TU Delft

GREEDY WIND FARM LAYOUT OPTIMIZATION USING PRE-AVERAGED LOSSES

FEDERICO TILLI

Master of Science Thesis

Greedy Wind Farm Layout Optimization Using Pre-Averaged Losses

by

Federico Tilli

to obtain the degree of Master of Science in
Sustainable Energy Technology
at Delft University of Technology.

Student number:	4742621	
Thesis committee:	Dr. ir. E. Quaeghebeur,	TU Delft, supervisor
	Prof. dr. S. J. Watson,	TU Delft, chairman
	Dr. ir. J. T. van Essen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Summary

Wind turbine placement in a wind farm can be optimized to limit power losses due to wakes and improve the economic value of the plant. Seeing as wind farms are increasing in number and size, fast methods to generate good quality layouts can be beneficial for designers. Wind Farm Layout Optimization (WFLO) consists in finding a layout that maximizes the Annual Energy Production (AEP) of a wind farm. The procedure is driven by an algorithm that generates possible layouts and by a framework that evaluates their AEP. If the traditional method to assess AEP is adopted, layout optimization is computationally demanding due to the impact of each turbine's position on the productivity of the surrounding ones. Indeed, for any change in the layout, this inter-dependency forces designers to calculate wake effects and the wind resource-average energy production of the wind farm.

This thesis proposes an approach to reduce the computational load of WFLO by pre-computing the power losses. Indeed, the approach avoids recalculating the expected power loss among turbines during the optimization procedure. This optimization strategy employs a novel approach, called Pre-Averaged Model (PAM), that expresses the expected power loss of a wake source at representative points around it. Firstly, the wind farm is discretized, and fictitious turbines are placed at each given spot. Secondly, PAM calculates the expected power loss caused by each fictitious turbine for the surrounding ones. Discontinuities introduced by wind resource discretization and top-hat wake deficit profiles substantially affect PAM's accuracy. Binning wind measurements in 72 wind directions solves the problem for typical engineering wake models. Then, a greedy algorithm uses the power losses of the fictitious turbines to build layouts constructively by adding an extra turbine per iteration. The effect of multiple wake sources on a wake target is modelled by linear superimposition of the pre-computed power losses. PAM is tested in combination with three greedy algorithms, namely, Basic Greedy (BG), Add-Remove-Move Greedy (ADREMOG), and ADREMOG II.

This research demonstrates that the PAM and the superposition of the power losses can be reliably used for WFLO. Also, the joint use of PAM and greedy algorithms achieve an interesting trade-off between speed and quality of the layouts. Indeed, PAM is beneficial as it speeds up greedy algorithms. Furthermore, greedy algorithms allow generating better layouts at the cost of slowing down the algorithm. The balance between speed and quality can be regulated by using a finer discretization, testing different locations for the first turbine placement, or acting on the nature of the algorithm. In particular, the use of a re-location stage at each constructive iteration increases the quality of the layouts substantially but reduces the speed of execution. As a result, the proposed algorithms present different characteristics: the BG is the fastest but its median layout is the worst; ADREMOG produces the best layouts in the longest time; ADREMOG II is a compromise between the two former algorithms.

Acknowledgements

This is the final milestone of my two-year journey in Delft. This multicultural pole of excellence has boosted my competence in sustainable energy and, equally important, it has connected me with extraordinary people. My friends, you made this experience in Delft unforgettable. I am glad I had the chance to meet you, learn from you, and have fun together.

This research has undoubtedly been the focal point of the current year. Nevertheless, I hardly remember moments in which I let stress take over positive feelings, despite the workload and my self-imposed pressures. This sense of happiness would not have been possible without certain people.

In particular, I felt privileged for being supervised by Erik. The help he gave me went beyond the precious feedback, the continuous availability, and the patient explanations. Indeed, his contagious passion for research and eagerness for teaching allowed me to express myself to the best. All in all, I thank him for having been a constant source of serenity throughout this research.

Still, serenity abroad could not have been possible without strong roots. I am grateful to my family (Piero, Imma, Livia, and Onelia) for their love and support as I journeyed away from my comfort zone.

Finally, it is time to thank my friends Andres and Winand. Moving to Lipkensstraat with them represented the best turning point I could imagine after a tough first year in Rijswijk. Indeed, living together was priceless.

Contents

Nomenclature

List of abbreviations

Acronym	Description
ADREMOG	Add-Remove-Move Greedy optimization algorithm
AEP	Annual Energy Production
BG	Basic Greedy optimization algorithm
BP	Bastankhah and Porté-Agel
CFD	Computational Fluid Dynamics
EW	Engineering Wake
GHG	Greenhouse Gas
IEA	International Energy Agency
LCoE	Levelized Cost of Energy
NPV	Net Present Value
PAM	Pre-Averaged Model
PDF	Probability Density Function
PG	Pseudo-Gradients
PMF	Probability Mass Function
WFLO	Wind Farm Layout Optimization

List of greek symbols

Symbol	Description	Units
α	Angle between θ and ψ	$^\circ$
$\Delta\theta$	Wind direction bin size	$^\circ$
ΔU	Wind speed bin size	$^{\mathrm{m}}/_{\mathrm{s}}$
ε	Correction factor for the Gaussian wake model	–
ζ	Normalized wind speed deficit w.r.t. the free stream wind speed	–
$\zeta_{\max}$	Normalized wind speed deficit at the centre of a wake	–
η	Wind turbine efficiency	–
θ	Wind direction	$^\circ$
θ_i	i-th wind direction sector angle	$^\circ$
θ_{ref}	Wind direction according to traditional convention	$^\circ$
ρ	Air density	$^{\mathrm{kg}}/_{\mathrm{m}^3}$
σ	Standard deviation of a Gaussian velocity deficit profile	m
ψ	Direction of the vector connecting an upwind to a downwind turbine	$^\circ$
Ω	Design space of a design optimization problem	–

List of roman symbols

Symbol	**Description**	**Units**
A	Rotor area	m^2
A_{l}	Intersection area between a wake and a partially waked rotor	m^2
$A_{\mathrm{l},i}$	Intersection area between the i-th wake and a partially waked rotor	m^2
c	Scale parameter of a Weibull curve	m/s
C_{P}	Power coefficient	–
C_{T}	Thrust coefficient of a wind turbine	–
d	Magnitude of $\vec{d}$	m
d_{min}	Minimum spacing between turbines	m
$d_{\parallel}$	Downwind distance	m
$d_{\perp}$	Crosswind distance	m
$\vec{d}$	Euclidean vector connecting an upwind to a downwind turbine	m
D	Diameter of a turbine	m
E	Expectation	–
E_{farm}	AEP for a wind farm	MWh
E_{solitary}	AEP for a solitary wind turbine	MWh
f	Bivariate PDF of wind speed and wind direction	–
F	Bivariate PMF of wind speed and wind direction	–
F	Objective function of a general optimization problem	–
F_{T}	Thrust force	N
g	Inequality constraint	–
h	Equality constraint	–
H	Hub height	m
H_{ref}	Reference height	m
k	Shape parameter of a Weibull curve	–
K	Wake expansion coefficient	–
L	List of turbines that are tested for re-location	–
m	Number of vertices of a wind farm boundaries	–
m_1	Number of equality constraints	–
m_2	Number of inequality constraints	–
n	Number of design variables	–
n_{w}	Number of wakes affecting a turbine	–
N_{p}	Number of possible locations selected in a wind farm	–
N_{t}	Number of wind turbines	–
N_{wd}	Number of wind direction sectors	–
N_{ws}	Number of wind speed bins	–
P	Power of a single turbine	MW
P_{farm}	Power of a wind farm	MW
P_{rated}	Rated power of a wind turbine	MW
P_{∞}	Wakeless power of a turbine	MW
Q	Power loss of a turbine due to wakes	MW
r	Radial distance from the centre of the rotor	m
R	Set of real numbers	–
R	Radius of a wind turbine rotor	m
R_{w}	Radius of a wake	m
S	List of indices corresponding to positions selected for a layout	–
T	Number of hours in one year	h
T_{d}	Downwind turbine	–
T_{u}	Upwind turbine	–
u	Wind speed inside a wake	m/s
u_i	Wind speed in wake i	m/s

u_{m}	Wind speed in a mixed wake	$^{\mathrm{m}}/_{\mathrm{s}}$
$\bar{u}$	Average wind speed at a rotor	$^{\mathrm{m}}/_{\mathrm{s}}$
$\bar{u}_k$	Average wind speed at the rotor of the k-th turbine	$^{\mathrm{m}}/_{\mathrm{s}}$
U	Free stream wind speed	$^{\mathrm{m}}/_{\mathrm{s}}$
U_j	j-th wind speed bin	$^{\mathrm{m}}/_{\mathrm{s}}$
U_{in}	Cut-in wind speed	$^{\mathrm{m}}/_{\mathrm{s}}$
U_{out}	Cut-out wind speed	$^{\mathrm{m}}/_{\mathrm{s}}$
U_{rated}	Rated wind speed	$^{\mathrm{m}}/_{\mathrm{s}}$
v	Vertices of a wind farm	–
V	Set of points in the wind farm site	–
x	Horizontal coordinate	m
x_{min}	Westernmost point of a wind farm	m
x_{max}	Easternmost point of a wind farm	m
y	Vertical coordinate	–
y_{min}	Southernmost point of a wind farm	m
y_{max}	Northernmost point of a wind farm	m
z	Design variable of an optimization problem	–
z_0	Surface roughness length	m
Z	Design vector of an optimization problem	–
Z^*	Optimal solution of an optimization problem	–

1
Introduction

1.1 Context

In 2014, the European Union committed to cutting the greenhouse gas (GHG) emissions by 40% for 2030 compared to the levels in 1990 [1]. The plan combines two strategies: improving energy efficiency and increasing to 27% the contribution of renewable sources in the energy sector. Recently, the European Union revised this target upwards to 32%, proving that high expectations exist regarding the implementation of green energy on a larger scale throughout the next decade. Indeed, renewable friendly policies [23] as well as a steady reduction of the production cost [19] are boosting the use of renewable energy sources for electricity production. A lower price of wind energy will encourage further investments in the wind energy sector, resulting in a faster spread of this technology into the global energy mix.

The energy conversion to electricity occurs in wind farms, defined as a cluster of wind turbines located in a geographical area, either on-shore or off-shore. With the aim of making wind energy more cost-effective, research has been done to optimize the design of a wind farm. In particular, layout optimization is important. The first attempt to generate optimized layouts belongs to Mosetti et al. [29], who accounted for the negative impact that wakes have on the energy yield of a wind farm. Nowadays, wind farm layout design has evolved into a multidisciplinary field, due to the existence of various objectives colliding with AEP maximization. Indeed, AEP maximization may clash with cable layout optimization, as the former increases the distance among the turbines to reduce wake effects, whereas the latter tends to shorten the connections in a wind farm to cut out the infrastructure cost. Also, intangible design variables, such as visual impact or noise assessment, are now taken into consideration.

In spite of the multi-objective nature of wind farm design, the assessment of the energy yield and its maximization remains a driving factor of the wind farm layout optimization and an independent field of research.

1.2 Problem analysis

Since the WFLO was firstly investigated in 1994, optimization techniques have progressed. Furthermore, technological advances resulted in more powerful computers. Also, engineering models resort to simplifications that reduce their complexity but still manage to describe the wake deficit accurately. Nevertheless, WFLO is still considered a very computationally intensive problem. Hence, the introduction of further complexity might discourage researchers from investigating more accurate models, as it results in longer simulations. Besides, seeing as wind farm dimensions are increasing and multi-objective layout optimizers are being developed, the wind energy industry might also be interested in methods capable of dealing with fast WFLO without compromising accuracy.

The interaction between turbines due to wake effects causes the WFLO to be computationally demanding. Indeed, optimization algorithms generate possible solutions that are then evaluated by comparing the values of the objective function (e.g., AEP maximization). During the optimization procedure, the energy yield is derived by summing up the production of the entire wind farm for each wind direction. Due to the relative dependency among wind turbines, this method requires to model wake deficits and average the production of each turbine over the wind resource every time the layout changes. Therefore, this traditional framework to assess the AEP is a barrier to accelerating the optimization algorithms.

The approach proposed in this thesis can speed up the optimization procedure by using a different approach to assess the AEP: the Pre-Averaged Model (PAM). The PAM is a function that expresses the expected power loss due to the wake of a turbine at representative points in its surroundings. To calculate the expected power loss at some point, assume that a fictitious turbine is placed there, and it experiences wake deficit due to the presence of another turbine at the centre of PAM. Then, the expected wake loss of the target turbine is derived by averaging over the wind resource. The same approach can be used to calculate the expected power loss in a wind farm. Indeed, the wake loss of each turbine is obtained through the PAM as the sum of the power losses due to other turbines. Finally, the AEP is derived by subtracting the energy loss from the theoretical energy yield in the absence of wakes. As the PAM relies on the superposition of power losses, it gives the possibility to precompute the expected power losses among turbines and use this information during the optimization procedure. Chapter 4 proposes one method which allows to run wake simulations and average the turbines power output over different wind conditions only once when generating a layout.

1.3 Objectives and methodology

This thesis elaborates a method that combines the use of optimization algorithms to drive the Wind Farm Layout Optimization WFLO and the PAM to assess the objective function (AEP). Firstly, this research creates an algorithm that makes use of the PAM to calculate the AEP. Secondly, we search for a suitable optimization approach to combine with PAM. After choosing the class of greedy algorithms, two of them are devised. The two algorithms differ regarding the quality of the layouts that they obtain and the time required for their generation. Indeed, the one named BG is faster but less accurate than the other, which is referred-to as ADREMOG. Later on, the different approaches are tested in a case study proposed by Mosetti et al. [29], and findings are used to develop a new optimization algorithm, called ADREMOG II. Finally, the algorithms are tested in two case studies released by the International Energy Agency (IEA) in the context of Task 37 on Wind Energy.

The driving questions that this work aims at answering are listed below:

(1) *Is PAM applicable to the WFLO?*

The goal is to validate the use of PAM for WFLO.

(2) *What is the dependency of the PAM on wind resource binning and the wake model?*

By answering this question, it is possible to provide a set of good practices concerning the use of PAM.

(3) *Is it possible to trade off quality of the results and speed of execution?*

If so, the resulting optimization algorithm can be adapted to different applications.

(4) *How does a WFLO algorithm based on PAM perform with respect to algorithms that use a traditional approach to calculate the AEP?*

The aim is to evaluate whether the PAM reaches a useful trade-off between quality of the results and speed of execution compared to current WFLO algorithms.

The answer to (1) emerges from a comparison over different layouts between the AEP calculated by PAM and by following the traditional approach. If the values show a correlation among them, the PAM can trustfully be employed in WFLO. The second question is addressed as follows. The optimization approach is used to generate layouts for Mosetti's problem by employing different Engineering Wake (EW) models and a varying number of wind directions. Then, the effect of these factors is investigated by comparing the AEP of the outcomes. Moving to the third research question, the performances of BG and ADREMOG for Mosetti's case study are compared. Subsequently, we can identify the main characteristics that contribute to slowing down the computation time of the algorithms and, on the other hand, to increasing the quality of the layouts. By acting on these elements, different ratios of speed and accuracy are obtained. Finally, participation in the IEA's Task 37 could help to answer the fourth question. Indeed, Task 37 tests different optimization strategies in the same case studies, allowing a comparison among them.

1.4 Thesis outline

This thesis has the following structure:

- Chapter 2 starts with an introductory discussion on design optimization, which then gives way to the presentation of WFLO.
- Chapter 3 presents the two single wake models, respectively, the Jensen and the Bastankhah and Porté-Agel (BP) wake model, that are employed throughout the thesis for the derivation of the results. It also explains how single wake models can be used to derive the wake deficits in a wind farm.
- Chapter 4 explains how to derive the AEP by following the PAM and the traditional approach. It includes a discussion on processing the wind resource measurements correctly, as well as a final recap on the differences between the two approaches.
- Chapter 5 consists of a preliminary discussion on the different optimization algorithms used in the literature to cope with WFLO problem. Then, it presents two optimization algorithms that have been selected. Finally, the last section presents one method to combine the PAM with greedy algorithms for WFLO.
- Chapter 6 elaborates on the results of the proposed optimization approach for Mosetti's problem.
- In Chapter 7, the algorithms are tested in increasingly complex case studies. The outcomes of the previous chapter are also used to adapt a new algorithm to the new case studies.
- Chapter 8 draws the final conclusions, provides instructions for a correct use of PAM, and gives recommendations for future research.

2

The Wind Farm Layout Optimization Problem

The chapter gives an overview of the Wind Farm Layout Optimization (WFLO) problem. Firstly, it introduces the topic by giving general information on design optimization, such as the definition and the terminology. Then, the discussion is moved into the specifics of WFLO regarding the objective function, the design variables, and the constraints.

2.1 Optimization: definition and terminology

This section gives a short introduction to design optimization.

An optimization problem is characterized by a set of *design variables* $z_1, z_2, ..., z_n$, which are confined into a *design vector* $Z = (z_1, z_2, ..., z_n)$. Any combination of the design variables is defined as a possible *solution* to the optimization problem, and belongs to the *set constraint* $\Omega \subseteq \mathbb{R}^n$. Nevertheless, some combinations are better than others. The yardstick to assess the solutions is the *objective function* $\mathbb{F}$. Depending on the optimization problem, the aim is either to find the combination that maximizes the value of the objective function or that minimizes it. The best combination encountered is called the *optimal solution* Z^*. Furthermore, a problem can be constrained. The term indicates that restrictions are applied to the set constraint Ω. These limitations are, respectively, m_1 *equality constraints* $h_i(Z)$, and m_2 *inequality constraints* $g_j(Z)$. If no restriction is applied, hence m_1 and m_2 equal to zero, then the optimization problem is called *unconstrained.* Finally, if a solution inside Ω respects the constraints, then is defined as *feasible.* Otherwise, the solution is *infeasible.*

To summarize, a generalized optimization problem consists of three main elements: the design vector, the objective function, and the constraints. It can be mathematically formulated as follows [33]:

$$\begin{aligned} &\textit{find } Z^* \textit{such that } \mathbb{F}(Z^*) = \max(\mathbb{F}(Z)) \\ &\textit{subject to} \quad h_i(Z) = 0, \quad i = 1, ..., m_1 \\ &\qquad\qquad\quad\; g_j(Z) \leq 0, \quad j = 1, ..., m_2 \\ &\textit{with } Z = (z_1, z_2, ..., z_n) \quad \textit{and} \quad Z \in \Omega \subseteq \mathbb{R}^n \end{aligned}$$

2.2 Introduction to the WFLO problem

This section presents possible options concerning the choice of the objective function, the design vector, and the constraints. It also indicates the approach we selected for each of these elements.

2.2.1 The objective function

WFLO is a broad topic that involves many disciplines. In practice, the optimal design emerges from a trade-off between different conflicting objectives. As an example, consider a simple layout composed of two turbines. If the criterion is to maximize energy production, then the further the turbines are, the better the layout is in terms of energy yield. Indeed, a longer distance allows an increased wind speed recovery inside the wake. Nevertheless, if the problem only accounts for the cost of the infrastructure, then the optimized solution minimizes the length of cabling between the two turbines and, hence, their distance.

Five different objective functions are identified in the literature: Annual Energy Production (AEP), Instantaneous Power Conversion, Levelized Cost of Energy (LCoE), Financial Balance, and Net Present Value. [18]. The first two optimization functions are energy-related, whereas the remaining consider economic factors (e.g., cost of electricity) as well as the energy production. Energy-related indicators are of secondary importance to project developers, whose aim is to diminish the cost of energy as much as possible. Hence, LCoE is generally recognized as the most appropriate objective function to evaluate the economic feasibility and competitiveness of a wind farm project. Indeed, bids in renewable energy auctions are based on this indicator [20].

Here, the choice is to opt for the AEP as the objective function, which accounts for the impact wake effects in the energy yield. Indeed, ignoring the additional complexity of economic variables or multiple objective functions eases the analysis of the PAM framework and the comparison with the related literature.

2.2.2 The design variables

The selection of the optimization function influences the WFLO design vector. If an economic objective function is selected, its value is influenced by the number of turbines N_t that are placed. In this case, the optimal number of turbines becomes a new design variable. On the other hand, energy-related objective functions require to fix the number of turbines as, generally, the objective function value increases every time a turbine is added. Independently of the chosen objective function, one may choose to introduce new design variables such as the hub height or even turbine model.

In this work, the decision is to opt for a fixed number N_t of identical turbines. Hence, the design vector consists of a set of N_t points corresponding to each turbine's location in the wind farm. Therefore, *design vector* and *layout* are synonyms in this specific case. Each design variable z_i belongs to $\mathbb{R}^2$, as it is represented by an abscissa x_i and an ordinate y_i. An example of the design vector relative to Figure 2.1 is provided:

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} (x_2, y_1) \\ (x_4, y_3) \\ (x_5, y_2) \end{bmatrix} \qquad \textit{where} \quad |Z| = N_\mathrm{t} \quad \textit{and} \quad N_\mathrm{t} = 3.$$

2.2.3 The constraints

In a WFLO problem, the domain of the objective function is limited by two types of constraints, respectively, the minimum spacing between turbines and the wind farm boundaries.

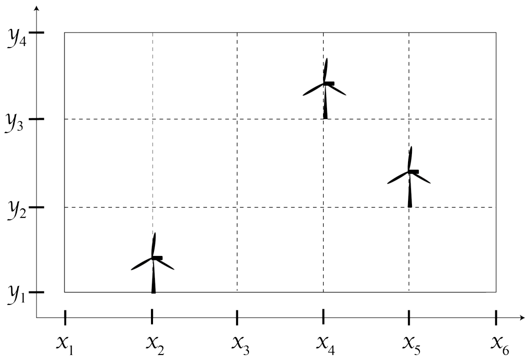


Figure 2.1: Example of a wind farm layout.

The first constraint can be expressed as follows:

$$g_{(i,k)}(Z) : d_{\text{min}} - \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \leq 0 \qquad \forall i \neq k \quad i, k \in \{1, 2, ..., N_{\text{t}}\} \tag{2.1}$$

where d_{min} indicates the minimum distance between turbines. Indeed, d_{min} is often regulated for safety reasons. For instance, d_{min} needs to be at least four times the diameter of the turbine rotor in the Netherlands [31]. In this thesis, its value is adapted to each of the case studies.

The second type of constraint requires turbines to be placed inside the wind farm boundaries. Let $(v_1, v_2, ..., v_{\text{m}})$ be the vertices of a wind farm boundaries. Assume that the position z_i of each turbine is chosen among N_{p} possible locations. Then, the following relation can be used to verify the compliance of z_i with the constraint [11].

$$h_i(Z) : \sum_{k=1}^{\text{m}} \measuredangle v_k z_i v_{k+1} - 360^\circ = 0 \qquad \forall i \in \{1, 2, ..., N_{\text{p}}\} \tag{2.2}$$

$\measuredangle v_i z v_{i+1}$ corresponds to the angle between two consecutive points v_i, v_{i+1}, whose vertex is z. Then, z belongs to the wind farm if the sum of the angles between all the consecutive vertices of the polygon is 360°.

Figure 2.2 gives a visual representation of all the constraints applied to a WFLO which are considered in the present work.

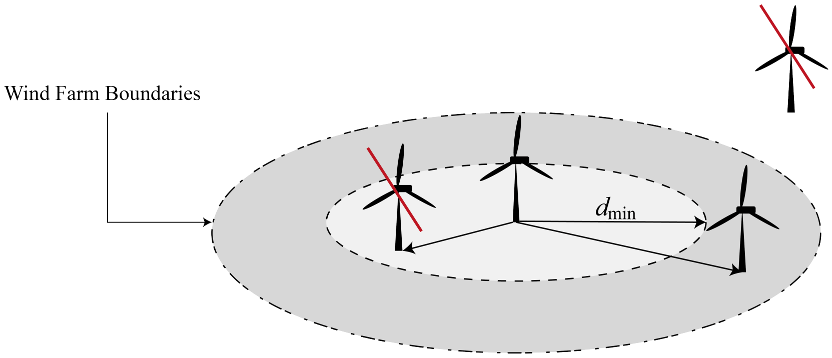


Figure 2.2: Visualization of the constraints used in this thesis. The crossed-out turbines violate a constraint. Adapted to this thesis from Cirillo [11].

3
Wake Models

As the wind passes through a turbine, a disturbed region appears downstream. This region, which is named *wake*, is characterized by increased turbulence, as well as a lower speed than the undisturbed wind flow. This chapter starts with an overview of wake models. Afterwards, two models, respectively, the Jensen wake model and the Bastankhah and Porté-Agel (BP) wake model are treated. Finally, the single wake models are used for a cluster of turbines. This involves defining how wakes affect the rotor of downwind turbines, and how to account for multiple, mixing wakes.

3.1 Introduction to wake modelling

Attempts to modelling the wake of a turbine have produced many different approaches, which can be grouped into two categories: Engineering Wake (EW) models and Computational Fluid Dynamics (CFD) models. EW models describe the wake evolution employing Newtons' laws and momentum conservation, or analytical functions based on empirical studies, whereas the second approach makes use of Raynolds Averaged Navier Stokes equations or Large Eddy Simulations [18]. By following common practice in the Wind Farm Layout Optimization (WFLO) field [18], this work considers only the implementation of EW models.

Indeed, the complexity of CFD models makes them hardly used in WFLO, although they have proven to be better at describing wakes. On the contrary, EW models are generally simple and easy to process but tend to overestimate the wake losses [5]. Nevertheless, studies show that a EW model produces accurate results when describing the wind deficit in the far wake [5]. Although the exact definition of the far wake is a topic of discussion, the literature sets its start at between 3 and 5 diameters downstream of the rotor [12]. Therefore, these models are valid if applied to real wind farms, where the spacing between the turbines is usually larger [28].

Commonly, an EW model derives the wind speed u inside a wake as a function of the free stream wind speed U, the distance $d_{\|}$ downstream from the rotor, and the radial position r respect to the wake centre. Nevertheless, this thesis focuses on layout optimization of turbines at the same height. Therefore, there is no need to derive the wind speed of wakes at different altitudes. In this case, r can be replaced with the distance $d_{\perp}$ from the wake centre on the crosswind direction, as the two variables are equivalent. Finally, the standard form of a EW model is as follows:

$$u(U, d_{\|}, d_{\perp}) = U \cdot (1 - \zeta(d_{\|}, d_{\perp})). \tag{3.1}$$

Therefore, $u(U, d_{\|}, d_{\perp})$ shows the dependency of the wake wind speed on the three parameters above-mentioned. Instead, ζ represents the normalized wind speed reduction with respect to U:

$$\zeta = 1 - \frac{u}{U}. \tag{3.2}$$

As the wake proceeds downstream, the wind speed recovers due to its interaction with the free stream wind. This phenomenon is modelled by employing a wake decay coefficient K.

3.2 Selected wake models

In this section, two models, respectively, the Jensen wake model and the BP wake model are treated. In spite of both being EW models, how they describe the wake differs substantially. Indeed, the Jensen model results in constant wind speed over the wake section perpendicular to the wind flow given $d_{\|}$. On the other hand, the second generates a Gaussian wind speed profile inside the wake at the same downwind distance.

3.2.1 Jensen wake model

The *Jensen model* was introduced by Jensen in 1983 [22] and, then, further developed by Katic et al. [24]. At the time the model was created, lack of computational resources required highly simplified wake models that could be used by the industry to assess the productivity of a wind farm. Therefore, the model relies on semi-empirical assumptions, prioritizing the description of the kinetic energy balance in the wake rather than on representing its wind speed field precisely [24]. Currently, it is by far the wake model that is used the most in WFLO research [18]. The Jensen model prescribes a wake geometry as it is illustrated in Figure 3.1. The initial diameter of the wake is equal to the turbine rotor. Going downwind, the wake expands linearly according to the wake decay coefficient. Furthermore, the wind speed profile is considered independent on the radial position, resulting in a top-hat wake profile.

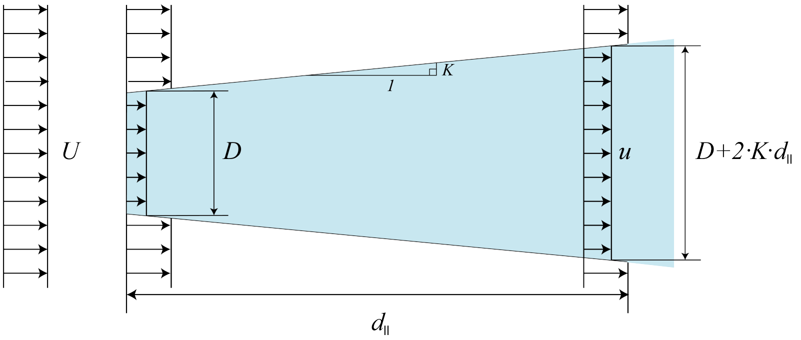


Figure 3.1: Geometry of the Jensen wake.

By applying the conservation of mass in stream tube that encloses the wake, Katic et al. [24] derive the wind speed deficit in it as a function of the downwind distance from the rotor $d_{\|}$ as:

$$\zeta(d_{\|}) = \frac{1 - \sqrt{1 - C_{\mathrm{T}}}}{(1 + \frac{2 \cdot K \cdot d_{\|}}{D})^2}. \tag{3.3}$$

An example of the wake deficit obtained by the Jensen model is depicted in Figure 3.2. As the wake is confined into a restricted area surrounded by free stream wind, the wind speed is:

$$u(U, d_{\|}, d_{\perp}) = \begin{cases} U \cdot (1 - \zeta(d_{\|})), & \text{if } |d_{\perp}| \leq r + d_{\|} \cdot K \\ U, & \text{otherwise} \end{cases}. \tag{3.4}$$

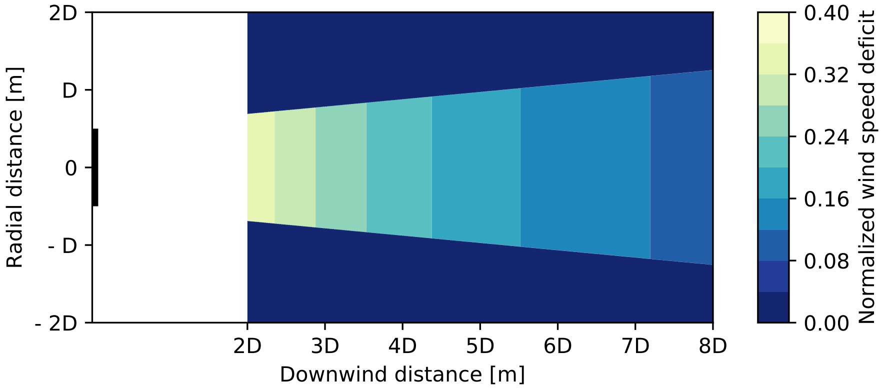


Figure 3.2: Wake deficit for the Jensen wake model.

3.2.2 Bastankhah and Porté-Agel wake model

The assumptions made by Jensen [22] and by Katic et al. [24] result in a simple but widely-used wake model. Indeed, simplicity is certainly the most valuable feature of the Jensen model and the main reason why it is often preferred to other wake models. Nevertheless, the description of the wake deficit through a top-hat profile does not correspond to reality. This approximation may introduce errors when evaluating the AEP [6].

The wake model developed by BP [6], here labelled as the BP model, intends to give a more realistic description of the velocity deficit inside wakes. In line with the outcomes of wind-tunnel measurements, numerical simulations, and data of operating wind farms, the authors describe the velocity deficit in the wake using an axisymmetrical Gaussian function. Thus, the normalized velocity deficit in the wake can be described by:

$$\zeta(d_{\parallel}, d_{\perp}) = \zeta_{\max}(d_{\parallel}) \cdot \exp\left(-\frac{{d_{\perp}}^2}{2\sigma^2(d_{\parallel})}\right) \tag{3.5}$$

where $\zeta_{\max}$ and σ depend on the downwind distance from the rotor. The former describes the normalized maximum velocity deficit at the center of the wake, whereas the latter is the standard deviation of the velocity deficit distribution. $\zeta_{\max}$ can be derived by applying mass and momentum conservation to the stream tube and neglecting the effect of pressure forces, turbulent shear, and gravity. These simplifications lead to:

$$\zeta_{\max}(d_{\parallel}) = 1 - \sqrt{1 - \frac{C_{\mathrm{T}}}{8({}^{\sigma(d_{\parallel})}/_{D})^2}}. \tag{3.6}$$

As for the Jensen model, the BP model assumes linear expansion of the wake, which is taken into account by Equation 3.7.

$$\frac{\sigma}{D} = K \cdot \frac{d_{\parallel}}{D} + \varepsilon \tag{3.7}$$

Nevertheless, the expansion coefficient K differs from the Jensen model, as it indicates the growth rate of the standard deviation (${}^{\mathrm{d}\sigma}/_{\mathrm{d}d_{\parallel}}$) rather than the wake radius (${}^{\mathrm{d}R_{\mathrm{w}}}/_{\mathrm{d}d_{\parallel}}$). Instead, ε is an empirical correction factor to account for the value of ${}^{\sigma}/_{D}$ as $d_{\parallel} \to 0$. It can be expressed as [6]:

$$\varepsilon = 0.2 \cdot \sqrt{\frac{1}{2} \cdot \frac{1 + \sqrt{1 - C_{\mathrm{T}}}}{\sqrt{1 - C_{\mathrm{T}}}}}. \tag{3.8}$$

Finally, merging equations 3.6 and 3.7 into 3.5 and rearranging leads to:

$$\zeta(d_{\parallel}, d_{\perp}) = \left(1 - \sqrt{1 - \frac{C_{\mathrm{T}}}{8({}^{\sigma(d_{\parallel})}/_{D})^2}}\right) \cdot \exp\left(\frac{1}{2} \cdot \left(\frac{d_{\perp}}{\sigma(d_{\parallel})}\right)^2\right). \tag{3.9}$$

However, this work employs a simplified version of the BP model, which will be named *Gaussian model.* It corresponds to the reference wake model created by Baker et al. [4], and assumes $\varepsilon = {}^1/\sqrt{8}$. In support of the International Energy Agency's (IEA's) Wind Energy Task 37, the model was created to run a cross-comparison among different WFLO strategies and, subsequently, choose upon the best combination of wake model and optimization algorithm [4]. The Gaussian model has been preferred to the BP mainly due to the participation to the Task 37, which will be introduced more extensively in Chapter 7.

An example of the wake deficit obtained by the Gaussian model is depicted in Figure 3.3. As the wake deficit only occurs downstream the rotor, the wind speed can be calculated as:

$$u(U, d_{\parallel}, d_{\perp}) = \begin{cases} U \cdot \left(1 - \zeta(d_{\parallel}, d_{\perp})\right), & \text{if } d_{\parallel} > 0 \\ U, & \text{otherwise} \end{cases}. \tag{3.10}$$

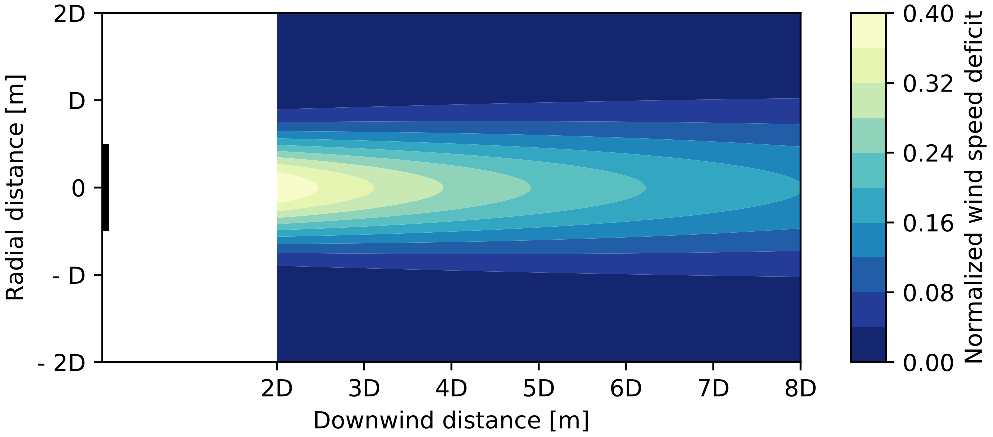


Figure 3.3: Wake deficit depicted by the Gaussian wake model.

3.3 Applying wake models to a cluster of turbines.

Section 3.2 describes two wake models that are commonly used to derive the wind speed deficit downwind. This section now illustrates how to apply these models to describe the wind velocity field in a wind farm. It involves making choices about the types of wake incidence and the procedure for mixing the effects of multiple wakes. These core aspects are treated after an introduction on the angle convention adopted in this work.

3.3.1 Angle convention

The wake effects in a wind farm depend on its layout and the wind resource. Generalizing a wake model to a wind farm inevitably involves setting consistent conventions between these two aspects. In particular, these conventions regard the wind directions θ and the angle between two turbines ψ.

The traditional convention takes into consideration the origin of the wind rather than its direction when defining θ. Indeed, it sets the 0° to a wind blowing from north to south. All the other wind directions are defined in a clockwise fashion with respect to it.

In this work, the choice is to deviate from traditional convention. Indeed, it is preferred to adopt the conventional Cartesian coordinate system and to place the 0 angle between the first and fourth quadrant. Also, positive angles rotate counterclockwise. Furthermore, the use of downwind direction is preferred to the upwind direction. Therefore, contrary to the traditional convention, the wind direction is overturned so that θ indicates the direction in which the wind is blowing rather than its origin. Hence, the relation between the two conventions is:

$$\theta = 270^\circ - \theta_{\text{ref}} \tag{3.11}$$

where on the one hand θ is the adopted convention for the wind direction, on the other θ_{ref} is the traditional. A comparison between the two approaches is presented in Figure 3.4.

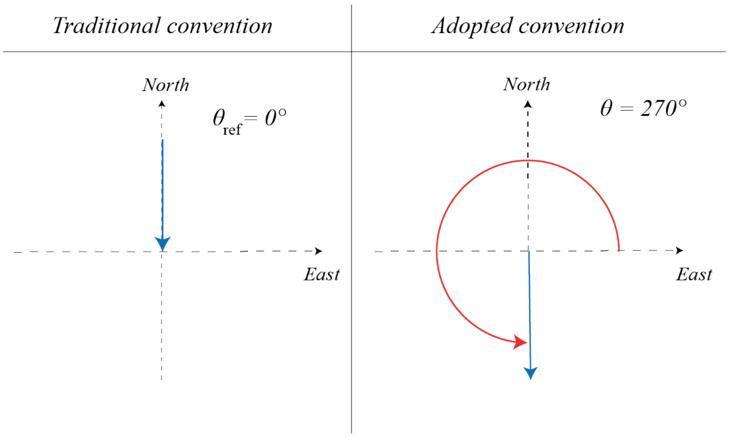


Figure 3.4: Conventions for representing a northerly wind.

The same convention is chosen for the angle ψ, which indicates the angle between a pair of turbines. Given two turbines, a wake source and a wake target, placed in $[x_\text{u}, y_\text{u}]$ and $[x_\text{d}, y_\text{d}]$ respectively, the vector $\vec{d}$ connects them from upwind to downwind. In addition to the starting point, $\vec{d}$ is described by its length d and by its direction ψ. The choice is to set the reference 0° point to the positive x-axis half-line, and counterclockwise rotation for positive angles. As a consequence, ψ can be obtained through the following equation:

$$\psi = \arctan\left(\frac{y_\text{d} - y_\text{u}}{x_\text{d} - x_\text{u}}\right). \tag{3.12}$$

A graphical representation is provided in Figure 3.5. Determining ψ is important, as it allows to relate the relative position between two turbines to the wind direction and, subsequently, determine if there is an interaction between them. Its relevance is revealed in the next discussion on the types of wake incidence.

3.3.2 Types of wake incidence

The choice of a suitable system of conventions eases the assessment of the wake effects. In particular, consistency between angles θ and ψ allow to easily derive α as the angle between the wake longitudinal axis and the vector $\vec{d}$ connecting a source and a target turbine:

$$\alpha = \theta - \psi. \tag{3.13}$$

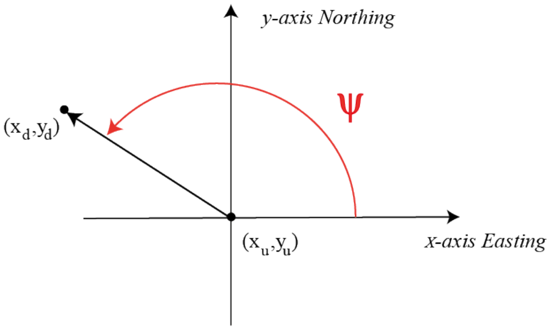


Figure 3.5: Angle convention for ψ.

This angle is commonly used to evaluate the downwind $d_{\|}$ and crosswind $d_{\perp}$ distance, which are the two variables used in the previous sections to derive the wind speed inside a wake. Using the adopted angle convention, the former parameter represents the projection of $\vec{d}$ onto the downwind direction, whereas the latter is the projection of $\vec{d}$ on the crosswind direction:

$$d_{\|} = d \cdot \cos(\alpha); \tag{3.14}$$

$$d_{\perp} = d \cdot \sin(\alpha). \tag{3.15}$$

Therefore, these geometric parameters can also be used to describe the relative position among the two turbines and, subsequently, to assess the impact of a wake source on a wake target. Indeed, not only $d_{\|}$ and $d_{\perp}$ allow to determine the magnitude of the wake deficit at one point, but also the type of interaction between an upstream wake and a downstream turbine. Depending on which wake model is used, the types of interaction among turbines vary. Given a wake source turbine T_{u} and a wake target turbine T_{d}, the Gaussian model only distinguishes two states: waked, if the target turbine is downstream ($d_{\|} > 0$); non-waked otherwise. The target turbine being downstream is also the *conditio sine qua non* waking occurs. The Jensen model distinguishes three types of interaction for $d_{\|} > 0$: full wake (1), no wake (2) or partial wake (3).

(1) If $d_{\perp} < d_{\|} \cdot K$, then the wake caused by T_{u} hits the rotor of T_{d} completely (see Figure 3.6).

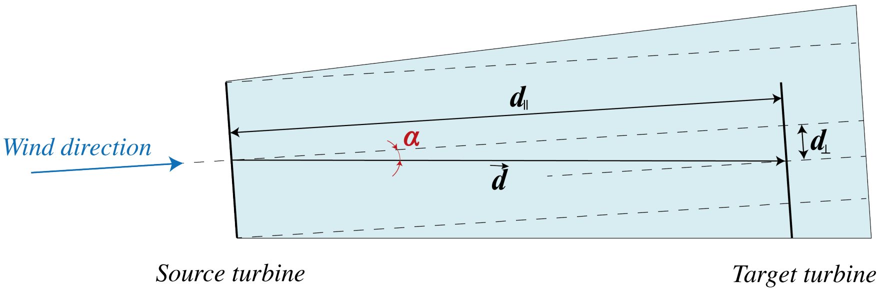


Figure 3.6: Full Wake.

(2) If $d_{\perp} \geq 2 \cdot r + d_{\|} \cdot K$, then the wake caused by T_{u} does not affect T_{d} at all (see Figure 3.7).

(3) If $2 \cdot r + d_{\|} \cdot K < d_{\perp} < d_{\|} \cdot K$, then the wake caused by T_{u} does not enclose the rotor of T_{d} entirely. In this case, the rotor presents two zones with different wind speeds. The rotor

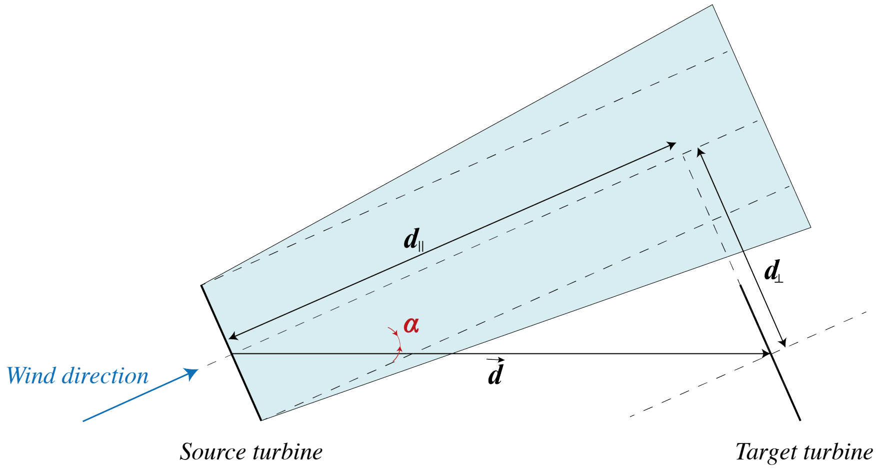


Figure 3.7: No Wake.

experiences the wake wind speed over A_l, which is the area of an asymmetrical lens obtained by intersecting the wake and the rotor, whereas the remaining surface of the rotor experiences the free stream flow. Figure 3.8 gives a representation of the wind speed at the rotor under partial wake. To calculate A_l, we define R_w as the wake radius. The area of the rotor affected by partial waking is obtained as [15]:

$$A_\mathrm{l} = 2 \cdot \beta \cdot {R_\mathrm{w}}^2 + 2 \cdot \gamma \cdot R^2 - 2 \cdot \delta \tag{3.16}$$

in which

$$\beta = \arccos \frac{{R_\mathrm{w}}^2 + {d_\perp}^2 - R^2}{2 \cdot R_\mathrm{w} \cdot d_\perp},$$

$$\gamma = \arccos \frac{R^2 + {d_\perp}^2 - {R_\mathrm{w}}^2}{2 \cdot R \cdot d_\perp},$$

$$\delta = \sqrt{p \cdot (p - R_\mathrm{w}) \cdot (p - d_\perp) \cdot (p - R)},\ p = \frac{(R_\mathrm{w} + d_\perp + R)}{2}$$

One can derive the average wind speed $\bar{u}$ at the downwind rotor as follows. Firstly, the wind speed inside the wake at the target turbine's rotor is obtained by using Equation 3.4. Secondly, the rotor area A_l hit by the wake is calculated. Then, the area of the wakeless rotor surface is obtained by subtracting A_l to the rotor area A. Finally, $\bar{u}$ can be determined through the weighted average over the different wind speed areas:

$$\bar{u}(U, d_\parallel, d_\perp) = u(U, d_\parallel, d_\perp)\ \frac{A_\mathrm{l}}{A} + U\ \frac{A - A_\mathrm{l}}{A} \tag{3.17}$$

3.3.3 Wake interaction

EW models are used to determine the wake deficit of a single wake on a turbine. Nevertheless, wakes might mix within a wind farm. Therefore, a wind farm wake model requires to determine how wakes interact.

The standard approach consists in combining the effect of single wakes using assumptions on superposition. As Crespo et al. outlined [12], this approach does not describe wake interactions precisely. Indeed, single wake models (e.g., Jensen's model or the BP model) assume that the

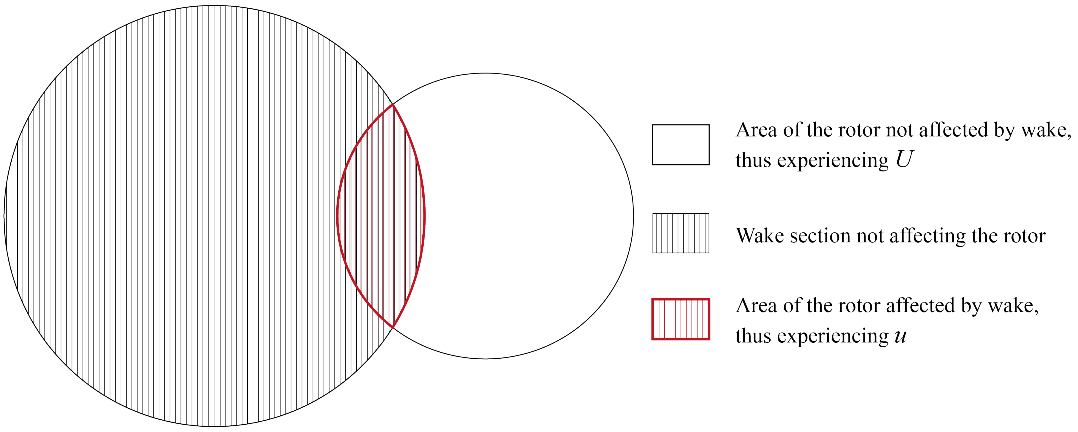

Area of the rotor not affected by wake, thus experiencing U
Wake section not affecting the rotor
Area of the rotor affected by wake, thus experiencing u

Figure 3.8: Representation of a rotor affected partially by a wake. Adapted to this thesis from Engelen [13].

characteristics of the environment where the wake diffuses do not change with respect to where the wake is originated. When a wind farm is considered, neighbouring wakes constantly change the properties of the ambient where a wake develops, such as its turbulence intensity. Hence, the characteristics of the wake might largely diverge from the ones obtained through a single wake model.

Different superposition methods have been proposed. An overview of the most common superposition approaches is presented by Shao et al. [38]. One method consists in applying linear superposition to the wake deficits [26]. The wind speed of the mixed wake u_{m} can be calculated as presented in Equation 3.18. Nevertheless, this procedure might generate negative wind speeds at downwind rotors in case of large perturbations, as it tends to overestimate the wake effects [12].

$$(1-\frac{u_{\mathrm{m}}}{U})=\sum_{i=0}^{n_{\mathrm{w}}}(1-\frac{u_i}{U}) \tag{3.18}$$

Later on, Katic et al. [24] assume that the energy deficit of a mixed wake corresponds to the sum of the energy deficits for each of the wakes that compose it. From this assumption, it derives the wind speed in mixing wakes as the superposition of the squares of the velocity deficits at the same location. If compared to linear superposition, this approach underestimates the velocity deficit caused by multiple wakes, but giving better results. Nevertheless, negative wind speed values can still emerge when turbines are located densely. Therefore, it is necessary to cap the maximum value of the wake deficit to avoid negative wind speeds:

$$(1-\frac{u_{\mathrm{m}}}{U})^2=\min\left(1,\sum_{i=0}^{n_{\mathrm{w}}}(1-\frac{u_i}{U})^2\right). \tag{3.19}$$

In case the Jensen's model is used, a turbine might be affected by multiple wakes of different nature, as represented by Figure 3.9. Two different strategies can be adopted to find the average wind speed at the rotor. One approach combines the wakes by superposition at first. Secondly, it calculates the area relative to each wind speed. Hence, in the most general case, the rotor can experience the free stream wind, as well as wake deficit due to n_m mixed wakes and due to n_w single wakes. The areas of intersection between the rotor and wakes are indicated as $A_{\mathrm{l},i}$ for the n_w single wakes, and as $A_{\mathrm{m},j}$ for the n_m mixed wakes. The remaining surface of the rotor is experiencing the free stream wind speed. Finally, this approach obtains the wind speed at the rotor by averaging over the different wind speed areas as:

$$\bar{u} = \left(\sum_{i=1}^{n_\mathrm{w}} u_{\mathrm{w},i} \cdot \frac{A_{\mathrm{l},i}}{A}\right) + \left(\sum_{j=1}^{n_\mathrm{m}} u_{\mathrm{m},j} \cdot \frac{A_{\mathrm{m},j}}{A}\right) + U \cdot \left(1 - \frac{\sum_{i=1}^{n_\mathrm{w}} A_{\mathrm{l},i} + \sum_{j=1}^{n_\mathrm{m}} A_{\mathrm{m},j}}{A}\right). \tag{3.20}$$

Nevertheless, calculating the intersection area between multiple mixing wakes and the rotor can be challenging. Therefore, the second approach ignores mixing wakes and considers the effect of each wake singularly. It consists of averaging the wind speed deficits ζ_i over the rotor for each wake singularly. Finally, the root sum square is performed in order to obtain the average wind speed at the rotor:

$$1 - \frac{\bar{u}}{U} = \min\left(1, \sqrt{\sum_{i=1}^{n_\mathrm{w}} \left(\frac{\zeta_i \cdot A_{\mathrm{l},i}}{A}\right)^2}\right). \tag{3.21}$$

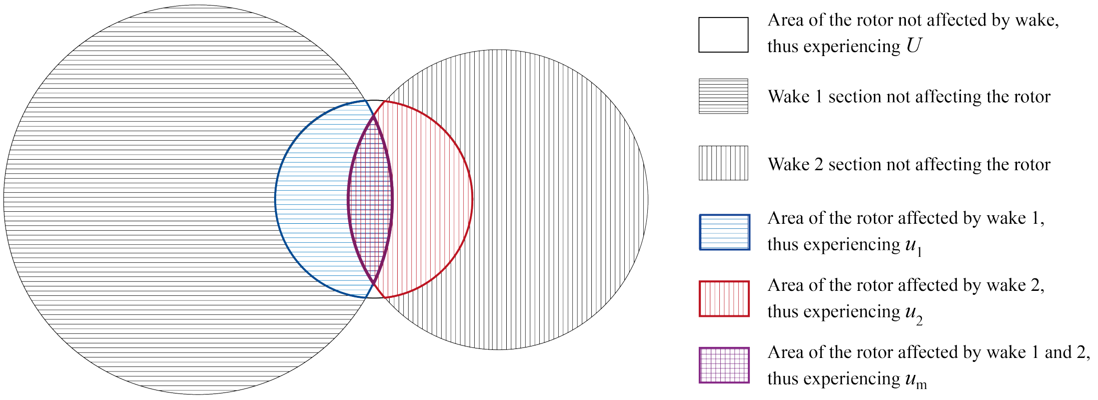

Area of the rotor not affected by wake, thus experiencing U
Wake 1 section not affecting the rotor
Wake 2 section not affecting the rotor
Area of the rotor affected by wake 1, thus experiencing u_1
Area of the rotor affected by wake 2, thus experiencing u_2
Area of the rotor affected by wake 1 and 2, thus experiencing u_m

Figure 3.9: Representation of a rotor affected by multiple wakes. Adapted to this thesis from Engelen [13].

4 Annual Energy Production Assessment

Assessing the AEP (Annual Energy Production) is essential when designing a wind farm. Indeed, as discussed in Section 2.2.1, it is a core aspect of any Wind Farm Layout Optimization (WFLO) problem. Moreover, this procedure is needed to investigate the economic feasibility of a wind farm project. This chapter describes how the energy yield of a wind farm can be calculated, both according to the traditional approach and to the proposed framework: the Pre-Averaged Model (PAM).

In particular, the energy yield depends on two main ingredients, which are wind availability, and the wind farm efficiency. This chapter starts by illustrating how the wind data is processed to obtain a wind resource which is representative of the real condition. Afterwards, the traditional procedure to derive the AEP is introduced. Then, this chapter derives the PAM. Finally, the last section compares the two approaches to clearly remark the innovation introduced by the proposed approach.

4.1 Wind Resource Modelling

Processing the wind data is required for Wind Farm Layout Optimization (WFLO). As the wind resource is aleatory in speed and direction, measurements have to take place for a sufficient period to reliably estimate the potential energy yield of a location. The common practice is to average both the speed and the direction over ten minutes. Direction values do not need further processing. On the contrary, the wind speed has to be adjusted to the hub height H [14]. Assuming that measurements are taken at a reference height H_{ref}, then the logarithmic wind shear law states that the wind speed at H is as follows:

$$U(H) = U(H_{\text{ref}}) \cdot \frac{\ln(H/z_0)}{\ln(H_{\text{ref}}/z_0)}. \tag{4.1}$$

Afterwards, wind data is binned. The bin size for wind speed ΔU and wind direction $\Delta\theta$ has to be chosen carefully, as it impacts the WFLO.

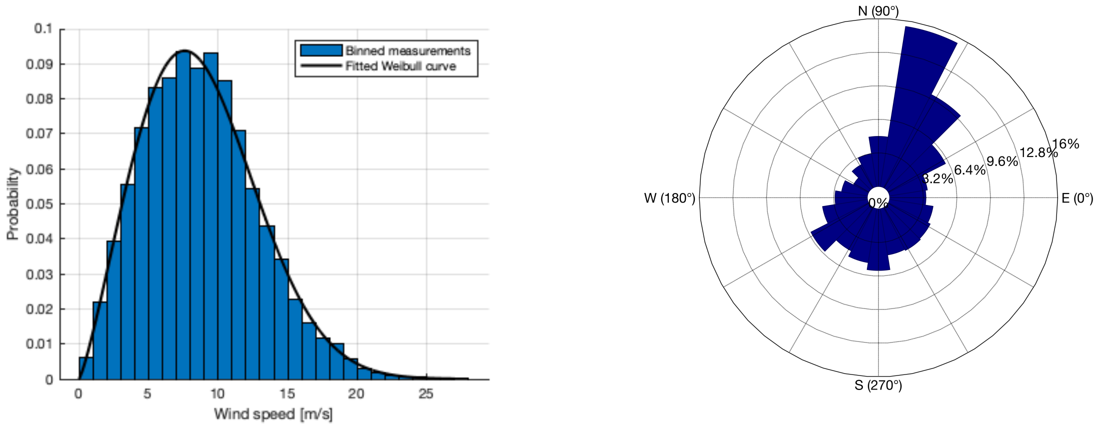
If N_{ws} is the number of wind speed bins and N_{wd} the number direction sectors, binning results in a collection of $N_{\text{ws}} \cdot N_{\text{wd}}$ bins, each one accounting for the frequency of occurrence of a specific wind speed along a precise direction [14]. By dividing the frequency of occurrence by the total number of measurements, a bivariate Probability Mass Function (PMF) $F(U_j, \theta_i)$ is created. The

PMF associates to each bin the probability that a measurement will fall into it.

If the use of continuous variables is preferred to discretization, data can be processed to derive a wind rose, which depicts the probability distribution of the wind direction. Afterwards, fitted Weibull distributions are obtained to describe the wind speed Probability Density Function (PDF) for each i-th direction section.

$$f(U, \theta_i) = \frac{k}{c}(\frac{U}{c})^{(k-1)} e^{-(\frac{U}{c})^k} \tag{4.2}$$

The advantage is that only two parameters, namely, the shape k and the scale factor c are needed to define the wind PDF for each direction. As a result, the wind resource can be summarized effectively by N_{wd} Weibull curves (Figure 4.1a) and the wind rose (Figure 4.1b).



(a) Wind speed binning and fitted Weibull curve for a wind rose sector.

(b) Wind rose.

Figure 4.1: An example of wind resource.

4.2 AEP using the standard approach

Here, the traditional procedure to derive the energy yield is treated. Before discussing how the AEP is assessed for a cluster of turbines, the current section presents the simplified case of a solitary turbine.

4.2.1 Single turbine

The output power P of a wind turbine is a function of the wind speed over the rotor, commonly approximated by the wind speed at the hub despite the wind shear along the vertical axis. The power curve depicts the dependency of power on wind speed. It can be characterized by a cut-in wind speed U_{in}, a cut-out wind speed U_{out}, and a rated wind speed U_{rated}. When the wind speed reaches U_{in}, the turbine is switched on. Above U_{out}, the turbine stops producing for safety reasons. Furthermore, the turbine reaches its full capacity between U_{rated} and U_{out}. In mathematical notation:

$$P(U) = \begin{cases} 0, & U < U_{\text{in}}, U > U_{\text{out}} \\ P_{rated}, & U_{\text{rated}} < U < U_{\text{out}} \end{cases}. \tag{4.3}$$

Finally, the turbine operates in the so-called partial load region between U_{in} and U_{rated}. In theory, the power production in the variable region is described by the following equation:

$$P(U) = \frac{1}{2}\rho \cdot A \cdot C_{\text{P}} \cdot U^3 \qquad (U_{\text{in}} < U < U_{\text{rated}}) \tag{4.4}$$

where ρ is the air density, A is the surface of the rotor, and C_{P} is the power coefficient. Depending on the wind speed, C_{P} varies in the partial load region. Nevertheless, in practice, the power curve is approximated to a pure cubic function or derived from interpolating measurements [27]. The power curve of a Vestas V80-2 MW is presented as an example (see Figure 4.2).

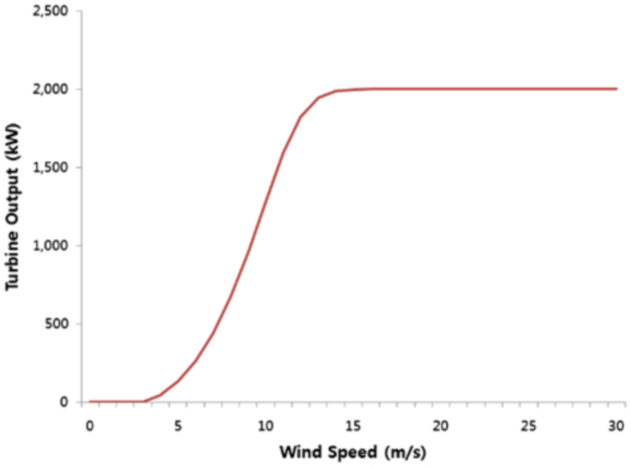


Figure 4.2: Power curve for the Vestas V80-2 MW turbine.

By knowing the power output as a function of the wind speed, one can calculate the expected power $\mathbb{E}(P)$ of a turbine as:

$$\mathbb{E}(P) = \sum_{i=1}^{N_{\text{wd}}} \sum_{j=1}^{N_{\text{ws}}} P(U_j) \cdot F(U_j, \theta_i). \tag{4.5}$$

It has to be mentioned that equation 4.5 is consistent with the wind resource measurements being processed into a wind rose as described in section 4.1. Nevertheless, the energy yield of a solitaire turbine is not affected by the wind direction. Therefore, it is also possible to bin the wind speed measurements ignoring the wind direction. The resulting Weibull curve is then used to calculate the expected power as follows:

$$\mathbb{E}(P) = \sum_{j=1}^{N_{\text{ws}}} P(U_j) \cdot F(U_j). \tag{4.6}$$

Finally, the AEP of a single turbine E_{solitary} corresponds to the product between its expected power $\mathbb{E}(P)$ and the number of hours in one year T:

$$E_{\text{solitary}} = T \cdot \mathbb{E}(P). \tag{4.7}$$

4.2.2 Wind farm

When the study is expanded to a cluster of wind turbines, the AEP assessment procedure is more elaborate due to wake effects among the wind turbines. Indeed, as the wind characteristics vary, the number of turbines affected by wakes might change, as well as the intensity of the wind speed deficit at the rotors. Hence, the traditional approach consists in singularly assessing the wind farm power output relative to each of the wind rose sectors and wind speed bins. Then, the expected value of the power output over the wind resource can be derived through the PMF. Finally, the AEP is obtained by multiplying the expected power by the number of hours in one year.

For the sake of clarity, the procedure is described step by step. For a given layout, a wind speed bin U_j, and a wind rose sector θ_i, the wind speed profile at the rotor of each turbine of the wind farm is derived according to a wake model, as described in Chapter 3. In particular, turbines in a wind farm can experience multiple wakes simultaneously. Therefore, the average wind speed $\bar{u}$ at the rotor needs to be assessed for each of the turbines that compose the wind farm. Once $\bar{u}$ is known, the power production can be derived through the wind turbine power curve (Equations 4.3 and 4.4) as:

$$P_{\text{farm}}(U_j, \theta_i) = \sum_{k=1}^{N_{\text{t}}} P\left(\bar{u}_k(U_j, \theta_i)\right) \tag{4.8}$$

where $\bar{u}_k$ indicates the average wind speed at the rotor of the k-th turbine and P_{farm} represents the power output of the wind farm. The dependence of the wind speed deficit at each rotor on θ and U implies that $\bar{u}$ has to be re-assessed every time the wind resource conditions change. Once the power output corresponding to each of the $N_{\text{wd}} \cdot N_{\text{ws}}$ wind conditions is known, its expected value can be derived:

$$\mathbb{E}(P_{\text{farm}}) = \sum_{i=1}^{N_{\text{wd}}} \sum_{j=1}^{N_{\text{ws}}} P_{\text{farm}}(U_{\text{j}}, \theta_{\text{i}}) \cdot F(U_{\text{j}}, \theta_{\text{i}}). \tag{4.9}$$

Once the expected power of a wind farm is obtained, the AEP is easily derived as:

$$E_{\text{farm}} = T \cdot \mathbb{E}(P_{\text{farm}}). \tag{4.10}$$

4.3 Proposed framework: the pre-averaged model

In this section, firstly, the basic form of PAM is illustrated. Afterwards, this model is employed to derive the AEP of a wind farm.

4.3.1 Single wake source

Let T_{u} be a wake source and T_{d} a wake target, connected by the vector $\vec{d}$. Assume that the two turbines are placed at the same height. The PAM aims at calculating the expected power loss caused by T_{u} on T_{d} as a function of their relative position $\vec{d}$, characterized by its magnitude d and its direction ψ. Figure 4.3 illustrates the parameters that will be used in this section to derive the PAM.

The previous chapter treated how the deficit caused by a wake source on a wake target depends on the wind speed U, the downwind distance $d_{\|}$, and the crosswind distance $d_{\perp}$. Nevertheless, the value of the two latter parameters can be determined by using α and d, as presented in Equations 3.14 and 3.15. In turn, the wind speed at the rotor of T_{d} in the presence of T_{u} can be expressed as a function of α and d. The previous chapter also addressed how the average wind speed over the rotor can be determined. As the PAM is interested in the effect of T_{u} on T_{d}, only one wake is considered. Hence, T_{d} might experience one wake, but, according to the model, it would not experience either mixing wakes or multiple wakes. Then, if a discrete wake model is employed (e.g., the Jensen model), it might be necessary to average between zones at two different wind speeds: the wake wind speed and the free stream wind speed. On the other hand, if the wake model is continuous (e.g., the Gaussian model), the wind speed at the hub adequately approximates the

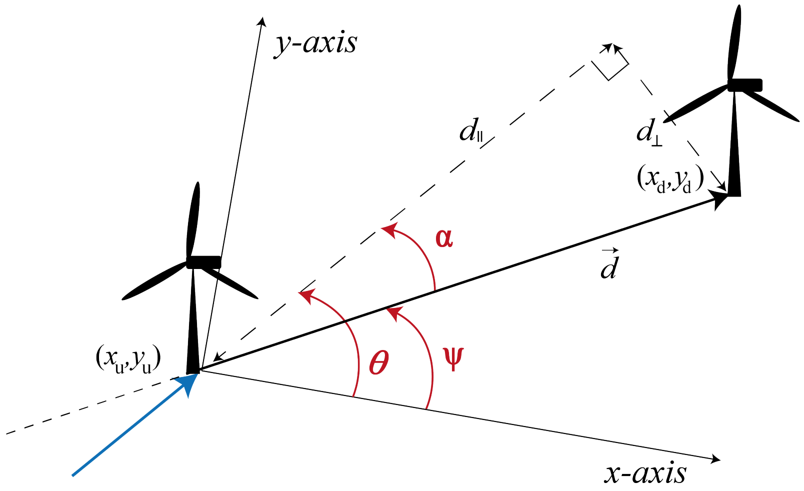


Figure 4.3: Angle conventions used in the PAM.

average over the rotor. To summarize, two equations can be used to assess the average wind speed over the rotor of a downwind turbine:

$$\bar{u}(U,\alpha,d) = \begin{cases} \text{Equation 3.10} & \text{if the Gaussian model is used;} \\ \text{Equation 3.17} & \text{if the Jensen model is used.} \end{cases} \tag{4.11}$$

Note that, according to the new notation, u is a function of U, α and d. The role of α is essential for PAM. Indeed, Equation 3.13 shows that this parameter can be derived by subtracting ψ to the wind direction θ. As the position between T_u and T_d is fixed, so will be ψ and d. Therefore, one can obtain α_i corresponding to each wind direction θ_i as:

$$\alpha_i = \theta_i - \psi \qquad \forall i \in 1 \leq i \leq N_\text{wd}. \tag{4.12}$$

Clearly, the relation above allows to calculate $\bar{u}$ at T_d's rotor for each wind speed U_j and wind direction θ_i using Equation 4.11. Once $\bar{u}$ is known, the power production of T_d can be derived through the power curve. Afterwards, the expected power of T_d is obtained by averaging over the wind resource. This procedure corresponds to the following:

$$\mathbb{E}(P) = \sum_{i=1}^{N_\text{wd}} \sum_{j=1}^{N_\text{ws}} P(\bar{u}(U_j, \theta_i - \psi, d)) \cdot F(U_j, \theta_i). \tag{4.13}$$

In this equation, the argument α_i of the function $\bar{u}$ is replaced with $\theta_i - \psi$ to outline that, for a given wind resource, the expected power $\mathbb{E}(P)$ of T_d depends only on its position respect to T_u. Finally, the expected power loss caused by T_u on T_d is obtained:

$$Q\,(\psi, d) = P_\infty - \mathbb{E}(P) \tag{4.14}$$

where P_∞ is equivalent to Equation 4.5. Given the wind resource, P_∞ is constant, and it indicates the expected wakeless power of T_d. As $\text{E}(P)$ implicitly depends on ψ and d, these variables impact Q likewise.

4.3.2 Wind farm

In section 4.3.1, the effect of a source turbine on a target turbine is derived in function of their position. In this section, the same method is generalized to N_t turbines.

Given a layout of N_t turbines, they are labelled as both a wake source and a wake target. The relative distances $d_{\|\,k,l}$ and directions $\psi_{k,l}$ between each k-th wake source and l-th target are obtained. These geometric variables are then used by the PAM to individually assess the expected power loss $Q(d_{\|\,k,l}, \psi_{k,l})$ caused by each wake source on a target. The effect of single wake sources on a target is then combined by summing up each contribution to power loss. No other attempt of performing power loss superimposition has been found in the literature. Although this research does not investigate other combination rules, a different superposition method may be applied to the PAM. If the proposed approach is chosen, the expected power loss Q_l experienced by the l-th target turbine on a wind farm is obtained.

$$Q_l = \sum_{\substack{k=1 \\ l\neq k}}^{N_\mathrm{t}} Q(d_{\|\,k,l}, \psi_{k,l}) \tag{4.15}$$

Finally, the AEP of a wind farm can be derived as:

$$E_\mathrm{farm} = T \cdot (N_\mathrm{t} \cdot P_\infty - \sum_{l=1}^{N_\mathrm{t}} Q_l). \tag{4.16}$$

4.4 Comparison of the standard approach and the pre-averaged model

By comparing section 4.2.2 and 4.3.2, differences between the traditional approach and PAM become apparent.

Indeed, the traditional approach derives the AEP by averaging the P_farm for any wind speed along every direction. It implies that the wake model has to be run for every wind condition and, subsequently, $\bar{u}$ derived for each of the turbines that compose the wind farm. The effect of both multiple and mixing wakes affecting the turbines is considered. Therefore, if one turbine is added, moved, or removed, the whole procedure for deriving the AEP needs to be carried out again.

On the contrary, the PAM does not calculate directly P_farm, but its value is derived by subtracting the expected power loss Q_l of each turbine from the wakeless total. The power loss of each turbine is assessed by superimposing the power losses caused individually to it by the surrounding turbines. Therefore, if the traditional method relies on the superposition of the wake deficits, the PAM is based on the superposition of the power losses. The next chapter will present a possible approach to take advantage of this property for reducing the computational load of WFLO.

5
Optimization Algorithms

This chapter is central to the thesis. Indeed, optimization algorithms use the information elaborated by the Pre-Averaged Model (PAM) to generate layouts. The chapter starts with an introduction about different classes of algorithms employed in the literature to solve Wind Farm Layout Optimization (WFLO) problems. Then, it motivates the choice of adopting the greedy algorithms, and two of them are devised. Finally, it addresses the strategies followed to merge the PAM and the greedy algorithms in the same optimization procedure.

5.1 Overview of optimization algorithms

Many optimization techniques have been proposed to tackle the wind farm layout problem, as well as several ways to categorize these algorithms. If the criterion is on the use of the derivative of the objective function, then the optimization methods can be grouped into two macro categories: gradient-based and gradient-free [42].

5.1.1 Gradient-based methods

Gradient-based algorithms include methods that use the first and second derivative of the objective function to find the optimal solution in the search domain (e.g., gradient descent and Newton's method). As a consequence, gradient-based optimization requires that the objective function is Lipschitz continuous and differentiable [18]. These restrictions pose challenges to the applicability of such methods for highly complex problems. Furthermore, convergence to a globally optimal solution is guaranteed just in case of a convex/concave objective function. Otherwise, gradient-based algorithms are susceptible to converge to local optima [34]. Nevertheless, a multi-start approach or the introduction of elements of randomness during the search may avoid this pitfall. However, it results in more computationally intensive algorithms.

In the literature, the gradient-based algorithms used to be limited to highly simplified Wind Farm Layout Optimization (WFLO) problems [25] or discouraged [18]. Indeed, WFLO problems are often subject to discontinuities introduced by binning the wind resource or by the wake model (e.g., Jensen's model) that makes the derivation of a differentiable objective function difficult. Currently, new studies demonstrate that gradient-based approaches are more accurate than gradient-free methods [4]. However, the accuracy comes at the cost of a slower convergence to a solution.

5.1.2 Gradient-free methods

Gradient-free methods are also known as heuristics or meta-heuristics. In theory, the term meta-heuristic was coined to indicate higher-level heuristics and defined as “a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality” [16]. Meta-heuristics apply elements of randomness that increase the ability of the algorithm to escape from local optima, whereas most heuristics are deterministic. Nevertheless, no distinction is made in this work between heuristics and meta-heuristics, as the two terms often overlap in the literature.

Gradient-free methods tend to reach a better trade-off between computational complexity and accuracy of the solution than gradient-based approaches. Although heuristics cannot guarantee convergence to the global solution, they dodge the strict requirements on the search space of the gradient-based methods. With heuristic methods, it is possible to take advantage of knowledge about a specific problem to apply tailor-made semi-empirical rules in the optimization process. Unfortunately, as the *No Free Lunch Theorem* states, every heuristic algorithm performs as any other if results are averaged over an infinite set of problems [41]. In turn, this means that no heuristic method is overall the best. Nevertheless, tuned algorithms may outclass other approaches for the WFLO.

A vast range of heuristics applied to the WFLO is available in the literature. Two different strategies can be distinguished: iterative and constructive [18]. Concerning the former, search for an optimal solution starts from a complete, non-optimal layout. At each iteration, the solution evolves to a better one by assessing the objective function in the local search space. The process stops when the stop criterion is reached. Some methods that have successfully been applied to WFLO are genetic algorithms [29], simulated annealing optimization [36] or particle swarm optimization [40]. These non-deterministic methods reach accurate solutions after many iterations. On the other hand, constructive approaches build a complete solution by starting from a partial solution. In WFLO, it results in adding a turbine to the previous layout at every iteration. This category encompasses greedy algorithms and bionic optimization [39].

A greedy heuristic was tested by Ozturk and Norman [32]. In their research, the profit maximization of a wind farm is obtained through three operations: add, remove, and move the turbines. The wind farm domain is chosen to be continuous, resulting in a infinite search area. Therefore, a restricted set of possible turbine placements is randomly generated. Then, the candidate locations are individually assessed by calculating the value of the objective function that would result from adding the turbine in that specific location. Afterwards, the greedy algorithm places a turbine where the wind farm profit is maximized. Subsequently, each turbine is individually removed. If no improvements occur, turbines are moved one by one in each of the wind directions. When no better move is possible, the iteration terminates. To escape from local optima, the algorithm can perturb the solution if only little improvement of the evaluation function occurs after many iterations. Eventually, the greedy algorithm outputs the best layout encountered.

Use of a randomized discrete search space has been overcome by later research [9], [10], where a meshed domain is preferred. It results in a limited number of available locations for the wind turbines. Contrarily to Ozturk and Norman’s algorithm, this choice avoids the use of randomness while choosing a list of possible placements and gives the possibility to build deterministic algorithms. Therefore, given a WFLO, the output of these greedy algorithms is always the same. In this case, the optimization procedure starts with a partial solution. At every iteration, a turbine is added where it maximises the objective function. In turn, each turbine is removed by following the order in which they were added to the layout and moved to a greedier position if it exists. The algorithm terminates when the desired number of turbines is reached. This approach is effectively used by the lazy greedy algorithm [9], which takes advantage of the submodularity of the WFLO to accelerate the optimization procedure. Chen et al. [10] obtained further improvements by speeding up the energy yield calculation.

Song et al. [39] built the so-called bionic optimization algorithm, which applies a constructive strategy as well. Although the optimization procedure follows the same steps depicted for the greedy approach, the turbine is placed where it maximizes its power output. Hence the effect of the new turbine on the rest of the wind farm is neglected. As a result, no guarantee exists regarding the fact that relocating a turbine has a positive impact on the evaluation function. Therefore, this algorithm may fail converging if repeated adjustments are performed.

Finally, constructive heuristics have been used to seed iterative approaches with reasonable initial solutions. Saavedra et al. [37] obtained good results by combining a greedy heuristic with a genetic algorithm.

5.2 Selected and developed algorithms

This section presents the optimization procedures on which PAM is tested. By bearing in mind that the PAM is created to speed up the WFLO, optimization methods are selected accordingly.

In this regard, gradient-based methods and iterative heuristics are discarded. The former method provides the most accurate results, whereas most of the literature focuses on the second approach. Nevertheless, both methods need many iterations to converge to accurate results. On the other hand, constructive heuristic methods are the most suitable. Indeed, heuristic methods can be tuned by applying empirical rules. As a result, the algorithm can be adjusted to seek an ideal trade-off between speed and accuracy. Furthermore, the greediness of the constructive approaches combines perfectly with the PAM. The reason lies in the fact that PAM gives the wake loss as a function of the relative position between turbines. This statement is motivated in Section 5.3.1.

Two algorithms have been built. Both use the greedy strategy, but the level of accuracy and speed differs. Both are inspired by the work of Chen et al. [10]. The former is labelled as Basic Greedy BG, whereas the second Add-Remove-Move ADREMOG.

5.2.1 Basic Greedy algorithm

The BG algorithm was tested for the first time on WFLO by Chen et al. [10], which obtained good results. As the authors demonstrate, this algorithm offers room for improvement due to its simplicity. The reason for implementing this algorithm is to evaluate the performances of the simplest constructive algorithm possible, which, in turn, is also the fastest.

The following notation is adopted to present the algorithm. Let a be a vector, then $\tilde{a}$ is the corresponding set of a, and $|\tilde{a}|$ the cardinality of $\tilde{a}$. We define s as the index associated to a coordinate $z = [x, y]$ in V. The constant vector V encompasses all the locations z of the wind farm so that, if $s = 1$, the first element of V is indicated as $V[s]$. On the other hand, the variable vector S includes the indices s of the positions z selected for the layout Z. Then, layout Z can be determined by picking the selected indices from V.

Firstly, the BG randomly selects the position of one turbine. Indeed, as a solitary turbine does not experience wake losses, any position is optimal. Then, the BG builds a complete solution by adding at every iteration one turbine where it maximizes the AEP of the partial layout. In particular, the index s^* of the greediest position is chosen at each iteration among a set $\left\{1, 2, ..., \left|\tilde{V}\right|\right\} \setminus \tilde{S}$ which encompasses the indices of the placements that are still available. This procedure is undertaken by the function $\textsc{Best\_Turbine\_Position}(V, \tilde{S})$ (Pseudo-algorithm 1). The optimization process ends when the fixed number of turbines N_t is reached. Finally, the algorithm returns the coordinates of all the turbines in the layout. The structure of the algorithm is presented in

Pseudo-algorithm 2.

Algorithm 1: Best_Turbine_Position function

```
function BEST_TURBINE_POSITION($V, \tilde{S}$)
    $s^* := \arg\max_i \left(E_{\text{farm}}(\tilde{S} \cup \{i\})\right)$    where $i \in \left\{1, 2, ..., \left|\tilde{V}\right|\right\} \setminus \tilde{S}$
    return $s^*$
```

Algorithm 2: Basic Greedy algorithm

```
function BASIC_GREEDY($V$,$N_{\text{t}}$)
    Initialize : $S_1 :=$ RANDOM($1, \left|\tilde{V}\right|$) # Randomly selects one index from 1 to $\left|\tilde{V}\right|$
    for $j$ from 2 to $N_{\text{t}}$ do :
        $s :=$ BEST_TURBINE_POSITION($V, \tilde{S}_{j-1}$)
        $\tilde{S}_j := \tilde{S}_{j-1} \cup \{s\}$
    end for
    $Z :=$ elements in $V$ corresponding to indices in $\tilde{S}_{N_{\text{t}}}$
    return $Z$
```

As stated at the beginning of the current section, this algorithm outputs the worst results among the greedy algorithms. Changshui et al. [9] demonstrate the submodularity of $E_{\text{farm}}(Z)$. Submodularity is a property of set functions that can be used to assess the theoretical worst output of BG. The definition of a submodular function is the following [9]: given a set function $F : 2^{\tilde{V}} \to \mathbb{R}$, where $\tilde{V}$ indicates a finite set and $2^{\tilde{V}}$ the power set of $\tilde{V}$, F is submodular if $F(\tilde{A} \cup \{z\}) - F(\tilde{A}) \geq F(\tilde{B} \cup \{z\}) - F(\tilde{B})$ for any $\tilde{A} \subseteq \tilde{B} \subseteq \tilde{V}, z \notin \tilde{B}$. In turn, the submodularity of $E_{\text{farm}}(Z)$ indicates that the marginal expected power of any turbine in the layout either remains the same or decreases every time a new turbine is added. Indeed, if the wakes of the new turbine influence the surrounding turbines, these will reduce their power output. On the other hand, if the new turbine is placed far away, turbines will not be affected by the new turbine in the layout. Besides being submodular, the set function $E_{\text{farm}}(Z)$ is also monotone [9]. As a consequence, the worst result obtained through a greedy algorithm is capped at a percentage of an optimal solution Z^* [30]:

$$E_{\text{farm}}\left(Z\right) \geq \left(1 - \frac{1}{\text{e}}\right) \cdot E_{\text{farm}}(Z^*) \approx 63\% \cdot E_{\text{farm}}(Z^*). \tag{5.1}$$

Nevertheless, results show that such an algorithm performs much better than its theoretical worst case limit [9].

However, the quality of the layouts generated by BG has room for improvement. Indeed, BG places a turbine in the optimal position according to the partial layout of the previous iteration, but it ignores that the positions of the partial layout are not optimized for the extra turbine.

5.2.2 Add-Remove-Move algorithm

The Add-Remove-Move Greedy (ADREMOG) is an evolution of the BG. It is designed to compensate for the problem outlined at the end of the previous section concerning the greediness of the algorithm, which finds the best placement when adding a turbine to the partial solution, but does not modify the pre-existing layout according to the position of the newly added turbine. This ability is given to ADREMOG.

The optimization procedure is shown in Pseudo-algorithm 4, which makes use of the notation presented in the previous section. At every iteration, the algorithm places a turbine in the best position among a set of available locations. During the same iteration, each turbine is removed and moved to a greedier position, if it exists, by following the order they have been added to the partial layout. The removing and moving procedures are carried on by REMO(V, S) (Pseudo-algorithm 3), which adjusts the partial solution at every iteration. The adjustments continue until the layout does not change anymore. A layout that meets this condition is called *stable*. When a stable layout is reached, ADREMOG returns the complete layout if it consists of the desired number N_t of turbines. Otherwise, an extra turbine is added and the procedure is repeated.

Algorithm 3: Remove-move function

```
function REMO($V, S$)
    do :
        $T := S$
        for $k$ from 1 to $\left|\tilde{S}\right|$ do :
            $S[k]$ :=BEST_TURBINE_POSITION($V, \tilde{S} \setminus \{S[k]\}$)
        end for
    until $\tilde{S} = \tilde{T}$   # Condition to reach a *stable layout*
    return $S$
```

Algorithm 4: Add-Remove-Move Greedy algorithm

```
function ADREMOG($V$,$N_\mathrm{t}$)
    Initialize : $S[1]$ := RANDOM(1, $\left|\tilde{V}\right|$) # Randomly selects one index from 1 to $\left|\tilde{V}\right|$
    for $l$ from 2 to $N_\mathrm{t}$ do :
        $S[l]$ := BEST_TURBINE_POSITION($V, \tilde{S} \setminus \{S[l]\}$)
        $S$ := REMO($V, S$)
    end for
    $Z$ := elements in $V$ corresponding to indices in $\tilde{S}_{N_\mathrm{t}}$
    return $Z$
```

5.3 Combining PAM with greedy algorithms

As discussed in the previous chapter, PAM can be used to assess the AEP of a wind farm by summing up the power losses that turbines cause among them. This section proposes a method to combine PAM with a greedy algorithm. It allows to run wake models only once during WFLO. The method requires the discretization of the search space and employs strategies to maintain the problem unconstrained. These two aspects will be treated in the order thay were mentioned.

5.3.1 Selection of available locations

The manner in which the exploration of the domain is discretized affects the performance of the greedy algorithms. The exploration of the search space means how the vector $V = (z_1, z_2, ..., z_n)$ containing the possible wind turbine placements is determined. The literature provides two ways:

- Ozturk and Norman [32] opt for a continuous search space, resulting in infinite possible locations. At every iteration, the algorithm randomly selects a finite set $\{V\}$ of positions where the objective function is assessed.

- Changshui et al. [9] and Chen et al. [10] use a grid to discretize the search space and limit the number of available spots. At every iteration, the algorithm assesses the objective function in each of these positions.

Ozturk and Norman [32] motivate the decision by arguing that a mesh discretization limits the number of turbines that can be placed on the wind farm. Moreover, this choice implies further complexity, as the use of a mesh requires an analysis of the influence of the grid size. However, later research demonstrates that the discretization guarantees a more uniform exploration and gives the possibility to build deterministic algorithms [10].

We choose to discretize the search space by using a square grid. The turbines can be placed only at the grid intersections (see Figure 5.1). Not only this method has proven to be more effective, but it is also more appropriate for combining PAM and greedy algorithms. Indeed, the combined approach of PAM and greedy algorithm is as follows. Firstly, a set of possible positions is created. Second, fictitious turbines are placed in all the possible positions, and PAM is used to calculate the expected power loss caused by each one of the turbines on the others. Then, a greedy algorithm uses the information on the fictitious turbines to build a layout constructively. The advantage of this approach is that there is no need to re-run the wake model every time the greedy algorithm changes the layout. Indeed, PAM is based on the superposition of the expected power losses, which are known for any possible pair of turbines since they have been calculated using the fictitious turbines. Instead, if the method proposed by Ozturk and Norman [32] is used, the possible locations will change at every iteration. Therefore, this approach requires the PAM to assess the expected power loss of the fictitious turbines (N_t-1) times more with respect to the selected strategy.

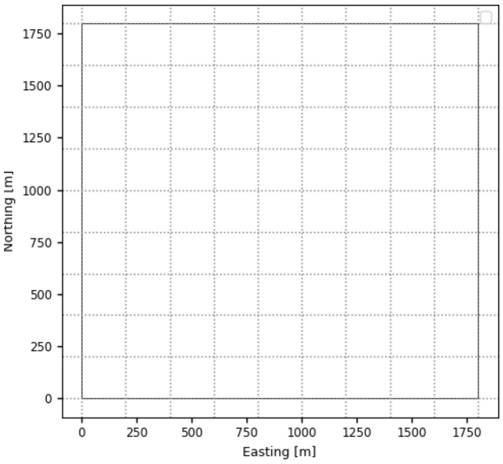


Figure 5.1: Grid discretization of the wind farm.

5.3.2 Constraint-handling

As described in 2.2.3, the WFLO is subject constraints, which regard the boundaries of the wind farm and the distance among turbines. Also, Section 5.3.1 describes how available points can be generated by using a square mesh. This section presents how the points generated by the mesh are selected to comply with the boundary constraints. Afterwards, the approach used to ensure the safety distance among turbines is treated.

The boundary's characteristics determine the approach used to avoid the placement of a turbine outside of the wind farm. First, consider a square wind farm, like the one proposed by Mosetti et al. [29]. The four vertices indicate the westernmost and easternmost coordinates of the wind farm, respectively, $x_{\min}$ and $x_{\max}$, as well as the southernmost and the northernmost coordinates $y_{\min}$ and $y_{\max}$. In this case, there is no need for recurring to boundary constraints. Instead, the following approach is used. First of all, the coarseness of the square mesh is set by choosing the side length of the cells. Then, one can count the maximum number of cells that enter side by side in the wind farm. Finally, the side length of the cell is adjusted by dividing $(x_{\max} - x_{\min})$ by the counted number of cells. This last step ensures that the search space is uniformly discretized. Now, consider a wind farm with an irregular shape. If the same approach is used, points beyond the boundaries will be selected, and compliance with the boundary constraints will not be ensured. Therefore, first, the wind farm site is treated as it is square-shaped, and points in $x_{\min} \leq y \leq x_{\max}$ and $y_{\min} \leq x \leq y_{\max}$ are generated through the square mesh. Then, each point is tested by using Equation 2.2. Hence, z is confined into the wind farm if the sum of the angles between all the consecutive vertices of the polygon corresponds to $360°$.

Therefore, selecting available points through a mesh can make WFLO unconstrained. Indeed, for both the case of a square wind farm and the irregular-shaped site, generating a set of possible positions before the greedy algorithm is launched ensures that each turbine of the final layout is located inside the boundaries. Also, the safety distance between turbines can be respected by selecting a grid size of the same, or larger, dimensions.

Nevertheless, this approach cannot ensure that the minimum distance is respected if the mesh for selecting the available points is finer than the minimum distance. In this case, the problem is maintained unconstrained by adding a static penalty to the objective function. In particular, expected power loss of a turbine that does not comply with the minimum distance is replaced with infinite value when searching for the greediest position. Indeed, such position is discarded in favour of valid positions, as they provoke a much smaller impact on the objective function.

The static penalty should be much larger than the expected power loss of the wind farm. Nevertheless, setting an infinite value has two advantages: first, it does not need to be adapted to the sizes of the wind farm; second, it is immediately recognizable if the minimum distance between turbines is not respected.

6
Results and Analysis

This chapter comments on the combined usage of the Pre-Averaged Model (PAM) and greedy algorithms. This approach is tested on an academic case study proposed by Mosetti et al. [29]. This so-called Mosetti's problem considers a simple, made-up wind resource and square-shaped wind farm boundaries.

Greedy algorithms are evaluated according to their speed of execution and the quality of the results. Nevertheless, the former parameter depends on the computer used to generate the layouts. Therefore, the relevant characteristics of the computational resource are summarized in the appendix. Moving to the quality of the layouts, the efficiency η is used. It is defined as the ratio between the AEP of a wind farm and the annual energy yield that would occur in the absence of wake loss:

$$\eta = \frac{E_{\mathrm{farm}}}{T \cdot N_{\mathrm{t}} \cdot P_{\infty}} \tag{6.1}$$

The use of a dimensionless parameter is preferred to AEP as it allows a better comparison among different cases, as well as a more precise analysis of the optimization algorithms. For instance, assume that two algorithms, whose performances are largely distinct respect to the quality of the results, displace a few turbines over a vast area. If the layouts are assessed in terms of AEP, the difference among them will be narrower than if many turbines are positioned in a small area, as in the former case wake losses are expected to have a low impact on energy production. Thus, one may wrongly conclude that the two algorithms achieve results of similar quality. On the other hand, η adds information on the results by indicating the proximity of the layout to its maximum theoretical AEP.

The reader is introduced to the Mosetti's problem in the first section. Then, Section 6.2 presents the results of an investigation aimed at identifying the main aspects that impact the proposed approach. Four aspects are analyzed. The first regards the placement of the first turbine added to the layout, as it determines the final output unequivocally. Then, the discretization of the search space, which also plays a role. Afterwards, Section 6.2 continues providing information on how the approach is affected by the number of wind rose sectors. At last, different wake models are tested. Then, the chapter moves to the validation of the proposed model, which is performed by comparing the Annual Energy Production AEP calculated through the PAM and the traditional approach. Finally, the last section summarizes the findings regarding the factors influencing the PAM.

6.1 Mosetti's problem

Mosetti et al. [29] pioneered the research on Wind Farm Layout Optimization (WFLO). Problems that they proposed in their research have been used by several authors as a reference point [17], [15]. Furthermore, Mosetti's case studies are a suitable starting point when testing a WFLO algorithm due to their simplicity.

Mosetti et al. [29] obtained optimized layouts using a genetic algorithm which employs a variable number of turbines and seeks for a cost of energy minimization. In order to test the optimization algorithm, they created three different problems: the first with one single wind direction; the second with multiple wind directions but constant wind intensity; the third with multiple wind direction and intensities. In this work, the first and second case studies are discarded as they are considered too far from reality. Instead, the third case study is used.

The Jensen model is employed to describe the wake deficit. Nevertheless, the paper does not explicitly indicate how it deals with partial waking. Here, the effect of partial wakes on the average wind speed at the rotor is taken into account and modelled as presented in Equation 3.17.

With regard to the wind resource, a description of the multiple wind direction and intensities problem is given below. The wind resource is binned in 36 wind rose sectors and 3 wind speeds. The characteristics of the wind rose are presented in Figure 6.1.

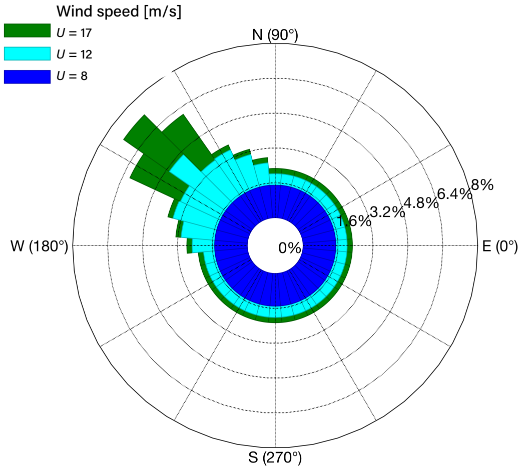
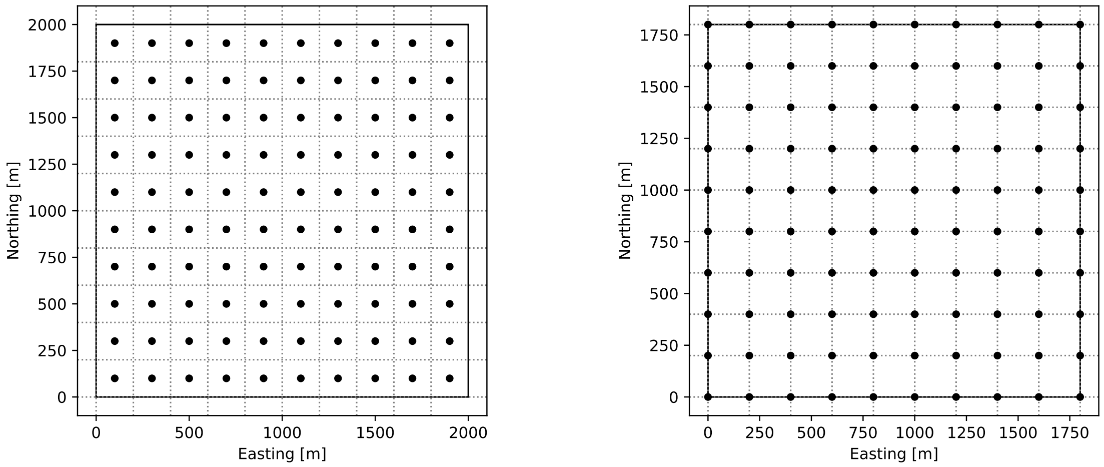


Figure 6.1: Mosetti's problem wind resource.

Moving to the wind farm geometry, it consists of a square of 2 × 2 km. A set of 100 possible positions is obtained by dividing the wind farm into 100 cells using a grid. Each available position is selected at the centre of each cell (Figure 6.2a). The size of the cell, whose side is $5D$ long, ensures safety distance between turbines. Nevertheless, this approach does not explore the whole search space fully, as the outermost region of the wind farm is not exploited. Also, the approach differs from the one used in the current work, which selects the available positions on the vertices of the cells (see Subsection 5.3.1). Therefore, the dimensions of the wind farm have been adjusted to 1.8 × 1.8 km in order to obtain the same problem site. The wind farm site and the selected available points are represented in Figure 6.2b.

Finally, an ideal turbine is employed. The characteristics of the turbine are given by its power curve (see Figure 6.3) and Table 6.1.



(a) Original.

(b) Adapted.

Figure 6.2: Comparison of the location of available positions for Mosetti's problem [29].

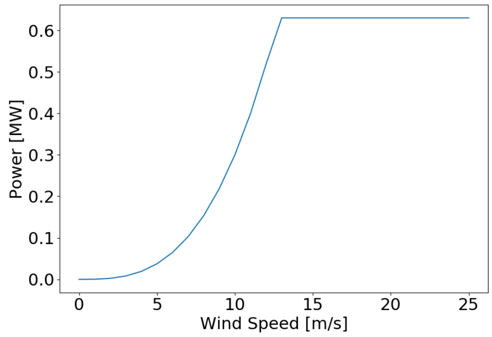


Figure 6.3: Mosetti's turbine power curve.

Variable	Value
Rotor diameter (D)	40
Hub Height (H)	60
Rated power (P_{rated})	0.63
Cut-in wind speed (U_{in})	0
Rated wind speed (U_{rated})	12.8
Cut-out wind speed (U_{out})	Not specified
Thurst coefficient (C_{T})	0.88

Table 6.1: Mosetti's turbine characteristics.

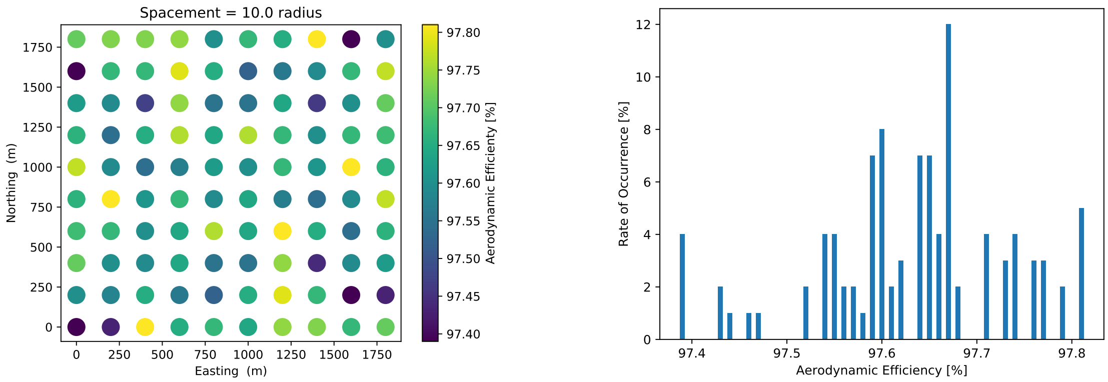
6.2 Results of the greedy algorithms

6.2.1 Impact of the initial placement

Both Basic Greedy (BG) and Add-Remove-Move Greedy (ADREMOG) are deterministic. In turn, the output layout does not vary for the same initial conditions. Nevertheless, the algorithms require to be initialized with the position where the first turbine is added. Indeed, the first turbine does not experience any wake. Therefore, the power loss is zero in every location, and the greediest position does not exist. Hence, the decision on the first placement is arbitrary, but it affects the efficiency of the output layout.

Knowledge of the influence of the first placement on optimality can speed up an algorithm. Indeed, investigating this aspect might help to assess the minimum number of initial positions that have to be evaluated to find a sufficiently good layout. Therefore, a mesh of 10 × 10 is created. Afterwards, BG and ADREMOG are asked to generate layouts for different initial placements until all the initial placements have been explored.

For BG, the impact of the first placement is relevant. On the left side, Figure 6.4 displays the efficiency of the obtained layout based on where the first turbine is placed. Looking at the figure, no pattern is visible that relates the initial position to better layouts. Often, there is even a relatively large discrepancy between the final result of neighbouring starting points. Furthermore, when looking at the distribution of the results (right-hand side of Figure 6.4), the number of different outcomes is considerable.

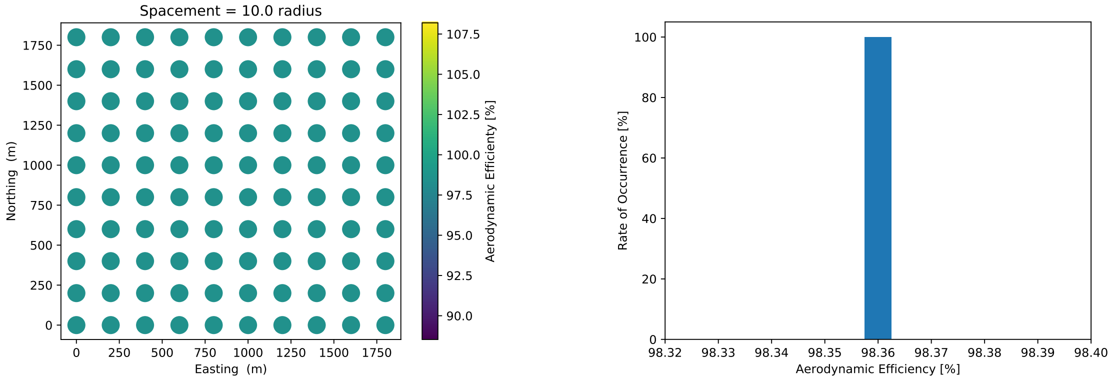


(a) Efficiency as a function of initial placement.

(b) Distribution of results.

Figure 6.4: Impact of the initial placement on BG.

Moving to the ADREMOG, the result is different. Indeed, as visible in Figure 6.5, the algorithm produces only one output for the given mesh. When the spacing among possible positions is reduced, ADREMOG does not maintain the peculiarity of generating only one layout. In any case, the range of outputs remains limited to a few layouts (Figure 6.6).

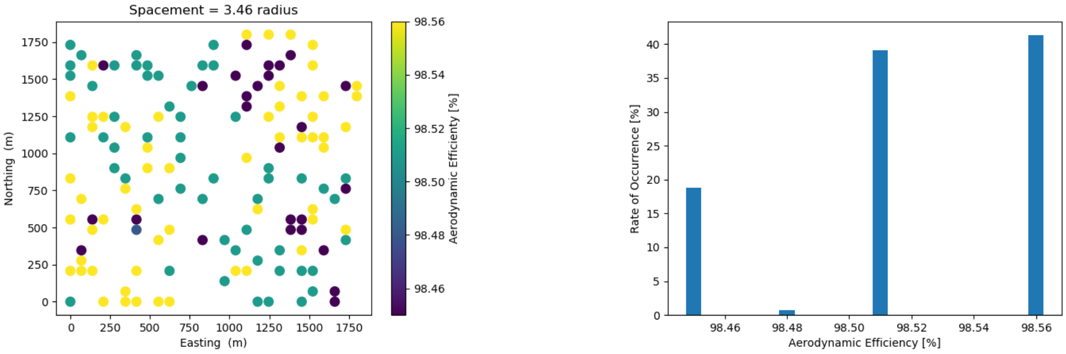


(a) Efficiency as a function of initial placement.

(b) Distribution of results.

Figure 6.5: Impact of the initial placement on ADREMOG for a 10 × 10 mesh.

Based on these results, it has been decided to generate the results by testing many random initial placements. As the ADREMOG is more likely to generate similar results and converges to results more slowly than BG, only a 5% of initial placements is investigated. For BG the percentage is taken to be 20%. All the results presented from now are based on these percentages, except for the graphs illustrating the speed of execution. In this case, the time refers to the generation of one single layout.



(a) Efficiency as a function of initial placement.

(b) Distribution of results.

Figure 6.6: Impact of the initial placement on ADREMOG for a 27 × 27 mesh.

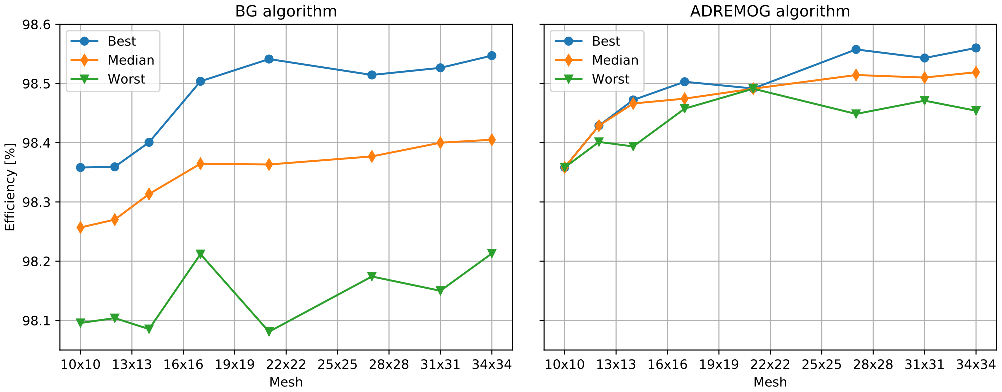


Figure 6.7: Efficiency as a function of the grid sizing.

6.2.2 Impact of grid spacing

To evaluate the impact of the grid sizing, both BG and ADREMOG are run with different meshes. From Mosetti's 10 × 10 mesh, finer grids are tested until reaching a 34 × 34 mesh. Afterwards, the data is processed to obtain the best, worst, and median efficiency values for each grid dimension. Figure 6.7 charts the trends of the efficiency.

Both algorithms generate the best layout with the finest mesh. Nevertheless, the improvement becomes negligible after the 17 × 17 grid for BG, whereas it is still evident in ADREMOG until 27 × 27. Then, the efficiency of the best layout reaches a plateau. With regard to the median efficiency of the layouts, its value keeps on improving constantly and slowly as the mesh becomes finer. Finally, the efficiency of the worst layouts fluctuates when the number of available positions varies.

Comparing the performances of BG and ADREMOG (see Figure 6.7), it appears that the latter does not improve much the efficiency of the best layouts obtained. Indeed, the blue lines are comparable. Nevertheless, ADREMOG narrows down the difference between the efficiency of the best and worst layouts. In turn, the outcome of the BG can fluctuate largely, whereas the ADREMOG produces more stable results, and the median efficiency of the layouts that it generates is better (see Figure 6.8).

Nevertheless, relocating the turbines comes at the price of slowing down the ADREMOG, as shown

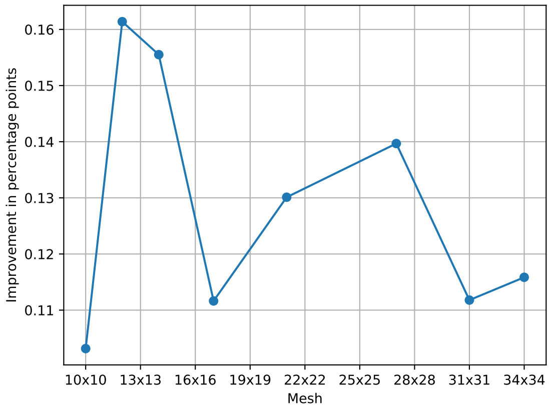


Figure 6.8: Improvement of the median achieved by ADREMOG over BG.

in Figure 6.9. Indeed, it appears that the ADREMOG is nearly four times slower than BG for the grid sizes investigated. In both cases, the time is affected by the number of available positions.

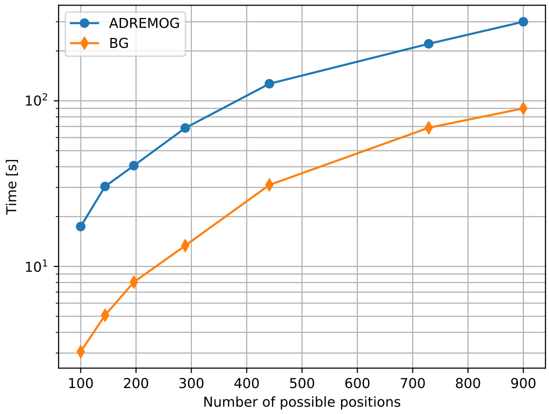


Figure 6.9: Time to generate one layout.

6.2.3 Impact of the number of wind rose sectors

Section 4.1 illustrates how wind measurements can be processed to create sector-wise Weibull distributions. The wind direction discretization can have an importaint impact on WFLO results. In particular, the size of the wind rose sectors $\Delta\theta$ can have a remarkable influence. An excellent example to motivate the above statement is provided by Feng et al. [14].

Consider three identical turbines, placed on the perimeter of a 400 m radius circle at the same distance among them (blue layout in Figure 6.10). A second layout is obtained by rotating the former by 15° (red layout in Figure 6.10). The small circles represent the space that the turbine rotor occupies. Assuming that the wind is blowing at a constant speed, and the wind directions are uniformly distributed, then it can be concluded that the two layouts must have the same power output.

Now, let the the wind resource be discretized in 12 wind rose sectors. Figure 6.11 shows the wake zones that the turbines would originate behind them if the Jensen wake model is employed. It appears that the blue layout is affected by the wake effects, whereas none of the turbines that

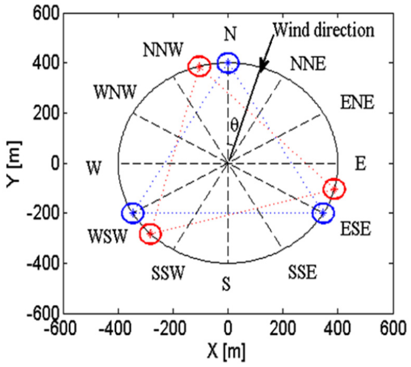


Figure 6.10: Two fictitious layouts. The red is rotated by 15° [14].

compose the red layout experiences any wake. Clearly, this result is incompatible with the fact that the two layouts are expected to generate the same power output. Hence, in this case the discretization provokes the red layout to be better than the blue one, although this result is not representative of reality.

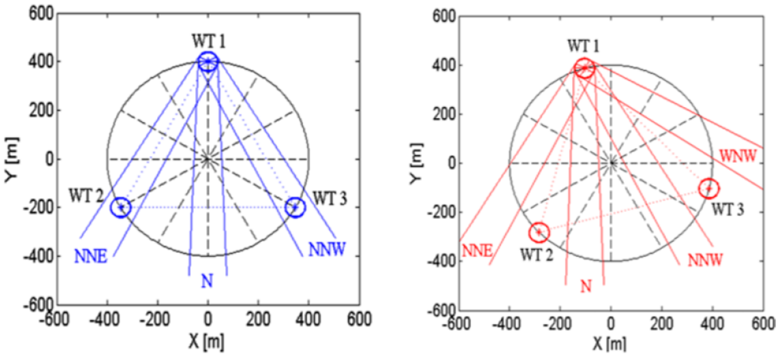


Figure 6.11: Wakes affecting the two layouts [14].

As a consequence, it is interesting to search for dependence between the layouts obtained and the number of wind rose sectors. Therefore, we progressively increase N_{wd} for different meshes. The subdivided wind direction and wind speed distributions are obtained using linear interpolation. Results are presented in Figure 6.12.

Limitations in the computational resource do not allow to expand the investigation to finer grids. Nonetheless, it is already visible that the value of η as a function of grid spacing reaches a plateau faster when the number of wind direction sectors is increased (see Figure 6.12). Indeed, both BG and ADREMOG keep on generating better layouts for $\Delta\theta = 10°$ as the grid becomes finer. On the contrary, the efficiency of the layouts generated for $\Delta\theta = 5°$ and $\Delta\theta = 2°$ look alike, and it is almost independent on grid spacing.

The cause of this behaviour is revealed when the effect of one turbine on its surroundings is plotted. Indeed, as discussed in Chapter 4, the PAM can be used to derive the expected power loss caused by a wake source as a function of vector $\vec{d}$. The plots (Figure 6.13) have been obtained between $5R$ and $25R$, which can be considered as the area of interest in a WFLO problem. Indeed, looking at Figure 6.13, some V-shaped indentations appear in the contour areas. These indentations are almost removed when the number of wind direction sectors N_{wd} increases. Also, the number of

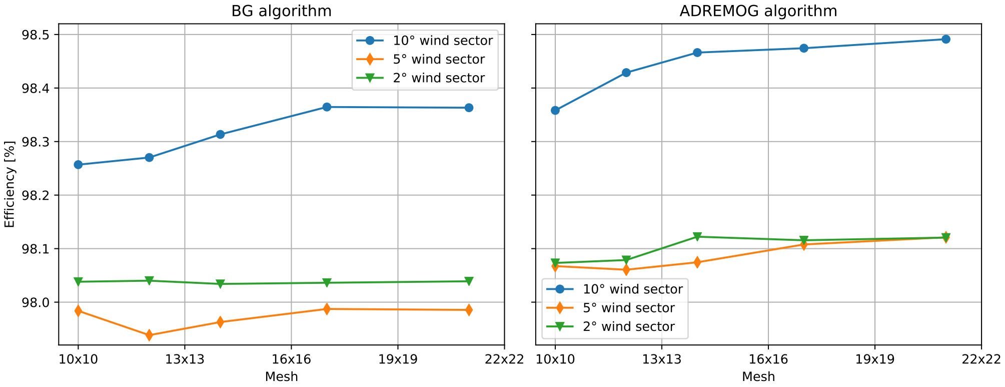


Figure 6.12: Median efficiency obtained with different wind direction sector width.

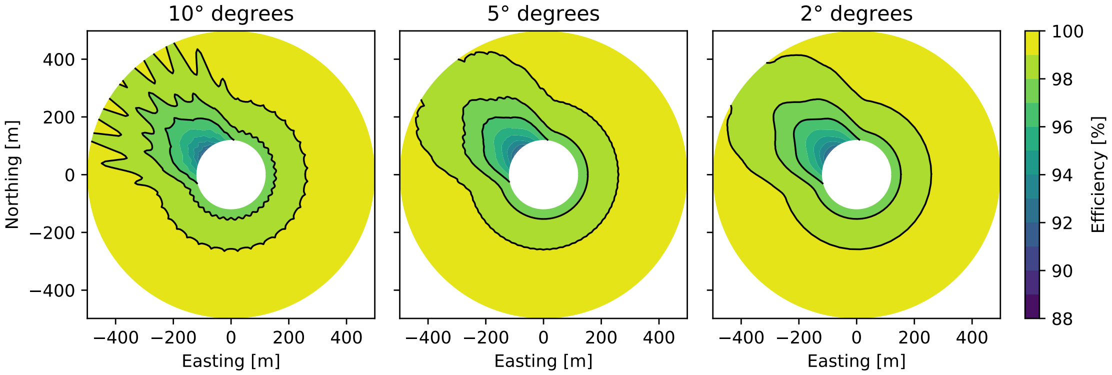


Figure 6.13: Expected efficiency around a wake source according to the Jensen wake model.

indentations equal N_{wd}. Such indentations are not consistent with the wind resource of Mosetti's problem. Hence, they are artificial.

Nonetheless, 72 wind directions are enough to realistically describe the wind resource. Increasing the number of wind directions further does not have a major impact on the model. Indeed, differences between η when $\Delta\theta = 5°$ and $\Delta\theta = 2°$ are slight.

6.2.4 Impact of the wake model

Processing the wake effects is the most time-consuming procedure that the PAM carries out when assessing the AEP. As discussed in Chapter 3, Engineering Wake EW models are chosen to limit the computational load. Some of these models describe the wake effects more precisely among the EW models, whereas others resort to simplifications.

Apart from the Jensen wake model, which has been employed to derive the results presented so far, two more wake models are tested. The first, called *Simplified Jensen*, is derived from the Jensen model. Indeed, the way the two models describe the trend of the wind speed profile inside the wake is analogous. Nonetheless, the interaction between wake and rotor differs substantially. Indeed, according to the Simple Jensen, a turbine is either fully waked or wakeless depending on whether its hub is waked or not. The second model is the *Gaussian* as described in Section 3.2.2. The effect of a wake source on the efficiency η of fictitious turbines placed on its surroundings is plotted.

Compared to the Jensen wake model, its simplified version introduces an additional source of dis-

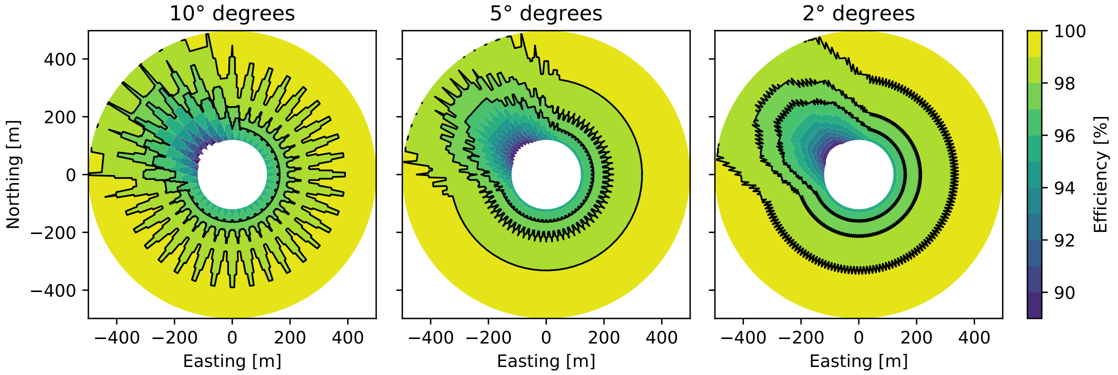


Figure 6.14: Expected efficiency around a wake source according to the Simplified Jensen wake model.

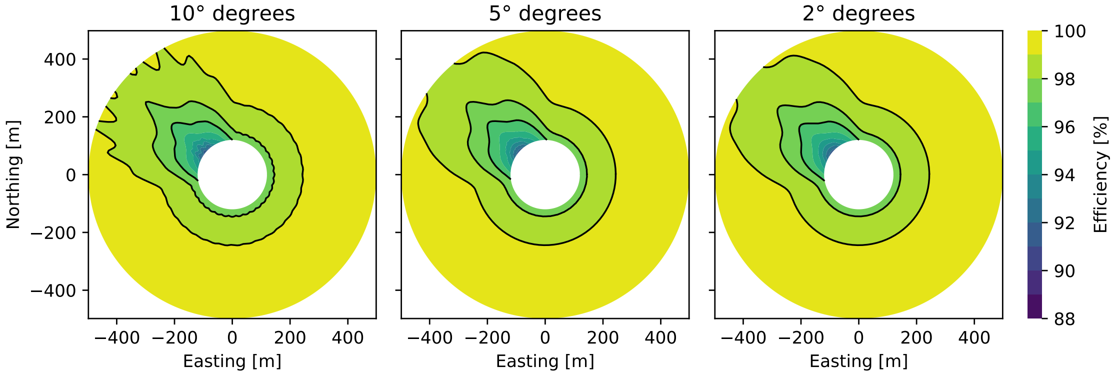


Figure 6.15: Expected efficiency around a wake source according to the Gaussian wake model.

continuity as partial waking is ignored. This discontinuity has a significant impact on the model. Indeed, the V-shaped indentations described for Figure 6.13 are more evident in Figure 6.14. As the number of wind direction sectors increases, the size of the indentations diminishes, but they are still visible even when $\Delta\theta$ equals 2°.

Instead, the Gaussian wake model describes the wake deficit through a continuous Gaussian distribution. This characteristic has a positive impact on the outcome of PAM. Indeed, when the number of sectors is increased to 72, the indentations totally disappear. No improvement is observed when progressing from 72 wind rose sectors to 180.

6.3 Validation

The PAM relies on assumptions, such as the averaging and superposition of the power loss, which have not yet been validated. To ensure that the PAM is reliable, a cross comparison is performed by evaluating the generated layouts' AEP with both the PAM and the traditional approach. The same wake model (Jensen's model) has been used for both frameworks. In the traditional approach, the total wind speed deficit of multiple wakes in a turbine's rotor is calculated as the root sum square of the deficits (Equation 3.21). Mosetti's layout [29] is taken as a reference (Figure 6.16).

When assessing the AEP of Mosetti's layout with the two approaches, some discrepancies occur

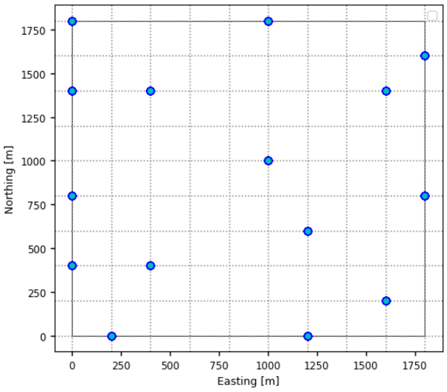


Figure 6.16: Mosetti's Layout [29].

Table 6.2: Value of the objective function for Mosetti's layout.

	AEP	Efficiency [%]
Traditional	43679.1	97.77
PAM	43558.5	97.50

(see Table 6.2). In particular, PAM overestimates the wake loss with respect to the traditional method. The difference in efficiency is about 0.3%.

Although this difference cannot be neglected, it is more interesting to see whether this discrepancy is typical or whether Mosetti's layout is an isolated case. Therefore, the best layouts obtained during the simulations discussed in Section 6.2.2 are evaluated with both the traditional method and the PAM. Results are presented in Figure 6.17, and show that the PAM always calculates the power loss in excess. Nevertheless, both ways of assessing the energy yield share the same trends. In particular, the difference between the efficiency assessed with the traditional approach and with PAM is nearly constant over different layouts. Therefore, the outcome of an optimization procedure that employs PAM will be consistent with one that uses the traditional approach.

As a result, Mosetti's layout can be used as a yardstick to evaluate the efficiency of the layouts generated by BG and ADREMOG. Below, the best layouts are shown for a 10 × 10 mesh. By comparing Table 6.2 and Figure 6.17 it appears that the greedy algorithms can improve the efficiency of Mosetti's layout by about 0.7 percentage points with the same mesh. The improvement increases as the mesh becomes finer.

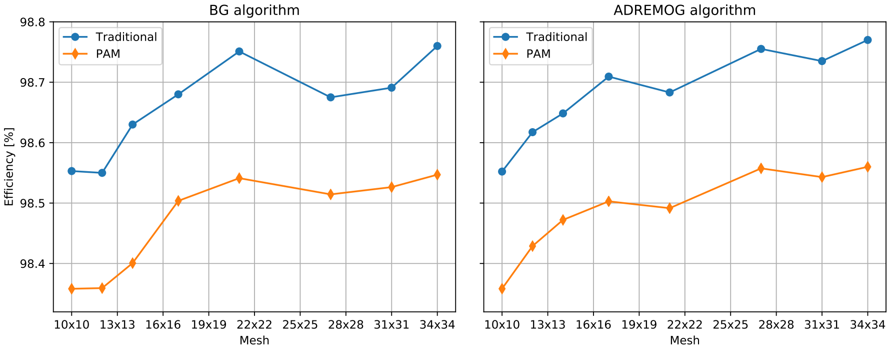
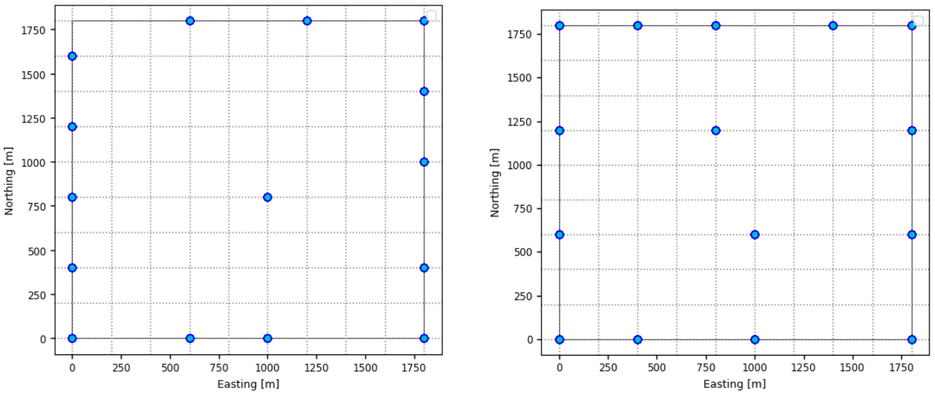


Figure 6.17: Efficiency of the best layouts assessed by the traditional method and PAM.



(a) BG's layout.

(b) ADREMOG's layout.

Figure 6.18: Best layouts generated for the Mosetti's problem.

6.4 Analysis of the Pre-Averaged Model

As presented in Section 6.2.3 and 6.2.4, the choice of the wake model and of the number of wind sectors affect the PAM. In particular, PAM conveys the presence of discontinuities introduced by binning the wind directions and the wake models, as discontinuities cause the presence of artificial zones with smaller wake loss. The impact of discontinuities can be reduced by choosing smoother wake models and by increasing the number of wind rose sectors. Hence, the wake model drives the choice of an optimal $\Delta\theta$. The size of the indentations on the model of the expected wake power around a wake source is used as an indicator of the impact that discontinuities have.

If the Simplified Jensen model is employed, it does not lead to realistic results even with a large number of wind rose sectors. On the other hand, results obtained with both the Jensen and the Gaussian model are trustworthy for $\Delta\theta \leq 5°$. Indeed, the optimization algorithms place turbines in locations where the wake loss is minimized, but such location might take advantage of the discontinuities introduced in the PAM by wind direction binning or by the wake model. When N_{wd} grows, and wake effects are more representative of the real case, the optimization algorithms reduce their dependency on grid spacing (Figure 6.19).

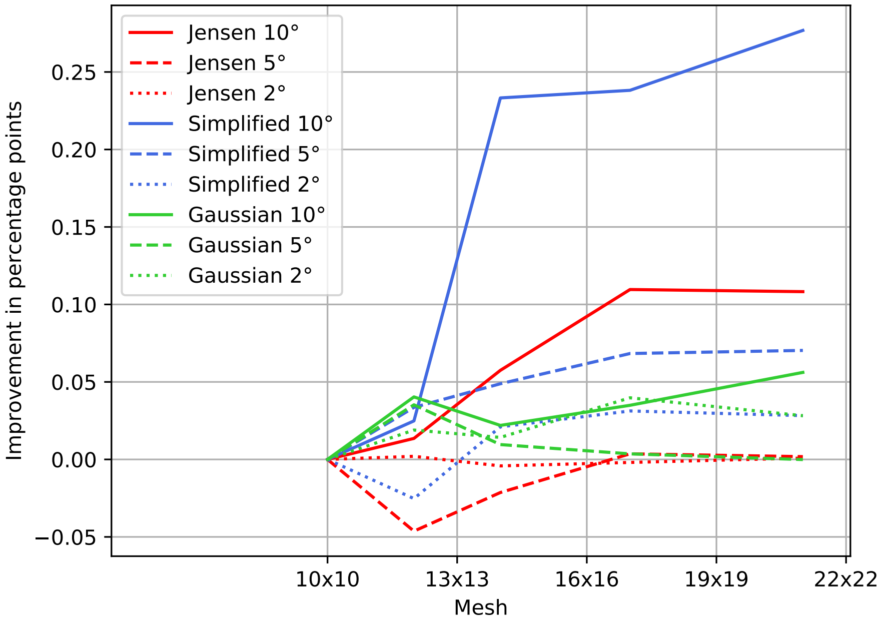


Figure 6.19: Improvement of the median efficiency for the layouts obtained by BG for different wake models and N_{wd}.

7

New Case Studies: Borssele

This chapter continues presenting the results of the Basic Greedy (BG) and Add-Remove-Move Greedy ADREMOG. However, the optimization strategies are tested on more challenging cases. Indeed, Mosetti's problem, introduced in Section 6.1, represented a useful starting point for testing the performance of both BG and ADREMOG. Nevertheless, doubts remain on whether the proposed optimization strategy is a valid option when the complexity of the problem increases. In particular, the aim here is to explore the performances of the combined employment of Pre-Averaged Method PAM and greedy algorithms when the boundaries of the wind farm become irregular, the number of turbines increases, and the wind farm site is divided into many parcels. Two cases have been selected to introduce extra complexity to the implementation of the PAM progressively. Real wind farm sites are chosen for both cases. The first case study considers a parcel of Borssele III, whereas the second regards the totality of the concessions of Borssele III and IV.

Also, Borssele III and IV were selected as reference wind farm site for two investigations conducted in the context of International Energy Agency's (IEA's) Task 37 on Systems Engineering in Wind Energy. The purpose of Task 37 is to boost the research on wind farms from a holistic perspective by improving the collaboration among organizations and research groups, as well as providing sets of good practices, analytical tools, and reference models [2]. In particular, the two cases on Borssele III and IV were recently released to expand the results collected in an article by Baker et al. [4] regarding the selection of wake models and optimization strategies for Wind Farm Layout Optimization WFLO. We choose to participate to IEA's Task 37, which further justifies why Borssele has been elected among other wind farm sites. Indeed, the participation in this research will allow to the combined use of PAM and greedy algorithms to other cutting-edge WFLO algorithms.

First, the two cases are presented. Second, the extra challenges posed by the new WFLO problems on the greedy algorithms, as well as the proposed solutions, are discussed. The final section is dedicated to presenting the outcomes of the optimization algorithms.

7.1 Presentation of the new case studies

Borssele is a project for a future cluster of offshore wind farms, that is about to be built at more than 22 km off the coast of Zeeland. The totality of the project area has been split into 5 concessions. Only Borssele III and IV are used in this work. In particular, these two concessions were awarded to Blauwwind, a consortium that joins Shell, Eneco, Van Oord and Diamond Generating Europe Limited [7]. A totality of 77 Vestas V164-9.5 MW will be installed [3]. With a rated capacity of 731.5 MW and an AEP of approximately 300 GWh, this wind farms will provide electricity to more than 800.000 households [21].

This thesis refers to the two case studies as IEA 3 and IEA 4. The former takes into consideration the easternmost parcel of Borssele III. Indeed, Borssele III consists of two separate parcels divided by a forbidden area due to the presence of cable routes [31]. The aim of the case study is to maximize the AEP by adjusting the placement of 25 turbines. Instead, the second case study regards the maximization of the AEP of 81 wind turbines in the concessions of Borssele III and IV, which consist of 5 different parcels.

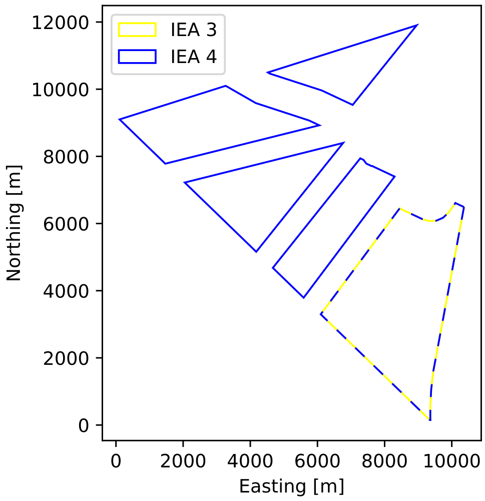


Figure 7.1: Boundaries of Borssele III and IV.

Both case studies employ the same wind resource, which is described by 20 wind rose sectors and 20 wind speed values. The characteristics of the wind resource are summarized in Figure 7.2.

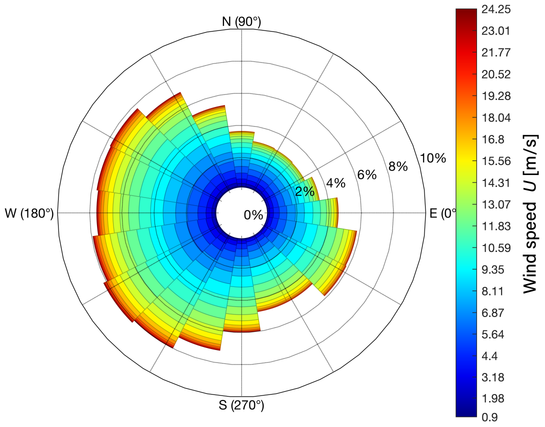


Figure 7.2: Wind resource for IEA I and IEA II.

Finally, the IEA 10 MW offshore reference turbine [8] is used. The power curve of the turbine is presented in Figure 7.3, whereas its specifications are gathered in Table 7.1

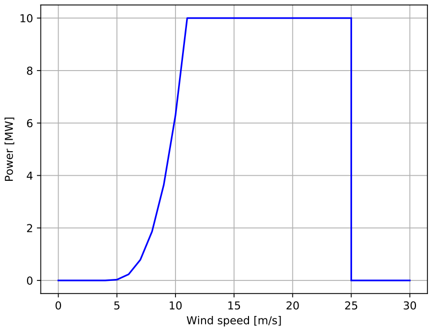


Figure 7.3: IEA's turbine power curve.

Variable	Value
Rotor diameter (D)	198
Hub Height H	119
Rated power P_{rated}	10
Cut-in wind speed U_{in}	4
Rated wind speed	11
Cut-out wind speed U_{out}	25
Thurst coefficient C_{T}	0.88

Table 7.1: IEA's turbine characteristics.

7.2 Adaptations to the new challenges

The use of PAM in combination with greedy algorithms has proved to be fruitful for Mosetti's problem (see Chapter 6). Nevertheless, the IEA's cases introduce obstacles, as they require to deal with irregular boundaries, larger layouts, and multiple parcels. This section illustrates the countermeasures that have been identified to these three factors.

7.2.1 Irregular boundaries

The greedy algorithms select locations that minimize the power loss by testing a set of possible positions $\{V\}$, which correspond to the intersections of a mesh constructed inside the boundaries of the wind farm (see Subsection 5.3.1). The employment of squared mesh for the discretization of the search space is undoubtedly suitable for the Mosetti's wind farm boundaries (Figure 6.4), but it appears to be ineffective for irregular shapes. Indeed, the available positions are not well distributed on the wind farm, as Figure 7.4 reveals. In particular, the selection through a mesh ignores the vertices of the wind farms as well as the points in the boundaries.

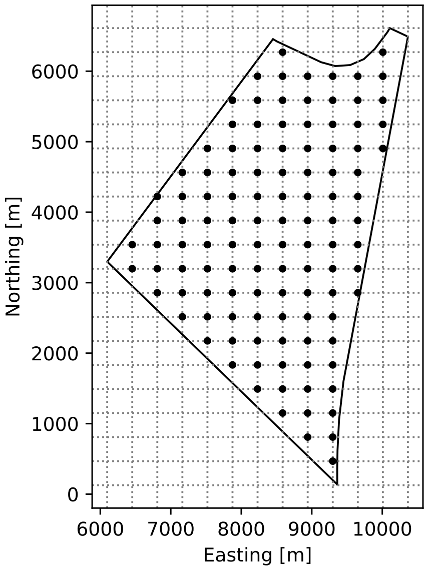
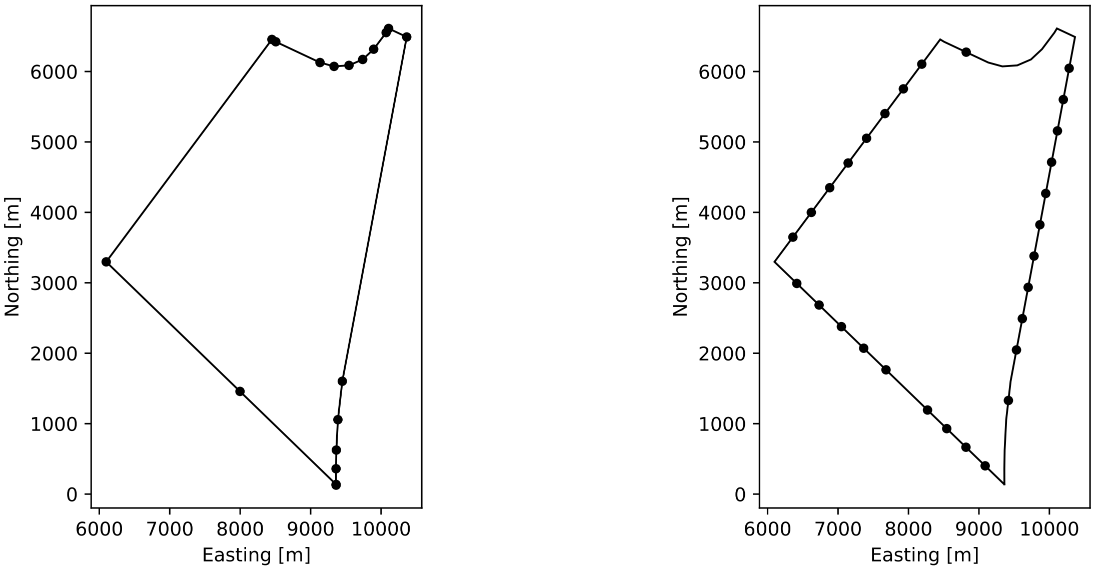


Figure 7.4: Possible positions generated as in Mosetti's problem.

To solve this problem, more points are chosen (Figure 7.5). The vertices of the wind farm are automatically selected. Furthermore, the distance between consecutive vertices is calculated. If the distance is larger than the grid spacing, intermediate points along the boundary are added to the available locations. The distance among intermediate points corresponds to the grid spacing.



(a) Vertices.

(b) Intermediate points on boundaries.

Figure 7.5: Example of positions added to V for IEA 3.

Looking at Figure 7.6, it is visible that the selection of points in the boundaries enhances the efficiency of the layouts. As the size of V increases, the effect diminishes. Indeed, the squared mesh becomes more and more suitable for irregular boundaries as its spacing narrows down.

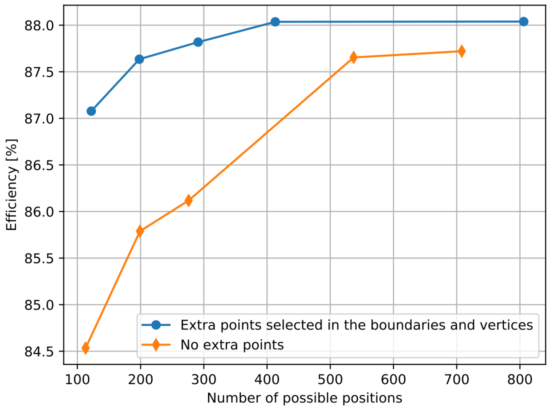


Figure 7.6: Comparison between BG's outputs with and without the inclusion of boundary points in V.

7.2.2 Large number of turbines

Big wind farms, composed of hundreds of turbines, have been built in the past decade. Seeing as the dimensions of the wind farms are rising, a well-working WFLO algorithm should cope with larger N_t than in the past. The new case studies progressively test this ability. Indeed, compared to Mosetti's problem, IEA 3 and 4 increase the number of turbines N_t from 15 to, respectively, 25 and 81.

A preliminary analysis is done to investigate the impact this challenge could have. As Figure 7.7 shows, the time needed to construct a layout for the Mosetti's problem sky rockets when ADREMOG is used and N_t increases. Indeed, as the dimension of the partial layout enlarges, the time required for every iteration rises, since the algorithm relocates more turbines than in the previous iteration. On the other hand, BG iterates at a constant pace. Hence, it linearly depends on N_t.

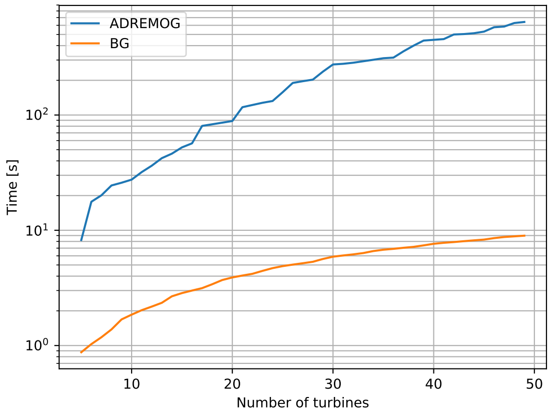


Figure 7.7: Time to generate a layout with 100 available positions as a function of the number of turbines.

To summarize, on the one hand, the number of turbines does not affect adversely the BG primarily. Hence, BG is expected to be suitable for layout optimization of large wind farms. On the other hand, the dependency of ADREMOG on N_t could make it fail to provide layouts in a reasonable time, especially when the discretization of the search space becomes finer. Comparing the performances of the two algorithms, it appears that the re-location stage is a double-edged sword: it improves the efficiency of the layouts, but it narrows down the applicability of the algorithm to low values of N_t.

Bearing in mind that the interest of this work is on an algorithm that balances between speed and quality of the solution, heuristic rules are applied to speed up the algorithm. Investigating on the displacement phase of ADREMOG, it comes out that this stage is excessively time-demanding due to two reasons.

Firstly, the re-location of the turbines is carried on by until it does not vary anymore (see Pseudo-algorithm 3). This strict requirement may be superfluous. Indeed, each partial layout generated at every constructive iteration is a locally optimal solution. Nevertheless, it will likely lose this property when a new turbine is added. Therefore, search for local optima in the first $(N_\mathrm{t} - 1)$ iterations may be unnecessary, especially seeing as the re-location impacts the performances of the code causing fluctuation on time required to generate layouts. Indeed, Figure 7.8 shows that the distribution of the time needed to generate a layout is wide. In particular, the choice of the initial position may lead to a smaller or larger number of iterations.

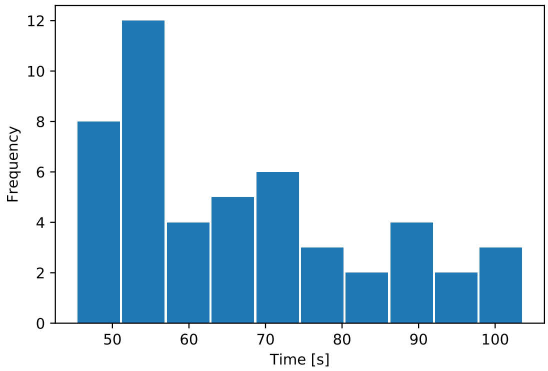


Figure 7.8: Time distribution of ADREMOG to generate a layout for Mosetti's problem and 100 possible positions.

Secondly, ADREMOG tests for re-location all the turbines that compose the partial solution in search of a greedier position. Nevertheless, it has been observed that a small percentage of turbines is re-located for each iteration, whereas most of them remain in their original position. This behaviour can be observed in Figure 7.9. It charts the rate of displacement for ADREMOG, defined as the ratio between the number of turbines that are moved and the number of turbines N_t that compose the partial solution.

$$\text{rate of displacement} = \frac{\text{number of turbines that are moved during relocation}}{N_\mathrm{t}} \tag{7.1}$$

Indeed, the rate of displacement rarely rises above 20%. Also, although it is not visible from Figure 7.9, the same analysis proved that the re-adjusting stage does not produce any change 33% of the times. This characteristic can undoubtedly be exploited to optimize the algorithm.

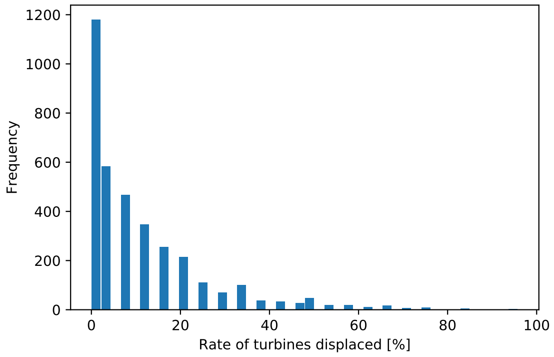


Figure 7.9: ADREMOG's rate of displacement.

Based on these two considerations, a new algorithm, named ADREMOG II, has been devised. The structure of ADREMOG II is shown in Pseudo-algorithm 6, which makes use of the notation introduced in Subsection 5.2.1. The core elements of the algorithm do not differ broadly from its precursor. Indeed, it still consists of two stages: at first, a turbine is added to the layout and, then, the layout is adjusted. Nevertheless, it presents two main changes. The first change regards the research of a stable layout at each iteration during the re-adjustment stage. Indeed, ADREMOG II does not loop, and the re-location is performed just once. Not only this variation speeds up the algorithm, but it avoids fluctuations in the time required to generate a layout. Moving to the second change, not all the turbines are tested for re-location. In particular, ADREMOG II tests for a greedier position only for a restricted list L of turbines that compose the partial layouts. Function CREATE_LIST(Z) generates the list L. Given a layout, it first assesses the marginal expected power output for each turbine or, in other words, the gain in expected farm power that the turbine provides. Secondly, the turbine with the smallest and the one with the largest marginal expected power are identified. The two turbines are inserted, respectively, as the first and last element of L. Furthermore, the interior of the list is obtained by randomly choosing 20% of turbines in the layout Z.

Algorithm 5: CreateList function

```
function CREATE_LIST($V, \tilde{S}$)
    $s^{\max} := \arg\max_s \left(\mathbb{E}(P_{\text{farm}}(\tilde{S} \setminus \{s\})\right)$    where $s \in S$
    $s^{\min} := \arg\min_s \left(\mathbb{E}(P_{\text{farm}}(\tilde{S} \setminus \{s\})\right)$    where $s \in S$
    for $j$ from 1 to $round(0.2 \cdot N_{\text{t}})$ do :
        $L_{\text{core}}\,[j] :=$ RANDOM($\tilde{S}$)      # Randomly selects one element in S
    end for
    $L := [s^{\min}, L_{\text{core}}, s^{\max}]$
    return $L$
```

Algorithm 6: ADREMOG II

```
function ADREMOG_II($V, N_{\text{t}}$)
    Initialize : $S\,[1] :=$ RANDOM($1, \left|\tilde{V}\right|$) # Randomly selects one index from 1 to $\left|\tilde{V}\right|$
    for  $i$ from 2 to $N_{\text{t}}$ do :
        $S\,[i] :=$ BEST_TURBINE_POSITION($V, \tilde{S}$)
        $L :=$ CREATE_LIST($V, \tilde{S}$)
        for  $l \in L$ do :
            Find $k$ such that $S[k] = l$
            $\tilde{S} := \tilde{S} \setminus l$
            $S[k] :=$ BEST_TURBINE_POSITION($V, \tilde{S}$)
        end for
    end for
    $Z :=$ elements in $V$ corresponding to indices in $\tilde{S}$
    return $Z$
```

Undoubtedly, the ambition of L is to provoke as many re-locations as possible without excessively compromising the speed of execution of the algorithm. With this in mind, the dimensions and the order of L have to be chosen carefully.

Although the re-adjustment loop removal modifies ADREMOG II substantially, the percentage of turbines that compose L determines whether its performances resemble BG or ADREMOG the most. The increase in the length of L results in a more accurate but slower algorithm. A value of 20% has been chosen as it speeds up the algorithm by nearly five times than when re-locating the wind farm in its totality. More investigation of this aspect may lead to a tailored optimal value.

The other factor influencing the performances of ADREMOG II is the order in which turbines are tested for re-location. An example is given to support this statement. Assume that the turbine added at last to the layout is the first one that the algorithm investigates for re-location. As the position of the last turbine is already optimized for the existing layout, its marginal power may correspond to the largest. Therefore, it will not be moved. Instead, an ideal ordering should aim at triggering the re-location by testing at first the turbines whose position is more likely to change. Finally, a greedier position should be searched for the last-added turbine. Further investigation would likely result in a more effective ordering than the one proposed in this thesis. Nevertheless, the deadline imposed by IEA's Task 37 required to take decisions in a short time. Therefore, ordering the wind turbines according to their marginal power primarily aims at pointing out one possible path of improvement rather than at giving a consolidated strategy. Nevertheless, some recommendations are given in the conclusive chapter.

Moreover, it appears that the marginal expected power of a turbine depends on its position as well. In particular, turbines that experience fewer wakes have higher marginal expected power. As illustrated in Figure 7.10, these turbines are usually placed in the boundaries and in the vertices of the wind farm. In turn, marginal power could be efficiently used as an indicator to develop new strategies for testing turbines for re-location to further speed up ADREMOG II. If turbines with the highest marginal power are moved, then the algorithm will better explore the outermost parts of the wind farms. On the other hand, moving turbines with the smallest marginal power will adjust the layout in the inner regions of the wind farm.

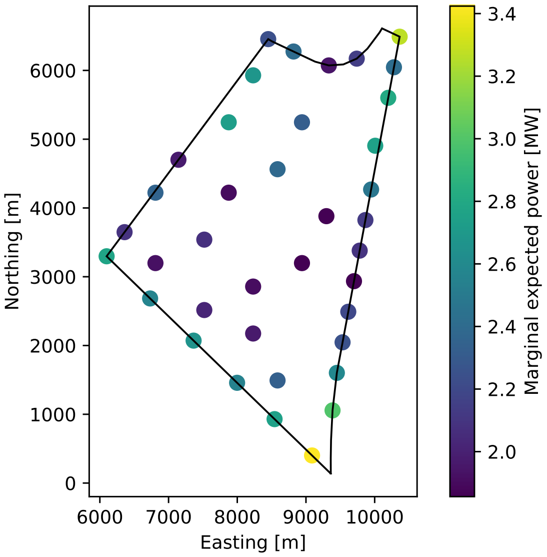


Figure 7.10: Marginal expected power of each turbine for IEA 3. 40 turbines have been placed.

7.2.3 Multiple parcels

A WFLO algorithm may underperform when a wind farm is composed of multiple parcels. IEA 4 gives the possibility to test the response of the greedy algorithms to this extra complexity. Indeed, complexity rises because the optimization algorithms not only have to place the turbines into each parcel optimally but also distribute the turbines among them. In particular, some algorithms might find it challenging to move a turbine from one parcel to another.

To analyze whether the proposed approach is affected by this extra complexity, the Pseudo-Gradients (PG) optimization algorithm created by Quaeghebeur [35] is employed. The PG algorithm requires to be initialized with a complete layout. The initialization layout is generally obtained by selecting the position of N_t turbines through a hexagonal mesh. Afterwards, it uses pseudo-gradients to find positions with higher expectation over the wind resource by "pushing" the turbines away from wakes.

Although a better analysis might result from the results of IEA Task 37's case studies, a preliminary investigation shows that ADREMOG II and PG algorithm obtain layouts with similar efficiency when IEA 3 is taken into consideration. Nevertheless, ADREMOG II generates better layouts for IEA 4. Finally, a combination of the two approaches is also tested by initializing the PG algorithm with the layouts obtained by ADREMOG II. As presented, the joint use of the two optimization strategies is beneficial (Table 7.2).

Table 7.2: Efficiency of the best layouts obtained with ADREMOG and PG algorithm.[1]

Case study	ADREMOG II	PG	ADREMOG II + PG
IEA 3	90.55%	90.91%	91.03%
IEA 4	72.16%	70.71%	73.88%

From the latter consideration, it emerges that discretization might limit the accuracy of ADREMOG II with respect to methods that employ a continuous search space. Nevertheless, as results are comparable for IEA 3, the fact that ADREMOG II generates better results for IEA 4 indicates that it copes more effectively with multiple parcels than the PG algorithm.

To conclude, subdivisions in parcels does not challenge the greedy algorithms. Indeed, they distinguish among points belonging to different parcels neither in the initialization of the available points nor in the constructive stage. Concerning the former aspect, the discretization is undertaken for all the parcels with the same strategy. Consequently, the search space is uniformly divided, as the number of available points in each parcel is proportional to its area, perimeter, and number of vertices. Then, all the available points are gathered in V, and evaluated according to their expected power irrespective of the parcel it belongs to.

7.3 Results

This section presents the results obtained for IEA 3 and 4. Compared to Chapter 6, the results of the optimization algorithms are only shown as a function of the number of available points. Indeed, outcomes regarding the influence of the number of wind directions and wake models on PAM are expected to be equivalent to the Mosetti's problem. Hence, the greedy algorithms, respectively, BG, ADREMOG, and ADREMOG II, are initialized with progressively enlarged V. After generating layouts by testing a 20% of initial positions for BG, 2% for ADREMOG and 5% for ADREMOG II, results are processed to obtain the highest, the lowest, and the median value of the efficiency. Instead, figures that illustrate the speed of execution of the algorithms refer to the generation of one single layout. The characteristics of the computers employed in the thesis can be found in the appendix.

7.3.1 IEA 3

The efficiency of the layouts generated for IEA 3 by BG, ADREMOG, and ADREMOG II is illustrated in Figure 7.11. Three conclusions can be drawn. First, the trends of the best and median efficiency are monotonically increasing for each of the optimization algorithms. For BG and ADREMOG, values increase sharply until V is expanded to about 400 available positions, but the improvement is limited as V is further enlarged. On the contrary, the accuracy of ADREMOG II keeps on improving at a more constant pace. Secondly, ADREMOG II provides a range of efficiency that is narrower than BG, but broader than its precursor ADREMOG. And last but not at least, the partial displacement stage run by ADREMOG II is not enough to achieve the same

[1] The values are not calculated through PAM, so they are not suitable for a direct comparison with the values given elsewhere in this chapter.

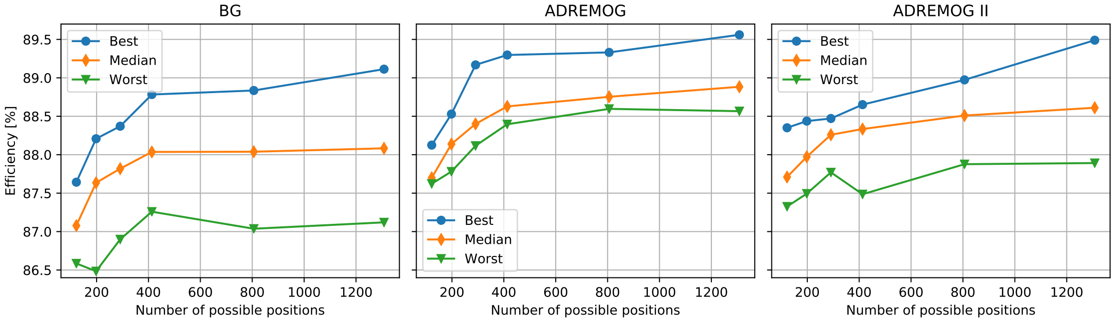


Figure 7.11: Efficiency as a function of the number of available points.

quality of results as ADREMOG but does improve the outcome with respect to BG (Figure 7.12).

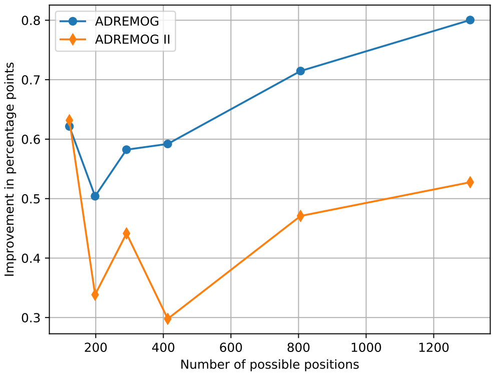


Figure 7.12: Improvement of the median achieved over BG.

The best layouts, which have been obtained when the number of available points is the highest, are shown in Figure 7.13.

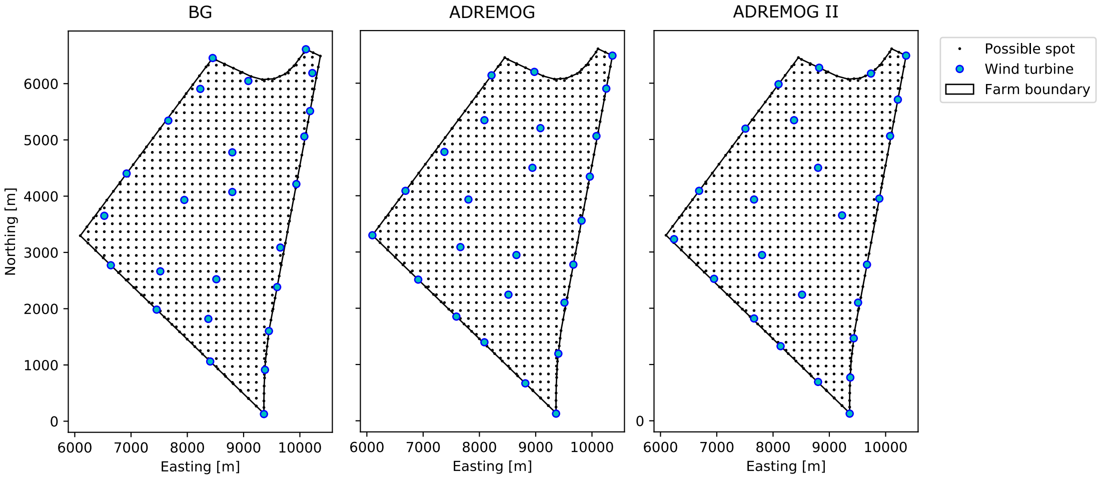


Figure 7.13: Layouts generated for IEA 3.

To recap, results reflect that ADREMOG II is a hybrid between BG and ADREMOG. Indeed, the efficiency of the layouts and the time to generate them is a trade-off between the other two greedy algorithms. Surely, limiting the number of displacements impacts the accuracy of the algorithm with respect to ADREMOG, but speeds up the generation of the layouts (Figure 7.14).

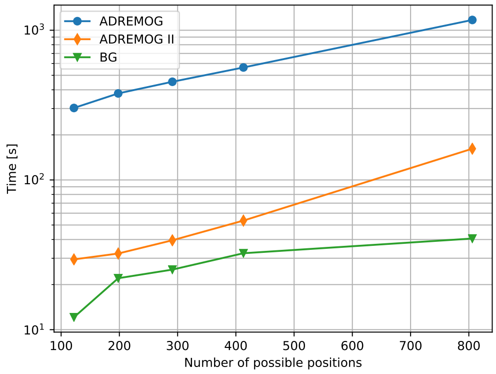


Figure 7.14: Time to generate one layout.

7.3.2 IEA 4

Figure 7.15 charts the efficiency of the layouts obtained for IEA 4 by BG and ADREMOG II. The same conclusions mentioned for IEA 3 case study apply to IEA 4. In short, layouts generated with BG still present a wider range of efficiency than ADREMOG II due to the displacement stage. Moreover, the efficiency of the results takes advantage of the re-location procedure undertaken by ADREMOG II.

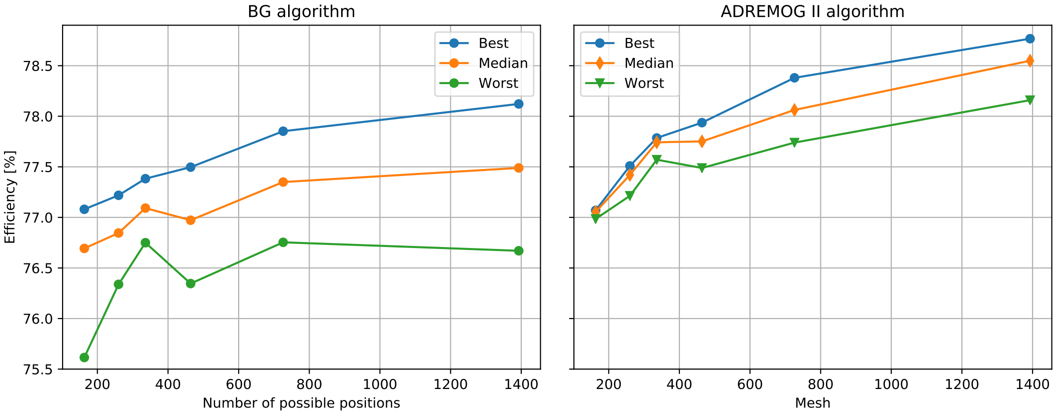


Figure 7.15: Efficiency as a function of the number of possible positions.

However, new information can be gathered by comparing the two cases and, in particular, Figure 7.12 and 7.16. Indeed, the improvement due to re-location is more remarkable when applied to IEA 4. Higher values of improvement might be a consequence of the more substantial wake loss that occurs in the current case study. Indeed, a larger wake loss should give the algorithms more room for improvement. Nevertheless, algorithms produce narrower ranges of efficiency for IEA 4 despite the increasing complexity of the problem.

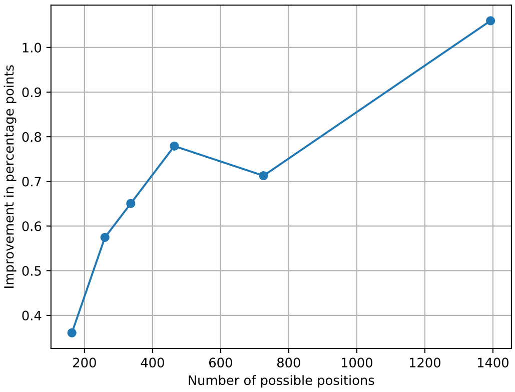


Figure 7.16: Improvement of the median achieved over BG.

A significant result concerns the time to derive one layout (see Figure 7.17). As usual, BG is by far the fastest, but ADREMOG II manages to generate layouts in a reasonable time if one considers the improvement that the latter produces over the former. Indeed, the modifications devised for ADREMOG II limit the adverse impact of large N_{t} on the speed of execution.

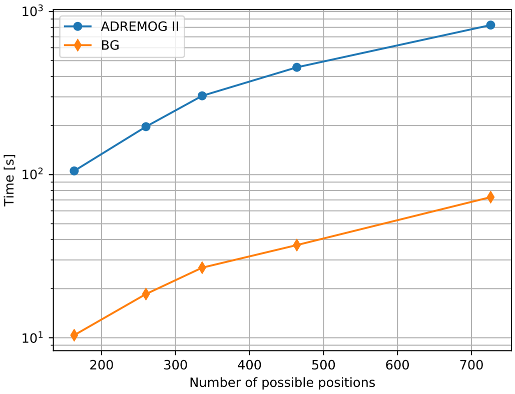


Figure 7.17: Time to generate one layout.

The best layouts encountered are presented in Figure 7.18. As the reader might have noticed, the results of ADREMOG are not shown for IEA 4. Indeed, its Achilles heel, hence N_{t}, makes the algorithm unsuitable for IEA 4. The derivation of a population of results significant enough to allow a comparison between ADREMOG and the other greedy algorithms would have been too time-demanding. Nevertheless, some layouts have been generated. Seeing as this work looks for a balance between speed and accuracy, Table 7.3 justifies why a more in-depth analysis of ADREMOG has not been undertaken. Indeed, the time to generate a layout is much larger than for the other methods.

Table 7.3: Efficiency of the layouts and ADREMOG's time of execution for IEA 4.[2]

Generated layout	Efficiency η [%]	Time [s]
Layout I	78.14	39970
Layout II	78.31	56648
Layout III	78.08	34575

[1] The number of initial possible positions corresponds to 726.

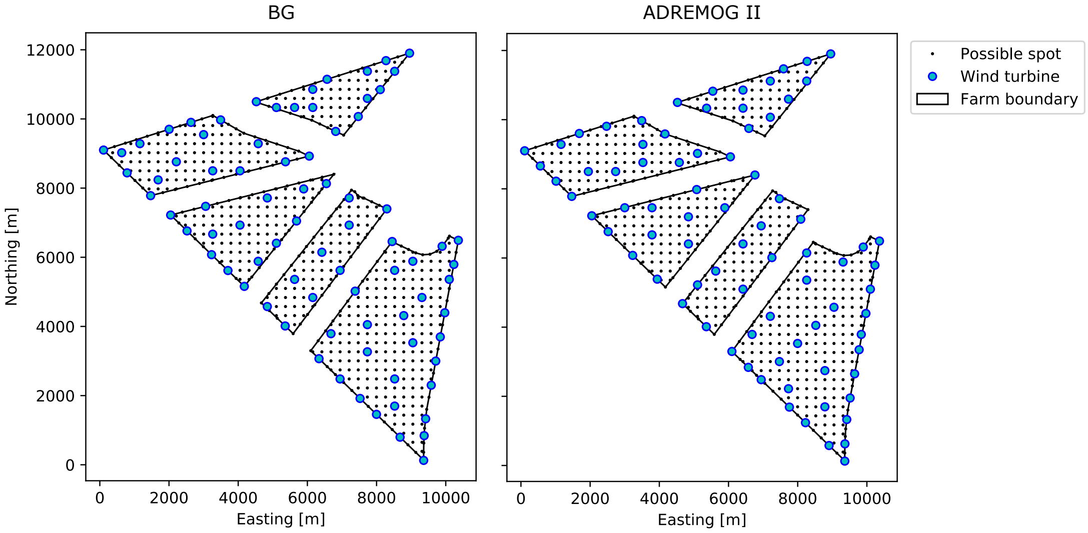


Figure 7.18: Best layouts generated for IEA 4.

8 Conclusions and Recommendations

In this research, the Pre-Averaged Model (PAM) is introduced to reduce the computational load of Wind Farm Layout Optimization (WFLO). With respect to optimization approaches that use the traditional method to assess the Annual Energy Production (AEP), the proposed method avoids re-processing wake models when generating layouts. The PAM is combined with different greedy algorithms separately, which build a layout constructively. The optimization strategy uses grid discretization and a static penalty to make the WFLO unconstrained. Performances of the optimization strategies are tested on three case studies. Case studies are used to, first, answer the research questions (Section 8.1) and, then, give recommendations on the use of PAM, on possible ways to accelerate the algorithms, and on how to expand the research (Section 8.2).

8.1 Conclusions

Outcomes of the case studies allow answering the research questions:

(1) *Is PAM applicable to the WFLO?*

The present study reveals that the PAM can be used to calculate the AEP within WFLO. Furthermore, it produces trustworthy results. On the one hand, when comparing the AEP assessed by the PAM and the traditional approach over various layouts, the two frameworks coincide in the selection of the best layout. On the other hand, they output slightly different AEP values, although the same wake model is employed. Nevertheless, the discrepancy is almost constant for any of the layouts assessed.

As a result, assumptions used by PAM and, specifically, the linear superposition of power losses, can also be employed in WFLO.

(2) *What is the dependency of the PAM on wind resource binning and the wake model?*

The research concludes that the selection of the wake model and the number of wind directions have a significant impact on PAM. Indeed, artificial zones where a wake target turbine would have lower expected power appear around a wake source turbine. These zones, which are V-shaped indentations, enlarge when more substantial discontinuities are introduced. The use of a continuous wake model (e.g., Bastankhah and Porté-Agel (BP)) diminishes the impact of the indentations with respect to a discontinuous wake model (e.g., Jensen model). The same positive phenomenon is observable when the number of wind directions increase.

The sensitivity of the PAM has an impact on greedy algorithms when the two frameworks are combined. If discontinuities are considerable, greedy algorithms take advantage of the discontinuities to pick positions which are less waked. The AEP of the layouts increases as the discretization of the search space becomes fine, whilst the improvement is artificial. On the other hand, increasing the number of available locations does not produce remarkable improvements if the number of wind directions is above 72 and either Jensen's wake model or BP's wake model is used. Instead, the improvement is still noticeable when adopting 72 wind rose sectors and a simplified Jensen model, which ignores partial waking.

Therefore, a proper selection of the wake model and the number of wind rose sectors is essential. Avoiding discontinuities results in a more realistic assessment of the energy yield. Furthermore, as discontinuities loose relevance, functional layouts can be obtained with a coarser grid discretization. Thus, the performances of the greedy algorithms also benefit, as the fewer the number of spots the greedy algorithm considers, the faster the generation of the layouts.

(3) *Is it possible to trade off quality of the results and speed of execution?*

The class of greedy algorithms offers different opportunities to balance the quality of the results and the time needed to generate them. Indeed, the research shows how greedy algorithms benefit from discretizing the wind farm with fine grids. Also, the the layouts' efficiency depends on the placement of the first turbine. It implies that several initial positions should be investigated before selecting the best layout. Nevertheless, finer grids require more time to obtain one layout, whereas the exploration of many initial points requires the generation of likewise layouts. Therefore, an acceleration of the algorithms can be obtained at the cost of quality by selecting a coarser mesh and fewer initial placements.

Nevertheless, this thesis identifies more elaborate solutions to balance quality and speed. The use of a re-adjustment stage improves the placement of the turbines with respect to the last one added. It is performed by testing for a greedier position all the turbines that compose a layout. Moving turbines to greedier positions positively impacts the quality of the layouts but protracts the optimization procedure. The research proposes different strategies to trade off these two elements. It is observed that layouts with the highest median efficiency are obtained when seeking locally optimal layouts at every iteration. This approach corresponds to the Add-Remove-Move Greedy (ADREMOG) algorithm, which extends the re-adjustment stage until a stable layout is reached. The Basic Greedy (BG) algorithm follows a different approach, as it does not perform any re-adjustment. The resulting greedy algorithm is fast, but the quality of the results is generally lower than the ADREMOG. Finally, we trade off quality and time by intervening on the nature of the re-adjustment stage. In particular, the number of turbines that are tested for re-location can be restricted. As this restricted group is enlarged, the quality of the layouts increases at the cost of time of execution. The same occurs if the re-adjustment is performed once for every constructive iteration rather than iterating until a stable layout is reached. Both the two updates are implemented in ADREMOG II. Moreover, ADREMOG II allows the user to reach a trade-off between speed and accuracy by adjusting the percentage of turbines that is tested for re-location.

(4) *How does a WFLO algorithm based on PAM perform with respect to algorithms that use a traditional approach to calculate the AEP?*

This question is not fully answered. Indeed, only Mosetti's case study allows comparing the proposed strategy directly with other optimization algorithms. It appears that the greedy algorithms can improve the efficiency of Mosetti's layout by about 0.7 % using the same discretization of the wind farm. Also, the time required is much shorter. The improvement increases as the grid becomes finer. Nevertheless, this comparison is biased by the fact that the computers evolved enormously in the two last decades. Moreover, Mosetti et al. [29] state that no attempt was done to optimize their code for speed. Moving to case studies IEA 3 and 4, no direct comparison to other optimization strategies is done in this reseach. However, pending results of the IEA's Task 37 might help to assess the response of the proposed approach in terms of quality and time with respect to other algorithms.

Nevertheless, it is worth to draw conclusions on the performances of the optimization algorithms that are developed in this thesis. It emerges that none of the optimization algorithms overcomes the others in absolute. The most suitable algorithm should be chosen case by case, depending on the characteristics of the problem, as well as on the focus of the user: either the accuracy of the results or the speed of execution.

For Mosetti's case study, employment of BG and ADREMOG might lead to comparable results in the same timespan, despite the latter generating layouts with higher median efficiency. Indeed, the two algorithms obtain similar best layouts as a function of the grid spacing. Furthermore, BG is nearly four times faster than ADREMOG. As faster greedy algorithms allow evaluating more initial positions, BG can generate more layouts than ADREMOG and then select the best.

IEA's case studies test ADREMOG II as well. For IEA 3, BG's and ADREMOG II's best layouts have lower efficiency than ADREMOG's. Indeed, the advantage of a complete re-location stage is evident when the complexity of the wind farm site grows. In this case, the highest efficiency of the layouts may compensate for ADREMOG being the slowest approach. ADREMOG benefits from the complete re-location stage in IEA 3 but, on the other hand, it makes the algorithm inappropriate for IEA 4 due to the large dimensions of the wind farm. Indeed, the derivation of the results is limited by the time required. For the latter application, ADREMOG II has proved to be the best, as it improves the results obtained by BG but is much faster than ADREMOG.

8.2 Recommendations

This section starts identifying good practices for PAM usage. As the relevance of speeding up WFLO is emphasized throughout this research, the second part provides some food for thought on possible strategies that could accelerate the generation of layouts. To conclude, this section suggests possible directions to expand the research in the future.

8.2.1 Recommendations for designers

Combining PAM and greedy algorithms can result in fast and reliable algorithms, which are suitable for multi-parcel wind farm sites and irregular boundaries. Depending on the focus of the designer and the characteristics of the final layout, three different algorithms can be employed. If the speed of generation is the driving parameter, BG is preferable. On the other hand, better layouts can be sought by ADREMOG at the cost of an extended time of execution. ADREMOG has shown to be suitable for layouts of up to 25 turbines, but one should avoid it for larger layouts. In this case, updates introduced in ADREMOG II make this algorithm the most suitable. Indeed, it increases the efficiency of the layouts obtained by BG but speeding it up with respect to ADREMOG.

The designer can also intervene on the characteristics of the optimization procedure to trade off speed and quality. Outcomes of complex problems keep on improving as the number of available positions increase, and a plateau is not been reached in this research. Hence, a fine discretization of the search space slows the algorithm down. Moreover, if one is searching for high quality layouts, it is suggested to test several initial positions before selecting the best one. Nevertheless, ADREMOG and ADREMOG II require few initial placements to be investigated. On the other hand, multi-start approach is beneficial for BG primarily as the range of outcomes is wider.

It is also suggested to create a set of possible spots for the turbines initially and to not modify them. The reason is twofold. First, it allows processing the PAM just once. Otherwise, PAM has to be re-run every time the set of possible placements changes. Second, it avoids dealing with boundary constraints during the optimization procedure. Indeed, compliance to boundary constraints can be tested immediately after selecting the available spots in the wind farm. Instead, a static penalty is handy in combination with greedy algorithms to make turbines respect proximity constraints.

Selection of the available spots can be performed using grid selection. Nevertheless, this approach may not generate sufficient spots in the outermost region of a wind farm with irregular boundaries. As a full exploration of the wind farm area is beneficial, it is advised to include the vertices of wind farm sites and points in its boundaries into the set of available positions.

If Jensen's model or BP's model is used, binning wind measurements into at least 72 direction sectors results in reliable optimization. Adoption of simpler wake models, which introduce further discontinuities, has to be compensated by increasing the number of wind directions. Nevertheless, the choice of more accurate Engineering Wake models does not compromise the speed of execution with respect to highly simple models, as PAM processes them just once. Therefore, the use of a continuous EW model should be preferred. Reducing discontinuities in the wake modelling allows using coarser discretization as wake deficits are more uniform over the wind farm.

Finally, the selection of many possible locations and an elevated number of wind directions caps the computational capacity of the device processing the PAM. Therefore, selection of a number of wind directions above the recommendations might be counter-productive. The number of available points should be increased progressively so that the computer response can be monitored. The appendix presents the characteristics of the computational resources employed in this research.

8.2.2 Recommendations for accelerating greedy algorithms.

Submodularity of the expected power output $\mathbb{E}(P_{\text{farm}})$ is the property with the highest potential for accelerating greedy algorithms. As the current approach tests all the available spots in search of the greediest position, submodularity could reduce the number of spots investigated. Changshui et al. [9] explain how to take advantage of this feature. Firstly, a list of available placements is generated, and the first turbine is placed randomly. Each available spot is associated with the marginal expected power that placing a turbine there would ensure to the wind farm. This list of marginal expected power values is sorted in descending order so that the first element in the list corresponds to the greediest position. Then, the layout is enlarged by placing a second turbine there. As the submodularity of $\mathbb{E}(P_{\text{farm}})$ holds, the marginal expected power associated with each point becomes smaller after adding an extra turbine. The algorithm evaluates the marginal power of the first element in the list and sorts the list again. If the re-assessed marginal power is still the initial element of the list, the corresponding spot is the greediest. Hence, the algorithm adds the position to the layout.

Submodlarity is immediately applicable to BG. In this case, assigning a static penalty may be the best strategy to deal with proximity constraints. Indeed, locations that do not respect this constraint will be assessed once and pushed to the tail of the list containing the marginal expected power values.

Nevertheless, submodularity does not stand if re-adjustment of the layout is performed (e.g., ADREMOG and ADREMOG II). For these algorithms, different strategies that make the turbines respect the minimum distance can be experimented. One approach could be implementing a strategy which enables or removes possible locations from a flexible set, continuously adapted to the layout. In particular, if an extra turbine is placed, spots within the minimum distance are removed from the set of available positions. On the contrary, if the same turbine is removed, positions that were not respecting proximity constraint are included to the set again. A flexible set requires extra steps in the algorithm but avoids recalculating the marginal expected power of positions that are not suitable. Hence, an acceleration of the algorithm is not guaranteed.

8.2.3 Recommendations for future research

This research has demonstrated that linear superposition of the power loss can be used to model the effect of multiple wake source turbines on a wake target turbine. This *ad hoc* approach differs from the traditional: summing up the square of the wake deficits. Although PAM can be employed for WFLO, it overestimates wake losses with respect to the common practice. In turn, the common practice often makes use of EW models, which tend to exaggerate the impact of wakes. As a result, PAM doubly overestimates the wake loss. Therefore, any research aimed at assessing the discrepancy between PAM and real measurements would be useful. Alternatively, different superposition strategies that reduce the impact of wake effects can be tested (e.g., root sum square of the wake losses).

Furthermore, this research employs marginal power to select a list of turbines tested for re-location. Nevertheless, no effort has been made to optimize the two characteristics of the list: dimensions and order. Future research could find better strategies to exploit this feature in ADREMOG II.

Finally, combining the Pseudo-Gradients (PG) algorithm devised by Quaeghebeur [35] to greedy algorithms offers room for improving the outcomes of this research. On the one hand, PG could benefit from greedy algorithms being better at placing the turbines over multiple parcels. On the other, PG could improve greedy algorithms' layouts by refining them. Indeed, greedy algorithms do not allow small adjustments unless discretization is highly dense. The combined use of greedy algorithms and PG could result in a faster algorithm than when employing fine discretization and could also result in higher quality layouts.

Appendix: Computational Resources

This research has made use of two computers. As speed of execution depends on the processor employed, this appendix summarizes their characteristics. It allows to evaluate the time of execution of the proposed method by considering the characteristics of the computational resource. Also, this appendix justifies the use of different computers and indicates which one is employed when deriving the results.

This research combines two procedures. In the first stage, PAM obtains the mutual wake losses among fictitious turbines placed in every possible position of the wind farm. Then, a greedy optimization algorithm employs this information to generate layouts. On the one hand, the first stage is the most computationally intense. In particular, the virtual memory of a computer saturates depending on coarseness of the wind farm discretization, the number of wind speeds and wind directions. Indeed, increasing the wind resource discretization has to be compensated with a coarser selection of possible spots to allow the computer to process the results. On the other hand, the second stage is generally the most time demanding.

A MacBook Air 6.2 (see Table A.1a) is employed up to about 800 possible positions, 3 wind speeds, and 36 wind directions. Its memory limitations do not allow us to fully investigate the dependency of the greedy algorithms on the number of possible positions and wind direction sectors. Therefore, the more powerful ASUS X580VD is employed (see Table A.1b). It can push forward the discretization to nearly 1400 points under the same wind resource characteristics. Nevertheless, as the latter computer is faster than the former, we only present the time of execution of the outcomes generated by the MacBook Air to maintain uniformity among the results.

Table A.1: Hardware summary

Processor	
Model	Intel Core i5
Speed	1.3 GHz
Cores	2 units

RAM	
Type	DDR3
Size	4 GB
Speed	1600 MHz

(a) MacBook Air 6.2

Processor	
Model	Intel Core i7 770HQ
Speed	2.8 GHz
Cores	4 units

RAM	
Type	DDR4
Size	16 GB
Speed	2400 MHz

(b) ASUS X580VD

Bibliography

[1] 2030 Energy Strategy. Retrieved May 17, 2019 from the official site of the European Commission: `https://ec.europa.eu/energy/en/topics/energy-strategy-and-energy-union/2030-energy-strategy`.

[2] IEA Wind Task 37 Systems Engineering. Retrieved Aug 2, 2019 from Windbench website: `https://windbench.net/iea37`.

[3] Financiering windparken Borssele III en IV rond. 29 Jun 2016. Retrieved from Maritiem Netherland website: `http://www.maritiemnederland.com/nieuws/financiering-windparken-borssele-iii-en-iv-rond/item2843`.

[4] N. F. Baker, A. P. Stanley, J. J. Thomas, A. Ning, and K. Dykes. Best Practices for Wake Model and Optimization Algorithm Selection in Wind Farm Layout Optimization: Preprint. 2019. `doi:10.2514/6.2019-0540`.

[5] R. J. Barthelmie, K. Hansen, S. T. Frandsen, O. Rathmann, J. G. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. S. Politis, and P. K. Chaviaropulos. Modelling and measuring flow and wind turbine wakes in large wind farms offshore. *Wind Energy*, 12(5):431–444, 2009. `doi:10.1002/we.348`.

[6] M. Bastankhah and F. Porté-Agel. A new analytical model for wind-turbine wakes. *Renewable Energy*, 70:116–123, 2014. `doi:10.1016/j.renene.2014.01.002`.

[7] M. Beunderman. Shell bouwt tweede windpark bij Borssele. *NRC*, 12 Oct 2016. Retrieved from `https://www.nrc.nl/nieuws/2016/12/12/shell-bouwt-tweede-windpark-borssele-5767847-a1536318`.

[8] P. Bortolotti, H. C. Tarres, K. L. Dykes, K. Merz, L. Sethuraman, D. Verelst, and F. Zahle. IEA Wind TCP Task 37: Systems Engineering in Wind Energy-WP2. 1 Reference Wind Turbines. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2019. Retrieved from `https://www.nrel.gov/docs/fy19osti/73492.pdf`.

[9] Z. Changshui, H. Guangdong, and W. Jun. A fast algorithm based on the submodular property for optimization of wind turbine positioning. *Renewable Energy*, 36(11):2951–2958, 2011. `doi:10.1016/j.renene.2011.03.045`.

[10] K. Chen, M. X. Song, Z. Y. He, and X. Zhang. Wind turbine positioning optimization of wind farm using greedy algorithm. *Journal of Renewable and Sustainable Energy*, 5(2): Article ID 023128, 2013. `doi:10.1063/1.4800194`.

[11] R. Cirillo. Offshore Wind Farm Layout Optimization: smart design strategies over real case studies. Master's thesis, Delft University of Technology, 2018. Retrieved from `http://resolver.tudelft.nl/uuid:4beb2cbd-dd1b-425d-be01-8f01fe1bfa79`.

[12] A. Crespo, J. Hernández, and S. Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. *Wind Energy*, 2(1):1–24, 1999. doi:10.1002/(sici)1099-1824(199901/03)2:1<1::aid-we16>3.3.co;2-z.

[13] C. M. Engelen. The nonlinear effect of combining uncertainties on the energy yield of an offshore wind farm. Master's thesis, Delft University of Technology, 2015. Retrieved from http://resolver.tudelft.nl/uuid:232e2b1a-95f4-4e2e-910f-4f2f3158cb69.

[14] J. Feng and W. Shen. Modelling Wind for Wind Farm Layout Optimization Using Joint Distribution of Wind Speed and Wind Direction. *Energies*, 8(4):3075–3092, 2015. doi:10.3390/en8043075.

[15] J. Feng and W. Shen. Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, 78:182–192, 2015. doi:10.1016/j.renene.2015.01.005.

[16] F. Glover and M. Laguna. Special Tabu Search Topics. *Tabu Search*, pages 227–266, 1997. doi:10.1007/978-1-4615-6089-0_7.

[17] S. A. Grady, M. Y. Hussaini, and M. M. Abdullah. Placement of wind turbines using genetic algorithms. *Renewable energy*, 30(2):259–270, 2005. doi:10.1016/j.renene.2004.05.007.

[18] J. Herbert-Acero, O. Probst, P. E. Réthoré, G. Larsen, and K. Castillo-Villar. A Review of Methodological Approaches for the Design and Optimization of Wind Farms. *Energies*, 7(11):6930–7016, 2014. doi:10.3390/en7116930.

[19] IRENA. Electricity storage and renewables: Costs and markets to 2030. Technical report, International Renewable Energy Agency, Abu Dhabi, United Arab Emirates, 2017. Retrieved from https://www.irena.org/publications.

[20] IRENA and CEM. Renewable Energy Auctions: A Guide to Design. Technical report, 2015. Retrieved from https://www.irena.org/publications.

[21] K. Jaspers. Zwitserse financier stapt in Zeeuwse windparken Borssele 3 en 4. *FluxEnergie*, 08 Jan 2018. Retrieved from https://www.fluxenergie.nl/zwitserse-financierder-stapt-in-zeeuwse-windparken-borssele-3-en-4/.

[22] N. O. Jensen. *A note on wind generator interaction.* Risø National Laboratory, 1983. Retrieved from https://findit.dtu.dk/en/catalog/2389477356.

[23] N. Johnstone, I. Haščič, and D. Popp. Renewable energy policies and technological innovation: Evidence based on patent counts. *Environmental and Resource Economics*, 45(1):133–155, Jan 2010. doi:10.1007/s10640-009-9309-1.

[24] I. Katic, J. Højstrup, and N. O. Jensen. A Simple Model for Cluster Efficiency. *EWEC'86. Proceedings*, 1:407–410, 1987. Retrieved from https://findit.dtu.dk/en/catalog/2389451418.

[25] 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.

[26] P. B. S. Lissaman. Energy effectiveness of arbitrary arrays of wind turbines. *Journal of Energy*, 3(6):323–328, 1979. doi:10.2514/3.62441.

[27] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar. A comprehensive review on wind turbine power curve modeling techniques. *Renewable and Sustainable Energy Reviews*, 30:452–460, 2014. doi:10.1016/j.rser.2013.10.030.

[28] J. Meyers and C. Meneveau. Optimal turbine spacing in fully developed wind farm boundary layers. *Wind Energy*, 15(2):305–317, 2011. doi:10.1002/we.469.

[29] 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. doi:10.1016/0167-6105(94)90080-9.

[30] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978. doi:10.1007/bf01588971.

[31] Netherlands Enterprise Energy. Project and Site Description HKZ III & IV. Oct 2018. Retrieved from https://offshorewind.rvo.nl/generalzh.

[32] U. A. Ozturk and B. A. Norman. Heuristic methods for wind energy conversion system positioning. *Electric Power Systems Research*, 70(3):179–185, 2004. doi:10.1016/j.epsr.2003.12.006.

[33] P. Y. Papalambros and D. J. Wilde. *Principles of optimal design: modeling and computation.* Cambridge University Press, second edition, 2000.

[34] B. Pérez, R. Mínguez, and R. Guanche. Offshore wind farm layout optimization using mathematical programming techniques. *Renewable Energy*, 53:389–399, 2013. doi:10.1016/j.renene.2012.12.007.

[35] E. Quaeghebeur. Robust Wind Farm Layout Optimization Using Pseudo-Gradients. Abstract from The 11th International Symposium on Imprecise Probability. Ghent, Belgium, 2019. Retrieved from http://resolver.tudelft.nl/uuid:17472c3e-53aa-4f43-a684-4959ca474471.

[36] R. A. Rivas, J. Clausen, K. S. Hansen, and L. E. Jensen. Solving the turbine positioning problem for large offshore wind farms by simulated annealing. *Wind Engineering*, 33(3):287–297, 2009. doi:10.1260/0309-524X.33.3.287.

[37] B. Saavedra-Moreno, S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto, and A. Portilla-Figueras. Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms. *Renewable Energy*, 36(11):2838–2844, 2011. doi:10.1016/j.renene.2011.04.018.

[38] Z. Shao, Y. Wu, L. Li, S. Han, and Y. Liu. Multiple Wind Turbine Wakes Modeling Considering the Faster Wake Recovery in Overlapped Wakes. *Energies*, 12(4): Article ID 680, 2019. doi:10.3390/en12040680.

[39] M. X. Song, K. Chen, Z. Y. He, and X. Zhang. Bionic optimization for micro-siting of wind farm on complex terrain. *Renewable Energy*, 50:551–557, 2013. doi:10.1016/j.renene.2012.07.021.

[40] C. Wan, J. Wang, Geng Yang, and X. Zhang. Optimal micro-siting of wind farms by particle swarm optimization. In *Advances in Swarm Intelligence*, pages 198–205, 2010. doi:10.1007/978-3-642-13495-1_25.

[41] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE Trasactions on Evolutionary Computation*, 1(1):67–82, Apr 1997. doi:10.1109/4235.585893.

[42] X. S. Yang. Metaheuristic Optimization. *Scholarpedia*, 6(8):11472, 2011. doi:10.4249/scholarpedia.11472.