

PAPER • OPEN ACCESS

Wind farm layout optimization with load constraints using surrogate modelling

To cite this article: Riccardo Riva *et al* 2020 *J. Phys.: Conf. Ser.* **1618** 042035

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Wind farm layout optimization with load constraints using surrogate modelling

Riccardo Riva¹, Jaime Liew¹, Mikkel Friis-Møller¹, Nikolay Dimitrov¹, Emre Barlas², Pierre-Elouan Réthoré¹ and Arvydas Beržonskis²

¹Department of Wind Energy, Technical University of Denmark (DTU), Frederiksborgvej 399, 4000 Roskilde, Denmark.

²Goldwind Energy ApS, Delta 4, 8382 Hinnerup, Denmark.

E-mail: {ricriv, jyli, mikf, nkdi, pire}@dtu.dk, abe@goldwinddenmark.com

Abstract. Minimizing the cost of energy of a wind farm is a difficult task, which involves reducing the wake effects while satisfying several constraints. Due to its multidisciplinary nature, this problem is usually solved through numerical optimisers. TOPFARM is one of these tools, and in this paper, we have added to it a constraint on the fatigue loads. The efficiency of the implementation is guaranteed by an extensive use of gradients and load surrogate models. The paper is concluded by showing some case studies.

1. Introduction

Designing a wind farm is a particularly challenging task, which requires addressing several problems related to the site conditions and wind turbine wakes. Depending on the wind direction, a wind turbine might operate in free flow, partial wake or full wake. In the free flow case, the turbines do not interfere with each other. In partial wake, the blades operate periodically in and out of the wind deficit, which causes a severe increase of the fatigue loads. In full wake, the turbines experience an overall slower wind with a higher turbulence intensity, which causes a strong reduction in power output and higher fatigue loads [1, 2]. The design of a wind farm should consider these factors, and weight their contribution through the wind rose and wind speed distribution. For this reason, there is a growing interest in wind farm design optimization tools with the goal of minimizing the Levelized Cost Of Energy (LCOE).

The tools for optimizing wind farms rely on several interconnected components [3, 4]. Typically, a flow component provides wind-related quantities for a given site and turbine layout. Given the wind characteristics, the turbine component estimates the power and loads. Other components provide additional quantities, like the domain boundaries, turbine distances, and cost. Finally, a numerical optimizer minimizes the objective function, while satisfying the constraints.

LCOE is typically minimized by increasing the Annual Energy Production (AEP), by reducing the transportation and installation costs, or by changing the number of turbines and their types. The optimization problem can depend on the turbine loads for two reasons. First, the fatigue loads are a well known design driver for some components, hence they should be constrained to be within safety limits. Secondly, the fatigue loads might affect the operation and maintenance



(O&M) costs, see Ref. [5] and related publications on O&M cost estimators. In this case, the fatigue loads might be embedded in the cost model, or used for a multi-objective optimization together with the AEP. Alternatively, they can again be constrained, to keep the O&M cost under a certain level. Reducing layout-induced fatigue loads is achieved by minimizing wake effects, which has also a positive effect on the AEP. A possibility is thus to maximize the AEP and then conduct a detailed load analysis a posteriori. If any of the turbines will exceed the design loads, the solution is discarded, and a new optimization will have to be carried out. This heuristic method is clearly sub-optimal, and has unpredictable effects on the convergence of the optimization problem. A better solution consists of including loads directly in the optimization routine by implementing load-driven constraints, to maintain them within the design limit during the whole optimization.

Since the load constraint has to be evaluated several times during the optimization process, it is computationally infeasible to repeatedly carry out full wind farm load simulations. We thus resort to estimating the loads through surrogate models trained on pre-simulated load scenarios. These surrogates are nonlinear regression models, which allow for quick load evaluations, a smooth dependence from the input parameters, and often can accommodate gradient-based optimizations. They normally depend only on the individual turbine model, and hence an arbitrary wind farm layout can be evaluated with a single surrogate model.

In this work, we have implemented a load constraint in TOPFARM, the DTU Wind Energy software platform for optimization of wind farm design and operational strategies in both onshore and offshore applications. This new functionality allows designing wind farms subject to constraints on the Lifetime Damage Equivalent Loads (LDEL). These constraints can exclude regions of the site, and layouts, that would otherwise generate excessive loads.

In Sec. 2 we illustrate the TOPFARM objectives and architecture. Sec. 3 shows some case studies and finally in Sec. 4 we draw the conclusions.

2. The TOPFARM platform

In Ref. [3] Réthoré *et al.* presented the first version of TOPFARM, a tool that allows optimizing wind farms layout using a multi-fidelity approach. This tool influenced several works, and over the last few years, its design principles evolved, to improve its functionalities and streamline the development. This new version is implemented in Python instead of MATLAB®, and many of its components are open source. TOPFARM is formed by several components, that allow a high degree of flexibility in designing the optimization problems. The main components are: PyWake for estimating the AEP, and TopFarm2 for formulating the optimization problem. In the remainder of the section, we will describe some of its components.

2.1. PyWake

PyWake [6] is an open-source AEP calculator for wind farms, similar to FLORIS [7] in its modular composition and in its capability to be used in wind farm optimization. It is composed of five components, shown in Fig. 1 and summarized hereafter.

Site Given a reference wind speed and direction, this component provides the local wind speed, wind direction, turbulence intensity and probability for each point. It can model both flat and complex terrains.

WakeModel Provides the effective wind speed at each location, by proceeding in down-wind order. It implements the wake models by Niels Otto Jensen (NOJ) and Bastankhah [8, 9], and can also interact with Fuga [10].

WindTurbines Estimates the power and thrust coefficient of each wind turbine. One of the models already implemented is the DTU 10 MW RWT [11].

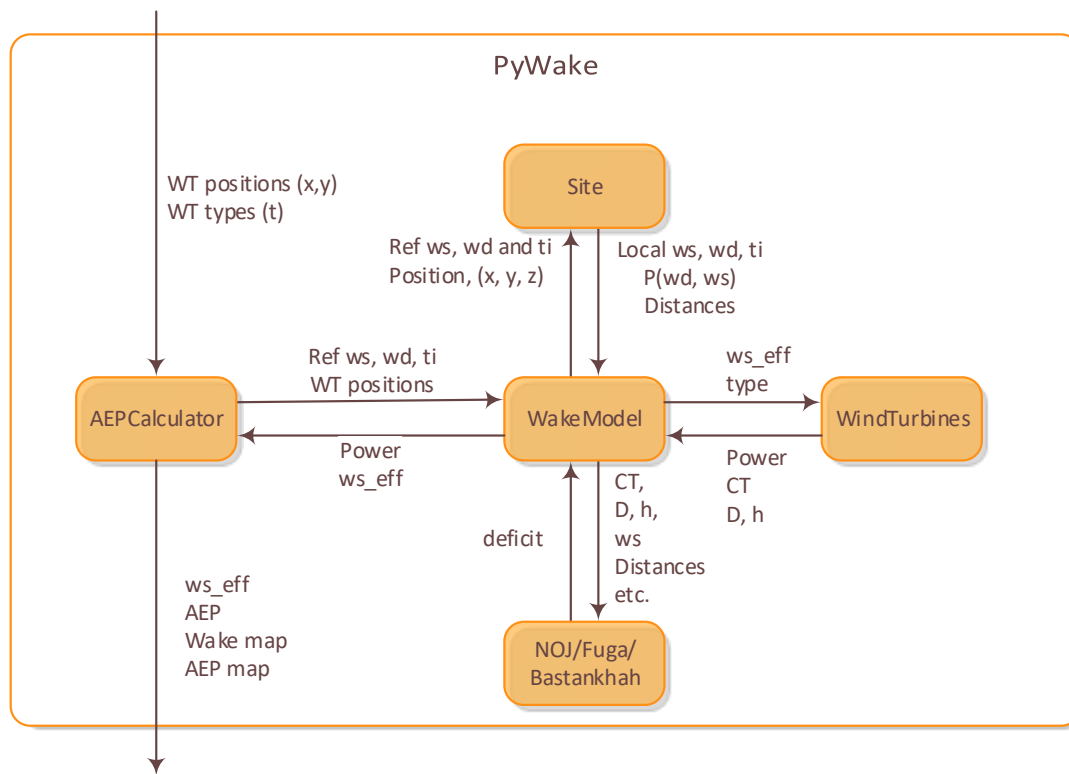


Figure 1. Architecture of PyWake.

AEP Calculator Computes the AEP by summing the individual turbines' power for each wind speed and wind direction and weighting them by the respective probability.

PyWake has been developed using an object-oriented paradigm, focusing on modularity and speed, achieved through extensive use of vector operations. In its basic version it uses engineering models, but in Ref. [12] has been coupled with EllipSys3D to enable RANS-based CFD.

2.2. TopFarm2

During the last few years, there has been an increasing need for tools able to efficiently perform multidisciplinary optimizations. **OpenMDAO** [13] has been created to address this problem, and due to its versatility and speed, it quickly became one of the most popular choices. **TopFarm2** [14] has been built on top of it, to tailor **OpenMDAO** to wind farm optimization problems. **OpenMDAO** can be used with different optimization algorithms, including Genetic Algorithms, but it is best used with gradient-based ones, hence many components of **TopFarm2** have been linearized. An optimization problem can also be restarted, to change the convergence tolerances, and possibly the optimization algorithm.

The design variables in **TopFarm2** are the number of turbines, their position, and types. A parametrization like the one in Ref. [15] could be introduced by adding an ad hoc component. **TopFarm2** can maximize the AEP, but also minimize the LCOE using the NREL cost model [16]. Spacing and various kinds of boundary constraints are available and support gradients. A capacity constraint has been added, for problems that involve the number of turbines and types. A reimplementation of the minimum spanning tree algorithm, from Ref. [3], allows minimizing roads and cables cost.

The initial turbine locations can be user-defined, randomly selected, or chosen through a heuristic algorithm. This algorithm receives a scalar function, for example the local AEP

(without wake losses) discretized over a set of coordinates, and places the turbines on the maximum of this function, while respecting the spacing constraint.

2.3. Load constraint

Numerical simulations produce load time histories, which are typically converted into DELs through the rainflow-counting algorithm. Since these DEL depend not only on the load channel, but also on the flow case, it would be impractical to constraint all of them. For this reason, the DELs for all flow cases are aggregated into LDELs, which depend only on the load channel. A constraint on the LDELs was already implemented in the original TOPFARM, but it was based on lookup tables, and hence inadequate for gradient-based optimizations. For the present implementation, we have focused on two aspects. First, to leverage the capabilities of OpenMDAO in assembling the total Jacobian, the load constraint has been separated in its basic components, and secondly, we have relied on surrogate models. The components are: WakeComp for computing the AEP, LoadComp for estimating the loads and LifetimeLoadComp for computing the LDEL.

Computing the wake-induced loads requires a wake model for load estimation, and for this work we have selected the Frandsen equivalent turbulence approach [17]. This choice is motivated by the fact that it is described in the IEC 61400-1, and differently from the Dynamic Wake Meandering model, it does not require a wake aggregation strategy, and thus leads to a simpler implementation.

WakeComp receives the turbine locations $\mathbf{x}_i \in \mathbb{R}^2$, for each turbine $i \in [1, \dots, n_t]$, and uses PyWake to compute the flow and then the AEP. Since PyWake is not linearized yet, the gradient of the AEP is estimated with finite differences. By indicating with θ a wind direction and with V_{ref} a reference wind speed, the effective wind speed is $V_{\text{eff}}(\mathbf{x}_i, \theta, V_{\text{ref}})$. The probability $p(\mathbf{x}_i, \theta, V_{\text{ref}})$ is integrated over all wind directions, to provide

$$p(\mathbf{x}_i, V_{\text{ref}}) = \int_0^{2\pi} p(\mathbf{x}_i, \theta, V_{\text{ref}}) d\theta. \quad (1)$$

Similarly, the effective wind speed is weighted over the wind directions

$$V_{\text{eff}}(\mathbf{x}_i, V_{\text{ref}}) = \frac{1}{p(\mathbf{x}_i, V_{\text{ref}})} \int_0^{2\pi} p(\mathbf{x}_i, \theta, V_{\text{ref}}) V_{\text{eff}}(\mathbf{x}_i, \theta, V_{\text{ref}}) d\theta. \quad (2)$$

For a Wöhler (material) exponent m , and reference turbulence intensity $I_{\text{ref}}(\mathbf{x}_i, \theta, V)$, the effective turbulence intensity is computed as

$$I_{\text{eff}}(\mathbf{x}_i, V_{\text{ref}}) = \left(\int_0^{2\pi} p(\mathbf{x}_i, \theta, V_{\text{ref}}) I_{\text{ref}}^m(\mathbf{x}_i, \theta, V) d\theta \right)^{1/m}. \quad (3)$$

The effective wind speed, effective turbulence intensity, and all other inputs of the surrogate models are passed to LoadComp, which evaluates the surrogate models and outputs the DELs. In the following, we will indicate the 1 Hz DEL for a generic load channel with $M(\mathbf{x}_i, V_{\text{ref}})$. LoadComp outputs the DELs for all turbines, load channels, and all combinations of input variables. Since this might result in a very large Jacobian matrix, particular care has been posed on determining its sparsity pattern.

Fitting wind turbines loads with surrogate models have been addressed in several papers, see for example [18] and references therein. Since each surrogate model type can perform differently depending on the particular condition, TopFarm2 allows to predict the loads, and their gradient, using a generic interface. Wrappers are available for the OpenTURNS [19], TensorFlow [20] and Wind2Loads [2] libraries. The supported model types are listed in Table 1.

Table 1. Types of surrogate models supported by TopFarm2.

| Type | Library | Gradient support |
|----------------------------|--------------|------------------|
| Polynomial Chaos Expansion | OpenTURNS | ✓ |
| Kriging | OpenTURNS | ✓ |
| Artificial Neural Network | TensorFlow | ✓ |
| | scikit-learn | ✗ |
| | Wind2Loads | ✓ |

The generation of the surrogate model is done using any of those libraries, following the procedure described in Ref. [18]. When defining the training dataset, particular care must be posed on the selection of the input variables, which is site-dependent. To limit the number of aeroelastic simulations for model training, and also not stress the capabilities of the surrogate model, the number of input variables should be kept to a minimum.

Since the surrogate models are used within an optimization context, choosing their type cannot be based solely on accuracy consideration, but also on the evaluation speed. In our experience, kriging, which is an interpolation method requiring the assembly of a cross-correlation matrix before each evaluation, is the slowest, while artificial neural networks are the fastest.

LifetimeLoadComp receives all the DEL and probabilities, and computes the LDEL, according to

$$L(\mathbf{x}_i) = \left(\phi \int_{V_{\text{ref}}} p(\mathbf{x}_i, V_{\text{ref}}) M(\mathbf{x}_i, V_{\text{ref}})^m dV_{\text{ref}} \right)^{1/m}, \quad (4)$$

with ϕ the number of seconds in 20 years, divided by the number of cycles, typically 10^7 . The integral is computed using the Monte Carlo method [18]. To improve the conditioning of the problem, each LDEL is normalized with respect to a reference value, dependent on the load channel. For example, the design limit, or the maximum LDEL at the initial guess. Similarly to the previous component, also for LifetimeLoadComp, the Jacobian matrix is allocated in a sparse format.

3. Case studies

For developing the case studies we have chosen the DTU 10 MW RWT. Numerical simulations using the HAWC2 aeroservoelastic multibody software [21] have been conducted in Ref. [18]. The input variables are: wind speed, turbulence intensity, wind shear exponent, turbulence length scale, anisotropy factor, wind veer, yaw angle, terrain inclination, and air density. 1 Hz DEL for several load channels have been computed using the rainflow-counting algorithm. The reliability of the estimates is ensured by having simulated 48 turbulence seeds.

Artificial Neural Networks have been trained using the Wind2Loads toolbox, with two hidden layers containing 10 to 20 neurons and tanh activation function. The tanh function has been chosen because it allows a smooth gradient. The model complexity, learning rate and the number of epochs have been tuned separately for each output channel, following the guidelines from Ref. [22]. The prediction error of the surrogates is approximately zero-mean and normally distributed, with a variance compatible with the seed to seed dependence.

The optimization problem that will be solved is:

$$\max_{\mathbf{x}_i} \text{AEP}, \quad (5a)$$

$$\text{subject to: } \text{dist}(\mathbf{x}_i, \mathbf{x}_j) \geq 3D, \quad (5b)$$

$$\mathbf{x}_i \in \text{site}, \quad (5c)$$

$$\bar{L}_k(\mathbf{x}_i) \leq 1. \quad (5d)$$

The cost function for the optimization problem is the wind farm annual energy production. Eq. (5b) is the spacing constraint, which imposes a minimum distance of 3 diameters among all pairs of turbines. Eq. (5c) is the boundary constraint, which requires that all turbines lie inside the site. Eq. (5d) is the load constraint, which requires that the normalized LDEL \bar{L} for all turbines i and all load channels k is within the limit. As discussed in Ref. [2], the blade root flapwise, blade root edgewise, tower bottom side-side and tower bottom fore-aft bending moments, and shaft torsional moment, are all subject to large increases due to wakes, therefore they have been constrained.

The Extended Design Structure Matrix (XDSM) diagram [13] for the present optimization problem is shown in Fig. 2.

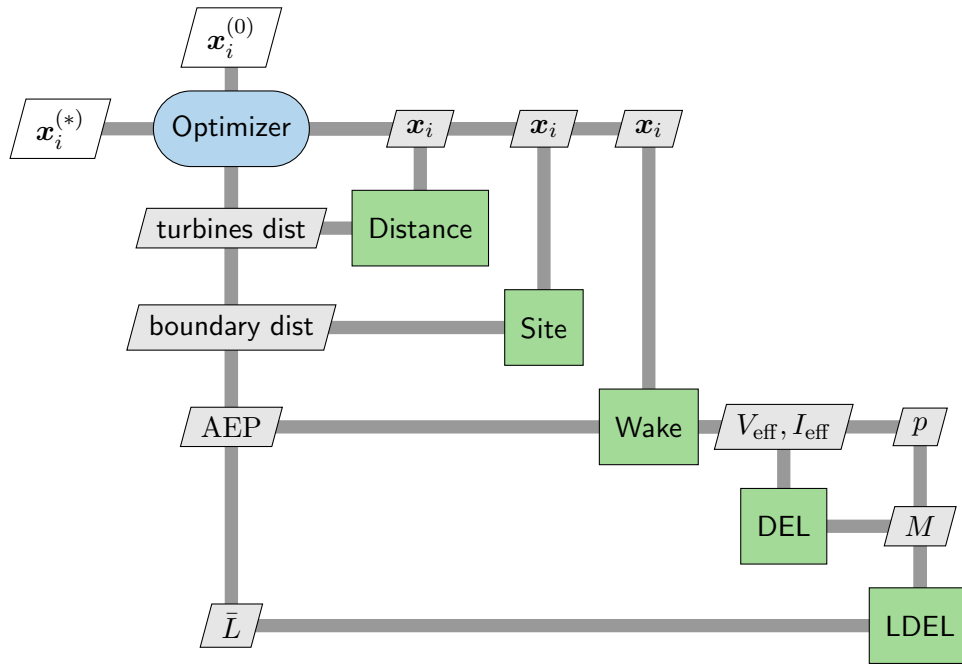


Figure 2. XDSM diagram for problem (5).

We will now provide two academic examples for which the optimization problem has a known global optimum.

3.1. Example 1

For the first example, we consider a 1-dimensional site where the wind speed increases linearly from left to right at an arbitrary rate a , and both wind directions are equally likely. The site has a length of $18D$, i.e. 3.6 km. The reference wind speed at $x = 0$ is 12 m/s and the turbulence

intensity follows the Normal Turbulence Model (NTM)

$$I(x) = \frac{I_{\text{ref}}}{V_{\text{ref}}(x)} (0.75 V_{\text{ref}}(x) + 5.6), \quad (6)$$

with $I_{\text{ref}} = 15\%$. The wind shear has a power law, with exponent 0.2. In one dimension, the turbines always operate in full wake, which causes unrealistically high LDELs. For this reason, the load constraint has not been considered in this example, which focus instead on the maximization of the AEP under wake effects.

To solve this optimization problem we have selected a gradient-free algorithm: the **Constrained Optimization BY Linear Approximation (COBYLA)**.

For the initial guess, we position the first turbine on the left boundary and three more at a distance of $3D$ each. by choosing $a = 0$ m/s/m, the optimal layout is symmetric around $x = 0$, with the external turbines located on the site boundaries. The outer turbines generate the same AEP, and the inner ones an equal, albeit lesser, amount. For $a = 0.001$ m/s/m, the optimizer moves the last turbine to the right boundary, and the others at a distance of $3D$ each, which means that the solution is dominated by the wind speed increase. For intermediate values of a , the first turbine moves progressively towards the right, the last is always on the right boundary, and the inner ones are located asymmetrically. Fig. 3 shows the turbines locations and AEP, as a function of the iteration number in the optimization, for $a = 0.0004$ m/s/m. In this case, PyWake estimates the initial AEP to be 345.12 GWh, and an increase for the optimal layout of 5.9%.

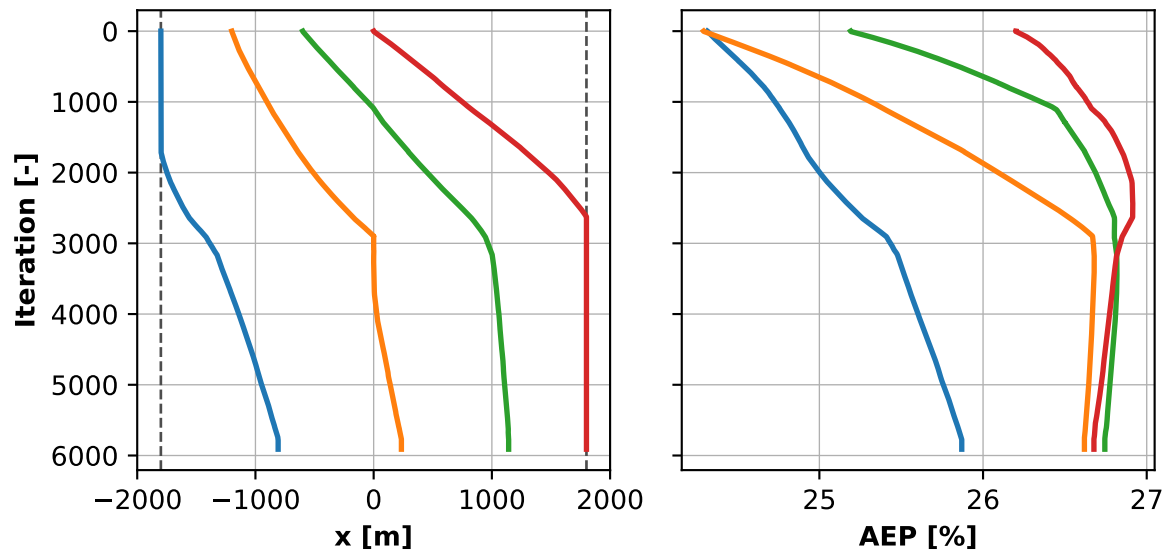


Figure 3. Turbines location and AEP, as a function of the iteration number, for the first example with $a = 0.0004$ m/s/m. The AEP of the individual turbines are normalized with respect to the total AEP for the initial layout.

3.2. Example 2

In this second example the site is rectangular, centered at the origin, with sides of length $18D$ and $30D$. The reference wind speed varies from left to right at a rate of 0.0002 m/s/m. The reference turbulence intensity increases linearly from bottom to top from 8% to 12% , and its

actual value is given by the NTM. The wind speed probability follows a Weibull distribution and the wind rose is uniform.

As observed in Ref. [15], the cost function in a wind farm optimization problem can have several local minima. Therefore, we have chosen a gradient-based algorithm, namely the Sequential Least Squares Programming (SLSQP) optimization algorithm, and looked for the global optimum with a multi-start approach.

For this example we have considered 12 turbines. The initial guesses are selected using a Latin hypercube sampling with a uniform distribution, and ensuring a minimum distance of $3D$. To set the reference value for nondimensionalizing the LDELs, i.e. the constraint limits, we have placed one turbine in the origin of the site, evaluated the LDELs, and applied a safety factor of 20%.

The optimal layout is shown in Fig. 4, along with the AEP of each turbine, normalized with respect to the total value. It can be seen that most turbines are located on the right boundary, where the wind speed is higher. The distance of the turbines on the top and bottom boundaries is dictated by the wake effects. Among the local minima, there is an AEP variation of 0.7%.

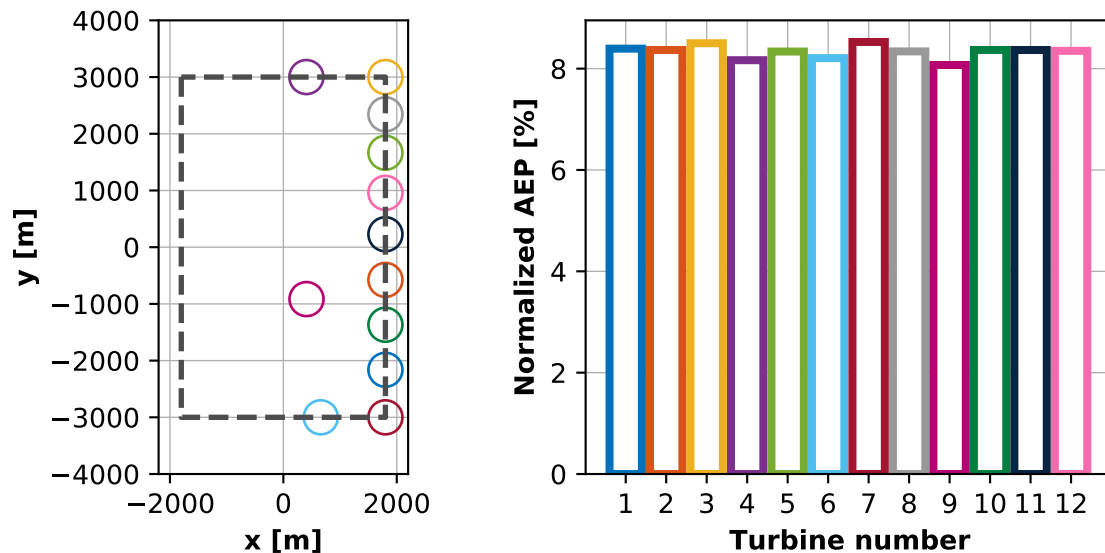


Figure 4. Optimal layout for the second example on the left, and normalized AEP of each turbine on the right. The circles indicate the spacing constraint.

The LDELs for the optimal layout are shown in Fig. 5. As we can see, in this layout the load constraint is not active. The load differences among the turbines are consistent with the wind speed and turbulence intensity variations over the site.

While setting up this example, we have experimented with several layout and LDEL limits. We have observed that when a load constraint is active, the optimizer tends to move the turbines closer to each other, in an attempt to reduce the effective wind speed, as it is the most important factor for these surrogates. This behavior is also due to the Frandsen approach, which does not model partial wakes, and hence cannot take into account the huge load increase that this choice would cause [1]. It has been observed that the SLSQP can successfully optimize a layout with an active load constraint, but often cannot recover from cases where the constraint violation is large, especially if this happens for the initial guess.

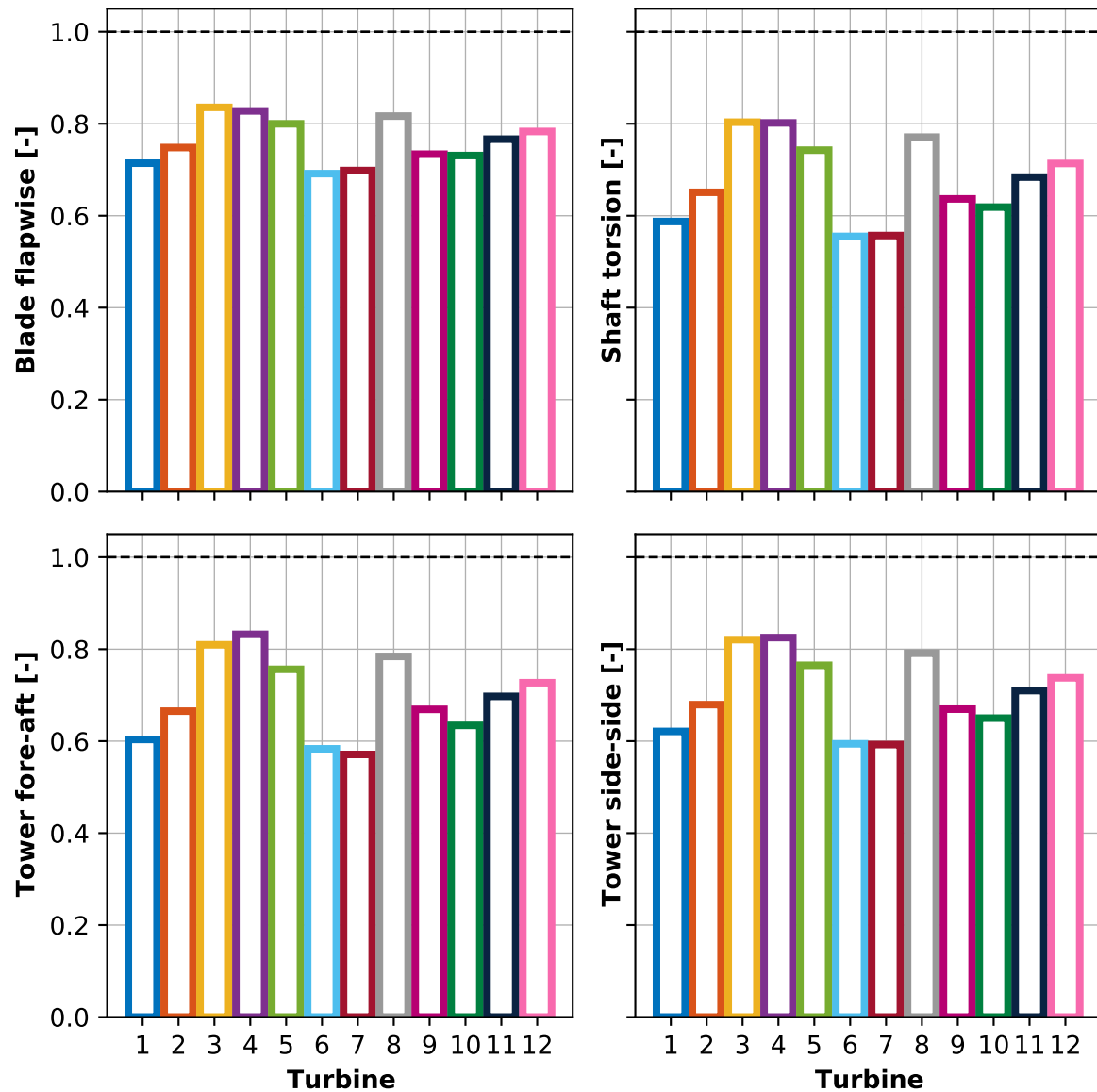


Figure 5. Normalized LDEL for each turbine in the optimal layout of the second example.

4. Conclusions

In this work, we have presented the latest version of TOPFARM, with a particular focus on the implementation of the load constraint. To provide a robust and efficient environment, this new version has been built on top of OpenMDAO. The multidisciplinary approach that characterized the first version has been maintained, by separating the platform into several interconnected components. The load constraint has been added employing various types of surrogate models, this way achieving a fast and accurate load prediction, and enabling gradient-based optimizations. Furthermore, the use of surrogate models makes the constraint layout-independent.

For the case studies, we have trained artificial neural networks on the Damage Equivalent Loads estimated through nonlinear aeroelastic simulations. The results show that the

optimization problems converge to the expected solutions.

This work will be extended in four directions:

- Although the Frandsen equivalent turbulence approach provides good results, it does not include important wake characteristics. We will thus implement the Dynamic Wake Meandering model.
- In several cases, it is useful to adopt more than one type of turbine in the same wind farm. We will thus extend the load constraint to support multiple types.
- A surrogate model for the power would allow taking into account more input variables than just the wind speed. Especially in complex terrains, this would provide better estimates of the AEP.
- This paper has focused on academic examples, but the importance of adding a load constraint should be assessed on realistic wind farms.

All of these points will be the object of a forthcoming publication.

Acknowledgments

This work has been partly supported by the Multi-Objective Wind Farm Layout Optimization Tool (MOWFLOT) project, built on TOPFARM, a collaboration between DTU Wind Energy and Goldwind Denmark.

References

- [1] Larsen T, Larsen G, Aagaard Madsen H and Petersen S 2015 *Scientific Proceedings. EWEA Annual Conference and Exhibition 2015* (European Wind Energy Association (EWEA)) pp 95–99 ISBN 9782930670003
- [2] Dimitrov N 2019 *Wind Energy* **22** 1371–1389 ISSN 1095-4244
- [3] Réthoré P E, Fuglsang P, Larsen G C, Buhl T, Larsen T J and Madsen H A 2014 *Wind Energy* **17** 1797–1816 ISSN 10991824
- [4] Stanley A P J and Ning A 2019 *Wind Energy Science* **4** 99–114
- [5] Rademakers L W M M, Braam H, Obdam T S and Pieterman R 2009 *European Offshore Wind 2009 Conference* vol 1 pp 14–16
- [6] Pedersen M M, van der Laan P and Friis-Møller M 2019 PyWake <https://doi.org/10.5281/zenodo.2562661>
- [7] NREL 2019 FLORIS. Version 1.0.0 URL <https://github.com/NREL/floris>
- [8] Jensen N O 1983 A note on wind generator interaction Tech. rep. Risø National Laboratory
- [9] Bastankhah M and Porté-Agel F 2014 *Renewable Energy* **70** 116–123
- [10] Ott S, Berg J and Nielsen M 2011 Linearised CFD models for wakes Tech. Rep. Risoe-R, No. 1772(EN)
- [11] Bak C, Zahle F, Bitsche R, Kim T, Yde A, Henriksen L C, Hansen M H, Blasques J P A A, Gaunaa M and Natarajan A 2013 Description of the DTU 10 MW reference wind turbine Tech. Rep. DTU Wind Energy Report-I-0092
- [12] Krabben I G W, van der Laan M P, Koivisto M, Larsen T J, Pedersen M M and Hansen K S 2019 *Journal of Physics: Conference Series* **1256** 012028
- [13] Gray J S, Hwang J T, Martins J R R A, Moore K T and Naylor B A 2019 *Structural and Multidisciplinary Optimization* **59** 1075–1104
- [14] Riva R, Liew J, Friis-Møller M, Dimitrov N K, Barlas E, Réthoré P E and Pedersen M M 2019 Topfarm2 <https://doi.org/10.5281/zenodo.3553712>

- [15] Stanley A P J and Ning A 2019 *Wind Energy Science* **4** 663–676
- [16] Fingersh L, Hand M and Laxson A 2006 Wind turbine design cost and scaling model Tech. Rep. NREL/TP-500-40566 National Renewable Energy Laboratory
- [17] Frandsen S T 2007 Turbulence and turbulence-generated structural loading in wind turbine clusters Tech. Rep. Risø-R-1188(EN) Risø National Laboratory
- [18] Dimitrov N, Kelly M, Vignaroli A and Berg J 2018 *Wind Energy Science* **3** 767–790
- [19] Baudin M, Dutfoy A, Iooss B and Popelin A L 2015 OpenTURNS: An industrial software for uncertainty quantification in simulation Tech. rep.
- [20] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems Tech. rep.
- [21] Larsen T J and Hansen A M 2018 How 2 HAWC2, the user’s manual
- [22] Schröder L, Dimitrov N K, Verelst D R and Sørensen J A 2018 *Journal of Physics: Conference Series* **1037** 062027