

Diese Arbeit wurde vorgelegt am
Lehr- und Forschungsgebiet Theorie der hybriden Systeme

Simulation und Optimierung von Offshore Windparks **Simulation and Layout Optimization of Offshore Wind** **Farms**

Masterarbeit
Software Systems Engineering

February 2019

Vorgelegt von Presented by	Marwa Maghnie Campus-Boulevard 70 52074 Aachen Matrikelnummer: 375065 marwa.maghnie@rwth-aachen.de
Erstprüfer First examiner	Univ.-Prof. Dr. rer. nat. Erika Ábrahám Lehr- und Forschungsgebiet: Theorie der hybriden Systeme RWTH Aachen University
Zweitprüfer Second examiner	Univ.-Prof. Dr. rer. nat. Thomas Noll Software Modeling and Verification Group (MOVES) RWTH Aachen University
Externer Betreuer External supervisor	Dr. rer. nat. Pascal Richter Steinbuch Centre for Computing Karlsruhe Institute of Technology

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen und Abbildungen. Diese Arbeit hat in dieser oder einer ähnlichen Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Aachen, im Februar 2019

Marwa Maghnie

Dedicated to my grandfather
Who left us suddenly and too soon

Acknowledgements

I would like to thank my thesis advisor Pascal Richter for his thorough guidance and generous time. As well Professor Erika Ábrahám for providing the opportunity to pursue this topic that combines computer science and renewable energy, and also for her feedback that opens new perspectives.

Many thanks to Joschka Theissen, Linus Franke, Julian Düstersiek, Marten Schulz, Merlin Bögershausen, and Florian Hövelmann for their time and helpful discussions.

A thank you as well to Professor Thomas Noll for agreeing to correct this thesis. In addition thanks to Frau Ohlenforst for her time and helpfulness in facilitating many administrative tasks related to this thesis.

*I can't change the direction of the wind,
but I can adjust my sails to always reach my destination.*

JIMMY DEAN

Contents

Nomenclature	VIII
List of Figures	IX
List of Tables	XI
1. Introduction	1
1.1. Thesis Outline	2
1.2. Related Work	2
1.2.1. Modeling wakes in wind farms	2
1.2.2. Optimization of wind farm layouts	3
1.2.3. Contribution to literature	4
2. Wind Farm Model	5
2.1. Wind Model	5
2.1.1. Weibull Based Model	6
2.1.2. Representation of the Wind Model	6
2.2. Wake Model	8
2.2.1. Turbine Settings	8
2.2.2. PARK Wake Model	9
2.3. Power Generation Model	12
2.4. Cost Model	14
2.4.1. Net Annual Energy Production	14
2.4.2. Levelized Cost of Electricity	14
2.4.3. Net Present Value	14
2.4.4. Internal Rate of Return	15
2.4.5. Cost model conclusion	15
2.5. Validation of the models	15
2.5.1. Validation settings	15
2.5.2. Basics: Power, speed, and direction models	17
2.5.3. Wake Model	20
2.5.4. Wake and wind speed model	25
2.5.5. Wake and wind direction models	27
2.6. Acceleration of the Model	28
2.6.1. Number of wind speeds and wind directions	29
2.7. Model conclusion	31
3. Optimization	31
3.1. Square Grid	32
3.1.1. Square Grid Algorithm	32
3.1.2. Square Grid Parameter Studies	35
3.2. Hexagon Grid	37
3.2.1. Hexagon Grid Parameter Study	38

3.3. Close Packing	39
3.3.1. Close Packing Implementation Details	40
3.3.2. Verification of Close Packing based on Circle Packing	42
3.3.3. Close Packing parameter studies	44
3.4. Spiral Grid	46
3.4.1. Spiral Grid Parameter Study	47
3.5. Refinements with Local Search	48
3.5.1. SA Parameter Study	49
3.6. Optimizer conclusion	50
4. Feasibility Study	51
4.1. Efficiencies after optimization with patterns and local search	51
4.2. Improvements from local search	51
5. Conclusion	53
5.1. Outlook	53
A. Appendix	56
A.1. Validation supplementary figures	56
A.2. Close Packing supplementary figures	56
References	60

Nomenclature

Symbols

β_k	Shadowing factor	
C_t	Thrust coefficient	
D	Diameter of the turbine rotor	[m]
δu	Velocity deficit	
κ	Von Kármán constant	
k	Wake decay constant	
ρ	Air density	[kg/m ³]
u	Velocity in downstream direction	[m/s]
u_0	Free stream velocity	[m/s]
x	Downstream distance	[m]
z	Hub height of turbine	[m]
z_0	Surface roughness of the site	[m]
z_m	Wind measurement height	[m]

List of Figures

1.	Wake effects visualized by high humidity	1
2.	Offshore Wind Farm Model	5
3.	Distributions of wind direction measurements from January 2010 to December 2017	6
4.	Wind speeds fitted with the Weibull distribution	7
5.	HAWT wind turbines	8
6.	Wake effects as described in PARK	9
7.	Partial wake effect on turbines	12
8.	Performance of the Vestas V80 wind turbine	12
9.	Power levels of the turbine type Vestas V80	16
10.	Simulated power production by a Vestas V80 turbine.	17
11.	Wake test 1: Partial shading	21
12.	Wake test 2: Fully shading	22
13.	Wake decay test between two turbines	24
14.	Wake test 3: Partial intersecting shading	25
15.	Wake test 4: Full intersecting shading	25
16.	Wind distribution for tested wind farm layouts	26
17.	Layouts of real wind farms in validation tests	26
18.	Validation test results for real wind farms	27
19.	Wake and wind speed models test	27
20.	Layout of four turbines	28
21.	Wake and wind direction models test	28
22.	Effect of simulated wind speeds and directions on AEP (1)	29
23.	Effect of simulated wind speeds and directions on AEP (2)	30
24.	Effect of simulated wind speeds and directions on AEP using Horns Rev 1 settings	31
25.	SquareGrid example result and main parameters	32
26.	Visualization of the growth factor in the SquareGrid optimization	33
27.	Square Grid main optimization algorithm flow. After the initialization with the required data, the optimization loops over the parallelogram parameters to try various combinations of positions.	34
28.	Effect of simulated wind speeds and directions on optimization score	35
29.	Effect of growth factor on layout score	36
30.	Effect of growth factor on layout score (detailed)	36
31.	Hexagon Grid example result and main parameters	37
32.	Effect of number of angle steps on layout score	38
33.	Close-Packing model reference positions and their interpolation	40
34.	Iteration approach of Close Packing	41
35.	Dense circle packing	42
36.	Artificially interpolated matrix of efficiencies	43
37.	Result of verifying Close Packing	43
38.	Effect of number of threshold steps on layout score	45

39.	Effect of number of threshold bounds on layout score	45
40.	Placement of positions with a low score threshold	46
41.	Example layout of Spiral Grid	47
42.	Effect of varying the density factor in Spiral Grid	48
43.	Efficiencies of different SA settings	50
44.	Intermediate efficiencies of one SA run	50
45.	Intermediate efficiencies of SA run (comparison)	51
46.	Layouts of a real wind farm	52
47.	Improvement of efficiencies with local search for the Horns Rev 1 wind farm.	53
48.	Effect of simulated wind speeds and directions on AEP (supplementary)	57
49.	Close Packing pre-computation flow	58
50.	Close Packing optimization flow	59

List of Tables

1.	Wind frequency table example	7
2.	Validation settings	18
3.	Wind direction interpolation tests	19
4.	Wind direction interpolation test results	19
5.	Wind speed interpolation tests	20
6.	Wind speed interpolation test results	21
7.	Wake test 1: Partial shading results	22
8.	Wake test 2: Full shading results	22
9.	Wake test 2: Partial intersecting results	23
10.	Wake test 2: Full intersecting shading results	25
11.	Hexagon Grid parameter study results	39
12.	SA default parameters	49
13.	Efficiencies of Horns Rev 1 after optimization	52

1. Introduction

Renewable energy is a concept dating back at least to the middle ages when wind mills were first used. It has been steadily regaining its popularity in recent decades, mainly because it is becoming a necessity due to environmental constraints such as the diminishing amount of fossil fuels and more importantly the detrimental effect of these and other non-renewable resources on the Earth [15]. There are many forms of sustainable energy sources, including wind power, which can be a viable option in often-windy areas such as Europe [43].

However wind energy is not free. In addition to costs of materials and building the actual wind turbines, increasing the area of a wind farm can increase cabling and maintenance costs [27].

Therefore it is imperative to build wind farms in such a way to minimize costs and maximize favorable metrics such as the *annual energy production* (AEP) of a wind farm energy. A main component of this challenge is the placement of wind turbines, because of their *wake effects*. Wakes are the turbulence in air flow caused by a turbine which decreases the speed of the air hitting the turbines in its down-stream, thus decreasing their power production. This effect is pictured in Figure 1 where condensed air shows that most of the turbines are affected by wakes. The need for a better placement of the turbines is obvious. However, it is not so simple as there are many variables, including the wind direction. In this thesis we attempt to accurately simulate the wind scenario in offshore wind farms to obtain their energy output and optimize their layouts. The endeavor of finding the optimal placement of turbines is commonly referred to as the *wind farm layout optimization problem* (WFLOP).



Figure 1: Turbulent airflow from wakes is exposed due to extreme humidity in the air.
Picture by Christian Steiness (12.02.2008) [13]

This leads to the main question to be answered in this thesis: given some constraints about resources (e.g. area and turbine type/count) and deliverables (e.g. AEP), what could be the optimal parameters to be used in the simulation and optimization of wind farm layouts?

1.1. Thesis Outline

In this thesis we first give an overview of the literature surrounding the WFLOP. Then we introduce the theoretical background that the model used in this thesis is based upon. Followed by the validation of each of the sub-models: turbine power, wind speed, wind direction, and wake effect. Next, the description of the optimization algorithms developed in this thesis which are: *slanted grid*, *hexagon grid*, and *close-packing*. These are optimized pattern algorithms that aim to balance between high quality solutions and run-time. In the penultimate chapter we present the feasibility studies that apply the model and optimization algorithms to generate turbine layouts for three offshore wind farms in the North Sea and simulate their AEP. We compare the energy production from these wind farms before and after optimizing their layouts using the methods described earlier. Finally in the concluding chapter, we summarize the main findings in this thesis and briefly explore options for future work.

1.2. Related Work

In this section we discuss classical and recent work in approaching the WFLOP. A comprehensive review of literature has been published in the last few years [23] that covers most models and optimization algorithms normally used in the WFLOP. Here we focus on wake models and optimization algorithms as the main components of the problem.

1.2.1. Modeling wakes in wind farms

To optimize the layout of a wind farm, a suitable wake model has to be chosen. Many models have been proposed to model wind farms, including most notably the *Jensen model* [25, 26] (also known as the *PARK* model, with its variant known as *Modified PARK*), the *eddy viscosity model* [1], and the *deep-array wake model* [17].

The Jensen model disregards wind turbulence, thus is suitable for calculating the velocity deficit at a relatively far distance (3-4 rotor diameters) behind a wind turbine. However the eddy viscosity model can calculate the velocity deficit starting from a closer distance of $2D$ [7]. The deep-array wake model also has an advantage on the PARK model, by improving on how the wake is calculated for a turbine that has several upstream turbines with multiple intersecting wakes [9].

Parada et al. [35] and Gao et al. [18] presented Gaussian-based models that performed better than the Jensen model in simple test scenarios. In test cases where the wind and direction were constant, the Gaussian-based model by Parada et al. lead to lower run-times of the optimization and higher energy production. However, test cases with more complex wind scenarios did not show significant improvement from optimizations based on the Jensen model. Gao et al. used a hybrid Jensen-Gaussian based model, that resulted in lower energy production values than those calculated with the original Jensen model. This decrease in yield was reported to correspond with results from literature. However, the results are inconclusive, because the tests

were performed only under constant wind speed scenarios.

Furthermore, the Jensen model provides a balance between efficiency and accuracy when compared with other far wake models, as shown by Shakoor et al. in their 2016 review [40]. It also remains the most popular wake model used in WFLOP research [23]. The Jensen model is described in more detail in Section 2.2.

1.2.2. Optimization of wind farm layouts

Research in the optimization of wind farm layouts began sporadically in the late eighties, and it was after the work of Mosetti et al. in 1994 [31] that more attention was focused on this field [23]. Mosetti et al. discretized the wind farm area into squared cells where each cell center is a candidate turbine position, then used a nature-inspired approach to solve the WFLOP called a *genetic algorithm*, which is a population-based metaheuristic that mimics the phenomena of natural selection to produce better results over a number of iterations.

Also like GAs there are other evolutionary algorithms (EAs) including *particle swarm optimizations* (PSOs) [48] and *biogeography based optimizations* (BBOs) [3, 4] which are inspired by the migration of species over time [41]. In the WFLOP context, GAs are the most commonly used approaches [45]. However they are very sensitive to the initial solutions (population) used. Other multi-step non-GA optimizations also share this vulnerability to initial solutions.

Initial solutions A standard initial turbine layout for optimizations is a simple grid pattern [31, 20, 49] or random positions [44]. Wang et al. used a GA optimization [50], and they argued that while a grid layout is a straightforward way to apply the constraint that turbines should be at least X meters away from each other, this only applies to positions that are horizontally or vertically adjacent. Positions that are diagonally adjacent are in fact farther away by a factor of $\sqrt{2}$, which leads to under-utilized space in the wind farm. Sparse layouts can lead to higher operational costs [2]. Therefore Wang et al. used a dense interleaving pattern of positions as their initial layout, similar to a mesh of packed circles with candidate turbine positions in the circle centers. The centers of the circles were connected by a mesh of equilateral triangles. They also introduced a rotation factor in their layout to align with the dominant wind direction and reduce wake effects. Using this mesh of packed triangles, their simulation results performed better than other GAs which were seeded with simple grid layouts.

Circle packing was also used by Fagerfjäll in his thesis to enforce the minimum distance constraint among turbines [16]. Others like Ozturk and Norman used circle packing as well for their greedy heuristic approach to the WFLOP [34]. They seeded their algorithm with three initial layout types: random positions, random positions followed by local search perturbations to improve the layout, and grid-like circle packing. Their preliminary tests showed that using the circle packing layouts lead to the best final results of their algorithm. However, this could greatly depend on the nature of their greedy algorithm and may not be generalized.

Optimized patterns While packed circles have been shown to be an efficient pattern, literature search did not lead to significant results where circle packing was optimized. However patterns in general can be preferred due to their uniformity which can simplify wind farm construction and maintenance including cabling. Neubert et al. compared the usage of a structured layout and a stochastic layout in their optimization [32]. Their structured layout was in a grid form that can be rotated and scaled. While the stochastic initial solution resulted in layouts more efficient by 1%, the structured layouts were implied to be computationally much less expensive, which may be an acceptable compromise. Vanaret et al. [47] also used an optimized grid layout as an initial step, where they find the optimal slanting degree and distance between the lines of the grid. This results in a 3% increase in energy output for simple cases and also minimizing processing time by an order of magnitude over other approaches in literature. This makes it a suitable approach for a multi-step optimizer.

Post-processing step Local search algorithms can be used on existing layouts to improve the positions individually. For example the *simulated annealing* (SA) algorithm, which is a type of global optimization that behaves similarly to cooling hot material: Bilbao and Alba [8] used this meta-heuristic where a random solution is generated in the beginning for the turbine positions, then each position is perturbed and if the new location has a better score then it is kept, otherwise it may be kept with a probability depending on a variable that represents the current temperature of the cooling material. This leads to a decreasing possibility of accepting worse solutions as the optimization progresses and the temperature cools. Rivas et al. used SA and they concluded that it is better for turbines to be located at the perimeters of a wind farm even if it means they have a higher proximity to each other [38].

1.2.3. Contribution to literature

The literature search has indicated that the Jensen model is a reasonably reliable option for modeling wake effects, and as such it is used in this thesis. It was further shown that a multi-step optimization process can yield satisfactory results, which is why in this thesis the wind farm layout is designed and improved by a multi-step optimizer. The optimizer first produces multiple initial candidate layouts utilizing a number of patterns: *slanted grid*, which is a variation of grid layouts but with many more degrees of freedom than what was found in literature; *hexagon grid*, which is a form of abstraction from slanted grid that minimizes the degrees of freedom thus the computation time; and *close-packing*, which is a circle packing pattern search optimization that iteratively adds positions as close as possible to each other yet far enough to minimize wake effects. Then the turbine positions are further enhanced by local-search-based heuristics.

Our search in related work has shown that: Using this multi-step optimization approach with the mentioned patterns and local search for solving the WFLOP, does not seem to be represented in literature. This indicates possible room for exploring new solutions for the WFLOP.

2. Wind Farm Model

The model (originally implemented by Heiming [22]) is comprised of four sub-models, outlined in Figure 2. We simulate wind speed and direction, wake effects amongst the turbines, their generated power, and selected economical statistics. The model’s main output is the annual energy production (AEP) value for a wind farm, which is obtained specifically through the wind, wake and power generation submodels. The AEP value can then be used in the cost submodel to generate the economical statistics.

The wind model is described in Section 2.1. Followed by the description of the wake model in Section 2.2. In Section 2.3 the power generation model is introduced. The cost model with its economic indicators are presented in Section 2.4. The validation of the core models (wind, wake, and power) is reported in Section 2.5, followed by a parameter study of the simulated wind speeds and directions in Section 2.6. The summary of the model and recommended default model settings are outlined in Section 2.7.

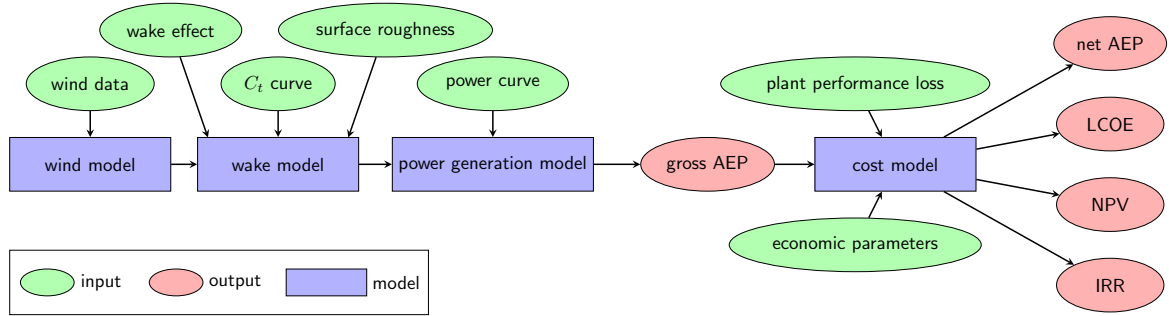


Figure 2: Structure of the offshore wind farm model. The three power-specific submodels (wind, wake and power generation) produce the annual energy production value (AEP). The cost model receives the AEP and produces economical indicators including the levelized cost of electricity (LCOE), the net present value (NPV), and the internal rate of return (IRR). Source: Richter et. al [37]

2.1. Wind Model

The wind model is based on thousands of measurements of wind speeds and their directions. This data was collected over a period of seven years from the research platform FINO3¹ in the North Sea, approximately 80 km west of the German island Sylt. The offshore wind farms DanTysk and Sandbank are in the vicinity of FINO3, and their data was used for a case study in Section 4. Figure 3 shows the wind direction distribution collected from this area. The probability of each wind direction is shown by the size of its sector.

¹<https://www.fino3.de/>

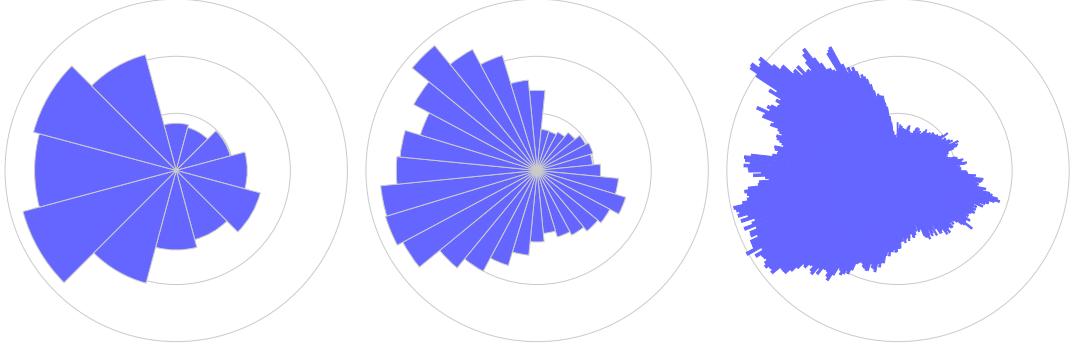


Figure 3: Distributions of wind direction measurements collected from January 2010 to December 2017 clustered into 12, 32, and 360 wind direction sectors respectively. The wind direction $\alpha = 220$ has one of the highest probabilities. Source: Richter et al. [37]

2.1.1. Weibull Based Model

Wind speed can be modeled with a Weibull distribution [30]. The probability density function of the Weibull distribution is given by:

$$u(\alpha) = \frac{\kappa}{\lambda} \left(\frac{u}{\lambda} \right)^{\kappa-1} e^{-(u/\lambda)^\kappa} \quad (1)$$

where $\kappa > 0$ is its shape parameter, and $\lambda > 0$ is its scale parameter. To fit the measured wind data with the Weibull distribution, its parameters were estimated by the maximum likelihood method (Heiming [22]), leading to these equations for the parameters:

$$\lambda = \left(\frac{1}{N} \sum_{i=0}^{N-1} u_i^\kappa \right)^{1/\kappa} \quad (2)$$

$$\frac{1}{\kappa} + \frac{1}{N} \sum_{i=0}^{N-1} \log(u_i) - \frac{\sum_{i=0}^{N-1} u_i^\kappa \log(u_i)}{\sum_{i=0}^{N-1} u_i^\kappa} = 0 \quad (3)$$

where N is the number of speed data. Using these parameters, it was possible to fit the wind speed data with the Weibull distribution, as seen in Figure 4. A good wind model is one of the crucial parts of modeling wind farms to estimate their energy output.

2.1.2. Representation of the Wind Model

Richter et al. [37] used the classical approach of modeling wind by computing a Weibull distribution for each direction, based on raw wind measurements. These distributions are then used iteratively where each wind speed's probability is weighted by its direction's Weibull distribution. Because a Weibull distribution is used to represent each wind speed probability, this can cause inaccuracies as seen in Figure 4.

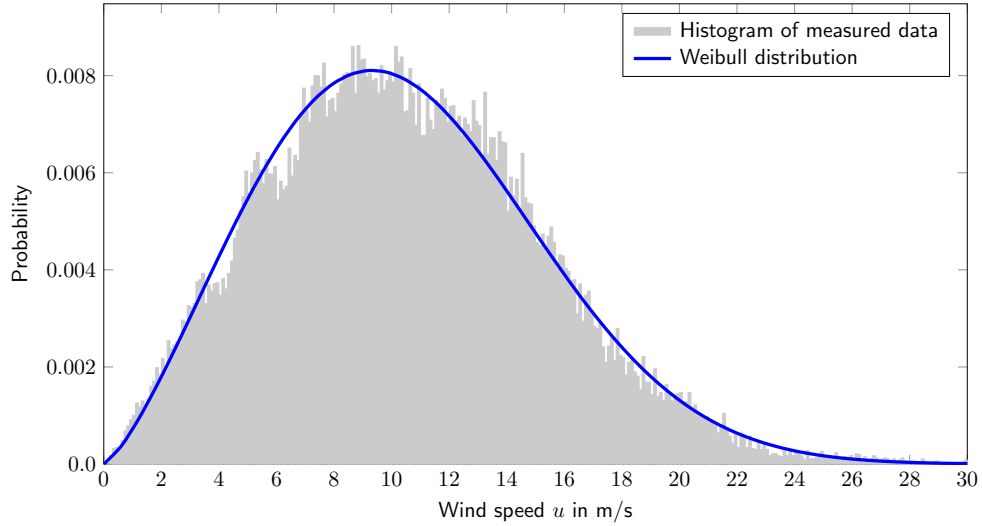


Figure 4: Fitted wind speed measurements at the FINO3 platform with the Weibull distribution. The distribution parameters λ and κ were estimated with the maximum likelihood method. This plot is for a single wind direction, sector $\alpha \in [225^\circ, 255^\circ)$. Source: Richter et al. [37]

The measured probabilities show that speeds in the range of 8 to 14 m/s are most likely to occur, but the Weibull curve underestimates this behavior.

Another way to work with the data is to use the wind speed and direction measured probabilities directly, which can be in the form of a *frequency table*, as shown in Table 1. A table contains the probabilities of all speeds for all directions, and each cell contains the probability of a specific wind speed range for a particular wind direction. Other versions of frequency tables can also include the raw measurement counts as occurrences per one-thousand [30]. The downside of using wind measurements directly could be the amount of data required to store the information about directions and speeds. The classical Weibull distributions can be a light-weight option for simulating wind scenarios, and they are used for the majority of investigations in this thesis.

Speed bin [m/s]	Direction 0°	Direction 30°	...	Direction 330°
0-1	0.03	0.42	...	0.08
1-2	0.001	0.07	...	0.2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
24-25	0.09	0.1	...	0.25
Speeds probability sum	1	1	...	1
Directions probability sum	0.05	0.13	...	1

Table 1: This frequency table shows the probabilities of wind speeds measured in $[1, 25]$ m/s , for 12 wind directions. Each direction has an individual probability for each speed bin; e.g. bin 1-2 m/s has probability 0.001 of occurring in direction 0° *among the other bins in this direction*.

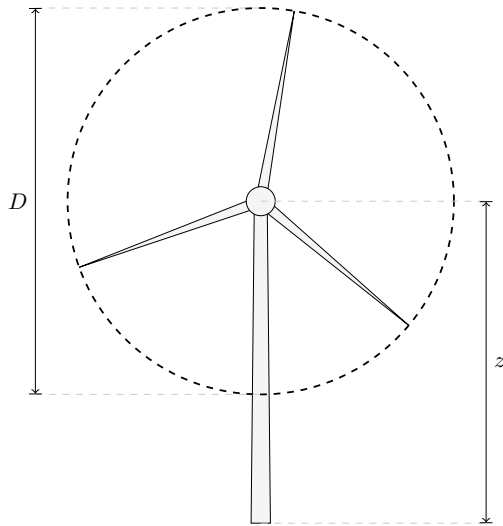
2.2. Wake Model

As briefly described in the thesis introduction, air that passes through a turbine becomes increasingly turbulent, creating a wake behind this turbine. The power output from turbines largely depends on the speed of the incoming air stream. That is why representing wakes is an important part of wind farm modeling.

Section 2.2.1 summarizes relevant wind turbine properties, followed by Section 2.2.2 which describes the wake model used in this thesis.

2.2.1. Turbine Settings

Figure 5 shows a horizontal-axis wind turbine (HAWT), which is the most common type of turbine for wind farms. Each type of turbine has a cut-in speed u_{cutin} and a cut-out speed u_{cutout} to constrain its operation. u_{cutin} is the minimum air speed required for a wind turbine to work and produce consistent power. Conversely u_{cutout} is the maximum air speed in which a turbine may function, as air velocities higher may damage the turbine. Each turbine also has a thrust coefficient curve, which is a function of the incident wind speed, and affects the wake calculation. The wake calculation is presented next in Section 2.2.2, while the power generation specifics are detailed in Section 2.3.



(a) The structure of a horizontal-axis wind turbine is mainly defined by its rotor diameter D and hub height z .
Source: Heiming [22]



(b) Wind turbine in the Thorntonbank wind farm near the Belgian coast.
Picture by Hans Hillewaert (20.03.2014) [21]

Figure 5: HAWT wind turbines

2.2.2. PARK Wake Model

We use a simple representation for the wake, based on the PARK model originally created by Jensen [25] in 1983 and further developed by Katic et. al [26] in 1986. As mentioned in Section 1.2, this model is well suited for wind farm simulations, mainly due to its balanced delivery of simplicity and accuracy [5]. It only considers velocity changes downstream from a turbine as a function of the distance x , as shown in Figure 6. Therefore, we do not perform an exact calculation of the velocity flow field with different directions. Furthermore, the PARK model was designed for the far wake case, i.e. turbines must be spaced at least 3 rotor diameters apart.

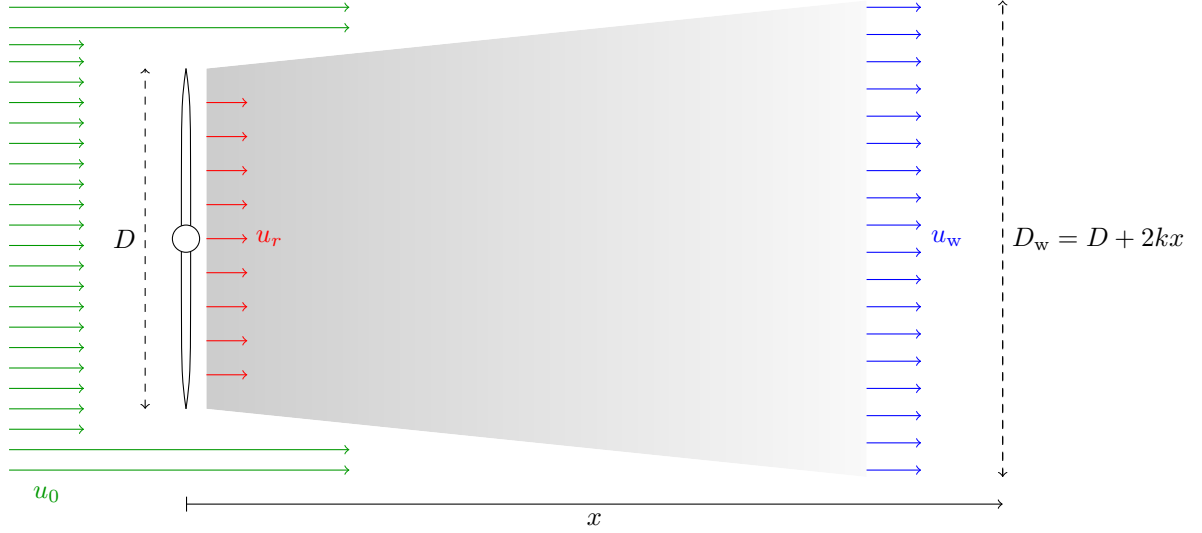


Figure 6: A top-view illustration of the wake effect as described in PARK. u_0 is the free stream wind velocity (also the incident velocity). u_r is the decreased velocity directly downstream from the turbine where airflow is highly turbulent. u_w is the wake-affected velocity where the airflow becomes steadier relative to that of u_r . Source: Heiming [22]

The wind speed data that was used had been measured at 100 meters above sea level, while the turbine features used in testing the model include hub heights around 80 m. While it would be ideal that wind speeds are measured at different heights directly, we can estimate these changes in speed using the *Logarithmic Law* [24]. This law can be used to approximate the wind speed at any height ≤ 100 m, given a reference speed and height. To estimate the speed at turbine height z knowing the wind speed at z_m :

$$u(z) = u(z_m) \frac{\ln((z - d)/z_0)}{\ln((z_m - d)/z_0)}, \quad (4)$$

where $u(z_m)$ is the reference wind speed, d is the zero plane displacement, and z_0 is the surface roughness. d accounts for large obstacles such as trees or buildings, therefore it is 0 for offshore wind farms.

The surface roughness is a measure of the condition of the site ground. In the offshore case, this measure would depend on the waves of the sea for example. Bernhard et al. [29] used a formula originally developed by Charnock [11] to calculate the surface roughness, but for the purpose of offshore wake modeling, it is estimated that the surface roughness is a constant at $z_0 = 0.0002$ m. Experiments carried out by DTU Wind Energy showed that for wind speeds of moderate to high values, using the fixed value of 0.0002 m was as good as using the equation [30]. The surface roughness is an important parameter for calculating the wake decay.

As seen in Figure 6, the wake diameter D_w of a turbine grows linearly by a factor of $2k$, where k is the wake decay factor and is defined as [39]:

$$k = \frac{0.5}{\ln \frac{z}{z_0}}, \quad (5)$$

with the hub height z and surface roughness z_0 . The wake decay is one of the parameters needed to calculate the velocity deficit from wakes.

As shown by Heiming, to derive the velocity deficit at a turbine affected by a wake, we first assume conservation of momentum within the area of the wake (illustrated by the gray-scale trapezium in Figure 6):

$$\sum \text{mass} \cdot \text{velocity} = \sum \text{density} \cdot \text{area} \cdot \text{velocity} = 0. \quad (6)$$

We further assume incompressibility of the fluid (air) and arrive at:

$$-\rho\pi\left(\frac{D}{2}\right)^2 u_r - \rho\pi\left(\left(\frac{D_w}{2}\right)^2 - \left(\frac{D}{2}\right)^2\right) u_0 + \rho\pi\left(\frac{D_w}{2}\right)^2 u_w = 0, \quad (7)$$

where ρ is the air pressure, and u_r is the wind speed directly behind the rotor of the turbine (as opposed to u_w , which is the wind speed further away from the turbine). This simplifies into:

$$D^2 u_r + (D_w^2 - D^2) u_0 = D_w^2 u_w. \quad (8)$$

The initial velocity deficit directly behind the turbine $\delta u_r = 1 - \frac{u_r}{u_0}$ is placed in Equation (8) and solved for $\frac{u_w}{u_0}$, yielding:

$$\frac{u_w}{u_0} = 1 - \delta u_r \left(\frac{D}{D + 2kx} \right)^2. \quad (9)$$

The initial velocity deficit δu_r is replaced with the relative loss at the turbine, where C_t is the thrust coefficient of the turbine:

$$a(u_0) = 1 - \sqrt{1 - C_t(u_0)} \quad (10)$$

which results in the final equation for the velocity deficit at any point inside the wake of a turbine with initial velocity u_0 :

$$\delta u(x) = 1 - \frac{u_w(x)}{u_0} = \frac{1 - \sqrt{1 - C_t(u_0)}}{\left(1 + \frac{2kx}{D}\right)^2}. \quad (11)$$

However, this derivation for the velocity deficit is only applicable for the wake of a turbine in the free stream. We need to consider the more likely case where turbines are downstream from other turbines. Thus to generalize the derivation, we examine the case of two interacting turbines. The first turbine that generates the wake is denoted by i and the turbine inside the wake is represented by j . The wind speed at turbine i is called incident wind speed $u_{\text{inc},i}$. In case turbine j is completely within the wake of i , we simply use Equation (11). However, if turbine j is only partially affected, we need to introduce a shadowing factor $\beta_k \in [0, 1]$ (Choi and Shan [12], OpenWind^o [46]):

$$\beta_k = \frac{A_{\text{Int}}}{A_{\text{Turbine}}}, \quad (12)$$

where A_{Int} is the area of the wake that intersects with the area of turbine j , denoted by A_{Turbine} and shown in Figure 7. The equation is updated to:

$$1 - \frac{u_w}{u_{\text{inc},i}} = \frac{\beta_k \left(1 - \sqrt{1 - C_t(u_{\text{inc},i})}\right)}{\left(1 + \frac{2kx}{D}\right)^2}. \quad (13)$$

However, yet another change is needed because Equation (13) depends on the incident velocity $u_{\text{inc},i}$ at turbine i . It should be generalized by making it dependent on the free stream velocity u_0 :

$$\delta u_{ij} = 1 - \frac{u_{w,i}}{u_0} = 1 - \frac{u_{\text{inc},j}}{u_0} = \frac{u_0}{u_{\text{inc},i}} \left(\frac{\beta_k \left(1 - \sqrt{1 - C_t(u_{\text{inc},i})}\right)}{\left(1 + \frac{2kx}{D}\right)^2} \right). \quad (14)$$

When two wakes are interacting, it assumed that the resulting mixed wake can be derived by simply summing the velocity deficits [26]:

$$\begin{aligned} \delta u_j^2 &= \delta u_{1,j}^2 + \delta u_{2,j}^2 \\ \Leftrightarrow \left(1 - \frac{u_{\text{inc},j}}{u_0}\right)^2 &= \left(1 - \frac{u_{w,1}}{u_0}\right)^2 + \left(1 - \frac{u_{w,2}}{u_0}\right)^2, \end{aligned} \quad (15)$$

where $u_{w,1}$ and $u_{w,2}$ are the velocities inside the first and second wake respectively. With an arbitrary number of interacting wakes, this formulation generalizes into:

$$\delta u_j = \sqrt{\sum_{i=1}^N \delta u_{ij}^2}, \quad (16)$$

with N being the number of interacting wakes and for each ij , δu_{ij} is the velocity deficit caused by the wake of turbine i which affects turbine j .

The velocity deficit at a turbine can then be used to calculate its generated power.

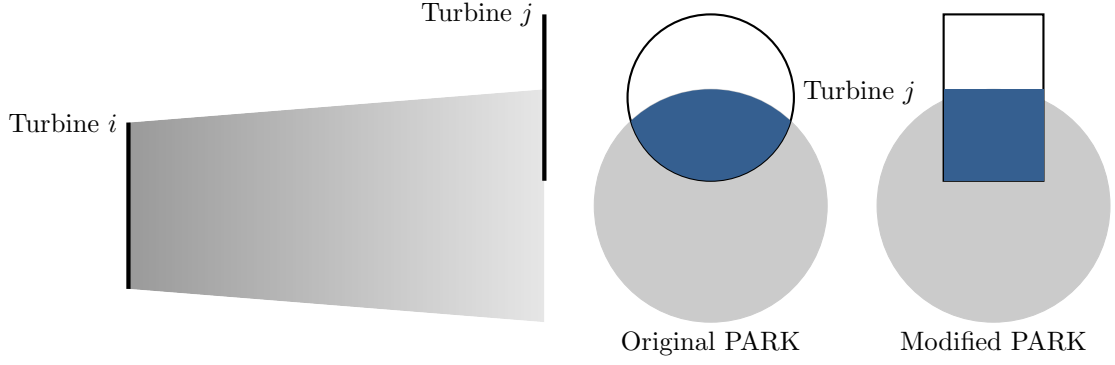


Figure 7: Visualization of the partial wake intersection according to the PARK model. The wake from turbine i crosses a section of turbine j . Source: Heiming [22]

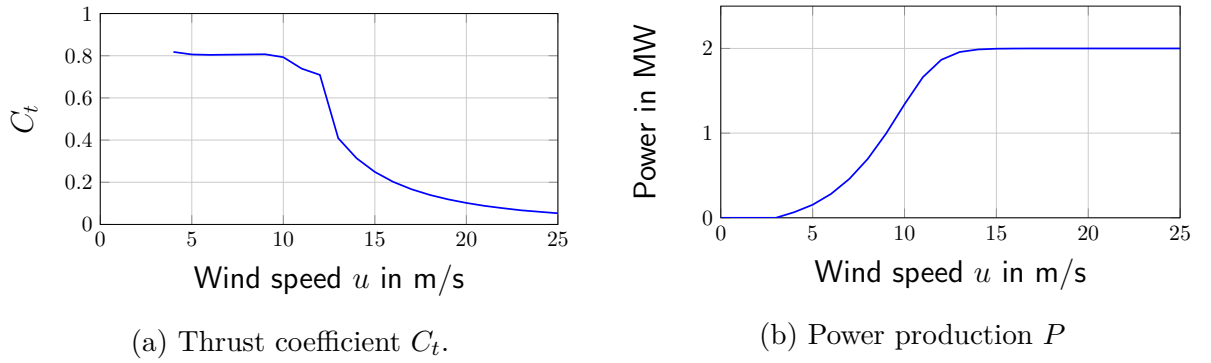


Figure 8: Performance of the Vestas V80 wind turbine with a cut-in speed of 4 m/s and cut-out speed of 25 m/s. Note that the power generation is not linearly dependent on the wind speed. Source: Heiming [22]

2.3. Power Generation Model

Turbines properties were briefly introduced in Section 2.2.1, and as mentioned turbines have a lower bound u_{cutin} and upper bound u_{cutout} that limit the range of speeds a turbine may function and produce power. Cut-in speeds are usually around 4 m/s (14 km/h), while cut-out speeds are around 25 m/s (90 km/h). Figure 8 shows how wind speed affects the power generation of a turbine, in comparison with its thrust coefficient.

The annual energy production (AEP) is based on calculating the expected power value E for all wind directions:

$$\begin{aligned}
 E &:= \int_0^{2\pi} E_\phi d\phi \\
 &\approx \sum_{i=1}^{N_\phi} w_{\phi_i} \cdot E_{\phi_i}
 \end{aligned} \tag{17}$$

with N_ϕ as the number of wind direction sectors, w_{ϕ_i} the weight for wind direction ϕ_i , and E_{ϕ_i} as the expected power in direction ϕ_i .

This mean power for a particular wind distribution E (measured in MW) is projected to the length of one year to achieve the AEP (measured in MWh):

$$\begin{aligned} AEP_{gross} &= (8760 \text{ h} + 6 \text{ h}) \cdot E \\ &\approx (8760 \text{ h} + 6 \text{ h}) \cdot \sum_{i=1}^{N_\phi} \sum_{j=1}^{N_u} w_{\phi_i u_j} \cdot \sum_{k=1}^{N_{\text{turbines}}} P(u_{inc, \phi_i, u_j, k}), \end{aligned} \quad (18)$$

where N_ϕ is the number of wind direction steps and N_u is the number of wind speed steps. $w_{\phi_i u_j}$ is the weight of the free-stream speed u_j at wind direction ϕ_i , and $u_{inc, \phi_i, u_j, k}$ signifies the incident velocity at turbine k , at the mentioned free-stream speed u_j and wind direction ϕ_i . 8760 is the hours in a year and the additional six hours is a correction to take into account leap years.

$w_{\phi_i u_j}$ is calculated differently depending if the wind is represented by Weibull distributions or bin data. If using Weibulls then:

$$w_{\phi_i u_j} = w_{\phi_i} \cdot \left(W(u_j - \frac{u_{\text{step}}}{2}, \lambda_{\phi_i}, \kappa_{\phi_i}) - W(u_j + \frac{u_{\text{step}}}{2}, \lambda_{\phi_i}, \kappa_{\phi_i}) \right), \quad (19)$$

where u_{step} is the wind speed step size in m/s and w_{ϕ_i} is the wind sector probability from the wind data. W is the cumulative Weibull function, where λ_{ϕ_i} is the scale parameter of wind sector ϕ_i , and κ_{ϕ_i} is its shape parameter, such that:

$$W(u, \lambda, \kappa) = 1 - e^{-(u/\lambda)^\kappa}. \quad (20)$$

However, if the wind mode is bins then:

$$\sum_{\phi_b = \phi_i - \frac{\varphi_{\text{step}}}{2}}^{\phi_i + \frac{\varphi_{\text{step}}}{2}} \sum_{u_b = u_j - \frac{u_{\text{step}}}{2}}^{u_j + \frac{u_{\text{step}}}{2}} w_{\text{bin} \phi_b u_b}, \quad (21)$$

such that φ_{step} is the wind direction step size, and $w_{\text{bin} \phi_b u_b}$ is the probability of the sections spanned by ϕ_b and u_b in the wind bin data. Finally, the power production is defined by:

$$P(u_{inc, \phi_i, u_j, k}) = \frac{P_c(u_{inc, \phi_i, u_j, k} - \frac{u_{\text{step}}}{2}) + P_c(u_{inc, \phi_i, u_j, k} + \frac{u_{\text{step}}}{2})}{2}, \quad (22)$$

where $P_c(u)$ is the power production of velocity u taken directly from the power curve.

Perhaps it would be simpler to take probabilities and power values directly from the wind data and power curves without incorporating values $\pm \frac{\text{step size}}{2}$, but this extensive interpolation is the approach taken by OpenWind, the software WindFlower is validated against. As mentioned in their validation report [46], they consider that when the wind speed is presented as u , realistically it is in a range $u \pm \frac{u_{\text{step}}}{2}$. A similar reasoning is applied to the power curve values, which is examined further in Section 2.5. Therefore for the purpose of the validation, the same implementation is adopted in WindFlower.

2.4. Cost Model

The AEP value computed with the three submodels (Figure 2) can be used further in the cost model to calculate other economical indicators to evaluate the performance of a wind farm, presented in the following sections.

2.4.1. Net Annual Energy Production

The gross AEP is not quite a realistic measure for wind farm energy production as it does not consider energy losses that usually occur due to grid downtime throughout the year. These losses can be summarized in one general performance loss factor p_{loss} , resulting in a new energy production measure:

$$AEP_{net} = AEP_{gross} \cdot (1 - p_{loss}) \quad (23)$$

The AEP_{net} is the basis for the other economical indicators presented next.

2.4.2. Levelized Cost of Electricity

The levelized cost of energy (LCOE) is a measure of the minimum cost per kWh over the life cycle of a wind park, such that its initial costs and costs of operation are covered. The model uses the formula developed by Lackner and Elkinton [28]:

$$K_{LCOE} = \frac{C_{capital} \cdot \frac{(1 + r_{rate})^l \cdot r_{rate}}{(1 + r_{rate})^l - 1} + C_{om}}{AEP_{net}} \quad (24)$$

where $C_{capital}$ is the total cost for turbines, cabling, substations etc., C_{om} the annual cost of operation and maintenance, and r_{rate} the discount rate including debt, taxes and insurance over an expected life cycle of l years.

2.4.3. Net Present Value

The net present value (NPV) is a measurement of profit, for the current financial value of a project. It is essentially the difference between the project's current cash inflows and outflows. Gonzales et al. [19] calculate it as such:

$$C_{NPV} = C_{prv} - C_{capital} + \sum_{t=1}^l \frac{AEP_{net} \cdot K_{energy} - C_{om}}{(1 + r_{rate})^t} \quad (25)$$

where C_{prv} consists of the current value of the project after the lifetime l (cash inflow). $C_{capital}$ is the installation cost and C_{om} is the operation and maintenance costs of the project ($C_{capital}$ and C_{om} being the basis of the cash outflow). K_{energy} translates to the current price of energy on the market. The parameters K_{energy} , C_{om} and AEP_{net} are usually dependent on time. However in the model we consider them as constants and replaced them with the respective averaged values collected from the wind data over several years.

2.4.4. Internal Rate of Return

The internal rate of return (IRR) is also a measure of profitability like the NPV, and in fact the same formula is used. However to obtain the IRR, r_{rate} in Equation (25) is set as r_{IRR} . To obtain r_{IRR} , we solve Equation (25) by setting the NPV as zero and r_{IRR} as unknown:

$$C_{\text{NPV}} = C_{\text{prv}} - C_{\text{capital}} + \sum_{t=1}^l \frac{AEP_{\text{net}} \cdot K_{\text{energy}} - C_{\text{om}}}{(1 + r_{\text{IRR}})^t} \stackrel{!}{=} 0 \quad (26)$$

For the project to be profitable, the r_{IRR} should be greater than the real discount rate r_{rate} and any additional risk deficits combined.

2.4.5. Cost model conclusion

The cost model provides a range of metrics to judge the fitness of a wind farm's turbine layout. Yet it will not be included in the validation, because OpenWind (the commercial program used as a reference) does not have the cost features available. However the AEP is used as the objective function in the validation tests, and this can be changed to any of the other cost functions presented earlier.

2.5. Validation of the models

In this section we validate the power, wind direction, wake, and wind speed. All the models have been implemented in a C++ program called WindFlower [22]. We validate the test results of our program against OpenWind, a commercial wind farm layout simulation software². The makers of OpenWind have validated it against another commercial software in addition to real wind and energy production data from wind power plants [46]. Therefore we take the results of OpenWind as a satisfactory reference solution.

2.5.1. Validation settings

OpenWind enforces some constraints that are taken into consideration when running the simulations.

Precision of input and output parameters The final results (such as the AEP) in OpenWind can be displayed up to only 10 decimal places with an energy unit of GWh, which for the simple test cases is not enough to show significant variation from WindFlower, thus makes it harder to pinpoint reasons for diverging results with complex test cases; the input it accepts is also limited to a relatively small number of significant digits, the most constricting being the Weibull parameters which can be up to only 5 significant digits; and finally for the wake decay constant parameter k , it is usually 0.04 for offshore wind farms [6], but our tests have implied that OpenWind

²<https://aws-dewi.ul.com/software/openwind/>

internally rounds up 0.04 to an unknown value less than 0.043, therefore we have chosen to use 0.045 instead as a wake decay constant value in the wake calculations.

Effect of u_{cutin} and u_{cutout} In OpenWind the cut-in and cut-out speeds are not treated as strict limits for a turbine's power production. To observe this relation we used OpenWind to simulate the power production of a single wind turbine over multiple wind speeds, with 1 m/s steps. Figure 9 shows the power curve of the turbine used in this test.

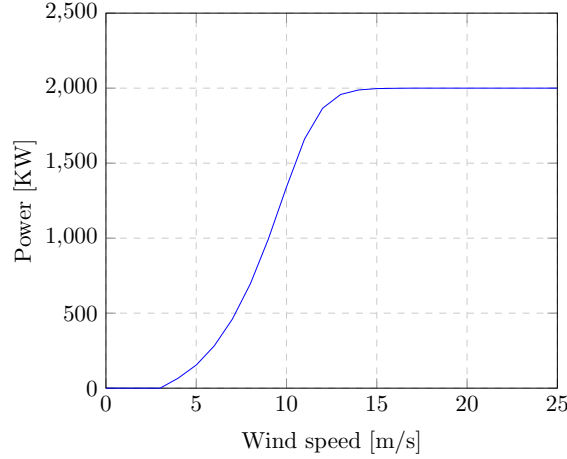


Figure 9: Power levels of the turbine type Vestas V80. The turbine's cut-off speeds are 4 m/s for u_{cutin} and 25 m/s for u_{cutout} .

The results of this test are shown in Figure 10. The power production was largely as expected for wind speeds ≤ 23 m/s. However, for wind speeds 24-25 m/s it produced less power than what the power curve in Figure 9 indicates. Furthermore, there was power produced in the interval 3-4 m/s and 25-26 m/s, even though both are beyond the cut-in and cut-out speeds respectively. As indicated briefly in Section 2.3, the official OpenWind validation report [46] offers some explanation to these results: When the program calculates the power production for a specific wind speed interval, it calculates the probability of this interval including probability values of speeds slightly above and below the speed interval. The size of this extra area on each end of the speed interval is determined by the wind speed step size u_{step} in the program's settings. For this test, the step size was the aforementioned 1 m/s.

Taking probabilities of speeds other than the specified range may result in unexpected behavior if the step size u_{step} is too large. This could lead to power output results that are far below the capacity of a turbine, or even higher than what its specification suggests.

Turbine height in Weibull tests The Vestas turbine settings are used in the validation which has a height of 70 m by default, but it will be adjusted to match the wind

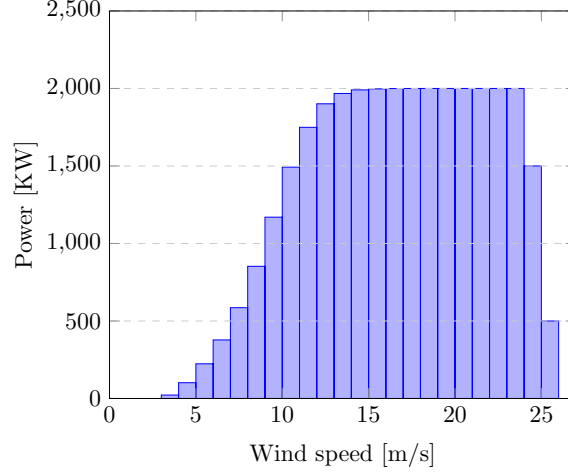


Figure 10: Simulated power production by a Vestas V80 turbine.

measurement height of 100 m only for the test cases using the Weibull distribution both for OpenWind and WindFlower. This is due to OpenWind-specific technical constraints when using Weibull wind data that does not allow adjusting wind speeds when turbine height and wind measurement height differ.

However it does allow the wind speed to be adjusted when using bin data, therefore tests with bins will have the original turbine height, which is an opportunity to implicitly test OpenWind's and WindFlower's methods of adjusting wind speeds.

Performance losses Non-wake energy losses are not considered in the validation, as currently OpenWind uses a simple cumulative function to apply the losses which does not have a significance in investigating [46]:

$$E_{\text{net}} = E_{\text{gross}} \cdot (1 - L_1) \cdot (1 - L_2) \cdot \dots \cdot (1 - L_n), \quad (27)$$

where E_{gross} is the AEP and n losses are applied, such electrical and availability losses. A more realistic loss modeling is presented by Richter et al. [37].

Given the above constraints, the validation settings are listed in Table 2.

2.5.2. Basics: Power, speed, and direction models

First we ensure that these individual submodels function correctly before incorporating the wake effects. To do this verification, a number of tests was carried out with different settings. The minimum accuracy between the results of WindFlower ($\text{AEP}_{\text{WindFlower}}$) and OpenWind ($\text{AEP}_{\text{OpenWind}}$) was 99.99999% .

Power model The calculation of a turbine's power production was examined by a simple test case of calculating the AEP of one turbine, with the wind coming from

Parameter	Symbol	Value
Wake decay constant	k	0.045
Number of wind direction sectors	N_ϕ	12
Number of simulated wind direction steps	N_φ	12
Cut-in speed	u_{cutin}	4 m/s
Cut-out speed	u_{cutout}	25 m/s
Simulated speed step size	u_{step}	1 m/s
Number of simulated speed steps	$N_{u_{\text{step}}}$	1 m/s
Turbine Type		Vestas V80-2MW
Site air density		1.225 kg/m ³
Air density lapse rate		0 kg/m ³ /km
Number of hours in year		8766
Turbine height	z	70 m
Wind measurement height	z_m	100 m
Surface roughness	z_0	0.001 m
Elevation		0 m
Energy Losses		0

Table 2: Settings used in the validation test cases, unless otherwise noted

one direction, having one main speed, and a real power curve such as that shown in Figure 9. For this test the error between OpenWind and WindFlower was $0.47 \cdot 10^{-8} \%$. Therefore for an extremely simple wind and speed scenario, the power interpolation works correctly. It will continue to be validated indirectly in the following tests for the direction and speed models, by virtue of the AEP results being accurate.

Direction model The wind direction model was validated as outlined in Table 3. The detailed test results are shown in Table 4. The results indicate a pattern between the number of wind direction sectors N_ϕ and the number of simulated direction steps N_φ : For test cases where $N_\varphi > N_\phi$, the AEP produced both in WindFlower and OpenWind was equal or largely similar to the AEP produced when $N_\varphi == N_\phi$, which is the expected result when the number of simulated steps is larger than the number of sectors. The maximum error between WindFlower and OpenWind results for any test case in the direction model test suite is $0.33 \cdot 10^{-9} \%$.

Speed model Another set of tests was carried out to ensure that WindFlower produces results that are accurate even when the simulated speed step size u_{step} is smaller or larger than the input speed step size, and when they have a non-linear relation. For example, the wind data measurements may be provided in intervals of 1 m/s, while the simulated speed step size u_{step} could be 0.1 m/s or 1.1 m/s. For Weibull distributions, the speed model is fairly simple because only the arguments in the Weibull cumulative function of a specific direction have to be changed when changing the simulated speed step size. However for speed data represented by bins of measurements, the interpo-

Test case	Wind mode		N_ϕ		N_φ				
	Weibull	Bins	12	32	12	32	48	501	720
1	✓		✓		✓				
2	✓		✓			✓			
3	✓		✓				✓		
4	✓		✓					✓	
5	✓		✓						✓
6	✓			✓	✓				
7	✓			✓		✓			
8	✓			✓			✓		
9	✓			✓				✓	
10	✓			✓					✓
11		✓	✓		✓				
12		✓	✓			✓			
13		✓	✓				✓		
14		✓	✓					✓	
15		✓	✓						✓
16		✓		✓	✓				
17		✓		✓		✓			
18		✓		✓			✓		
19		✓		✓				✓	
20		✓		✓					✓

Table 3: Wind direction interpolation tests for one turbine with one main speed in each sector. The number of direction sectors N_ϕ in the wind rose and the number of direction steps simulated N_φ were varied, both for bin and Weibull distribution wind data. The minimum and maximum direction steps possible in OpenWind are 12 and 720 respectively

Test	AEP _{OpenWind} (GWh)	AEP _{WindFlower} (GWh)	Error (%)
1 to 3	7.4978210576	7.49782105761832	$0.24 \cdot 10^{-9}$
4	7.4978210576	7.49782105761816	$0.24 \cdot 10^{-9}$
5	7.4978210576	7.49782105761850	$0.25 \cdot 10^{-9}$
6	12.2770654948	12.27706549478860	$0.09 \cdot 10^{-9}$
7 to 9	12.8490669586	12.84906695855700	$0.33 \cdot 10^{-9}$
10	12.8490669586	12.84906695855710	$0.33 \cdot 10^{-9}$
11 to 15	8.4178524835	8.41785248350562	$0.07 \cdot 10^{-9}$
16	14.3305207633	14.33052076327900	$0.15 \cdot 10^{-9}$
17 to 20	14.9142848386	14.91428483862240	$0.15 \cdot 10^{-9}$

Table 4: Wind direction interpolation test results

lation of these measurements must be implemented in a way to ensure adaptability to speed steps u_{step} of any size.

Table 5 shows the settings of the speed model tests. Some elements of the direction model tests were also incorporated in the speed tests to ensure that the speed interpolation functions correctly even alongside varied direction interpolation ranges. As shown in the results from Table 6, the maximum error between WindFlower and

Test case	Wind mode		u_{step}			N_ϕ		N_φ	
	Weibull	Bins	0.1	1.1	2	12	32	12	501
21	✓		✓			✓		✓	
22	✓		✓			✓			✓
23	✓		✓				✓	✓	
24	✓		✓				✓		✓
25	✓			✓		✓		✓	
26	✓			✓		✓			✓
27	✓			✓			✓	✓	
28	✓			✓			✓		✓
29	✓				✓	✓		✓	
30	✓				✓	✓			✓
31	✓				✓		✓	✓	
32	✓				✓		✓		✓
33		✓	✓			✓		✓	
34		✓	✓			✓			✓
35		✓	✓				✓	✓	
36		✓	✓				✓		✓
37		✓		✓		✓		✓	
38		✓		✓		✓			✓
39		✓		✓			✓	✓	
40		✓		✓			✓		✓
41		✓			✓	✓		✓	
42		✓			✓	✓			✓
43		✓			✓		✓	✓	
44		✓			✓		✓		✓

Table 5: Wind speed interpolation tests for one turbine with one main speed in each sector while varying the speed step size of the simulation u_{step} . The simulated speeds were in the range $[4, 25]$ m/s, represented by bins of size 1 m/s. The number of direction sectors N_ϕ in the wind rose and the number of direction steps simulated N_φ were varied, both for bin and Weibull distribution wind data.

OpenWind results for any test case in the speed model test suite is $0.4 \cdot 10^{-8}$ %.

2.5.3. Wake Model

Building upon the validation results of the previous submodels, the wake model is investigated in this section. The wake tests are divided into 5 scenarios: scenarios 1-4 are simple layouts where the wind comes from one direction with one speed, while scenario 5 uses real wind farm layouts with their real wind data. Only Weibull wind data is used in the following tests unless otherwise noted.

Scenario 1: Partial shading In this test case we have two turbines in the configuration shown in Figure 11. The energy production results for this layout are shown in Table 7. As mentioned before, it appears that OpenWind rounds their output to 10 decimal places. If the same rounding is done to WindFlower’s results, then in fact the

Test	$\text{AEP}_{\text{OpenWind}}$ (GWh)	$\text{AEP}_{\text{WindFlower}}$ (GWh)	Error (%)
21	0.9235897675	0.923589767463320	$0.397 \cdot 10^{-8}$
22	0.9235897675	0.923589767463299	$0.397 \cdot 10^{-8}$
23	12.2092655568	12.209265556806000	$0.005 \cdot 10^{-8}$
24	12.7873891468	12.787389146812000	$0.009 \cdot 10^{-8}$
25	0.8873646911	0.887364691091066	$0.101 \cdot 10^{-8}$
26	0.8873646911	0.887364691091045	$0.101 \cdot 10^{-8}$
27	12.2700740856	12.270074085626400	$0.022 \cdot 10^{-8}$
28	12.8398088793	12.839808879283700	$0.013 \cdot 10^{-8}$
29	0.4938744326	0.493874432614137	$0.286 \cdot 10^{-8}$
30	0.4938744326	0.493874432614125	$0.286 \cdot 10^{-8}$
31	12.2611454600	12.261145460021500	$0.018 \cdot 10^{-8}$
32	12.8405811235	12.840581123493700	$0.005 \cdot 10^{-8}$
33	0.7929431912	0.792943191223933	$0.302 \cdot 10^{-8}$
34	0.7929431912	0.792943191223957	$0.302 \cdot 10^{-8}$
35	14.0940633217	14.094063321667800	$0.023 \cdot 10^{-8}$
36	14.6778025357	14.677802535675100	$0.017 \cdot 10^{-8}$
37	0.7642568213	0.764256821291527	$0.111 \cdot 10^{-8}$
38	0.7642568213	0.764256821291518	$0.111 \cdot 10^{-8}$
39	14.3247924756	14.324792475586900	$0.009 \cdot 10^{-8}$
40	14.9085555949	14.908555594902400	$0.002 \cdot 10^{-8}$
41	0.4027414222	0.402741422209054	$0.225 \cdot 10^{-8}$
42	0.4027414222	0.402741422209045	$0.225 \cdot 10^{-8}$
43	14.3871579465	14.387157946528500	$0.020 \cdot 10^{-8}$
44	14.9712223287	14.971222328676700	$0.016 \cdot 10^{-8}$

Table 6: Wind speed interpolation test results

error rate for both wake model settings reduces to absolute 0.

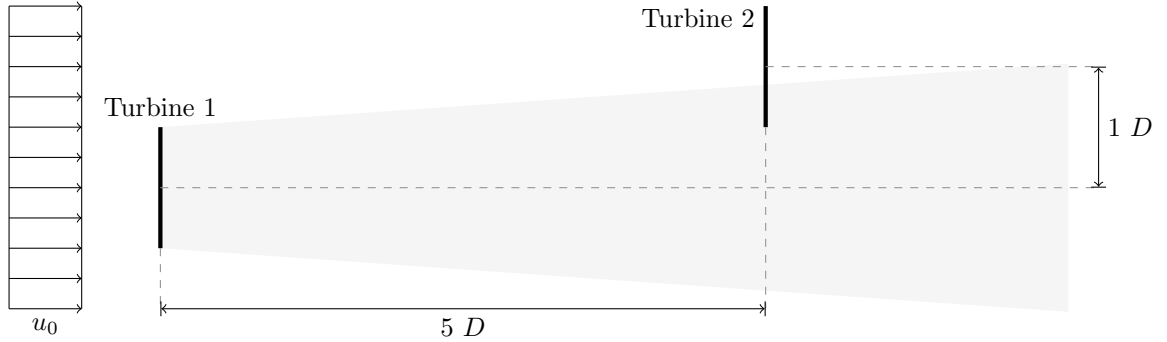


Figure 11: Wake scenario 1 with a partial shading layout. Source: Heiming [22]

Scenario 2: Full shading For this test there are also two turbines but the downstream turbine is completely inside the wake of the other turbine as shown in Figure 12. The

Model	$\text{AEP}_{\text{OpenWind}}$ (GWh)	$\text{AEP}_{\text{WindFlower}}$ (GWh)	Error (%)
Original Park	1.6801159448	1.68011594482096	$0.12 \cdot 10^{-8}$
Modified Park	1.5882090978	1.58820909781698	$0.11 \cdot 10^{-8}$

Table 7: Test results of partial shading

results for this test are listed in Table 8. The AEP output for the original Park model is equal to that of the modified model, unlike the case in scenario 1 shown in Table 7.

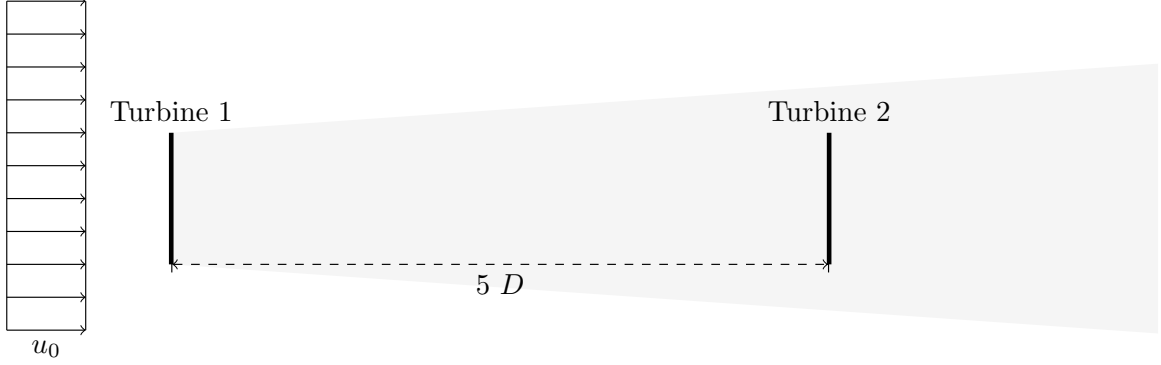


Figure 12: Wake scenario 2 with full shading. Source: adapted from Heiming [22]

Model	$\text{AEP}_{\text{OpenWind}}$ (GWh)	$\text{AEP}_{\text{WindFlower}}$ (GWh)	Error (%)
Original Park	0.9722061099	0.972206109949242	$0.51 \cdot 10^{-8}$
Modified Park	0.9722061099	0.972206109949242	$0.51 \cdot 10^{-8}$

Table 8: Test results of full shading

The results of test scenarios 1 and 2 highlight the difference between the original Park wake model and the modified version: The area of overlap between the downstream turbine and the wake of the upstream turbine is calculated differently in both model versions as illustrated in Figure 7. In the case of a fully shaded configuration such as in scenario 2, the area of overlap is 100%, thus the original and modified versions of the model give the same result for scenario 2, but for scenario 1 the area of overlap is partial and calculated differently for the two model versions.

Wake decay test Before continuing with test scenario 3, there is another aspect to explore in test scenario 2 which raises the question about the only tunable parameter in both versions of the Park model. This value is the wake decay constant k . It affects how far a wake takes to dissipate, thus at what distance for simple layouts it could be optimal to place a downstream turbine.

Starting again with the basic case of scenario 2 with a pair of turbines separated with $5D$ while one turbine is fully shaded by the other, we gradually increase the distance between the turbines only in the downstream wind direction, to note at what point the

wake effect is relatively eliminated and the power generation at full capacity. We use the Vestas V80 turbine model which has hub height of 70 m usually, but modified to 100 m for this test so that the free-stream wind speed is not affected by the different height of wind data measurement, as described in Section 2.2.2. The power curve for this turbine is outlined in Figure 9, and these values were used as references for the wake-affected speeds and powers. This test was done using the bins mode, and the results are plotted in Figure 13.

It can be noted that around a distance of $15 D$, the rate of increase in power begins to decrease, and the velocity deficit starts plateauing as well.

Comparing the power plots of the different speeds in Figures 13a, 13c and 13e, it shows a pattern that the higher the speed, the faster its power production converges. For speed 24 m/s, the power generation is always ideal even at a relatively small distance like $5 D$. However, the wake-affected wind speed plot of 24 m/s in Figure 13f shows that indeed there is a velocity deficit. The power generation is not affected because even at lower speeds like 23 m/s, the turbine produces its full power capacity as shown in the power curve in Figure 9.

This test also shows that theoretically the wake effect can not be completely eliminated if the turbines are directly behind each other.

At a downstream distance of $15 D$ the velocity deficit is $\approx 10\%$, and this result is similar to sources in literature where it is common that the velocity deficit from wakes is minimized around $10 D$ [2].

Scenario 3: Partial intersecting shading This test combines aspects from scenarios 1 and 2, where we have three turbines and only one is shaded by the other two such that its surface is fully affected by the intersecting wakes of the other turbines as shown in Figure 14. Table 9 lists the test results.

Model	AEP _{OpenWind} (GWh)	AEP _{WindFlower} (GWh)	Error (%)
Original Park	2.3255128370	2.32551283704678	$0.20 \cdot 10^{-8}$
Modified Park	2.2465738863	2.24657388626320	$0.16 \cdot 10^{-8}$

Table 9: Test results of partial intersecting shading

Scenario 4: Full intersecting shading In this scenario the layout consists of ten turbines in a row, where all of them (except the first) are fully shaded, as seen in Figure 15. This test illustrates a worst-case scenario where the wind direction is angled such that almost all the turbines are affected by strong wakes. It also yields the smallest percentage error between the results of WindFlower and OpenWind, as shown in Table 10.

Scenario 5: Real wind farm layouts In this scenario we move into testing more complex layouts and wind data, which belong to the Horns Rev 1, Sandbank, and DanTysk wind farms. The wind direction probabilities for these sites are shown in

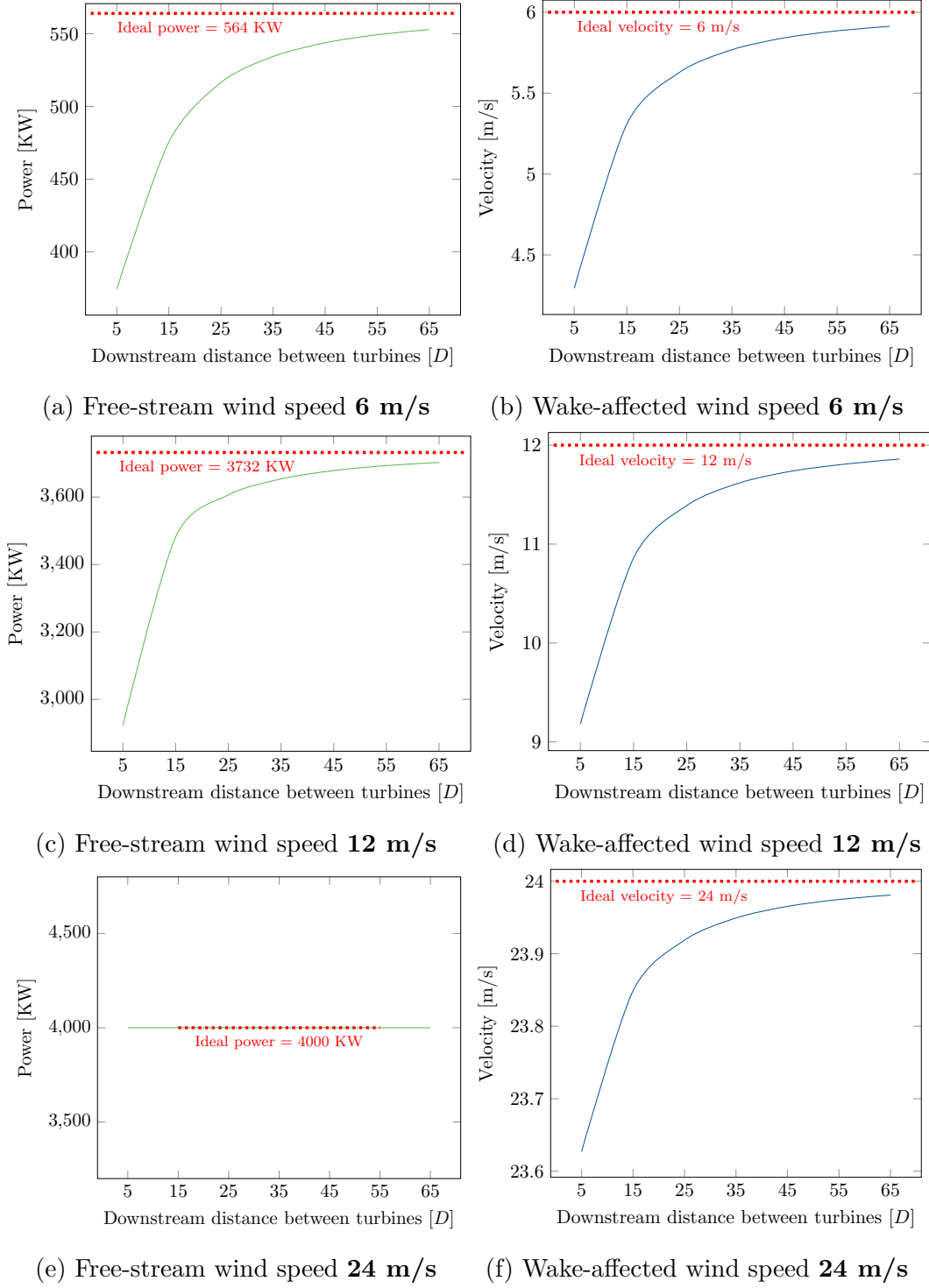


Figure 13: Wake decay test: Effect of varying the distance between two turbines where one is fully shaded by the other. Figures 13a, 13c and 13e on the left show the power generation of the site. While Figures 13b, 13d and 13f on the right show the velocity-deficit affected wind speed. Free-stream velocity and full power capacity at the wake-affected turbine therefore take more than 65 D to reach, where D is the turbine diameter.

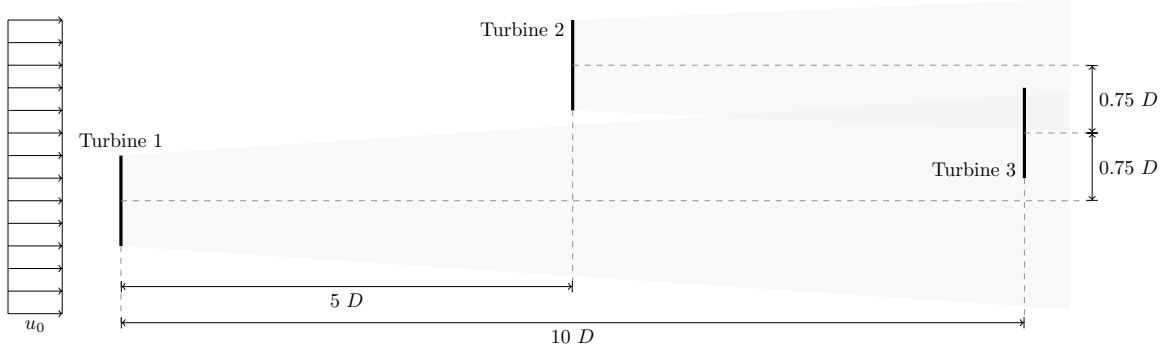


Figure 14: Wake scenario 3 with partial intersecting shading. Source: Heiming [22]

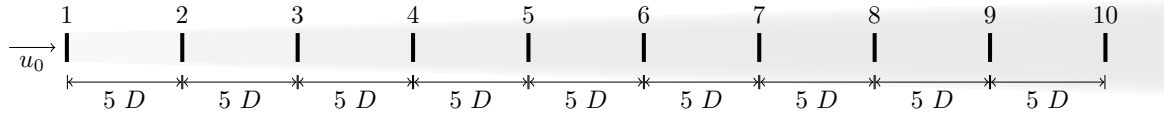


Figure 15: Wake scenario 4 with full intersecting shading. Source: Heiming [22]

Model	AEP _{OpenWind} (GWh)	AEP _{WindFlower} (GWh)	Error (%)
Original Park	1.9293283509	1.92932835088315	$0.9 \cdot 10^{-9}$
Modified Park	2.5530129596	2.55301295959630	$0.9 \cdot 10^{-9}$

Table 10: Test results of full intersecting shading

Figure 16. The turbine positions in each layout can be seen in Figure 46. The fact that wind is coming from all directions and turbines are positioned at different angles from each other makes the wake effect more complex than in the previous scenarios.

All three wind farm layouts were simulated for their AEP and the results are consolidated in Figure 18. The maximum margin of error between OpenWind and WindFlower was 0.0004%, significantly higher than the previous simple scenarios. It can also be seen that the Horns Rev1 tests have significantly higher accuracy than the other two wind farms, but currently the reason is undetermined. All in all, the differences in the results are negligibly small.

2.5.4. Wake and wind speed model

Previously in Section 2.5.2 it was established that the speed and direction models give reasonable results for simple cases. In this section we test the speed model in more detail, incorporating the wake model as well. For these tests we use multi-directional wind data shown previously in Figure 16, varying mainly the number of simulated wind speeds. The tests are performed for the fully shaded and partially shaded layouts, and are shown in Figure 19.

The number of wind speed steps $N_{u_{\text{step}}}$ used to calculate the ideal AEP was 211, which is the maximum value permissible by OpenWind for this scenario. There was

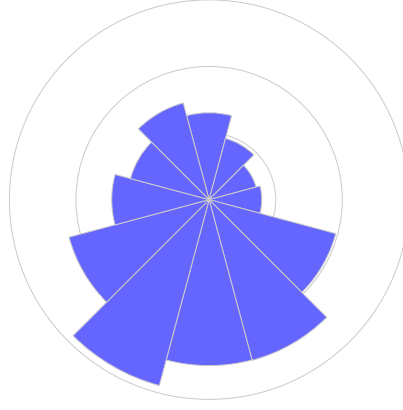


Figure 16: Wind distribution for wind farm layouts. The size of each sector determines the probability of wind coming from that direction. For example, the most probably direction in this case is South Southwest with probability $\sim 14\%$

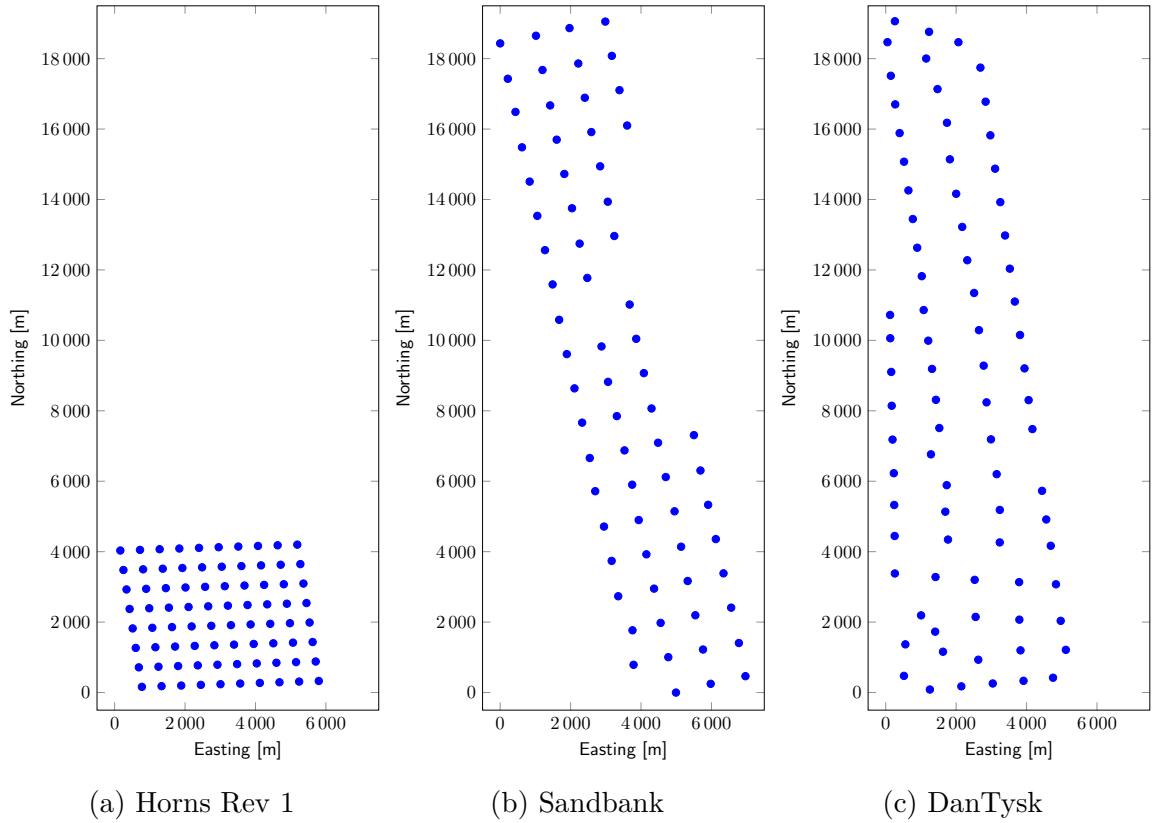


Figure 17: Layouts of real wind farms used in the validation tests

no significant difference between the results of OpenWind and WindFlower again, thus they could not be distinguished in the plot.

Interesting to note that the results start to converge around 100 simulated speeds with a relatively low error of $\approx 0.02\%$. Therefore this value could be an optimal

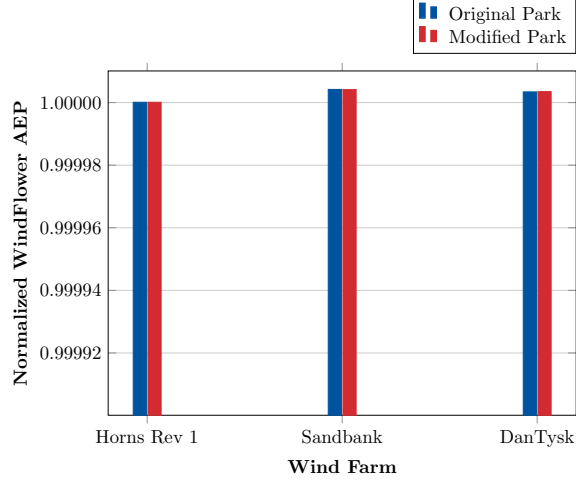


Figure 18: Validation test results for real wind farm layouts. Displayed here are the results from WindFlower normalized against those from OpenWind.

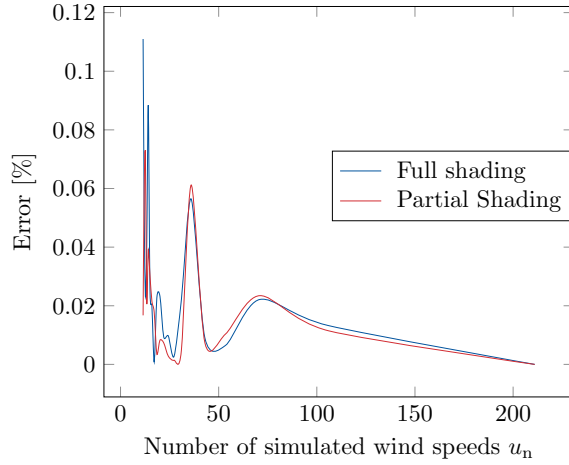


Figure 19: Testing the effect of varying the number of simulated wind speeds $N_{u_{\text{step}}}$ on AEP, for layouts with full shading and partial shading (shown in Figures 11 and 12) for wind coming from all directions. The reference AEP was calculated from simulations using 211 wind speed steps.

default for the number of speeds parameter. However there is also a local minima for the error around 30 simulated speed steps $N_{u_{\text{step}}}$, which could be a good compromise if processing time is a critical resource. Simulating 100 speeds can take around 2 seconds, while simulating 30 speeds is almost four times faster at about 0.5 seconds.

2.5.5. Wake and wind direction models

In this section we test the effect of varying the number of direction sectors N_ϕ in the wind rose and of the wind direction steps N_φ on the energy production. The fixed

aspects of the tests are: having wind speed steps constant and always the same simple configuration of four wind turbines shown in Figure 20.

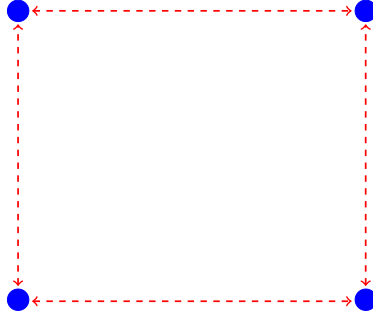


Figure 20: Layout of turbines in a grid pattern. A dashed line represents the distance between two turbines, which is an integer multiple of the turbine's diameter.

Results are presented in Figure 21, which show significant divergence before converging around 100 simulated wind direction steps N_φ . This inconsistent behavior is expected for small numbers of simulated directions, as much of the available wind direction data is not included in the simulations in different steps for each number of directions.

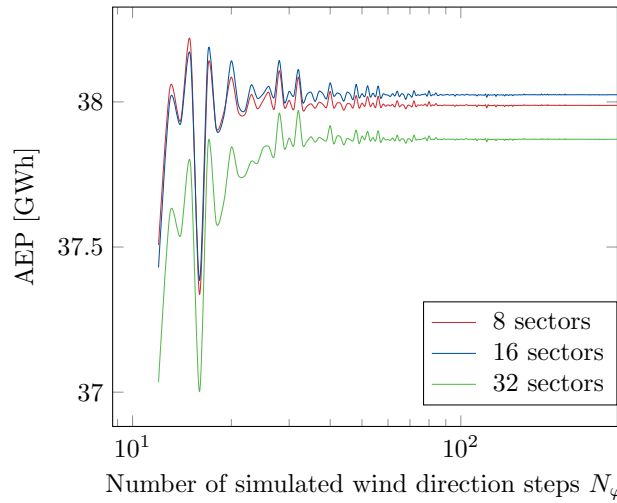


Figure 21: Effect of varying the number of simulated direction steps on AEP, for layouts using wind data made of 8, 16, and 32 sectors respectively. The number of direction steps ranged from 12 to 300

2.6. Acceleration of the Model

The purpose of the model's acceleration is to derive the optimal default settings for running simulations in WindFlow.

2.6.1. Number of wind speeds and wind directions

In this test we study the effect of varying the number of simulated wind speed steps $N_{u_{\text{step}}}$ and wind direction steps N_{φ} on the AEP, while keeping all other parameters constant. This test was repeated three times for the layout shown in Figure 20, alternating among these distances between the turbines: $3 D$, $5 D$, and $7 D$. These values particularly may be interesting because $3 D$ is around the typical minimum separation distance between turbines, while $5 D$ and $7 D$ are approximately the average distance between neighboring turbines in Horns Rev 1 and Sandbank respectively. Figures 22 and 23 show the results of this test.

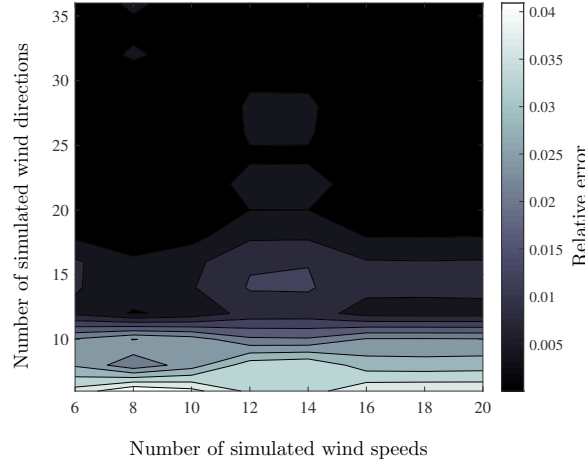


Figure 22: Effect of varying the number of simulated wind direction and wind speed steps on the relative error of the AEP output for 4 turbines in a grid layout separated by $3 D$. The ideal AEP was defined as the result from a simulation with 720 direction steps and 211 speed steps.

The ideal AEP output in these tests was defined as the simulation result with 211 wind speed steps $N_{u_{\text{step}}}$ and 720 wind direction steps N_{φ} . Both these values were chosen because they were the maximum options in OpenWind. The full tests were run in WindFlower, and the ideal (reference) AEP was also calculated in OpenWind for confirmation.

The Weibull distribution rose in these tests has 32 sectors, and each sector has a different Weibull distribution. However, for all three cases it holds that an error rate of 0.5% can be reached with as little as 28 direction steps N_{φ} . This may imply that the number of simulated direction steps N_{φ} can be lower than the number of wind sectors N_{ϕ} and still get an acceptable simulation result.

It can also be seen that increasing the number of direction steps N_{φ} has a greater impact on the relative error than increasing the number of wind speed steps $N_{u_{\text{step}}}$.

All three tests suggest that 28 simulated direction steps and 6 simulated speed steps are enough to give the most accurate result which matches the reference AEP with 0.5% error. This eliminates the need to simulate hundreds of directions and speeds to reach a reliable result.

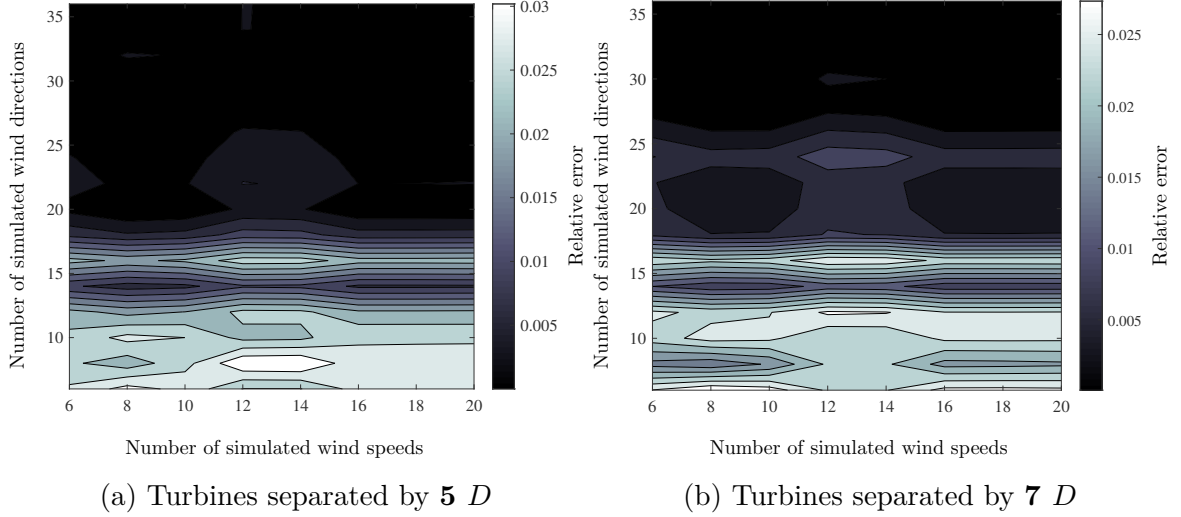


Figure 23: Effect of varying the number of simulated wind direction and wind speed steps on the relative error of the AEP output.

The results also consistently show varying degrees of lowered accuracy when simulating with 12 or 14 wind speed steps $N_{u_{\text{step}}}$. This could be a property of the raw wind data used in these tests. To examine if these estimates can be generalized for other scenarios, the previous three tests were repeated twice: once with a wind distribution of 8 sectors, then with 64 sectors. The full results are available in the appendix in Figure 48. These plots do not show consistently the same pattern of low accuracy for 12 and 14 simulated speeds, which could indicate that the optimal number of wind speed steps $N_{u_{\text{step}}}$ may not only depend on the raw wind data but also on the underlying wind data representation.

As for the number of simulated directions N_{φ} , all plots show a pattern that as the distance between turbines increases, the minimum number of wind direction steps required to give an accurate result also increases. This intuitively coincides with the model's approach in simulating the wind scenario, as it takes each turbine and detects which other turbines are in its wake, for each simulated direction. The farther the turbines are apart, the less likely a small number of wind direction steps would be sufficient to give a reliable score.

The results so far confirm the earlier observation that simulating 28 directions and 6 speeds is sufficient to produce a result that matches the reference output with at least 99.5% accuracy. However these are all simple scenarios. To study what a more realistic effect would be, we repeat the test with 32 wind sectors on the Horns Rev 1 layout. The results for the real wind farm are shown in Figure 24. Even for such a complex scenario, 28 wind direction steps N_{φ} and 6 wind speed steps $N_{u_{\text{step}}}$ give a relatively high accuracy of 99%. Therefore these values seem appropriate for simulating sites that have similar wind distributions.

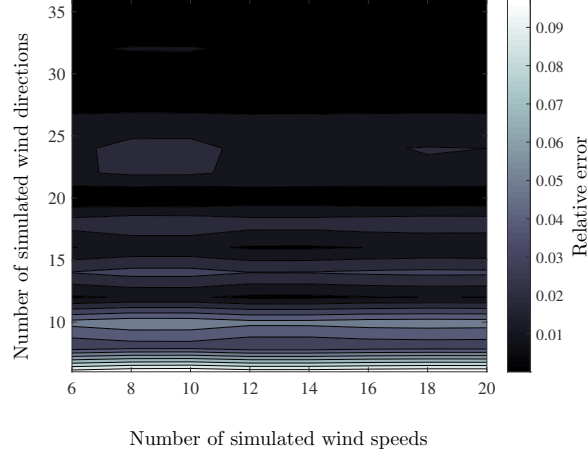


Figure 24: Effect of varying the number of simulated wind direction and wind speed steps on the relative error of the AEP output for the turbine layout of Horns Rev 1, shown in Figure 17a. The ideal AEP was defined as the result from a simulation with 720 direction steps and 211 speed steps.

2.7. Model conclusion

The brief parameter studies done in this section suggested that a separation of $5 D$ between turbines resulted in $\approx 25\%$ velocity deficit, which makes it a good initial guess in an optimization.

Furthermore, to decrease the computation time we can simulate 28 directions N_φ and 6 speeds $N_{u_{\text{step}}}$, and still produce results with an accuracy of at least 99%.

Overall, given that the maximum error between our calculations and OpenWind was 0.0004%, the model can be used in optimizations fairly reliably.

3. Optimization

Utilizing the model and its validation, three optimization algorithms in the context of a multi-step-optimizer were developed and evaluated for their efficiency: Square Grid, Hexagon Grid, and Close Packing. Although they have different approaches for optimizing turbine layouts, all three algorithms generate positions that must be:

- At least $2 D$ apart for safety reasons;
- Within the specified site area, defined by an irregular concave polygon;
- Not inside any restricted areas.

After producing these initial solutions, the second stage in the multi-step-optimizer is to further improve the pattern-generated positions with local search heuristics.

3.1. Square Grid

The Square Grid algorithm is a pattern-based optimization motivated by commercial constraints to have wind farm layouts with simple channels for navigation of ships. [36].

3.1.1. Square Grid Algorithm

It starts with an empty area and places the required amount of turbines along intersections of a grid that can be scaled and skewed. Figure 25 shows a typical optimization result from Square Grid where the main parameters are highlighted.

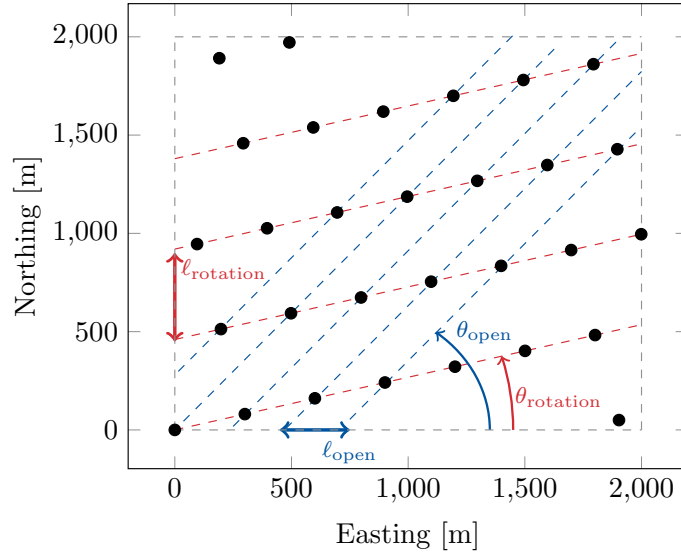


Figure 25: SquareGrid example result and main parameters. The gray dashed lines represent the site borders. The green dashed lines are the main orientation lines of the grid, where the lines are separated by α_{rotation} sized steps, and are at an angle of θ_{rotation} from the origin. The same interpretation applies to the blue dashed lines which however represent the grid opening lines.

The algorithm first detects the main wind exit direction (angle) $\theta_{\text{main wind}}$ in the site, then the Square Grid angle parameters search-space is defined as:

$$\begin{aligned}\theta_{\text{rotation}} &\in [\theta_{\text{main wind}} - 90^\circ, \theta_{\text{main wind}} + 90^\circ) \\ \theta_{\text{open}} &\in (\theta_{\text{rotation}}, \theta_{\text{main wind}} + 90^\circ).\end{aligned}\tag{28}$$

The intervals are iterated by a user-defined number of steps N_θ . The range of the iterated angles is mainly $\pm 90^\circ$ away from the main wind exit direction $\theta_{\text{main wind}}$ because angles larger than 90° just lead to repeated lines, i.e. a line that is 95° away from the origin is the same line that is -85° from the origin. As for the distances between the lines, they are limited by:

$$\ell_{\text{rotation}} \in [2D, 20D)$$

$$\ell_{\text{open}} \in [2D, 20D), \quad (29)$$

where $2D$ is chosen because it is approximately the minimum distance acceptable between turbine positions, and $\approx 20D$ is a large distance that could be attainable if a small number of turbines is required, relative to the size of the field. Furthermore, the wake decay tests in Figure 13 from Section 2.5.3 show that after a distance of $\approx 15D$ between two turbines, increasing the distance further does not result in as much efficiency gain. Therefore $20D$ could be a suitable upper-bound for the distance between turbines. The number of steps between the lower and upper bounds N_ℓ is specified by the user. However, the steps are only used for the main lines of rotation ℓ_{rotation} , because the grid opening lines ℓ_{open} are found by optimization, such that the required amount of turbines N_{turbines} fit into the site area.

The last main parameter is a growth factor ξ_{growth} , which allows the distances between consecutive parallel lines to gradually increase or decrease along with the direction of the wind entry into the site, as shown in Figure 26. The growth factor can either be negative, which will cause consecutive line distances to shrink, or it can be positive, allowing the distances to grow. If it is 0 then the distances between consecutive lines remains constant. The growth factor parameter iterates over the range:

$$\xi_{\text{growth}} \in [\xi_{\text{start}}, \xi_{\text{end}}], \quad (30)$$

where ξ_{start} and ξ_{end} are user-defined parameters. Furthermore, the number of steps N_ξ between the bounds is also customizable. This growth factor ξ_{growth} is applied such that the positions that have more up-stream turbines can also have larger distances in front of them, such that the intersecting wakes affecting them are more likely to dissipate. This aspect is tested in Section 3.1.2.

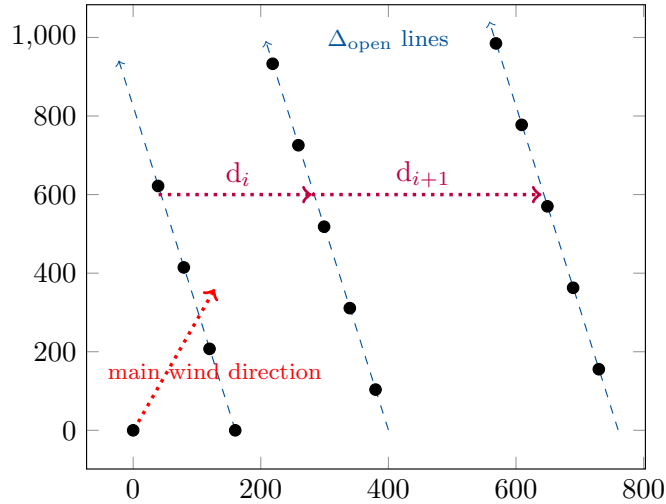


Figure 26: Visualization of the growth factor in the SquareGrid optimization. The distances between consecutive lines grows by $(1 + \xi_{\text{growth}})$ of the previous distance. Thus $d_{i+1} = d_i \cdot (1 + \xi_{\text{growth}})$

The algorithm's abstract idea is shown in Figure 27. In addition to the customizable parameters, the algorithm's input are mainly the corner coordinates of the wind farm, and the required number of turbines to place. The Square Grid algorithm then iterates in equidistant steps over the four main parameters: θ_{rotation} , θ_{open} , ℓ_{rotation} , and ξ_{growth} . Once in the innermost of these nested loops, it optimizes for ℓ_{open} using either the simple Bisection method or the Regula Falsi method, depending on a user flag. The Bisection method was chosen because it is a simple reliable approach for root-finding, while the Regula Falsi method may be more complex but it can converge faster. The Secant method was also tried in an earlier version of Square Grid, and while it converged faster than the Bisection method, it did not seem to produce stable results.

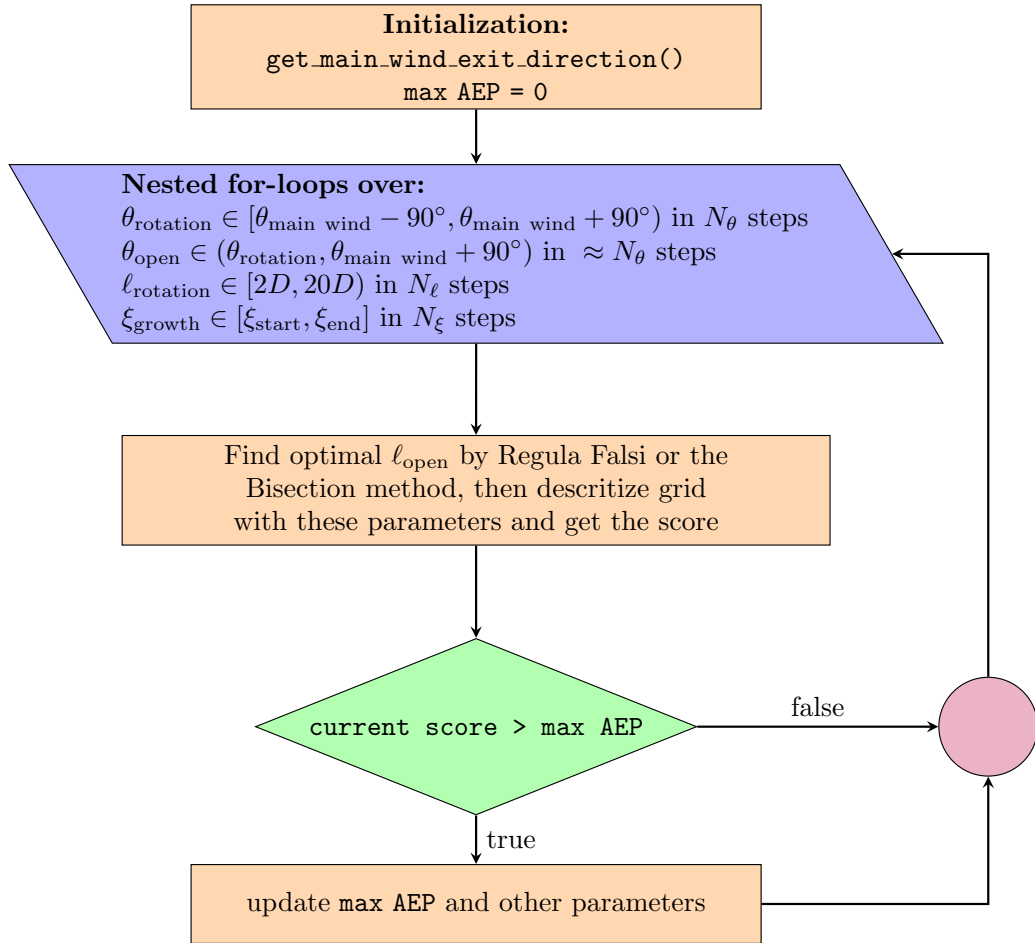


Figure 27: Square Grid main optimization algorithm flow. After the initialization with the required data, the optimization loops over the parallelogram parameters to try various combinations of positions.

The function used in the core-optimization is the `discretizeGrid(...)` method, which produces a list of positions based on the combination of the input parameters defined earlier. It works mainly by creating two sets of lines Δ_{rotation} and Δ_{open} and then assigning the intersections as positions. The line sets are created such that:

- Within each set the lines are parallel to each other;
- Between the two sets, each line in one set intersects all the lines in the other;
- The set Δ_{rotation} has equidistant lines while the set Δ_{open} can have a variable growth factor ξ_{growth} .

The generated set of positions is then verified against the conditions introduced at the start of this section, to ensure all positions are in valid areas. If any position is invalid, the whole set is not accepted, because uniformity of the pattern is a strong requirement of the Square Grid algorithm.

3.1.2. Square Grid Parameter Studies

Effect of number of simulated wind speeds and directions on Square Grid In the validation Section 2.6.1 we arrived that simulating 28 wind directions N_φ and 6 speeds $N_{u_{\text{step}}}$ can give around 99% accuracy to the reference solution. We re-examine this relation between speeds and directions but in the context of the Square Grid optimization, by optimizing Horns Rev 1 with different numbers of speeds and directions for 12 and 32 wind sectors, but keeping the Square Grid parameters themselves fixed. Figure 28 shows that around $N_\varphi = 28$ directions and $N_{u_{\text{step}}} = 6$ speeds we have approximately 98% accuracy, which is only 1% off the validation results. The error reduces further for higher numbers of simulated speeds. The reference solution for each plot was an optimization with $N_\varphi = 100$ wind directions and $N_{u_{\text{step}}} = 211$ wind speeds (which translates to 0.1 m/s wind speed steps u_{step}). We also see that for both wind sector discretizations of 12 and 32, the patterns are very similar w.r.t. the relative error. For the purpose of the optimizations, we will take 28 simulated directions and 22 simulated speeds (the equivalent of 1 m/s wind speed step sizes u_{step}) because the area of minimal error from the plots is larger around these values.

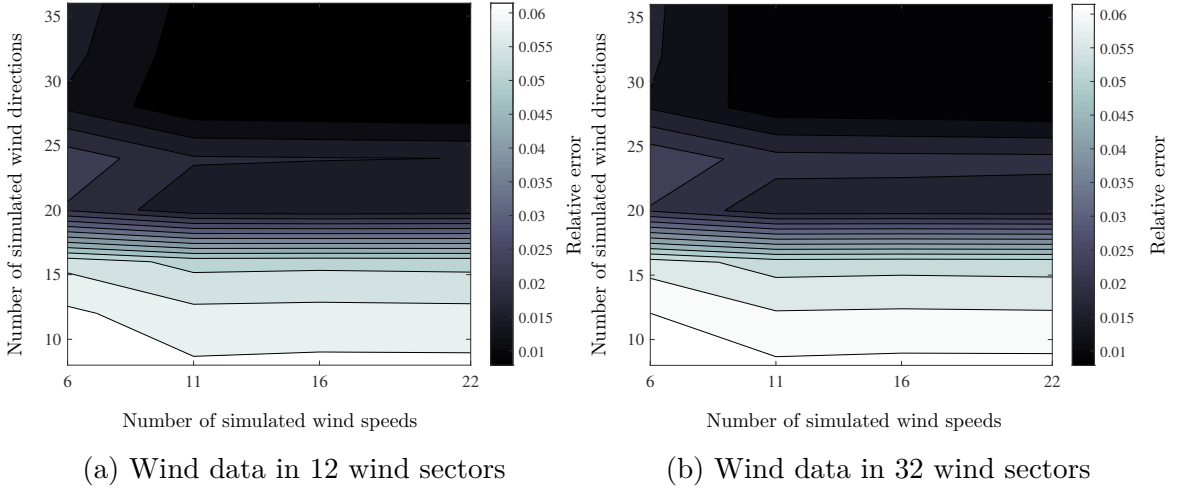


Figure 28: Effect of varying the number of simulated wind direction and wind speed steps on the relative error of the optimization's efficiency score.

Growth factor The growth factor ξ_{growth} was introduced to mitigate the effects of intersecting wakes on turbines at the back of a wind farm relative to the wind entry direction. To examine this parameter's effect on efficiency, we optimize the Horns Rev 1 wind farm, trying different growth factor values $\xi_{\text{growth}} \in [-0.1, 0.4]$ in steps of 0.02, and record the maximum efficiency each growth factor value ξ_{growth} was able to achieve, as shown in Figure 29. The plot shows that for growth factors < 0 , the efficiency is relatively low. This seems to correlate with the idea that turbines in the back of the farm relative to the wind entrance direction, should not be closer together than those turbines that benefit more from the free-stream wind. The efficiency improves significantly at $\xi_{\text{growth}} = 0$ where it is also the maximum. Beyond this point, the efficiency begins to gradually decrease.

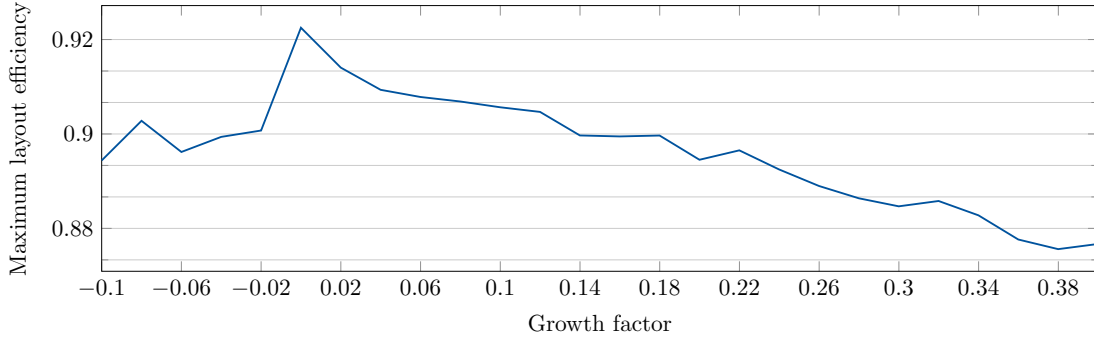


Figure 29: Effect of using different growth factors ξ_{growth} on optimizing for Horns Rev 1.

Based on these results, we investigate again in more detail the range $\xi_{\text{growth}} \in [-0.02, 0.04]$ in steps of 0.005, to get a clearer idea of the region where the maximum efficiency is achieved. The plot in Figure 30 shows the detailed results. The growth factor $\xi_{\text{growth}} = 0$ still produces the maximum efficiency, but it can be seen that $\xi_{\text{growth}} = 0.1$ is a very close second, with efficiency only 0.00151 less than that of $\xi_{\text{growth}} = 0$. Furthermore, $\xi_{\text{growth}} = -0.01$ also produced comparably good results.

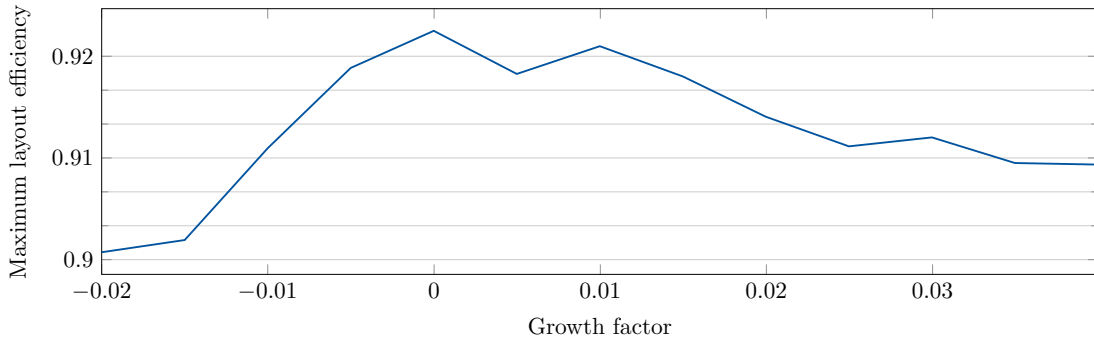


Figure 30: Effect of using different growth factors ξ_{growth} on optimizing for Horns Rev 1 using fine steps.

Therefore we recommend that the range for the growth factor ξ_{growth} is within $\xi_{\text{start}} = -0.01$ and $\xi_{\text{end}} = 0.02$, with number of steps $N_{\xi} = 3$.

3.2. Hexagon Grid

The idea of Hexagon Grid is that the wind farm area is divided into regular hexagons, such that the hexagons can be rotated and scaled, but not skewed. A turbine position is placed on each hexagon corner. If positions were also placed in the centers of hexagons, it would look like a special case of Square Grid. Even though Hexagon Grid complies with the constraint that positions are organized into a grid structure, it requires less run-time than Square Grid because it has only two main optimization parameters: θ_{hex} and ℓ_{side} , where θ_{hex} (similarly to θ_{main}) is the main angle of rotation of the grid pattern and ℓ_{side} is the length of one hexagon side. The search space of θ_{hex} is constricted to:

$$\theta_{\text{hex}} \in [\theta_{\text{main wind}} - 60^\circ, \theta_{\text{main wind}} + 60^\circ], \quad (31)$$

such that we only try hexagon patterns at $\pm 60^\circ$ away from the main wind exit direction $\theta_{\text{main wind}}$. Higher angles than 60° would only cause the Hexagon pattern to repeat.

The side of the hexagons ℓ_{side} is optimized similarly to the distance between Square Grid lines ℓ_{open} , using either the Bisection method or Regula Falsi. Therefore we only iterate over one parameter θ_{hex} , where the number of steps over its search space interval is a user-defined parameter $N_{\theta_{\text{hex}}}$.

Figure 31 shows an example layout of Hexagon Grid, noting the main optimization parameters.

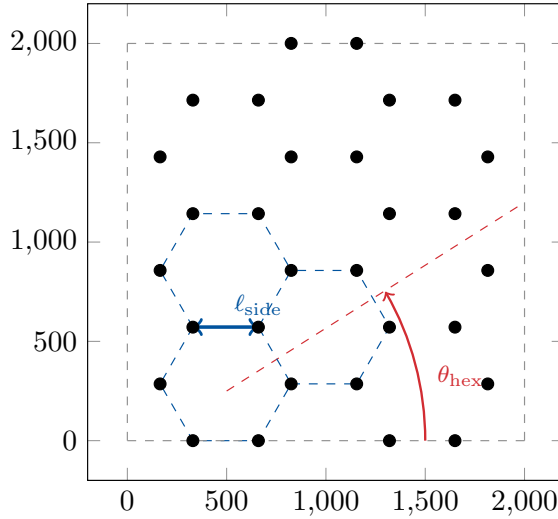


Figure 31: Hexagon Grid example result and main parameters. The gray dashed lines represent the site borders. The green dashed line is the main orientation line of the grid which is θ_{hex} from the origin. The blue line ℓ_{side} represents the side of a hexagon. All hexagons are regular.

3.2.1. Hexagon Grid Parameter Study

The number of angle steps $N_{\theta_{\text{hex}}}$ is the only Hexagon Grid variable which can be changed by the user, thus we attempt to find a good default value for this parameter. Again optimizing for Horns Rev 1, we vary the number of steps $N_{\theta_{\text{hex}}}$ and compare the efficiency reached each time, as shown in Figure 32.

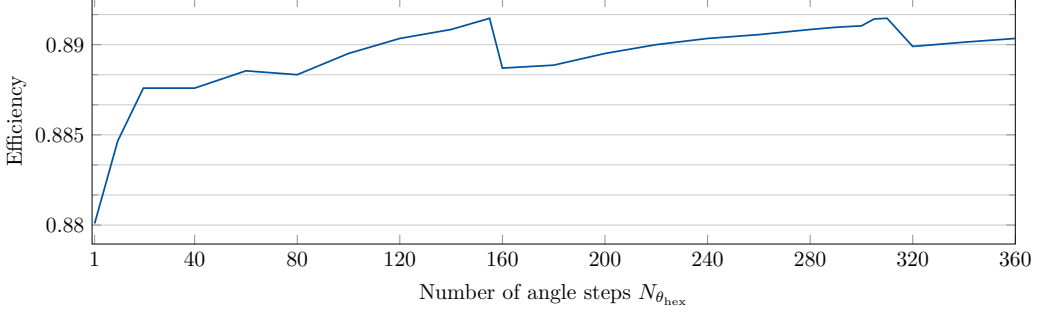


Figure 32: Effect of number of angle steps $N_{\theta_{\text{hex}}}$ on layout score.

There is a pattern of improvement until number of steps $N_{\theta_{\text{hex}}} \approx 160$, but no significant improvements after that. As noted in Table 11, which shows the detailed results, the run-time (achieved with only 1 thread) gradually increases with the number of steps $N_{\theta_{\text{hex}}}$ faster and more uniformly than the score. The difference between the score of $N_{\theta_{\text{hex}}} = 20$ and $N_{\theta_{\text{hex}}} = 360$ is only 0.005849, while the run-time for $N_{\theta_{\text{hex}}} = 360$ is almost 14 times greater. However, because all run-times are under one minute, the small improvement in score is worth the time.

It can also be seen that some scores repeat, for example $N_{\theta_{\text{hex}}} = 20$ and $N_{\theta_{\text{hex}}} = 40$ give 0.887591, as well as $N_{\theta_{\text{hex}}} = 140$ and $N_{\theta_{\text{hex}}} = 280$ both result in 0.890841. This follows from the fact that if $N_{\theta_{\text{hex}1}}$ is a multiple of $N_{\theta_{\text{hex}2}}$, then the angle step size of $N_{\theta_{\text{hex}1}}$ is a divisor of the angle step size of $N_{\theta_{\text{hex}2}}$. Therefore the search space of $N_{\theta_{\text{hex}1}}$ includes the search space of $N_{\theta_{\text{hex}2}}$, and the score of $N_{\theta_{\text{hex}1}}$ should be equal or greater than that of $N_{\theta_{\text{hex}2}}$. Building on this simple concept, that smaller step sizes could lead to greater coverage of the search space, we optimized again but with $N_{\theta_{\text{hex}}} = 12000$ which leads to a step size of 0.01° . This optimization run took ≈ 35 minutes to finish, and resulted in an efficiency score of 0.89167, higher than those in Table 11.

The non-uniformity of score shown in Table 11 indicate that Hexagon Grid is sensitive to the value of the number of angle steps $N_{\theta_{\text{hex}}}$, and a higher value does not guarantee finding the best score. However, if time is no constraint, $N_{\theta_{\text{hex}}} = 12000$ steps will at least give 89.2% accuracy after 35 minutes of run-time. Otherwise, $N_{\theta_{\text{hex}}} = 240$ steps will achieve a similar efficiency of about 89% in under a minute of run-time. Therefore these are the values recommended as the default options for the number of angle steps in Hexagon Grid.

$N_{\theta_{\text{hex}}}$	Efficiency	Run-time [seconds]
1	0.880086	0.45
20	0.887591	3.56
40	0.887591	6.89
60	0.888549	9.30
80	0.888336	13.00
100	0.889512	16.41
120	0.890344	19.78
140	0.890841	23.20
160	0.888701	28.75
180	0.888864	28.70
200	0.889512	35.36
220	0.890001	40.45
240	0.890344	37.53
260	0.890557	42.03
280	0.890841	50.78
290	0.890964	48.45
300	0.891043	47.00
320	0.889901	58.34
340	0.890137	54.58
360	0.890344	58.95

Table 11: Effect of changing the number of angle steps $N_{\theta_{\text{hex}}}$ on the best layout found and approximate run-time of Hexagon Grid.

3.3. Close Packing

Mosetti et al. and others [31, 20, 49] define the wind farm as a grid and discretize it into large enough cells to comply with the constraint that turbines should not be too close to each other. However, as established by Wang et al. [50], diagonal neighboring positions are too far from each other which causes underutilized site areas.

Close Packing Basic Idea The algorithm starts off with partitioning the wind farm area into cell sizes that can be as small as $0.1 D$, which results in a finer discretization and allows more flexibility in turbine positioning. It generates an *efficiency matrix* to represent possible wake effects among neighboring turbine positions. Then, turbine placement begins in the area corresponding to the wind entrance direction into the field. Close Packing ensures that all placed positions are as close to each other as possible while maintaining relatively high efficiencies among the positions, using the efficiency matrix which was simulated once before the optimization. Only final generated layouts are simulated with the model. This reduces the run-time while still maintaining approximate values of wake effect losses.

3.3.1. Close Packing Implementation Details

Efficiency matrix As alluded to earlier, this matrix represents the possible efficiency outcomes in relation to any two positions. The position at the center of the matrix is the reference position, and the remaining positions represent individually the efficiency outcome from placing only that particular position and the reference position in the field. For example, as shown in Figure 33a, the position in red near (250,0) has an efficiency of about 0.75, which means that the AEP of this and the reference position together is 0.75 of the ideal AEP.

There are two user-defined parameters that control the configuration of this simulated matrix: radius of the matrix M_R and number of simulated positions per side M_N .

Efficiency mesh From the efficiency matrix, the optimizer then interpolates among the position efficiencies to create an even finer resolution, as shown in Figure 33b. This efficiency mesh is then used to “stamp” the efficiency changes around newly placed positions to evaluate their possible wake effects.

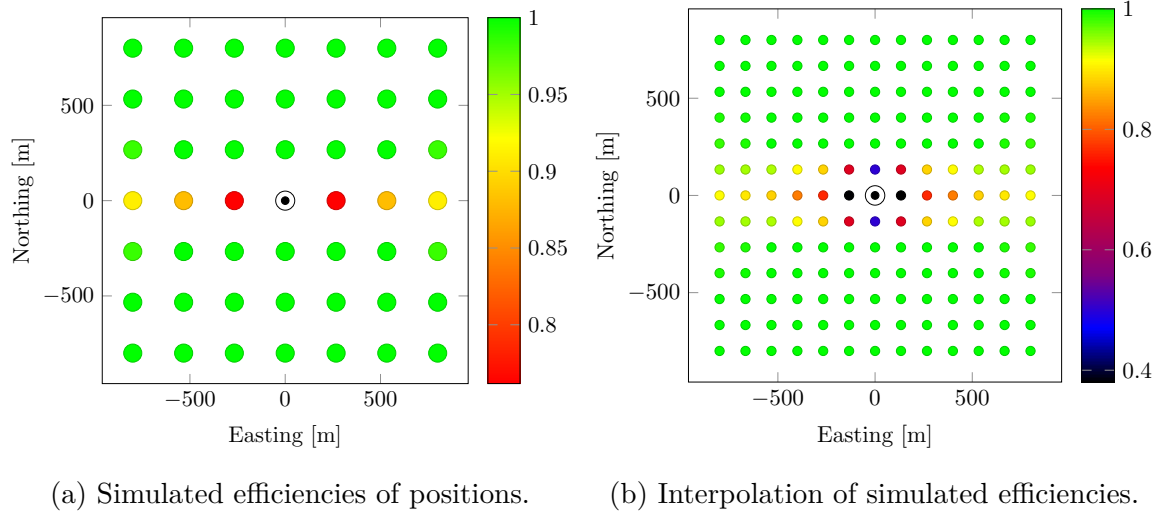


Figure 33: Reference pair-wise simulated efficiencies (Figure 33a) and their interpolation (Figure 33b). For visualization, the matrix in Figure 33a is simulated with an artificial wind distribution such that the main wind entrance direction is the West. Therefore only those positions with the same y -Coordinate as the center reference position cause the efficiency to decrease.

As for the interpolated matrix in Figure 33b, The number of positions per side for the finer efficiency mesh M_n is dictated by a user-defined multiplication factor M_F :

$$M_n = (M_N - 1) \cdot M_F + 1, \quad (32)$$

where M_F in Figure 33b is 2 therefore M_n evaluates to 13 positions.

Field rotation To make the actual optimization simpler, the field is rotated at the beginning such that the main wind *entrance* direction becomes the left, where this direction is defined as:

$$(\theta_{\text{main wind}} + 180^\circ) \mod 360^\circ,$$

with $\theta_{\text{main wind}}$ again being the most probable wind exit direction. This procedure makes iterating over the possible positions straightforward, as shown in Figure 34.

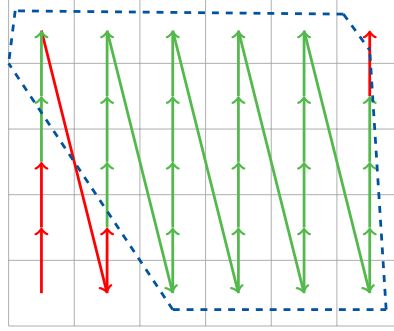


Figure 34: Iteration ordering of Close Packing where the main wind entrance direction is the West (left). The blue-dashed lines represent the wind farm borders, while the gray grid is the discretization applied by Close Packing (large cell sizes are used here for visualization). Steps outside of the valid area are signified by the red arrows, and they are skipped from evaluation as candidate positions by the algorithm.

The algorithm places the first turbine position in the lower-left corner of the field because that is one of the fore-most positions w.r.t. the wind entrance direction, then it begins to iteratively evaluate candidate positions near the first position along the same column of cells, followed by the next column etc. This allows an automatic ordering of the positions by their likelihood of having a high efficiency.

Optimization Using the interpolated efficiency mesh and the iteration approached described earlier, the optimization can begin. Close Packing goes over all candidate positions, and decides to place new positions under the conditions that they:

- Are in a valid area;
- Have efficiency equal to or above a user-defined minimum score threshold $\Omega_{\text{threshold}}$;
- Do not introduce new shading effects on previously placed positions such that they severely decrease their efficiencies.

This is where the efficiency mesh is used: a new position is tentatively placed and it is only kept if it abides by the above conditions.

All possible positions are evaluated until the end of the field is reached or the required amount of turbines with a tolerance of a few more $N_{\text{turbines}} + \epsilon$ is placed.

Different efficiency thresholds As the positions become farther away from the area of wind direction entrance, and more into the center of the field and beyond, the wakes they are affected by can increase. Thus the values in the efficiency mesh may not apply as well to them. Therefore, Close Packing may be used such that each column in the grid can have a different minimum limit on efficiency score $\Omega_{\text{threshold}}$, which is controlled by three user-defined parameters: $\Omega_{\text{threshold}_{\min}}$ and $\Omega_{\text{threshold}_{\max}}$ which are the minimum and maximum thresholds attempted, in addition to $N_{\Omega_{\text{threshold}}}$ which is the number of steps between the attempted values. The minimum and maximum thresholds are constrained by:

$$0 < \Omega_{\text{threshold}_{\min}} \leq \Omega_{\text{threshold}_{\max}} \leq 1$$

These three parameters control another two internal parameters:

$$\begin{aligned} \Omega_{\text{threshold}_{\text{start}}} &\in [\Omega_{\text{threshold}_{\max}}, \Omega_{\text{threshold}_{\min}}] \text{ in steps of } -N_{\Omega} \\ \Omega_{\text{threshold}_{\text{end}}} &\in [\Omega_{\text{threshold}_{\min}}, \Omega_{\text{threshold}_{\max}}] \text{ in steps of } +N_{\Omega} \end{aligned} \quad (33)$$

This allows the acceptance threshold $\Omega_{\text{threshold}}$ to either increase or decrease along the main wind entrance direction.

An overview of the flow of Close Packing is represented by Figures 49 and 50 in Appendix A.2.

3.3.2. Verification of Close Packing based on Circle Packing

Chang and Wang provided a simple proof of the theorem that the optimal pattern for circle packing is hexagonal [10]. Such a packing is shown in Figure 35, where 95 circles are packed densely together.

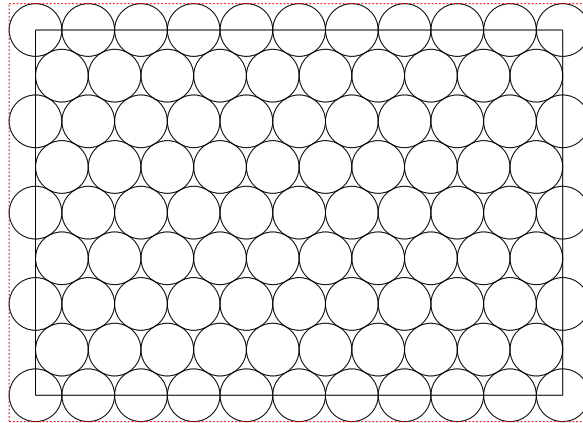


Figure 35: Dense circle packing with 95 circles. All the circle centers are within or at the black-bordered rectangle, while the red-dashed rectangle contains the circles whole.

Each circle has a diameter of $3 D$, which means the width of the black rectangle is $10 \cdot 3 = 30 D$ and its height is $8 \cdot \frac{\sqrt{3}}{2} D \approx 20.8 D$, where $\frac{\sqrt{3}}{2} D$ is the height of the equilateral triangle formed by the centers of three packed circles. Therefore translating this circle packing problem into a Close Packing optimization problem, we begin with a site of dimensions $\approx 20.8 D \cdot 30 D$, set the required number of turbines N_{turbines} to 95, and use an artificial wind distribution where the wind comes equally from all directions, as shown in the efficiencies matrix in Figure 36.

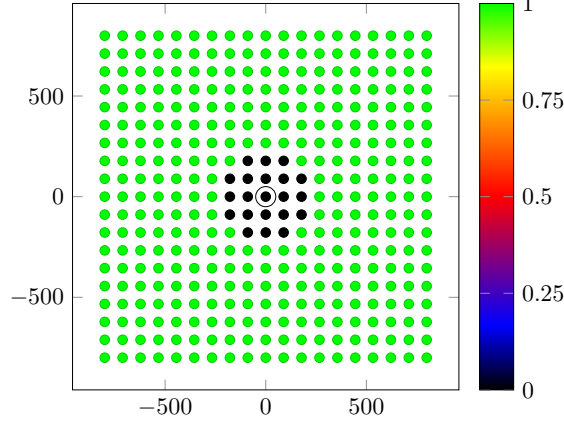


Figure 36: Manually created matrix of efficiencies to mimic the effect of wind equally likely to enter from all directions into the field. All possible positions with a radius of $3 D$ from the center have been assigned with 0 efficiency, so that the positioning adheres to the circle packing constraint of no overlapping.

The result of the optimization is shown in Figure 37. The borders of the site had to be increased by 0.8% to account for floating-point errors which may otherwise cause some positions on the borders to be considered invalid and thus left out. As for the pattern produced, it is indeed hexagonal.

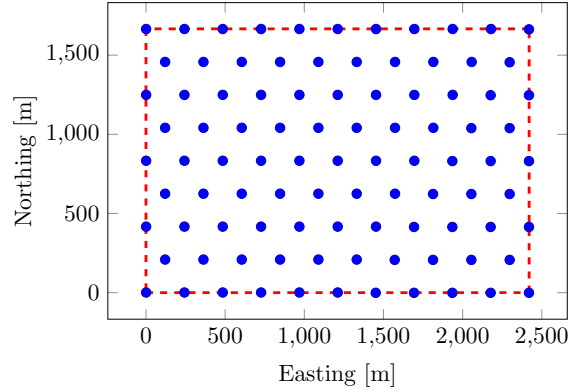


Figure 37: The circle packing problem of fitting 95 circles into an $\approx 20.8 D \cdot 30 D$ rectangle optimized by Close Packing. The red dashed lines are the site borders.

3.3.3. Close Packing parameter studies

As introduced earlier, Close Packing has six main parameters that can heavily influence the optimization results. Testing all the possible combinations of the six parameters could give a more comprehensive overview of the intricacies, but ultimately is too time-consuming. Therefore, in the following we test the two subsets of the parameters separately (threshold limits and mesh settings), while keeping the other subset of parameters fixed.

Score threshold limits The minimum and maximum accepted efficiency scores are defined by $\Omega_{\text{threshold}_{\max}}$ and $\Omega_{\text{threshold}_{\min}}$ respectively. The number of steps between the minimum and maximum thresholds is dictated by $N_{\Omega_{\text{threshold}}}$, which has the heaviest influence on the optimization: If we have $\Omega_{\text{threshold}_{\min}} = 0.9$ and $\Omega_{\text{threshold}_{\max}} = 1$ with number of steps $N_{\Omega_{\text{threshold}}} = 10$, then the step size of $\Omega_{\text{threshold}}$ will be 0.01. We can also have the same step size using $\Omega_{\text{threshold}_{\min}} = 0.8$, $\Omega_{\text{threshold}_{\max}} = 1$, and $N_{\Omega_{\text{threshold}}} = 20$. The former setting produces a search space that is a subset of the latter setting:

Setting 1 : $\Omega_{\text{threshold}} \in [0.9, 1]$ in steps of 0.01
 Setting 2 : $\Omega_{\text{threshold}} \in [0.8, 1]$ in steps of 0.01
 $\rightarrow$ Setting 1 $\subset$ Setting 2

Therefore, we fix the threshold limits and choose a wide range with $\Omega_{\text{threshold}_{\min}} = 0.75$ and $\Omega_{\text{threshold}_{\max}} = 1$, while varying the number of steps between them $N_{\Omega_{\text{threshold}}}$. This test is performed again on the Horns Rev 1 wind farm settings, and the results showing the real final efficiencies are plotted in Figure 38. It shows that the best efficiencies found with $N_{\Omega_{\text{threshold}}} \leq 5$ are relatively low in the range from 0.7 to 0.85. The efficiencies begin to converge beyond that point giving values around 0.875, with the maximum efficiency out of all the attempted threshold steps (with a small margin) at $N_{\Omega_{\text{threshold}}} = 50$, which gave a final efficiency of 90.2485%. This number of steps translates into a very fine step size of $(1 - 0.75)/50 = 0.005$ for the possible thresholds bounds $\Omega_{\text{threshold}_{\text{start}}}$ and $\Omega_{\text{threshold}_{\text{end}}}$.

It is expected that a finer step size would lead to larger search space and a higher chance of finding a better solution. Furthermore, it is possible that a higher value than $N_{\Omega_{\text{threshold}}} = 50$ would be unnecessary because of the nature of Close-Packing where for the main part of the optimization uses approximated efficiencies.

To get a clearer idea of what an optimization with $N_{\Omega_{\text{threshold}}} = 50$ entails, Figure 39 shows the simulated efficiencies from all the possible combinations of $\Omega_{\text{threshold}_{\text{start}}}$ and $\Omega_{\text{threshold}_{\text{end}}}$ that were attempted in this optimization run.

The most visible aspect of Figure 39 at first glance is the large area of effectively equal efficiencies for $\Omega_{\text{threshold}_{\text{start}}}$ values around 0.9. That is because for most of these threshold bounds, the threshold is too low and Close Packing is able to place positions extremely close together, as shown in Figure 40. Therefore, the efficiency of such layouts is also low at about 0.5.

However, having the threshold bounds too high will result in the inability to place the required amount of turbines, as indicated by the black area in the top-right corner

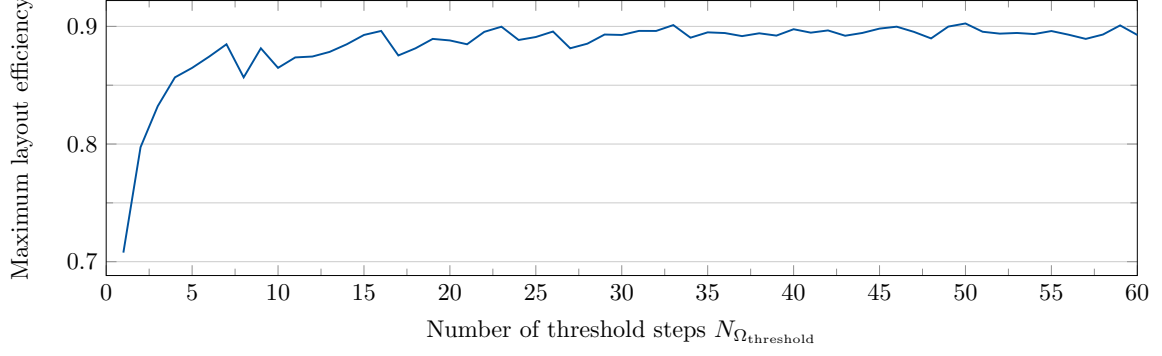


Figure 38: Effect of number of threshold steps $N_{\Omega_{\text{threshold}}}$ on layout score.

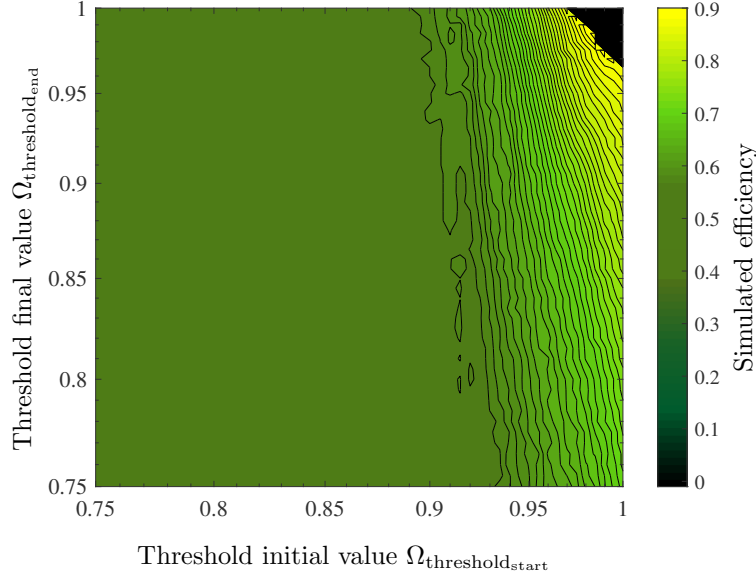


Figure 39: Effect of threshold bounds on layout score. The large uniform area is caused by the semi-equal scores of semi-identical layouts, which are in turn due to relatively low score thresholds.

of Figure 39. Other than the extremes of the threshold initial value $\Omega_{\text{threshold_start}}$ and final value $\Omega_{\text{threshold_end}}$, it is clear that the highest efficiencies are achieved when $\Omega_{\text{threshold_start}} \geq \Omega_{\text{threshold_end}}$, but not when $\Omega_{\text{threshold_start}} < \Omega_{\text{threshold_end}}$. To examine why, we look at specific values for these bounds.

The threshold bounds for the setting which resulted in the highest score of 90.2485% with number of threshold steps $N_{\Omega_{\text{threshold}}} = 50$, were simply 0.985 for both $\Omega_{\text{threshold_start}}$ and $\Omega_{\text{threshold_end}}$, resulting in a constant threshold for all the positions in the field.

The second-best score found was at $N_{\Omega_{\text{threshold}}} = 33$ which gave a final efficiency of 90.1108%, and it also was achieved with a setting where $\Omega_{\text{threshold_start}} = \Omega_{\text{threshold_end}}$.

This seems to suggest that changing the threshold in the way described does not lead to the optimal results. However, in the addition to the pattern observed in Figure 39

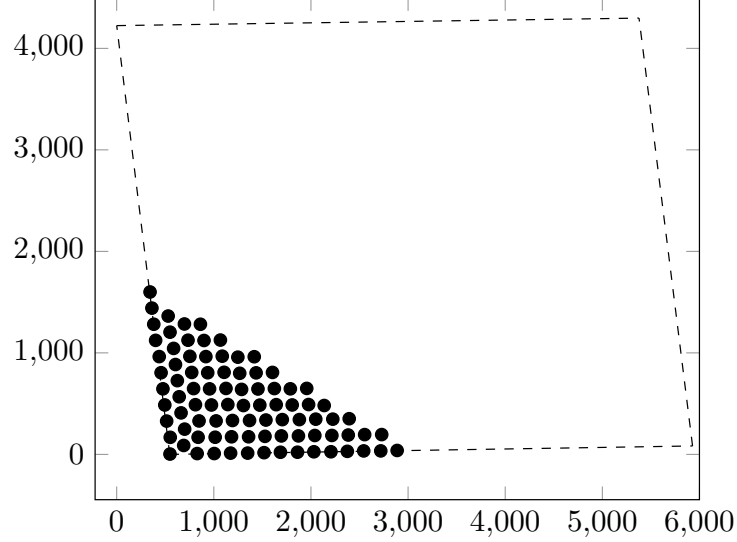


Figure 40: Turbine positioning with a relatively low score threshold $N_{\Omega_{\text{threshold}}}$ of 0.8 causes the positions to be only far apart enough to abide by the minimum distance constraint. The dashed lines are the site borders, which indicate how much space is not being utilized.

that leans towards a higher $\Omega_{\text{threshold}_{\text{start}}}$, the lead of the two best settings was not by a large gap from the other settings: The third best score was produced with a number of steps $N_{\Omega_{\text{threshold}}} = 59$ which resulted in a final efficiency of 90.0836%, using threshold bounds $\Omega_{\text{threshold}_{\text{start}}} = 0.991525$ and $\Omega_{\text{threshold}_{\text{end}}} = 0.974576$. This leads to the efficiency gradually decreasing away from the main wind entrance direction into the wind farm area. Having a higher efficiency threshold in the front of the field w.r.t. the wind entrance direction could place higher restrictions on turbine positioning such that they are farther apart, thus have higher efficiencies.

Taking all these results into account, a good layout is likely to be found with high threshold bounds. Therefore it is recommended to use a minimum threshold value of $\Omega_{\text{threshold}_{\text{min}}} = 0.95$ and a maximum of $\Omega_{\text{threshold}_{\text{max}}} = 1$, with number of steps $N_{\Omega_{\text{threshold}}} = 10$.

3.4. Spiral Grid

The final algorithm (not developed, but utilized in the context of this thesis) is inspired from the approach used by Noone et al. [33] to optimize heliostat field layouts, which is in turn inspired by the spirals of seed heads in sunflowers. In a way similar to Close Packing, Spiral Grid is also a pattern-based optimization that relies on the degree of closeness of its positions. It depends on two parameters: scaling or “zoom” factor a , and density factor b . For any setting of a and b , a set of positions can be generated

where the k^{th} position is defined by polar coordinates:

$$\alpha_k = 2\pi\varphi_g^{-2}k \quad (34)$$

$$r_k = ak^b, \quad (35)$$

where $\varphi_g = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Even though its original use-case was for heliostat positioning, manipulating the scaling a and density b parameters of Spiral Grid can make this approach fit for wind farms as well. Figure 41 shows an optimized layout with efficiency score 90.36%. It was achieved with density $b = 0.45$, scaling factor $a = 400.4$,

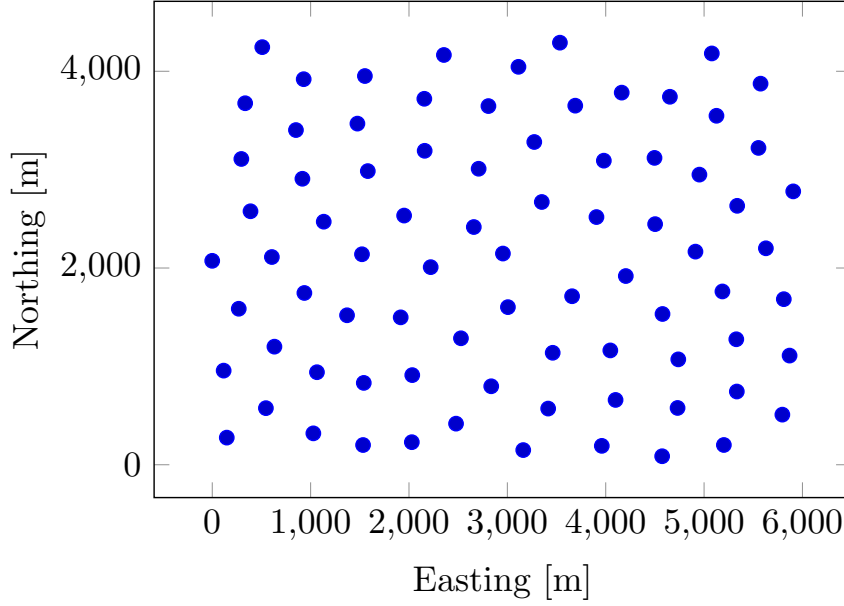


Figure 41: Layout optimized with Spiral Grid using density factor $b = 0.45$ and scaling factor $a = 400.4$.

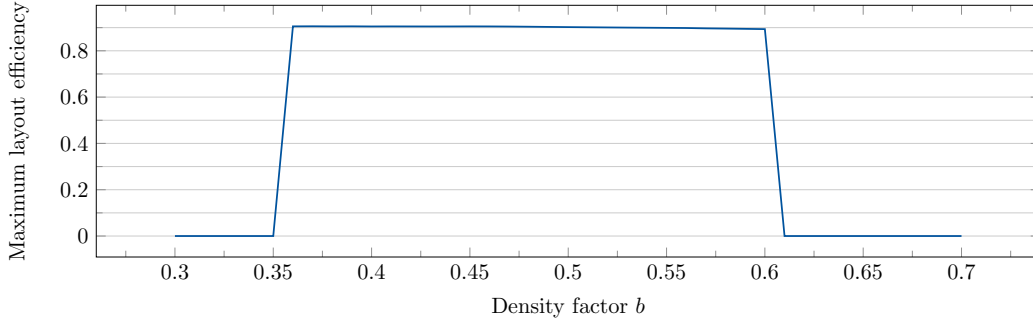
3.4.1. Spiral Grid Parameter Study

In this optimization, only the density parameter b is iterated within a for-loop while the scaling factor a is optimized using the Bisection method similarly to the approaches of optimizing side lengths in Square Grid and Hexagon Grid.

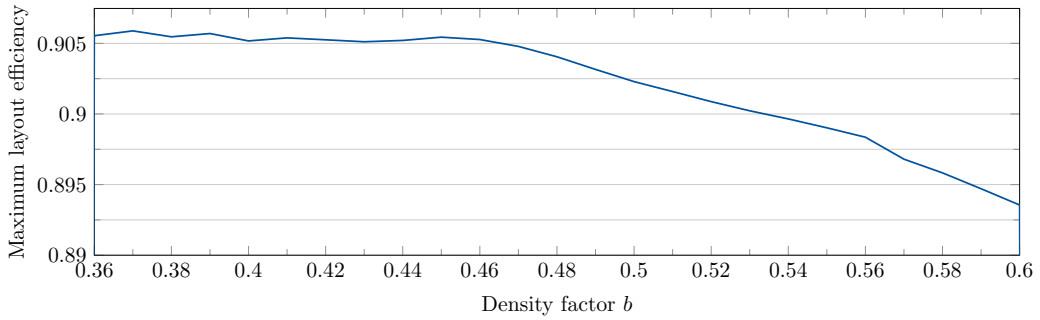
Therefore we study the effect of varying the density b on the layout score. As for the range of b , Noone et al. used $[0.5, 0.7]$ because for $b = 0.5$ the distance between the spirals is constant, and increasing the density b causes the distances to gradually increase away from origin of the field, which can be beneficial for heliostats to avoid blocking and shading as they become farther away from the receiver tower.

However, for wind farms, we would like to see the effect of decreasing the distances between the spirals, especially because there is no constraint on being close to a tower.

Thus, we optimize for Horns Rev 1 with Spiral Grid as shown in the results from Figure 42. The density parameter b was in the range $[0.3, 0.7]$ in steps of 0.1. It can be seen from Figure 42a that for $b < 0.36$ and $b > 0.6$, the efficiency is 0, because the generated layouts include too many or too few positions. Figure 42b shows the advantage of a lower density factor b in achieving higher efficiencies, at least for this scenario. This coincides with the results for the Square Grid growth parameter ξ_{growth} in Section 3.1.2, where it was shown that turbines towards the center of the field benefit from having increased distances among each other. Therefore, it is recommended to let the search space for density factor b be in $[0.35, 0.6]$.



- (a) The range from 0.3 to 0.35 and 0.61 to 0.7 for density factor b results in too many or too few turbine positions. Thus it is scored 0.



- (b) Taking a closer look at the valid range for b from 0.36 to 0.6 shows that for lower values of the density factor, higher efficiencies are achieved.

Figure 42: Effect of varying the density factor b on layout efficiency in Spiral Grid.

3.5. Refinements with Local Search

The second step in the multi-step optimizer is the local search heuristics improvements, which the simulated annealing (SA) algorithm will be used for, as briefly introduced in Section 1.2.

The SA algorithm can perturb positions by three operations:

1. Add: which adds a new turbine to the set

2. Remove: removes a turbine from the set
3. Move: simply moves an existing turbine into a new location

Each operation has a specific temperature t which affects how likely the operation change is to be accepted, and a cooling factor cf which increases/decreases the temperature t through-out the iterations of the SA algorithm.

3.5.1. SA Parameter Study

Others have recommended [14] specific default values for these parameters, shown in Table 12.

Parameter	Recommended Value
t_{add}	0.3
t_{remove}	0.002
t_{move}	0.003
cf_{add}	0.9
cf_{remove}	0.999
cf_{move}	0.8

Table 12: SA default parameters.

Although it would be ideal to examine the relation of all three parameters to each other, due to time constraints, only the move operation parameters t_{move} and cf_{move} . The simple reasoning is that a move operation is somewhat equivalent to a remove operation followed by an add operation, therefore it may be the most comprehensive operation of all three.

We use the same initial values recommended [14] in addition to a few values above and below. Therefore, $t_{\text{move}} \in [0.001, 0.003, 0.006, 0.009]$ and $cf_{\text{move}} \in [0.2, 0.35, 0.5, 0.8, 0.95]$. All in all it makes 24 combinations of settings, and this was tested on the RWTH cluster with 24 threads and 12 nodes, such that each combination would be a test run on a specific thread. Each run of the 24 lasted for about 6 hours individually, and Figure 43 shows the final best scores achieved by the different combinations.

For $t_{\text{move}} = 0.003$ and $cf_{\text{move}} = 0.65$ the efficiency is highest, at 93.1317%. We plot the intermediate efficiencies per iteration of this specific run in Figure 44. There is a significant jump in efficiency after 6 iterations, and the score starts to converge around 12 iterations.

In comparison, the previously recommended cooling factor $cf = 0.8$ (with $t_{\text{move}} = 0.003$) produced a different pattern of efficiencies in its run as shown in Figure 45. Even though the score is lower than with using $cf = 0.65$, the efficiencies have not shown convergence which means they may increase more. However, given that $cf = 0.65$ reached a higher score in the same amount of time (6 hours), we take this as a new rough estimate for the cooling factor value of the move operation, and use it in the feasibility tests.

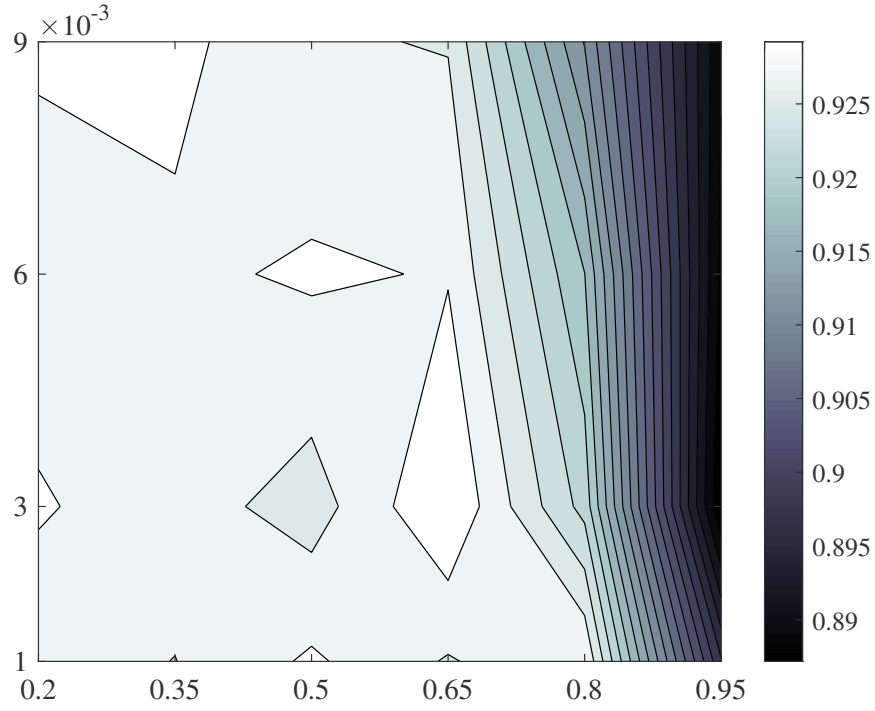


Figure 43: Efficiencies of different SA settings.

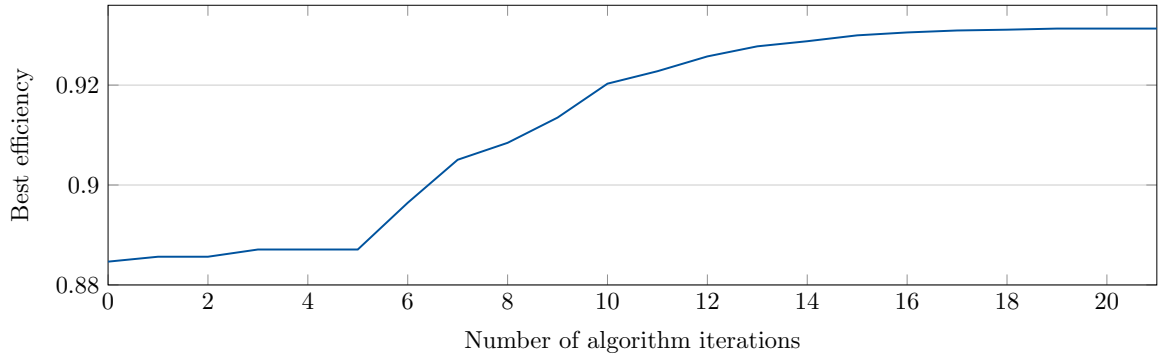


Figure 44: Intermediate efficiencies of one SA run with settings $t_{\text{move}} = 0.003$ and $cf_{\text{move}} = 0.65$.

3.6. Optimizer conclusion

In the context of this thesis, three practical pattern-optimizations have been developed and integrated into a multi-step optimizer that ends with local search improvements. The optimizers are capable of:

- Generating layouts with $> 88\%$ AEP efficiency within five minutes on a standard computer utilizing one thread.
- Catering to real world commercial constraints such as having the turbines in

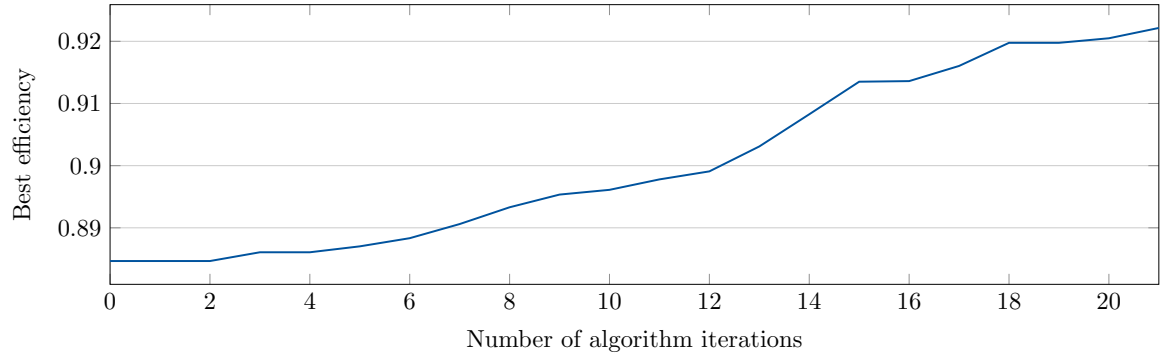


Figure 45: Intermediate efficiencies of one SA run with settings $t_{\text{move}} = 0.003$ and $cf_{\text{move}} = 0.8$.

semi-regular grids (Square Grid and Hexagon Grid), or exploiting the available wind farm area as much as possible (Close Packing).

- Abiding by geographical constraints such as valid or restricted areas for turbine placement, that may be defined by concave irregular polygons.

4. Feasibility Study

In this section we use the algorithms described earlier in the multi-step optimization, for the wind farm Horns Rev 1 in the North Sea. The real layout of these wind farm is shown in Figure 46.

4.1. Efficiencies after optimization with patterns and local search

The efficiencies achieved with each pattern and also after applying the SA algorithm with the Horns Rev 1 farm are shown in Table 13. All local search algorithms ran for about 5 hours. The settings used in the patterns were the recommended settings derived earlier.

4.2. Improvements from local search

These tests were run on the RWTH cluster for around 5 hours per optimization pattern. Applying the simulated annealing algorithm to the optimized layouts introduced some additional improvements, as shown in Figure 47. We can see that both Square Grid and Close Packing converge toward a higher number of iterations longer than Hexagon Grid. It is not clear why this happens. The file output with intermediate scores for Spiral Grid was not produced by the cluster and thus the convergence for Spiral Grid could not be determined. However, we know that starting efficiency for local search with Spiral Grid was 89.87%, and in the end it became 92.88%. Spiral Grid, Hexagon Grid and Close Packing all received about 4% increases in efficiencies as seen

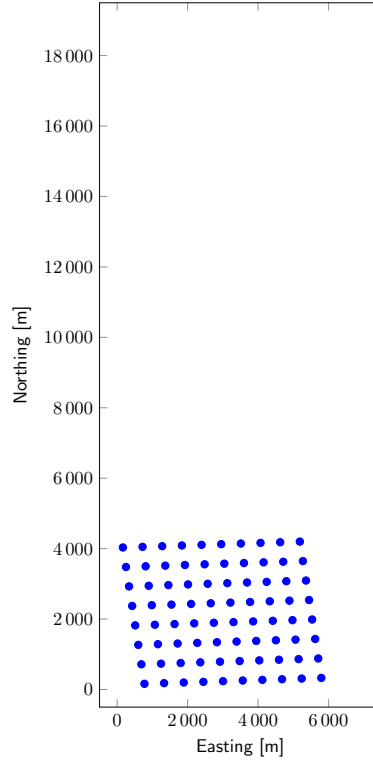


Figure 46: Layout of Horns Rev 1

Optimization	Efficiency [%]
None	88.80
Hexagon Grid	89.03
Square Grid	91.47
Close-Packing	90.18
Spiral Grid	89.86
Square Grid + SA	92.46
Hexagon Grid + SA	92.72
Close-Packing + SA	92.92
Spiral Grid + SA	92.88

Table 13: Efficiencies of Horns Rev 1 before optimization, and after optimizing with patterns alone then with Simulated Annealing.

in Figure 47. Even though Square Grid did not achieve as much improvement, it could be because the starting efficiency of Square Grid was the largest, which may suggest that the initial efficiency of a layout can affect the behavior of the local search algorithm. From the results, it is implied that if a local search heuristic is applied, then Close Packing is the best algorithm. Otherwise, Square Grid has the best initial score and it is recommended if run time is a constraint and does not allow extra hours

for local search.

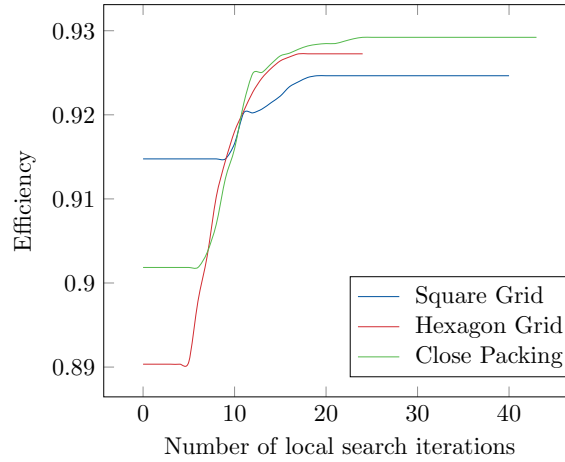


Figure 47: Improvement of efficiencies with local search for the Horns Rev 1 wind farm.

5. Conclusion

In this thesis we presented a validated model with at most 0.0004% error from the reference outputs by OpenWind. Additionally, we described the components of a multi-step optimizer consisting of pattern optimizations that can generate turbine layouts with 90% efficiency in less than five minutes, in addition to a local search step which was able to increase the efficiencies by about 1-4%, depending on the initial layout and site.

5.1. Outlook

During the work on this thesis, there were some aspects that may be worth exploring further:

Bins vs Weibull distributions Section 2.1.2 described the differences between using frequency tables (bins) and Weibull distributions to model wind direction and speed. Although WindFlower was validated using both formats, where the wind modes were only compared against the respective wind modes in OpenWind, it can also be interesting to compare bins and Weibull distributions against each other. This may be done by taking very fine direction and speed steps then simulating in bin mode and Weibull distribution mode. It may be the wind mode versions will not give equal results, unless the speed step size is sufficiently small.

Validation The model can be further validated against other commercial software in addition to OpenWind, or even real wind power plant data. Furthermore, other

models can be used besides Park as Beaucage et al. [7] show that Park may undermine the effects of wakes, and that Deep-Array Wake Model captures the wake effect more accurately. Although they only simulated one directional sector, therefore the results are inconclusive.

Incorporating a genetic algorithm All in all, the optimizers have shown that they can give layouts with about 90% efficiency in under five minutes of run-time. Genetic algorithms on the other hand are known to be more powerful yet take more processing time. However, typical offshore wind farms have life-spans of around 25 years [42], so it may be worth it to spend an extra month in run-time at the beginning to get the very best layout possible. Therefore we recommend the validated model and developed optimization patterns in this thesis as a strong basis for more complex and powerful algorithms.

Our population and our use of the finite resources of planet Earth are growing exponentially, along with our technical ability to change the environment for good or ill.

STEPHEN HAWKING

A. Appendix

A.1. Validation supplementary figures

A.2. Close Packing supplementary figures

The algorithm flow of Close Packing as described in Section 3.3.1 is abstracted and shown in Figure 50 and Figure 49.

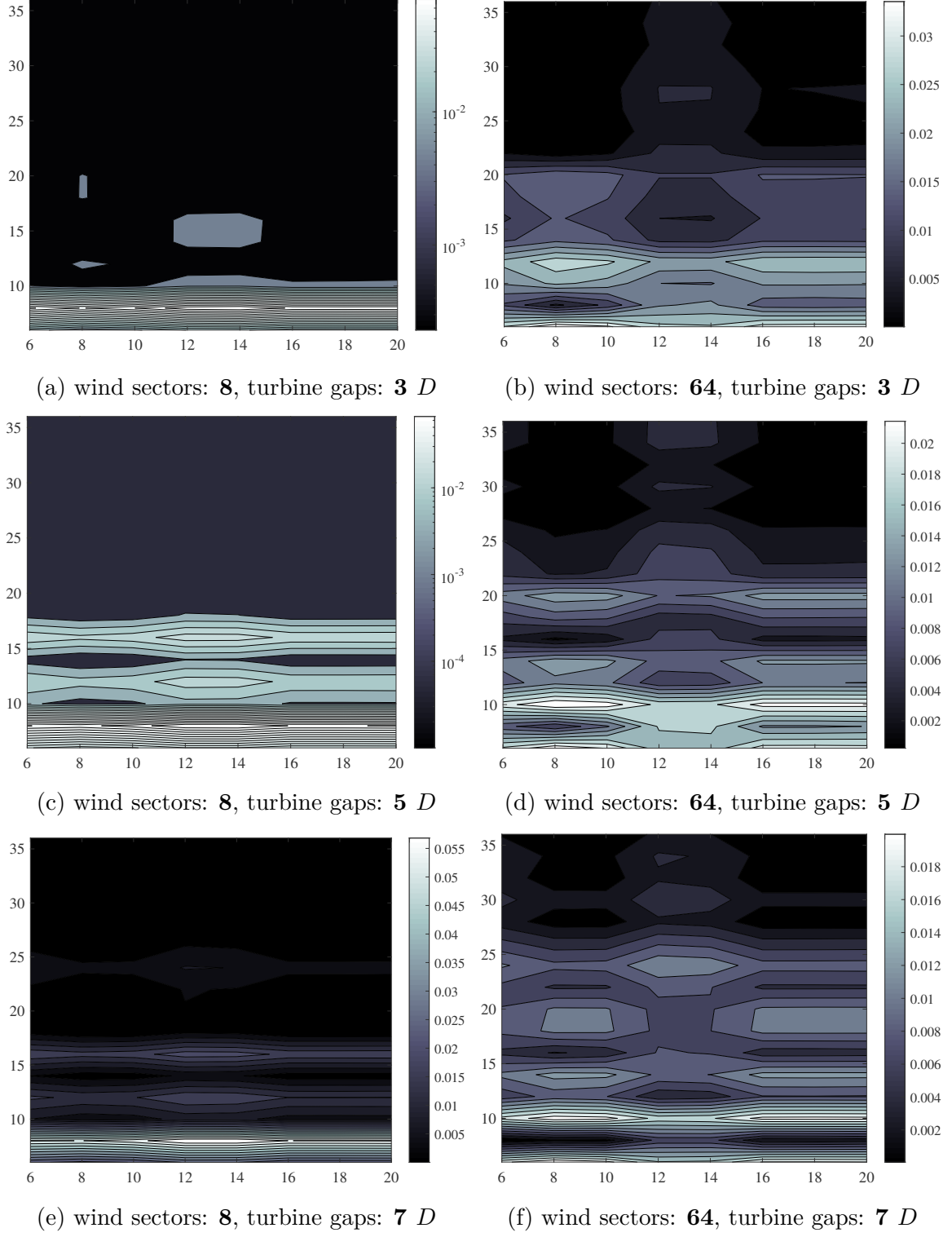


Figure 48: Effect of varying the number of simulated wind direction and wind speed steps on the relative error of the AEP output, for cases with different wind sectors and turbine-separation distances.

X-axis = Number of simulated wind speeds u_n

Y-axis = Number of simulated wind directions φ_n

Colorbar = Relative error

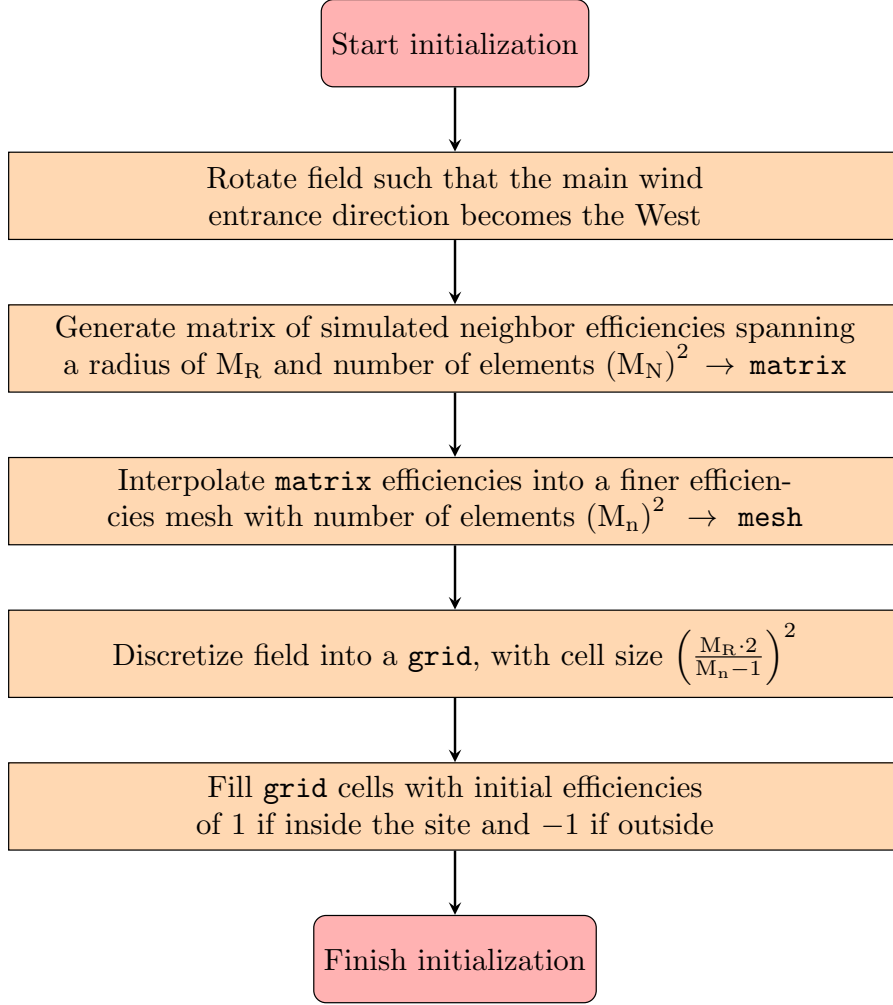


Figure 49: Flow of Close Packing initialization procedures.

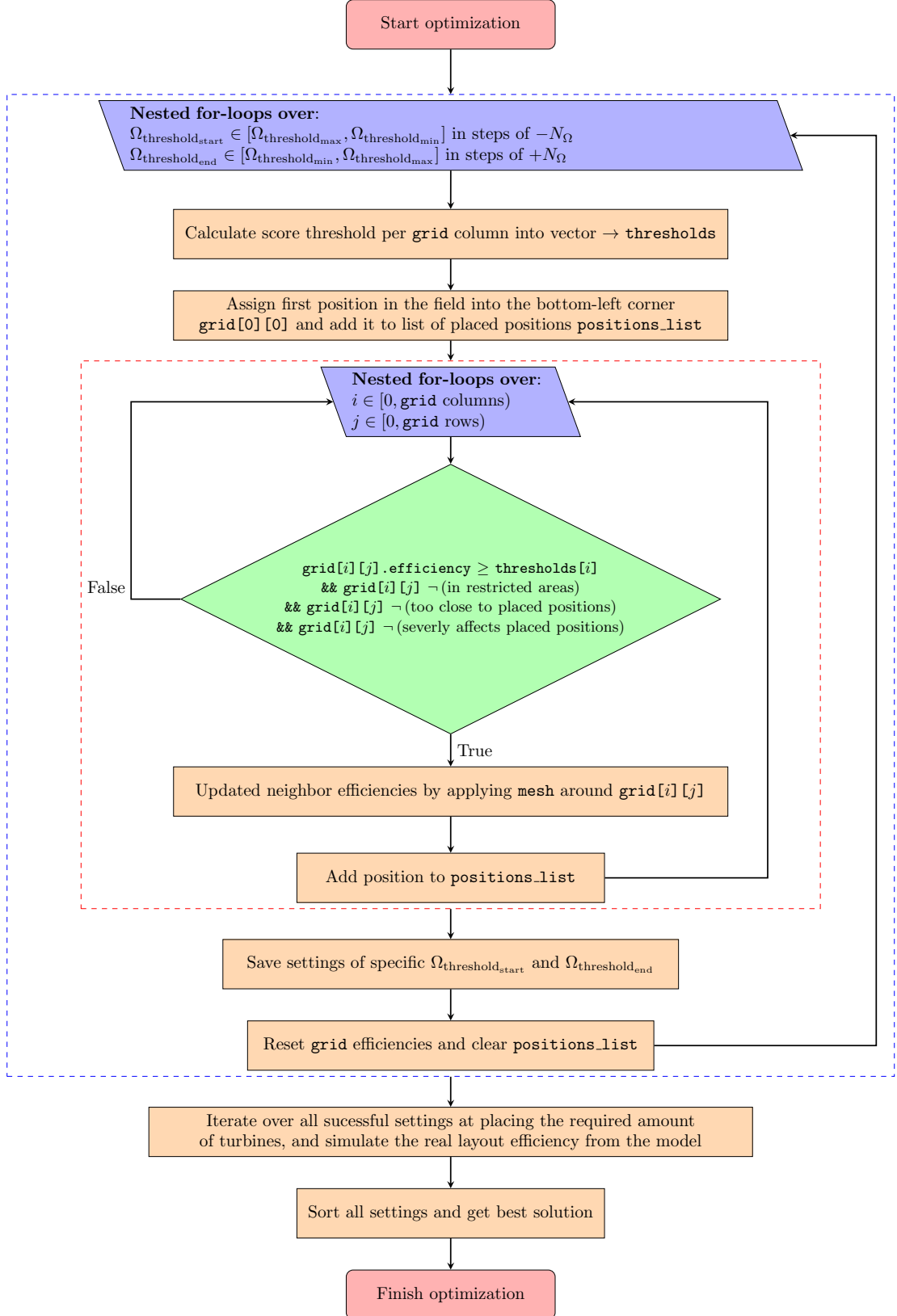


Figure 50: Close-Packing optimization flow, which consists of four nested for-loops: The first pair iterate over possible threshold limits for $\Omega_{\text{threshold}}$, and the inner loops simply iterate over the rectangular grid.

References

- [1] J. F. Ainslie. Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1):213–224, 1988.
- [2] Idriss Ammara, Christophe Leclerc, and Christian Masson. A viscous three-dimensional differential/actuator-disk method for the aerodynamic analysis of wind farms. *Journal of Solar Energy Engineering*, 124(4):345–356, 2002.
- [3] J. C. Bansal and P. Farswan. Wind farm layout using biogeography based optimization. *Renewable Energy*, 107:386 – 402, 2017. ISSN 0960-1481.
- [4] J. C. Bansal, P. Farswan, and A. K. Nagar. Design of wind farm layout with non-uniform turbines using fitness difference based bbo. *Engineering Applications of Artificial Intelligence*, 71:45 – 59, 2018. ISSN 0952-1976.
- [5] R. J. Barthelmie, G. C. Larsen, S. T. Frandsen, L. Folkerts, K. Rados, S. C. Pryor, B. Lange, and G. Schepers. Comparison of wake model simulations with offshore wind turbine wake profiles measured by sodar. *Journal of Atmospheric and Oceanic Technology*, 23(7):888–901, 2006. doi: 10.1175/JTECH1886.1.
- [6] R. J. Barthelmie, S. C. Pryor, S. T. Frandsen, K. S. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. *Journal of Atmospheric and Oceanic Technology*, 27(8):1302–1317, 2010. doi: 10.1175/2010JTECHA1398.1. URL <https://doi.org/10.1175/2010JTECHA1398.1>.
- [7] Philippe Beaucage, Michael Brower, Nick Robinson, and Chuck Alonge. Overview of six commercial and research wake models for large offshore wind farms. *Proceedings of the European Wind Energy Associate (EWEA)*, 2012.
- [8] Martin Bilbao and Enrique Alba. Simulated annealing for optimization of wind farm annual profit. In *Logistics and Industrial Informatics, 2009. LINDI 2009. 2nd International*, pages 1–5. IEEE, 2009.
- [9] Michael C Brower and Nicholas M Robinson. The openwind deep-array wake model: development and validation. *AWS Truepower*, 2012.
- [10] Hai-Chau Chang and Lih-Chung Wang. A simple proof of thue’s theorem on circle packing. *arXiv preprint arXiv:1009.4322*, 2010.
- [11] H Charnock. Wind stress on a water surface. *Quarterly Journal of the Royal Meteorological Society*, 81(350):639–640, 1955.
- [12] J. Choi and M. Shan. Advancement of jensen (park) wake model. In *Proceedings of the European Wind Energy Conference and Exhibition*, pages 1–8, 2013.

- [13] Christian Steiness. Picture of hornsrev1 wake effects. <https://imgur.com/qruVcnu>. Accessed: 21.01.2019.
- [14] A. Drichel, B. Loup, and L. Franke. Multi-objective layout optimization of offshore wind farms, 2017. Unpublished.
- [15] M Dolores Esteban, J Javier Diez, Jose S López, and Vicente Negro. Why offshore wind energy? *Renewable Energy*, 36(2):444–450, 2011.
- [16] Patrik Fagerfjäll. Optimizing wind farm layout: more bang for the buck using mixed integer linear programming. *Chalmers University of Technology and Gothenburg University*, 2010.
- [17] Sten Tronæs Frandsen et al. *Turbulence and turbulence-generated structural loading in wind turbine clusters*. PhD thesis, Risø National Laboratory, 2007.
- [18] X. Gao, H. Yang, and L. Lu. Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model. *Applied Energy*, 174:192 – 200, 2016. ISSN 0306-2619.
- [19] J. S. Gonzalez, A. G. G. Rodriguez, J. C. Mora, J. R. Santos, and M. B. Payan. A new tool for wind farm optimal design. In *2009 IEEE Bucharest PowerTech*, pages 1–7, June 2009. doi: 10.1109/PTC.2009.5281977.
- [20] SA Grady, MY Hussaini, and Makola M Abdullah. Placement of wind turbines using genetic algorithms. *Renewable energy*, 30(2):259–270, 2005.
- [21] Hans Hillewaert. Picture of a wind turbine in thorntonbank. <https://www.flickr.com/photos/bathyporeia/14088768866>. Accessed: 21.01.2019.
- [22] G. Heiming. Modeling and simulation of offshore wind farms. Bachelor thesis, RWTH Aachen University, 2015.
- [23] J. F. Herbert-Acero, O. Probst, P.-E. Réthoré, G. C. Larsen, and K. K. Castillo-Villar. A review of methodological approaches for the design and optimization of wind farms. *Energies*, 7:6930–7016, 2014.
- [24] John D Holmes. *Wind loading of structures*. CRC press, 2015.
- [25] N. O. Jensen. *A Note on Wind Generator Interaction*. Number 2411 in Risø-M. Risø National Laboratory, Roskilde, 1983.
- [26] I. Katic, J. Højstrup, and N. O. Jensen. A simple model for cluster efficiency. In *European wind energy association conference and exhibition*, pages 407–410, 1986.
- [27] Soren Krohn, Poul-Erik Morthorst, and Shimon Awerbuch. The economics of wind energy. 2009.

- [28] M. A. Lackner and C. N. Elkinton. An analytical framework for offshore wind farm layout optimization. *Wind Engineering*, 31(1):17–31, 2007. doi: 10.1260/030952407780811401. URL <https://doi.org/10.1260/030952407780811401>. Accessed: 02.05.2018.
- [29] Bernhard Lange, Hans-Peter Waldl, Algert Gil Guerrero, Detlev Heinemann, and Rebecca J Barthelmie. Modelling of offshore wind turbine wakes with the wind farm program flap. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 6(1):87–104, 2003.
- [30] N. G. Mortensen, D.N. Heathfield, O. Rathmann, and M. Nielsen. Wind atlas analysis and application program: Wasp 11 help facility, 2014.
- [31] G. Mosetti, C. Poloni, and B. Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51(1):105–116, 1994.
- [32] A Neubert, A Shah, and W Schlez. Maximum yield from symmetrical wind farm layouts. In *Proceedings of DEWEK*, 2010.
- [33] Corey J Noone, Manuel Torrilhon, and Alexander Mitsos. Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy*, 86(2):792–803, 2012.
- [34] U Aytun Ozturk and Bryan A Norman. Heuristic methods for wind energy conversion system positioning. *Electric Power Systems Research*, 70(3):179–185, 2004.
- [35] L. Parada, C. Herrera, P. Flores, and V. Parada. Wind farm layout optimization using a gaussian-based wake model. *Renewable Energy*, 107:531 – 541, 2017. ISSN 0960-1481.
- [36] A. C. Pillai, J. Chick, M. Khorasanchi, S. Barbouchi, and L. Johanning. Application of an offshore wind farm layout optimization methodology at middelgrunden wind farm. *Ocean Engineering*, 139:287 – 297, 2017. ISSN 0029-8018.
- [37] P. Richter, J. Wolters, R. Cakar, A. Verhoeven-Mrosek, and M. Frank. Uncertainty quantification of offshore wind farms. John Wiley & Sons, 2019.
- [38] R. A. Rivas, J. Clausen, K. Hansen, and L. Jensen. Solving the turbine positioning problem for large offshore wind farms by simulated annealing. *Wind Engineering*, 33(3):287–297, 2009.
- [39] M. Samorani. The wind farm layout optimization problem. In *Handbook of Wind Power Systems*, pages 21–38. Springer, 2013.
- [40] R. Shakoor, M. Y. Hassan, A. Raheem, and Y.-K. Wu. Wake effect modeling: A review of wind farm layout optimization using jensen’s model. *Renewable and Sustainable Energy Reviews*, 58:1048 – 1059, 2016. ISSN 1364-0321.

- [41] D. Simon. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6):702–713, Dec 2008. ISSN 1089-778X. doi: 10.1109/TEVC.2008.919004.
- [42] Bodil Skousen and Diletta Zonta. Offshore wind energy.
- [43] RJ Swart, C Coppens, H Gordijn, M Piek, P Ruysenaars, JJ Schrande, P de Smet, M Hoogwijk, M Papalexandrou, E de Visser, et al. Europe’s on-shore and offshore wind energy potential: An assessment of environmental and economic constraints. Technical report, European Environment Agency, 2009.
- [44] C Szafron. Offshore windfarm layout optimization. In *Environment and Electrical Engineering (EEEIC), 2010 9th International Conference on*, pages 542–545. IEEE, 2010.
- [45] Angelo Tesauro, Pierre-Elouan Réthoré, and Gunner Chr Larsen. State of the art of wind farm optimization. *Proceedings of EWEA*, pages 1–11, 2012.
- [46] AWS Truepower. Openwind theoretical basis and validation. *Albany, NY, Technical Report*, 2010.
- [47] Charlie Vanaret, Nicolas Durand, and Jean-Marc Alliot. Windmill farm pattern optimization using evolutionary algorithms. 07 2014. doi: 10.1145/2598394.2598506.
- [48] C. Wan, J. Wang, G. Yang, and X. Zhang. Optimal micro-siting of wind farms by particle swarm optimization. In *Advances in Swarm Intelligence*, pages 198–205. Springer, 2010.
- [49] Chunqiu Wan, Jun Wang, Geng Yang, Xiaolan Li, and Xing Zhang. Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5092–5096. IEEE, 2009.
- [50] Jun Wang, Xiaolan Li, and Xing Zhang. Genetic optimal micro-siting of wind farms by equilateral-triangle mesh. In *Wind Turbines*. InTech, 2011.