Renewable Energy 216 (2023) 119061

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

# Offshore wind farm layout optimization using ensemble methods

Kjersti Solberg Eikrem [a,*], Rolf Johan Lorentzen [a], Ricardo Faria [b], Andreas Størksen Stordal [a], Alexandre Godard [c]

[a] NORCE Norwegian Research Centre AS, Postboks 22 Nygårdstangen, 5838 Bergen, Norway
[b] ARDITI, Oceanic Observatory of Madeira, Agência Regional para o Desenvolvimento da Investigação Tecnologia e Inovação, Edif. Madeira Tecnopolo, Piso 2, Caminho da Penteada, 9020-105 Funchal, Portugal
[c] Wunderocean, Rua Dr Calado, 26. Figueira da Foz, Portugal

ARTICLE INFO

*Keywords:*
Wind farm layout optimization
Ensemble optimization (EnOpt and EPF-EnOpt)
Constrained optimization
Levelized cost of energy (LCOE)
Floating offshore wind

ABSTRACT

When planning wind farms it is important to optimize the layout to increase production and reduce costs. In this paper we minimize the levelized cost of energy (LCOE) for a floating wind farm using wind data in an area around Porto Santo in Portugal. We use ensemble based optimization (EnOpt), which is frequently applied in the geophysical community to find optimal controls of oil reservoirs. EnOpt is usually used for unconstrained optimization problems or for problems with simple constraints, for example upper and lower bounds on the optimization variables. Here we consider a layout problem with many constraints on the distances between turbines. To handle the constraints, we use an extension of EnOpt called EPF-EnOpt, in which the constrained problem is replaced by a series of unconstrained problems with increasing penalty terms. We compare the performance of this method with EnOpt with a fixed penalty term, and with a deterministic gradient method. All the tested methods reduce the LCOE, but EPF-EnOpt gives better results than both a single run of EnOpt with a fixed penalty term and the deterministic gradient method, and at a lower computational cost than using the gradient method. We also consider the problem of maximizing the annual energy production without taking into account any costs. EPF-EnOpt performs the best also for this problem.

## 1. Introduction

With the worlds increasing energy demand and the problems caused by $CO_2$ emissions and climate change, there is a great need for sustainable energy production. Wind energy could constitute an important contribution for solving these problems, and the production is expected to increase in coming years, in particular for offshore wind. To make the most of the wind farms, it is beneficial to consider both the lifetime energy production and the lifetime cost of the wind farm when planning the layout. Often the levelized cost of energy (LCOE) is used to evaluate a layout, and optimization can be performed to make it as low as possible.

Wind farm layout optimization (WFLO) is a complex and computationally demanding problem, and several optimization methods have been studied over the last decades. Genetic algorithms have been used by many authors [1–6]. Other methods include particle swarm optimization (PSO) [7–10], greedy algorithms [4,11], mathematical programming [12] and hybrid methods [13]. We refer to several review papers for an overview of the existing literature [14–20].

In this paper we will use a variant of ensemble-based optimization (EnOpt), a method that has been used extensively in the geophysical community for optimizing control of reservoirs and planning wells, but only recently has been applied to wind farm optimization problems. EnOpt is a stochastic optimization method where an ensemble of controls is used to calculate an approximate gradient. It was first introduced in Lorentzen et al. [21] to optimize water flooding in an oil reservoir. It has been further developed in Chen et al. [22],Chen and Oliver [23],Fonseca et al. [24],Stordal et al. [25],Fonseca et al. [26]. A major benefit of EnOpt is that the number of forward simulations is independent of the number of parameters, and typically 100 (or less) simulations are used to compute an approximate gradient. Our application of EnOpt is motivated by this fact, and although a relatively simple and fast wake model is used in this study, we aim at developing a workflow that is applicable to a wide range of model fidelities and complexities. EnOpt is a fast alternative to global methods and deterministic gradient methods, in particular if there are many variables to optimize or the model computations are costly. EnOpt often performs better than deterministic gradient methods on problems with many local minima, as these methods can easily get stuck in the closest local minimum, while EnOpt is able to jump over some local minima since

* Corresponding author.
*E-mail address:* kjei@norceresearch.no (K.S. Eikrem).

https://doi.org/10.1016/j.renene.2023.119061
Received 12 April 2023; Received in revised form 16 July 2023; Accepted 17 July 2023
Available online 21 July 2023
0960-1481/

it approximates the gradient using samples in a larger region. Another benefit of EnOpt is the potential of investigating robust optimization, e.g., to include uncertainty in the wind profiles for a specific site by representing the wind by an ensemble of realizations. The procedure for robust optimization is well documented in the references above, and the extension does not require additional computation time, contrary to many other optimization algorithms. To the best of our knowledge, the only applications of EnOpt to wind farm layout optimization are [27, 28]. The current work differs from these in the way the constraints are handled and by having a larger number of parameters to optimize as the turbines can move freely.

EnOpt is usually used for unconstrained optimization problems or for problems with simple constraints, for example upper and lower bounds on the optimization variables. In the wind farm optimization problem considered here, there are many constraints that are not simple bounds. The turbines are allowed to take any position in a designated area, but with a lower limit on the distances between the turbines, and for $N$ turbines, this gives $N*(N-1)/2$ constraints. The standard EnOpt method will not be able to fulfill the constraints when there are many turbines. Therefore, we will use a method called EPF-EnOpt, which is an extension of EnOpt designed to solve constrained optimization problems. It combines EnOpt with the exterior penalty function method (EPF) [29]. EPF-EnOpt was introduced in Oguntola and Lorentzen [30] to optimize the production in an oil reservoir. In EPF-EnOpt, the constrained problem is replaced by a series of unconstrained problems with increasing penalty terms. The penalty term is zero in the regions where the constraints are not violated, and nonzero if they are violated.

We apply the method to a test case north of the island Porto Santo in Portugal using realistic data based on reanalysis of historical wind speed measurements. The methodology used to generate the wind data consists of a long-term data analysis for the atmosphere and for the ocean, to constitute a Typical Meteorological Year (TMY). A TMY is composed by the 12 more representative months from different years over a certain (long-term) period of observational data. To obtain a TMY that characterizes the distribution and variability associated with the islands, hourly wind data was necessary.

For the numerical experiments a floating wind farm is considered, and costs related to floating offshore wind is used in the calculation of the LCOE. The performance of EPF-EnOpt is compared with a deterministic gradient method from Scipy [31] and with EnOpt with a fixed penalty term. We also consider the problem of maximizing the annual energy production (AEP) without considering any costs.

The paper is organized as follows. In Section 2 we present the EnOpt algorithms. In Section 3 the wind data and the optimization problem is presented. The results follow in Section 4 and conclusion in Section 5. More details of the wind data simulations are presented in the appendix.

## 2. Ensemble-based optimization

### 2.1. Constrained optimization problem

A common problem in science and engineering is to minimize or maximize a function while satisfying some constraints. A general minimization problem can be described by an objective function $J$ which depends on a vector of control variables $\mathbf{u} \in \mathbb{R}^{N_u}$. The goal is to find $\mathbf{u}$ that minimizes $J$ while respecting the constraints, i.e.

$$\min_{\mathbf{u}\in\mathbb{R}^{N_u}} \; J(\mathbf{u}) \tag{1}$$

$$\text{subject to:} \quad g_i(\mathbf{u}) \geq 0, \quad \forall i \in \mathrm{I} \tag{2}$$

$$h_j(\mathbf{u}) = 0, \quad \forall j \in \mathrm{E}, \tag{3}$$

where $g_j$ and $h_j$ are constraint functions from $\mathbb{R}^{N_u}$ to $\mathbb{R}$ and I and E are indexing sets.

In our application, we will minimize the LCOE for a wind farm by finding the optimal position of the turbines. Our focus is on EnOpt algorithms, but for comparison we also test another constrained optimization method from the SciPy python package [31], scipy.optimize.minimize with method 'trust-constr'. In the next subsection we describe unconstrained optimization with EnOpt, and then the constrained optimization with EPF-EnOpt.

### 2.2. Ensemble optimization (EnOpt)

The original formulation of EnOpt was derived in Chen et al. [22] as an approximation to the preconditioned steepest ascent method [32], and it was used to maximize the net present value for the production of an oil reservoir. In the following we describe it as a minimization method as our main goal is to minimize the LCOE. The preconditioned steepest descent is described by

$$\mathbf{u}_{l+1} = \mathbf{u}_l - \beta_l\, \mathbf{C}\, \nabla J(\mathbf{u}_l), \quad \forall l = 1, 2, \ldots, \tag{4}$$

where $\nabla J$ is the gradient of the objective function $J$, $\mathbf{C}$ is a preconditioning or smoothing matrix, $l$ is the iteration number and $\beta$ is the step length. In EnOpt, the control variable $\mathbf{u}$ is considered a random variable with covariance matrix $\mathbf{C}_u$. This covariance is used as the preconditioner in (4). The covariance matrix can also change with iterations [24,25], and we denote it $\mathbf{C}_u^l$. In each iteration of EnOpt, an ensemble of $N$ controls is drawn from a Gaussian distribution with mean $\mathbf{u}_l$ and standard deviation $\mathbf{C}_u^l$

$$\mathbf{u}_{l,j} \sim \mathcal{N}(\mathbf{u}_l, \mathbf{C}_\mathbf{u}^l), \; j = 1, 2, \ldots, N.$$

A common way to calculate $\mathbf{C}_u^l \nabla J(\mathbf{u}_l)$ in (4) is to approximate it by the cross covariance [26]

$$\mathbf{C}^l_{\mathbf{u},J(\mathbf{u})} = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{u}_{l,j} - \mathbf{u}_l)\Big(J(\mathbf{u}_{l,j}) - J(\mathbf{u}_l)\Big). \tag{5}$$

Then the update is calculated as

$$\mathbf{u}_{l+1} = \mathbf{u}_l - \beta_l \mathbf{C}^l_{\mathbf{u},J(\mathbf{u})} \tag{6}$$

We will instead estimate $\nabla J(\mathbf{u}_l)$ using linear regression on the points $\{\mathbf{u}_{l,j} - \mathbf{u}_l\}$ and $\{J(\mathbf{u}_{l,j}) - J(\mathbf{u}_l)\}$. In many scenarios, there are fewer ensemble members than variables to optimize, hence the regression is not unique, but the Moore–Penrose pseudoinverse can be used to get the minimum norm solution. Define the $N \times N_u$ matrix $\hat{U} = [\mathbf{u}_{l,1} - \mathbf{u}_l \;\; \mathbf{u}_{l,2} - \mathbf{u}_l \;\; \ldots \;\; \mathbf{u}_{l,N} - \mathbf{u}_l]^T$ and $\hat{J} = [J(\mathbf{u}_{l,1}) - J(\mathbf{u}_l) \;\; J(\mathbf{u}_{l,2}) - J(\mathbf{u}_l) \;\; \ldots \;\; J(\mathbf{u}_{l,N}) - J(\mathbf{u}_l)]^T$ and estimate

$$[\nabla J(\mathbf{u}_l); b] \approx [\hat{U}\; \mathbf{1}]^+ \hat{J}, \tag{7}$$

where + denotes the pseudoinverse, b is the offset in the linear regression and $\mathbf{1}$ is a vector of ones with length $N$ that is appended to $\hat{U}$. ($\hat{U}$ is scaled before this is done to be of the same order of magnitude as 1 for numerical reasons.) The estimate of $\nabla J(\mathbf{u}_l)$ from (7) is then used, without preconditioning, to update

$$\mathbf{u}_{l+1} = \mathbf{u}_l - \beta_l \nabla J(\mathbf{u}_l). \tag{8}$$

The step length $\beta_l$ is calculated by a backtracking line search [29]. We got better results by using (8) instead of (6).

In Chen et al. [22] the covariance $\mathbf{C}_u$ was kept the same in all iterations, but it has been shown that better results can be obtained by updating it [24,25]. We update it using the following formula

$$\mathbf{C}_u^{l+1} = \mathbf{C}_u^l - \gamma_l \frac{1}{N}\sum_{j=1}^{N}(J(\mathbf{u}_{l,j}) - J(\mathbf{u}_l))\left((\mathbf{u}_{l,j} - \mathbf{u}_l)(\mathbf{u}_{l,j} - \mathbf{u}_l)^T - \mathbf{C}_u^l\right), \tag{9}$$

where $\gamma$ is the step size. For more information on how to derive the covariance adaption, see Stordal et al. [25]. To make sure this matrix is positive definite we force all negative eigenvalues to a small positive number. If the step length $\beta_l$ is reduced because of backtracking, we reduce $\gamma_l$ with the same factor.

If a number of backtracking iterations have been tested without finding an accepted step, we perform a resampling of the ensemble

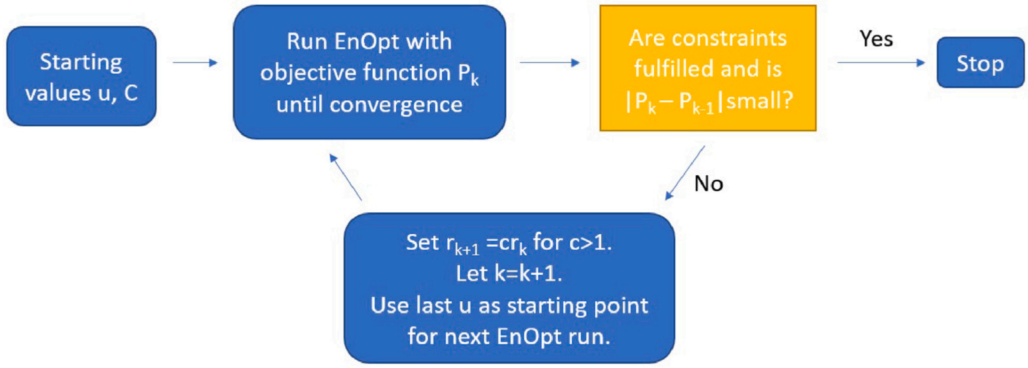



**Fig. 1.** Flow diagram for EPF-EnOpt. The starting value for $u$ can be selected manually or be the result of an initial run of unconstrained EnOpt.

with the original covariance matrix and calculate a new gradient. If it is again unsuccessful, we resample in a smaller area by dividing all elements of the original covariance matrix by four, (i.e., the standard deviation is halved) and calculate the gradient again. The algorithm continues until a maximum number of consecutive resamplings have been performed.

### 2.3. Constrained ensemble optimization (EPF-EnOpt)

To handle more complex constraints in EnOpt, EPF-EnOpt was introduced in Oguntola and Lorentzen [30]. This extension of EnOpt is based on the exterior penalty function (EPF) method [29], in which a constrained optimization problem is replaced by a series of unconstrained problems with increasing penalty terms. When the penalties are the squares of the constraint violations it is also denoted quadratic penalty method. In each subproblem, the objective function is modified by adding extra terms for equality and inequality constraints multiplied with a weight $r_k$

$$P_k(\mathbf{u}, r_k) = J(\mathbf{u}) + r_k\Big(\sum_{i\in\mathrm{I}}(\min\{g_i(\mathbf{u}),0\})^2 + \sum_{j\in\mathrm{E}}|h_j(\mathbf{u})|^2\Big). \tag{10}$$

If the constraints are fulfilled, i.e. $g_i(\mathbf{u}) \geq 0$ and $h_j(\mathbf{u}) = 0$ for all $j$ in the index sets, the penalty terms are zero. The sequence $\{r_k\}_{k=1}^{\infty}$ is increasing with $\lim_{r\to\infty} r_k = \infty$. A standard construction is to choose positive constants $r_0 > 0$ and $c > 1$ and letting $r_{k+1} = cr_k$. The updating is done as in (8), but $\nabla J(\mathbf{u}_l)$ is calculated by a linear regression on $\{\mathbf{u}_{l,j} - \mathbf{u}_l\}$ and $\{P_k(\mathbf{u}_{l,j}, r_k) - P_k(\mathbf{u}_l, r_k)\}$. The algorithm stops when the improvement in a run of EnOpt is less than a predefined number $\epsilon_1$ and the constraint violation is small enough, i.e.

$$|P_k(\mathbf{u}_0, r_k) - P_k(\mathbf{u}, r_k)| < \epsilon_1 \tag{11}$$

and

$$\sum_{i\in\mathrm{I}}(\min\{g_i(\mathbf{u}),0\})^2 + \sum_{j\in\mathrm{E}}|h_j(\mathbf{u})|^2 < \epsilon_2. \tag{12}$$

A small modification compared to Oguntola and Lorentzen [30] is implemented here by running unconstrained EnOpt as the first iteration in EPF-EnOpt, and then the remaining iterations are performed with increasing penalty terms as described above. This modification resulted in lower LCOE at the cost of more iterations. Another difference is that we decreased the initial covariance matrix $C_{\mathbf{u}}^0$ with a predefined factor each time a new EnOpt run with higher penalty was initiated. This variance reduction was implemented to avoid many samples in the highly penalized regions and resulted in fewer resampling steps. Fig. 1 shows a flow diagram for EPF-EnOpt.

For comparison, we also consider a simpler version of EPF-EnOpt, where we fix a weight $r$ and run a single optimization with this penalty. This requires careful selection of the penalty, since choosing it too large will result in nonoptimal positions as too much weight is put on the constraint. But choosing it too small will result in positions that do not fulfill the constraints.

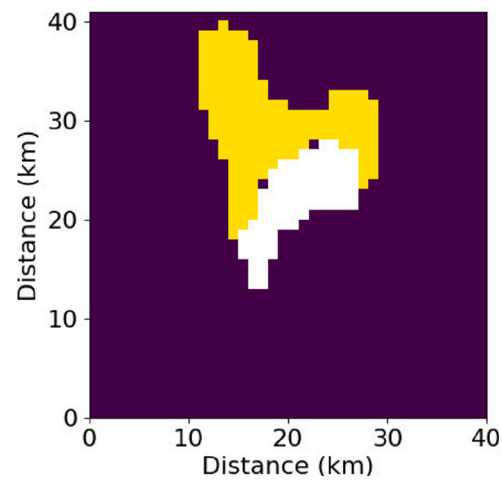



**Fig. 2.** The yellow area is the area considered for the wind farm, and the white is the island Porto Santo. This area is from AREAM Agência Regional da Energia e Ambiente da Região Autónoma da Madeira [33]. It is resampled on the same grid as the wind data.

## 3. Wind farm layout problem

### 3.1. Wind resources

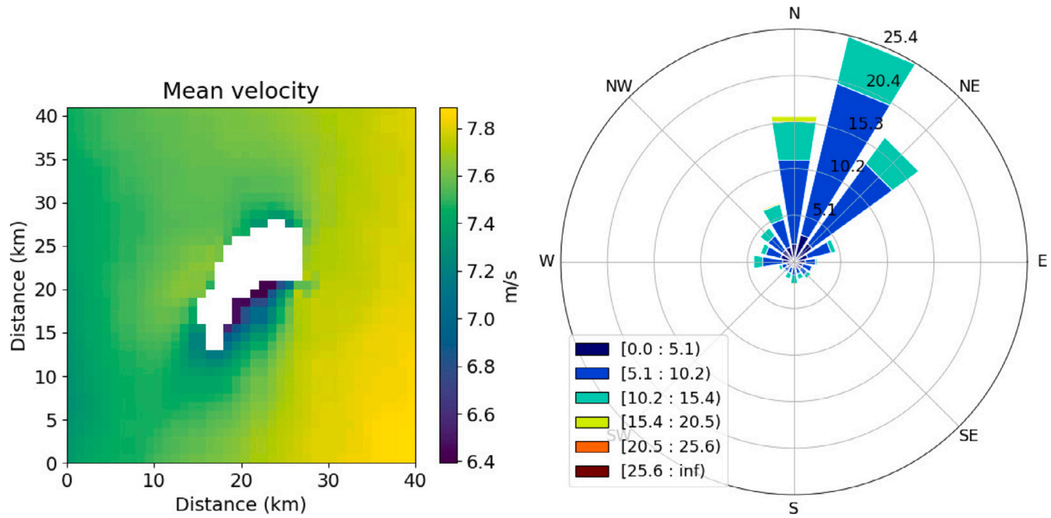
For the numerical experiments we use wind data in an area around Porto Santo in Portugal. The wind data are generated for a larger area around the Madeira Islands, and the generation of these data is described in the appendix. The area we consider for the wind farm is shown in Fig. 2. The average wind speed and the wind rose for the area are shown in Fig. 3. We calculate the speed $v_2$ at hub height $h_2$ from the formula

$$v_2 = v_1 * \frac{ln(h_2/z_0)}{ln(h_1/z_0)}, \tag{13}$$

where $z_0$ is the surface roughness set to 0.002, $h_1$ is 80 m and $v_1$ is outputted from the wind simulations described in the appendix. The wind data is hourly data for one year. The wind data are given by hourly wind speeds and directions for one year.

### 3.2. Problem description

The goal is to minimize the LCOE for a fixed number of turbines in the area shown in Fig. 2. We use 15 MW turbines from NREL Wind Turbine Power Curve Archive [34] with a hub height of 150 m and a rotor diameter of 240 m. In order to simulate the wake effects and calculate the produced energy we use PyWake [35] with the Bastankhah Gaussian wake model [36]. The computational speed is




(a) Mean wind at 80 m above sealevel around Porto Santo.

(b) Mean wind rose showing the distribution of the wind speeds and directions for the selected area in Figure 2.

**Fig. 3.** Wind resources. The area with wind data is approximately latitude 33.21487 to 32.89658 and longitude −16.17131 to −16.54167.

fast, and it takes less than a second to calculate the annual energy production for 50 turbines on a desktop computer. The wind data are used to calculate a probability of each wind direction and speed, which was used as input for the wake model. Bins of 1 m/s and 1 degree were used for the wind speed and angle of direction, respectively. We calculated an average probability for the whole area in Fig. 2, but scaled the wind speeds for each grid block with the average speed in that grid block divided by the average for the whole area. (We could not find any documentation on whether it is possible to have different probabilities for each gridblock in this software.)

The constraints are that turbines are required to be inside the selected area, and that all turbines are at least 5 rotor diameters apart (1200 m). The constraints on the distance between turbines can be described by functions which are negative when the distance is too small and 0 when it is fulfilled. We use the distance in kilometer minus the minimum distance as constraint functions, and if this is positive it is set to 0. In our case we do not need the equality constraints (3). For $N$ turbines, the number of constraints on distances between pairs of turbines is $N * (N-1)/2$.

The levelized cost of energy can be calculated as

$$LCOE = \frac{\sum_{t=0}^{T} \frac{I_t + M_t}{(1+r)^t}}{\sum_{t=0}^{T} \frac{E_t}{(1+r)^t}},$$

where $I_t$ is the investment cost in year $t$, $M_t$ is the maintenance cost, $E_t$ is the produced energy, $r$ is the discount rate and $T$ is the number of years considered. Table 1 presents the costs applied in this study. The costs of turbines, balance of system, soft costs and maintenance are from Beiter et al. [37]. The cost of balance of system includes e.g. substructure and foundation, assembly and installation, development and engineering management, and the soft costs include insurance during construction, decommissioning, construction financing and contingency.

In order to find the shortest cable layout connecting all the turbines, we use a minimum spanning tree algorithm [39]. We also add two times the depth of the water column to the length of the cable for each turbine position assuming that the cable will be on the sea bottom. (This is a small simplification since some turbines are connected to more than two other turbines, and some only to one.) The bathymetry data is shown in Fig. 4. We assume that a substation is positioned close to the turbine that is closest to the shore, and the length of the cable from the substation to the shore is included in the cable cost.

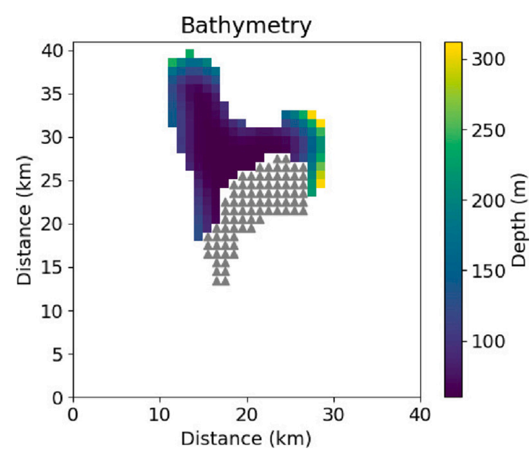
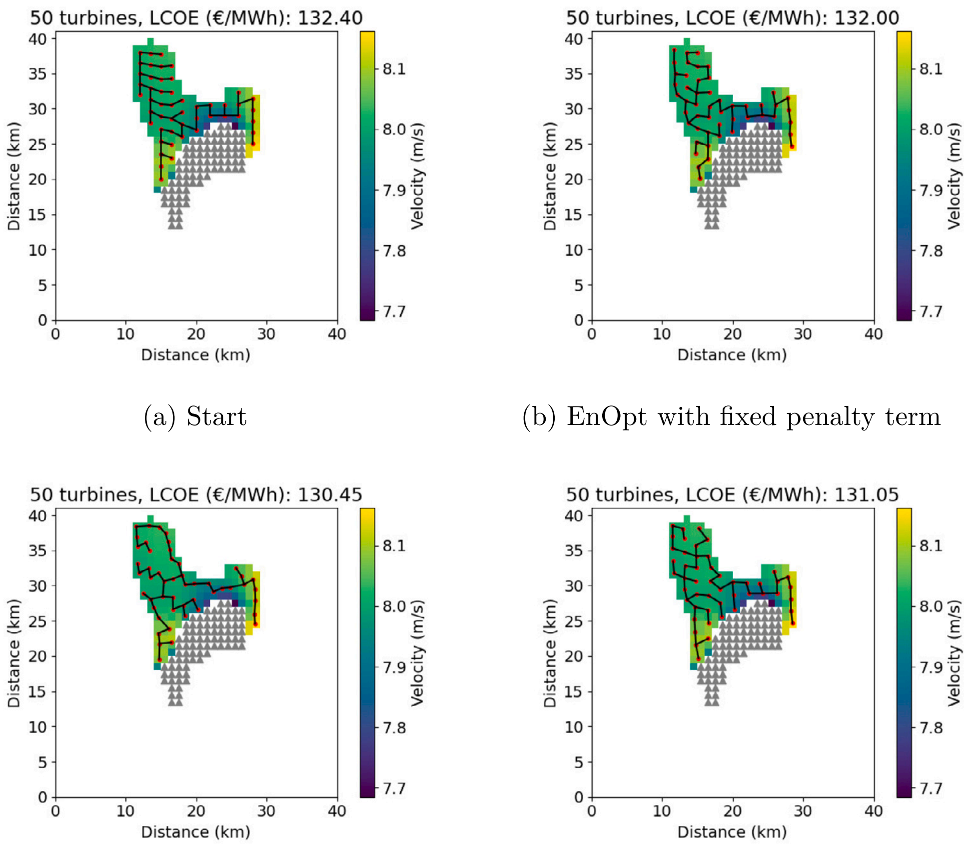



**Fig. 4.** Water depth in the selected area.

**Table 1**
Costs used in LCOE optimization. The four first costs are from [37]. We have subtracted the costs of electrical infrastructure in the second cost (balance of plant) since we are considering these costs separately. The costs of the cables are based on [38] (but we have not included all details). We use an exchange rate of 1 €/$ and 1.15 €/£, and a discount rate of 6%.

| Costs | Value |
|---|---|
| Turbines (per kW) | $1301 |
| Balance of plant (per kW) | $2258 |
| Soft costs (per kW) | $790 |
| Maintenance (per kW) | $130 |
| Internal cables (per km) | £1e6 |
| Cable to shore (per km) | £1.1e6 |
| Other electrical infrastructure | £100e6 |




(a) Start

(b) EnOpt with fixed penalty term

(c) EPF-EnOpt

(d) Scipy 'trust-constr'

**Fig. 5.** Manually selected start positions and final layout after optimization with three different methods. The gray triangles show the island and the colored background in the wind farm area shows the mean wind speed.

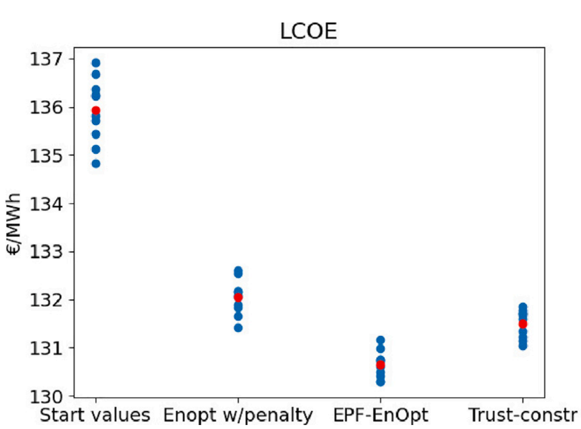



**Fig. 6.** Start values and final objective functions (LCOE) for 10 runs with random starting positions for different methods. The red dots show the means of the 10 runs.

**Table 2**
EnOpt and EPF-EnOpt parameters used in the LCOE optimization. The parameters above the dashed line are for EnOpt, and below are for EPF-EnOpt, except the last which is for EnOpt with fixed penalty.

| Parameter | Value |
|---|---|
| Number of ensemble members ($N$) | 25 |
| Initial variance (values on diagonal of $C_u^1$) | $1000^2$ |
| Initial step size $\beta_l$ | $2000/\|\|\nabla J(\mathbf{u}_l)\|\|_\infty$ |
| Initial step size for covariance adaptation $\gamma$ | 0.01 |
| Maximum number of backtracking iterations | 6 |
| Maximum number of resamplings | 4 |
| Difference between EnOpt runs in EPF-EnOpt ($\epsilon_1$ in Eq. (11)) | 0.1 |
| Maximum constraint violation ($\epsilon_2$ in Eq. (12)) | 0.001 |
| Variance reduction factor in each EnOpt run | 0.8 |
| Initial $r_0$ EPF-EnOpt | $J(\mathbf{u}_0)/1500$ |
| Increase factor for $r_k$ in EPF-EnOpt | 2 |
| Initial $r_0$ EnOpt with fixed penalty | $J(\mathbf{u}_0)/60$ |

## 4. Results

We consider 50 turbines in the allowed area, which means we have 100 variables to optimize, as we must determine the x- and y-positions. Both manually selected starting positions and random starting positions were tested. For few turbines, for example 5, the unconstrained EnOpt method often finds a solution that fulfills the distance requirement. But for 50 turbines, optimizing with unconstrained EnOpt gives solutions that do not fulfill the constraints, hence we need to consider other methods. We optimize the positions using EPF-EnOpt, the trust region method from Scipy and EnOpt with a fixed penalty term. For EnOpt and EPF-Enopt an ensemble of size 25 is used. For the weights in the objective function (10) we used $r_0$ equal to the starting objective value divided by 1500 and increased it with a factor 2 for each outer iteration. The number 1500 was selected based on a few experiments with different values, and seemed to be a good trade-off between accuracy and speed. High starting penalty (dividing by a low number) gives faster convergence, but less good result, and vice versa. The other parameters used in the ensemble algorithms are specified in Table 2.

The results with manually selected starting positions are shown in Fig. 5 and in Table 3. As can be seen from the figures, the LCOE

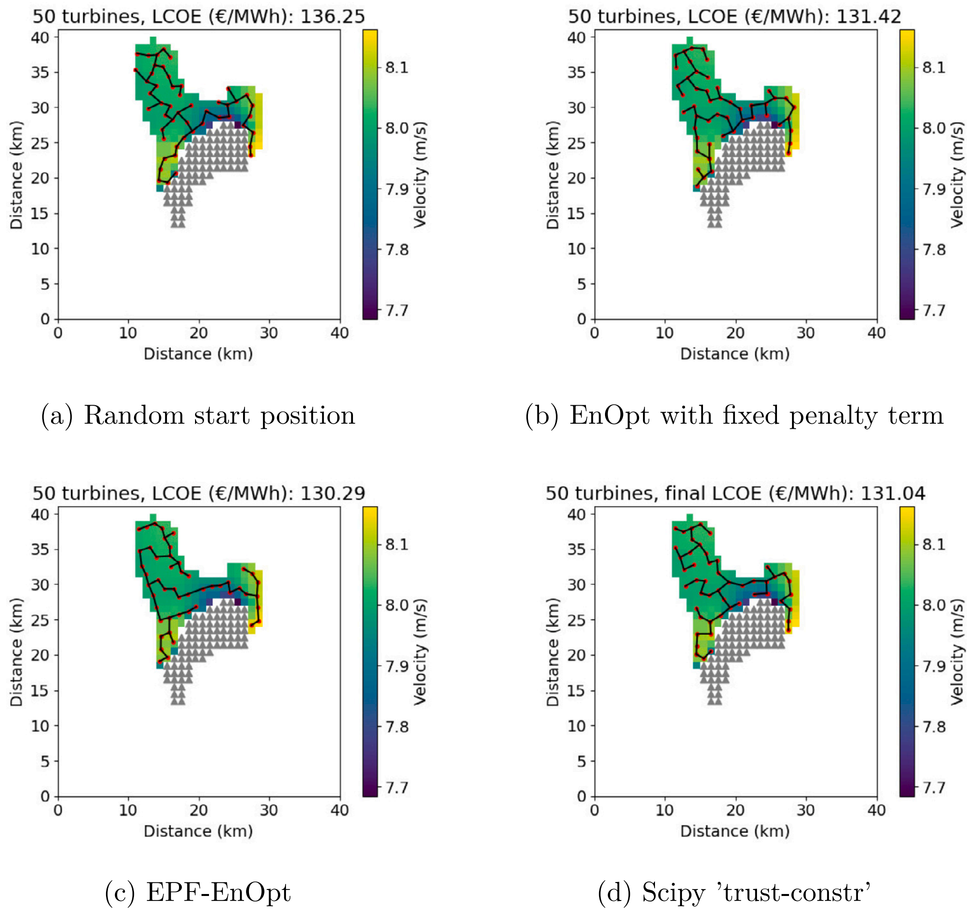



**Fig. 7.** Random start positions and final layout after optimization of LCOE. Figure (b), (c) and (d) show the best final layout of the 10 runs performed with each method and (a) shows the starting positions. (All methods obtained the best results from this starting layout.)

**Table 3**
The table shows initial value and results for minimization of the LCOE for manually selected start positions.

| | LCOE (€/MWh) | Function evaluations |
|---|---|---|
| Start | 132.40 | |
| EnOpt w/penalty | 132.00 | 1355 |
| EPF-EnOpt | 130.45 | 9113 |
| Scipy trust-constr | 131.05 | 19392 |

is reduced after optimization for all methods. The best results are obtained with EPF-EnOpt with an improvement of 1,5%.

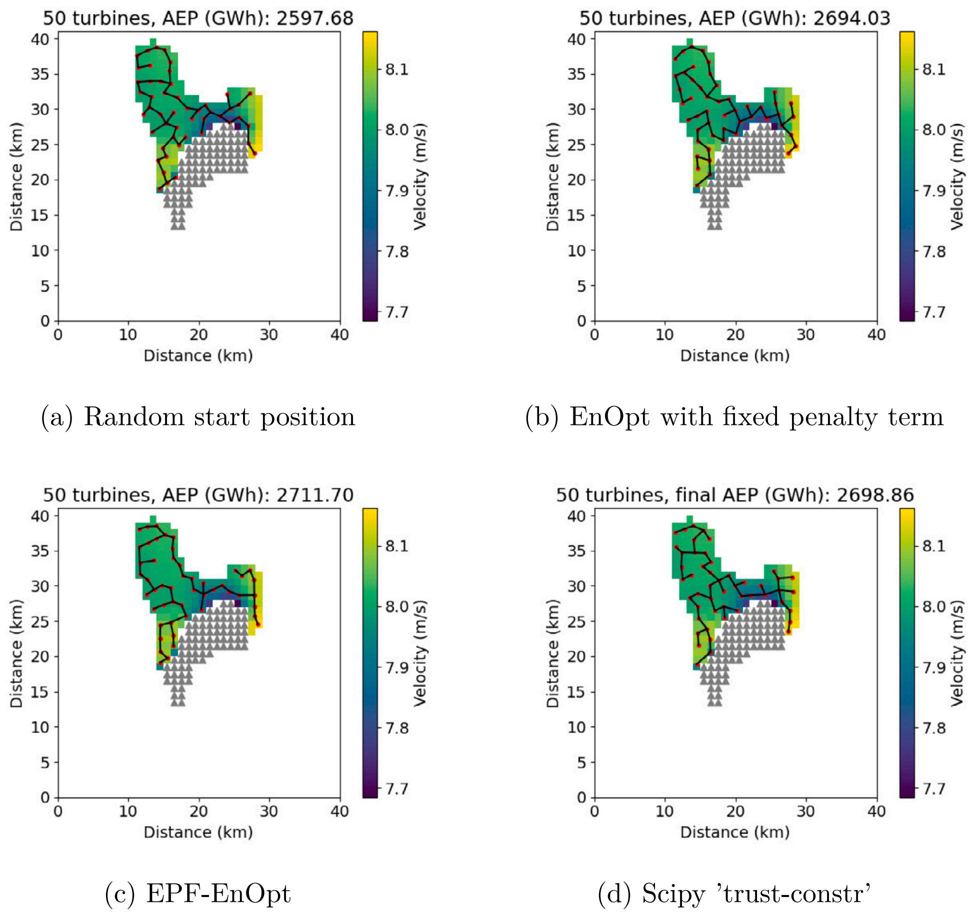
To further test the methods, we perform the optimization 10 times with randomly selected starting positions. We select starting positions by drawing the turbine positions one by one, and check that they are within the right area and satisfy the distance constraints. If not, we draw a new one. (We could have started with a layout that did not fulfill the distance requirement as well, since the increasing penalty in EPF-EnOpt will make the distances large enough in the final layout, but this was not tested here.) The objective functions of the 10 runs are shown in Fig. 6 and in Table 4, and the best layouts obtained for each of the three methods are shown in Fig. 7. Also for the random starting positions EPF-EnOpt performs the best, and at a smaller cost than the trust region method. The average reduction in LCOE is 3,9% with EPF-EnOpt. The results for EPF-EnOpt are equally good with random starting positions as with the manually selected starting positions. EnOpt with a fixed penalty term is the computationally fastest method, but results in significantly higher LCOE then EPF-EnOpt. A fixed penalty term also requires tuning of $r_0$ for a balanced weight between the objective function and the penalty term in order to satisfy the constraints.

We also test the methods on another problem, where we maximize the annual energy production (AEP) without considering any costs. The results are shown in Fig. 8 and Table 5. Also here EPF-EnOpt performs the best, and the increase in AEP is on average 4,0% with this method.

## 5. Conclusion

We have introduced and demonstrated ensemble based methodology to find the optimal layout of wind farms that minimizes levelized cost of energy (LCOE) or maximizes annual energy production (AEP). We have seen that EPF-EnOpt provides better optimization results at a lower computational cost than the constrained trust region method from the Scipy python package. Both with manually selected starting positions and randomly placed turbines EPF-EnOpt consistently obtained good results and fulfilled the distance constraints. The results of EPF-EnOpt were also better than using a single run of EnOpt with a fixed penalty term. The gradual increase of the penalty seems to be beneficial in order to find good layouts. **The reduction in LCOE with EPF-EnOpt was around 1,5% compared to a manually selected starting layout and on average 3,9% with randomly placed turbines. The increase in AEP was on average 4,0% for EPF-EnOpt with random starting positions.**

In this work we have used a fast method for calculating the wakes (PyWake), but if one wants to apply more advanced wake models, the computational cost will be much larger, and ensemble methods could provide an affordable way of optimizing the layout. We note that gradient-free methods will be very time consuming when advanced wake models are used. With EPF-EnOpt, a large part of the




(a) Random start position

(b) EnOpt with fixed penalty term

(c) EPF-EnOpt

(d) Scipy 'trust-constr'

**Fig. 8.** Random start positions and final layout after optimization of annual energy production. Figure (b), (c) and (d) show the best final layout of the 10 runs performed with each method. Figure (a) shows one of the random starting layouts (corresponding to the final layout obtained with EPF-EnOpt).

**Table 4**
The table shows results for minimization of the LCOE. We run 10 experiments with random starting positions. The number of function evaluations is an average over the 10 runs.

| | Mean LCOE (€/MWh) | Std.dev. | Best run | Function evaluations |
|---|---|---|---|---|
| Start | 135.94 | 0.64 | – | – |
| EnOpt w/penalty | 132.05 | 0.347 | 131.42 | 2769 |
| EPF-EnOpt | 130.65 | 0.273 | 130.29 | 11166 |
| Scipy trust-constr | 131.51 | 0.279 | 131.04 | 30209 |

improvement in the objective function is obtained in the first iterations, and it could be possible to reduce the number of iterations further if needed. Another advantage of ensemble methods that we have not utilized here, is the possibility of including uncertainty in the wind resources. Ensemble methods are frequently used in climate modeling and weather predictions, where the uncertainty is represented by an ensemble, as in Swamy et al. [27], and we will include this in future work. EPF-EnOpt could also potentially be combined with a heuristic global method to obtain a good starting position for EPF-EnOpt.

We have done some simplifications in the calculation of the costs, since our main goal of this paper is not to get the most accurate price, but to demonstrate new methodology. For example, we used the same cable between all turbines independently of the amount of electricity it will transfer. One can also include more detailed costs and account for loss in the cables. In future work these simplifications can be removed.

## CRediT authorship contribution statement

**Kjersti Solberg Eikrem:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Rolf Johan Lorentzen:** Conceptualization, Methodology, Software, Writing – review & editing. **Ricardo Faria:** Conceptualization, Software, Formal analysis, Resources, Data curation, Writing – original draft. **Andreas Størksen Stordal:** Conceptualization, Writing – review & editing. **Alexandre Godard:** Conceptualization, Methodology, Writing – review & editing, Project administration, Funding acquisition.

## Acknowledgments

The authors would like to acknowledge the grant received by EEA, Portugal Grant (https://www.eeagrants.gov.pt/en/). The grant number is PT-INNOVATION-0061-Wunderocean, Lda/LOOP Wind. The first author also acknowledge The Research Council of Norway and the industry partners of ImpactWind SouthWest – RCN project number 332034 – for their support. The authors would also like to thank to IPMA (Portuguese Institute for Sea and Atmosphere) for providing the access to meteorological data used to validate the atmospheric model; DROTe (Direção Regional do Ordenamento do Território; Regional Agency for Territorial Planning) for providing all the static data used in the numerical system of models; and CESGA (Supercomputing Center of Galicia) for the access to computational resources to conduct the weather model simulations.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix. Wind data

### A.1. Long-term analysis

The methodology used to generate the wind data in this study consists of a long-term data analysis for the atmosphere and for the ocean, to constitute a Typical Meteorological Year (TMY). To predict the potential of any energetic system, it depends on a local well-established and extensive climatic data base. This analysis must be done with the support of at least 10 years of quality measured data; when this type of data is not available reanalysis data can be used, but with several limitations when compared to in-situ/observational data. Long-term analysis can be performed through really long simulations periods, but the computational cost and time for completing all the calculations increases. However, interannual climatic variability is important for this study to represent mid and long-term signature, that can be represented by one TMY. A TMY is composed by the 12 more representative months from different years over a certain (long-term) period of observational data. Each month of the calendar year it is calculated as the smallest weighted sum of the Finkelstein Schafer (FS) considering 10 years of observational data [40]. Data from all meteorological masts are concatenated into one single dataset, this approach allows the consideration of the local and surrounding climatic features over a larger (or small) regions. After the TMY analysis is completed, we can represent one year considering different 'typical months' considering 10 years of observational data. Then the obtained period from TMY is downloaded to force the regional numerical model's boundary conditions.

### A.2. Climatic data sources

For the TMY analysis, initially 30 years climatology data was considered for wind and waves using reanalysis from ERA-Interim, provided by Dee et al. [41]. However, ERA-Interim spatial and temporal resolution of 0.75° and 6 h are not representative of the local island processes, namely small and short-term events, and variations. To obtain a TMY that characterizes the distribution and variability associated with the islands, wind hourly data was necessary. The data was obtained from "Instituto Português do Mar e da Atmosfera" (IPMA). IPMA has good coverage of meteorological stations in Madeira and Porto Santo Island as represented in Fig. A.9. They cover the period of 10 years of consecutive data with a 10 min timestep. For oceanographic conditions, OSCAR [42] is used. Founded by ESA Data User Element (ESA 2015), OSCAR uses advanced processing tools and models that incorporate satellite and in-situ data to calculate global ocean circulation patterns. It has a coverage period from 1993 to 2016 with a spatial resolution of 0.25°, producing daily outputs.

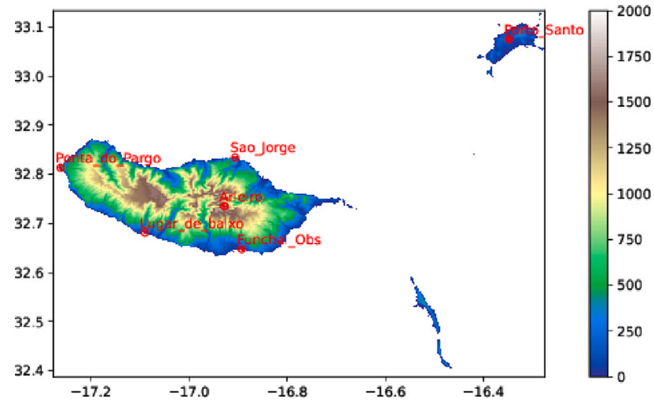



**Fig. A.9.** Madeira topography with IPMA meteorological stations numbered.

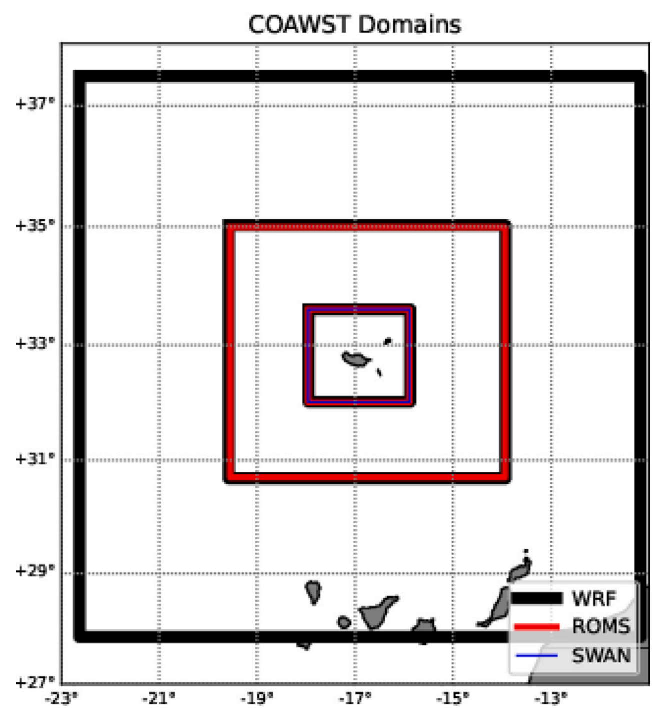



**Fig. A.10.** COAWST domains configuration. WRF has 3 domains presented in black, ROMS 2 domains in red and WW3 the inner domain in blue.

### A.3. COAWST configuration

Hereafter, we focus on the description of the configuration of the Coupled-Ocean–Atmosphere–Wave–Sediment Transport Modeling System (COAWST) numerical framework and its components. Weather Research and Forecast (WRF) has 3 domains centered in Madeira, an outer Atlantic domain with 9 km grid-spacing, nested onto 3 and subsequently 1 km grid-spacing sub-domains (Fig. A.10). The geographic location of the two grids used in the Regional Ocean Modeling System (ROMS) are almost identical to the two inner grids used in WRF and

**Table 5**
The table shows results for maximization of the annual energy production (AEP). We run 10 experiments with random starting positions. The number of function evaluations is an average over the 10 runs.

| | Mean AEP (GWh) | Std.dev. | Best run | Function evaluations |
|---|---|---|---|---|
| Start | 2599.19 | 18.53 | – | – |
| EnOpt w/penalty | 2682.10 | 7.422 | 2692.75 | 3140 |
| EPF-EnOpt | 2704.15 | 8.134 | 2711.70 | 10435 |
| Scipy trust-constr | 2691.18 | 4.709 | 2698.86 | 24765 |

WAVEWATCH III (WW3) is almost identical to the inner domain. To account for the minor differences among grid cell's locations between models, COAWST uses the Spherical Coordinate Remapping and Interpolation Package (SCRIP) [43] that generates the interpolated weights used to remap the data exchanged between the different grids. All grids use a Mercator projection. To allow data exchange between models, COAWST uses the Model Coupling Toolkit (MCT) [44,45]. In our configuration, data is transferred every 30 min between the models, to provide an accurate simulation of ocean–atmosphere fluxes. Simulations done in this article are based on our previous studies based on fully two-way coupled simulations with the Coupled Ocean–Atmosphere Mesoscale Prediction System (COAMPS) or using COAWST in Madeira Archipelago, and the reader is referred to Pullen et al. [46,47],Alves et al. [48],Azevedo et al. [49] for further information. Also, Fig. A.10 shows the geographic limit of each computational domain in relation to the location of Madeira Island, in the North Atlantic at  900 km SW of Iberian Peninsula and  700 km to the west of the Northwestern African coast.

## References


[1] GPCDB Mosetti, Carlo Poloni, Bruno Diviacco, Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm, J. Wind Eng. Ind. Aerodyn. 51 (1) (1994) 105–116.

[2] Ajit.C. Pillai, John. Chick, Lars. Johanning, Mahdi. Khorasanchi, Sebastien. Pelissier, Optimisation of offshore wind farms using a genetic algorithm, International Journal of Offshore and Polar Engineering 26 (03) (2016) 225–234.

[3] S. Rodrigues, Pavol Bauer, Peter A.N. Bosman, Multi-objective optimization of wind farm layouts–complexity, constraint handling and scalability, Renew. Sustain. Energy Rev. 65 (2016) 587–609.

[4] Andrew P.J. Stanley, Owen Roberts, Jennifer King, Christopher J. Bay, Objective and algorithm considerations when optimizing the number and placement of turbines in a wind power plant, Wind Energy Sci. 6 (5) (2021) 1143–1167.

[5] A. Al Shereiqi, B. Mohandes, A. Al-Hinai, M. Bakhtvar, R. Al-Abri, M.S. El Moursi, M. Albadi, Co-optimisation of wind farm micro-siting and cabling layouts, IET Renew. Power Gener. 15 (8) (2021) 1848–1860.

[6] Nicolas Kirchner-Bossi, Fernando Porté-Agel, Wind farm area shape optimization using newly developed multi-objective evolutionary algorithms, Energies 14 (14) (2021) 4185.

[7] Peng Hou, Weihao Hu, Mohsen Soltani, Zhe Chen, Optimized placement of wind turbines in large-scale offshore wind farm using particle swarm optimization algorithm, IEEE Trans. Sustain. Energy 6 (4) (2015) 1272–1282.

[8] Ajit.C. Pillai, John. Chick, Lars. Johanning, Mahdi. Khorasanchi, Offshore wind farm layout optimization using particle swarm optimization, J. Ocean Eng. Mar. Energy 4 (1) (2018) 73–88.

[9] Siyu Tao, Qingshan Xu, Andrés Feijóo, Gang Zheng, Joint optimization of wind turbine micrositing and cabling in an offshore wind farm, IEEE Trans. Smart Grid 12 (1) (2021) 834–844.

[10] Philip. Asaah, Lili. Hao, Jing. Ji, Optimal placement of wind turbines in wind farm layout using particle swarm optimization, J. Mod. Power Syst. Clean Energy 9 (2) (2021) 367–375.

[11] K. Chen, M.X. Song, X. Zhang, S.F. Wang, Wind turbine layout optimization with multiple hub height wind turbines using greedy algorithm, Renew. Energy 96 (2016) 676–686.

[12] I. Ulku, C. Alabas-Uslu, A new mathematical programming approach to wind farm layout problem under multiple wake effects, Renew. Energy 136 (2019) 1190–1201.

[13] Tengjun Zuo, Yuchen Zhang, Ke Meng, Ziyuan Tong, Zhao Yang Dong, Yang Fu, A two-layer hybrid optimization approach for large-scale offshore wind farm collector system planning, IEEE Trans. Ind. Inform. 17 (11) (2021) 7433–7444.

[14] Alireza. Emami, Pirooz. Noghreh, New approach on optimization in placement of wind turbines within wind farm by genetic algorithms, Renew. Energy 35 (7) (2010) 1559–1564.

[15] Michele Samorani, The wind farm layout optimization problem, in: Handbook of wind power systems, Springer, 2013, pp. 21–38.

[16] José F. Herbert-Acero, Oliver Probst, Pierre-Elouan Réthoré, Gunner Chr Larsen, Krystel K. Castillo-Villar, A review of methodological approaches for the design and optimization of wind farms, Energies 7 (11) (2014) 6930–7016.

[17] Javier Serrano González, Manuel Burgos Payán, Jesús Manuel Riquelme Santos, Francisco González-Longatt, A review and recent developments in the optimal wind-turbine micro-siting problem, Renew. Sustain. Energy Rev. 30 (2014) 133–144.

[18] Peng. Hou, Jiangsheng. Zhu, Kuichao. Ma, Guangya. Yang, Weihao. Hu, Zhe. Chen, A review of offshore wind farm layout optimization and electrical system design methods, J. Mod. Power Syst. Clean Energy 7 (5) (2019) 975–986.

[19] Karthik Balasubramanian, Sudhakar Babu Thanikanti, Umashankar Subramaniam, N. Sudhakar, Sam Sichilalu, A novel review on optimization techniques used in wind farm modelling, Renew. Energy Focus 35 (2020) 84–96.

[20] F. Azlan, J.C. Kurnia, B.T. Tan, M.-Z. Ismadi, Review on optimisation methods of wind farm array under three classical wind condition problems, Renew. Sustain. Energy Rev. 135 (2021) 110047.

[21] Rolf Johan Lorentzen, Aina Berg, Geir Nævdal, Erlend H. Vefring, A new approach for dynamic optimization of water flooding problems, in: Intelligent Energy Conference and Exhibition, OnePetro, 2006.

[22] Yan Chen, Dean S. Oliver, Dongxiao Zhang, Efficient ensemble-based closed-loop production optimization, Spe J. 14 (04) (2009) 634–645.

[23] Yan Chen, Dean S. Oliver, Ensemble-based closed-loop optimization applied to brugge field, SPE Reserv. Eval. Eng. 13 (01) (2010) 56–71.

[24] R.M. Fonseca, Olwijn Leeuwenburgh, P.M.J. Van den Hof, Jan-Dirk Jansen, Improving the ensemble-optimization method through covariance-matrix adaptation, SPE J. 20 (01) (2015) 155–168.

[25] Andreas S. Stordal, Slawomir P. Szklarz, Olwijn Leeuwenburgh, A theoretical look at ensemble-based optimization in reservoir management, Math. Geosci. 48 (4) (2016) 399–417.

[26] Rahul Rahul-Mark Fonseca, Bailian Chen, Jan Dirk Jansen, Albert Reynolds, A stochastic simplex approximate gradient (stosag) for optimization under uncertainty, Internat. J. Numer. Methods Engrg. 109 (13) (2017) 1756–1776.

[27] S. Krishna Swamy, S. Szklarz, R.M. Fonseca, B.H. Bulder, Combined wind turbine design and wind farm layout optimisation under wind resource uncertainty, J. Phys. Conf. Ser. 1618 (4) (2020) 042030, http://dx.doi.org/10.1088/1742-6596/1618/4/042030.

[28] S.P. Szklarz, E. Barros, D. Berawala, B.K. Hegstad, K.R. Petvipusit, How could reservoir engineers harvest wind energy? Pract. Parametr. Approaches Wind Farm Layout Optim. 2022 (1) (2022) 1–5.

[29] Jorge Nocedal, Stephen J Wright, Numerical optimization, 2nd edition., Springer, 2006.

[30] Micheal B. Oguntola, Rolf J. Lorentzen, Ensemble-based constrained optimization using an exterior penalty method, J. Pet. Sci. Eng. 207 (2021) 109165.

[31] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R.J. Nelson, Eric Jones, Robert Kern, Eric Larson, C.J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, SciPy 1.0 contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.

[32] Albert Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, 2005.

[33] AREAM Agência Regional da Energia e Ambiente da Região Autónoma da Madeira, Área potencial de energia renovável offshore, 2017, URL https://webgis.dgrm.mm.gov.pt/arcgis/rest/services/PSOEM_PosCPub/PSOEM_Madeira_PCP2/MapServer/10, Accessed: 2021-05-04.

[34] The National Renew. Energy Laboratory (NREL), Nrel wind turbine power curve archive, 2022, URL https://https://github.com/NREL/turbine-models/blob/master/Offshore/IEA_15MW_240_RWT.csv. Accessed: 2022-2-7.

[35] Mads M. Pedersen, Paul van der Laan, Mikkel Friis-Møller, Jennifer Rinker, Pierre-Elouan Réthoré, Dtuwindenergy/pywake: pywake, 2019, http://dx.doi.org/10.5281/zenodo.2562662.

[36] Majid Bastankhah, Fernando Porté-Agel, A new analytical model for wind-turbine wakes, Renew. Energy 70 (2014) 116–123.

[37] Philipp Beiter, Tyler Stehly, Patrick Duffy, 2019 Cost of wind energy review, 2020, https://www.nrel.gov/docs/fy21osti/78471.pdf.

[38] K. Nieradzinska, C. MacIver, S. Gill, G.A. Agnew, O. Anaya-Lara, K.R.W. Bell, Optioneering analysis for connecting dogger bank offshore wind farms to the gb electricity network, Renew. Energy 91 (2016) 120–129.

[39] Krishna. Naidoo, MiSTree: A python package for constructing and analysing minimum spanning trees, J. Open Source Softw. 4 (42) (2019) 1721, http://dx.doi.org/10.21105/joss.01721.

[40] Jack M. Finkelstein, Ray E. Schafer, Improved goodness-of-fit tests, Biometrika 58 (3) (1971) 641–645.

[41] Dick P. Dee, S.M. Uppala, Adrian J. Simmons, Paul Berrisford, Paul Poli, Shinya Kobayashi, U. Andrae, M.A. Balmaseda, G. Balsamo, D.P. Bauer, et al., The era-interim reanalysis: Configuration and performance of the data assimilation system, Q. J. R. Meteorol. Soc. 137 (656) (2011) 553–597.

[42] Eric S. Johnson, Fabrice Bonjean, Gary S.E. Lagerloef, John T. Gunn, Gary T. Mitchum, Validation and error analysis of oscar sea surface currents, J. Atmos. Ocean. Technol. 24 (4) (2007) 688–701.

[43] Philip W. Jones, First-and second-order conservative remapping schemes for grids in spherical coordinates, Mon. Weather Rev. 127 (9) (1999) 2204–2210.

[44] Robert. Jacob, Jay. Larson, Everest. Ong, M× n communication and parallel interpolation in community climate system model version 3 using the model coupling toolkit, Int. J. High Perform. Comput. Appl. 19 (3) (2005) 293–307.

[45] Jay. Larson, Robert. Jacob, Everest. Ong, The model coupling toolkit: A new fortran90 toolkit for building multiphysics parallel coupled models, Int. J. High Perform. Comput. Appl. 19 (3) (2005) 277–292.

[46] Julie Pullen, Richard Allard, Hyodae Seo, Arthur J. Miller, Shuyi Chen, Luciano Ponzi Pezzi, Travis Smith, Philip Chu, José Alves, Rui Caldeira, Coupled ocean-atmosphere forecasting at short and medium time scales, J. Mar. Res. 75 (6) (2017) 877–921a.

[47] Julie Pullen, Rui Caldeira, James D. Doyle, Paul May, Ricardo Tomé, Modeling the air-sea feedback system of madeira island, J. Adv. Modelling Earth Syst. 9 (3) (2017) 1641–1664b.

[48] José M.R. Alves, Rui M.A. Caldeira, Pedro M.A. Miranda, Dynamics and oceanic response of the madeira tip-jets, Q. J. R. Meteorol. Soc. 146 (732) (2020) 3048–3063.

[49] Cátia C. Azevedo, Carolina M.L. Camargo, José Alves, Rui M.A. Caldeira, Convection and heat transfer in island (warm) wakes, J. Phys. Oceanogr. 51 (4) (2021) 1187–1203.