavior of increasing the standard deviation for a Gaussian distribution as done in [13].
- 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 spread is no longer altered from the original model. Each optimization after the first should be hot-started using the result of the previous optimization.

By spreading the wakes, we can effectively fill in the gaps between wakes so that a gradient-based optimization algorithm can bypass local optima and proceed to a better solution. Another way of thinking about this is that by spreading the wakes, more turbines see the influence of each wake, which provides more information to the gradient-based optimization algorithm about the design space through the Jacobian.

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 spreading the wake: (1) increasing the wake spreading angle (WEC-A), (2) multiplying the initial wake diameter (WEC-D), and (3) a hybrid of 2 and 3 that is accomplished by multiplying the wake diameter at an estimated point of far wake onset and allowing the wake to follow an angled line from the initial wake diameter to the expanded far wake diameter (WEC-H). The impact on the wake shape of each of these three methods of expanding the wake are shown in fig. 1.

3. Simulation Models

Because of the general nature of the proposed method, it could theoretically be applied to a range of wake models. In this study we have applied WEC to the 2016 version of the Bastankhah and Porté-Agel wake model [15] and the cosine version of the Jensen model [16]. We also used various other models necessary for combining wakes and calculating wind farm AEP.

3.1. The Bastankhah and Porté-Agel Wake Model

In the Bastankhah and Porté-Agel wake model, the 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 and Porté-Agel wake model, as defined in eq. (1) [15], along with the Niayifar and Porté-Agel wind farm model [17].

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

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, $y - \delta$ and $z - z_h$ are the distances of the point of interest from the wake center in the cross-stream horizontal and vertical directions respectively,

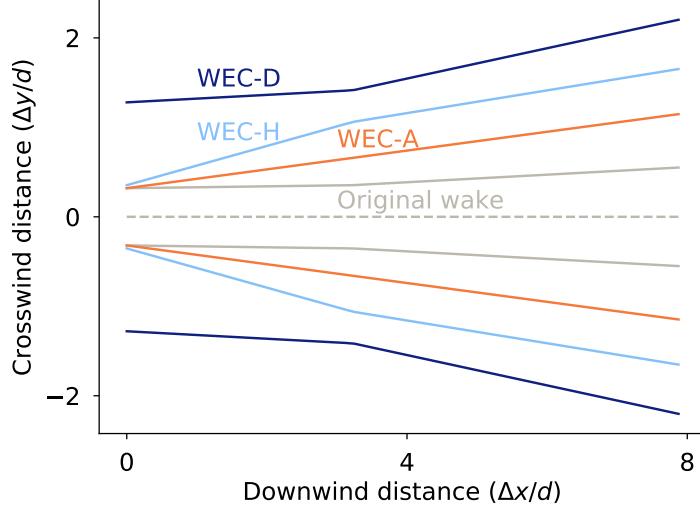


Figure 1. The impact on wake shape of expanding the wake by increasing the spreading angle (WEC-A), the diameter (WEC-D), or a downstream diameter that also changes the near wake spreading angle (WEC-H). The relative amount of expansion in the figure is for convenience in comparing the WEC methods. The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

and σ_y and σ_z are the standard deviations of the wake deficit in the cross-stream horizontal and vertical directions as defined in eqs. (2) and (3).

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

$$\sigma_z = k_z(\Delta x - x_0) + \frac{d}{\sqrt{8}} \quad (3)$$

In eqs. (2) and (3), 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)[17].

$$k^* = 0.3837I + 0.003678 \quad (4)$$

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. However, a smooth version of the local turbulence intensity was used in a final optimization step following all WEC steps in studies performed using WEC with the Bastankhah and Porté-Agel 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 and Porté-Agel 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 in [18] and reproduced in eq. (5).

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

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

$$\sigma_{yd} = k_y(x_d - x_0) + \frac{D_r \cos \gamma}{\sqrt{8}} \quad (6)$$

We then use the definition of the wake spread at the point of discontinuity to provide an estimate for the wake spread and velocity deficit at the rotor hub. This is an important assumption because we need to have some slope in the wake spread between the rotor hub and the point of far wake onset for more effective gradient-based optimization. With the assumption that σ_{yd} is the value of the wake spread at the rotor hub, we can define the slope of the near wake, k_{yNEAR} , as shown in eq. (7).

$$k_{yNEAR} = \frac{\sigma_{yo} - \sigma_{yd}}{x_o} \quad (7)$$

For more details on near wake approximation of the Bastankhah and Porté-Agel wake model used in this study, please see [18].

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 cross wake deficit shape to follow a cosine curve. We used the Jensen cosine wake model as shown in eq. (8), along with the Katic wake combination model [19].

$$\frac{\bar{u}}{\bar{u}_\infty} = 1 - 2a \left(\frac{r_0}{r_0 + \alpha \Delta x} \right)^2 f_\theta \quad (8)$$

In eq. (8), \bar{u} represents the wind velocity at a distance Δx downstream of the waking turbine, \bar{u}_∞ represents the free-stream wind velocity, r_0 represents the radius of the wind turbine, α represents the wake entrainment constant [16], a represents the axial induction (for which we assumed an ideal value of $a = 1/3$), and f_θ is the cosine factor defined in eq. (9).

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

In eq. (9), θ is the angle from the wake center line to the point of interest measure at the wake vertex a distance z upstream of the wind turbine, n represents an adjustment factor derived from the wake spreading angle β , as shown in fig. 2. The value for n can be calculated as in eq. (10), where β is in radians.

$$n = \pi/\beta \quad (10)$$

The wake spreading angle used in this work, and by Jensen, was $\beta = 20^\circ$ [16]. See fig. 2 for a diagram showing the overhead geometry of the Jensen Cosine model.

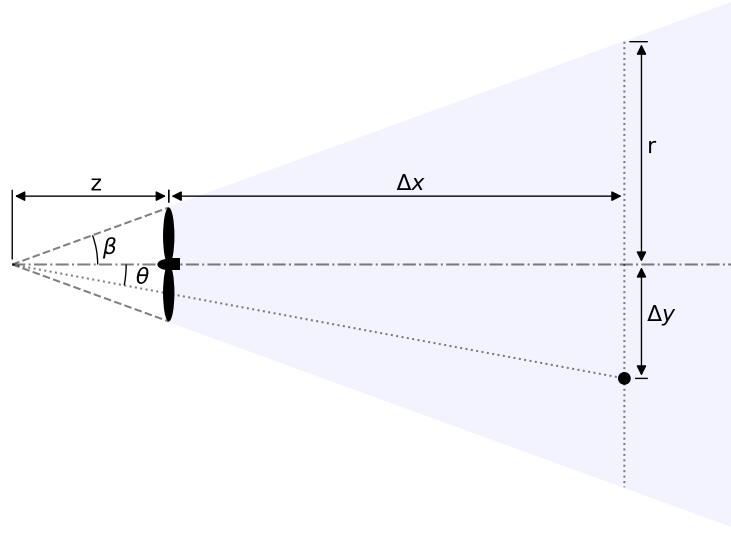


Figure 2. Diagram showing the overhead geometry of the Jensen cosine wake model. The wind is blowing to the right. The dashed line down the middle represents the center line of the wake. The large black dot represents any given point of interest.

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 in [17] 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 and Porté-Agel wake model. We used a constant axial induction value of 1/3 when using the Jensen cosine wake model.

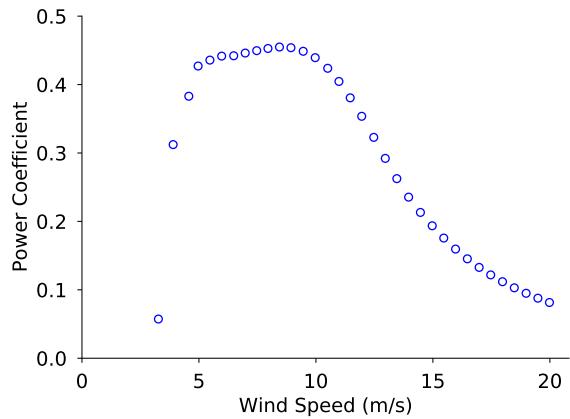


Figure 3. C_P curve for the Vestas V80 2MW wind turbine [17]

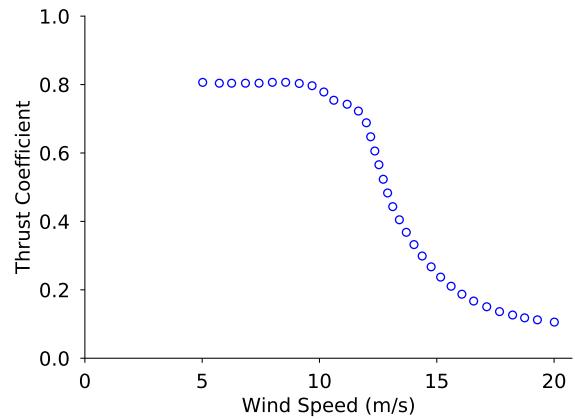


Figure 4. C_T curve for the Vestas V80 2MW wind turbine [17]

3.4. Other Models

We combined the wake deficits using a linear combination method as discussed in [17] with the Bastankhah and Porté-Agel wake model, and a sum of squares method with the Jensen wake

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

model as discussed in [19]. We used a reference height of 80 m for all wind speed measurements and adjusted to different heights with a power law and a wind shear exponent of 0.31 as shown in eq. (11).

$$u = u_r \left[\frac{z}{z_r} \right]^\psi \quad (11)$$

In Equation (11), 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 sample 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 and Porté-Agel wake model, but in no particular order for the Jensen cosine model. The power output of each turbine was then calculated based on eq. (12)

$$P_i = \frac{1}{2} \rho A_{r,i} C_P U_i^3 \quad (12)$$

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 shown in eq. (13).

$$AEP = (24)(365) \sum_{j=1}^{n_d} \left(p_j \sum_{i=1}^{n_t} P_i \right) \quad (13)$$

In eq. (13), n_d is the number of wind directions, or states, p_j is the probability of wind for a given state (direction and speed), and n_t is the number of wind turbines.

To achieve an unbiased baseline for results comparisons, we calculated wake loss (energy lost due to wake effects) across all the optimization results. Wake loss percentage was calculated as shown in eq. (14).

$$L = 100 \left(1 - \frac{AEP_0}{AEP_t} \right) \quad (14)$$

In eq. (14), 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. (15).

$$AEP_t = (24)(365) \sum_{j=1}^{n_d} \left(p_j P \right) \quad (15)$$

In eq. (15), P is the power of a single un-waked wind turbine calculated using eq. (12).

4. Applying WEC to the Wake Models

In this section we apply WEC, step by step as discussed in section 2, to the Bastankhah and Porté-Agel wake model and the Jensen cosine wake model. To test the wake spreading approaches discussed, we implemented each of them in the Bastankhah and Porté-Agel wake model. Once we understood which approach is most effective, we implemented and tested only that approach in the Jensen cosine model.

4.1. Applying WEC to the Bastankhah and Porté-Agel Wake Model

4.1.1. Bastankhah and Porté-Agel WEC Step (1) The first parenthetical term of eq. (1) defines the magnitude of the velocity deficit. The exponential terms determine the wake spread. The wake spread and velocity deficit are coupled through σ_y and σ_z . However, because the spread and magnitude are expressed in separate terms, it is possible to adjust one without impacting the other.

4.1.2. Bastankhah and Porté-Agel WEC Step (2) - WEC-D In the diameter spreading method (WEC-D), independent control of the wake diameter is obtained by applying a factor, ξ , to σ_y and σ_z inside the exponential terms of eq. (1), as shown in eq. (16).

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left(1 - \sqrt{1 - \frac{C_T \cos \gamma}{8\sigma_y \sigma_z / d^2}} \right) \exp \left(-0.5 \left(\frac{y - \delta}{\xi \sigma_y} \right)^2 \right) \exp \left(-0.5 \left(\frac{z - z_h}{\xi \sigma_z} \right)^2 \right) \quad (16)$$

By increasing the value of ξ , we can widen the wakes without changing the magnitude of the velocity deficit in the center of the wakes. As the wakes widen, they mix and smooth out the local optima, as shown in fig. 6.

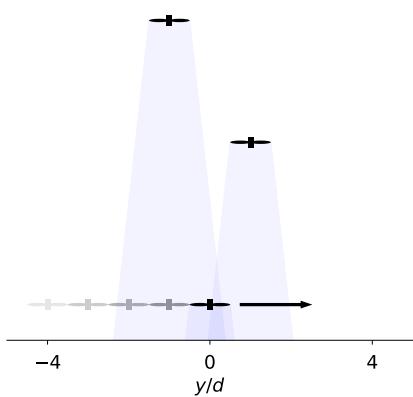


Figure 5. Simple wind farm, seen from above, used to demonstrate the effects of the WEC factors, θ_ξ and ξ , on the wind farm layout design space (see fig. 6). Wind is from the top.

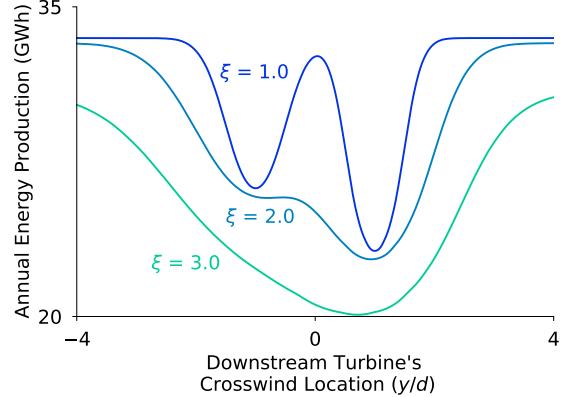


Figure 6. The impact of the relaxation factor, ξ , for WEC-D on a simple wind farm layout design space with one movable turbine and one wind direction (see fig. 5) using the Bastankhah and Porté-Agel wake model. The local optimum between the wakes disappears as the WEC factor increases.

4.1.3. Bastankhah and Porté-Agel WEC Step (2) - WEC-A To gain direct control of wake spreading angle, we arranged the equations in such a way that the user specifies the wake spreading angle, rather than using a multiple of the default angle.

To gain direct control of the wake spreading angle, we calculate the wake spreading slope that would correspond to the desired spreading angle (θ_ξ) as

$$k_{y_\xi} = \tan \theta_\xi \quad (17)$$

To get a final version, where a user can specify a desired spreading angle, we compare this against the original formula for $k_{y_{NEAR}}$ to obtain

$$k_{y_\xi} = \max(\tan \theta_\xi, k_{y_{NEAR}}) \quad (18)$$

We then use k_{y_ξ} to re-define the wake spread at the point of far wake onset, σ_{yo_ξ} , as shown in eq. (19).

$$\sigma_{yo_\xi} = k_{y_\xi} x_o + \sigma_{yd} \quad (19)$$

The definition for σ_{yo_ξ} allows us to define the general form of the wake spread for all wake regions as shown in eq. (20).

$$\sigma_{y_\xi} = \begin{cases} k_y(x - x_o) + \sigma_{yo_\xi} & x \geq x_o, k_y \geq k_{y_\xi} \\ k_{y_\xi}(x - x_o) + \sigma_{yo_\xi} & x \geq x_o, k_y < k_{y_\xi} \\ k_{y_\xi}x + \sigma_{yd} & x < x_o \end{cases} \quad (20)$$

The same process can also be applied to obtain σ_{z_ξ} .

Now that we have direct control of the wake spreading angle, for angles greater than the default angle, we can apply the results directly to the exponential terms of eq. (1) to obtain a model that allows the user to specify a spreading angle without impacting the velocity deficit in the wake center as shown in eq. (21).

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left(1 - \sqrt{1 - \frac{C_T \cos \gamma}{8\sigma_y \sigma_z / d^2}} \right) \exp \left(-0.5 \left(\frac{y - \delta}{\sigma_{y_\xi}} \right)^2 \right) \exp \left(-0.5 \left(\frac{z - z_h}{\sigma_{z_\xi}} \right)^2 \right) \quad (21)$$

For WEC-A, The impact of θ_ξ is much more pronounced for turbines that are further upstream due to the increased wake growth rate, but are otherwise similar to the effects shown for WEC-D in fig. 6.

4.1.4. Bastankhah and Porté-Agel WEC Step (2) - WEC-H The third approach, WEC-H, is a sort of hybrid of the other two approaches of widening the wake. The wake diameter is multiplied in the far wake, but the near wake expands linearly until the pre-selected point where the wake diameter multiplier is applied. The basic theory for applying WECH to the Bastankhah and Porté-Agel model is shown in eqs. (22) to (24). The first step, eq. (22), is based on the original wake diameter calculation as shown in eq. (2).

$$\sigma_{yo} = d \frac{\cos(\gamma)}{\sqrt{(8)}} \quad (22)$$

$$\sigma_{yo,\xi} = \xi \sigma_{yo} \quad (23)$$

$$\sigma_{y,\xi} = \begin{cases} \xi(k_y(x - x_0) + d \frac{\cos(\gamma)}{\sqrt{(8)})} & x > x_0 \\ \sigma_0 + x \frac{\sigma_{yo,\xi} - \sigma_0}{x_0} & x \leq x_0 \end{cases} \quad (24)$$

Where σ_{yo} is the original wake spread at x_0 , the onset of the far wake, $\sigma_{yo,\xi}$ is the wake spread at the onset of far wake after the application of the expansion factor, and $\sigma_{y,\xi}$ is the horizontal wake

standard deviation to be used in the exponential terms of the Bastankhah and Porté-Agel model as shown in eq. (25). A similar approach can be used to fine $\sigma_{z,\xi}$.

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left(1 - \sqrt{1 - \frac{C_T \cos \gamma}{8\sigma_y \sigma_z / d^2}} \right) \exp \left(-0.5 \left(\frac{y - \delta}{\sigma_{y,\xi}} \right)^2 \right) \exp \left(-0.5 \left(\frac{z - z_h}{\sigma_{z,\xi}} \right)^2 \right) \quad (25)$$

Note that the $\sigma_{y,\xi}$ term is only applied to the wake width terms and the original model values are used in calculating the magnitude of the center-line wake deficit. As in WEC-A and WEC-D, increasing the value of the WEC factor, ξ , allows us to widen the wake without changing the magnitude of the velocity deficit in the center of the wake. As the wakes widen, they mix and smooth out the local optima. The effects are similar to those shown for WEC-D in fig. 6.



4.1.5. Bastankhah and Porté-Agel WEC Step (3) To gain the benefits of WEC without losing the accuracy of the wake model, the optimization must be run multiple times, in a continuation approach, with values of ξ or θ_ξ decreasing to 1 or 0 respectively. The results from optimizing with each value of ξ or θ_ξ can then be used to warm start the next optimization. In this way, the gradient-based optimization can intelligently explore the design space and then refine the model to find a final, more accurate, result. The iterative optimizations are generally fairly fast due to starting from the previous optimized solution. In section 7, we will show how we determined which WEC method to use and the corresponding values of ξ or θ_ξ to use in practice.

4.2. Applying WEC to the Jensen Cosine Wake Model

In this section we apply WEC-D to the Jensen Cosine wake model. A diagram displaying the Jensen cosine wake model from an overhead perspective is shown in fig. 2. 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 spread for the Jensen Cosine wake model. By breaking down the cosine factor into its constituent parameters, it is possible to adjust the wake spread 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 determined using eq. (26).

$$\theta = \tan^{-1} \left(\frac{\Delta y}{\Delta x + z} \right) \quad (26)$$

In eq. (26), Δx and Δy represent the downwind and crosswind spacing between the upwind turbine and the downwind 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. An expression for z based on the initial wake diameter can be found below in eq. (27).

$$z = \frac{r_0}{\tan(\beta)} \quad (27)$$

4.2.2. Jensen Cosine WEC Step (2) We can directly adjust the wake spread without impacting the magnitude of the deficit in the center of the wake by applying a relaxation factor, ξ , to the rotor radius, r_0 , in eq. (27), as seen below in eq. (28).

$$z = \frac{\xi r_0}{\tan(\beta)} \quad (28)$$

Through a series of substitutions, this new vertex distance, eq. (28), may be combined with the Jensen cosine wake model to obtain eq. (29).

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = 1 - 2a \left[\left(\frac{r_0}{r_0 + \alpha x} \right) \right]^2 \frac{1}{2} \left(1 + \cos \left[\frac{\pi}{\beta} \tan^{-1} \left(\frac{\Delta y}{\Delta x + \frac{\xi r_0}{\tan(\beta)}} \right) \right] \right) \quad (29)$$

Because we have inserted the relaxation factor, ξ , into eq. (29), we may adjust the wake spread 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. (29) goes to zero, regardless of the value of ξ . Because this is the only place where the relaxation factor is found in eq. (29), this means that the velocity deficit will be constant with respect to ξ at the wake center. This behavior is shown in figs. 7 and 8, which also shows that local optima can be smoothed out through WEC for the Jensen Cosine model.

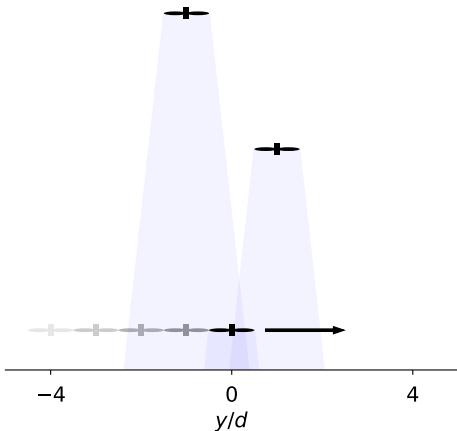


Figure 7. Simple wind farm, seen from above, used to demonstrate the effects of the WEC factor ξ , on the wind farm layout design space (see fig. 8). Wind is from the top.

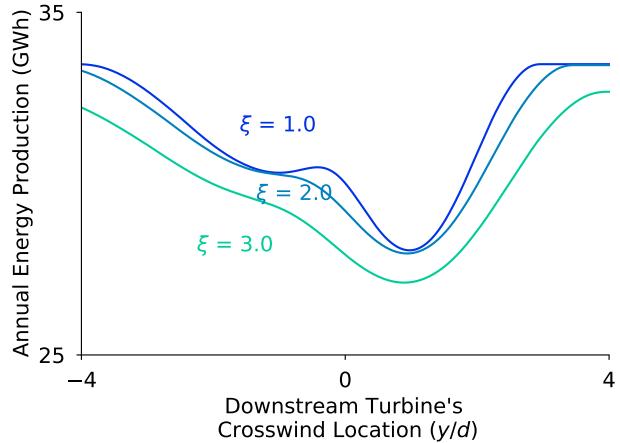


Figure 8. The impact of changing the wake spread angle, ξ , for WEC on a simple wind farm layout design space with one movable turbine and one wind direction (see fig. 7) using the Jensen cosine wake model. The local optimum between the wakes disappears as the WEC factor increases.

4.2.3. Jensen Cosine WEC Step (3) 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.

5. Case Studies

To demonstrate the effectiveness of WEC on a range of problems, we present three wind farm optimization case studies. For each case, we used the Vestas V-80 2MW wind turbine. The cases were chosen to represent a range in size, complexity, and difficulty of the optimization problem.

We created 199 pseudo-random starting wind farm layouts and one planned layout for each case for a total of 200. Each of the starting layouts had all the turbines inside the wind farm boundary constraint and did not have any turbines spaced less than one rotor diameter apart.

5.1. Case 1: 16 Turbines and 20 Wind Directions

Case 1 was carefully selected 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. 9). 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. 10).

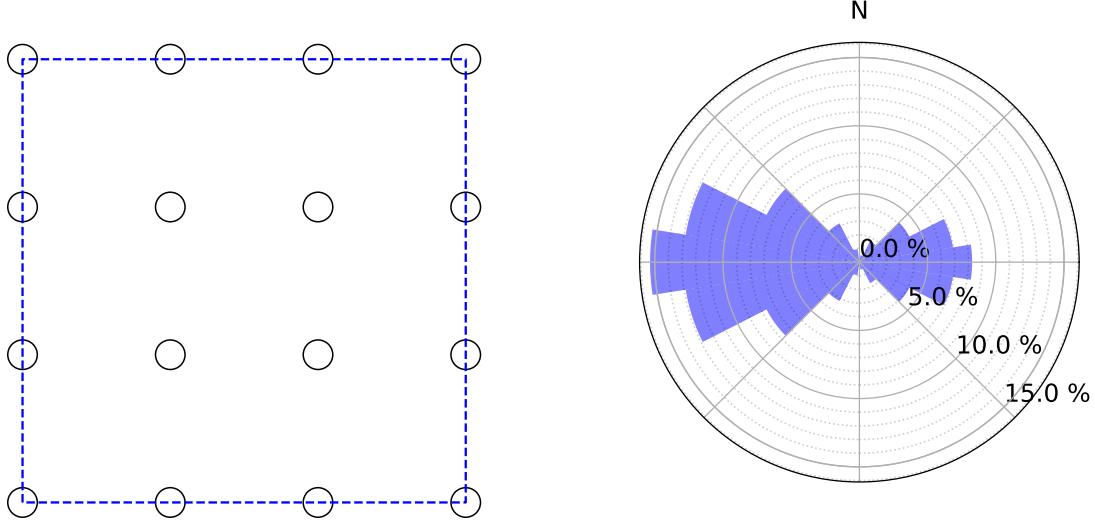


Figure 9. Baseline wind farm layout for case 1. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

Figure 10. 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 shown in eq. (30)

$$\begin{aligned}
 & \underset{x_i, y_i}{\text{maximize}} \quad AEP(x_i, y_i,) \quad i = 1 \dots 16 \\
 & \text{subject to} \quad s_{i,j} \geq 2d \quad i, j = 1 \dots 16, \quad i \neq j \\
 & \quad x_{min} \leq x_i \leq x_{max} \quad i = 1 \dots 16 \\
 & \quad y_{min} \leq y_i \leq y_{max} \quad i = 1 \dots 16
 \end{aligned} \tag{30}$$

In eq. (30), (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

Case 2 was created to be significantly more challenging than the first case, but still fairly simple. We defined a wind farm with 38 wind turbines and a circular boundary as shown in fig. 11. The

size of the boundary allowed for at least a five-diameter 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. 12.

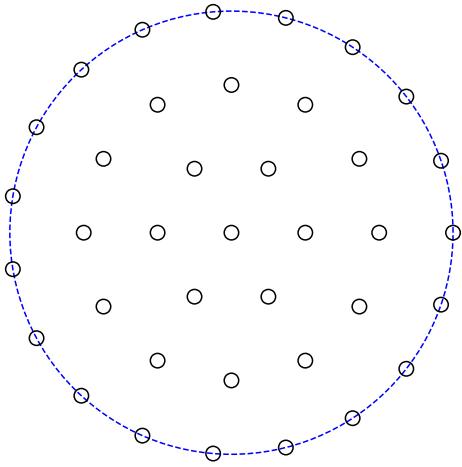


Figure 11. Baseline wind farm layout for cases 2 and 3. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

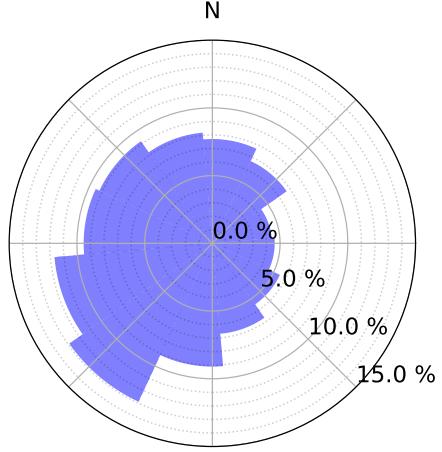


Figure 12. Directional probability wind rose for case 2. This is the Nantucket wind rose binned into 12 directions [20].

The optimization problem for case 2 was formulated as

$$\begin{aligned} & \underset{x_i, y_i}{\text{maximize}} \quad AEP(x_i, y_i,) \quad i = 1 \dots 38 \\ & \text{subject to} \quad s_{i,j} \geq 2d \quad i, j = 1 \dots 38 \quad i \neq j \\ & \quad [x_c - x_i]^2 + [y_c - y_i]^2 \leq r_b^2 \quad i = 1 \dots 38 \end{aligned} \quad (31)$$

In eq. (31), (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. 13 and 14. The optimization problem for case 3 was formulated as shown in eq. (31) with the same number of variables and constraints.

5.4. Case 4: 60 Turbines and 72 Wind Directions

Case 4, based on the Princess Amalia Wind Park, was selected 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 a convex polygonal boundary fig. 15. We used wind data binned into 72 wind directions with wind speed for each direction determined as the average of all samples in that sector as shown in figs. 16 and 17

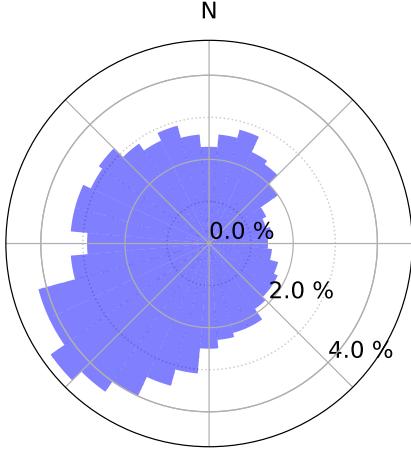


Figure 13. Directional probability wind rose for case 2. This is the Nantucket wind rose binned into 36 directions [20].

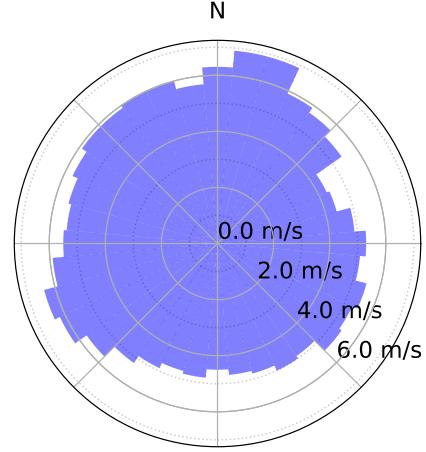


Figure 14. Average speed wind rose for case 2. This is the Nantucket wind rose binned into 36 directions [20]

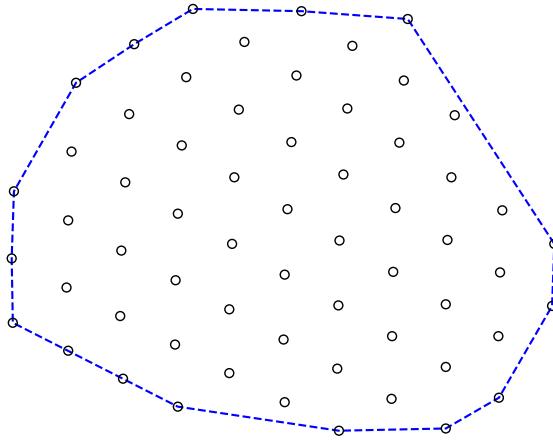


Figure 15. Baseline wind farm layout for case 4. The circles marking turbine locations are to scale, with diameters equal to the rotor diameter.

For case 4, the optimization problem was formulated as shown in eq. (32)

$$\begin{aligned}
 & \underset{x_i, y_i}{\text{maximize}} \quad AEP(x_i, y_i,) \quad i = 1 \dots 60 \\
 & \text{subject to} \quad s_{i,j} \geq 2d \quad i, j = 1 \dots 60, \quad i \neq j \\
 & \quad b_{i,k} \geq 0 \quad i = 1 \dots 60, k = 1 \dots 14
 \end{aligned} \tag{32}$$

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.

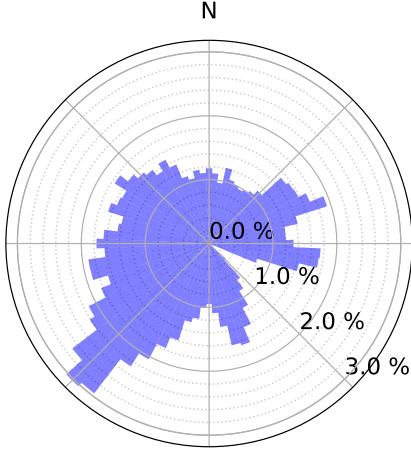


Figure 16. Direction probability wind rose for case 4. The wind rose is comprised of data from [21]. Measurements were taken from July 1, 2005, to June 30, 2006.

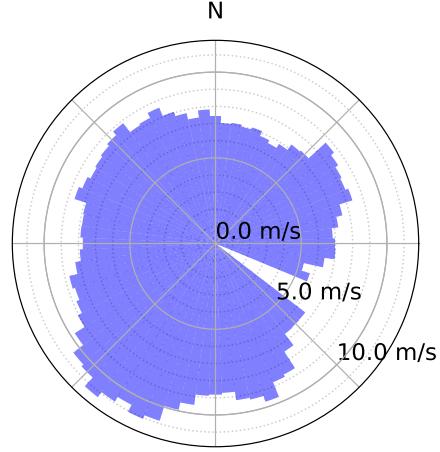


Figure 17. Average speed wind rose for case 4. The wind rose is comprised of data from [21]. Measurements were taken from July 1, 2005, to June 30, 2006.

6. Implementation Details and Optimization Algorithms

The code for each wake model was written in Fortran and wrapped in python. In this work, optimization problems were set up and solved in OpenMDAO [22] using two optimization algorithms, the gradient-based Sparse Nonlinear OPTimizer (SNOPT) [23] and the gradient-free Augmented Lagrangian Particle Swarm Optimizer (ALPSO) [24] **that uses an augmented Lagrangian method to handle constraints**. Both algorithms were used through pyOptSparse [25]. We obtained exact gradients for both wake models using algorithmic differentiation provided by Tapenade [26].

7. Tuning the Optimization Methods

7.1. SNOPT

We used two optimizations for each run with SNOPT when using the Bastankhah and Porté-Agel model. The first did not use local turbulence intensity, the second one did. SNOPT was run with different convergence tolerances for each case. **Tolerances were determined based on the tolerance required to achieve the majority of the improvement in several test runs.** Convergence tolerances used in each case are presented in table 2. The objective and constraint derivatives for all cases were scaled to be between ± 1 . The objective function was formulated in kWh and was subsequently scaled by **1E-4** for optimization with SNOPT.

7.2. ALPSO

ALPSO was run using the default parameters provided in pyOptSparse [27] with a few exceptions. First, the **craziness velocity** was set to be **1E - 2** as this value resulted in increased **performance** compared to the default value across all the cases we tested. We tried adjusting the initial particle velocity, but found no difference in results for the cases tested. As demonstrated in [24], 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



Table 2. SNOPT Tolerances And Scaling for Each Case Study

	Bastankhah and Porté-Agel		Jensen Cosine
	Without Local TI	With Local TI	
Case 1	1E-2	1E-3	
Case 2	9E-3	1E-3	1E-3
Case 3	9E-3	1E-3	
Case 4	1E-3	1E-4	

about 20000 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 determined based on the number inner iterations and desired function call cap for each case. The ALPSO convergence history for each case study with varied inner iteration counts is shown in figs. 18 to 21. The objective and design variables were all scaled to be between -1 and 1. All optimization runs using ALPSO had a population size of 30. The varied meta-parameters used for the various ALPSO runs are shown in table 3.

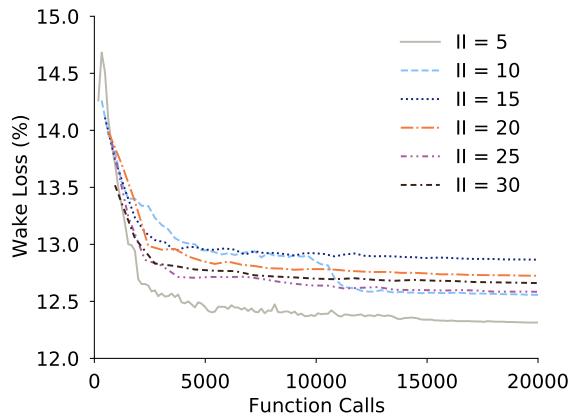


Figure 18. Comparing ALPSO convergence history for 16 turbines, 20 directions, and varying numbers of inner iterations.

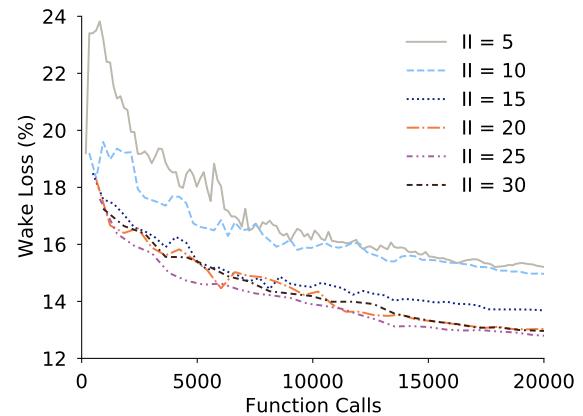


Figure 19. Comparing ALPSO convergence history for 38 turbines, 12 directions, and varying numbers of inner iterations.

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 to running 30 optimizations using a gradient-based algorithm in population size, we decided to run a full set

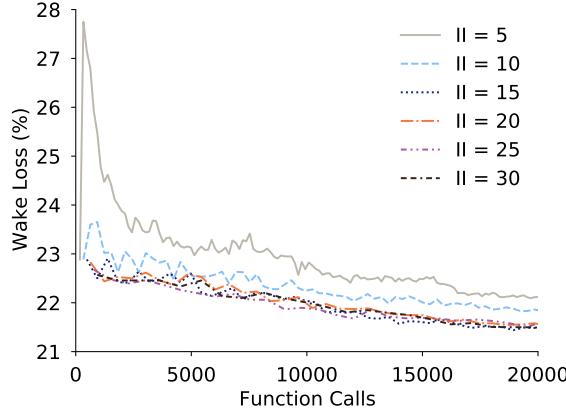


Figure 20. Comparing ALPSO convergence history for 38 turbines, 36 directions, and varying numbers of inner iterations.

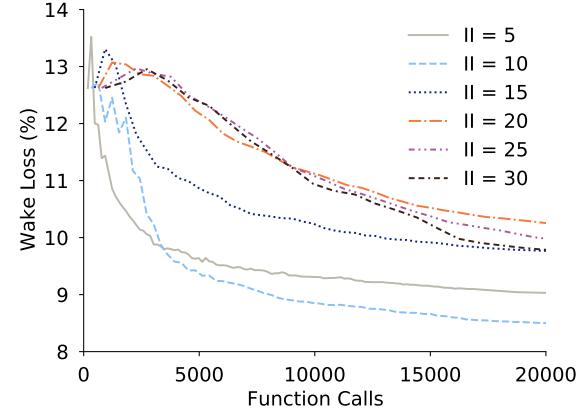


Figure 21. Comparing ALPSO convergence history for 60 turbines, 72 directions, and varying numbers of inner iterations.

of 200 optimizations for each test, just as was done for SNOPT. This decision was to reflect how optimizations are often performed in practice, with many optimizations being run regardless of the algorithm being used. 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 enables a direct comparison of function calls and the optimization objective.

7.3. SNOPT+WEC

We compared the performance of WEC-A, WEC-D, and WEC-H with the Bastankhah and Porté-Agel model by varying both the number of intermediate optimizations (steps) to run and the maximum values of the WEC factor (ξ or θ_ξ) to use. To investigate the WEC methods and parameter selection, we used Case 2 as discussed in section 5.2. 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 $\theta_\xi = 9^\circ$, $\xi = 3$, and $\xi = 3$, for WEC-A, WEC-D, and WEC-H respectively. All 200 starting layouts were tested with each parameter set. For comparing the wake expansion methods, the k^* value in eq. (4) was not adjusted to the local turbulence intensity during optimization, but was adjusted for calculating the results shown.

The mean wake loss results for the various wake spreading approaches are provided in figs. 22 and 23. Here we see that, on average, WEC-D results in less wake loss with a WEC value of 3 and using 5 or more steps, while WEC-A has its lowest average wake loss for a maximum WEC angle of 6° , and WEC-H minimizes average wake loss most with lower WEC values. On average, wake loss minimized using WEC-A or WEC-H does not appear to be impacted by changing the number of steps.

We found that the minimum wake loss is achieved when the maximum WEC angle is 9° for WEC-A and when the maximum WEC value is 3 for WEC-D and WEC-H. There does not appear to be a clear relationship between minimum wake loss and number of steps for WEC-A or WEC-H. WEC step numbers greater than or equal to 3 achieve approximately equal minimum wake loss for WEC-D and WEC-H individually, but on average WEC-D performs much better than either WEC-A or WEC-H.

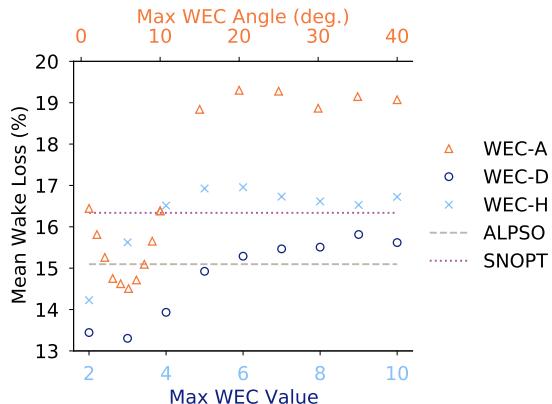


Figure 22. Mean wake loss results for varying the maximum WEC expansion parameter while holding the number of steps constant at six. Each data point represents 200 separate optimizations with different starting points. Note: the WEC value and WEC angle axes are not directly comparable.

The lowest standard deviations occur when the WEC angle is 5° for WEC-A, the WEC value is 3 for WEC-D, and when the WEC value is 2 for WEC-H. There is no clear relationship between number of steps and the standard deviation for WEC-A and WEC-H, but we found it best to use at least 4 or 5 steps with WEC-D to achieve a small standard deviation (meaning more consistent results).

We determined that the WEC-D method is the best wake spreading method we investigated as it had the lowest minimum, mean, and standard deviation of wake loss. We decided to use WEC-D with 6 steps and a maximum ξ value of three for the case studies. 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 as discussed in [13]. In the balance of this paper we will refer to WEC-D as WEC because the other methods have no more relevance in this discussion.

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

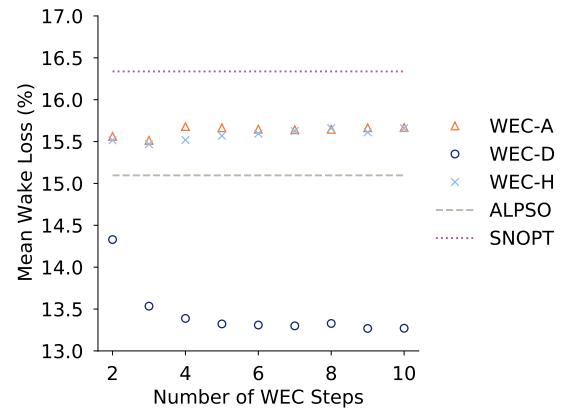


Figure 23. Mean wake loss results for varying the number of WEC steps while holding the expansion parameter constant at three. Each data point represents 200 separate optimizations with different starting points.

Table 4. WEC Parameters and Convergence Tolerances

7.4. ALPSO+WEC

Because WEC adjusts the design space, it can be applied with any optimization algorithm. We tested the impact of WEC with ALPSO in case 2 (see section 5.2). For this test we used only WEC-D and used a maximum WEC value 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 as in the ALPSO runs. We chose to use 25 inner iterations, the number of inner iterations that was best for ALPSO without WEC. 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, such as 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. 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.

7.5. 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 and Porté-Agel model in section 7.3. We tested WEC with the Jensen cosine model only on case 2 (see section 5.2). Further improvements with WEC are likely available if WEC were tuned to this model and case combination.

8. 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 mean results are more consistent and less dependent on the starting layout, which in turn means we need fewer optimizations to be confident of having a good wind farm design. A low number of function calls means 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. 24.

We can see in fig. 24 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. 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. For case 2, the SNOPT+WEC distribution does not even overlap with the SNOPT distribution. The low outliers in the starting location wake loss distributions are the designed layouts depicted in figs. 9, 11 and 15.

While looking at the final wake loss distributions is informative, we are also interested in how the various methods got to those final locations. The convergence history for SNOPT, SNOPT+WEC, and ALPSO for each case study are shown in figs. 25 to 29. The ALPSO runs all terminated with the same number of function calls because function calls was controlled by the number of inner and

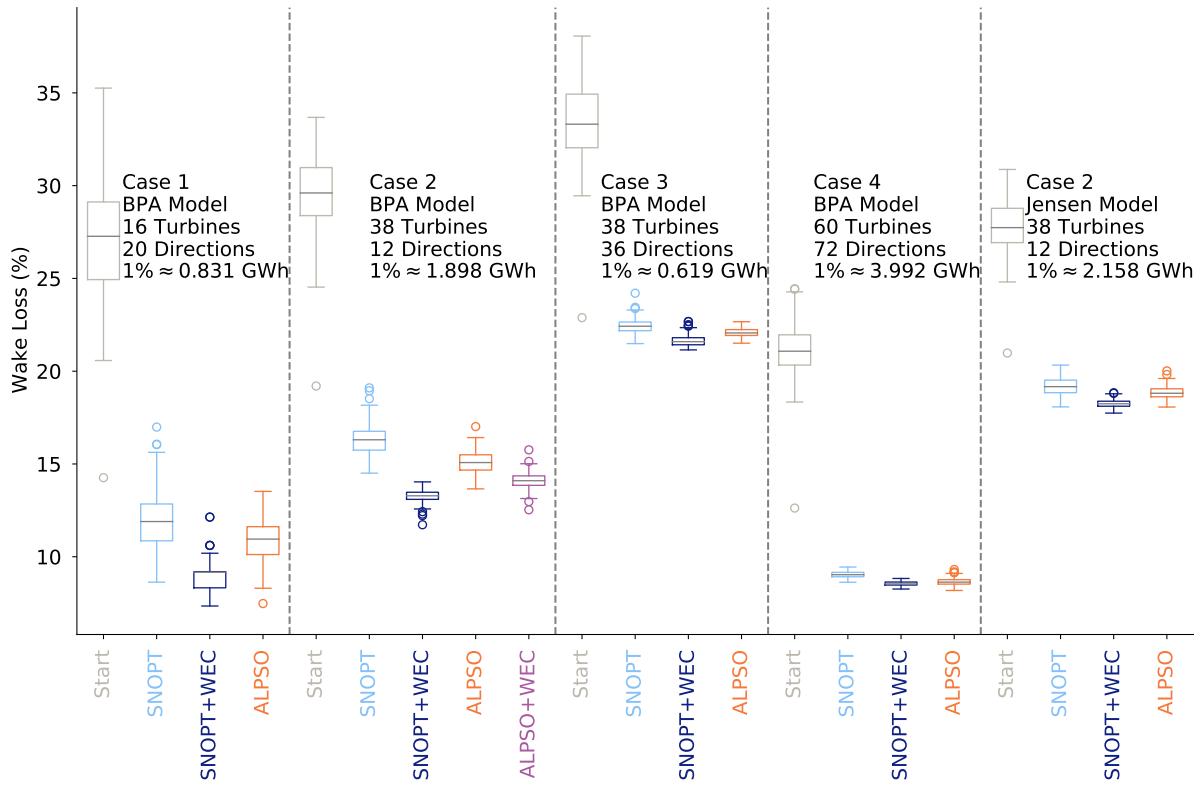


Figure 24. Results distributions of all case studies.

outer iterations (see table 3). The straight lines in the ALPSO histories (from 1×10^0 to about 1.5×10^2 in fig. 25) 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. Straight lines can also be seen in the SNOPT and SNOPT+WEC convergence histories in each of the case studies. These are just small steps taken that appear to have no change in the figures. We have adjusted wake loss values to be calculated by 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. 25 to 29 correspond with the y-axis in fig. 24, 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).

The convergence histories for case 1, shown in fig. 25, 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. 26 to 29. While SNOPT+WEC did terminate in fewer function calls than SNOPT in case 4 (fig. 29), the convergence rate was still higher for SNOPT alone. 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 also the most noticeable difference between the Bastankhah and Jensen model results. As can be seen in

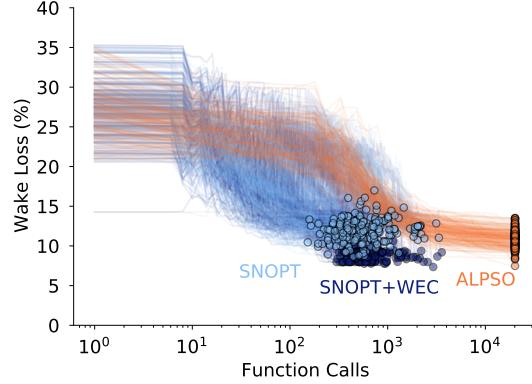


Figure 25. Convergence histories of optimizations for case 1 (16 turbines and 20 wind directions) using the Bastankhah model. Markers indicate optimized values.

comparing fig. 26 and fig. 27, 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. The difference in **convergencce** rate for the gradient-free 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 appear to remain at higher levels longer in the convergence histories than without WEC. This is a feature of the method and demonstrates the effectiveness of WEC at allowing optimization algorithms to escape local optima. The different characteristics in the convergence paths with and without WEC, for both SNOPT and ALPSO, are very apparent in fig. 26, where both algorithms with WEC stay higher longer and then drop rapidly to be below the results of the algorithms without WEC.

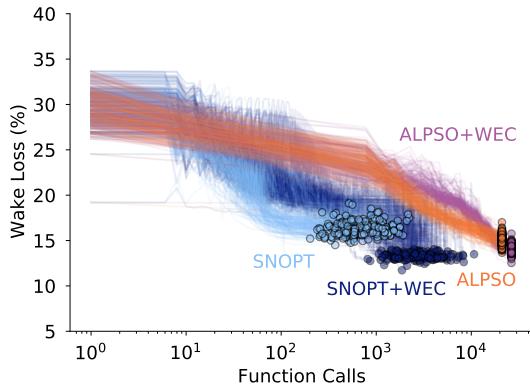


Figure 26. Convergence histories of optimizations for case 2 (38 turbines and 12 wind directions) using the Bastankhah model. Markers indicate optimized values.

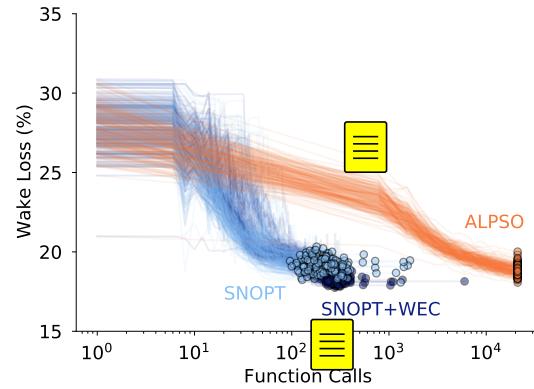


Figure 27. Optimization convergence histories for case 2 (38 turbines and 12 wind directions) using the Jensen model. Markers indicate optimized values.

Cases 3 and 4 (figs. 28 and 29) showed similar trends as cases 1 and 2 (figs. 25 to 27), 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. 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.

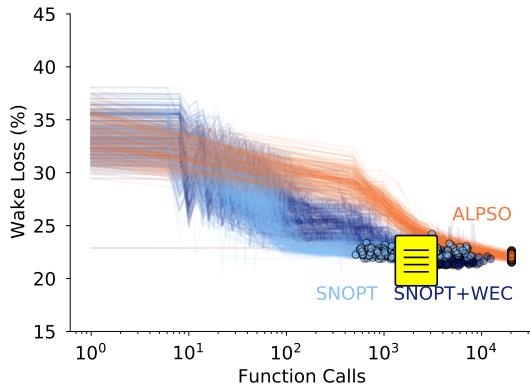


Figure 28. Convergence histories of optimizations for case 3 (38 turbines and 36 wind directions) using the Bastankhah model. Markers indicate optimized values.

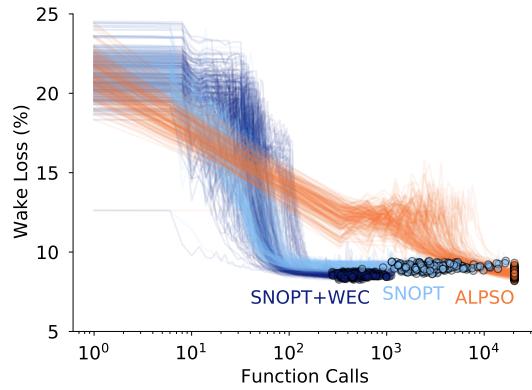


Figure 29. Convergence histories of optimizations for case 4 (60 turbines and 72 wind directions) using the Bastankhah model. Markers indicate optimized values.

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 **that the results are almost definitely not random chance**. 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 except for case 1 with SNOPT+WEC. In case 1, the SNOPT+WEC results had high outliers that skewed the mean upward. SNOPT+WEC resulted in the lowest average wake loss percentage across all cases. The largest difference was for case 2, where SNOPT+WEC resulted in a reduction in wake loss percentage of 3.058 compared with the results from SNOPT alone.

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

	Function Calls			Wake Loss (%) [*]						p
	Median	Low	High	Median	Mean	SD	Low	High		
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 (%) [*]						p
	Median	Low	High	Median	Mean	SD	Low	High		
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 (%) [*]						p
	Median	Low	High	Median	Mean	SD	Low	High		
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 (%) [*]						p
	Median	Low	High	Median	Mean	SD	Low	High		
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

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

	Function Calls			Wake Loss (%) [*]						p
	Median	Low	High	Median	Mean	SD	Low	High		
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

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 have smaller standard deviations than SNOPT alone for all cases and ALPSO for all but one case, 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 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.

9. Conclusion

The wake expansion continuation (WEC) method 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. 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. Results for the case studies using WEC show a statistically significant reduction ($p < 0.001$) in optimized wake loss compared to optimization without WEC for all test cases. **These results show WEC to be most effective with a gradient-based optimization algorithm.**

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. This work was also supported in part by the National Science Foundation under Grant No. 1539384. 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 National Science Foundation.

References

- [1] Rios L M and Sahinidis N V 2013 *Journal of Global Optimization* **56** 1247–1293 ISSN 1573-2916 URL <https://doi.org/10.1007/s10898-012-9951-y>
- [2] Herbert-Acero J, Probst O, Rethore P, Larsen G and Castillo-Villar K 2014 *Energies* **7**
- [3] Fleming P, Ning A, Gebraad P and Dykes K 2015 *Wind Energy*
- [4] Guirguis D, Romero D A and Amon C H 2016 *Applied Energy* **179** 110 – 123 ISSN 0306-2619 URL <http://www.sciencedirect.com/science/article/pii/S0306261916308765>
- [5] Gebraad P, Thomas J J, Ning A, Fleming P and Dykes K 2017 *Wind Energy* **20** 97–107
- [6] Mosetti G, Poloni C and Diviacco B 1994 *Journal of Wind Engineering and Industrial Aerodynamics* **51** 105 – 116 ISSN 0167-6105 URL <http://www.sciencedirect.com/science/article/pii/0167610594900809>
- [7] Grady S, Hussaini M and Abdullah M 2005 *Renewable Energy* **30** 259 – 270 ISSN 0960-1481 URL <http://www.sciencedirect.com/science/article/pii/S0960148104001867>
- [8] Martínez J E G and Zaayer M 2014 *Layout optimisation of offshore wind farms with realistic constraints and options* Master's thesis Delft University of Technology
- [9] Stanley A P J and Ning A 2019 *Wind Energy Science* **4** 663–676
- [10] Réthoré P E, Fuglsang P, Larsen G C, Buhl T, Larsen T J and Aagaard Madsen H 2014 *Wind Energy* **17** 1797–1816
- [11] Graf P, Dykes K, Scott G, Fields J, Lunacek M, Quick J and Rethore P E 2016 *Journal of Physics: Conference Series* **753** 062004 URL <http://stacks.iop.org/1742-6596/753/i=6/a=062004>
- [12] Mittal P and Mitra K 2017 *IFAC-PapersOnLine* **50** 159 – 164 ISSN 2405-8963 20th IFAC World Congress URL <http://www.sciencedirect.com/science/article/pii/S2405896317300393>
- [13] Mobahi H and III J F 2015 URL <https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9299>
- [14] Thomas J J and Ning A 2018 A method for reducing multi-modality in the wind farm layout optimization problem *Journal of Physics: Conference Series* vol 1037 (Milano, Italy: The Science of Making Torque from Wind)
- [15] Bastankhah M and Porté-Agel F 2016 *Journal of Fluid Mechanics* **806** 506–541
- [16] Jensen N O 1983 A note on wind generator interaction Tech. rep. Risø National Laboratory DK-4000 Roskilde, Denmark
- [17] Niayifar A and Porté-Agel F 2016 *Energies* **9** 1–13
- [18] Thomas J J, Annoni J, Fleming P and Ning A 2019 Comparison of wind farm layout optimization results using a simple wake model and gradient-based optimization to large-eddy simulations *AIAA Scitech 2019 Forum* (San Diego, CA)
- [19] Katic I, Højstrup J and Jensen N 1986 A simple model for cluster efficiency *European Wind Energy Association Conference and Exhibition* (Rome - Italy: European Wind Energy Association)
- [20] Western Regional Climate Center 2012 Station wind rose: Nantucket URL https://wrcc.dri.edu/cgi-bin/wea_windrose.pl?1aKACK
- [21] NoordzeeWind Weather data of the offshore wind farm URL https://www.noordzeewind.nl/en_nl/knowledge/weather-data.html
- [22] Gray J, Moore K T and Naylor B A 2010 *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*
- [23] Gill P, Murray W and Saunders M 2005 *SIAM Review* **47** 99–131
- [24] Jansen P and Perez R 2011 *Computers & Structures* **89** 1352 – 1366 ISSN 0045-7949 URL <http://www.sciencedirect.com/science/article/pii/S0045794911000824>
- [25] Perez R E, Jansen P W and Martins J R R A 2012 *Structural and Multidisciplinary Optimization* **45** 101–118
- [26] Hascoët L and Pascual V 2013 *ACM Transactions On Mathematical Software* **39** URL <http://dx.doi.org/10.1145/2450153.2450158>
- [27] Wu N, Kenway G, Mader C A, Jasa J and Martins J R R A 2020 *Journal of Open Source Software* **5** 2564