Wind Energ. Sci., 8, 1235–1250, 2023
https://doi.org/10.5194/wes-8-1235-2023
© Author(s) 2023. This work is distributed under
the Creative Commons Attribution 4.0 License.

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE

Stochastic gradient descent for wind farm optimization

Julian Quick, Pierre-Elouan Rethore, Mads Mølgaard Pedersen, Rafael Valotta Rodrigues, and Mikkel Friis-Møller

Technical University of Denmark, Risø National Laboratory for Sustainable Energy,
Frederiksborgvej 399, 4000 Roskilde, Denmark

Correspondence: Julian Quick (juqu@dtu.dk)

Received: 31 October 2022 – Discussion started: 9 November 2022
Revised: 17 April 2023 – Accepted: 20 June 2023 – Published: 1 August 2023

Abstract. It is important to optimize wind turbine positions to mitigate potential wake losses. To perform this optimization, atmospheric conditions, such as the inflow speed and direction, are assigned probability distributions according to measured data, which are propagated through engineering wake models to estimate the annual energy production (AEP). This study presents stochastic gradient descent (SGD) for wind farm optimization, which is an approach that estimates the gradient of the AEP using Monte Carlo simulation, allowing for the consideration of an arbitrarily large number of atmospheric conditions. SGD is demonstrated using wind farms with square and circular boundaries, considering cases with 100, 144, 225, and 325 turbines, and the results are compared to a deterministic optimization approach. It is shown that SGD finds a larger optimal AEP in substantially less time than the deterministic counterpart as the number of wind turbines is increased.

1 Introduction

Wind farms are groups of wind turbines that harness the power in the atmospheric boundary layer to provide renewable energy. When a wind turbine absorbs energy from the air, the air downstream of the wind turbine has reduced power, which often reduces the power production of downstream turbines. This is known as the wake effect (Sanderse, 2009; Hasager et al., 2013). When a new wind power plant is to be constructed, optimal turbine locations are determined using engineering wind farm models (Samorani, 2013; Ning. et al., 2020; Annoni et al., 2018). Turbine positions are optimized to exploit the benefits of the local wind resource while avoiding energy losses from turbine wakes. In the wind turbine placement problem, atmospheric conditions, such as the inflow speed and direction, are assigned probability distributions according to measured data. By propagating these probability distributions through the engineering wake model, the annual energy production (AEP) can be estimated. The AEP is often computed using rectangular quadrature, dividing the relevant speeds and directions into equal-sized bins, then computing the expected AEP as the product of the power and probability of each bin, added together, then multiplied by the number of hours per year. The cost of wind farm optimization generally increases with the number of atmospheric conditions considered during AEP computation, and this expense becomes more extreme as more complex wake models (e.g., Reynolds-averaged Navier–Stokes models) are considered. For example, there are some memory limitations when computing AEP gradients using automatic differentiation with very large wind farms. This has given rise to studies seeking convergence of the AEP, proposing methods such as polynomial chaos expansion (Padrón et al., 2019; Murcia et al., 2015) or Bayesian quadrature (King et al., 2020) to avoid discretizing the input distributions into evenly spaced intervals. In this study, we present an approach for wind farm optimization that estimates the gradient of the AEP using Monte Carlo simulation. This does not require the input to be discretized at all and allows for the consideration of an arbitrarily large number of atmospheric conditions.

Stochastic gradient descent (SGD) is an optimization algorithm commonly used in machine learning when selecting neural network weights (Ketkar, 2017). The algorithm samples the gradient of a stochastic objective, following the mean gradient by a specified distance, then repeating the process, which amounts to optimizing the expected value of the objec-

tive. The SGD algorithm is often enhanced to avoid oscillations caused by large changes in the gradient of the objective (Ruder, 2016). This includes methods to reuse previous gradient information (Qian, 1999), dampen oscillations (Riedmiller and Braun, 1993), or incorporate an estimate of the Hessian matrix (Moritz et al., 2016; Byrd et al., 2016; Liu et al., 2018; Najafabadi et al., 2017). Kingma and Ba (2014) introduced the Adam SGD algorithm, which reuses gradient evaluations and dampens oscillations, and which is the basis of the SGD method we propose in this study.

Interestingly, SGD is not often applied to problems with nonlinear constraints, although it can be fruitful to include nonlinear constraints in the context of training a machine learning algorithm. For example, when recognizing three-dimensional pictures of people, it can be useful to impose a constraint that any person's left arm should be close to the same length as their right arm (Márquez-Neila et al., 2017). Many frameworks have been proposed for constrained SGD, including the log-barrier function (Kervadec et al., 2019), penalty functions (Márquez-Neila et al., 2017), blending barrier and penalty functions (Kervadec et al., 2019), and Riemannian geometry (Roy and Harandi, 2017). In this study, we use a penalty term to transform the constrained problem into an unconstrained optimization.

The wind farm layout optimization problem presents a setting where the objective (AEP) can be formulated as being stochastic (e.g., the AEP is derived from a probability density function), while the constraints (e.g., boundaries and minimum turbine spacing) are firmly deterministic. This paper explores the potential benefits of formulating the wind farm layout optimization problem in this way. As part of this, the Adam algorithm is extended to optimize a stochastic objective with deterministic constraints. To the best of the authors' knowledge, this exact algorithm has not been published before.

This study benchmarks the performance of the proposed SGD approach when compared to conventional gradient-based optimization within the TOPFARM framework (DTU Wind Energy Systems, 2023b), considering wind farms with different shapes and sizes. We examine the open-source SLSQP algorithm (Kraft, 1988), which is employed in many engineering frameworks (King et al., 2017; Allen et al., 2020; Wu et al., 2020; Zhang et al., 2022; Zilong and Wei, 2022; Kölle et al., 2022; Clark et al., 2022; Simley et al., 2023) and has been used in previous comparisons of optimization algorithms (Lam et al., 2018; Li and Zhang, 2021; Fleming et al., 2022). The TOPFARM framework has been used with SLSQP in several wind farm optimization studies (Riva et al., 2020; Ciavarra et al., 2022; Criado Risco et al., 2023; Rodrigues et al., 2023). In future work, this approach can be extended to co-optimize layout and control strategy – the SGD framework can naturally incorporate uncertainty quantification when modeling the potential control strategies for potential layouts (similar to the work in Gebraad et al., 2017; Quick et al., 2020; Howland et al., 2022).

While there are some wind plant optimization studies that resemble our approach, we are not aware of any studies that have applied SGD to the wind farm optimization problem (although SGD has been applied to other problems in engineering optimization, e.g., De et al., 2020; Sivanantham and Gopalakrishnan, 2022). Several wind farm optimization studies have made use of gradient-based optimization techniques (Herbert-Acero et al., 2014; Guirguis et al., 2016; Graf et al., 2016; Gebraad et al., 2017; Baker et al., 2019; Riva et al., 2020; Stanley et al., 2021; Croonenbroeck and Hennecke, 2021). Feng and Shen (2015) present a random search approach, moving the wind turbines one by one using a greedy algorithm. Some studies have employed neural networks to forecast power production (Godinho and Castro, 2021), estimate local atmospheric conditions (Stengel et al., 2020), suggest control strategies (Najd et al., 2020), or optimize engineering wake models (Zhang et al., 2021; Zhang and Zhao, 2022; Hussain et al., 2022), which all use SGD algorithms to train the parameters of the neural networks.

The remainder of the paper is the following. Section 2 outlines the SGD and deterministic optimization approaches used in this study. Section 3 details the wind farm optimization application cases examined. Section 4 discusses the results of these optimization comparisons. Section 5 provides conclusions and future research directions.

2 Methods

When deciding where to put wind turbines, a typical strategy is to maximize wind farm AEP while ensuring turbines are within the prospective site and are not spaced too closely together. In this study, we examine square and circular wind farms, where the corresponding optimization problems are posed as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{maximize}} \quad & \text{AEP}(\boldsymbol{x},\boldsymbol{y}) \\ \text{subject to} \quad & \left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 \geq (N_\text{D} D)^2, \\ & \forall i \neq j \\ & x_l \leq x_i \leq x_u \\ & y_l \leq y_i \leq y_u \end{aligned} \tag{1}$$

and

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{maximize}} \quad & \text{AEP}(\boldsymbol{x},\boldsymbol{y}) \\ \text{subject to} \quad & \left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 \geq (N_\text{D} D)^2, \\ & \forall i \neq j \\ & \sqrt{x_i^2 + y_i^2} \leq R, \end{aligned} \tag{2}$$

respectively, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the turbine horizontal and vertical locations, x_l and x_u are the lower and upper horizontal square wind farm boundaries, y_l and y_u are the lower and upper square wind farm vertical boundaries, R is the radius of the circular wind farm, D is the rotor diameter, and N_D is the minimum allowable spacing between turbines measured

in rotor diameters. From this point forward, we will use a single variable to represent the x and y locations, $\boldsymbol{s} = \{\boldsymbol{x}, \boldsymbol{y}\}$. When optimizing square wind farms, x_l, x_u, y_l, and y_u are constant.

The AEP is defined as

$$\mathrm{AEP}(\boldsymbol{s}) = 8760 \int_0^{2\pi} \int_0^{\infty} P(\boldsymbol{s}, u_\infty, \theta)\, \pi(u_\infty, \theta)\, \mathrm{d}u_\infty \mathrm{d}\theta, \tag{3}$$

where P is power, π is probability, u_∞ is the freestream velocity, and θ is the freestream direction. The 8760 factor reflects the number of hours per year, converting from units of power to units of energy.

The AEP is typically estimated through rectangular quadrature, where the freestream velocity and direction are discretized using evenly spaced intervals,

$$\mathrm{AEP}(\boldsymbol{s}) \approx 8760 \sum_{d=1}^{D} \sum_{u=1}^{U} P(\boldsymbol{s}, \mathcal{U}_\mathrm{u}, \theta_\mathrm{d})\, \rho(\mathcal{U}_\mathrm{u}, \theta_\mathrm{d}), \tag{4}$$

where $\boldsymbol{\mathcal{U}}$ is a vector of evenly spaced wind speeds, $\boldsymbol{\theta}$ is a vector of evenly spaced wind directions, and $\rho(\mathcal{U}_\mathrm{u}, \theta_\mathrm{d})$ is a probability mass function.

The AEP can also be estimated through Monte Carlo integration,

$$\mathrm{AEP}(\boldsymbol{s}) \approx 8760 \frac{1}{K} \sum_{k=1}^{K} P\left(\boldsymbol{s}, u_\infty^{(k)}, \theta^{(k)}\right), \tag{5}$$

where $u_\infty^{(k)}$ and $\theta^{(k)}$ represent draw k of the probability distribution $\pi(u_\infty,\ \theta)$.

The associated AEP gradient can also be approximated through Monte Carlo simulation:

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{s}}\mathrm{AEP} \approx 8760 \frac{1}{K} \sum_{k=1}^{K} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{s}} P\left(\boldsymbol{s}, u_\infty^{(k)}, \theta^{(k)}\right). \tag{6}$$

2.1 Stochastic gradient descent

SGD is built upon the steepest descent algorithm. Early SGD algorithms added a moving average term (sometimes referred to as "momentum") to avoid spurious oscillations (Tian et al., 2023). The conventional Adam SGD algorithm uses two moving averages: one of the gradient and one of the squared gradient. The ratio of these moving averages is used to determine the search direction. SGD algorithms are often combined with a learning rate scheduler, where the step size of the gradient descent is gradually decreased, allowing the optimization algorithm to hone in on the best solution. While the Adam algorithm is already designed to dynamically change the step size, including a learning rate scheduler can further improve the performance. The conventional Adam algorithm is designed for unconstrained optimization algorithms. In the following, we extend the algorithm to allow for deterministic constraints, which is a case that is common in mechanical engineering and unusual in the context of training neural networks. The basic idea is to aggregate the constraints to a penalty term with units that are consistent with the objective. The penalty term is designed so that, initially, the penalty gradients are of similar magnitude to the AEP gradients and so that the penalty gradients overwhelm the AEP gradients as the optimization continues. The SGD algorithm is shown in Algorithm 1, where $\boldsymbol{s}_0$ is the initial turbine position; i is the iteration number; α_i is referred to as the constraint multiplier; $\gamma(\boldsymbol{s})$ is a penalty function; $P(\boldsymbol{s}, u_\infty^{(k)}, \theta^{(k)})$ is the wind farm power associated with the inflow speed and direction, $u_\infty^{(k)}$ and $\theta^{(k)}$; K is the number of samples employed in each SGD iteration; β_1 and β_2 are constants; T is the number of SGD iterations; $\mathcal{S}$ is the learning rate scheduler; and η_i is the learning rate. By default, the early stopping option is false.

Algorithm 1 TOPFARM stochastic gradient descent implementation.

$\boldsymbol{m} \leftarrow \boldsymbol{0}$, $\boldsymbol{v} \leftarrow \boldsymbol{0}$, $\boldsymbol{s} \leftarrow \boldsymbol{s}_0$
for i in $[0, 1, 2, \ldots, T-1]$
do
if early_stopping and $\eta_i/\eta_0 \leq$ threshold:
$\boldsymbol{j} = \alpha_i \frac{\partial \gamma}{\partial \boldsymbol{s}}$
if $|\boldsymbol{j}| \equiv 0$:
break
else:
$\boldsymbol{j} = -\frac{8760}{K} \sum_{k=1}^{K} \frac{\partial}{\partial \boldsymbol{s}} P\left(\boldsymbol{s},\ u_\infty^{(k)},\ \theta^{(k)}\right) + \alpha_i \frac{\partial \gamma}{\partial \boldsymbol{s}}$
$\boldsymbol{m} = \beta_1 \boldsymbol{m} - (1-\beta_1)\, \boldsymbol{j}$
$\boldsymbol{v} = \beta_2 \boldsymbol{v} - (1-\beta_2)\, \boldsymbol{j}^2$
$\hat{\boldsymbol{m}} = \frac{\boldsymbol{m}}{1-(\beta_1)^i}$
$\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{1-(\beta_2)^i}$
$\boldsymbol{s} = \boldsymbol{s} - \eta_i \hat{\boldsymbol{m}} / \sqrt{\hat{\boldsymbol{v}}}$
$\eta_i = \mathcal{S}(\eta_0,\ \delta, i)$
$\alpha_i = \alpha_0 \frac{\eta_0}{\eta_i}$

The spacing between turbines is enforced using a penalty term,

$$\gamma_\mathrm{s} = \sum_{\forall i,\, j>i} \min\left[\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 - (N_\mathrm{D} D)^2, 0\right]. \tag{7}$$

Similarly, the distance outside of boundaries is enforced using a penalty term. When considering square wind farms this penalty term is defined as

$$\gamma_\mathrm{b} = \sum_{i=1}^{N_\mathrm{t}} \Big[\max(x_i - x_\mathrm{ub}, 0)^2 + \max(x_\mathrm{lb} - x_i, 0)^2 + \max(y_i - y_\mathrm{ub}, 0)^2 + \max(y_\mathrm{lb} - y_i, 0)^2\Big],$$

and, in the case of circular boundaries, it is defined as

$$\gamma_\mathrm{b} = \sum_{i=1}^{N_\mathrm{t}} \max\left(\sqrt{x_i^2 + y_i^2} - R, 0\right)^2, \tag{8}$$

where N_t is the number of wind turbines.

The total penalty, γ, is defined as the sum of these two penalty terms,

$$\gamma(\boldsymbol{s}) = \gamma_\mathrm{s}(\boldsymbol{s}) + \gamma_\mathrm{b}(\boldsymbol{s}). \tag{9}$$

The gradient of the penalty term, γ, is scaled before being added to the negative gradient of the AEP using the scaling factor, α_i.

In Algorithm 1, the learning rate (η_i), constraint multiplier (α_i), number of SGD iterations (T), and the samples per SGD iteration (K) are all free parameters. These parameters can be optimized to perform well for individual wind farm optimization problems. But there is no guarantee that these particular parameters will perform well for other wind farm problems – and this meta-optimization can be expensive. In the machine learning community, these parameters are sometimes optimized using evolutionary, grid search, or Bayesian optimization approaches (Alibrahim and Ludwig, 2021). In addition, it is common to schedule the learning rate to decay as the optimization proceeds (You et al., 2019; Denkowski and Neubig, 2017).

We propose a method for setting free parameters to ensure that all units are consistent. The only free parameters we manually set are the number of optimization iterations and the number of power samples per iteration. The optimization generally becomes more accurate and more expensive as these parameters increase, and users are free to balance this trade-off as they see fit. Our formulation does not guarantee that all intermediate solutions satisfy the constraints, especially in the beginning of the optimization. The constraint multiplier begins on a comparable scale to the AEP and is scheduled to increase so that the constraint gradients overwhelm the AEP gradients as the optimization progresses. The number of iterations, T, can be based on a prescribed computational budget.

We initially attempted this approach using the widely used default parameter values in the original Adam algorithm, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The parameters can be thought of as adding momentum to the moving averages of the gradient and squared gradient, $\boldsymbol{m}$ and $\boldsymbol{v}$. We found that these default values gave too much emphasis to gradients from the penalty function, launching the turbines away from the boundaries in a dramatic fashion. Instead, we suggest the parameters $\beta_1 = 0.1$ and $\beta_2 = 0.2$, which encode a shorter memory of the presence of the penalty. With these new default parameters, and the learning rate defined below, we observed successful convergence for a wide variety of test cases.

The learning rate, η_i, can be interpreted as converting $\hat{\boldsymbol{m}}/\sqrt{\hat{\boldsymbol{v}}}$ (with unity units) to distance (units of m). In this study, the learning rate is scheduled to decay according to

$$\begin{aligned} S(t=0) &= \eta_0 \\ S(t=T-1) &= \eta_T, \end{aligned} \tag{10}$$

where T is the number of optimization iterations, η_0 is the initial learning rate, and η_T is the scheduled final learning rate. This final learning rate can be thought of as a solution tolerance for the design variables. In this study, we set $\eta_T = 0.1$ m.

The initial learning rate, η_0, is based on a length scale parameter, L, which corresponds to a reasonable initial step size for the optimization. By setting the initial learning rate according to

$$\eta_0 = L = D/5, \tag{11}$$

where D is the turbine rotor diameter, we encourage the turbines to move at most L distance every optimization iteration.

The learning rate is scheduled to decay as

$$S(\eta_0, \delta, t) = \eta_0 \prod_{i=0}^{t} \frac{1}{1 + i\delta}, \tag{12}$$

where δ is a parameter that controls the learning rate length, such that the final learning rate is η_T. The parameter δ is numerically set as

$$\delta(\eta_0, \eta_T, T) = \underset{\delta}{\operatorname{argmin}} |\eta_T - S(\eta_0, \delta, T)|. \tag{13}$$

The constraint multiplier, α_i, can be interpreted as converting the gradient of constrained square distances (in units of m) to AEP gradients. The initial constraint multiplier, α_0, is set as the mean absolute AEP gradient divided by the length scale, L, so that the separation constraint has a similar scale to AEP gradients,

$$\alpha_0 = \frac{\mathrm{mean}\left[|\nabla \mathrm{AEP}(\boldsymbol{s}_0)|\right]}{L}, \tag{14}$$

where mean$[|\nabla \mathrm{AEP}(\boldsymbol{s}_0)]$ is the mean of the absolute AEP gradient of the initial guess with respect to each component of the gradient. During each iteration, the constraint multiplier, α_i, is scheduled to increase based on the inverse of the learning rate,

$$\alpha_i = \alpha_0 \frac{\eta_0}{\eta_i}. \tag{15}$$

The wind rose samples, $(u_\infty^{(i)}, \theta^{(i)}) \sim \pi(u_\infty, \theta)$, are randomly selected based on the direction frequency and direction-specific Weibull shape and scale parameters. Note that the tilde ($\sim$) denotes a shared probability distribution. After a direction is sampled, the wind speed is sampled as a continuous Weibull-distributed random variable,

$$u_\infty(\theta) \sim \mathcal{W}\left[u_\infty, A(\theta), k(\theta)\right], \tag{16}$$

where the probability density of the Weibull distribution, $\mathcal{W}$, is given by

$$\mathcal{W}(u_\infty, A, k) = \frac{k}{A}\left(\frac{u_\infty}{A}\right)^{k-1} \exp\left[-\left(\frac{u_\infty}{A}\right)^{k}\right]. \quad (17)$$

2.2 Deterministic approach

The SLSQP algorithm (Kraft, 1988) is selected to be the deterministic optimization algorithm to act as a benchmark to the SGD approach. SLSQP is a conventional deterministic optimization approach. It is employed in many open-source engineering design codes (Allen et al., 2020; Wu et al., 2020; Ciavarra et al., 2022) and has been used in previous comparisons of optimization algorithms (Lam et al., 2018; Li and Zhang, 2021; Fleming et al., 2022).

The spacing and boundary constraints are passed to the optimizer as individual inequality constraints. The spacing constraints are defined as

$$C_{ij} = \left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 - (N_\mathrm{D} D)^2 \quad \forall i,\ j > i, \quad (18)$$

where $\boldsymbol{C}$ is an upper triangular matrix of nonlinear inequality constraints.

Square wind farm boundaries are represented using four inequality constraints per turbine,

$$\begin{aligned} D_{i1} &= x_i - x_u \\ D_{i2} &= x_l - x_i \\ D_{i3} &= y_i - y_u \\ D_{i4} &= y_l - y_i, \end{aligned} \quad (19)$$

and the circular wind farm boundaries are represented with one inequality constraint per turbine,

$$D_i = \sqrt{x_i^2 + y_i^2} - R, \quad (20)$$

where $\mathbf{D}$ is a matrix of boundary constraints that must be less than or equal to 0.

3 Application

We apply the optimization approaches discussed above to optimize wind power plants of various sizes using the TOPFARM framework (DTU Wind Energy Systems, 2023b). Each farm consists of turbines with 70 m hub heights, 80 m rotor diameters, and 2 MW rated powers. Power is computed using PyWake (Pedersen et al., 2019), which is an open-source wake modeling tool that has been used in several related studies (Riva et al., 2020; Rodrigues et al., 2022; Ciavarra et al., 2022; van der Laan et al., 2023; Fischereit et al., 2022). Power gradients are computed directly from PyWake using automatic differentiation. The power of each turbine is estimated by a combination of velocity deficits predicted by the Bastankhah Gaussian wake model (Bastankhah and Porté-Agel, 2014) using the default parameters in the PyWake tool and the squared sum superposition (Pedersen et al., 2019; DTU Wind Energy Systems, 2023a). We require each turbine to be spaced at minimum two rotor diameters apart ($N_\mathrm{D} = 2$). This is imposed as an optimization constraint. We considered wind farms with square and circular boundaries. The square wind farm boundaries are determined as

$$\begin{aligned} x_l &= 0 \\ y_l &= 0 \\ x_u &= D\left(\sqrt{N_\mathrm{t}} - 1\right)\Delta \\ y_u &= D\left(\sqrt{N_\mathrm{t}} - 1\right)\Delta, \end{aligned} \quad (21)$$

and, in cases with circular wind farm boundaries, the radius is determined as

$$R = D\left(\sqrt{N_\mathrm{t}} - 1\right)\Delta, \quad (22)$$

where the Δ parameter controls the average spacing of the turbines. In this study, $\Delta = 5$.

We use the pyOptSparseDriver (Wu et al., 2020) SLSQP (Kraft, 1988) implementation (Virtanen et al., 2020) in TOPFARM. The optimizer was set to run for 300 maximum iterations with a tolerance of 10^{-1}. The TOPFARM "expected_cost" parameter is set to 10. The turbine coordinates are normalized from 0 to 1. In each optimization iteration, the AEP, and the corresponding gradient, is computed using rectangular quadrature as described in Eq. (4), using 360 wind direction bins and 23 wind speed bins, resulting in 8280 power evaluations.

The wind rose, visualized in Fig. 1, is based on PyWake's Lillgrund example site. A probability mass function is assigned to different direction bins. Each direction bin is associated with Weibull scale and shape parameters describing the distributions of wind speeds within the sector. This probability mass is derived from 7 months of measured data used in a previous study (Göçmen and Giebel, 2016). Each direction bin is 30° wide. The reasoning behind this is similar to that behind the IEC 614 400 power curve standard (International Electrotechnical Commission, 2005) – it is crucial that the reference data consider a statistically significant number of data in each bin. This coarse direction discretization results in a faster convergence of the estimated probability mass function than a finer discretization would. The continuous probability density function $\pi(u_\infty, \theta)$ is approximated as $\rho(\theta)\pi(u_\infty|\theta)$, where ρ is the previously mentioned probability mass function, linearly interpolated across 1° bins, and $\pi(u_\infty|\theta)$ is parameterized by direction-specific Weibull shape and scale parameters that are also linearly interpolated from the provided data. With this formulation, the likelihood of different wind directions is provided as a probability mass function, $\rho(\theta)$. This probability mass is used as weights passed to the Numpy "choice" function (Harris et al.,

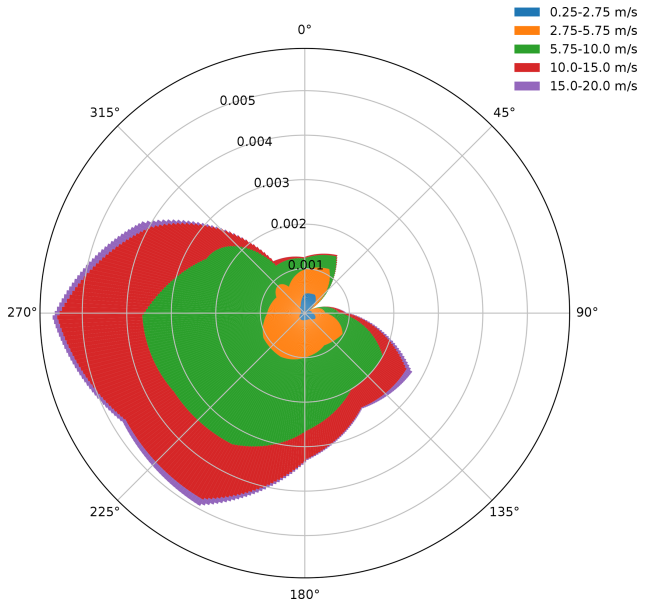

Figure 1. Lillgrund wind speed and direction probability mass function with 360 direction bins and five wind speed bins, where the probability mass function is a linear interpolation of coarser measurements.

2020), allowing the wind direction to be sampled as a discrete random variable. We note that this formulation could be extended to a fully continuous formulation by drawing the direction samples from the inverse of an empirical cumulative direction density function.

In all wind farm optimization problems considered, constraint gradients, and the associated penalty function gradients, are computed analytically. The AEP gradient is computed via automatic differentiation. The directions are discretized from 0 to 360°, with 1° increments. In the deterministic formulation, the discretized wind speed ranges from 3–25 m s^{-1} and is divided using increments of 1 m s^{-1}.

While each Monte Carlo estimate of AEP has significant error, the average error will be close to 0 throughout the course of the SGD optimization. We compare the accuracy of the Monte Carlo approach (Eq. 5) and the quadrature approach (Eq. 4) to estimate the AEP and the L-2 norm of the AEP gradient. The true values are estimated with a very fine discretization of speed and direction, 0.2 m s^{-1} and 0.2°. These are used as reference values to assess the accuracy of the Monte Carlo and quadrature approaches by comparing the errors associated with both approaches as functions of the number of samples and the discretization level, respectively, when analyzing a 100-turbine farm with square boundaries. This convergence analysis is shown in Fig. 2. While some realizations of the Monte Carlo approach yield more accurate results than the quadrature approach, the quadrature approach is generally more accurate than Monte Carlo sampling. The Monte Carlo approach requires on average around 10 times as many power evaluations to obtain the same accuracy as the deterministic approach.

In this study, we select 50 samples for every SGD iteration. Figure 3 shows the measured computational cost of computing AEP gradients using the circular wind farm described in this study, with different wind farm sizes. The minimum measured time is reported as the minimum of 30 identical runs on the DTU Sophia supercomputer (Technical University of Denmark, 2019). The computational time generally scales logarithmically with the number of turbines. This is to be expected, as there are more interaction terms in the wake model as more turbines are considered. The computational time does not scale logarithmically with the number of wind rose samples. For small numbers of turbines, evaluating 10 wind rose samples is about as expensive as evaluating 50 samples. This scaling changes as the wind farm grows in size, and it gradually becomes more expensive to sample the wind rose. The evaluation time appears to converge to a logarithmic scaling for large numbers of wind rose samples. These scaling results are likely influenced by memory limitations.

The optimization algorithms are timed based on the time elapsed between the first and final optimization gradient evaluations. Each optimization case is run on first-generation AMD EPYC 7351 processors.

4 Results and discussion

In the following subsections, the performance of SGD and SLSQP is compared for wind farms with square and circular boundaries, and the sensitivity of the SGD algorithm is assessed.

4.1 Square wind farm

The performance of SGD is compared to the deterministic counterpart, considering wind farms with 100, 144, 225, and 324 turbines, with square boundary constraints, using 20 different initial starting conditions to obtain statistically significant results. The AEP, constraint violation (γ), and time elapsed associated with each optimization solution are plotted in Fig. 4. The SGD approach consistently yields higher AEPs than the SLSQP approach when the number of scheduled SGD iterations, T, is 2000. There is a large range of computational times associated with the SLSQP approach, though the computational expense of SLSQP generally grows much larger than SGD as the number of turbines is increased. This is largely due to the nature of the turbine spacing constraint, the size of which grows as the number of turbines squared. SLSQP takes about as much computational time as the SGD approach with 500 scheduled iterations when there are 100 turbines in the square farm. As the number of turbines grows, the average time required by SLSQP becomes more costly than SGD with 2000 scheduled iterations. The computational cost of SLSQP is a strong function

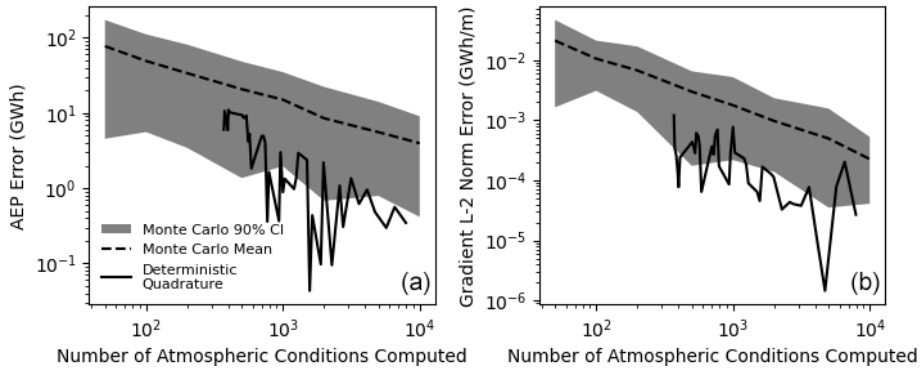

Figure 2. Convergence of AEP **(a)** and L-2 norm of the AEP gradient **(b)** with respect to the number of samples used in the quadrature and Monte Carlo techniques. The grey cloud shows the 90 % confidence interval associated with the Monte Carlo approach, using 50 samples. The dashed black line shows the average error associated with the Monte Carlo approach. The solid black line shows the error associated with the deterministic approach.

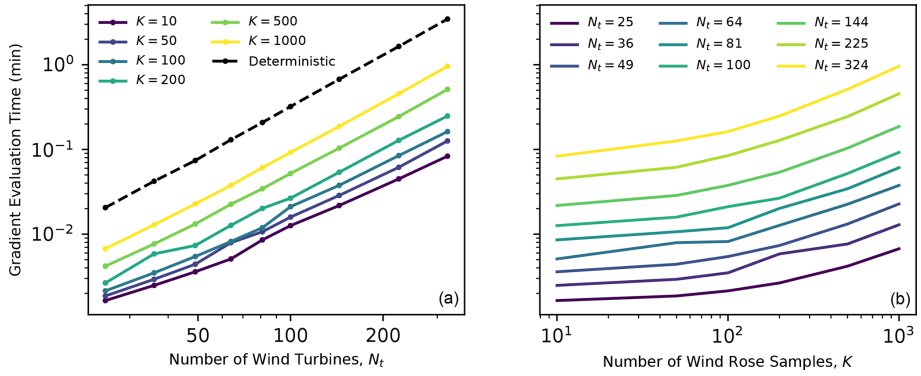

Figure 3. Computational time associated with computing the gradient for various wind farm sizes and wind rose sampling strategies. The left panel compares the cost of computing AEP gradients when using different numbers of Monte Carlo samples with the cost of the full factorial wind rose (8280 samples) for farms with various numbers of wind turbines. The right panel compares the cost of Monte Carlo estimates of the AEP gradient for different numbers of samples of the atmospheric conditions and turbines in the wind farm.

of the initial layout, and the variance of the SLSQP optimization time also increases with the wind farm size. This is due to the complex interaction between the linear boundary constraints and nonlinear spacing constraints. As the proposed SGD formulation does not offer an automatic way to set the number of SGD iterations, T, results are shown for different values of T. When T is increased, the optimizer finds solutions with larger AEPs, with a computational cost that is approximately proportional to T. The SGD solution consistently improves as more optimization iterations are scheduled (larger values of T). Results associated with 1000 SGD iterations tend to yield similar AEPs to the SLSQP approach, and results with 2000 SGD iterations tend to yield higher AEPs than the SLSQP designs.

The final layouts associated with one of the random initial conditions used in the 324-turbine analysis, when $T = 2000$ iterations, are shown in Fig. 5. The SGD approach generally identifies solutions with the majority of turbines packed into the side boundaries. The deterministic algorithm also packed turbines into the edges of the farm, although not as many turbines were packed into the east and west boundaries as in the SGD results. The layouts found using the SGD approach tend to have interior turbines that generally appear to be more aligned in the north–south direction than in the deterministic solutions.

The results of the 100-, 144-, 225-, and 324-turbine wind farm optimization cases are summarized in Table 1. The mean time, mean constraint violation, and mean and standard deviation of the AEP are reported with respect to the 20 random initial starting conditions. Constraint violation is reported as $\sqrt{\gamma/N_t}$ to quantify the mean length of the constraint violations of each turbine. The final constraint violations can be reduced by lowering the η_T parameter. In all of these cases, the SLSQP optimization resulted in solutions with zero constraint violations. This is likely because of the linear formulation of the boundary constraints – when a solution satisfies the spacing constraint, any solutions that satisfy the boundary constraints can quickly be found. SGD with

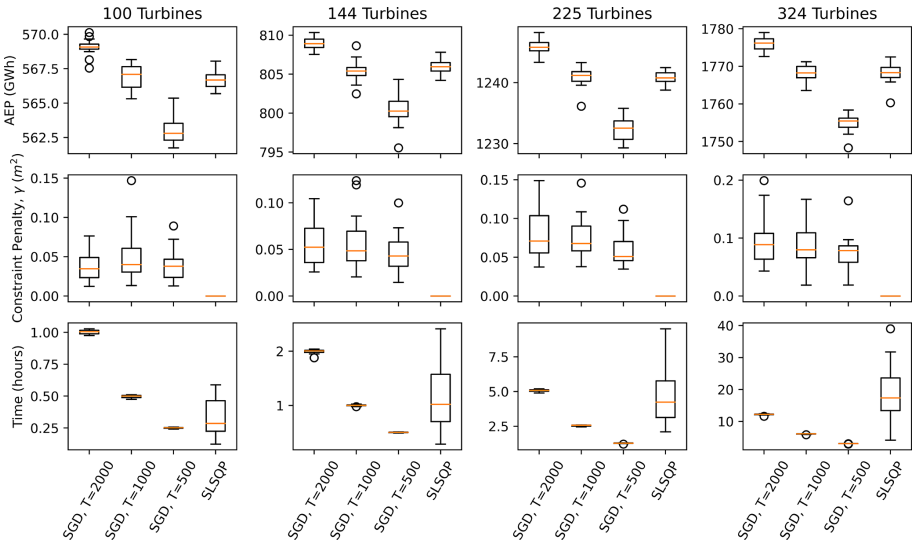

Figure 4. Optimization results associated with SGD and SLSQP for square wind farms with 100, 144, 225, and 324 turbines, using 20 random initial starting conditions. The AEP (top panels), constraint penalty (middle panels), and computational time (bottom panels) are plotted as boxplots. The SGD results are plotted for $T = 500$, 1000, and 2000 iterations.

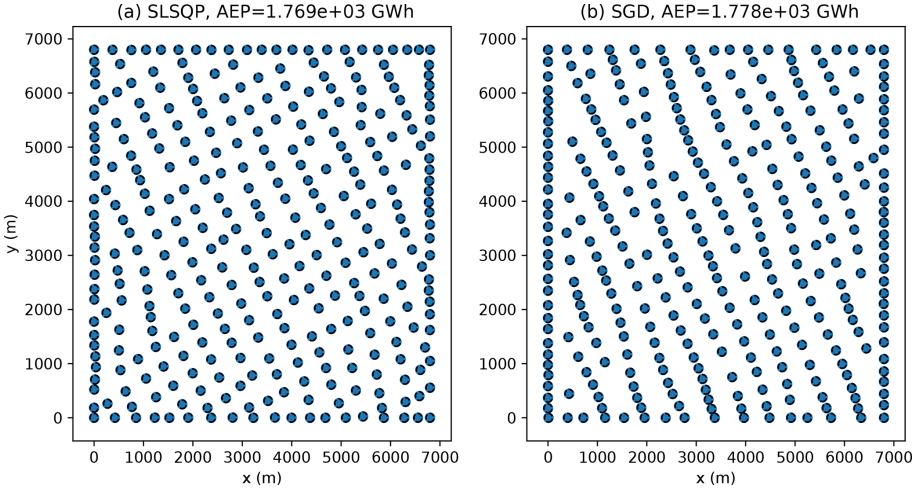

Figure 5. The final layouts found using SLSQP **(a)** and SGD **(b)** using one of the random initial layouts examined in a 324-turbine wind farm with square boundaries for $T = 2000$ iterations. Each circle has a radius of one rotor diameter.

2000 iterations generally yields solutions with AEP that are 0.3 %–0.5 % higher than the solutions found using SLSQP. This is likely because the SGD algorithm is able to better explore the design space by initially relaxing the constraints, allowing for some initial constraint violations.

4.2 Circular wind farms

To ensure that the previously presented results are not specific to square wind farms, we performed a similar set of analyses examining circular wind farms. The results yield similar trends to the analysis using square wind farms – SGD becomes significantly less time-consuming than SLSQP as the number of turbines increases and generally yields solutions with slightly larger AEPs.

Circular winds farms were optimized using 20 random initial layouts, examining different farm sizes, using the SGD and SLSQP optimization algorithms. The results are summarized in Fig. 6. The circular wind farm optimization generally took longer than the square wind farm when using the SLSQP optimizer. This is likely due to the more complicated nature of the circular boundary when using Cartesian coordinates. These results are similar to the results in Sect. 4.1 – as the number of wind turbines and scheduled SGD iterations increases, SGD tends to find solutions with larger AEPs in less computational time.

Table 1. Results of SGD and deterministic optimizations for various square wind farm sizes. Each optimization case is run using 20 random initial starting conditions, and the mean and standard deviation are reported with respect to these 20 initial points. The SGD results are associated with $T = 2000$ iterations.

N_t	Case	Mean time (hours)	Mean AEP (kWh)	AEP standard deviation (kWh)	Mean $\sqrt{\gamma/N_t}$ (m)
100	Deterministic	0.34	5.667×10^8	6.223×10^5	0.000×10^0
	SGD	1.00	5.691×10^8	5.347×10^5	1.919×10^{-3}
144	Deterministic	1.18	8.059×10^8	9.188×10^5	$0.000e \times 10^0$
	SGD	2.00	8.089×10^8	7.727×10^5	1.642×10^{-3}
225	Deterministic	4.79	1.241×10^9	1.004×10^6	0.000×10^0
	SGD	5.07	1.246×10^9	1.110×10^6	1.250×10^{-3}
324	Deterministic	18.69	1.768×10^9	2.556×10^6	0.000×10^0
	SGD	12.20	1.776×10^9	1.810×10^6	9.484×10^{-4}

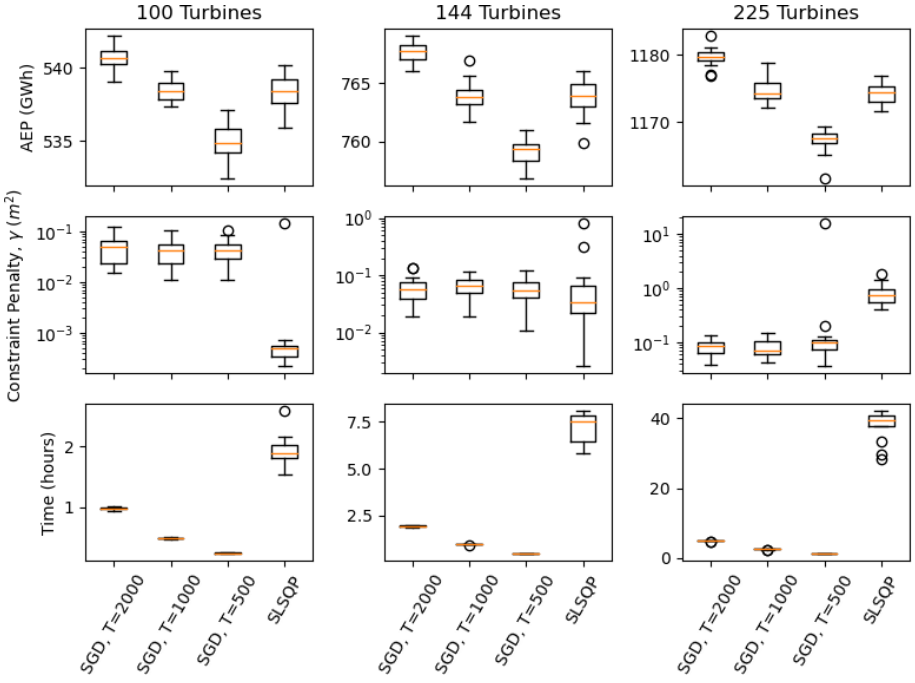

Figure 6. Optimization results associated with SGD and SLSQP for circular wind farms with 100, 144, and 225 turbines, using 20 random initial starting conditions. The AEP (top panels), constraint penalty (middle panels), and computational time (bottom panels) are plotted as box and whisker plots. The SGD results are plotted for $T = 500$, 1000, and 2000 iterations.

The results of the circular wind farm optimization are compared between the SGD and SLSQP optimizers in Table 2, where SGD is scheduled to run for 2000 optimization iterations. SGD generally results in about 0.5 % more AEPs in significantly less time than SLSQP as the number of turbines is increased. The result area is also compared in Fig. 7. The SGD optimizer generally results in more turbines on the boundary edge than the SLSQP optimizer.

4.3 Sensitivity analysis

There are several parameters in the SGD algorithm that were tuned to perform reasonably well. In this section, we investigate the sensitivity of the SGD optimization results with respect to the early stopping option in Algorithm 1, the number of Monte Carlo samples per optimization iteration, the learning rate schedule, and the initial and final learning rates.

Table 2. Results of SGD and deterministic optimizations for various circular wind farm sizes. Each optimization case is run using 20 random initial starting conditions, and the mean and standard deviation are reported with respect to these 20 initial points. The SGD results are associated with $T = 2000$ iterations.

N_t	Case	Mean time (hours)	Mean AEP (kWh)	AEP standard deviation (kWh)	Mean $\sqrt{\gamma/N_t}$ (m)
100	Deterministic	1.92	5.383×10^8	1.193×10^6	2.742×10^{-2}
	SGD	0.98	5.407×10^8	7.844×10^5	2.203×10^{-3}
144	Deterministic	7.25	7.638×10^8	1.512×10^6	8.648×10^{-2}
	SGD	1.96	7.676×10^8	8.777×10^5	1.726×10^{-3}
225	Deterministic	38.31	1.174×10^9	1.573×10^6	2.359×10^{-1}
	SGD	4.98	1.180×10^9	1.306×10^6	1.280×10^{-3}

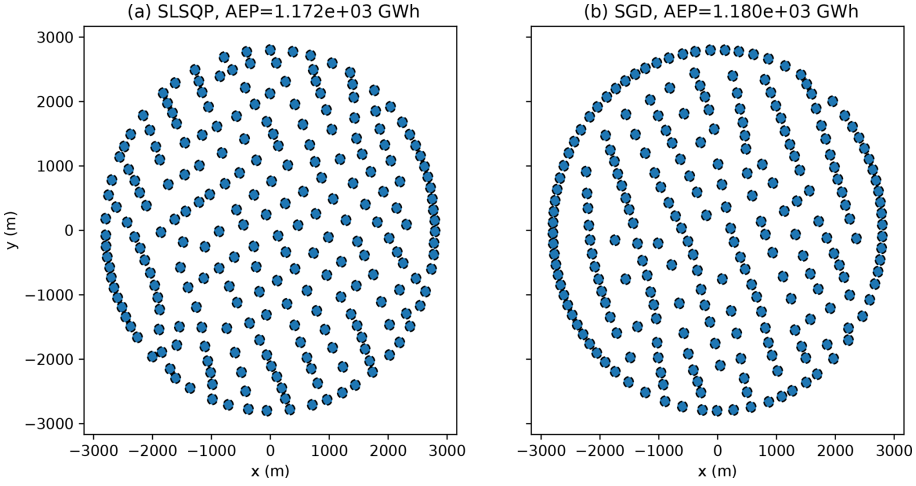

Figure 7. The final layouts found using SLSQP **(a)** and SGD **(b)** using one of the random initial layouts examined in the 225-turbine wind farm with circular boundaries using $T = 2000$ iterations. The final turbine layouts are shown as filled circles. Each circle has a radius of one rotor diameter.

As the optimization progresses, the constraint multiplier, α_i, becomes large (approaching 10 as the learning rate approaches 0.1), and the gradients of the AEP are overwhelmed by the gradients of the penalty, which take very little time to compute. This situation can be addressed by using the early stopping option in Algorithm 1. The solution tends to terminate quickly when the optimizer only follows the deterministic gradient (the optimization engine terminates when the constraint gradients are 0). Figure 8 shows the AEP, constraint violation, and computational time associated with five random initial layouts, using threshold parameters of 0.01, 0.05, and 0.1, as well as the SGD algorithm as applied in the previous sections, without the early stopping option activated. The use of each early stopping option results in layouts without constraint violations. As the threshold parameter is increased, the AEP is slightly reduced, and the total computational time decreases. A threshold parameter of 0.1 results in approximately 0.3 % reduction in AEP and 44 % reduction in computation time.

The optimization results presented in this study used 50 power samples per iteration ($K = 50$). We found this to produce high-quality results without incurring unacceptable computational expense. Figure 9 shows the behavior of the SGD approach associated with different values of K, considering 100 turbines with 2000 scheduled optimization iterations. As K increases, the optimization finds solutions with larger AEPs. There is a small increase in time elapsed and a large increase in the final AEP between the $K = 5$ and $K = 50$ cases, while there is a large increase in time elapsed and a small increase in the final AEP between $K = 50$ and $K = 200$. As K increases, we expect the maximum AEP to reach a plateau and the time and memory required to increase indefinitely. In future work, we plan to explore scheduling K to change as the optimization progresses.

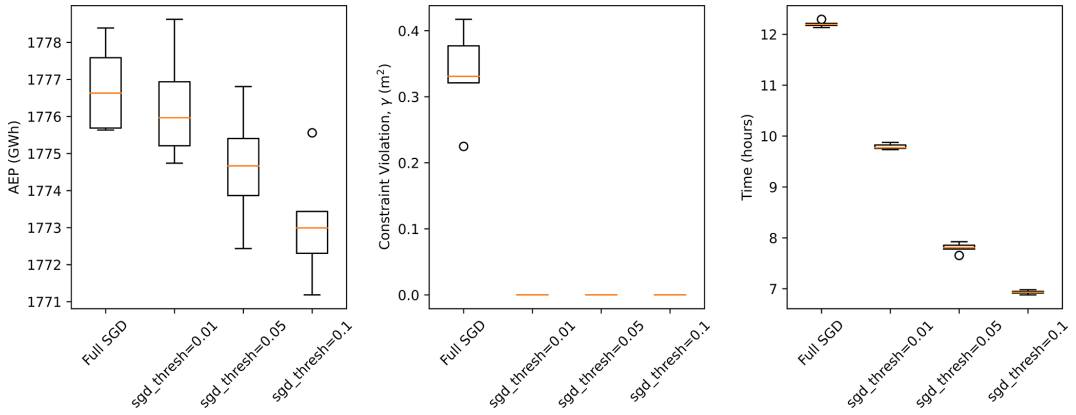

Figure 8. Results using different configurations of the early stopping option in TOPFARM, as well as the SGD optimization without early stopping, considering the circular wind farm with 225 turbines, with $T = 2000$, using five random initial layouts.

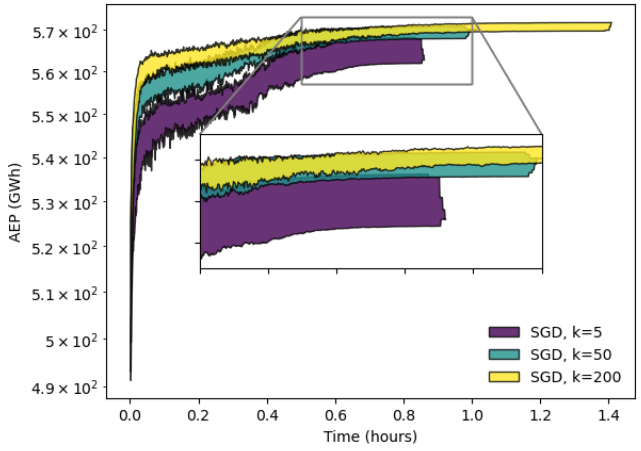

Figure 9. Optimization results associated with SGD for a square 100-turbine wind farm using 20 random initial starting conditions and $T = 2000$. The upper and lower bounds of the results are plotted as a function of the optimization iteration number. The SGD results associated with $K = 5$, 50, and 200 iterations are shown in purple, blue, and yellow, respectively.

This study used an exotic learning rate scheduler. We tried several schedulers and observed this one to be the best at finding sufficiently large AEP solutions that reasonably satisfied the imposed constraints. Figure 10 shows the behavior of the SGD algorithm associated with the presented learning rate scheduler, referred to here as the product scheduler, as well as an exponential and a linear decay scheduler. The exponential scheduler quickly diminishes the learning rate, causing the SGD algorithm to become stuck in local minima. The linear transition from large to lower learning rates prevents the SGD algorithm from having sufficient time to follow enlarged constraint gradients. It is possible that the algorithm could be improved by using separate schedulers for the learning rate and constraint multiplier. For instance, it might be more effective to use a linear scheduler to decrease the learning rate and an exponential scheduler to increase the constraint multiplier. We leave this question for future work.

The initial and final learning rates, η_0 and η_T, have units of distance and correspond to the initial and final step size of the optimization algorithm. The final learning rate can be interpreted as the degree to which the constraints are to be satisfied, since this will be the step size the optimization algorithm uses when α_i is large and the constraint gradients overwhelm the AEP gradients. This is illustrated in the left panel of Fig. 11, which shows the results of several SGD optimizations using different initial and final learning rates. On average, there is a linear relationship between the constraint violation of the solution and η_T, where the average final constraint violation is approximately 2 times η_T. The maximum observed constraint violation is approximately 4 times η_T. In addition, there is a trade-off between the AEP and constraint violation of the final solution. This trade-off is influenced by the initial and final learning rates. It is important to tune the initial learning rate. An initial learning rate that is too low will result in very little exploration. Initial learning rates that are too high will result in a rapid influx of penalty violations that overwhelm AEP gradients throughout the optimization. From our experiments, we found a step size of one-fifth of the rotor diameter to produce satisfactory results, as shown in the right panel of Fig. 11.

5 Conclusions

SGD is a promising optimization tool for wind farm design. Instead of evaluating all anticipated atmospheric conditions during every optimization iteration, SGD randomly samples the defined distributions of atmospheric conditions, resulting in substantially reduced computational time required for each optimization iteration. The total optimization time can be scheduled according to a prescribed computational budget. The presented formulation allows for continuous resolution of uncertain variables, eliminating the need to choose a

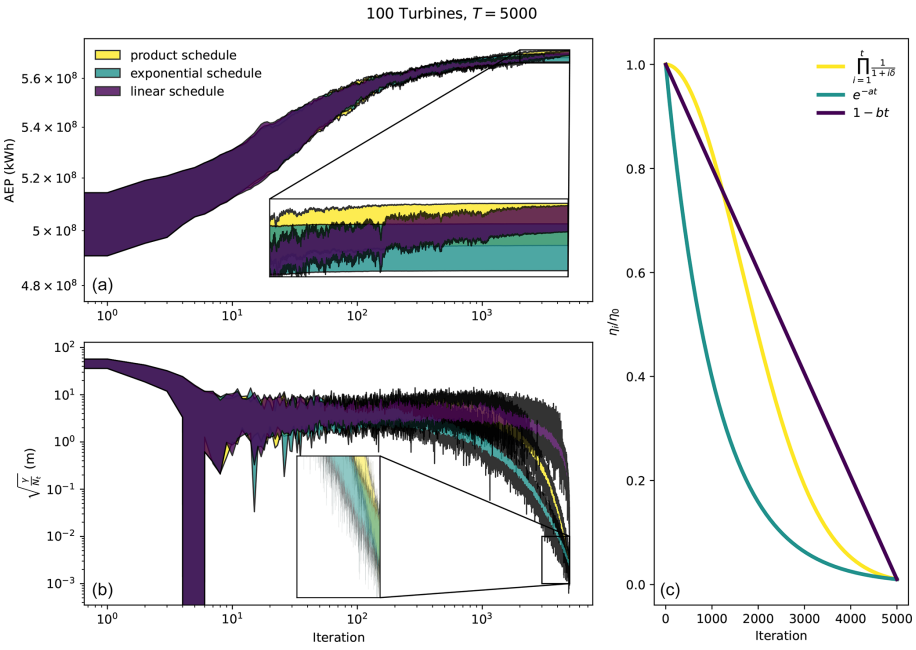

Figure 10. Influence of the learning rate scheduler on the SGD optimization, considering a 100-turbine wind farm with a square boundary. The product scheduler is shown in yellow. The exponential scheduler is shown in blue. The linear scheduler is shown in purple. The AEP **(a)**, constraint penalty function **(b)**, and the learning rate decay **(c)** are plotted as a function of the number of optimization iterations. The optimization iteration is denoted as t in the legend of panel **(c)**.

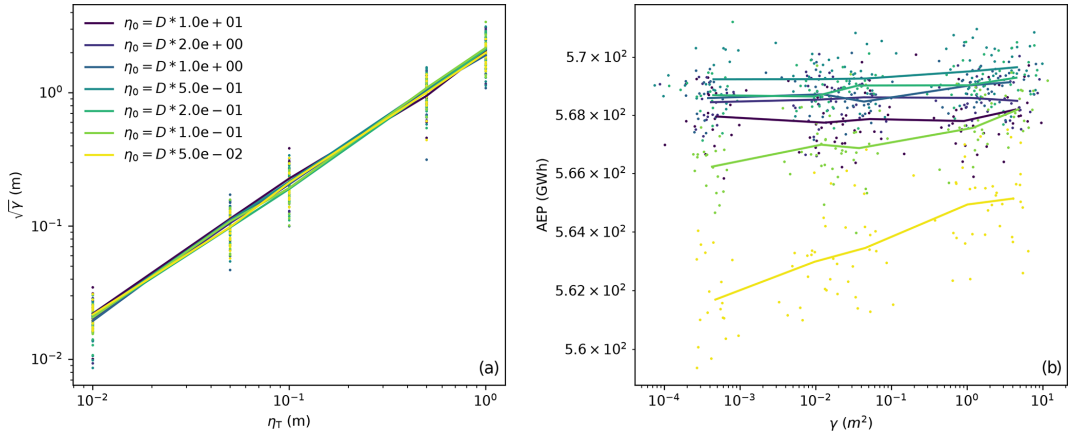

Figure 11. **(a)** The final constraint penalty plotted against the final learning rate. **(b)** The final AEP plotted against the final constraint penalty. The different colors represent different initial learning rates. The results of 20 initial starting positions are plotted as points. The average results of the 20 initial starting conditions are connected as lines. These data are associated with 100-turbine wind farms with square boundaries and $T = 2000$ iterations.

discretization resolution of atmospheric conditions, such as the wind speed and direction. This technique does not become exponentially more expensive as a greater number of uncertain parameters is included, allowing for consideration of other atmospheric conditions, such as turbulence intensity, air density, veer, and shear (Saint-Drenan et al., 2020; Duc et al., 2019).

The presented SGD approach was shown to become more effective than a deterministic counterpart as the number of wind turbines increased. SGD yielded slightly higher AEPs than the deterministic approach in substantially reduced computational time. The time required to optimize wind farm layouts can be a major bottleneck in corporate workflows, and the time savings associated with the SGD approach allows engineers to access optimization results sooner than a conventional approach. If the inflow conditions were discretized using extremely small bins, or if several atmospheric conditions were to be considered, we expect that the

SGD approach would perform the optimization even faster and more effectively than the deterministic approach.

The SGD approach is a simple framework that is well suited to large-scale stochastic wind power plant design optimization challenges. This framework is available in the open-source TOPFARM package. Future work includes exploring separate schedulers for the constraint multiplier and learning rate and scheduling the number of Monte Carlo samples, K, to change as the optimization proceeds.

Code availability. The code used in this study is available from DTU Wind Energy Systems' PyWake and TOPFARM repositories (https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake, DTU Wind Energy Systems, 2023a; and https://gitlab.windenergy.dtu.dk/TOPFARM/Topfarm2, DTU Wind Energy Systems, 2023b.)

Data availability. The data used in this study have been made available on Zenodo: https://doi.org/10.5281/zenodo.8202150 (Quick, 2023).

Author contributions. JQ and PER designed the experiments. JQ, PER, MMP, and RVR developed the problem formulation. JQ developed the SGD formulation. JQ, RVR, MFM, and MMP performed the simulations. JQ, MMP, and RVP prepared the paper with contributions from all co-authors.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The authors gratefully acknowledge the computational and data resources provided on the Sophia HPC cluster at the Technical University of Denmark, https://doi.org/10.57940/FAFC-6M81.

Review statement. This paper was edited by Cristina Archer and reviewed by Ahmad Vasel-Be-Hagh and two anonymous referees.

References

Alibrahim, H. and Ludwig, S. A.: Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization, in: 2021 IEEE Congress on Evolutionary Computation (CEC), 28 June–1 July 2021, Kraków, Poland, 1551–1559, https://doi.org/10.1109/CEC45853.2021.9504761, 2021.

Allen, J., King, R., and Barter, G.: Wind farm simulation and layout optimization in complex terrain, J. Phys.: Conf. Ser., 1452, 012066, https://doi.org/10.1088/1742-6596/1452/1/012066, 2020.

Annoni, J., Fleming, P., Scholbrock, A., Roadman, J., Dana, S., Adcock, C., Porte-Agel, F., Raach, S., Haizmann, F., and Schlipf, D.: Analysis of control-oriented wake modeling tools using lidar field results, Wind Energ. Sci., 3, 819–831, https://doi.org/10.5194/wes-3-819-2018, 2018.

Baker, N. F., Stanley, A. P., Thomas, J. J., Ning, A., and Dykes, K.: Best practices for wake model and optimization algorithm selection in wind farm layout optimization, in: AIAA Scitech 2019 forum, 7–11 January 2019, San Diego, California, USA, p. 0540, https://doi.org/10.2514/6.2019-0540, 2019.

Bastankhah, M. and Porté-Agel, F.: A new analytical model for wind-turbine wakes, Renew. Energy, 70, 116–123, https://doi.org/10.1016/j.renene.2014.01.002, 2014.

Byrd, R. H., Hansen, S. L., Nocedal, J., and Singer, Y.: A stochastic quasi-Newton method for large-scale optimization, SIAM J. Optimiz., 26, 1008–1031, https://doi.org/10.1137/140954362, 2016.

Ciavarra, A. W., Rodrigues, R. V., Dykes, K., and Réthoré, P.-E.: Wind farm optimization with multiple hub heights using gradient-based methods, J. Phys.: Conf. Ser., 2265, 022012, https://doi.org/10.1088/1742-6596/2265/2/022012, 2022.

Clark, C. E., Barter, G., Shaler, K., and DuPont, B.: Reliability-based layout optimization in offshore wind energy systems, Wind Energy, 25, 125–148, https://doi.org/10.1002/we.2664, 2022.

Criado Risco, J., Valotta Rodrigues, R., Friis-Møller, M., Quick, J., Mølgaard Pedersen, M., and Réthoré, P.-E.: Gradient-based Wind Farm Layout Optimization With Inclusion And Exclusion Zones, Wind Energ. Sci. Discuss. [preprint], https://doi.org/10.5194/wes-2023-5, in review, 2023.

Croonenbroeck, C. and Hennecke, D.: A comparison of optimizers in a unified standard for optimization on wind farm layout optimization, Energy, 216, 119244, https://doi.org/10.1016/j.energy.2020.119244, 2021.

De, S., Hampton, J., Maute, K., and Doostan, A.: Topology optimization under uncertainty using a stochastic gradient-based approach, Struct. Multidiscip. Optimiz., 62, 2255–2278, https://doi.org/10.1007/s00158-020-02599-z, 2020.

Denkowski, M. and Neubig, G.: Stronger Baselines for Trustable Results in Neural Machine Translation, in: Proceedings of the First Workshop on Neural Machine Translation, Association for Computational Linguistics, Vancouver, 18–27, https://doi.org/10.18653/v1/W17-3203, 2017.

DTU Wind Energy Systems: PyWake, DTU Wind Energy [code], https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake (last access: 31 July 2023), 2023a.

DTU Wind Energy Systems: TOPFARM, DTU Wind Energy [code], https://gitlab.windenergy.dtu.dk/TOPFARM/Topfarm2 (last access: 31 July 2023), 2023b.

Duc, T., Coupiac, O., Girard, N., Giebel, G., and Göçmen, T.: Local turbulence parameterization improves the Jensen wake model and its implementation for power optimization of an operating wind farm, Wind Energ. Sci., 4, 287–302, https://doi.org/10.5194/wes-4-287-2019, 2019.

Feng, J. and Shen, W. Z.: Solving the wind farm layout optimization problem using random search algorithm, Renew. Energy, 78, 182–192, https://doi.org/10.1016/j.renene.2015.01.005, 2015.

Fischereit, J., Schaldemose Hansen, K., Larsén, X. G., van der Laan, M. P., Réthoré, P.-E., and Murcia Leon, J. P.: Comparing and validating intra-farm and farm-to-farm wakes across different mesoscale and high-resolution wake models, Wind Energ. Sci., 7, 1069–1091, https://doi.org/10.5194/wes-7-1069-2022, 2022.

Fleming, P. A., Stanley, A. P., Bay, C. J., King, J., Simley, E., Doekemeijer, B. M., and Mudafort, R.: Serial-Refine Method for Fast Wake-Steering Yaw Optimization, J. Phys.: Conf. Ser., 2265, 032109, https://doi.org/10.1088/1742-6596/2265/3/032109, 2022.

Gebraad, P., Thomas, J. J., Ning, A., Fleming, P., and Dykes, K.: Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control, Wind Energy, 20, 97–107, https://doi.org/10.1002/we.1993, 2017.

Göçmen, T. and Giebel, G.: Estimation of turbulence intensity using rotor effective wind speed in Lillgrund and Horns Rev-I offshore wind farms, Renew. Energy, 99, 524–532, https://doi.org/10.1016/j.renene.2016.07.038, 2016.

Godinho, M. and Castro, R.: Comparative performance of AI methods for wind power forecast in Portugal, Wind Energy, 24, 39–53, https://doi.org/10.1002/we.2556, 2021.

Graf, P., Dykes, K., Scott, G., Fields, J., Lunacek, M., Quick, J., and Rethore, P.-E.: Wind farm turbine type and placement optimization, J. Phys.: Conf. Ser., 753, 062004, https://doi.org/10.1088/1742-6596/753/6/062004, 2016.

Guirguis, D., Romero, D. A., and Amon, C. H.: Toward efficient optimization of wind farm layouts: Utilizing exact gradient information, Appl. Energy, 179, 110–123, https://doi.org/10.1016/j.apenergy.2016.06.101, 2016.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., G'erard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E.: Array programming with NumPy, Nature, 585, 357–362, https://doi.org/10.1038/s41586-020-2649-2, 2020.

Hasager, C. B., Rasmussen, L., Peña, A., Jensen, L. E., and Réthoré, P.-E.: Wind farm wake: The Horns Rev photo case, Energies, 6, 696–716, https://doi.org/10.3390/en6020696, 2013.

Herbert-Acero, J. F., Probst, O., Réthoré, P.-E., Larsen, G. C., and Castillo-Villar, K. K.: A review of methodological approaches for the design and optimization of wind farms, Energies, 7, 6930–7016, https://doi.org/10.3390/en7116930, 2014.

Howland, M. F., Ghate, A. S., Quesada, J. B., Pena Martínez, J. J., Zhong, W., Larrañaga, F. P., Lele, S. K., and Dabiri, J. O.: Optimal closed-loop wake steering – Part 2: Diurnal cycle atmospheric boundary layer conditions, Wind Energ. Sci., 7, 345–365, https://doi.org/10.5194/wes-7-345-2022, 2022.

Hussain, M. N., Shaukat, N., Ahmad, A., Abid, M., Hashmi, A., Rajabi, Z., and Tariq, M. A. U. R.: Micro-Siting of Wind Turbines in an Optimal Wind Farm Area Using Teaching–Learning-Based Optimization Technique, Sustainability, 14, 8846, https://doi.org/10.3390/su14148846, 2022.

International Electrotechnical Commission: IEC 61400-12-1 Wind Turbines-Part 12-1: Power Performance Measurements of Electricity Producing Wind Turbines, IEC – International Electrotechinal Commission, Geneva, Switzerland, 1, https://webstore.iec.ch/publication/68499 (last access: 31 July 2023), 2005.

Kervadec, H., Dolz, J., Yuan, J., Desrosiers, C., Granger, E., and Ayed, I. B.: Constrained deep networks: Lagrangian optimization via log-barrier extensions, arXiv [preprint], arXiv:1904.04205, https://doi.org/10.48550/arXiv.1904.04205, 2019.

Ketkar, N.: Stochastic gradient descent, in: Deep learning with Python, Springer, 113–132, https://doi.org/10.1007/978-1-4842-2766-4_8, 2017.

King, R., Glaws, A., Geraci, G., and Eldred, M. S.: A probabilistic approach to estimating wind farm annual energy production with bayesian quadrature, in: AIAA Scitech 2020 Forum, 6–10 January 2020, Orlando, Florida, USA, p. 1951, https://doi.org/10.2514/6.2020-1951, 2020.

King, R. N., Dykes, K., Graf, P., and Hamlington, P. E.: Optimization of wind plant layouts using an adjoint approach, Wind Energ. Sci., 2, 115–131, https://doi.org/10.5194/wes-2-115-2017, 2017.

Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, arXiv [preprint], arXiv:1412.6980, https://doi.org/10.48550/arXiv.1412.6980, 2014.

Kölle, K., Göçmen, T., Eguinoa, I., Alcayaga Román, L. A., Aparicio-Sanchez, M., Feng, J., Meyers, J., Pettas, V., and Sood, I.: FarmConners market showcase results: wind farm flow control considering electricity prices, Wind Energ. Sci., 7, 2181–2200, https://doi.org/10.5194/wes-7-2181-2022, 2022.

Kraft, D.: A software package for sequential quadratic programming, Forschungsbericht, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, http://degenerateconic.com/uploads/2018/03/DFVLR_FB_88_28.pdf (last access: 31 July 2023), 1988.

Lam, R., Poloczek, M., Frazier, P., and Willcox, K. E.: Advances in Bayesian optimization with applications in aerospace engineering, in: 2018 AIAA Non-Deterministic Approaches Conference, 8–12 January 2018, Kissimmee, Florida, p. 1656, https://doi.org/10.2514/6.2018-1656, 2018.

Li, J. and Zhang, M.: Data-based approach for wing shape design optimization, Aerospace Sci. Technol., 112, 106639, https://doi.org/10.1016/j.ast.2021.106639, 2021.

Liu, J., Rong, Y., Takác, M., and Huang, J.: On the acceleration of l-bfgs with second-order information and stochastic batches, arXiv [preprint], arXiv:1807.05328, https://doi.org/10.48550/arXiv.1807.05328, 2018.

Márquez-Neila, P., Salzmann, M., and Fua, P.: Imposing hard constraints on deep networks: Promises and limitations, arXiv [preprint], arXiv:1706.02025, https://doi.org/10.48550/arXiv.1706.02025, 2017.

Moritz, P., Nishihara, R., and Jordan, M.: A linearly-convergent stochastic L-BFGS algorithm, in: Artificial Intelligence and

Statistics, PMLR, 249–258, https://proceedings.mlr.press/v51/moritz16.html (last access: 31 July 2023), 2016.

Murcia, J., Réthoré, P.-E., Natarajan, A., and Sørensen, J. D.: How many model evaluations are required to predict the AEP of a wind power plant?, J. Phys.: Conf. Ser., 625, 012030, https://doi.org/10.1088/1742-6596/625/1/012030, 2015.

Najafabadi, M. M., Khoshgoftaar, T. M., Villanustre, F., and Holt, J.: Large-scale distributed l-bfgs, J. Big Data, 4, 1–17, https://doi.org/10.1186/s40537-017-0084-5, 2017.

Najd, A. H., Goksu, G., and Hammood, H. F.: Pitch angle control using neural network in wind turbines, Mater. Sci. Eng., 928, 022118, https://doi.org/10.1088/1757-899X/928/2/022118, 2020.

Ning., A., Dykes., K., and Quick, J.: Systems engineering and optimization of wind turbines and power plants, Wind Energy Modeling and Simulation – Volume 2: Turbine and System, Institution of Engineering and Technology, 235–292, https://doi.org/10.1049/pbpo125g_ch7, 2020.

Padrón, A. S., Thomas, J., Stanley, A. P., Alonso, J. J., and Ning, A.: Polynomial chaos to efficiently compute the annual energy production in wind farm layout optimization, Wind Energ. Sci., 4, 211–231, https://doi.org/10.5194/wes-4-211-2019, 2019.

Pedersen, M. M., van der Laan, P., Friis-Møller, M., Rinker, J., and Réthoré, P.-E.: DTUWindEnergy/PyWake: PyWake, Zenodo [code], https://doi.org/10.5281/zenodo.2562662, 2019.

Qian, N.: On the momentum term in gradient descent learning algorithms, Neural Networks, 12, 145–151, https://doi.org/10.1016/S0893-6080(98)00116-6, 1999.

Quick, J.: Data Used for Article: Stochastic Gradient Descent for Wind Farm Optimization, Zenodo [data set], https://doi.org/10.5281/zenodo.8202150, 2023.

Quick, J., King, J., King, R. N., Hamlington, P. E., and Dykes, K.: Wake steering optimization under uncertainty, Wind Energ. Sci., 5, 413–426, https://doi.org/10.5194/wes-5-413-2020, 2020.

Riedmiller, M. and Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in: IEEE international conference on neural networks, 28 March–1 April 1993, San Francisco, California, USA, 586–591, https://doi.org/10.1109/ICNN.1993.298623, 1993.

Riva, R., Liew, J., Friis-Møller, M., Dimitrov, N., Barlas, E., Réthoré, P.-E., and Beržonskis, A.: Wind farm layout optimization with load constraints using surrogate modelling, J. Phys.: Conf. Ser., 1618, 042035, https://doi.org/10.1088/1742-6596/1618/4/042035, 2020.

Rodrigues, R. V., Friis-Møller, M., Dykes, K., Pollini, N., and Jensen, M.: A surrogate model of offshore wind farm annual energy production to support financial valuation, J. Phys.: Conf. Ser., 2265, 022003, https://doi.org/10.1088/1742-6596/2265/2/022003, 2022.

Rodrigues, R. V., Pedersen, M. M., Schøler, J. P., Quick, J., and Réthoré, P.: Speeding up large wind farms layout optimization using gradients, parallelization, and a heuristic algorithm for the initial layout, Wind Energ. Sci. Discuss. [preprint], https://doi.org/10.5194/wes-2023-61, in review, 2023.

Roy, S. K. and Harandi, M.: Constrained Stochastic Gradient Descent: The Good Practice, in: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 29 November–1 December 2017, Sydney, NSW, Australia, 1–8, https://doi.org/10.1109/DICTA.2017.8227420, 2017.

Ruder, S.: An overview of gradient descent optimization algorithms, arXiv [preprint], https://doi.org/10.48550/ARXIV.1609.04747, 2016.

Saint-Drenan, Y.-M., Besseau, R., Jansen, M., Staffell, I., Troccoli, A., Dubus, L., Schmidt, J., Gruber, K., Simões, S. G., and Heier, S.: A parametric model for wind turbine power curves incorporating environmental conditions, Renew. Energy, 157, 754–768, https://doi.org/10.1016/j.renene.2020.04.123, 2020.

Samorani, M.: The wind farm layout optimization problem, Handbook of wind power systems, Springer, 21–38, https://doi.org/10.1007/978-3-642-41080-2_2, 2013.

Sanderse, B.: Aerodynamics of wind turbine wakes, US Department of Energy Office of Scientific and Technical Information, https://www.osti.gov/etdeweb/biblio/21162007 (last access: 31 July 2023), 2009.

Simley, E., Millstein, D., Jeong, S., and Fleming, P.: The value of wake steering wind farm control in U.S. energy markets, Wind Energ. Sci. Discuss. [preprint], https://doi.org/10.5194/wes-2023-12, in review, 2023.

Sivanantham, G. and Gopalakrishnan, S.: Stochastic Gradient Descent Optimization Model for Demand Response in a Connected Microgrid, KSII Transactions on Internet and Information Systems (TIIS), 16, 97–115, https://doi.org/10.3837/tiis.2022.01.006, 2022.

Stanley, A. P., Roberts, O., King, J., and Bay, C. J.: Objective and algorithm considerations when optimizing the number and placement of turbines in a wind power plant, Wind Energ. Sci., 6, 1143–1167, https://doi.org/10.5194/wes-6-1143-2021, 2021.

Stengel, K., Glaws, A., Hettinger, D., and King, R. N.: Adversarial super-resolution of climatological wind and solar data, P. Natl. Acad. Sci. USA, 117, 16805–16815, https://doi.org/10.1073/pnas.1918964117, 2020.

Technical University of Denmark: Sophia HPC Cluster, https://doi.org/10.57940/FAFC-6M81, 2019.

Tian, Y., Zhang, Y., and Zhang, H.: Recent Advances in Stochastic Gradient Descent in Deep Learning, Mathematics, 11, 682, https://doi.org/10.3390/math11030682, 2023.

van der Laan, M. P., García-Santiago, O., Kelly, M., Meyer Forsting, A., Dubreuil-Boisclair, C., Sponheim Seim, K., Imberger, M., Peña, A., Sørensen, N. N., and Réthoréé, P.-E.: A new RANS-based wind farm parameterization and inflow model for wind farm cluster modeling, Wind Energ. Sci., 8, 819–848, https://doi.org/10.5194/wes-8-819-2023, 2023.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nat. Meth., 17, 261–272, https://doi.org/10.1038/s41592-019-0686-2, 2020.

Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R.: pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems, J. Open Sour. Softw., 5, 2564, https://doi.org/10.21105/joss.02564, 2020.

You, K., Long, M., Wang, J., and Jordan, M. I.: How does learning rate decay help modern neural networks?, arXiv [preprint], arXiv:1908.01878, https://doi.org/10.48550/arXiv.1908.01878, 2019.

Zhang, C., Kramer, S. C., Angeloudis, A., Zhang, J., Lin, X., and Piggott, M. D.: Improving tidal turbine array performance through the optimisation of layout and yaw angles, Int. Mar. Energ. J., 5, 273–280, https://doi.org/10.36688/imej.5.273-280, 2022.

Zhang, J. and Zhao, X.: Wind farm wake modeling based on deep convolutional conditional generative adversarial network, Energy, 238, 121747, https://doi.org/10.1016/j.energy.2021.121747, 2022.

Zhang, Z., Santoni, C., Herges, T., Sotiropoulos, F., and Khosronejad, A.: Time-averaged wind turbine wake flow field prediction using autoencoder convolutional neural networks, Energies, 15, 41, https://doi.org/10.3390/en15010041, 2021.

Zilong, T. and Wei, D. X.: Layout optimization of offshore wind farm considering spatially inhomogeneous wave loads, Appl. Energy, 306, 117947, https://doi.org/10.1016/j.apenergy.2021.117947, 2022.