Offshore Wind Farm Layout Optimization with Alignment Constraints

Paul Malisani[1], Tristan Bartement[2], and Pauline Bozonnet[3]

[1]IFP Energies nouvelles, Applied Mathematics Department, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France
[2]IFP Energies nouvelles, Scientific Computing Department, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France
[3]GreenWITS, Rond-Point de l'échangeur, les Levées, 69360 Solaize, France

Correspondence: Paul Malisani (paul.malisani@ifpen.fr)

Abstract. Wind farm layout optimization involves placing wind turbines in a defined domain to minimize the expected production losses due to wake effects within the wind farm. Due to navigational regulations and risks, tenders for offshore wind farms often impose so-called alignment constraints, i.e., wind turbines must be located at the intersections of a grid formed by parallelograms. The shape and orientation of these parallelograms are to be determined to minimize wake losses. Mathematically, this problem belongs to the class of non-convex, non-linear mixed-integer programming problems, known to be highly complex. The literature has not yet investigated the wind farm layout optimization problem under alignment constraints. This paper makes two contributions. First, we propose a modelization of the AEP maximization with alignment constraints as a mixed-integer nonlinear problem, where the continuous parameters are the parallelogram-based tiling parameters and the discrete variables are the turbines' positions at the tiling's intersections. Second, we provide a heuristic derived from the DEBO algorithm developed by the same team and presented in Thomas et al. (2023). The proposed method is the subject of international patent application number WO 2024/061627.

1 Introduction

Selecting the proper layout is a crucial task when building a wind farm. A layout far from optimal is prone to significant loss of expected Annual Energy Production (AEP) due to wake effects within the farm. Having an optimized method of turbine placement in a given area helps maximize energy production over the wind farm's lifespan. In its full generality, the problem of optimizing a wind farm layout is a complex one for several reasons. The first is that computing a given farm's mean annual energy production is numerically complex, i.e., evaluating the optimization problem's objective function is computationally time-consuming LoCascio et al. (2024); Porté-Agel et al. (2020). The second difficulty is that the problem is not convex, i.e., neither the objective function nor the minimization set are convex. As a result, the problem of wind farm layout optimization necessitates the development of dedicated optimization tools. Wind farm optimization has been the subject of much scientific research; see, for example, Herbert-Acero et al. (2014); Fischetti and Pisinger (2019); Hou et al. (2019) for a list of reviews of such methods. Let us now focus on recent contributions to the field. In Quick et al. (2023), the authors develop a stochastic gradient-based method for optimizing wind farm layouts. The presented algorithm is developed for circular or square domains and could be easily extended to convex domains but not to non-convex or non-connected ones. In Kumar and Sharma (2023),

the authors use a teaching-learning-based algorithm to solve the wind farm layout optimization problem for a circular domain. In Liang and Liu (2023), the authors use genetics and particle swarm algorithms to solve the problem on a square domain. In Fischetti and Fischetti (2022), the authors propose a mixed-integer linear programming model to solve turbine placement and cable routing optimization problems. In Kunakote et al. (2022), the authors compare twelve meta-heuristic methods for wind farm layout optimization. The presented methods do not rely on any assumption on the shape of the admissible domain. However, the method relies on a very coarse discretization of the domain and can be numerically intractable using a finer one. In Thomas et al. (2023), the authors compare eight wind farm optimization methods and provide a benchmark case study for comparing algorithm performances. The proposed benchmark is highly complex since the admissible domain is neither convex nor connected. However, this contribution does not account for any alignment constraints. Finally, in Stanley and Ning (2019), the authors propose a so-called inner-grid wind farm layout parameterization that satisfies strong alignment constraints. However, the authors assume a one-to-one correspondence between this parameterization and the layout configuration, which dramatically reduces the degree of freedom and potentially leads to far-from-optimal solutions. In fact, despite the extensive literature on wind farm optimization, to the best of our knowledge, no other method can handle turbine alignment constraints. By alignment constraint, we place the turbines at the intersections of a regular grid composed of parallelograms, whose shape and orientation are to be determined, while considering the possibility of not occupying all the grid's intersections. This possibility is a key feature of the proposed algorithm, and its interest is illustrated in the numerical examples presented in the paper. Far from being just an academic question, turbine alignment constraints are often imposed by maritime authorities on developers in practice to ensure the safe navigation of boats within the wind farm (see, for example, the following technical reports Ministry of Transport of the French Republic (2017); Maritime and Coastguard Agency (2012)). The contribution of this paper is to provide an optimization algorithm for wind farm layout optimization that can handle alignment constraints. Mathematically, these alignment constraints render the wind farm layout optimization problem a mixed-integer nonlinear programming (MINLP) problem. The integer variables represent the positions of the turbines, which are restricted to the finite set of grid intersections within the admissible domain. The continuous variables are the grid parameters, that is to say, the size and orientation of the grid's unit parallelogram. The non-linearity mainly stems from the wind farm's Annual Energy Production (AEP) as a function of the optimization parameters. These problems are generally extremely difficult to solve. As detailed in Burer and Letchford (2012), solving algorithms for non-convex MINLPs fall into two different categories: exact methods and heuristic-based methods. Exact methods often rely on branch-and-bound methods Papadimitriou and Steiglitz (1998) or separation properties of the objective function. Heuristic methods include tabu research Exler et al. (2008), particle swarm algorithms Yiqing et al. (2007); Young et al. (2007), genetic algorithms Schlüter et al. (2009) or local search methods Liberti et al. (2011). The strategy adopted in this paper is to adapt the DEBO method developed by the authors in Thomas et al. (2023) to the problem at hand. This method is a local-search-based approach coupled with an exploration heuristic for optimizing parameters. The paper is organized as follows. In section 2, we introduce useful notations and definitions. In section 3, we describe the aligned-layout optimization problem; that is to say, we present the objective function, the constraints, and the optimization variables. We fully describe the optimization algorithm in section 4. In section 5, we take up the benchmark presented in Thomas et al. (2023) and add the alignment constraints, and we conduct a thorough study on the setting of the

optimization algorithm hyper-parameters. In section 6, we illustrate the impact of the alignment-grid parameters exploration method on the AEP and prove that an efficient exploration method yields a strong improvement of the AEP. Finally, in section 7, we give the conclusions of this work and draw up research perspectives on the subject.

2 Notations and definitions

Throughout the paper we will use recurrently the following notations

- $\mathbb{R}, \mathbb{R}_+$ denote respectively the set of real numbers and the set of non negative real numbers.
- $\mathbb{Z}, \mathbb{Z}_*$ denote respectively the set of integers and the set of non zero integers
- $\Omega \in \mathbb{R}^2$: Two dimensional domain where turbines can be planted
- $N_{\max}$: Maximal number of turbines to be placed within the admissible domain E
- D_{turb}: Turbine diameter
- $D_{\min}$: Minimal distance between turbines
- $D_{\max}$: Maximal distance between turbines
- w_s: Wind speed
- w_d: Wind direction
- $\top$: Logical True
- $\bot$: Logical False
- $\neg$: Logical negation
- $\wedge$: Logical and operator
- $\vee$: Logical or operator

Definition 1 (Wind farm)**.** *A capital bold character associated with a subscript such as $\mathbf{F}_n$ denotes a n-turbines wind farm. Mathematically $\mathbf{F}_n$ is a function satisfying:*

$$\mathbf{F}_n : \{1, \dots, n\} \ni k \mapsto (x_k \;\; y_k)^\top \in \mathbb{R}^2 \tag{1}$$

where $(x_k \;\; y_k)^\top$ is the position of the k^{th}-turbine. Let $\mathbf{F}_n$ and let $(x \;\; y)^\top \in \mathbb{R}^2$, we denote $\mathbf{H}_{n+1} := \mathbf{F}_n \oplus (x \;\; y)$ the wind farm defined as follows

$$\mathbf{H}_{n+1}(k) := \begin{cases} \mathbf{F}_n(k) & \text{if } k \leq n \\ (x \;\; y)^\top & \text{if } k = n+1 \end{cases} \tag{2}$$

Definition 2 (Wind farm Power Production). *We denote* $\mathcal{P} : (\mathbf{F}_n, w_s, w_d) \mapsto \mathbb{R}_+$ *the power production of the wind farm* $\mathbf{F}_n$ *for a wind speed* w_s *and a wind direction* w_d.

Definition 3 (Expected Power Production and Annual Energy Production). *We denote* $\mathrm{W} := (w_s, w_d) \in \mathbb{R}_+ \times [0, 2\pi)$ *the wind random variable, and we denote* $\mathbb{P}_{\mathrm{W}}$ *the probability measure on* $\mathbb{R}_+ \times [0, 2\pi)$ *associated with the random variable* W. *Finally, we denote* $\mathbb{E}_{\mathrm{W}}$ *the expectation with respect to the probability* $\mathbb{P}_{\mathrm{W}}$ *and the Expected Power Production of a wind farm* $\mathbf{F}_n$ *writes*

$$\mathbb{E}_{\mathrm{W}}(\mathcal{P}(\mathbf{F}_n, w_s, w_d)) := \int_{\mathbb{R}_+ \times [0,2\pi)} \mathcal{P}(\mathbf{F}_n, w_s, w_d) \mathrm{d}\mathbb{P}_{\mathrm{W}}(w_s, w_d) \tag{3}$$

and the Annual Energy Production of a wind farm $\mathbf{F}_n$ *denoted* $\mathrm{AEP}(\mathbf{F}_n)$ *writes*

$$\mathrm{AEP}(\mathbf{F}_n) := 8760 \,.\, \mathbb{E}_{\mathrm{W}}(\mathcal{P}(\mathbf{F}_n, w_s, w_d)) \tag{4}$$

Finally, when the probability is discrete, i.e. $\mathbb{P}_{\mathrm{W}} := \sum_{n=1}^{N_{\mathrm{W}}} p(w_s^n, w_d^n)\delta(w_s^n, w_d^n)$, *where* $p(w_s^n, w_d^n)$ *is the probability of wind* $W = (w_s^n, w_d^n)$ *and* $\delta(w_s^n, w_d^n)$ *is the Dirac delta function at* (w_s^n, w_d^n), *the AEP computation writes*

$$\mathrm{AEP}(\mathbf{F}_n) := 8760 \sum_{n=1}^{N_{\mathrm{W}}} p(w_s^n, w_d^n) \mathcal{P}(\mathbf{F}_n, w_s^n, w_d^n) \tag{5}$$

3 Optimization model for layout optimization with alignment constraints

The optimization problem we are interested in consists of optimizing the grid configuration and the turbine placement on the intersections of this grid. This optimization problem is a non-linear mixed integer programming problem, a class of problems known to be challenging to solve. In this section, we describe the parameterization of our problem.

3.1 Grid parameterization

To write the optimization problem, we parameterize the grid using 6 parameters $(r_1, r_2, \theta_1, \theta_2, v_x, v_y)$ as represented on fig. 1. The grid is a parallelogram-based tiling of the plane, the parameters r_1, r_2 are the two sides' length of the parallelogram, the parameter θ_1 (resp. θ_2) is the angle formed between the side of the parallelogram of length r_1 (resp. r_2) and the x-axis. Finally the parameters v_x, v_y is the offset between the origin of the Cartesian and the parallelogram-based grids. Using this parameterization we define the change-of-basis matrix from the canonical basis denoted $\mathcal{B}_0$ to the grid basis denoted $\mathcal{B}(\theta_1, \theta_2, r_1, r_2)$ as follows

$$P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} = \begin{pmatrix} r_1\cos(\theta_1) & r_2\cos(\theta_2) \\ r_1\sin(\theta_1) & r_2\sin(\theta_2) \end{pmatrix} \tag{6}$$

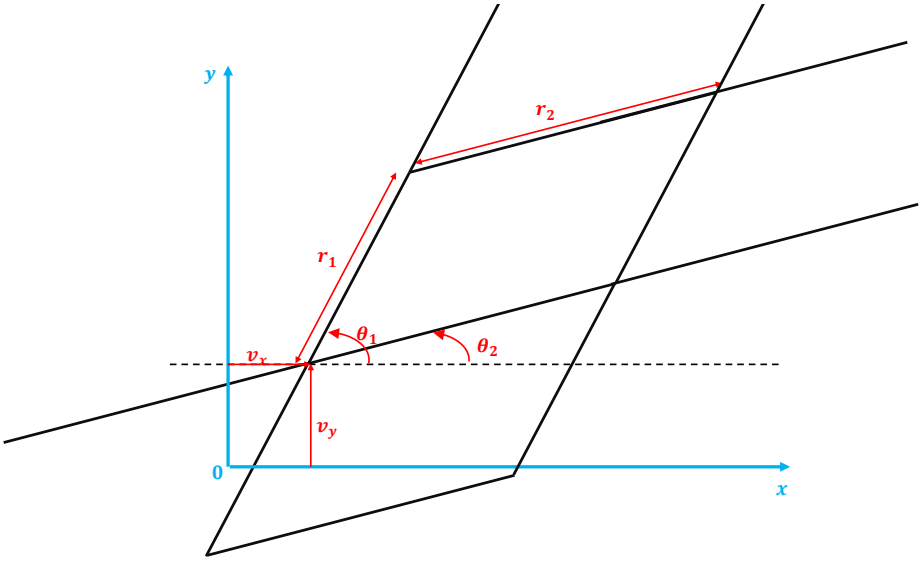


Figure 1. Grid parameterization for aligned layout.

In the grid-basis coordinates, any intersection point p writes as follows:

$$p := \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} + \left(P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)}\right)^{-1} \begin{pmatrix} v_x \\ v_y \end{pmatrix} \quad k_1, k_2 \in \mathbb{Z}$$

This parameterization in grid basis is, in turn, equivalent to

$$p = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} + \begin{pmatrix} \Delta_1 \\ \Delta_2 \end{pmatrix} \quad k_1, k_2 \in \mathbb{Z} \text{ and } \Delta_1, \Delta_2 \in [0,1) \tag{7}$$

3.2 Layout parameterization

Using the grid parameterization described in section 3.1, an aligned layout has its turbines located on the intersections of the grid which writes

$$\mathbf{F}[\mathcal{B}]_n(i) : \{1,\dots,n\} \ni i \mapsto \begin{pmatrix} k_1^i & k_2^i \end{pmatrix}^\top + \begin{pmatrix} \Delta_1 & \Delta_2 \end{pmatrix}^\top, \quad k_1^i, k_2^i \in \mathbb{Z};\ \Delta_1, \Delta_2 \in [0,1) \tag{8}$$

The corresponding wind farm in canonical coordinates $\mathbf{F}_n$ thus writes

$$\mathbf{F}_n(i) = P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} \mathbf{F}[\mathcal{B}]_n(i),\ \ i = 1,\dots,n \tag{9}$$

Thanks to the parameterization from eq. (8), the alignment constraint translates into an integer constraint on the optimization variables (k_1^i, k_2^i).

3.3 Optimization problem

We are now ready to write the general wind farm layout optimization problem with alignment constraints. This optimization problem consists of maximizing the wind farm's Annual Energy Production as defined in eq. (4). This writes

$$\max_{(k_1^i,k_2^i)_{i=1,\dots,N_{\max}},\Delta_1,\Delta_2,r_1,r_2,\theta_1,\theta_2} \mathrm{AEP}(\mathbf{F}[\mathcal{B}]_{N_{\max}}) \tag{10}$$

under the following constraints

$$k_1^i, k_2^i \in \mathbb{Z};\ \ i = 1,\dots,N_{\max} \tag{11}$$

$$\left|k_1^i - k_1^j\right| + \left|k_2^i - k_2^j\right| \geq 1;\ \ \forall i \neq j \tag{12}$$

$$\Delta_1, \Delta_2 \in [0,1) \tag{13}$$

$$\mathbf{F}_{N_{\max}}(i) \in \Omega;\ \ i = 1,\dots,N_{\max} \tag{14}$$

$$\theta_1 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{15}$$

$$\theta_2 \in \left[-\frac{\pi}{2}, \theta_1\right) \tag{16}$$

$$D_{\min} \leq \min_{z \in \mathbb{Z}_*^2} \left\| P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} z \right\| \tag{17}$$

Equations (11) to (13) ensure that the turbines are located at the intersections of the grid defined by the parameters $(r_1, r_2, \theta_1, \theta_2, v_x, v_y)$ and thus, that the turbines are aligned along the directions θ_1 and θ_2. The constraint from eq. (14) guarantees that the turbines are located in the admissible domain. Constraints defined in eqs. (15) and (16) allow to generate all possible parallelogram-based grids. We only consider $\theta_1 \in (-\pi/2, \pi/2]$ since $\theta_1 > \pi/2$ (resp. $\theta_1 < -\pi/2$) yields the same tiling as $\theta_1 - \pi/2$ (resp. $\theta_1 + \pi/2$). Finally, the constraint from eq. (17) ensures that no grid's intersections are closer to each other than $D_{\min}$. Since the turbines are necessarily located on these intersections, this constraint ensures that turbines are at least $D_{\min}$ distant from each other.

4 Solving Algorithm

4.1 General description of the proposed solving algorithm

The method presented in this paper belongs to the category of heuristic-based methods and consists of the following steps.

STEP 1 The first step consists of computing the set $R_{1,2}$ of parameters (r_1, r_2) by discretization of the space $[D_{\min}, D_{\max}]^2$ using a grid size of Δr. There is no upper limit to the $D_{\max}$ value. However, it is useless to set this parameter at a large value. Indeed, beyond a certain value $d > D_{\min}$, any shape parameter configuration such that $r_1, r_2 > d$ will produce coarse grids with fewer than $N_{\max}$ admissible intersections.

STEP 2 The second step consists in reducing the size of the angle-search space. To do so, for each couple $(r_1, r_2) \in R_{1,2}$, we discretize the search space $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \left[-\frac{\pi}{2}, \theta_1\right)$ using a discretization size of $\Delta\theta$. Then, for each grid

configuration $(r_1, r_2, \theta_1^k, \theta_2^k)_k$ satisfying eq. (17) we compute the AEP of an elementary 4 turbines wind farm. Then, we store the N_θ best angle configurations (θ_1, θ_2) in a angle set Θ. This latter set Θ is the reduced angle-search space.

STEP 3 Then, we compute a set of grid configurations $(r_1, r_2, \theta_1, \theta_2)$ denoted grids defined as grids $:= \{(r_1, r_2, \theta_1, \theta_2) : (r_1, r_2) \in R_{1,2},\ (\theta_1, \theta_2) \in \Theta,\ eq.\ (17)\ \text{holds}\}$

STEP 4 For each explored shape configuration $(r_1, r_2, \theta_1, \theta_2)$, we compute an optimal layout using a greedy algorithm for placing the $N_{\max}$-turbines on the intersections of the grid and using a local search optimization method to move the turbines on the intersections. The sequence of greedy initialization followed by a local search method has already been proved efficient for wind farm layout optimization without alignment constraints, see the DEBO algorithm from Thomas et al. (2023).

STEP 5 Return best overall wind farm layout.

Finally, the corresponding pseudo-code is displayed in algorithm 1. As one can see on line 13, the step 4, which is the most computationally demanding, can be run in parallel.

4.2 Angle search space reduction and grid configuration selection

In this section, we describe the first part of the algorithm which consists in finding a set of grid configurations $(r_1, r_2, \theta_1, \theta_2)$ of reasonable size and to perform a complete wind farm layout optimization for each element of this set. To do so, we first compute the set $R_{1,2}$ of parameters (r_1, r_2) by discretizing the space $[D_{\min}, D_{\max}]^2$ using a discretization of size Δr. Then, for each $(r_1, r_2) \in R_{1,2}$, discretize the following search space

$$S := \left\{ (\theta_1, \theta_2) \in \left[-\frac{\pi}{2}, \frac{\pi}{2} \right]^2 : \theta_1 \geq -\frac{\pi}{2} + \Delta\theta,\ \theta_2 \leq \theta_1 - \Delta\theta \right\} \tag{18}$$

with an angle discretization parameter $\Delta\theta$. We denote S_D this discrete search space. Then, for each $(r_1, r_2) \in R_{1,2}$, we define the corresponding angle search space Θ_{r_1, r_2} as follows

$$\Theta_{r_1, r_2} := \{(\theta_1, \theta_2) \in S_D : \text{eq. (17) holds}\} \tag{19}$$

Then, for each configuration $(r_1, r_2, \theta_1^k, \theta_2^k)_k$ with $(\theta_1^k, \theta_2^k) \in \Theta_{r_1, r_2}$ we compute the AEP of the elementary farms $(\mathbf{F}[\mathcal{B}(r_1, r_2, \theta_1^k, \theta_2^k)]_4)_k$ defined as follows

$$\mathbf{F}[\mathcal{B}(r_1, r_2, \theta_1^k, \theta_2^k)]_4(i) := \begin{cases} (0\ \ 0)^\top & \text{if } i = 1 \\ (1\ \ 0)^\top & \text{if } i = 2 \\ (0\ \ 1)^\top & \text{if } i = 3 \\ (1\ \ 1)^\top & \text{if } i = 4 \end{cases} \tag{20}$$

Algorithm 1 Aligned_Optimization$(\Omega, N_{\max}, N_\theta, D_{\min}, D_{\max}, \Delta r, \Delta\theta)$

{**STEP 1** : Compute $R_{1,2}$}
$R_{1,2} \leftarrow$ GenerateR1R2$(\Delta r, D_{\min}, D_{\max})$

{**STEP 2** : Compute Reduced Angles Search Space Θ}
$\Theta \leftarrow$ ReduceSearchSpace$(R_{1,2}, \Delta\theta, D_{\min}, D_{\max}, D_{\text{turb}}, N_\theta)$

{**STEP 3** : Generate compute admissible intersections for each grid configurations}
grids $\leftarrow$ GenerateConfigs$(R_{1,2}, \Theta, D_{\min}, D_{\max}, D_{\text{turb}})$
intersections_sets $\leftarrow$ ComputeAllIntersections(grids, $D_{\min}, \Omega$)

{**STEP 4** : Compute optimal layout for each set of intersections}
layouts $\leftarrow \emptyset$
for (intersections, $(r_1, r_2, \theta_1, \theta_2)$) $\in$ intersections_sets {Run in parallel} **do**
  $(\mathbf{F}_{N_{\max}}, \text{aep}) \leftarrow$ PlaceTurbines(intersections)
  layouts $\leftarrow$ layouts$\bigcup\{(\mathbf{F}_{N_{\max}}, \text{aep}, \text{intersections}, (r_1, r_2, \theta_1, \theta_2))\}$
end for
layouts $\leftarrow$ sort(layouts) by aep in descending order

{**STEP 5** : Get best overall layout}
return layouts(0)

and sort the couples $(\theta_1^k, \theta_2^k)_i$ by decreasing order of AEP. Finally, for each $(r_1, r_2) \in R_{1,2}$, we store in the set Θ the best N_θ angles configuration $(\theta_1^k, \theta_2^k) \in \Theta_{r_1,r_2}$. Then, the continuous variables search-space denoted grids consists in all the combinations of the $(r_1, r_2) \in R_{1,2}$ explored with all the angles configuration from Θ, i.e.

$$\text{grids} := \{(r_1, r_2, \theta_1, \theta_2) : (r_1, r_2) \in R_{1,2},\ (\theta_1, \theta_2) \in \Theta :\ eq.\ (17)\ \text{holds}\} \tag{21}$$

The corresponding algorithm in pseudo-code is described in algorithm A1, algorithm A2, algorithm A3, algorithm A4.

4.3 Compute intersections for each grid configuration

This part of the algorithm consists of traversing grids(k) and, for each configuration $(r_1^k, r_2^k, \theta_1^k, \theta_2^k)$, calculating the maximum number of intersections located in the admissible domain Ω and their positions. If this set of intersections has more than $N_{\max}$ elements and if all intersections are $D_{\min}$-apart from each other, this set of intersections is stored in a set of set-of-intersections that we denote intersections_sets. The corresponding algorithm is written in pseudo-code in algorithm A5.

4.4 Optimize turbines placement

4.4.1 Greedy Initialization

Given a grid configuration $(r_1, r_2, \theta_1, \theta_2, \Delta_1, \Delta_2)$, a greedy algorithm is used to sequentially place $N_{\max}$ turbines on the admissible intersections. This algorithm consists in sequentially placing the turbines on the best possible empty intersection, in the sense of AEP maximization, until $N_{\max}$ turbines are placed. The corresponding algorithm in pseudo-code is given in algorithm A6.

4.4.2 Local Search

This part of the algorithm sequentially moves each turbine in random order from its current intersection to a free one if it provides a strict increase in AEP. The algorithm stops when a complete course of all the turbines has been made without a single one being moved. When the number of intersections in the admissible domain is much bigger than $N_{\max}$, one can explore a subset of the free intersections. For example, one can explore the p closest intersections from the turbine to be moved or select p random free intersections. In this case, the size of the subset to explore and its definition, p, are user-defined parameters. The algorithm in pseudo-code is given in algorithm A7.

4.4.3 Turbine placement optimization

Finally, given a set of intersections, the optimization algorithm for optimal turbine placement consists of using sequentially the greedy initialization and the local search algorithm as described in pseudo-code in algorithm A9.

5 Numerical Examples

5.1 Problem presentation

For this numerical example, we use the same case study as in Thomas et al. (2023), whose data are available in Baker et al. (2021). This case study was created within the International Energy Association (IEA) Wind Task 37, and is based on the Borssele III and IV wind farms. Of particular interest in this case study is the presence of five disconnected boundary regions and concave boundary features. The turbines are 10 MW machines with 198 m rotor diameters based on the IEA 10 MW reference wind turbine (Bortolotti et al. (2019)). For the AEP computation, we also use the same algorithm as in Thomas et al. (2023). This method is based on a simple Gaussian wake model based on Bastankhah's Gaussian wake model (Bastankhah and Porté-Agel (2016)), and presented in the IEA case study 3 and 4 announcement documents (Baker et al. (2021)), to calculate wind speeds at each turbine in the wind farm. However, any other AEP computation software, such as FLORIS, can be used with the presented algorithm as long as the computation time of the AEP is fast enough. Indeed, our optimization algorithm

requires a large number of AEP evaluations. Finally, the AEP is computed using a windrose discretized as follows

$$\text{wind direction bins} = \{0, 1, \ldots, 359\}$$

$$\text{wind speed bins} = \{0.90,\ 1.98,\ 3.18,\ 4.40,\ 5.64,\ 6.87,\ 8.11,\ 9.35,\ 10.59,\ 11.83,\ 13.07,\ 14.31,\ 15.56,\ 16.80,\ 18.04,\ 19.28,\ 20.52,\ 21.77,\ 23.01,\ 24.25\}$$

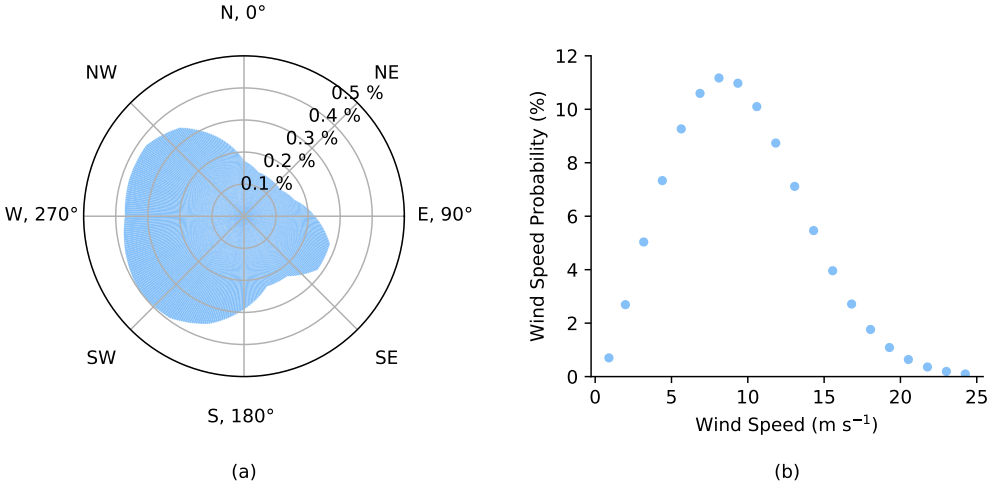


Figure 2. This figure, reproduced from Thomas et al. (2023), displays the full wind resource used for evaluating the final wind farm layouts. (a) The wind direction probability (360 bins). (b) A representative wind speed probability distribution (20 bins).

5.2 Influence of the hyper-parameters on the AEP and the computation time

The optimization procedure described in section 4 requires setting 5 hyper-parameters $D_{\min}$, $D_{\max}$, $\Delta\theta$, N_θ, and Δr. The turbine's manufacturer usually sets $D_{\min}$ at a fixed value. In this example, we set $D_{\min} = 2D_{\text{turb}}$. The parameter $D_{\max}$ must be chosen large enough to allow a good exploration range for the grid parameters (r_1, r_2). However, setting $D_{\max}$ with a significant value generates grids with a number of admissible intersections smaller than the number of turbines to be placed, i.e., in generating non-admissible layouts. Therefore, we have set $D_{\max} = 6D_{\text{turb}}$. The parameter $\Delta\theta$ should be chosen to the minimal value such that the wake model used to compute the AEP is valid. For this example, we set $\Delta\theta = 1°$. The remaining hyper-parameters, N_θ, and Δr, dramatically affect the optimization procedure regarding AEP value and computation time. Indeed, as explained in section 4.2, the larger N_θ, the larger the reduced angle-set denoted Θ, and the smaller Δr, the larger the set $R_{1,2}$. The number of configurations to optimize being the product of the cardinal of the sets Θ and $R_{1,2}$, the larger these sets, the longer the computation time. However, the more configuration to optimize, the greater the AEP. Therefore, any layout optimization needs to make a trade-off between computation time and size of the set of configuration to optimize. In this

section, we will show the effect of the parameters N_θ and Δr on the optimal AEP and the computation time, and give the user some guidelines to set these parameters. To do so, we run algorithm 1 for all possible configurations of the hyper-parameters valued in their respective value-set given in table 1, for wind farm sizes of 81, 100, 150, and 250 turbines respectively. The

$D_{\min}$	$D_{\max}$	$\Delta\theta$	N_θ	Δr
$\{2\,D_{\text{turb}}\}$	$\{6\,D_{\text{turb}}\}$	$\{1^\circ\}$	$\{1,5,10\}$	$\left\{\frac{D_{\text{turb}}}{2}, D_{\text{turb}}, 2\,D_{\text{turb}}\right\}$

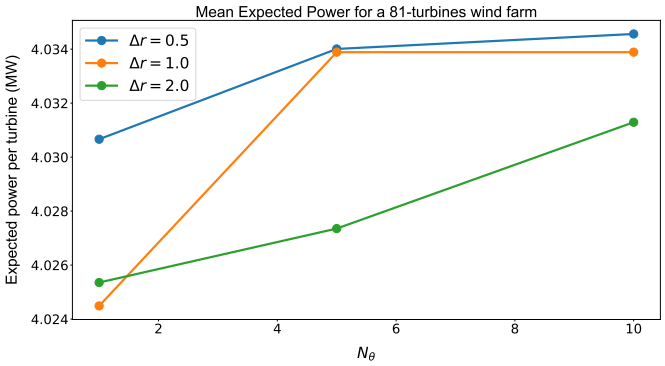
Table 1. Value set of each hyper-parameter

numerical results of these optimizations are gathered in table 2 and illustrated in figs. 3 and 4. On fig. 3, one can see that the expected power per turbine[1] is growing with respect to N_θ and decreasing with respect to Δr. Also, when $N_\theta \geq 5$ and $\Delta r \leq 1 D_{\text{turb}}$, the expected powers per turbine are similar whatever the value of these hyper parameters. However, as illustrated on fig. 4, the execution time is strongly increasing with respect to N_θ and strongly decreasing with respect to Δr. Therefore, if algorithm 1 is run using an AEP computation method more precise and computationally more expensive than the one we used, (see Baker et al. (2021); Thomas et al. (2023)), keeping N_θ reasonably small (≈ 5) and Δr reasonably large (≈ 1) should enable the solving algorithm to find a well-performing layout in a reasonable execution time. In addition, as illustrated in table 1, the solving algorithm often sets the optimal shape parameters (r_1, r_2) to their minimal authorized values. This behavior indicates that the method generates grids with many intersections and, thus, a large degree of freedom for the local search part of the algorithm. The larger the degree of freedom for the local-search algorithm, the better the solution. The best layout for each wind farm size is displayed on fig. 5 and the optimal parameters $(r_1, r_2, \theta_1, \theta_2)$, the optimal AEP, the optimal expected power per turbine, and the execution time are displayed on table 3. The execution time corresponds to the total execution time. However, Steps 1 to 3 from algorithm 1 are always computed in less than 1'30 minute (for $\Delta r = 0.5$ and $N_\theta = 10$), the majority of the execution time is spent on Step 4 of algorithm 1.

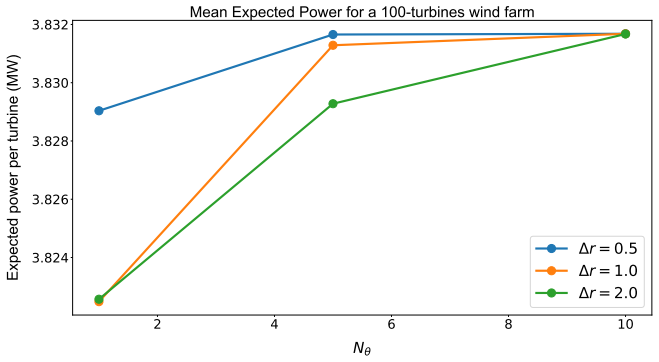
6 Exploration method's impact on the AEP

The optimization layout algorithm presented in this paper relies on the discrete exploration of the space of shape parameters $(r_1, r_2, \theta_1, \theta_2) \in [D_{\min}, D_{\max}]^2 \times [-\pi/2, \pi/2]^2$. Despite the angles' search-space reduction technique presented in section 4.2 and algorithm A2, exploring this space using fine discretization is numerically too demanding since it requires running STEP 4 of algorithm 1 for a number of shape parameters too large for a reasonable running time. Unfortunately, the performance of the optimization depends on the capacity to tune the shape parameters finely. Achieving such a fine-tuning with algorithm 1 requires using a small shape-parameter-space discretization step. Therefore, there is a strong incentive to develop heuristic methods to explore the shape parameters space other than by using the angle search space reduction associated with a brute force-like exploration method. Indeed, using more efficient shape-parameter-space exploration methods allows for a good tradeoff between a fine-tuning of these parameters and a reasonable computation time. In this section, we present optimization

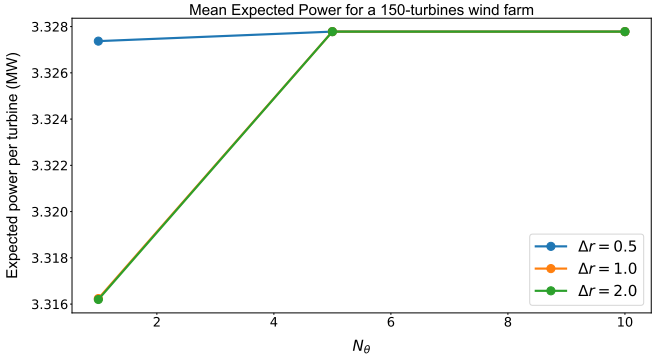
[1] For a N_t-turbines wind farm, the expected power per turbine is given by the formula $\text{AEP}(\text{MWh})/(8760 \times N_t)$ and allows for comparing wind farms of different sizes.



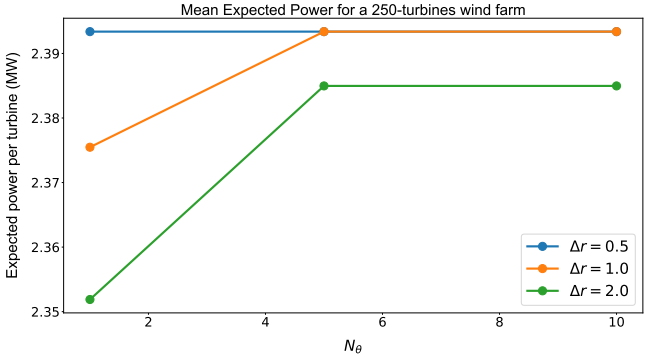
(a) Expected power per turbines for a 81-turbines wind farm as a function of N_θ and Δr



(b) Expected power per turbines for a 100-turbines wind farm as a function of N_θ and Δr



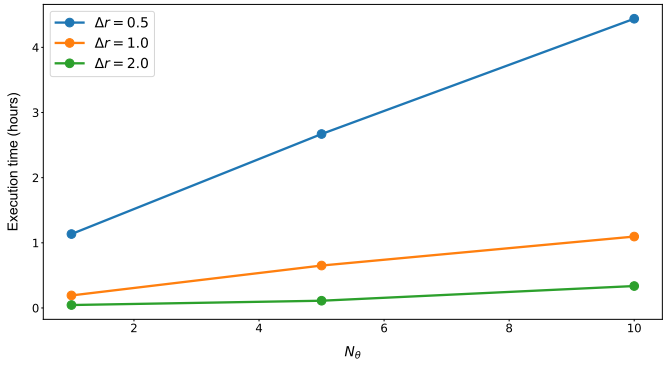
(c) Expected power per turbines for a 150-turbines wind farm as a function of N_θ and Δr



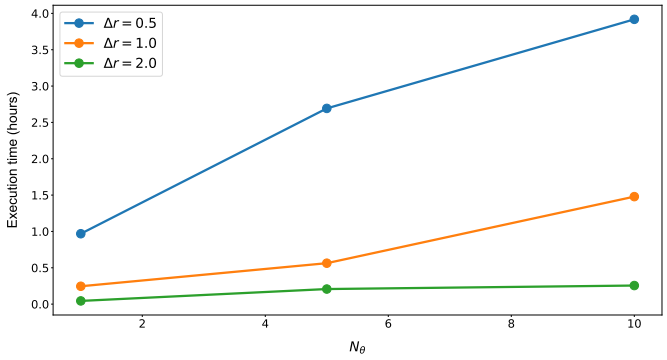
(d) Expected power per turbines for a 250-turbines wind farm as a function of N_θ and Δr

Figure 3. Expected power per turbine for 81, 100, 150, and 250 turbines wind farms as a function of the hyper-parameters N_θ, and Δr.

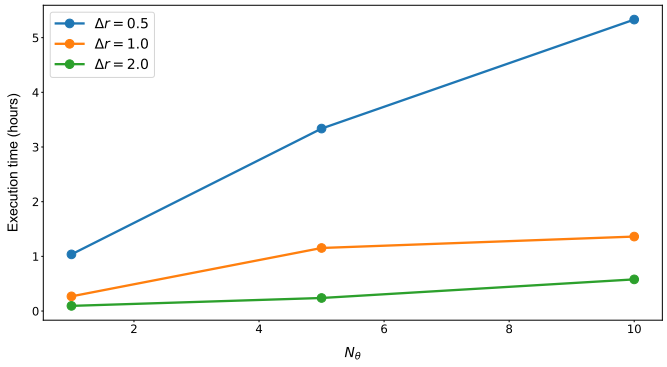
results using such a heuristic to provide a benchmark for an aligned optimization algorithm. Unfortunately, for industrial confidentiality reasons, we do not describe its principle and only focus on the improvement in terms of AEP. Again, we have run the optimization procedure for wind farms of 81, 100, 150, and 250 turbines. The results in terms of optimal shape parameters, AEP, expected power per turbine, and wake losses are summarized in table 4, and the optimal layouts are displayed in fig. 6. The optimal layout obtained using this heuristic exhibits the same behavior as those found in the previous section regarding r_1 and r_2. Indeed, these parameters are systemically found to be equal to the lowest possible value. On the contrary, the optimal angles are not the same. One of the alignment directions is conserved ($\approx 18°$), but the other one is quite different even when taking into account the $180°$ periodicity of the angles. Concerning the AEP, using a heuristic to explore the space of shape parameters more efficiently allows for improvement. Interestingly, as illustrated in fig. 7 and in the last column of table 4, the percentage of AEP increase grows almost linearly concerning the wind farm size and reaches 1% for the larger one. This behavior stems from the decreasing degrees of freedom in the turbine's optimal placing problem as the wind farm



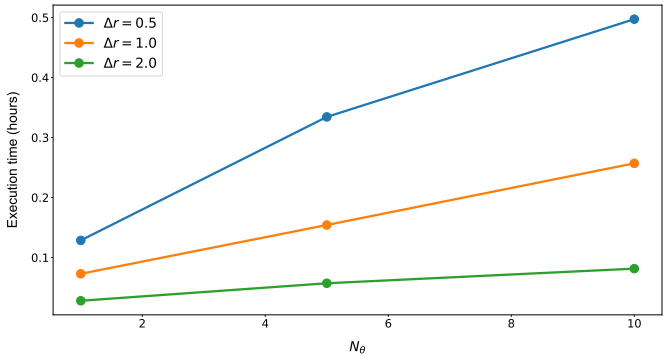
(a) Execution time of algorithm 1 as a function of the hyper-parameters N_θ, Δr for a 81-turbines wind farm



(b) Execution time of algorithm 1 as a function of the hyper-parameters N_θ, Δr for a 100-turbines wind farm



(c) Execution time of algorithm 1 as a function of the hyper-parameters N_θ, Δr for a 150-turbines wind farm



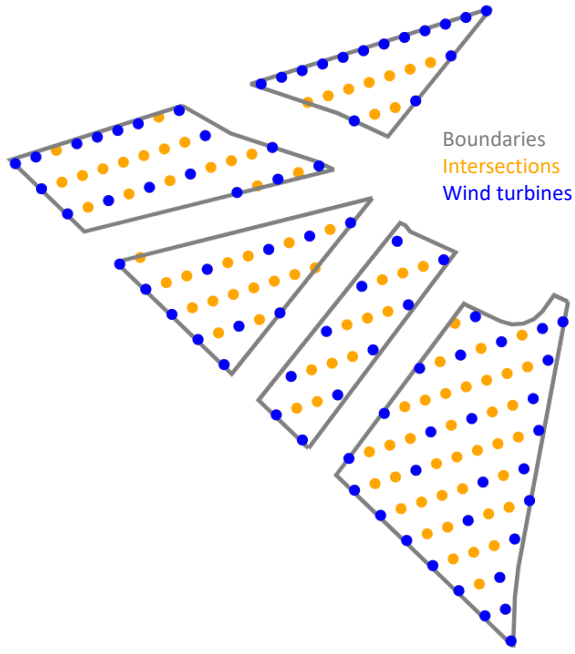
(d) Execution time of algorithm 1 as a function of the hyper-parameters N_θ, Δr for a 250-turbines wind farm

Figure 4. Execution time of algorithm 1 for 81, 100, 150, and 250 turbines wind farms as a function of the hyper-parameters N_θ, and Δr. All optimizations were run on a 12^{th} Gen Intel(R) i7-12700H 2.30 GHz core.

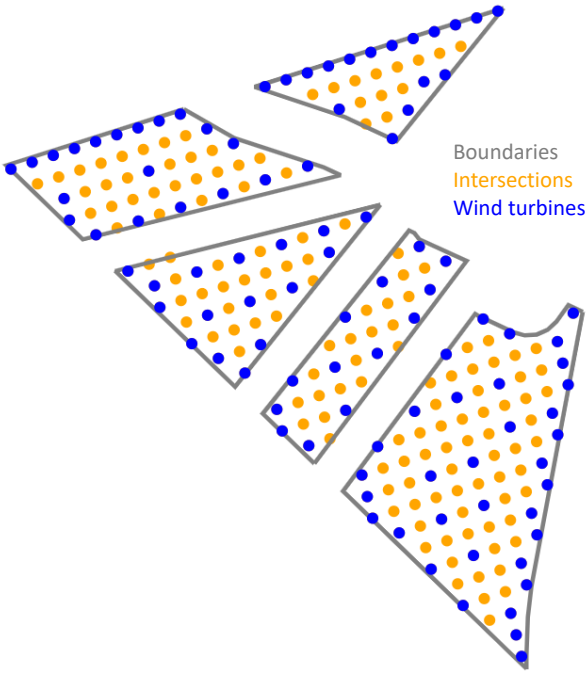
size grows. Therefore, for larger farms, the efficiency of the shape parameters optimization algorithm is of greater importance than for smaller farms; thus, there is a more substantial improvement of AEP for large farms when using a better exploration algorithm for the space of shape parameters. These results prove a strong interest in developing efficient heuristics to explore the space of shape parameters.

7 Conclusions

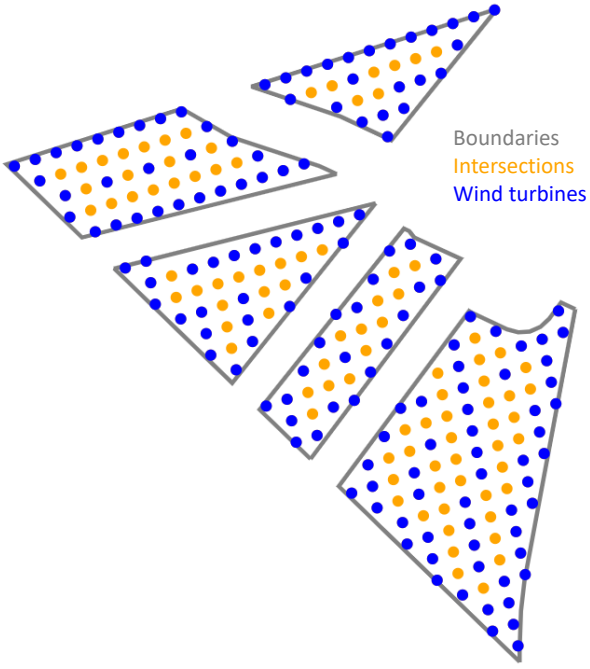
This work tackles the wind farm layout optimization problem with alignment constraints. We introduced a model of the corresponding optimization problem and adapted the DEBO algorithm from Thomas et al. (2023) to this new problem. The proposed method is based on an exploration heuristic for computing the grid parameters and a local-search method to place the turbines on the grid's intersections optimally. We have shown that this method performs well on the benchmark of IEA Wind task 37



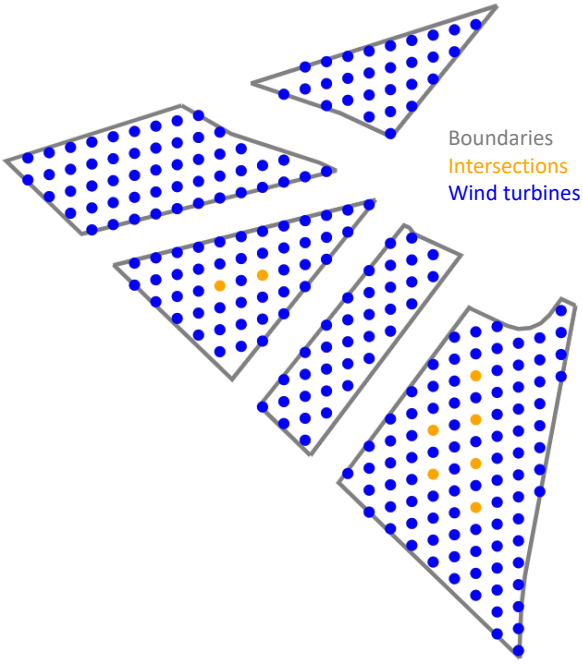
(a) Best 81-turbines wind farm layout



(b) Best 100-turbines wind farm layout

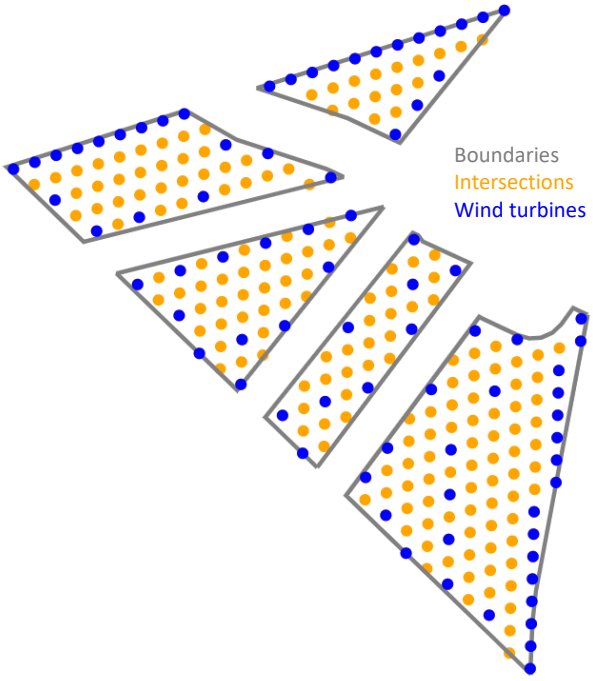


(c) Best 150-turbines wind farm layout

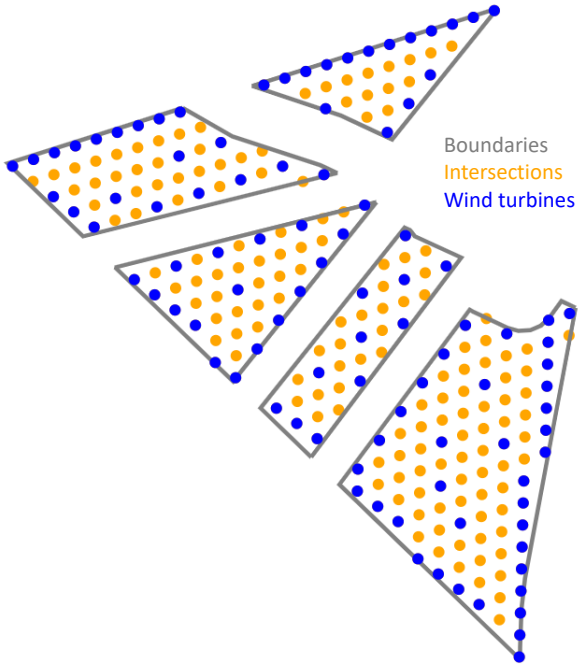


(d) Best 250-turbines wind farm layout

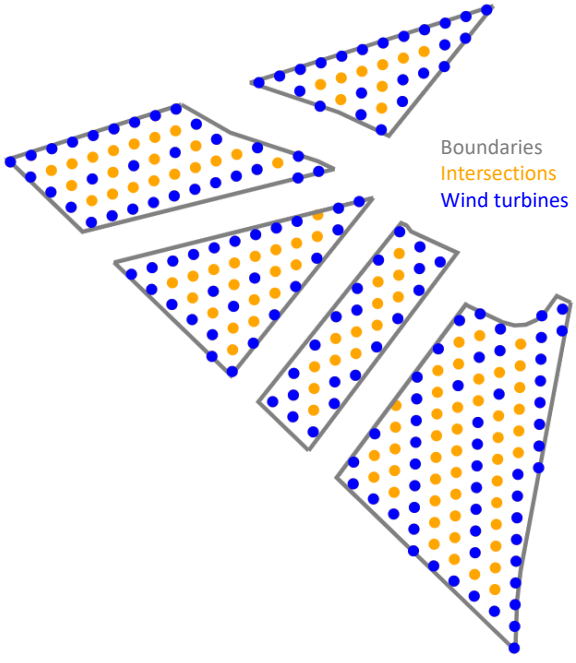
Figure 5. Best layouts for wind farm sizes of 81, 100, 150, 250 turbines using algorithm 1.



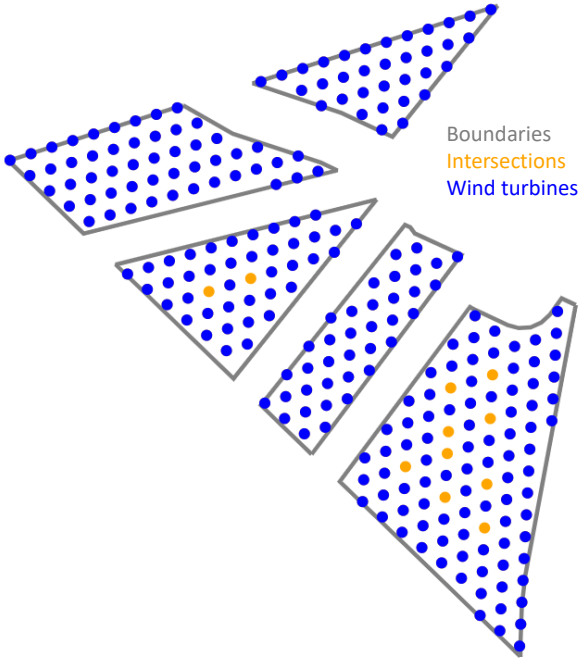
(a) Best 81-turbines wind farm layout



(b) Best 100-turbines wind farm layout



(c) Best 150-turbines wind farm layout



(d) Best 250-turbines wind farm layout

Figure 6. Best layouts for wind farm sizes of 81, 100, 150, 250 turbines using an efficient shape parameters exploration method and the same local-search algorithm.

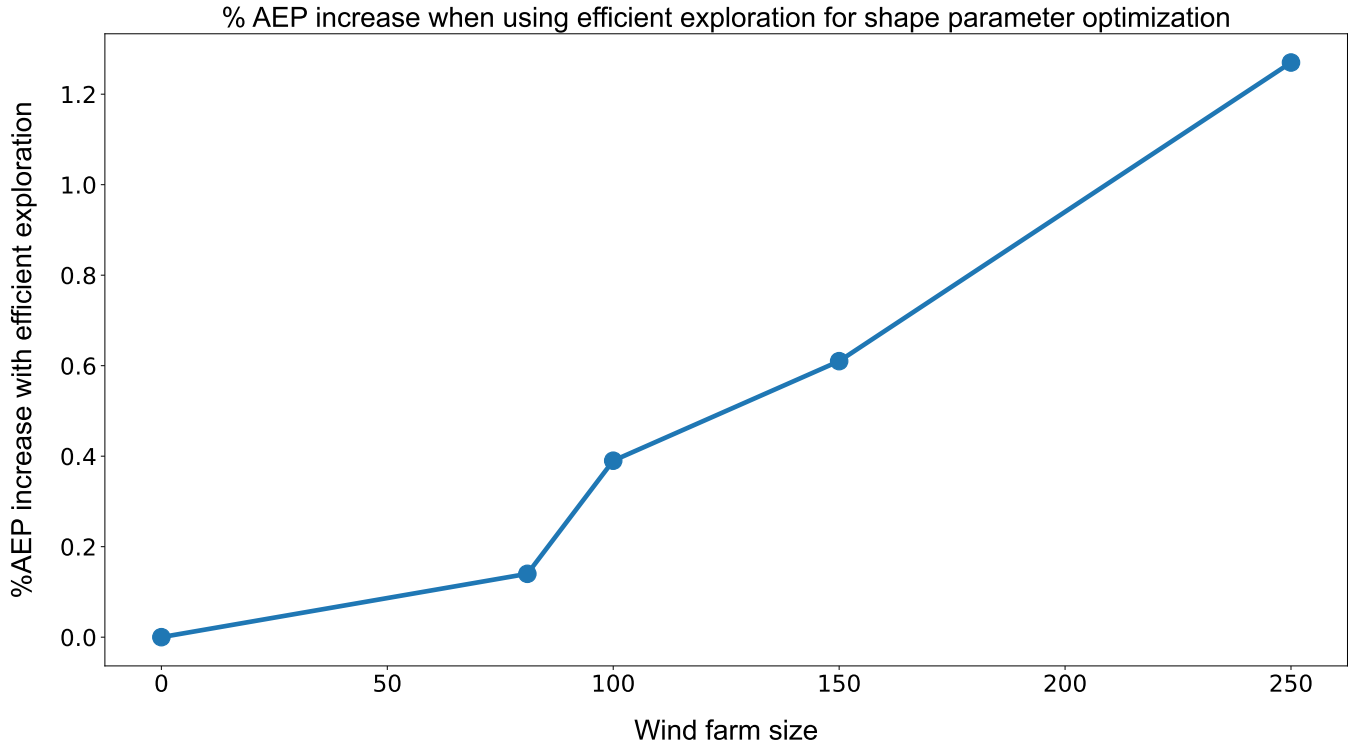


Figure 7. Percentage of AEP increase when using an efficient shape-parameter-space exploration method compared to the brute force discretization from algorithm 1.

(Thomas (2022)) since it produced an AEP increase of 0.41% compared to the baseline layout, even though the latter does not satisfy any alignment constraints and is potentially less prone to significant wake losses. Using this numerical example, we have also demonstrated the benefits of developing efficient heuristics for exploring the grid parameters. Indeed, using efficient heuristics allows for a better trade-off between wake-losses reduction and computation time. Therefore, these heuristics can be used to find layouts with higher AEP or to use more precise and computationally demanding AEP models. A more efficient algorithm can enable the introduction of other optimization parameters or constraints, such as cable routing or shared mooring for floating farms. Otherwise, to quantify the effect of alignment constraints on the wake losses, one could perform a sensitivity analysis by allowing small displacements of each turbine, resulting in an *almost* aligned layout. Finally, future works can also investigate the robustness of the optimal layout with respect to the wind probability measure $\mathbb{P}_{\mathrm{W}} := \sum_{n=1}^{N_{\mathrm{W}}} p(w_s^n, w_d^n)\delta(w_s^n, w_d^n)$. The sensitivity of the AEP with respect to the wind probability is easy to estimate if the support of the probability is unchanged, i.e., if the modified probability $\hat{\mathbb{P}}_{\mathrm{W}}$ writes $\hat{\mathbb{P}}_{\mathrm{W}} := \sum_{n=1}^{N_{\mathrm{W}}} \left[p(w_s^n, w_d^n) + \Delta p(w_s^n, w_d^n)\right] \delta(w_s^n, w_d^n)$, with $\sum \Delta p(w_s^n, w_d^n) = 0$. In this case we have

$$\Delta\mathrm{AEP}(\mathrm{F}_n) := 8760 \sum_{n=1}^{N_{\mathrm{W}}} \mathcal{P}(\mathrm{F}_n, w_s^n, w_d^n)\Delta p(w_s^n, w_d^n))$$

However, when the probability's support is also varying, computations are much more involved and future works could focus on using Distributionnaly Robust Optimization methods (see Rahimian and Mehrotra (2022) for a review on the subject) to produce robust layouts with respect to the wind probability measure.

Data availability. Provided physics model, turbines and boundary data are available at Thomas (2022), optimal layouts, AEPs, shape configurations, and intersections are available at Malisani (2024)

Wind farm size	Δr	N_θ	Optimal shape parameters $(r_1, r_2, \theta_1, \theta_2)$	AEP (GWh)	Expected power per turbine (MW)	Exec. time (s)
81	0.5	1	$(5.5\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 73°,\ -87°)$	2860.00	4.031	4083
81	0.5	5	$(2\,D_{\text{turb}},\ 3\,D_{\text{turb}},\ 18°,\ -80°)$	2862.37	4.034	9611
81	0.5	10	$(2\,D_{\text{turb}},\ 3\,D_{\text{turb}},\ 18°,\ -80°)$	2862.76	4.035	15 981
81	1	1	$(2\,D_{\text{turb}},\ 6\,D_{\text{turb}},\ 90°,\ 70°)$	2855.62	4.024	690
81	1	5	$(2\,D_{\text{turb}},\ 3\,D_{\text{turb}},\ 18°,\ -80°)$	2862.29	4.034	2343
81	1	10	$(2\,D_{\text{turb}},\ 3\,D_{\text{turb}},\ 18°,\ -80°)$	2862.29	4.034	3944
81	2	1	$(2\,D_{\text{turb}},\ 6\,D_{\text{turb}},\ 90°,\ 70°)$	2856.23	4.025	161
81	2	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	2857.65	4.027	399
81	2	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	2860.44	4.031	1211
100	0.5	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	3354.24	3.829	3486
100	0.5	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	3356.53	3.832	9693
100	0.5	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	3356.55	3.832	14105
100	1	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 19°,\ -78°)$	3348.50	3.822	884
100	1	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -82°)$	3356.21	3.831	2027
100	1	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	3356.55	3.832	5324
100	2	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 19°,\ -78°)$	3348.57	3.823	158
100	2	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	3354.45	3.829	746
100	2	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	3356.54	3.832	920
150	0.5	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.17	3.327	3729
150	0.5	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	12013
150	0.5	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	19188
150	1	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 19°,\ -78°)$	4357.54	3.316	966
150	1	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	4151
150	1	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	4902
150	2	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 19°,\ -78°)$	4357.48	3.316	339
150	2	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	858
150	2	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	4372.71	3.328	2084
250	0.5	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	5241.50	2.393	462
250	0.5	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	5241.50	2.393	1203
250	0.5	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	5241.50	2.393	1789
250	1	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 27°)$	5202.29	2.375	262
250	1	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	5241.50	2.393	555
250	1	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	5241.50	2.393	925
250	2	1	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 19°)$	5150.62	2.352	100
250	2	5	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 18°)$	5223.08	2.385	205
250	2	10	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 18°)$	5223.08	2.385	293

Table 2. Influence of the hyper-parameters N_θ and Δr on the AEP and algorithm 1 execution time. All optimizations were run on a 12^{th} Gen Intel(R) i7-12700H 2.30 GHz core.

Number of turbines	AEP (GW h)	Optimal shape parameters $(r_1, r_2, \theta_1, \theta_2)$	Expected power / turbine	% wake losses
1	42.55	–	4.86 MW	0
81	2862.76	$(2\,D_{\text{turb}},\ 3\,D_{\text{turb}},\ 18°,\ -80°)$	4.03 MW	17.1%
100	3356.55	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -76°)$	3.83 MW	21.2%
150	4372.71	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 18°,\ -78°)$	3.33 MW	31.5%
250	5241.50	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 90°,\ 14°)$	2.39 MW	50.8%

Table 3. Optimal AEP, shape configuration and mean power per turbine for an increasing number of turbines.

Number of turbines	AEP (GW h)	Optimal shape parameters $(r_1, r_2, \theta_1, \theta_2)$	Expected power per turbine	% wake losses	% AEP increase w.r.t. table 2
1	42.550	–	4.86 MW	0	0
81	2866.697	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 88.6°,\ 17.9°)$	4.04 MW	16.8%	0.14 %
100	3369.709	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 88.5°,\ 17.9°)$	3.85 MW	20.8%	0.39 %
150	4399.174	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 88.8°,\ 17.78°)$	3.35 MW	31.1%	0.61 %
250	5308.184	$(2\,D_{\text{turb}},\ 2\,D_{\text{turb}},\ 87.0°,\ 17.4°)$	2.42 MW	50.1%	1.27 %

Table 4. Optimal AEP, shape configuration, and mean power per turbine for an increasing number of turbines using a fast heuristic for the optimization of the shape's parameters.

Appendix A: Algorithms in pseudo-code

Algorithm A1 GenerateR1R2($\Delta r, D_{\min}, D_{\max}$)

$R_{1,2} \leftarrow \emptyset$
$r_1 \leftarrow D_{\min}.D_{\text{turb}}$
angles $\leftarrow \emptyset$
while $r_1 \leq D_{\max}.D_{\text{turb}}$ **do**
    $r_2 \leftarrow D_{\min}.D_{\text{turb}}$
    while $r_2 \leq D_{\max}.D_{\text{turb}}$ **do**
        $R_{1,2} \leftarrow R_{1,2} \cup \{r_1,\ r_2)\}$
        $r_2 \leftarrow r_2 + \Delta r$
    end while
    $r_1 \leftarrow r_1 + \Delta r$
end while
return $R_{1,2}$

Algorithm A2 ReduceSearchSpace($R_{1,2}, \Delta\theta, D_{\min}, D_{\max}, D_{\text{turb}}, N_\theta$)

$\Theta \leftarrow \emptyset$
for $(r_1,\ r_2) \in R_{1,2}$ **do**
    $\Theta_{r_1,r_2} \leftarrow$ GetBestAngles($r_1, r_2, \Delta\theta, D_{\min}, D_{\max}, D_{\text{turb}}, N_\theta$)
    $\Theta \leftarrow \Theta \cup \Theta_{r_1,r_2}$
end for
return Θ

Algorithm A3 GetBestAngles$(r_1, r_2, \Delta\theta, D_{\min}, D_{\max}, D_{\text{turb}}, N_\theta)$

$\text{farms} \leftarrow \emptyset$
$\Theta_{r_1,r_2} \leftarrow \emptyset$
$\theta_1 \leftarrow -\frac{\pi}{2} + \Delta\theta$
while $\theta_1 \leq \frac{\pi}{2}$ **do**
  $\theta_2 \leftarrow -\frac{\pi}{2}$
  while $\theta_2 \leq \theta_1 - \Delta\theta$ **do**
    $D_{\text{grid}} \leftarrow \min_{z \in \mathbb{Z}^2_*} \left\| P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} z \right\|$
    if $(D_{\text{grid}} \geq D_{\min}.D_{\text{turb}}) \wedge (\theta_1 - \theta_2 \leq \pi - \Delta\theta)$ **then**
      $n \leftarrow 1$
      for $i = 0$ **to** $i = 1$ **do**
        for $j = 0$ **to** $j = 1$ **do**
          $\mathbf{F}_4(n) \leftarrow P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} \begin{pmatrix} i \\ j \end{pmatrix}$
          $n \leftarrow n + 1$
        end for
      end for
      $\text{aep} \leftarrow \mathbb{E}_{\text{W}} \left[\mathcal{P}(\mathbf{F}_4, w_s, w_d) \right]$
      $\text{farms} \leftarrow \text{farms} \cup \{(\theta_1,\ \theta_2,\ \text{aep})\}$
    end if
    $\theta_2 \leftarrow \theta_2 + \Delta\theta$
  end while
  $\theta_1 \leftarrow \theta_1 + \Delta\theta$
end while
$\text{sort}(\text{farms})$ {by decreasing order of aep}
for $i = 1$ **to** $i = N_\theta$ **do**
  $\Theta_{r_1,r_2} \leftarrow \Theta_{r_1,r_2} \cup \{(\text{farms}(i).\theta_1, \text{farms}(i).\theta_2)\}$
end for
return Θ_{r_1,r_2}

Algorithm A4 GenerateConfigs($R_{1,2}, \Theta, D_{\min}, D_{\max}, D_{\text{turb}}$)

grids $\leftarrow \emptyset$
for $(r_1,\ r_2) \in R_{1,2}$ **do**
  for $(\theta_1,\ \theta_2) \in \Theta$ **do**
    $D_{\text{grid}} \leftarrow \min_{z \in \mathbb{Z}^2_*} \left\| P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} z \right\|$
    if $\min_{z \in \mathbb{Z}^2_*} \left\| P_{\mathcal{B}_0}^{\mathcal{B}(\theta_1,\theta_2,r_1,r_2)} z \right\| \geq D_{\min}.D_{\text{turb}}$ **then**
      grids $\leftarrow$ grids $\cup \{(r_1, r_2, \theta_1, \theta_2)\}$
    end if
  end for
end for
return grids

Algorithm A5 ComputeAllIntersections(grids, $D_{\min}$, Ω)

intersections_sets $\leftarrow \{\emptyset\}$
$n \leftarrow 1$
while $n \leq \text{card}(\text{grids})$ **do**
  {Compute basic shape vector}
  $r_1, r_2, \theta_1, \theta_2 \leftarrow \text{grids}(n)$
  $v_1 \leftarrow (r_1 \cos(\theta_1), r_1 \sin(\theta_1))$
  $v_2 \leftarrow (r_2 \cos(\theta_2), r_2 \sin(\theta_2))$
  {Function of the intersections}
  $I_{v_1,v_2}(\Delta_1, \Delta_2) := \{k_1 v_1 + k_2 v_2 \in \mathbb{R}^2 \text{ s.t. } [k_1 v_1 + k_2 v_2 + (\Delta_1, \Delta_2)] \in \Omega\ ;\ (k_1, k_2) \in \mathbb{Z}^2\}$
  {Compute offset maximising the number of admissible intersections}
  $(\Delta_1^*, \Delta_2^*) \leftarrow \arg\max_{\Delta_1, \Delta_2 \in [0,1)^2} \text{card}(I_{v_1,v_2}(\Delta_1, \Delta_2))$
  {Add intersections to the set of intersections}
  if $\text{card}(I_{v_1,v_2}(\Delta_1^*, \Delta_2^*)) \geq N_{\max}$ **then**
    intersections_sets $\leftarrow \{(I_{v_1,v_2}(\Delta_1^*, \Delta_2^*), (r_1, r_2, \theta_1, \theta_2))\} \bigcup$ intersections_sets
  end if
  $n \leftarrow n + 1$
end while
return intersections_sets

Algorithm A6 GreedyInitialization(intersections)

$(x_0, y_0) \leftarrow \arg\max_{x,y}\{x - y \text{ s.t. } (x,y) \in \text{ intersections}\}$
$\mathbf{F}_1(1) \leftarrow (x_0 \;\; y_0)$
$n_t \leftarrow 1$
intersections $\leftarrow$ intersections$\backslash\{(x_0 \;\; y_0)\}$
while $n_t < N_{\max}$ **do**
  $(x^* \;\; y^*)^\top \leftarrow \arg\max_{(x,y)\in \text{intersections}} \mathbb{E}_{\mathrm{W}}\{\mathcal{P}(\mathbf{F}_{n_t} \oplus (x \;\; y), w_s, w_d)\}$
  $\mathbf{F}_{n_t+1} \leftarrow \mathbf{F}_{n_t} \oplus (x^* \;\; y^*)$
  $n_t \leftarrow n_t + 1$
  intersections $\leftarrow$ intersections $\backslash\{(x^* \;\; y^*)\}$
end while
return $\mathbf{F}_{N_{\max}}$

Algorithm A7 LocalSearch($\mathbf{F}_{N_{\max}}$, intersections)

convergence $\leftarrow \perp$
aep $\leftarrow \mathbb{E}_{\mathrm{W}}\{\mathcal{P}(\mathbf{F}_{N_{\max}}, w_s, w_d)\}$
while $\neg$ convergence **do**
  $\mathbf{H}_{N_{\max}} \leftarrow \mathbf{F}_{N_{\max}}$
  random_indices $\leftarrow$ shuffle($[\![0, \ldots, N_{\max}[\![$)
  for indice $\in$ random_indices **do**
    {compute all possible layouts by moving i$^{\text{th}}$ turbine}
    children_layout $\leftarrow$ generate_children($\mathbf{F}_{N_{\max}}$, indice, intersections)
    for $(\mathbf{G}_{N_{\max}}, \text{intersections}_G) \in$children_layout **do**
      if $\mathbb{E}_{\mathrm{W}}\{\mathcal{P}(\mathbf{G}_{N_{\max}}, w_s, w_d)\} >$ aep **then**
        $\mathbf{F}_{N_{\max}} \leftarrow \mathbf{G}_{N_{\max}}$
        aep $\leftarrow \mathbb{E}_{\mathrm{W}}\{\mathcal{P}(\mathbf{F}_{N_{\max}}, w_s, w_d)\}$
        intersections$\leftarrow$ intersections$_G$
      end if
    end for
  end for
  convergence $\leftarrow \mathbf{F}_{N_{\max}} = \mathbf{H}_{N_{\max}}$
end while
return $(\mathbf{F}_{N_{\max}}, \text{aep})$

Algorithm A8 generate_children($\mathbf{F}_{N_{\max}}$, indice, intersections)

children_layout $\leftarrow \emptyset$
for $(x\ \ y) \in$ intersections **do**
  $\mathbf{G}_{N_{\max}} \leftarrow \mathbf{F}_{N_{\max}}$
  intersection$_G \leftarrow$ intersections
  $(x_{\text{new}}\ \ y_{\text{new}}) \leftarrow \mathbf{G}_{N_{\max}}$(indice)
  $\mathbf{G}_{N_{\max}}$(indice) $\leftarrow (x\ \ y)$
  intersections$_G \leftarrow$ intersections$_G \setminus \{(x\ \ y)\} \cup \{(x_{\text{new}}, y_{\text{new}})\}$
  children_layout $\leftarrow$ children_layout $\cup\{(\mathbf{G}_{N_{\max}}$, intersections$_G)\}$
end for
return children_layout

Algorithm A9 PlaceTurbines(intersections)

$\mathbf{F}_{N_{\max}} \leftarrow$ GreedyInitialization(intersections)
$(\mathbf{F}_{N_{\max}}, \text{aep}) \leftarrow$ LocalSearch($\mathbf{F}_{N_{\max}}$, intersections)
return $(\mathbf{F}_{N_{\max}}, \text{aep})$

Author contributions. P. Malisani developped the methodology and wrote the manuscript. T. Bartement was in charge of the software development. P. Bozonnet edited the full manuscript.

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

References

Baker, N. F., Thomas, J. J., Stanley, A. P. J., and Ning, A.: EA Task 37 Wind Farm Layout Optimization Case Studies, Zenodo [code and data set], Tech. rep., https://doi.org/10.5281/zenodo.5809681, 2021.

Bastankhah, M. and Porté-Agel, F.: Experimental and theoretical study of wind turbine wakes in yawed conditions, Journal of Fluid Mechanics, 806, 506–541, https://doi.org/10.1017/jfm.2016.595, 2016.

Bortolotti, P., Tarres, H. C., Dykes, K. L., Merz, K., Sethuraman, L., Verelst, D., and Zahle, F.: IEA Wind TCP Task 37: Systems Engineering in Wind Energy - WP2.1 Reference Wind Turbines, Tech. rep., https://doi.org/10.2172/1529216, 2019.

Burer, S. and Letchford, A.: Non-convex mixed-integer nonlinear programming: A survey, Surveys in Operations Research and Management Science, 17, 97–106, 2012.

Exler, O., Antelo, L., Egea, J., Alonso, A., and Banga, J.: A tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design, Computers and Chemical Engineering, 32, 1877–1891, 2008.

Fischetti, M. and Fischetti, M.: Integrated Layout and Cable Routing in Wind Farm Optimal Design, Management Science, 69, 2147–2164, 2022.

Fischetti, M. and Pisinger, D.: Mathematical Optimization and Algorithms for Offshore Wind Farm Design: An Overview, Business & Information Systems Engineering, 61, 469–485, 2019.

FLORIS: NREL 2023, Version 3.4.1., https://github.com/NREL/floris.

Herbert-Acero, J., Probst, O., Rethore, P., Larsen, G., and Castillo-Villar, K.: A Review of Methodological Approaches for the Design and Optimization of Wind Farms, Energies, 7, 2014.

Hou, P., Zhu, J., Ma, K., Yang, G., Hu, W., and Chen, Z.: A review of offshore wind farm layout optimization and electrical system design methods, J. Mod. Power Syst. Clean Energy, 7, 975–986, 2019.

Kumar, M. and Sharma, A.: Wind Farm Layout Optimization Problem Using Teaching-Learning-Based Optimization Algorithm, in: Communication and Intelligent Systems, edited by Sharma, H., Shrivastava, V., Bharti, K. K., and Wang, L., pp. 151–170, Springer Nature Singapore, Singapore, 2023.

Kunakote, T., Sabangban, N., Kumar, S., Tejani, G. G., Panagant, N., Pholdee, N., Bureerat, S., and Yildiz, A. R.: Comparative Performance of Twelve Metaheuristics for Wind Farm Layout Optimisation, Archives of Computational Methods in Engineering, 29, 717–730, https://doi.org/10.1007/s11831-021-09586-7, 2022.

Liang, Z. and Liu, H.: Layout Optimization Algorithms for the Offshore Wind Farm with Different Densities Using a Full-Field Wake Model, Energies, 16, https://doi.org/10.3390/en16165916, 2023.

Liberti, L., Mladenović, N., and Nannicini, G.: A recipe for finding good solutions to MINLPs, Mathematical Programming Computation, 3, 349–390, 2011.

LoCascio, M. J., Bay, C. J., Martinez-Tossas, L. A., Bastankhah, M., and Gorlé, C.: FLOWERS AEP: An Analytical Model for Wind Farm Layout Optimization, Wind Energy, 27, 1563–1580, https://doi.org/https://doi.org/10.1002/we.2954, 2024.

Malisani, P.: Data for Wind farm layout optimization with alignment constraints: Initial release Zenodo [data set], Tech. rep., https://doi.org/10.5281/zenodo.13122308, 2024.

Maritime and Coastguard Agency: Offshore renewable energy installations: impact on shipping, https://www.gov.uk/guidance/offshore-renewable-energy-installations-impact-on-shipping, 2012.

Ministry of Transport of the French Republic: Note technique du 28 juillet 2017 établissant les principes permettant d'assurer l'organisation des usages maritimes et leur sécurité dans et aux abords immédiats d'un champ éolien en mer, https://eolmernormandie.debatpublic.fr/images/documents/dmo/biblio/cir_42526.pdf, 2017.

Papadimitriou, C. and Steiglitz, K.: Combinatorial optimization, algorithms and complexity, Dover, 1998.

Porté-Agel, F., Bastankhah, M., and Shamsoddin, S.: Wind-Turbine and Wind-Farm Flows: A Review, Boundary-Layer Meteorolgy, 174, 1–59, 2020.

Quick, J., Réthoré, P.-E., Mølgaard Pedersen, M., Rodrigues, R. V., and Friis-Møller, M.: Stochastic gradient descent for wind farm optimization, Wind Energy Science, 8, 1235–1250, https://doi.org/10.5194/wes-8-1235-2023, 2023.

Rahimian, H. and Mehrotra, S.: Frameworks and Results in Distributionally Robust Optimization, Open Journal of Mathematical Optimization, 3, 4, 2022.

Schlüter, M., Egea, J. A., and Banga, J. R.: Extended ant colony optimization for non-convex mixed integer nonlinear programming, Computers & Operations Research, 36, 2217–2229, 2009.

Stanley, A. P. J. and Ning, A.: Massive simplification of the wind farm layout optimization problem, Wind Energy Science, 4, 663–676, https://doi.org/10.5194/wes-4-663-2019, 2019.

Thomas, J. J.: jaredthomas68/thomas2022-8-opt-algs-wflop: Initial release Zenodo [code and data set], Tech. rep., https://doi.org/10.5281/zenodo.7125349, 2022.

Thomas, J. J., Baker, N. F., Malisani, P., Quaeghebeur, E., Sanchez Perez-Moreno, S., Jasa, J., Bay, C., Tilli, F., Bieniek, D., Robinson, N., Stanley, A. P. J., Holt, W., and Ning, A.: A comparison of eight optimization methods applied to a wind farm layout optimization problem, Wind Energy Science, 8, 865–891, https://doi.org/10.5194/wes-8-865-2023, 2023.

Yiqing, L., Xigang, Y., and Yongjian, L.: An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints, Computers & chemical engineering, 31, 153–162, 2007.

Young, C.-T., Zheng, Y., Yeh, C.-W., and Jang, S.-S.: Information-guided genetic algorithm approach to the solution of MINLP problems, Industrial & engineering chemistry research, 46, 1527–1537, 2007.