

Carl von Ossietzky Universität Oldenburg

Fraunhofer Institute for Wind Energy Systems IWES

Bachelor Thesis

Wind Farm Optimization: Multi-Objective Problems of Induction Control and Wake Steering Control

Presented by: Alberto Jimenez Haro
(alberto.jimenez.haro@iws.fraunhofer.de)

First examiner: Prof. Dr. Martin Kühn
(martin.kuehn@uni-oldenburg.de)

Second examiner: Dr. Jonas Schmidt
(jonas.schmidt@iws.fraunhofer.de)

Place and Date: Oldenburg, October 12, 2021

Document history

Version	Date	Prepared	Checked	Approved
01	October 12, 2021	Alberto Jimenez Haro	Dr. Jonas Schmidt	Prof. Dr. Martin Kühn

BACHELOR THESIS

Wind Farm Optimization: Multi-Objective Problems of Induction Control and Wake Steering Control



Bachelor Thesis

Wind Farm Optimization: Multi-Objective Problems of Induction Control and Wake Steering Control

Fraunhofer Institute: Fraunhofer Institute for Wind Energy Systems
Küpkersweg 70, 26129 Oldenburg

Academic institution: Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstraße 114-118, 26129 Oldenburg

Candidate: Alberto Jimenez Haro

Submitted on: 12.10.2021

Abstract

The wake effects that occurs as a consequence of interactions between turbines in a wind farm is a key factor affecting the lifetime and power production of turbines. Strategies for mitigating this wake effects on turbines are called Active Wake Control strategies and have become very popular in the last years. This paper aims to implement and evaluate multi-objective wind farm optimization problems with the objectives of load minimization and yield maximization, and analyze the Pareto optimal solutions. The optimization problems are solved by using induction control and wake steering control as decision variables together with simulation provided by the software MoWiT.

For this purpose a multi-objective wind farm optimizer will be developed from its start, by using the python libraries of Flappy for wind farm and wake modelling calculations, and the optimization library of PYMOO, where the multi-objective optimization algorithm NSGA-II will be used for the calculations of the problem. The results finally indicate that the multi-objective optimizer does calculate problems of AWC, converging to the optimal solutions and diverging towards these solutions in a satisfactory manner. The comparison results show multiple Pareto fronts for both AWC types, which seem to produce solutions which almost always dominate the solution for the base case, i.e., without using AWC strategies, concluding that both methods have positive potential towards increasing the power production and reducing the loads on the turbines of wind farms.

Contents

Abstract	vii	
1	Introduction	16
2	Theoretical Background	17
2.1	PART I: Wind Energy Theory	17
2.1.1	Basics of Wind Energy	17
2.1.2	Wind and Wake	17
2.1.3	Wake Modelling	20
2.1.4	Turbine Loads and Damage	23
2.1.5	Active Wake Control	25
2.2	PART II: Optimization Theory	29
2.2.1	Multi-Objective Optimization and Algorithms	29
2.2.2	Evolutionary Multi-Objective Optimization (EMO)	32
2.2.3	EMO Algorithms: Genetic Algorithms (GA) and NSGA-II	34
2.2.4	Decision Making and Performance Indicators in MOO	36
3	Methods	39
3.1	Measurement Devices, Datasets and Software Tools	39
3.2	Developing the Multi-Objective Wind Farm Optimizer for Induction Control and Wake Steering Control Problems	42
3.2.1	Multi-Objective Optimizer for Induction Control Problems	43
3.2.2	Multi-Objective Optimizer for Wake Steering Control Problems	46
3.3	Testing and Evaluation of the Multi-Objective Wind Farm Optimizers	47
4	Results	48
4.1	Testing the Multi-Objective Optimizers Performance	48
4.1.1	Testing Wind Farm Flappy Calculations and Analyzing Behaviour Patterns	48
4.1.2	Testing Optimizers Convergence Towards Solutions	51
4.1.3	Testing Optimizers Divergence and Extent	53
4.2	Analyzing Pareto Fronts and its Effect on the Wind Farm	54
4.3	Comparing Multiple Pareto Front Scenario	58
5	Discussion and conclusions	62
6	References	66
Appendix A Optimization Procedure Example Case		69
Appendix B Testing Wind Farm Calculation Scenarios		72

List of Figures

2.1	a) Pressure and speed variation in an ideal model of a wind turbine [7] , and b) trajectory of an air particle passing through the rotor disc [5].	18
2.2	The control volume of the Jensen wake model. [17]	22
2.3	Schematic of the wake of a yawed turbine for Bastankhah and Porté-Agel 2016 wake model. [12]	23
2.4	Degrees of freedom of a wind turbines. [24]	25
2.5	Visualization of the wake redirection effect by yaw-based AWC [28]	28
2.6	Mapping of a decision space (a) onto an objective space (b) , where both objectives are meant to be minimized. It contains two decision variables (i.e., 2-dimensions) and two objective functions (i.e., 2-dimensions). [31]	30
2.7	a) The concept of domination illustrated in a minimization problem of two objective functions, and b) special solutions and points in MOO. [31]	32
2.8	a) Classical method to solve MOO problems by performing many independent single-objective optimizations [33], and b) EMO methods, where recombination takes place between solutions and other population members, leading to new solutions that typically expand through the objective space.	33
2.9	a) Flow chart of a typical GA [41], b) Schematic of the NSGA-II procedure [33], and c) The crowding distance calculation [33].	36
2.10	a) Reference-point based MOO used by the R-NSGA-II algorithm [42], and b) The hyper-volume indicator in the two objective case [43].	38
3.1	The working procedure of Flappy for the calculation of one chunk [45]	40
4.1	Results showing the a) decision space \mathbb{R}^n and b) objective space \mathbb{R}^M of the non-dominated solutions from fixed cases, where no optimization was performed, and the non-dominated solutions from the optimizer program developed. The optimization solutions have 200 generations with 120 population (60 offspring), using curtailment of the turbines as the decision variables, 12m/s wind speed, TI of 5% and a horizontal distance of 360m.	52
4.2	a) The hypervolume performance metric from Figure 4.1, and b) the objective space \mathbb{R}^M from the same case showing the Pareto Frontier (red) and all preceding solutions (empty blue)	53
4.3	Comparison of the Pareto front created using the MOO program (green points) with the single-objective optimizers for power maximization and loads minimization (yellow diamonds), in 12 turbine wind farm using yaw misalignment of the turbines as the decision variables, 12m/s wind speed, TI of 5% and a horizontal distance of 600m for all cases.	54
4.4	Comparison of the Pareto Frontiers created ,using the MOO program, when induction control methods are being used (black points) or when wake steering control methods are used (red points) in a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.	55
4.5	Figure showing individual wind farm solutions from the Pareto front solutions of induction control in Figure 4.4 for the cases of a) the trade-off solution offering maximum power production and its wake (d), c) the trade-off solution offering minimum DEL and its wake (e), and b) a case in between offering a balance between both objectives. For a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.	56

4.6	Figure showing individual wind farm solutions from the Pareto front solutions of wake steering control in Figure 4.4 for the cases of a) the trade-off solution offering maximum power production and its wake (d), c) the trade-off solution offering minimum DEL and its wake (e), and b) a case in between offering a balance between both objectives. For a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.	57
4.7	Figure showing solutions from a Pareto Front that offer maximum power production (top-left) using the same properties from Figure 4.4 for the wind farm but changing only the initial wind speed to a) 9m/s and b) 15m/s	58
4.8	Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 9m/s. Solutions where no AWC method was applied are also shown (diamond).	59
4.9	Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 12m/s. Solutions where no AWC method was applied are also shown (diamond).	60
4.10	Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 15m/s. Solutions where no AWC method was applied are also shown (diamond).	61
B1	Test showing wind farm calculations for multiple case scenarios using as independent variables wind speed and TI, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other.	72
B2	Test showing wind farm calculations for multiple case scenarios using as independent variables the yaw misalignment of the first and second turbine at the leftmost, and showing the resulting a) wind speed and b) TI for a 3 turbine case at a 600m distance from each other, initial 12m/s wind speed and an initial TI of 5%.	72
B3	Test showing wind farm calculations for multiple case scenarios using as independent variables (Top) curtailment of leftmost turbine and wind speed, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other and a TI of 5%. And using as independent variables (Bottom) curtailment of leftmost turbine and TI, and showing the resulting c) power and d) Flapwise DEL with same conditions and wind speed of 12m/s.	73
B4	Test showing wind farm calculations for multiple case scenarios using as independent variables (Top) yaw misalignment of leftmost turbine and wind speed, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other and a TI of 5%. And using as independent variables (Bottom) yaw misalignment of leftmost turbine and TI, and showing the resulting c) power and d) Flapwise DEL with same conditions and wind speed of 12m/s.	74
B5	Test showing wind farm calculations for multiple case scenarios using as independent variables the curtailment of the first and second turbine at the leftmost, and showing the resulting a) power and b) Flapwise DEL for a 3 turbine case at a 600m distance from each other, 12m/s wind speed and TI of 5%	75

- B6 Test showing wind farm calculations for multiple case scenarios using as independent variables the yaw misalignment of the first and second turbine at the leftmost, and showing the resulting a) power and b) Flapwise DEL for a 3 turbine case at a 600m distance from each other, 12m/s wind speed and TI of 5% 75

List of Tables

3.1	Wind farm characteristics for calculations	43
3.2	Fixed parameters employed for wake models calculations.	44
4.1	Test result correlations from wind farm calculations of 1287 scenarios using testing script from Section 3.3 with two turbines. It shows the resulting increase (\uparrow) or decrease (\downarrow) of Power, C_t and Flapwise DEL when the testing cases are increased in wind speed, TI and distance between turbines.	48
4.2	Test correlations from wind farm calculations of more than 5000 scenarios in wind farm cases with two or three turbines in a row. It shows the resulting increase (\uparrow) or decrease (\downarrow) of Power, C_t and Flapwise DEL when the testing cases are increasingly curtailed or yawed.	49

Symbols and Abbreviations

Symbols

P Power output from a wind turbine sort.

A Area of the intercepted airstream by the rotor.

a Axial Induction Factor.

C_p Power Coefficient.

C_t Thrust Coefficient.

DEL^{life} Damage Equivalent Loads over the design lifetime.

D^{life} Damage over the design lifetime.

d_r Rotor Diameter.

d_w Diameter of the wake behind rotor.

PF^* Pareto-Optimal Front or Pareto Frontier.

P^* Pareto-Optimal Set.

\mathbf{x}^* Pareto-Optimal Solution.

ρ Density of the air.

\mathbb{R}^M Objective Space containing M objectives (dimensions).

\mathbb{R}^n Decision Space containing n decision variables (dimensions).

σ Standard Deviation of Wind Speed variations.

T Thrust on the turbine blades.

TI Turbulence Intensity.

u_w Wind Speed at Downstream.

\bar{u} Mean Wind Speed.

u_r Wind Speed at Rotor.

u_∞ Wind Speed at Upstream.

\mathbf{W} Worst Objective Vector.

α Wake Decay Constant.

γ Yaw Angle.

\mathbf{z}^0 Ideal Objective Vector.

\mathbf{z}^{nad} Nadir Objective Vector.

\mathbf{z}^u Utopian Objective Vector.

Acronyms

AWC Active Wake Control.

DEL Damage Equivalent Loads.

DM Decision Maker.

EA Evolutionary Algorithm.

EMO Evolutionary Multi-Objective Optimization.

EO Evolutionary Optimization.

Flappy Farm Layout Program in Python.

Flapwise DEL Damage Equivalent Loads for Flapwise Blade Root Bending Moment M_y .

GA Genetic Algorithm.

MOO Multi-objective optimization.

NSGA-II Non-dominated Sorting Genetic Algorithm.

PYGMO Parallel Global Multiobjective Optimizer.

PYMOO Python Multi-Objective Optimization.

SOO Single-objective optimization.

1 Introduction

Wind energy is one most popular sources of renewable energy in the current era, especially in the North of Europe due to its strong winds [1]. This industry which harnesses the power from the wind is growing at a fast pace, where not only the number of wind parks is increasing, but also the number of turbines inside these wind parks and the turbines themselves. For this reason there is an increased demand for methods to correctly measure and calculate the power production of these wind farms and their lifespan [2].

One of the big challenges of wind farms is the wake effects, this is, the interaction that occurs between turbines where an upstream turbine casts a wind shade on the turbine it has behind. This factor is vital when designing a wind farm and during their functioning time also, as it affects the lifespan of the turbines affected and the power output obtained. This issue has been disregarded for long time and it was not until recently that methods for mitigating this wake appeared [3].

The strategies for mitigating the effect that the wake has on the loads and power production of turbines are called Active Wake Control (AWC) Strategies. These are very popular at the moment for the wind industry and researchers [4]. They are comprised of two different strategies being induction control (pitch-based) and wake steering control (yaw-based) AWC, and have been found to increase the power production of a wind farm and reduce its loads, to also lower costs of operations and maintenance, and to minimize capital expenses by placing the turbines closer together.

In order to find the best AWC strategies, one must make use of optimization processes and algorithms that will search the optimum strategies for obtaining power production or load reduction. But ideally, the best interest would be to simultaneously increase the power and reduce the loads, or achieve a trade-off between both that would be the most satisfactory. This can be achieved using multi-objective optimization processes, with objectives of power increase and load reduction.

For this purposes, the object of this paper consists in the implementation and evaluation of multi-objective wind farm optimization problems with the objectives of load minimization and yield maximization, and the following analysis of their Pareto front of Pareto optimal solutions. The problems will be solved by using induction control and wake steering control methods, meaning that as decision variables of the optimizer the curtailment of the turbines or the yaw misalignment of these will be utilized for calculating and improving the objectives. The simulation for the calculations are provided by the software MoWiT, a simulated generic direct drive IWT 7.5 MW turbine. In other words, the aim consists in developing a optimizer script which consists of a fully functional piece of code which is integrated into a wake modelling software called Flappy, that when executed performs multi-objective optimization calculations using the techniques of AWC.

Firstly, the background theory necessary for understanding concepts in wind energy and optimization will be laid in Chapter 2. In Chapter 3 the multi-objective wind farm optimizer will be developed from its beginnings, by using the python libraries of Flappy for wind farm and wake modelling calculations, and the optimization library of PYMOO, where the optimization algorithm 'Non-dominated Sorting Genetic Algorithm' (NSGA-II) will be used for multi-objective calculations of the problem. In Chapter 4 the optimizer will be tested in order to prove its correct performance and wind farm calculations, and finally some test cases will be executed and the results shown and later compared and discussed in Chapter 5.

2 Theoretical Background

2.1 PART I: Wind Energy Theory

2.1.1 Basics of Wind Energy

Just about one or two percent of the energy arriving from the sun is transformed into wind energy. This energy from the wind is extracted by a wind turbine using the force from the rotor blades. By getting rid of an amount of its kinetic energy the wind slows down, but this only happens to the portion of air which passes through the rotor disc. It's the difference in atmospheric pressure across both sides of the blade that creates the lift and the drag, which makes the rotor to spin around which is connected to a generator that creates electricity [5]. Turbine models are typically identified by its dimensions, its manufacturer and its power rating [3].

The conditions at a specific site including the wind's variability and speed average, and extreme events are key elements on the performance. Let's begin by considering how the amount of power that impinges on a turbine rotor placed in a steady airstream. The power output from a turbine is given by the expression:

$$P = \frac{1}{2} C_p \rho A u_\infty^3 \quad (2.1)$$

where C_p is the power coefficient, ρ is the density of the air, A is the area of the intercepted airstream (equal to the area 'swept' by the rotor), and U_0 is the speed of the wind. [6]

As we see in Equation (2.1) the energy available in the wind changes with the cube of the wind speed. Also important from this equation is the rotor diameter, that identifies the area spanned by the turbine blades. In general, the bigger the rotor, the higher the rated power of a turbine [5].

2.1.2 Wind and Wake

The most noticeable characteristic of the wind is its variability, both temporally and geographically [5]. The wind speed is always fluctuating, and so it is the energy content of the wind.

Fluctuations in wind speed on a relatively fast timescale (about less than about 10 minutes) are referred to as turbulence. Turbulence is mostly originated by friction with the earth's surface, which can be thought of as reaching as far as disturbances in the flow caused by topographical features such as hills and mountains. And also by thermal effects which can cause masses of air to move vertically as a result of changes in temperature and therefore in the air density. The process can be described as 'chaotic' in that the smallest differences in boundary conditions or initial conditions can result in large differences in the predictions after a relatively short time.

For this reason, it is usually more useful to develop descriptions of turbulence in terms of its statistical properties. The turbulence intensity is measured as the overall level of turbulence, and is defined as

$$TI = \frac{\sigma}{\bar{u}} \quad (2.2)$$

where σ is the standard deviation of wind speed variations about the mean wind speed \bar{U} , usually defined over ten minutes or an hour. Turbulent wind speed variations can be considered to be

roughly Gaussian. This parameter will be crucial in the software calculations, especially for the wake steering control calculations.

As we saw in the section above wind turbines employ a rotor with blades rotating at an angular velocity Ω about an axis normal to the rotor plane and parallel to the wind direction.

When a flow of air advances towards a wind turbine, it begins to slow down and the pressure increases. Then, when it crosses the turbine rotor, there is a sudden pressure drop as shown in Figure 2.1a. The blades sweep out a disc and develop a pressure difference across the disc. This pressure difference is accountable for the axial momentum's loss in the wake, creating a loss of energy that is gathered by an electrical generator which is attached to the rotor shaft. A thrust force and a torque in the direction of rotation is experienced by the rotor, that will oppose the torque that the generator applies.

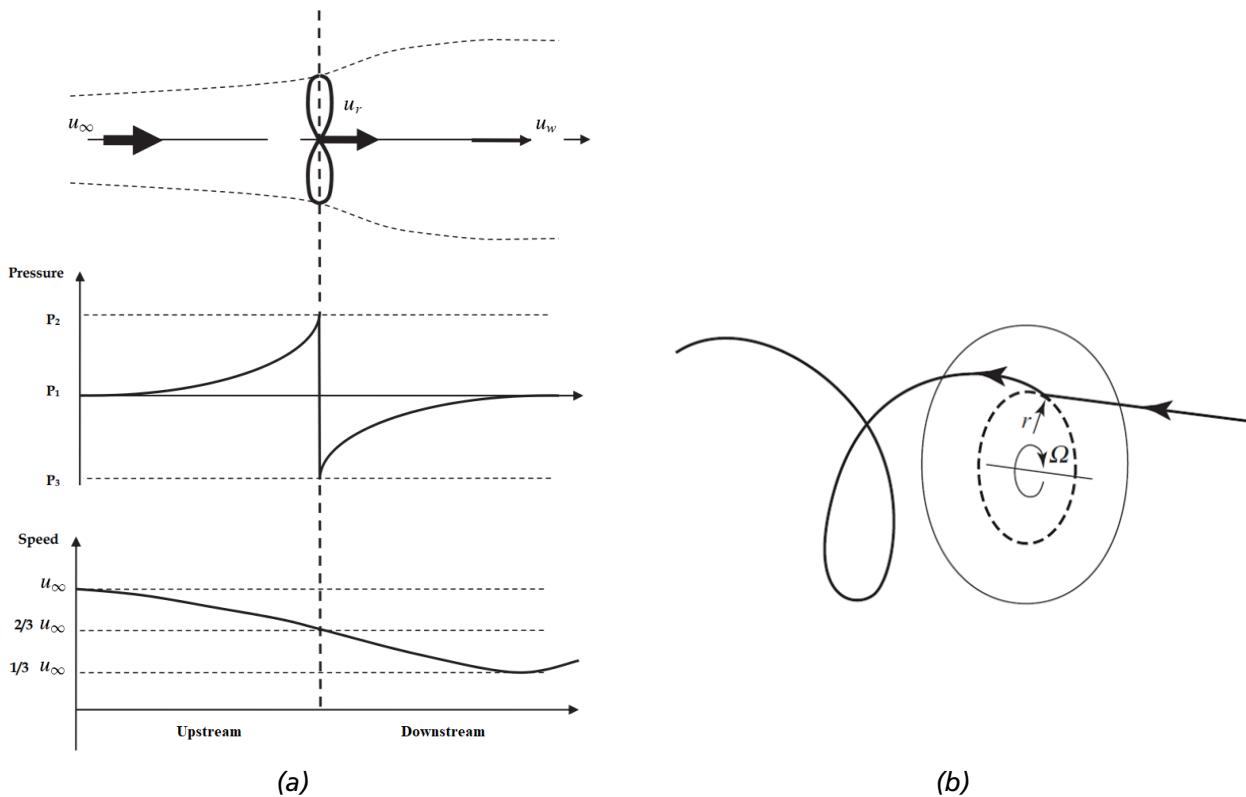


Figure 2.1 a) Pressure and speed variation in an ideal model of a wind turbine [7], and b) trajectory of an air particle passing through the rotor disc [5].

Exerting this torque on the rotor disc by means of the air passing through it requires an equal and opposite torque to be applied into the air. The consequence of this reactive torque is to cause the air to rotate in an opposite direction to that of the rotor. Then the air picks up angular momentum and so in the wake of the rotor disc the particles of air have a velocity component which is in a direction tangential to the rotation and moreover an axial component as seen Figure 2.1b.

Obtaining this tangential portion of velocity by the air translates into an increase in its kinetic energy. This is compensated for by a decrease in the static pressure of the air in the wake. As seen in Figure 2.1b the flow entering the actuator disc has no rotational motion at all while the exiting flow of the disc does have it, and that rotation remains constant as the fluid progresses along the wake. This transfer of rotational motion to the air takes place entirely across the thickness of the disc [5].

So as observed above, the wind leaving a turbine must have a lower energy content than the wind arriving in front of the turbine. This wake effect behind the turbine that was mentioned before could be described as a long trail of wind which is quite turbulent and slowed down containing tangential components to its initial direction, as compared to the wind that arrives in front of the turbine [1].

A different turbine that operates in this wake, or operating deep inside of a wind farm where the effects of various wakes may be felt simultaneously, will produce less energy and suffer a more considerable structural loading than that of a wind turbine that operates in the free stream.

The wake right behind a wind turbine can be viewed as a region of reduced wind speed which is larger in diameter than that of the wind turbine itself. The velocity reduction is related to the thrust coefficient C_t of the turbine [5].

The axial force that is exerted by the wind on the rotor of a wind turbine, and therefore also the force applied by the turbine on the wind is the thrust force. It has a relation to all the losses of kinetic energy in the wind flow. The thrust coefficient C_t hence determines the momentum that is extracted from the wind flow [2].

$$C_t = \frac{T}{\frac{1}{2} \rho A u_\infty^2} = \frac{\rho A u_r (u_\infty - u_w)}{\frac{1}{2} \rho A u_\infty^2} = 2 \frac{u_r}{u_\infty} \left(1 - \frac{u_w}{u_\infty} \right) \quad (2.3)$$

Where T stands for the Thrust force on the turbine blades, u_∞ for the inflow upstream wind speed, u_w is the wake downstream wind speed and u_r the wind speed at the rotor.

This thrust force on the turbine rotor normally grows with increasing solidity (total blade area divided by swept area) and is directly applied to the tower where the rotor is supported, significantly influencing the structural design of the tower. The concept in equations 2.1 and 2.3 are therefore quite important as the performance characteristics of a wind turbine are given by the power and thrust coefficients.

Behind the turbine, while the diminished wind speed region convects downstream, the wind speed gradient located between the free flow outside the wake and the wake itself a results in added shear-generated turbulence assisting to the transfer of momentum towards the wake from the neighbouring wind flow. Hence, the surrounding flow and the wake itself begin to mix, and this mixing region extends inwards to the core of the wake and outwards to enlarge the width of the wake. In this fashion, there is an erosion in the velocity deficit in the wake and this one becomes wider but shallower until the wind flow has completely recovered in the far downstream. The pace at which this takes place is dependent on the amount of ambient turbulence.

Additionally to this turbulence which is generated in the shear, the turbine itself also creates additional turbulence directly, being as a result of the general disturbance to the flow caused by the blades, nacelle and tower, along with the tip vortices that are shed by the blades. This element of the turbulence is of somewhat high frequency and decays in a relatively quick manner [5].

A wake region is usually divided into a far and a near wake. The region downstream of the rotor is the near wake and it expands for two to five rotor diameters. It is a region that is dominated by the turbulence made by the wind turbine itself. The far wake region begins at around five diameters after the rotor, where the wake flowing is entirely developed. The wind velocity then begins to recover and the wind flow decays to its free stream state. The ambient turbulence and the topographic effects become then dominant over the turbulence produced by the rotor [8].

When the turbulence generated by the shear has reached the core of the wake and begins to erode the centreline velocity deficit afar the near wake region, the variation in the mean velocity of the wake can be described by an axisymmetric profile with has a Gaussian cross-section. Developing the wake profile downstream of this spot can be fairly well predicted.

Inside a wind farm, the reduction in wind speed and the increase in turbulence intensity are the result of the superposition of wakes from many upwind turbines. This will be a huge factor in the optimization cases that will be further presented, for both Power and Damage Equivalent Loads (DEL) objectives, as the optimization process itself will strongly depend on how the superposition of wakes increases/decreases wind speed and turbulence intensity [5].

2.1.3 Wake Modelling

It's been shown above that the wake has a powerful impact on wind farms as it has a strong influence in the power output and the amount of turbulence that determines the lifetime of a turbine, making wake modelling a topic of vital significance to the industry of wind energy and especially to the subject of wind farm optimization.

Wake modelling has been a hot research topic in the last years, and there is basically two classical perspectives that can be identified for this problem. A generic approach to this problem consisted in assuming that the wind turbines behaved as distributed roughness components. These models usually employ a wind profile that is logarithmic and that is altered by an increase in the roughness because of the presence of the wind turbine itself.

The conventional approach to wake modelling nonetheless, has its basis on a single wake description, followed by calculating its interaction with the adjacent ones. These model types are referred to as individual models. Individual wake models can be separated into two different categories: the computational models and the analytical models.

Computational models are computationally expensive take up a lot of time, since they make the very little simplifications of the Navier-Stokes equations to completely identify the wake and turbulence of the turbine. These type of models calculate the flow quantities at every single point of the flow field.

On the other hand, the analytical approach, consists of wake models build on semi-empirical functions and in simplified Navier-Stokes equations. These models use analytical expressions for the calculations of the wind speed deficits after calculating the wind fields. They can be really effective in modelling the velocity deficit and the expansion of the wake, and they are also commonly preferred owing to its rapid resolution and computational efficiency. Withal, a turbulence model has to be usually combined with analytical models as the change in ambient turbulence is not considered [8].

The drawback of these analytical wake models is their dependence on coefficients that must be empirically determined for every case, making them not commonly applicable to all wind farm sites (e.g. the wake expansion coefficient of the Jensen model). Moreover, analytical models suffer additionally from inherent simplifications when calculating for example the turbulence characteristics of the wake and wake-to-wake interaction [9].

Examples of the Jensen model [10] and two other analytical wake gaussian models from Bastankhah and Porté-Agel [11] [12] will be presented. These will be used in the final software calculations or while testing wake models.

The Jensen Wake Model

The Jensen wake model is a popular engineering wake model that quantifies the reduction of the wind speed downstream of a turbine. It was formulated in 80s when few wake models were available and it is a very simplistic, fast and easy to implement model. It is also the base of the Park model [13] that was created for wind farm calculations for the Wind Atlas Analysis and Application Program (WAsP) [14], which is used broadly for estimating wind resources.

It is not perceived as really accurate at predictions of wake losses under determined atmospheric inflow conditions which is due to the simplicity of its physical considerations. It is nonetheless reasonably accurate at predicting annual energy production wake losses [15].

Using the control volume presented in Figure 2.2, where $d = d_r$ is the rotor diameter, and assuming a top-hat inflow profile the mass balance between the rotor plane and the downstream flow yields,

$$\left(\frac{d_r}{2}\right)^2 u_r + \left[\left(\frac{d_w}{2}\right)^2 - \left(\frac{d_r}{2}\right)^2\right] u_\infty = \left(\frac{d_w}{2}\right)^2 u_w \quad (2.4)$$

Also, the wake is presumed to have a linear expansion as a function of the downstream distance x at a rate α which is the wake decay constant that describes the expansion rate of the wake. Which means that $d_w = d_r + 2\alpha x$ and $\frac{u_r}{u_\infty} = 1 - 2a$ using the axial induction factor, the fractional decrease in wind speed, $a = \frac{u_\infty - u_r}{u_\infty}$. Putting them into Equation (2.4), the normalized velocity can be found as

$$\frac{u_w}{u_\infty} = 1 - \frac{2a}{\left(1 + \frac{2\alpha x}{d_r}\right)^2} \quad (2.5)$$

Supposing that the axial symmetric flow is ideal with no rotation, no turbulence and a conic shape for the wake profile, the axial induction factor can be noted as

$$a = \frac{1 - \sqrt{1 - C_t}}{2} \quad (2.6)$$

The wake decay coefficient α , which is the main factor in Jensen model since it defines the expansion rate of the wake, is associated to the ambient turbulence intensity TI . Both wake decay coefficient and turbulence are increased in the wake [16].

Gaussian wake models

As opposed to the Jensen model which assumes a uniform velocity profile inside of the wake, the Gaussian wake model evaluates the deficit in velocity by using an exponential function. Two Gaussian wind deficit wake models will be presented, both from Bastankhah and Porté-Agel in 2014 [11] and 2016 [12].

The Gaussian shape of the wind deficit using mass and momentum conservation looks like the following,

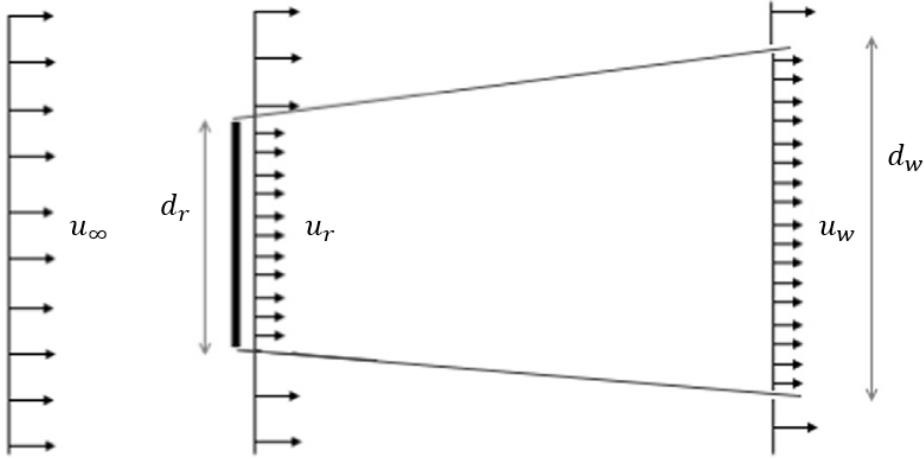


Figure 2.2 The control volume of the Jensen wake model. [17]

$$\frac{u_{\infty} - u_w}{u_{\infty}} = C(x) e^{-\frac{r_w^2}{2\sigma^2(x)}} \quad (2.7)$$

where r_w is the radial distance from the center of the wake and σ is the standard deviation of the Gaussian-like velocity deficit profiles at each x .

For the 2014 model, the wake decay component $C(x)$ and the Gauss profile expressed in the exponent appear as,

$$\frac{u_{\infty} - u_w}{u_{\infty}} = \left(1 - \sqrt{1 - \frac{C_t}{8 \left(\frac{k^* x}{d_r} + 0.2 \sqrt{\beta} \right)^2}} \right) \exp \left(-\frac{1}{2 \left(\frac{k^* x}{d_r} + 0.2 \sqrt{\beta} \right)^2} \left\{ \left(\frac{z - z_h}{d_r} \right)^2 + \left(\frac{y}{d_r} \right)^2 \right\} \right) \quad (2.8)$$

where β is a function of C_t , k^* represents the wake growth rate, y and z are spanwise and vertical coordinates, respectively, and z_h is the hub height.

The Jensen model together with other Top-hat model commonly underestimate the velocity deficit that occurs at the wake center and overestimate it nearer to the edges of the wake. This 2014 wake model is consistent and adequately accurate when estimating power whereas top-hat models like Jensen are less accurate and negatively sensitive to wind turbines relative position with respect to the wind direction [11].

The method of yaw angle control can be used to reduce power losses from interaction with the wake in wind farms, done by deflecting the wakes away from other downwind turbines. Although power production can be reduced by yawing a wind turbine, there is still the chance that yawing increases the total power produced by the whole wind farm. For this reason, the 2016 Gaussian wake model was proposed by Bastankhah and Porté-Agel. Compared with their 2014 model, this new one can predict the wake deflection and the far-wake velocity distribution for yawed turbines.

In Figure 2.3 an idealized schematic of the wake of a yawed turbine is showed. As the wake moves downstream, the potential core decreases until it finally ends at $x = x_0$. Prior to this moment

however, the centre portion of the potential core is not affected by the ambient flow. Consequently the velocity magnitude and the flow angle situated in the wake centre stays the same across the potential core. Once the potential core is terminated, the wake centre begins its recovery, and the wake deflection angle drops because of the interaction with the ambient flow. The wake deflection angle and velocity in the potential core are indicated by u_0 and θ_{c0} respectively in Figure 2.3.

The Gaussian shape of the wind deficit for the 2016 model which include the prediction of far-wake velocity for a yawed turbine is the following,

$$\frac{\Delta \bar{u}}{\bar{u}_\infty} = \left(1 - \sqrt{1 - \frac{C_t \cos \gamma}{8 \left(\frac{\sigma_y \sigma_z}{d_r^2} \right)}} \right) e^{-0.5 \left(\frac{y-\delta}{\sigma_y} \right)^2} e^{-0.5 \left(\frac{z-z_h}{\sigma_z} \right)^2} \quad (2.9)$$

with the incoming velocity $\frac{\Delta \bar{u}}{\bar{u}_\infty}$, the introduction of a yawed angle γ , the wake-centre deflection at each downwind location δ and different wake widths in the y and z directions denoted by σ_y and σ_z .

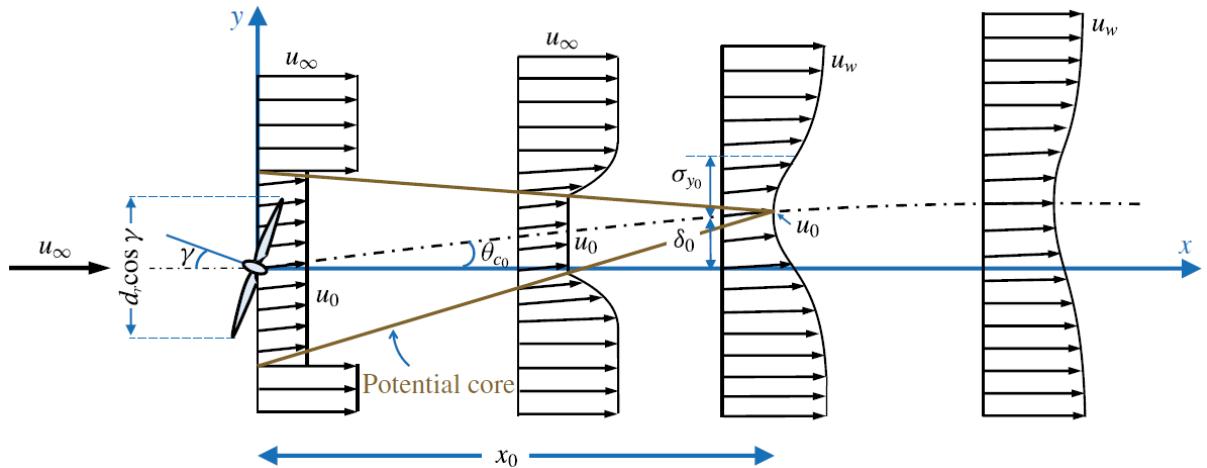


Figure 2.3 Schematic of the wake of a yawed turbine for Bastankhah and Porté-Agel 2016 wake model. [12]

As the 2016 model was explicitly developed to represent the wake deflection by intentional yawing, this wake model will therefore be used for the wake steering control calculations.

2.1.4 Turbine Loads and Damage

The stresses and tensions on a wind turbine affect the strength of and duration of these. A wind turbine should be in a fit and undamaged condition in order to be able to withstand the loads it endures. The origin point for the loads are commonly categorized in Gravitational, Aerodynamic, Inertia loads (including centrifugal and gyroscopic effects) and Operational loads (e.g. yawing, blade pitch control, braking).

Depending on its impact, one can find steady loads mainly due to centrifugal forces and gravity, and also as a result of gusts what is called as transient loads associated with braking [18]. Alternating winds which result in alternating forces, makes the components in a turbine to be under repeated bending forces that can lead to breaks. These are the cyclic load which lead to fatigue damage. It

is therefore vital in the design of wind turbines and farms to try and predict in advance how these loads and vibrations will affect the turbines strength and for how long [1], especially the fatigue damage as a result of alternating loads over the design life of the turbine.

The wake effects from the turbines not only have an effect in the power production, but also it affects the fatigue loads. These wakes although they decrease the wind speeds, they increase the TI in the air for the next turbines, which consequently increases the loads. Induction control and wake steering methods can influence the loads as they affect the wake, and will be later explained in more detail in Section 2.1.5.

Following Miner's Rule [19], the standard for fatigue analysis of turbines [20], the fatigue damage can be accumulated linearly and reduced into individual hysteresis cycles. For the calculation of the lifetime damage of a turbine, one must get collect smaller time-series data which encompass shorter periods than an entire lifetime, and later extrapolate them over the entire design lifetime [21]. These total amount of damage is given by the equation,

$$D^{life} = \sum_j \sum_i \frac{n_{ij}}{N_{ij}} \quad (2.10)$$

where,

$$N_{ij} = \left(\frac{L^{ult}}{\frac{1}{2}L_{ij}^{RF}} \right)^m \quad (2.11)$$

here i indicates the cycle, j indicates the time-series, n_{ij} is the cycle count, N_{ij} the number of cycles to failure, L_{ij}^{RF} represents the oscillation range of a load cycle, L^{ult} the ultimate load which would lead to a damage of $D^{life} = 1$ if it occurred once and m the Wöhler coefficient which is specific to the component.

The design parameter L^{ult} is the ultimate load which is unknown in most cases. This load value can be found by pre-calculated Damage Equivalent Loads (DEL). This is a fatigue load with constant amplitude occurring at a set frequency and producing the a damage that is equivalent to the variable spectrum loads. The DEL of a turbine can be determined by smaller sets of time-series data from much shorter periods in time than those of the design lifetime. It is resolved by using the lifetime count extrapolation factor f_j^{life} , so by adapting Equation (2.11) to include DEL, these can be obtained as,

$$N_{ij} = \left(\frac{L^{ult}}{\frac{1}{2}DEL^{life}} \right)^m$$

$$DEL^{life} = \left(\frac{\sum_j \sum_i f_j^{life} n_{ij} (L_{ij}^{RF})^m}{N_{eq}} \right)^{\frac{1}{m}} \quad (2.12)$$

Where N_{eq} is the equivalent number of load cycles (typically 10^7 cycles [22]).

When talking about turbine loads and DELs, one must first take into account all the different components and directions in which this fatigue damage can occur. In a wind turbine, especially the exterior components like the , tower, hub and each blade component, suffer each loads. For each

component there is a different load applied on them as a force F in each of the three dimensions (F_x, F_y, F_z), a moment M in each dimension (M_x, M_y, M_z) along sometimes with a resulting torsion T especially in the blades.

Although all loads are important for consideration, there are some that are the most crucial to consider to ensure the correct lifetime of the wind turbine. Considering the blades, the main form of vibration these are in-plane (mainly the edgewise blade root bending moment related with the moment M_x) and out-of-plane (mainly the flapwise blade root bending moment related with the moment M_y) and can be seen in Figure 2.4 [23].

The edgewise moment has barely no damping, while the flapwise moment is aerodynamically damped by nature. For this reason the flapwise modes will add to the fatigue damage more considerably and will be taken into account for the results of the optimizers calculations. One can also consider the tower loads as important, as these may combine with the motion of the blades to create unwanted vibrations [24][25].

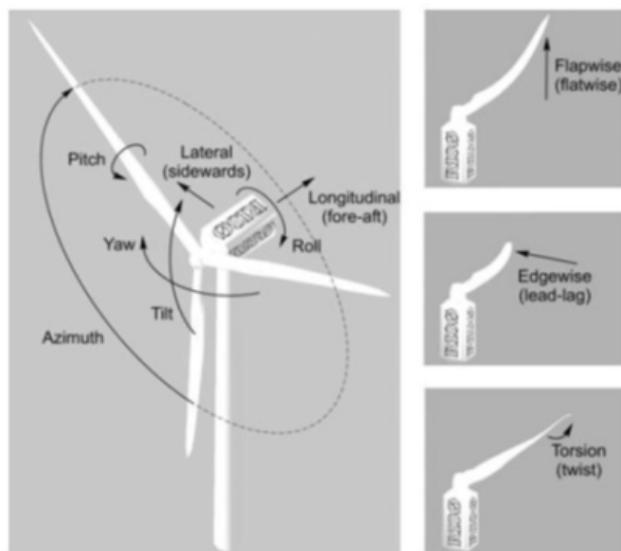


Figure 2.4 Degrees of freedom of a wind turbines. [24]

2.1.5 Active Wake Control

Nowadays wind turbines are being densely packed in wind farms and they have an effect on each other's power production and loading damage through their wakes, producing less power and experiencing increased fatigue loading on account of the reduced wind velocity and increased turbulence.

Nonetheless, it is the common practice to permit each turbine to maximize its individual power capture, and in doing so disregarding its effect on the other turbines. This is not an optimal strategy with respect to the power produced by the entire wind farm.

Different techniques have been developed towards maximizing an entire wind farm power production, while simultaneously attempting the reduction of the fatigue loading on the turbines. These techniques for mitigation of wake effects are called Active Wake Control (AWC) [4].

The concept of AWC offers the following user benefits:

- To maximize the power production of a wind farm.
- Reducing the costs of operations and maintenance (O&M) by minimizing the fatigue loads.
- It allows to decrease capital expenses by placing the turbines closer together (therefore reducing electrical infrastructure costs) when it is applied at the design phase of the wind farm.

The concept of AWC can be applied at below rated wind speeds, and it is comprised of two strategies being pitch-based and yaw-based AWC, which are both patented by ECN [26][27], and they can either be applied individually or in combination.

Pitch-based AWC is able to increase the power production while simultaneously reducing the loads. In practice, it is much less beneficial regarding the power production compared with yaw-based AWC. On the flip side, when yaw-based AWC is not properly designed, it can lead to an increase of the turbine loads. It is hence vital to develop efficient optimization algorithms for the design of yaw-based AWC settings targeted at power production maximization while keeping at the same time the loads inside their design limits.

Several simulation studies have concluded that the optimal settings for AWC have a weak dependence on the wind speeds but a strong one on the wind directions [4]. Additionally, a few studies indicate that a lifetime power increase of up to 1-1.5% is possible with AWC [28].

Induction Control

Induction controls goal consists in the reduction of the wake deficit downstream by limiting the axial induction factor of turbines upstream, known as axial induction control (also known as pitch-based AWC). This can be accomplished by operating the wind turbines at the windward side using an increased blade pitch angle. It is also possible in practice to achieve an increased pitch angle indirectly by reducing the power produced by the wind turbine, i.e., derating.

Small changes in pitch setting angle can have a huge effect on the power output when changing the blade pitch angle. By using positive pitch angle settings it is possible to increase the design pitch angle and thus decrease the angle of incidence. Contrarily, negative pitch angle settings can increase the angle of incidence and in some cases possibly cause stalling. A rotor designed to operate ideally at some given wind conditions can be suited to others by proper adjustments of rotational speed and blade pitch angle [5].

A result of induction control is that as the production of power from the derated machines is decreased, there is an increase in the wind velocity in their wakes, which allows to increase power production at downstream wind turbines and there is a chance of raising the overall farm production. The TI in the wake also decreases, prompting to lower fatigue loads downstream.

Regarding the use of induction control in wind farms, some authors like Kanev and Savenije [28] indicate that when it comes to maximizing the power yield, pitch-based AWC only needs to be applied to the first wind turbine in the row (maybe even the first two turbines). The reason that pitch-based AWC application to the second (and later) wind turbines in the row does not lead to noticeable power yield improvement they believe, is that the TI after the first turbine increases too much and therefore also the wake recovery. As a result of that, the wake effects decrease and so does the benefit of AWC. This concept will be further explained and tested in Chapter 4.

Wake Steering Control

The other class of AWC method consists in redirecting the wakes aside from downstream wind turbines, letting them increase their power production. Redirecting the wake can be attained by operating the turbines at yaw misalignment (what is called yaw-based AWC).

Because the majority of horizontal axis wind turbines use a yaw drive mechanism so that the wind turbine stay headed into the wind, the same principle can be used to yaw the turbine out of wind so to limit power output. Still, there are two factors which counter against the quick response of the system to limit power. In the first place, the huge moment of inertia from the rotor and nacelle about the yaw axis. In second place, the sinusoidal relationship that exists between the element of wind speed which is perpendicular to the rotor disc and the yaw angle. This last factor means that at initial yaw angles that are small in size, the yaw changes of small degree values only bring reductions in power of a few percentage, whereas blade pitch changes of this size can easily cut by half the power output.

In the case of active wake control using yaw misalignment, small changes like this in power reduction really could mean that the yaw angles can be modified, especially at small initial angles without having a heavy effect in the power performance, but as a consequence, the wake of the wind turbine could be directed partially outside or completley outside of the direction of the next incoming wind turbine, what in return would be beneficial for this turbine, although there is a possibility of the opposite effect happening. Also one can take into account that rapid yawing can create large gyroscopic loads.

Because yaw misalignment changes the direction of the wake, yaw-based AWC, although it increases the power output, it could also theoretically increase the loads for some of the wind directions. This means that it is key to be able to evaluate the effect that AWC has on the fatigue loads and if possible include these loads into the optimization process to guarantee they remain lower than the nominal fatigue loads.

When talking about the design of yaw-based AWC, the effects that they have on the loads of the turbines need to be clearly understood. Few conditions exist that being influenced by yaw-based AWC they affect the loads. Firstly, is the **yaw misalignment**. In first place, in regards to the loads of a yawed turbine, some studies [29][30] have concluded that when the turbulence intensity is low the blade flapwise DEL is the lowest at a yaw misalignment angle which is not zero, as a result of the effect of the wind shear. Above and below this optimal angle of yaw misalignment the blade flapwise DEL will increase. Furthermore the same results determined that depending on the yaw misalignment directionality, the blade bending DEL either decreases or increases.

An effect of yaw misalignment can also be found on the loads of the wind turbines in the wake of the ones that are yawed. The consequences of applying yaw-based AWC to a turbine derives in the diversion of its wake, which results in:

- A wind speed increase in the downstream, which increases DEL usually.
- A TI reduction in the downstream, which decreases DEL.
- And it creates half-wake operations downwind, which increases DEL. This half-wake situation can be found in Figure 2.5.

The second condition that affects the DEL is the **turbulence intensity**. The TI has a huge effect on the loads on blades and towers following studies from Kanev and Savenije [28]. A relationship

that is almost linear has been found between TI and DEL, which proves that AWC can reduce the TI on turbines in a farm.

The last condition which affects the DEL is the **wake location** with respect to rotor, and its bulge width and depth. Following same studies from Kanev and Savenije [28] the wake location affects the DEL on the blades and tower.

Asymmetric loading on the rotor by wind speed deficit in the wake can occur in partial wake states, which can heavily increase the DEL by as much as 50 to 60%.

However, the increased wake width spreads over the asymmetric loads distributed over the rotor, causing a lower DEL.

When creating the software for Multi-Objective Optimization of Induction Control and Wake Steering Control Problems, the above information and studies in this section will be crucial as some type of benchmark or reference comparison during the testing period of the software and the final results.

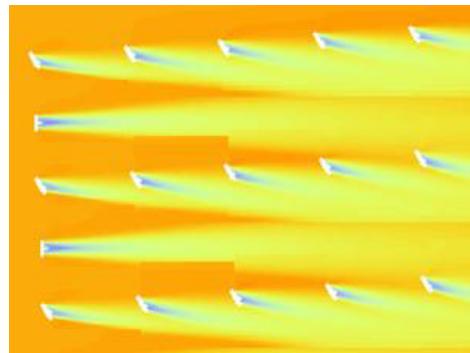


Figure 2.5 Visualization of the wake redirection effect by yaw-based AWC [28]

2.2 PART II: Optimization Theory

Improving the performance of wind turbines by using effective control strategies to reduce the power generation cost and improve profits is very demanded by the wind industry. Wind Farm Optimization is the methodology or process of making something (such as a design, system, or decision related with wind farms or wind turbines) as fully functional and effective as possible.

2.2.1 Multi-Objective Optimization and Algorithms

An optimization problem is the problem of finding the best solution from among the set of all feasible solutions. It may have only one objective function (i.e., single-objective problem), or it there may be multiple conflicting objective functions (i.e., multi-objective problem)

Multi-objective optimization (MOO), also called multi-criteria, many-objective or vector optimization, is the procedure for optimizing two or more conflicting objectives at the same time which are subject to some particular constraints. In a more mathematical sense a MOO problem consists of finding a vector of decision variables which pleases certain constraints and optimizes a vector function containing elements that represent the objective functions (being either maximized or minimized at once). The functions mentioned found a description of performance criteria that is usually in conflict with one another. In consequence, the meaning of optizing consists of finding such a solution that would give the values of the objective functions to be acceptable to the Decision Maker.

A MOO problem consists of four different parts: a set of decision variables, bounds on these decision variables, one or more objective functions and finally constraints. The objective function can be either maximized or minimized. MOO problems in its common form look like the following:

Find the decision variable vector,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{x} \in \mathbb{R}^n \quad (2.13)$$

which minimizes or maximizes the objective functions,

$$f_m(\mathbf{x}), m = 1, 2, \dots, M \quad (2.14)$$

and is subject to n number of bounds,

$$x_i^{lb} \leq x_i \leq x_i^{ub}, i = 1, 2, \dots, n \quad (2.15)$$

J number of inequality constraint(s),

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, J \quad (2.16)$$

and/or K number of equality constraint(s),

$$h_k(\mathbf{x}) = 0, k = 1, 2, \dots, K \quad (2.17)$$

The objective functions $f_m(\mathbf{x})$ or their objective values f_m with $m = 1, 2, \dots, M$, make an objective vector $\mathbf{f}(\mathbf{x})$, which can be written as:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_M(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_M \end{bmatrix} \quad (2.18)$$

There are two types of spaces considered in MOO problems. Firstly, the decision or design space \mathbb{R}^n , which is the real coordinate space of the decision variables containing n dimensions, where each coordinate axis corresponds to a component of vector x . And secondly, the solution or objective space \mathbb{R}^M which is the real coordinate space of the objective functions containing M dimensions, and composed by the objective function values f_m where each coordinate axis corresponds to a component of vector $\mathbf{f}(\mathbf{x})$.

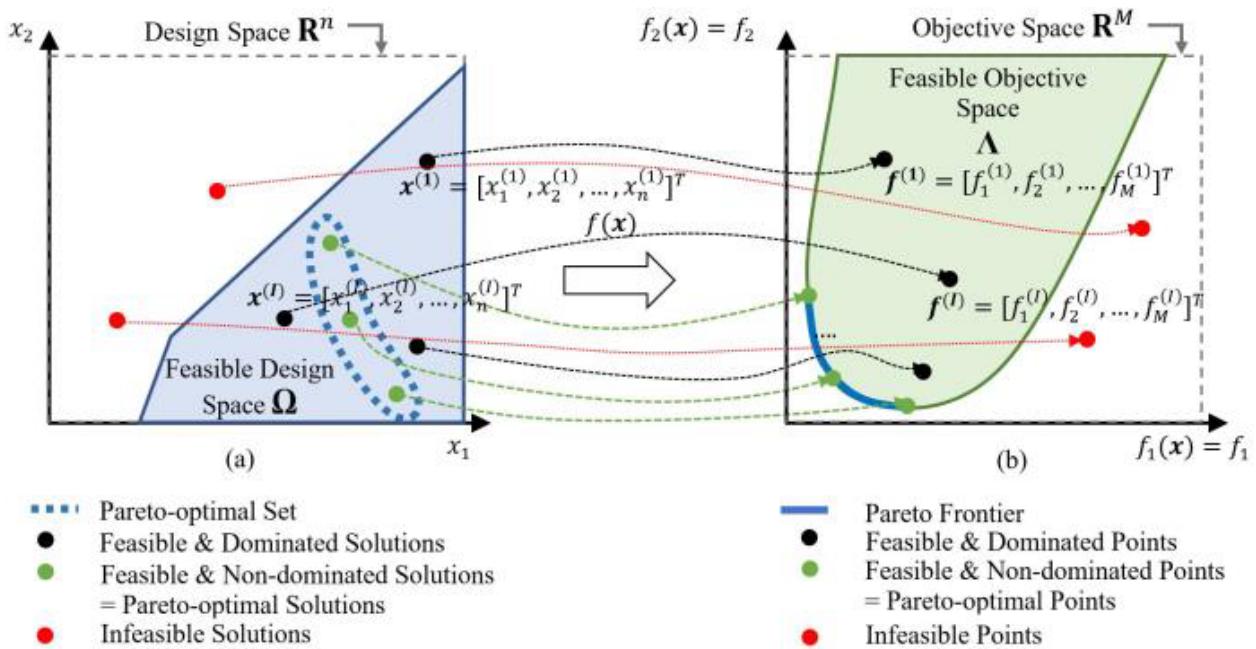


Figure 2.6 Mapping of a decision space (a) onto an objective space (b), where both objectives are meant to be minimized. It contains two decision variables (i.e., 2-dimensions) and two objective functions (i.e., 2-dimensions). [31]

There is a direct mapping between these two spaces, as each solution $\mathbf{x} \in \mathbb{R}^n$ in the decision space will give a corresponding point $\mathbf{f} \in \mathbb{R}^M$ in the objective space, described by: $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T$ i.e., $\mathbf{f} = [f_1, f_2, \dots, f_M]^T$ (T being the transposition of the column to row vector). This mapping dictates the quality of the solution in terms of the objective function values f_m . It is denoted a solution as a decision variable vector and its corresponding objective vector a point. The solutions that satisfy at the same time the objective functions' and constraints and variable bounds form a feasible decision variable space $\Omega \subset \mathbb{R}^n$ and the corresponding feasible objective space $\Lambda \subset \mathbb{R}^M$. This mapping between design and objective space is seen in Figure 2.6.

In MOO each objective has its own fitness landscape. Here, objectives usually compete with one another so that solutions that improve values of a objective function might make the other objective get worse. Therefore, in order to find an optimal solution the concept of dominance becomes paramount.

The concept of optimality has different meanings. In multi-objective problems the aim is to find solution with good compromises or ‘trade-offs’ of conflicting objective functions in an best possible manner, instead of finding a just one solution. Therefore the Decision Maker has the final option to choose the configurations that suits best for him or her.

One of the common meanings for optimality in MOO problems uses the expression ‘Pareto optimality’ [32]. Far from the single-objective optimization problems, where a solutions optimality over another solutions can easily decided by comparison of their objective functions, in MOO problems a solutions optimality or goodness is decided by the idea of dominance that was just mentioned.

This notion of domination is shown in Figure 2.7a. A solution $\mathbf{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ dominates the solution $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ (indicated as $\mathbf{x}^* \preceq \mathbf{x}$) if:

1. When comparing solutions based on their objective functions, the solution \mathbf{x}^* cannot be worse than \mathbf{x} in all objectives.
2. The solution \mathbf{x}^* must be better than \mathbf{x} in minimum one objective.

The mathematical representation of the above principles is,

$$\forall m = 1, 2, \dots, M, f_m(\mathbf{x}^*) \leq f_m(\mathbf{x}) \wedge \exists m = 1, 2, \dots, M : f_m(\mathbf{x}^*) < f_m(\mathbf{x})$$

In the decision space \mathbb{R}^n , the Pareto-optimal set P^* represents a complete set of Pareto-optimal solutions \mathbf{x}^* that are non-dominated in the entire feasible space. There exists a mapping from the Pareto-optimal set of points in the design space, to the Pareto-optimal points in the feasible objective space. This is the Pareto frontier or Pareto-optimal front. Both mathematical descriptions are shown below:

$$P^* := \{\mathbf{x}^* \in \Omega \mid \neg \exists \mathbf{x} \in \Omega, f(\mathbf{x}) \preceq f(\mathbf{x}^*)\} \quad (2.19)$$

$$PF^* := \{f(\mathbf{x}) \mid \mathbf{x} \in P^*\} \quad (2.20)$$

There are some related definitions of special solutions which are often used in MOO algorithms. The nadir objective vector $\mathbf{z}^{\text{nad}} = [z_1^{\text{nad}}, z_2^{\text{nad}}, \dots, z_M^{\text{nad}}]^T$ is the worst value of each objective function from the whole Pareto-optimal set P^* . Not to be confused with \mathbf{W} , which represents the worst objective value of the whole search space (uses the worst feasible function values f_i^{\max} in the complete search space).

The ideal objective vector $\mathbf{z}^0 = [z_1^0, z_2^0, \dots, z_M^0]^T$ represents the best value of each objective function which corresponds to the complete Pareto-optimal set P^* . In some special cases a solution having an objective value which is slightly better to any existent one in the decision space is needed. This solution is the utopian objective vector \mathbf{z}^u . The aforementioned special solutions can be found in Figure 2.7b.

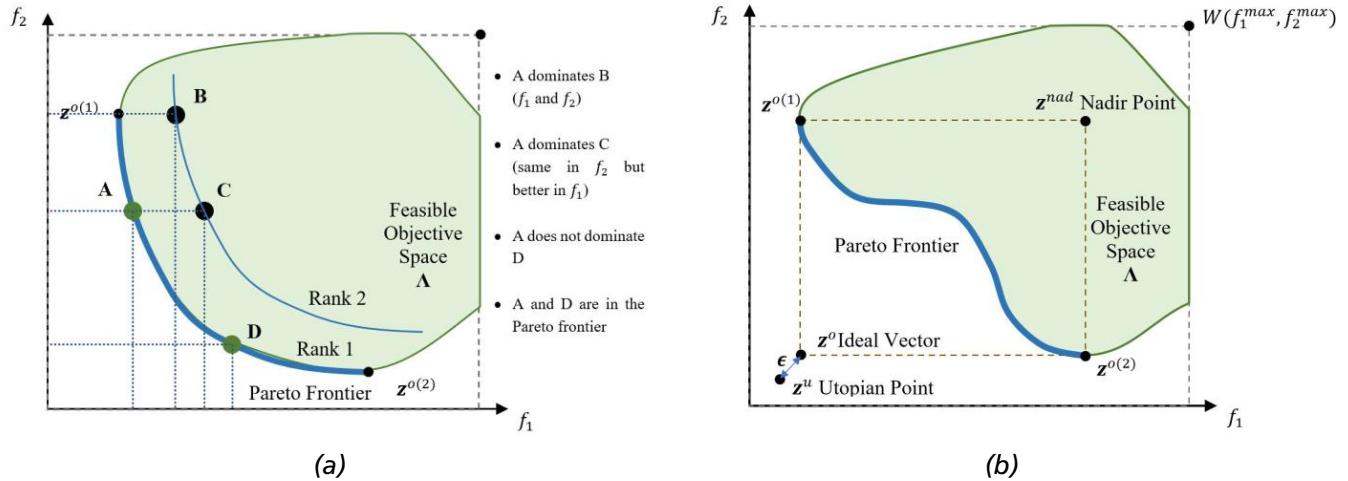


Figure 2.7 a) The concept of domination illustrated in a minimization problem of two objective functions, and b) special solutions and points in MOO. [31]

The ideal and nadir vectors can then be used for the normalization of the objectives in the Pareto-optimal region,

$$f_i^{norm} = \frac{\mathbf{f}_i - \mathbf{z}_i^0}{\mathbf{z}_i^{nad} - \mathbf{z}_i^0} \quad (2.21)$$

The normalized values can later be used for optimization purposes (weighted-sum approach) or to indicate the performance of the algorithm in some cases [31].

Because of the importance of trade-off solutions from the conflicting objectives, where no optimal solution can be applied to one objective alone, there are two suggested ideal goals of MOO [33]:

1. Get the set of solutions to converge towards the Pareto-optimal front (**convergence principle**)
2. Make these set of solutions diverse enough so that it represents the complete range of the Pareto-optimal front (**diversity principle**).

2.2.2 Evolutionary Multi-Objective Optimization (EMO)

There have been different methods over time in order to solve the MOO problems.

Classical Methods

Classically, obtaining Pareto-optimal solutions was done by executing independent single-objective optimizations multiple times, each of them trying to find a single Pareto-optimal solution. Some scalarizing approaches, like ϵ -constraint approach or weighted-sums, are used to transform multiple objectives into single-objectives.

Consider Figure 2.8a, where multiple independent single-objective optimization processes move/improve throughout the objective space reaching different Pareto-optimal solutions in end. Throughout the course of an optimization process the selected algorithms overcomes different struggles, like local optimal solutions, infeasible regions, etc., in order to converge towards the global optimum so-

lution. It is also key for an algorithm to be able to do all of this in the shortest amount of time possible.

Because these methods consist in a way of repeated single-objective optimization, they are performed independently from one another, the information about the failure or success of previous simulations is lost, and cannot be used to speed up the subsequent runs of the optimization. For this reason these memory-less methods demand a large computational time to get a collection of Pareto-optimal solutions, and furthermore, a good distribution among Pareto Front points can never be guaranteed (see 'diversity principle' in Section 2.2.1)

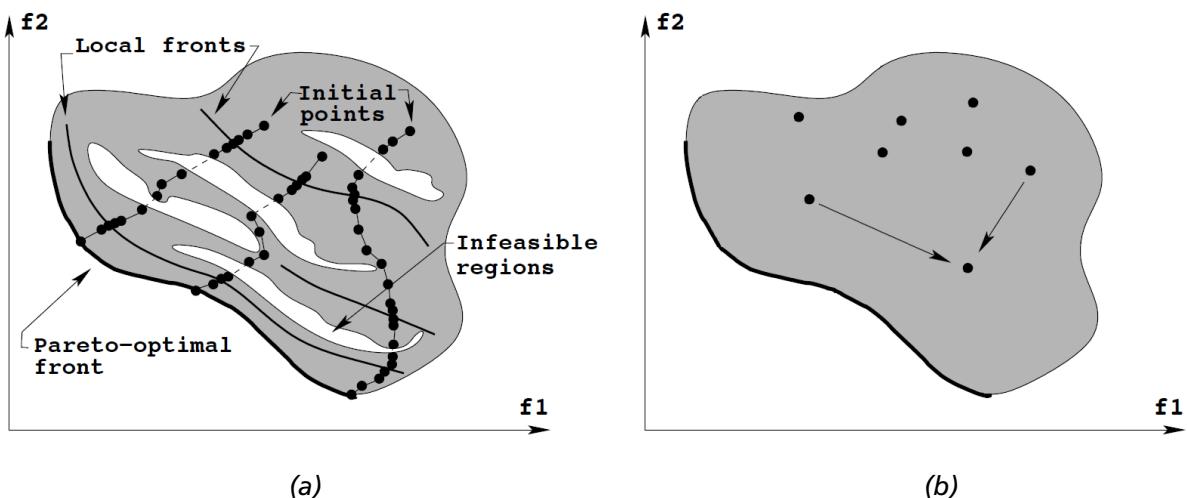


Figure 2.8 a) Classical method to solve MOO problems by performing many independent single-objective optimizations [33], and b) EMO methods, where recombination takes place between solutions and other population members, leading to new solutions that typically expand through the objective space.

Evolutionary optimization (EO)

Evolutionary Algorithms (EAs) are population-based meta-heuristic optimization algorithms which use mechanisms inspired by biological evolution. Solving optimization problems using EAs is known as Evolutionary optimization (EO). The EO principles are different from classical optimization techniques as:

- EO don't typically use gradients or derivative information in its search process
- The EO's use a population approach with more than one solution in an iteration, in contrast to the explained cases of classical algorithms where one solution is updated in each iteration. Among other advantages, using a population of solutions provides a parallel processing ability which accelerates the search time; it permits multiple optimal solutions; and it allows to normalization of decision variables, constraints and objective function.
- These procedures employ stochastic operators, in contrast to the deterministic operators in most classical methods.

An EO begins its search with a population of solutions usually created at random within a specified lower and upper bound on each variable. Thereafter, the EO procedure enters into an iterative

operation of updating the current population to create a new population by the use of four main operators: selection, crossover, mutation and elite-preservation. The operation stops when one or more pre-specified termination criteria are met.

EMO Methods

The ideal way of performing MOO [34] follows the below procedure where:

- 1** Detect various non-dominated points being the closest as possible to the Pareto-optimal front.
- 2** Select one of these non-dominated points by using higher-level information.

When using EO for more than one objective (i.e., Evolutionary Multi-Objective Optimization (EMO)) these ideal steps are always followed due to the nature of the evolutionary algorithms. The success in employing bio-inspired search models to solve MOO problems has been much greater than in other fields of computer science. Some examples of these algorithms that have become standard solvers when solving MOO problems are SPEA2, NSGA-II, MOEA/D between others [35].

The EMO is a parallel process by nature. While a population makes progress towards the Pareto Frontier, its values and their combination reflect this reality. When recombinations takes place between the present solution and other population components, this information gets shared through variable blending and exchanges, that can be seen in Figure 2.8b, making the task of finding multiple trade-off solutions a parallel one.

EMOs may not assure one to arrive to the Pareto Front and obtain the Pareto-optimal points, as other deterministic linear or convex problems may do. But these procedures have nevertheless crucial operators that constantly improve the non-dominated points every generation according to the principles of convergence and diversity from Section 2.2.1.

2.2.3 EMO Algorithms: Genetic Algorithms (GA) and NSGA-II

There is a variety of popular Evolutionary Algorithms, but undoubtedly the best-known EA is the Genetic Algorithm (GA) [36]. For the case of MOO, one of the most successful and popular algorithms [33] [37] [38] is a type of GA called NSGA-II which stands for 'Non dominated Sorting Genetic Algorithm' [39].

This MOO algorithm that tries to obtain the best Pareto-optimal solutions, is characterized by using the principle of elitism, using a direct diversity preserving mechanism, and prioritizes solutions that are non-dominated.

By means of using the parents population P_t and genetic operators, the offspring population Q_t is created during a particular generation t .

The offspring population Q_t follows the usual GA procedure described in Figure 2.9a (without the Termination Criteria) and is the following [40]:

1. The process starts with a set of individuals (i.e., a population), where each individual is a solution to the problem in hand. For our NSGA-II case it is P_t . These individuals are a set of parameters commonly known as Genes, which are joined together into a string to form a Chromosome or solution.

2. Next step is the computation of the fitness value $f(x)$ of each individual in the current population. Individuals are chosen for mating based upon their fitness values and as per the desired values of each one.
3. Now in the selection phase the fittest individuals are selected and let transfer their genes to the next generation. A pair of parent individuals are selected on the basis of their fitness scores. The individuals with the highest fitness scores have higher chances of being selected for reproduction.
4. Next, the crossover is performed by choosing a random point on the chromosome, in which the parents' will exchange parts. Through the process of crossover a new offspring is brought up based on the exchange point chosen from the parents.
5. Finally, mutation takes place after crossover is done. Mutation applies random changes to one or more genes in the chromosome in order to produce new offspring, helping to stay away from local optima.

For the case of a simple GA, after the crossover/mutation has taken place the termination criteria is checked as seen in Figure 2.9a. But for the multi-objective case of NSGA-II the process does not end here. The whole procedure for the NSGA-II algorithm is shown in the below pseudo-code and also in Figure 2.9b.

Pseudo-code of the NSGA-II algorithm

```

1:  $P(0)$  GenerateInitialPopulation()
2:  $t \leftarrow 0$ 
3: Evaluate( $P(0)$ )
4: while not StoppingCriteria() do
5:    $P'(t) \leftarrow$  SelectionTechnique( $P(t)$ )
6:    $P''(t) \leftarrow$  Crossover & Mutation( $P'(t)$ )
7:   Evaluate( $P''(t)$ )
8:    $P(t + 1) \leftarrow$  Ranking( $P''(t) \cup P(t)$ )
9:   if  $P(t + 1)$  not full then
10:     $R \leftarrow$  CrowdingDistanceSelection( $P''(t) \cup P(t)$ )
11:     $P(t + 1) \leftarrow P(t + 1) \cup R$ 
12:   end if
13:    $t \leftarrow t + 1$ 
14: end while

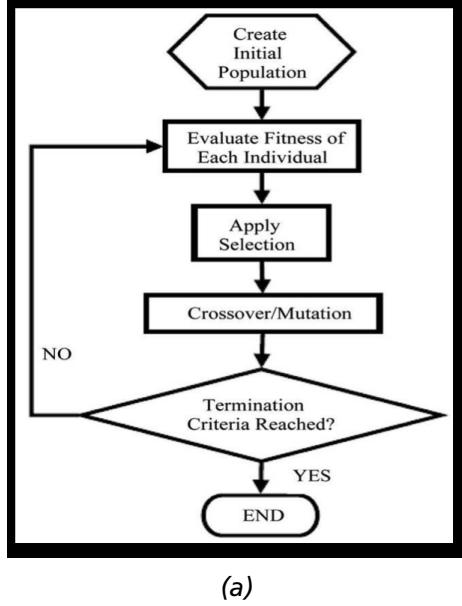
```

The two populations P_t (parents) and Q_t (offspring) are then combined to create a new population R_t , twice the size. This population R_t is then classified into different classes according to their domination.

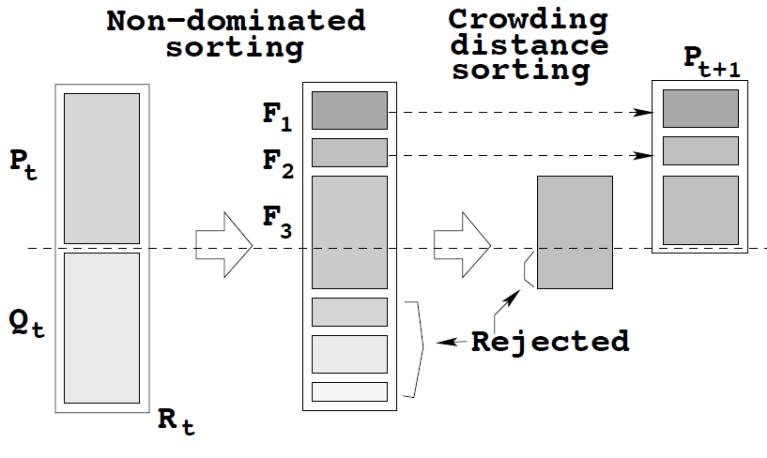
Afterwards, the new population of individuals (half the size of R_t) is created by adding the points from R_t . In the first place the non-domination front is added, and subsequently the points from the second non-domination front, and like this it follows until completion. Because not all fronts can be added, all fronts which could not be accommodated are removed.

When the last possible front that can be accommodated is considered, there may be more points in the front than available slots in the new population. This is shown in Figure 2.9b. For this purpose, points increasing the diversity of the front are preferred inside of the Pareto Front. This crowded-sorting of the points is done in descending order of their crowding distance values. This crowding

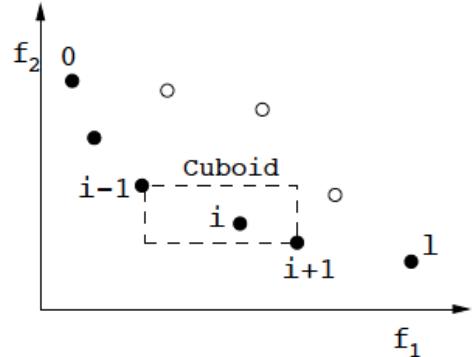
distance d_i of point i indicates the emptiness of other solution points in the objective space around i . The crowding distance d_i is simply calculated by an estimation of the perimeter of the cuboid that is formed by using the closest neighbors in the space as the vertices, and is shown in Figure 2.9c.



(a)



(b)



(c)

Figure 2.9 a) Flow chart of a typical GA [41], b) Schematic of the NSGA-II procedure [33], and c) The crowding distance calculation [33].

2.2.4 Decision Making and Performance Indicators in MOO

Decision Making

In optimization of any kind, the user usually requires one final solution. In MOO, where many solutions are available, the user must now choose between the different optimal solutions that are provided. This decision is not usually an easy one, as it requires knowledge of many factors that could be based in experience or non-technical aspects. An evaluation of the advantages and

disadvantages of each of these solutions must be done and compared to make a choice. The procedure of selecting a single solution from an obtained set of solutions is called Decision Making.

Often MOO methods are classified with respect to the role of the Decision Maker as [31]:

No-Preference Methods: In these cases the Decision Maker is not available, and provides only one computed solution (ex. global criterion method).

A-Priori Methods: Here the Decision Maker expresses preferences before the optimization process. The DM is first asked about his preferences, and then these are optimized accordingly. Only Pareto-optimal solutions which are of interest to the Decision Maker are made, which in many cases it is quite difficult for him to know before other solutions are already known.

In many cases like the 'Weighted Sum Method', the preferences of the Decision Maker are later employed in a technique that is called scalarization, consisting of the accumulation of multiple objectives, converting the MOO problem into a Single-objective optimization one. In the case of the 'Weighted Sum Method' weights $w_m, m = 1, 2, \dots, M$ are supplied by the Decision Maker for guiding the process of selection. There are also a-priori methods where the scalarization process doesn't turn the problem completely into a single-objective one, but there may be instead 'preferred regions' of space where the solutions can only exist, like in the case of the Cone Domination method. And in other cases also algorithms like R-NSGA-II, which follows the general outline of NSGA-II with modified survival selection, where there is the addition of 'reference points' which guide the optimization process as shown in Figure 2.10a.

A-Posteriori Methods: In these methods the Decision Maker expresses preferences after the optimization process. The participation of the Decision Maker is not needed during the process which creates all the solutions. It depends instead on methods that can be utilized to create a diversified set of Pareto optimal solutions which are distributed evenly on the Pareto Front. The Decision Maker then by examination and negotiation of the merits of each different solution, makes a final decision. While having an understanding of the entire Pareto Optimal set, it is still difficult to develop solutions that can create the Pareto Front, being in occasions a more of a burden being subjected to decide between such large number of solutions.

MOO evolutionary algorithms are one of the most used examples of utilizing 'a-posteriori' preference, example of which is the already described algorithm NSGA-II, which will be central for the developed program in this paper. To later decide between all of this points, one can use strategies like Compromise Programming, which consists of using a scalarization method and use it for post-processing, once all the Pareto Solutions have been found, or Pseudo-Weights, where the weights w_i gives the normalized distance to the worst solution regarding each objective i .

Interactive Methods: It is based on an progressive expression of preferences, where the Decision Maker's preferences are added into the searching process (ex. Satisficing trade-off method).

Performance Indicators

For SOO algorithms the comparison regarding performance is rather simple because each optimization run results in a single best solution. In MOO however, each run returns a non-dominated set of solutions, and these solutions should have according to the principles in section 2.2.1 a good 'convergence' to the Pareto-optimal front and a good 'diversity' in obtained solutions.

In order to get an idea of the performance of a MOO algorithm, a performance metric must be used. There are many different metrics proposed, but just a couple top indicators [44] will be shown according to the knowledge or lack of it of a previous PF.

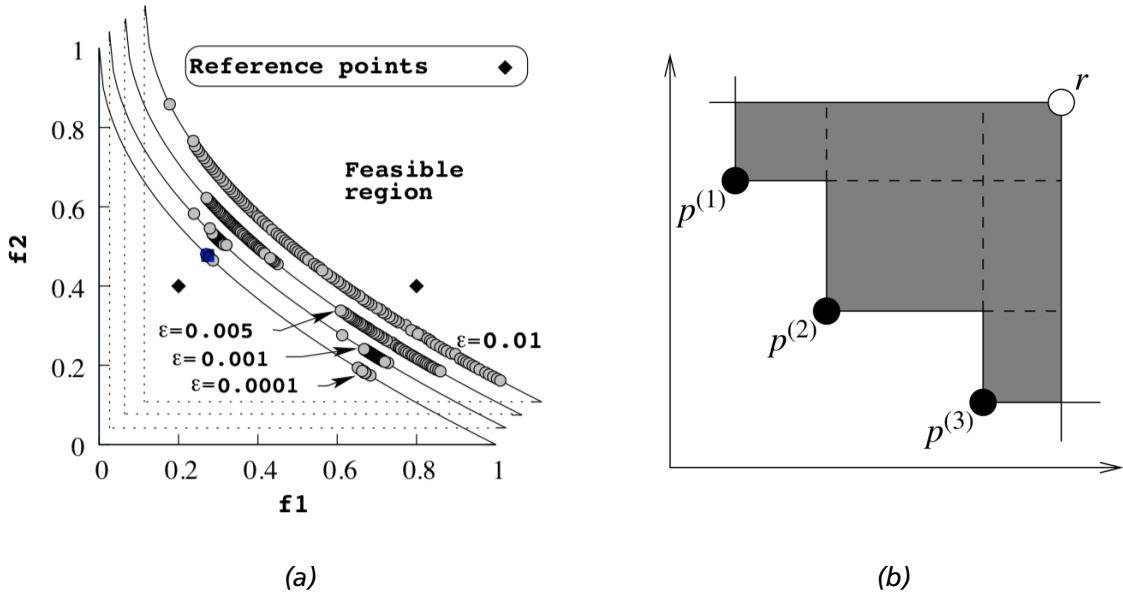


Figure 2.10 a) Reference-point based MOO used by the R-NSGA-II algorithm [42], and b) The hyper-volume indicator in the two objective case [43].

First, for the cases where the Pareto Front is already known or approximated, one can use the generational distance (GD) or the inverted generational distance (IGD) metric. The first one measures the distance between the actual solution PF_{sol} and the best Pareto front estimation which is done by a combination of points non-dominated from many multi-objective solutions of a particular test case, named PF^* . GD precisely relates to the average Euclidean distance between each point in PF_{sol} and the nearest point in PF^* . For the case of IGD, as its name suggests it is the inverted case, where now the average Euclidean distance between each point in PF^* and the nearest point in PF_{sol} is measured.

For the cases where the PF is not yet known or approximated, the popular hypervolume metric can be used. This metric represents the volume of the objective space which is covered by the individuals from a non-dominated solutions set (i.e., solutions that belong to a PF). The volume is fixed by a point that called the anti-optimal point r , defined as the worst solution inside the objective space, and the second optimal point that is computed by the proposed solution method. It is necessary to normalize the objective function values before the calculation of the hypervolume. It is shown in Figure 2.10b.

3 Methods

The aim of this thesis is the implementation and evaluation of multi-objective wind farm optimization problems with the objectives of yield maximization and load minimization, and the subsequent analysis of a so-called Pareto front of Pareto optimal solutions. The optimization problems are solved by using induction control and wake steering control as decision variables, and simulation data provided by the IWES in-house software MoWiT from a collaborating Fraunhofer IWES team in ‘Bremerhaven’.

In other words, the objective is to create and develop a fully functional piece of code as an integrated part of the wind farm and wake modelling software tool Flappy developed at Fraunhofer IWES. This code when executed, will perform MOO calculations in user-defined simulated wind farms, whose analysis and results focus in methods for mitigation of wake effects or Active Wake Control (AWC) methods, i.e., maximizing the power production of a whole wind farm, while at the same time, trying to reduce the fatigue loading on the wind turbines, by either curtailing turbines or performing yaw-misalignment on them.

Firstly, the software tools and packages that are needed will be explained together with the necessary data sets and other instrumentation. After that, the development of the multi-objective wind farm optimization programs will be described, for solving the case of induction control and of wake steering control. To finalize, the last section will deal with the testing phase of the optimizer programs, where the wind farm calculations and the optimization process itself will both be thoroughly evaluated for a correct behaviour and results.

3.1 Measurement Devices, Datasets and Software Tools

The development of this optimizer program is achieved by using different software instruments and tools together with provided data sets for the calculations. The following will be explained in this section.

Farm Layout Program in Python (Flappy)

Flappy [45] is a In-house wind farm and wake modelling tool developed here at Fraunhofer IWES. It is used for fast calculation of wind farm yield results, and other wind farm related quantities. It is a software tool similar to the more popular FLORIS [46] (FLOw Redirection and Induction in Steady State) developed by NREL, which also does wind farm modelling and optimization.

This in-house software can make fast wind farm and wake modelling calculations (no CFD calculations), and can handle large time series and statistical wind input data. It has a broad modeling grid-less platform for wind farms and wake simulations and can be coupled to optimization libraries, for use, for example, in single and multi-objective wind farm optimization problems. This last property will be necessary for our optimization calculations and optimization libraries such as PYMOO and PYGMO (which will be later explained) will be coupled for the usage [47].

Flappy creates and fills a wind farm, by adding turbines using selected models. It will first fill what is referred as a “model book”, which is a script containing all the models which are necessary to run the program. These models are:

- ***Rotor models*** which calculate effective data as seen by the wind turbine rotors: Rotor effective wind vector, wind speed, turbulence intensity and air density.
- ***Turbine models*** calculate variables from current Flappy variables and flow field.

- **Controllers** are responsible for running the turbine models.
- **Wake models** calculate wake deltas of wind vectors and turbulence intensity scalars, at any set of points.
- **Wake frames** calculate the coordinates (x, y, z) for input points, at which the wake models should be evaluated.
- **Wake superposition** combines wake effects.

Once the wind farm has been made with its specific characteristics and the "model book" has been filled, the next step will be to create the initial flow states of the wind farm which will be later altered by the wake calculations in all affected turbines. Each state is a single situation for which the wind farm is evaluated (a set of fixed values for wind speed, wind direction, turbulence intensity and air density for the case of a uniform inflow). In practice, Flappy evaluates many such states using time series and wind rose data which is vectorized and parallelized in order to reduce the computational costs of these calculations.

After creating the wind farm and the initial flow states, it is now possible to run calculations of wind farm quantities for all states. These main calculations happen with the now initialized models and data arrays, once the initial calculation data which does not involve wake interference has been processed. The main wind farm calculations are divided into groups of 'chunks' which allow to perform computations in parallelized manner (or in series for non-parallel cases).

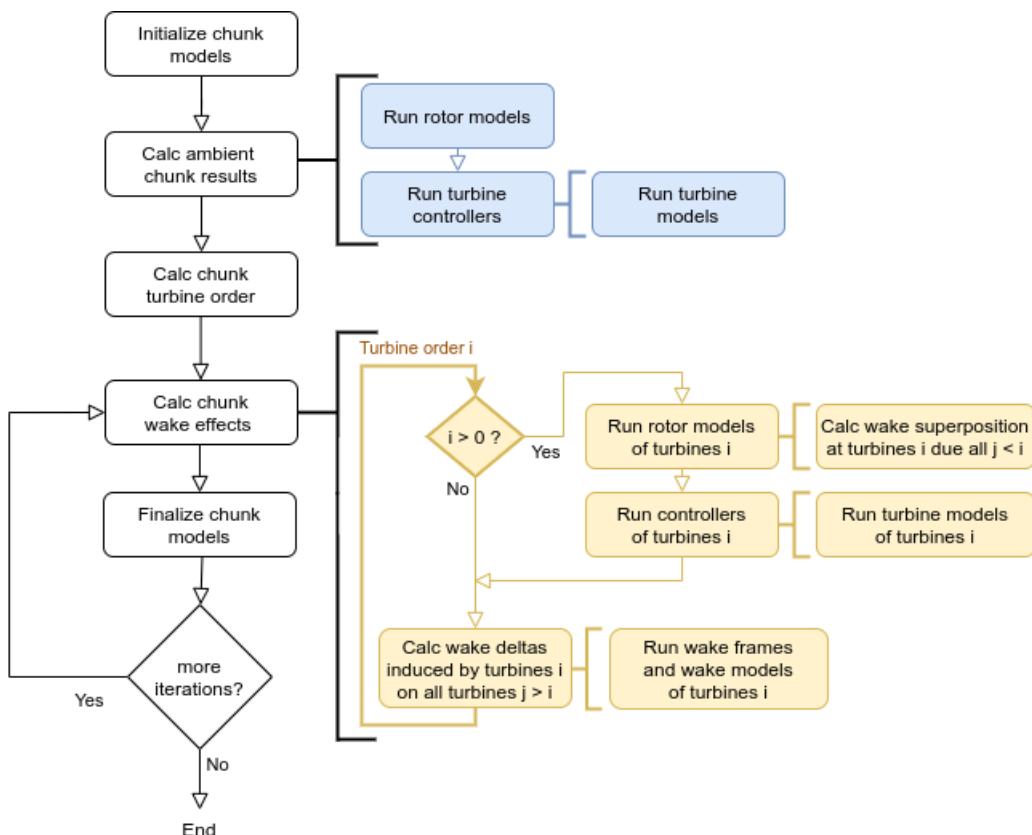


Figure 3.1 The working procedure of Flappy for the calculation of one chunk [45]

Inside each chunk calculation, the models above mentioned will be now used, first to calculate initial or "ambient" results, by running the rotor models which calculate the rotor effective data

and then with this one the turbine controllers run the turbine models selected. After this, once the turbine order is calculated, the models will be used again to calculate the wake effects running again the rotor models, but taking into account now the wake superposition and later again the turbines controller with all turbine models. The wake deltas induced by the turbines are calculated by running the wake frames which provides coordinates in the frame of reference in which the wake models that were selected will be evaluated. This process is repeated through all turbines, and if after that when the chunk is finalized there are no more chunk iterations, the process will end. This working process can be better seen in the flow diagram in Figure 3.1.

This so far, is the basic framework for any Flappy calculation. For the cases where optimization of the wind farm also takes place, more elements are needed (See section 2.2.1). It will be first necessary to define and initialize the optimization problem, using the wind farm and states created, and by determining the decision variables \mathbf{x} with its boundaries x_i^{lb} and x_i^{ub} . Once this is done, the objective function or functions $\mathbf{f}(\mathbf{x})$ from Equation (2.18) will be also defined and added to the problem, together with optimization constraints $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ if there are.

With our objective function and constraints defined, it is possible now to create and initialize the optimization solver. This solver uses a Python-based optimization library to run the optimization problem such as PYMOO or PYGMO which will be shortly explained in the next section.

Once the optimization process has finished according to its termination criteria, the final results together with other additional visualizations or supplementary actions will be presented for its analysis. Flappy is the core software instrument inside of which the multi-objective wind farm optimizer will be designed as a part of it.

Python Optimization Libraries

As mentioned in the section from before, the wake modelling tool Flappy can be coupled to optimization libraries in order to solve wind farm optimization problems. By using the software library IWES-OPT from Fraunhofer IWES which is the actual interface to external optimization libraries, two different Python optimization libraries are used in Flappy.

Firstly, the Parallel Global Multi-objective Optimizer, **PYGMO** [48], a scientific library for parallel optimization in C++ (originally) and Python. Created inside the Advanced Concepts Team of the European Space Agency, this library offers a large number of optimization algorithms and problems for single and a few multi-objective cases.

This optimization library is made for the easily distribute optimization tasks over many processors. It uses the so-called “generalized island-model paradigm” a special mechanism for parallel optimization. It is used for many single-objective problems and some multi-objective problems in Flappy.

The other optimization library which has been recently introduced for use in Flappy is **PYMOO** (Python Multi-objective Optimization) [49]. This framework provides single, multi and many (suited for three or more objectives) objective optimization algorithms and problems.

PYMOO provides a variety of options for key aspects of a multi-objective optimization task, offering performance metrics for evaluation of optimization problems, many visualization techniques for showing the Pareto fronts in multiple dimensions and its trade-off solutions, and a few approaches for decision-making.

It is this special emphasis in MOO that makes PYMOO special, particularly for the use with the algorithms NSGA-II (See section 2.2.3) and NSGA-III, as this optimization library has been devel-

oped collaboratively with the main author of these algorithms, and therefore PYMOO is formally recommended for the official benchmarks of these algorithms.

As the algorithm NSGA-II will be used for the development of the multi-objective wind farm optimizer, and due to the MOO significance of this framework, PYMOO will be selected as the optimization library of use in this development.

Simulation Data from MoWiT

As observed before, when using the software Flappy one must make use of 'Turbine models' which calculate variables from current Flappy variables and flow field. Our multi-objective wind farm optimizer has the objectives of yield maximization and load minimization, and as decision variables the induction control (i.e. curtailment) and in the second case wake steering control (yaw misalignment) of the turbines.

The 'Turbine models' required in this case will need to take as input wind speed, turbulence intensity and the corresponding decision variable (curtailment of turbines or yaw misalignment depending of the program), and provide as output the thrust coefficient and the objectives of power and the selected DEL. These models are derived from lookup-tables, which transform state-turbine input to output variables, based on a data table file, that for this program will arise from simulation data with MoWiT.

This database contains load simulation data obtained with the IWES in-house software MoWiT (which includes all major components necessary for load calculations of state of the art onshore and offshore wind turbines). It provides information about a generic direct drive IWT 7.5 MW simulated turbine, given in the form of a lookup-tables that are based on yaw misalignment/curtailment, rotor effective wind speed and turbulence intensity.

3.2 Developing the Multi-Objective Wind Farm Optimizer for Induction Control and Wake Steering Control Problems

Now that the environment and necessary tools for the program have been explained (Section 3.1) together with the necessary background about wind energy and optimization processes (Section 2), it is the moment develop inside of the IWES wind farm modeling Flappy the programming code that will solve multi-objective wind farm optimization problems, with the objectives of yield maximization and load minimization, by using induction control and wake steering control methods.

It is important to define first the starting point of the future program. As seen in Section 3.1 Flappy already allowed for single-objective wind farm layout optimization cases with the objective of yield maximization. But in the recent years, multi-objective wind farm layout optimization cases were introduced into Flappy, opening the possibility for MOO in Flappy.

Now, this new multi-objective optimizer being still in a very experimental phase, brings us to the object of this thesis project, where one of the aims is to expand MOO in Flappy for the use with different decision variables (curtailment and yaw misalignment) and additional objectives (DEL). To later perform more rigorous test in order to observe a correct behaviour of the optimizer. To use real simulation data obtained with the IWES in-house software MoWiT. And also the goal to bring new tools and gadgets to improve MOO which has been done introducing PYMOO optimization library (Section 3.1) and new concepts for the performance and decision making of solutions (Section 2.2.4).

There are two main different Flappy code programs that are the scope of the thesis project: A multi-objective wind farm optimizer for induction control problems, i.e., using curtailment of the wind turbines as decision variables. And a second program dealing with multi-objective wind farm optimization for wake steering control problems, using instead yaw misalignment of the turbine as a decision variable. Some of the basic wind farm characteristics that will be used for the results calculations in Section 4 are shown in Table 3.1

Table 3.1 Wind farm characteristics for calculations

rotor diameter (meters)	hub height (meters)	wind direction (degrees)	air density (kg/m ₃)
120	100	270	1.225

3.2.1 Multi-Objective Optimizer for Induction Control Problems

The program developed here will solve and analyze the multi-objective wind farm optimization problem using the curtailment of the turbines, within an allowed boundary, as decision variables, and yield and loads as the objectives.

The development of this code program will follow the structure used in Flappy as described in Section 3.1. A model book containing the turbine models which calculate variables from current Flappy variables and flow field is made. As described above in Section 3.1, the turbine model will use lookup tables with simulation data provided by MoWiT.

For the case of induction control problems, there is a turbine model taking as input the wind speed and amount of curtailment of the turbine, and by linear interpolation of this data it outputs the corresponding Power generated and thrust coefficient C_t , by means of a '.csv' file containing the data simulated results . A second turbine model gets the interpolation values for the DEL. Taking as input the wind speed, TI and amount of curtailment, and in this case by means of a polynomial interpolation (which varies according to the type of DELs being calculated) it outputs the corresponding DEL values.

The turbine models will all be run by the turbine controller, Flappy contains so far a simple default controller that runs all models. But firstly, for the wind farm calculations a rotor model which calculates the effective data as seen by the wind turbine rotors is required. A Gaussian model is used to obtain partial wake calculations.

The reason for using a Gaussian rotor model is that the wake model used is also a Gaussian one. The wake model calculates wake deltas of wind and TI for partial wake calculations. The one being used here is the Gaussian model from Section 2.1.3, specifically the wind deficit wake model from Bastankhah and Porté-Agel in 2016 [12], which includes the prediction of far-wake velocity for a yawed turbine (particularly useful for the wake steering control case) following Equation (2.9).

For this wake model there are some fixed parameters that must be set. Firstly, the wake growth model constant k , or instead the factor k_{TI} that computes k from TI if k is not provided. Also there is a coefficient k_b , that is added to the computation of k from k_{TI} . The near wake scaling parameters α and β for TI or the C_t term respectively. And also indicating if the wake recovery coefficient is computed on ambient TI values, or just TI values. The parameters used are found in Table 3.2.

Also a TI wake model is used to calculate the TI scalars, at any set of points. A TI empirical correlation wake model based on Crespo and Hernández [50] is used. They proposed simple expressions for the estimation of the turbulent kinetic energy k , and its dissipation rate ε . To achieve the turbulence spectrum found in the wake, it is necessary to assume a law for atmospheric turbulence, where

Table 3.2 Fixed parameters employed for wake models calculations.

Wake Model Parameters	Bastankhah and Porté-Agel 2016 Wake Model values	Crespo and Hernandez 1996 TI Wake Model values
k	None	None
k_{TI}	0.2	0.3837
k_b	0	0.003678
initial/ambient TI	Yes	Yes
α	0.58	
β	0.077	

particular values of turbulence length and velocity are calculated by combining k and ε , so that when the wake effect is insignificant the spectrum of the flow that was not perturbed is then recovered. The fixed parameters used for this TI wake model can also be found in Table 3.2.

Next, a wake frame is selected to calculate the coordinates for input points, at which the wake models should be evaluated. The type of wake frame model used for this program is oriented with respect to the ambient wind vector at the wake causing rotor.

The program will only deal with a single uniform ambient flow state. This means there will be fixed parameter for wind direction (typically at 270°), wind speed, TI and air density throughout the entire optimization calculations. The simple reason for the moment being, is that adding states increases the computational time required, especially using large timeseries, as the optimization process must compute each Flappy calculation iteratively for each individual of a population, every single generation.

A wake superposition model is used to combine the wake effects. For the case of the wind superposition a quadratic model is used, meaning that the direction of the total delta wind vector used is determined by the weighted sum of the delta wind vectors, where the weights are linear for the direction. It is also necessary to obtain the superposition for TI of every turbines effect at each target point. A model that calculates the nearest turbine's wake is considered for this.

At this point, the problem can be constructed, as all elements for the Flappy calculations are now established. As described in Section 3.1 "Farm Layout Program in Python (Flappy)" and following the optimization principles from Section 2.2.1, the optimization problem is initialized using as decision variables \mathbf{x} the curtailment of turbines, from a maximum allowed range (due to the simulation data provided) between $x_i^{lb} = 0$ and $x_i^{ub} = 0.5$ (where 0 indicates an uncurtailed turbine working at full capacity, and 1 indicating a total shut down of the turbine).

Objective functions $\mathbf{f}(\mathbf{x})$ are also added to the problem. The first objective consists of maximizing mean wind farm power. The value of this objective function is expressed from Equation (2.14) as:

$$f_1(\mathbf{x}) = 1 - \frac{\sum_{i=0}^n P_i}{n \cdot P_0} \quad (3.1)$$

Where n is the number of turbines in the wind farm, and P_0 is used for scaling, indicating an initial Power value when the turbines are completely undisturbed. This function is then minimized, creating a double minimization problem for optimization together with the loads Equation 3.2. The maximum generated power by the uncurtailed simulated turbine IWT 7.5 MW form MoWiT reaches a maximum of just below 7500 kWh, therefore the parameter of this initial Power is set to $P_0 = 7500$. The scaling of the Power P to the value in Equation (3.1) is done to facilitate the optimization process.

The second objective consists in the minimization of the mean wind farm DELs. Its value as objective function is:

$$f_2(\mathbf{x}) = \frac{1}{n} \sum_{i=0}^n \text{DEL}_i \quad (3.2)$$

In the case of the DEL (See Section 2.1.4), there are many types of loads that could be considered for the calculations. Primarily, the loads considered individually as objective functions are the edgewise blade root bending moment M_x (edgewise DEL from now on), the flapwise blade root bending moment M_y (flapwise DEL from now) and the combined tower bending moments, i.e., $\sqrt{M_x^2 + M_y^2 + M_z^2}$ (tower DEL from now), all in units of newton meter (Nm). For the results and discussion of this paper just one objective being the flapwise DEL M_y will be used for the analysis of the Pareto Frontiers together with the power production.

Now, all the components for solving a MOO problem have been established, as the last component would be the equality and inequality constraints $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$, but they are not defined for this problem. At this point the multi-objective optimizer can be called to begin the optimization process. The optimization library PYMOO will be used (See section 3.1)

An evolutionary multi-objective optimizer will be used due to its distinctive advantages (See Section 2.2.2). The evolutionary algorithm used for this problem is the Non dominated Sorting Genetic Algorithm or NSGA-II. The procedure this algorithm will follow is fully described in Section 2.2.3 and in the pseudo-code of this section.

It is first crucial to determine the operators and parameters of the algorithm NSGA-II. The sampling process defines the initial set of solutions which are the starting point of the optimization algorithm. A random sampling will be used in this case. After that, at the beginning of the mating process, parents(individuals) from a population need to be selected to be mated using the crossover operation. The method used will be tournament selection, which involves running several "tournaments" among a few individuals chosen at random from the population, and the winner of each tournament (the one with the best fitness values) is selected for crossover.

Using the crossover operator the information of two parents is combined to generate new offspring. A simulated binary crossover will be used. Real values can be represented by a binary notation and then a point crossovers can be performed. Simulated binary crossover simulates this operation by using a probability distribution simulating the binary crossover. Also mutation is necessary to maintain genetic diversity. Mutation alters one or more gene values in a chromosome(individual) from its initial state. Then polynomial mutation will be employed, which follows the same probability distribution as the simulated binary crossover just described.

The crossover and mutation both have a probability factor, in the case of the crossover it means the number of times a crossover occurs for chromosomes in one generation, i.e., the chance that two chromosomes exchange some of their parts (100% crossover rate means that all offspring are made by crossover; 0% the new generation of individuals is exactly the same as the old one) In the case of mutation, this rate determines how many chromosomes should be mutated in one generation. Moreover, there are eta hyper-parameter for mutation and crossover used to determine the similarity among selected parents and children. A high eta value will produce children resembling their parents, while a small eta value will produce much more different solutions. These parameter values in genetic algorithms greatly determine the quality of the final solution [51]. For the optimizer in hand the following will be applied:

- Crossover rate: **0.9**
- Crossover eta: **15**
- Mutation rate: **0.033**
- Mutation eta: **20**

These default values have been selected following the guidance of common practices for single and multi objective genetic problems [40] [49] [34]. Lastly the concept of elitism, which is of key importance in MOO, consisting of a reserved number of individuals which will always be transferred to the new created population in the new generation. A ratio of half the population used as elitism will always be maintained for the program analysis.

As the optimization process develops, each individual of the population at a point contains the decision variables (selected turbines), where each variable or gene contains a different value for curtailment. As these populations go through the entire process of the algorithm (See pseudo-code in Section 2.2.3), a new improved population is being created after every generation, approaching progressively towards the Pareto Front. This process repeats itself until the termination criteria has been reached, which can be set as a number of generations, based in the improvement of solutions after generations, after a given time frame, etc. The whole process explained in this section is exemplified in Appendix A, where a simple case is shown with just a population of 6 individuals and over 3 generations.

3.2.2 Multi-Objective Optimizer for Wake Steering Control Problems

A second script dealing with multi-objective wind farm optimization for wake steering control problems is also developed, using instead yaw misalignment of the turbine as a decision variable and yield and loads as the objectives. The general methodology for the development of this program follows the one explained in Section 3.2.1. The specifics and differences for the case of wake steering control will be explained in this section.

Similar to the induction control program, this one uses a Gaussian rotor model along with the Gaussian wake model from Bastankhah and Porté-Agel in 2016 [12], which is chosen specifically to help with wake steering control problems, as it includes the prediction of far-wake velocity for a yawed turbine shown in Figure 2.3. Also the same TI wake model from Crespo and Hernández [50] is used, which is specially important for yaw misalignment due to its impact when yawing the turbines.

The turbine models used in this case are to obtain the generated power and thrust coefficient C_t from the wind speed. A second turbine model gets the interpolation values for the DEL by taking as input the wind speed, TI and amount of yaw misalignment and by means of polynomial interpolation (varying according to the type of DELs) it outputs the corresponding DEL values. And finally a turbine model that calculates the absolute yaw angle from wind direction and yaw misalignment.

Next, a wake frame is needed that calculates wake coordinates according to the corresponding wind frame at the wake causing rotor and its yaw angle. The type of wake frame model used for this program is also based from the authors of the wake model (Bastankhah and Porté-Agel). It is a model for the calculation of the wake centreline and the wake coordinate, as it solves the problem that the wake centreline has after yawing, where it does not have a straight line any longer. The wake superposition models to combine the wake effects are the same ones that for the induction control program, i.e., a quadratic model for wind superposition, and a model for TI that calculates the nearest turbine's wake.

Now that the problem can start being constructed, the process from here is analogous to the one in Section 3.2.1. The difference being naturally the decision variables \mathbf{x} being used. In this case it is

the yawing of the turbines by certain number of degrees, with a range of allowed values (provided in the data from MoWiT) between $x_i^{lb} = -30^\circ$ and $x_i^{ub} = 30^\circ$. An example case of the optimization process using Flappy and PYMOO is found in Appendix A using curtailment as decision variables.

3.3 Testing and Evaluation of the Multi-Objective Wind Farm Optimizers

At this moment, the multi-objective wind farm optimizer programs for solving induction control and wake steering control problems using power and DEL as objectives have been created and are able to run calculation and perform optimization. But so far, it is not yet certain if the wind farm calculations are accurate and precise, or if they are free of errors in the code. The same can be said about the optimizer and the optimization process, it is not guaranteed that the solutions obtained are the optimal ones, or if there is a good enough convergence to the Pareto-optimal front and a good diversity according to the principles of MOO in Section 2.2.1

For these reasons, a number of different programs have been created in addition to the optimizers, in order to be able to test and evaluate the multi-objective optimizers. Firstly, the part of our original code which performs only the wind farm calculations (i.e. excluding the formation of the problem, the objectives and the solver and its optimization) was taken to be used in a different program where it is iteratively run in a loop. While running, three user chosen parameters will act as variables (each one inside of another's loop), easily creating thousands of cases when these variables consist of lists with a dozen elements each. The variables can be selected to be wind speed, TI, horizontal distance between turbines (the program tests wind farms consisting of only 2 or 3 turbines) and finally the amount of curtailment/yaw misalignment that a specific turbine in the farm has.

The results of this program are saved into a file, which can later be used by other testing programs created with interactive environments for evaluation. The first program developed, whose results can be seen in Appendix B, creates 3 dimensional animations, using a Python framework for creating interactive web applications called 'Dash'. Here, the three selected variables from the other program are the independent variables (the x-axis, y-axis and the animation frames from the figure), and the Power, C_t , and different types of DEL between other quantities act as the dependent variables (the z-axis and the color bar). This allows to compare many different scenarios to obtain results that are shown in the 'Results' segment (Section 4).

The other interactive environment created also uses this data file, but in this case the data is used to compare the decision/design space \mathbb{R}^n of the chosen independent variables with the objective space \mathbb{R}^M of the also user chosen dependent variables, in a similar way as it is shown in Figure 2.6. It uses a different Python framework for interactive web applications called 'Streamlit'. Due to the limited amount of visual dimensions, up to 3 decision variable cases comparing up to 3 objectives can be shown if selected by the user. This software will be vital at ensuring a correct behaviour of the optimizer, especially in achieving a desired convergence and diversity towards the Pareto Front, and an example using this software can be seen in Figure 4.1

4 Results

The goal of developing and implementing the code for a multi-objective wind farm optimizer with the objectives of yield maximization and load minimization for induction control and wake steering control problems has so far been achieved. But thus far, it is unknown if the running behaviour of this optimizer is precise and flawless. Meaning that, there is no yet knowledge if the wind farm calculations from Flappy are truly correct and accurate, it is also so far unsure if the optimizer is arriving to the corresponding solutions, and furthermore, in the case it arrives to the solutions, it is not known if the optimizer observes the principles of MOO (see section 2.2.1) and converges to and diversifies these solutions. In order to assure its correct behaviour, the multi-objective optimizer must be tested and evaluated rigorously by means of different tests and utilizing the programs developed in Section 3.3. These results will be explained below in this section.

Once the correct operation of the optimizer has been demonstrated, it is finally possible to inspect and analyze the Pareto-optimal fronts created for curtailment and yaw misalignment cases and observe the influence it exerts in the power and loads of wind turbines in the results. Lastly, different optimization scenarios with multiple Pareto Frontier result cases will be shown for later analysis in Chapter 5.

4.1 Testing the Multi-Objective Optimizers Performance

In order to guarantee that the multi-objective optimizers calculations are reliable, these ones must undergo different testing procedures for its assurance.

4.1.1 Testing Wind Farm Flappy Calculations and Analyzing Behaviour Patterns

An optimization process requires a set of decision variables (the curtailment or yaw misalignment) that improve an objective or objectives (power and DEL) at each evaluation. Naturally, this whole process is pointless if the objective value calculated from the decision variables is incorrect in the first place. The calculations of this objective values consist of wind farm and wake modelling calculations performed by the Flappy library.

The code scripts created in Section 3.2 use these Flappy calculations. In Section 3.3 a testing environment is created by isolating the code from the script that performs these Flappy calculations, and running this part of the code thousands of times in varying scenarios with changing wind speed, TI and distance between turbines (non-optimization calculations) to observe the plausibility of the results. In Table 4.1 the general relationships obtained from the test results are shown for uncurtailed, unyawed turbines. Some of these results can be closer examined in Figure B1.

Table 4.1 Test result correlations from wind farm calculations of 1287 scenarios using testing script from Section 3.3 with two turbines. It shows the resulting increase (\uparrow) or decrease (\downarrow) of Power, C_t and Flapwise DEL when the testing cases are increased in wind speed, TI and distance between turbines.

	Increasing wind speed	Increasing TI	Increasing distance between turbines
Power	\uparrow	\uparrow	\uparrow
C_t	\downarrow	\downarrow	\downarrow
Flapwise DEL	\uparrow	\uparrow	\uparrow^*

*Special case with low wind speed and low TI, where DEL decrease (\downarrow) with increasing distance between turbines, if the distance is bigger than ~500 meters

The table above shows the effect that wind speed, TI and varying horizontal distance have on a single turbine affected by the wake of another (i.e., a simple 2 turbine wind farm scenario). The table shows that the power of the wake affected turbine increases with increasing wind speed, TI and distance, and correspondingly C_t decreases with the increase of all three. The DEL also increases with the increase of wind speed, TI and distance, except for the special case with low TI and wind speed at distances bigger than ~500 meters. This means that the election of power production maximization and DEL minimization for MOO objective function values is a wise one, as they are in most cases in direct conflict with each other, generating Pareto Frontiers with multiple valid trade-offs.

Now that the general case has been shown in Table 4.1, the cases where the active wake control strategies of curtailment and wake steering are applied to simple cases of 2 or 3 turbine wind farm scenarios can be analyzed. In Table 4.2 these relationships are shown.

Table 4.2 Test correlations from wind farm calculations of more than 5000 scenarios in wind farm cases with two or three turbines in a row. It shows the resulting increase (\uparrow) or decrease (\downarrow) of Power, C_t and Flapwise DEL when the testing cases are increasingly curtailed or yawed.

	Increasing curtailment		Increasing yaw misalignment	
	2 turbines	3 turbines	2 turbines	3 turbines
Power	\uparrow	T1: \uparrow T2: \uparrow	\uparrow	T1: \uparrow T2: \downarrow
C_t	\downarrow	T1: \downarrow T2: \downarrow	\downarrow	T1: \downarrow or =** T2: \downarrow
Flapwise DEL	low TI: \downarrow **** high TI: \uparrow	T1: = or \uparrow *** T2: \downarrow ***	\uparrow	T1: \uparrow (slightly) T2: clockwise: \downarrow * anticlockwise: \uparrow *

*At a clockwise and anticlockwise wake steering approaching -30°/30° in turbine T2, the behaviour begins to reverse, i.e., for clockwise wake steering at T2, increasing yawing increases (\uparrow) DEL, and for anticlockwise wake steering it decreases (\downarrow) DEL. Also, at fairly high wind speeds (over 17m/s), the complete opposite to the entire behaviour takes place.

**At high wind speeds the increase in yaw misalignment doesn't change C_t or increases it (\uparrow) very slightly.

***Special case where T1 has a low initial curtailment and T2 has a high initial curtailment, the effect is opposite as normal, where increasing the curtailment at this point would decrease (\downarrow) DEL for T1, and increase DEL (\uparrow) for T2.

****At higher wind speeds the DEL can increase with increasing curtailment.

In the table above, the effect that increasing curtailment and wake steering have on power production, C_t and the Flapwise blade root bending moment is shown for the scenario containing two and three turbines in a row. For the two turbine cases, as in Table 4.1 there is a single uncurtailed/unyawed turbine from which the increase or decrease of power, C_t and DEL is shown in the table, and this turbine is affected by the wake of another. But for the three turbine in a row case, also the third uncurtailed/unyawed turbine is accounted for only in the results, but in this case T2 is located just in front of it at its left position, and T1 is located at the left of T2 (i.e., T1 receives undisturbed wind by wake).

For the case of increasing curtailment in Table 4.2, the power production of the uncurtailed turbine seems to increase always with increasing curtailment from the turbines in front, suggesting that curtailed turbine wake is positive for power production increase for the turbines affected. Correspondingly, for the C_t the opposite occurs and it decreases with increasing curtailment. In the

case of the flapwise DEL, a variety of cases presents. For the two turbine case, the DEL of the uncurtailed turbine decreases if presented with scenarios of low TI and wind speeds, but increases in the rest of the cases, which indicates potential cases where a simultaneous increase of both objectives (power maximization and loads minimization) can occur which could be beneficial. This combination of possibilities is further confirmed in the 3 turbine case, where as a general rule the uncurtailed wind turbine increases its flapwise DEL with increasing curtailment of T1, while it decreases with increasing curtailment of T2, and a special case where the opposite effect happens (DEL decreases with increasing curtailment of T1, while increases with increasing curtailment of T2) when the curtailment of T1 is initially low (almost uncurtailed), and the curtailment of T2 is initially high.

Now, the behaviour when altering the mechanism of yaw misalignment is different than with induction control. In the first place, the power production of the unyawed turbine does not seem to increase always with increasing curtailment from the turbines in front, as while this might be true for the 2 turbine scenario, it is not true for the 3 turbine scenario, where it decreases its power with increasing curtailment of T2. This indicates already a lesser consistency compared with the more dependent and stable relation of induction control's objective variables. The C_t is similarly not as consistent as with induction control methods, as while it does consistently decrease with increasing yaw misalignment, at high wind speeds the increase wake steering doesn't change C_t or increases it very slightly.

Regarding the effect that the wake steering has on the Flapwise DEL, it is perceptible this lack of consistency from using a mixture of yawed turbines which create different yawed patterns that are more 'chaotic' and unpredictable than the uniform ones from induction control methods (see Figure 2.5). While for the simple case of two turbines, where no combination of yawed wakes happen, there is a simple and consistent increase in DEL for the unyawed turbine receiving the wake effects when the yaw misalignment of the first one increases. For the case of 3 turbines, where only a mixture of 2 yawed turbine wakes occurs, the results on the unyawed third turbine become much more complicated. Especially the yaw misalignment of the turbine T2 has lots of different scenarios, as the unyawed turbine, when increasing turbine T1, it just slightly increases the Flapwise DEL.

Surprisingly, not only the amount of wake steering is important from turbine T2 in the 3 turbine case, but also the directionality of the yawing that is happening. For the case where the yaw misalignment occurs clockwise, the flapwise DEL of the unyawed turbine seems to be reduced with the increasing misalignment. And unexpectedly, the opposite happens when the wake steering occurs anticlockwise, increasing the DEL with increasing misalignment from the turbines in front. But this is not it, as when the clockwise and anticlockwise wake steering of turbine T2 approaches -30° and 30° the behaviour begins to reverse, meaning that for clockwise wake steering at T2, increasing yaw misalignment increases DEL, and for anticlockwise wake steering it decreases DEL. And on top of all this behaviour for turbine T2, at fairly high wind speeds, over $17m/s$ approximately, the absolute opposite to the entire behaviour yet explained occurs. This means that initially increasing clockwise yaw misalignment increases DEL, and increasing anticlockwise reduces DEL in the unyawed turbine. And T2 approaching even sooner yawed angles of around 20° and 20° this behaviour reverses again, with increasing clockwise wake steering of T2 decreasing flapwise DEL and increasing anticlockwise wake steering increasing flapwise DEL. Some of the interpreted behaviours from Table 4.2 can be visualized in Appendix B.

The application of wake steering in a turbine results in the following consequences for the downstream affected turbines, according to the developers of the AWC strategy, i.e. ECN [28]:

- An increase of wind speed, which as a general rule increase the DEL.
- A decrease of TI, which results in a reduction of DEL.

These results have been confirmed by the results from the testing environments and can be seen in Figure B2.

4.1.2 Testing Optimizers Convergence Towards Solutions

When running the optimizer script, it is yet unknown if the optimizer being used is genuinely finding the best solutions possible or not. The next testing phase consists in assuring that the optimization process indeed reaches the solutions that are optimal, i.e., that it improves the objectives and that it actually tries to reach the Pareto optimal solutions.

In order to do so, the second interactive environment developed and explained in Section 3.3 will be used. As explained in the mentioned section, this script will use hundreds or thousands of wind farm calculated cases, and it will compare the decision space \mathbb{R}^n of the chosen independent variables (for this case curtailment or yaw misalignment of the turbines) with the objective space \mathbb{R}^M of the user selected dependent variables, being the objectives of interest the power production (Eq. 3.1) and flapwise DEL (Eq. 3.2) values from the entire wind farm, in a similar fashion as in the comparative case in Figure 2.6.

As in Section 4.1.1 this comparative analysis is made for cases with a maximum of 3 turbines. This is for various reasons, firstly, there is a physical boundary of using a maximum of 3 decision variables in the selected problems or 3 objectives in the objective space comparison, as each of these is a visual dimension. Lastly, as with the past testing section, simple cases which are relatively easy to analyze and obtain useful information are of our interest, as increasing decision variables further massively complicates any conclusion or analysis since the possible behaviours of the mixing wakes are boundless. The cases used for verification are the same calculations cases from last section with a 3 turbines, i.e., a wind farm with 3 turbines, where the rightmost turbine is uncurtailed/unyawed, and each of the other two are changed in each fixed scenario. Later, in the optimization comparative scenarios, the same exact case will have to be made, meaning that the two leftmost turbines are in this case the decision variable (making it a 2-dimensional decision space problem), and their values of curtailment/yaw misalignment will be continuously improved until their objectives reach the Pareto Front, at least in theory.

In the tests created, one of these shown in Figure 4.1, it can be seen in the left (Figure 4.1a) the decision space \mathbb{R}^n where the Pareto-optimal set of solutions from the fixed cases, where no optimization was performed, appear in red, and the non-dominated solutions from the Pareto-optimal set appears also for the optimized solutions from our optimizer in light blue. For the optimization 200 generations with 120 population (60 of these offspring) was used, where curtailment is used as the decision variable, where in both cases there is 12m/s wind speed, a TI of 5% and a horizontal distance of 360m (it is not so important the quantities used, but the fact that both cases (optimized and non-optimized) used the same parameters). The Pareto Front solutions of the corresponding Pareto-optimal sets can be seen in Figure 4.1b in the corresponding objective space \mathbb{R}^M , for the fixed case in red, and the optimized case in blue.

It can be observed from the Figures already, how in both spaces the optimized solutions (blue) and the non-dominated solutions from the fixed case (red) reside in the same neighbourhood so to say, indicating the optimizer indeed performs in a correct manner towards the optimum solutions. Certainly, the simplicity of this cases is not useful for real life scenarios, but it is an indication about the optimizer converging towards the Pareto Frontier.

A different way to test the convergence of solutions towards the Pareto Front, is to observe to totality of its solutions since its beginning, i.e., showing each population from each generation there is. For the same optimized case in Figure 4.1, the totality of its solutions is shown in Figure 4.2b.

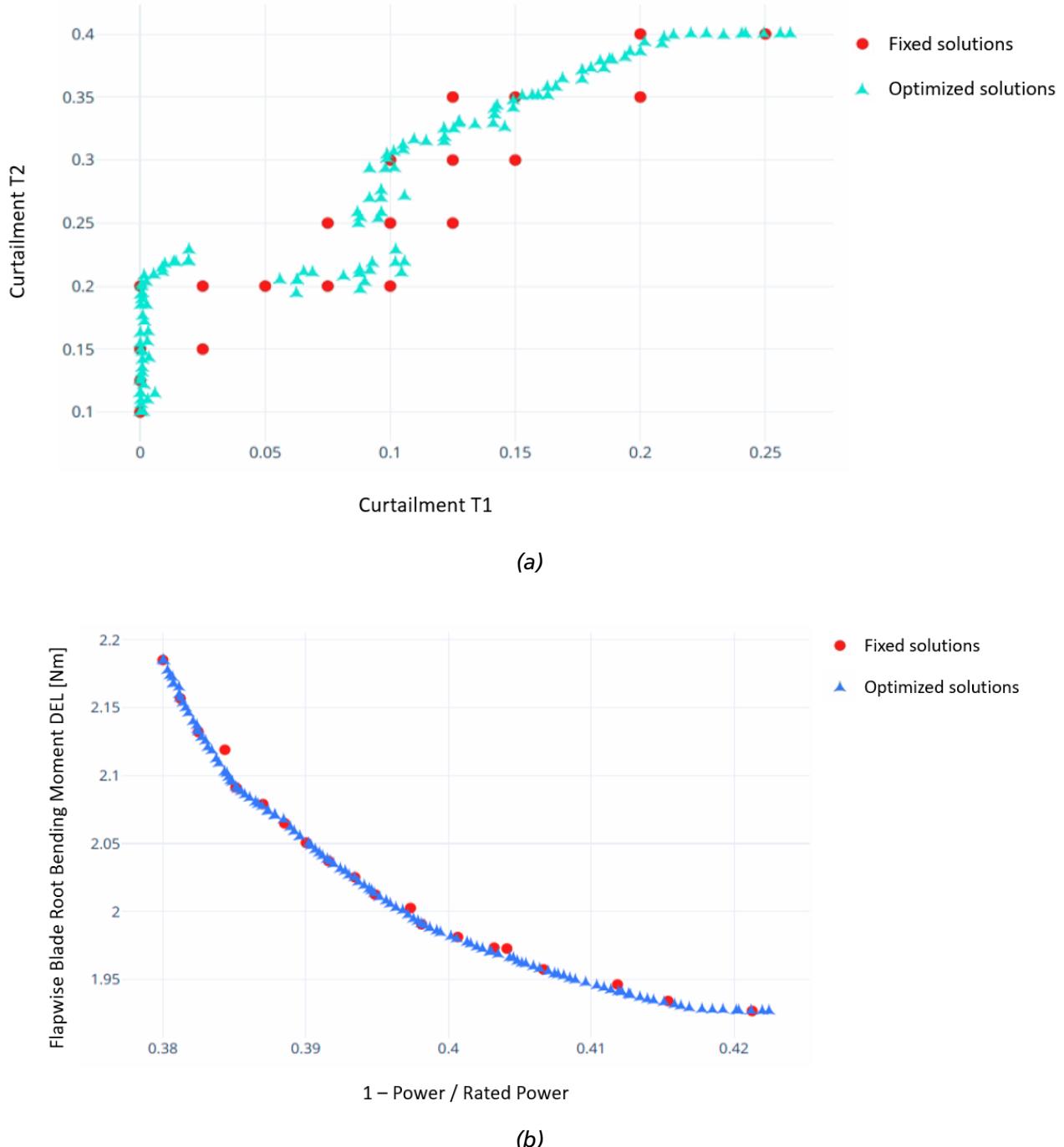


Figure 4.1 Results showing the a) decision space \mathbb{R}^n and b) objective space \mathbb{R}^M of the non-dominated solutions from fixed cases, where no optimization was performed, and the non-dominated solutions from the optimizer program developed. The optimization solutions have 200 generations with 120 population (60 offspring), using curtailment of the turbines as the decision variables, 12m/s wind speed, TI of 5% and a horizontal distance of 360m.

The red points consist of the Pareto front, and the blue empty point are the multiple population from all previous generations that lie in the objective space. Since it is quite a simple case with 2 decision variables and 3 turbines, most of the past populations lie already very near to the Pareto front.

One can observe this ways how much improvement searching the objective space there has been since the start of the optimization process, and determine in some occasions if there is a need to continue evaluating for a further number of generations, or if otherwise the Pareto Frontier solutions are rapidly approached. This guides us in the performance of the optimization procedure (see Section 2.2.4), where one can observe this need for further evaluations (i.e., amount of population multiplied by the number of generations) so that the optimizer works efficiently for a specific case. The hypervolume metric for the case in Figure 4.1 is shown in Figure 4.2a. It can be seen how as the evaluations increase in number, the hypervolume (in this case the area created between the pareto front points and the anti-optimal point which for our case has the value of worse than the nadir point) keeps improving, usually at slower and slower rate as the evaluations increase and the Pareto optimal solutions are being approached.

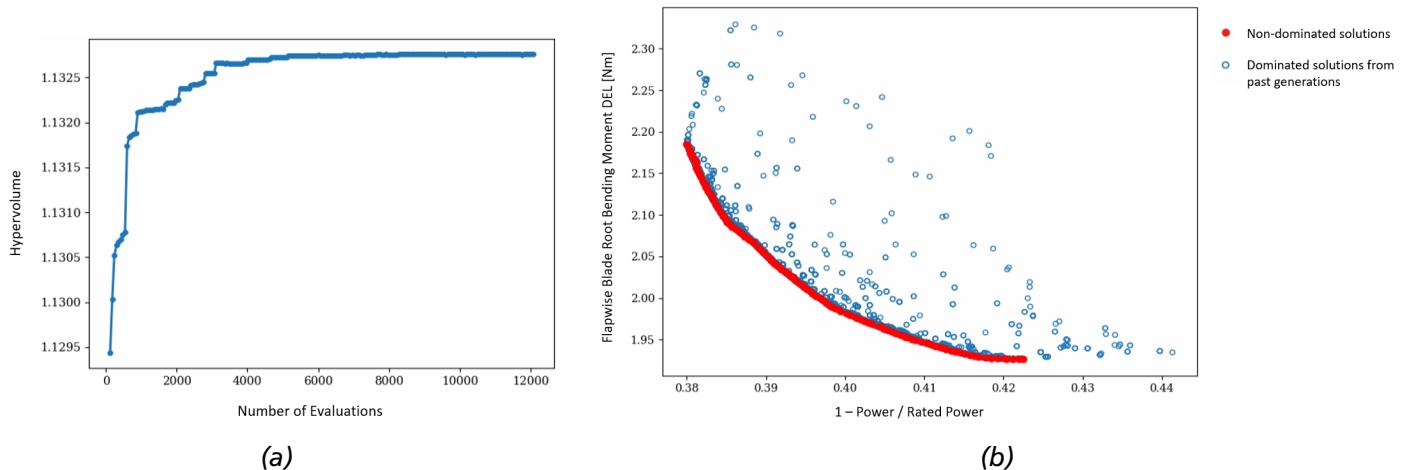


Figure 4.2 a) The hypervolume performance metric from Figure 4.1, and b) the objective space \mathbb{R}^M from the same case showing the Pareto Frontier (red) and all preceding solutions (empty blue)

4.1.3 Testing Optimizers Divergence and Extent

An Important test to check that the MOO process is working at its optimal state and that the solutions diverge and extent as much as possible in the objective space \mathbb{R}^M , is to test it against the single-objective optimization process of each of its objectives (K.Deb [34]), i.e., to create a single-objective optimization case for power maximization, and a single case for loads minimization. Once this is done, one can compare the optimal single solution of each of the two single-objective optimizers with the entire Pareto optimal solutions created by the multi-objective optimizer.

These single-objective optimizers were developed for both decision variables (curtailment and yaw misalignment) and their results examined. The resulting Pareto Frontiers seem to indicate that the MOO program developed indeed reaches solutions which are very close to the optimal solutions from the single-objective optimizers and are satisfactorily scattered around.

To illustrate the point take Figure 4.3. Here, a Pareto front is created using the MOO program (green points), and also the optimal solutions from the single-objective optimizers for power maximization

and loads minimization (yellow diamonds) can be seen, for a 12 turbine wind farm using wake steering control. It can be noticed, that although the solutions from the MOO procedure do not reach the very near neighbourhood of the single cases, the trade-off benefit in doing so would be almost minimum as there is a great cost in the other objective to be sacrificed for a almost minimal improvement in each of the single cases. Also naturally, these cases were run using similar evaluations (around 250 population and 100 generations), and by allowing more evaluations to the multi-objective optimizer it does improve the diversification, but it is at a high computational cost for small improvements in cases like Figure 4.3, and even so the exact same results has not been achieved in our tests so far.

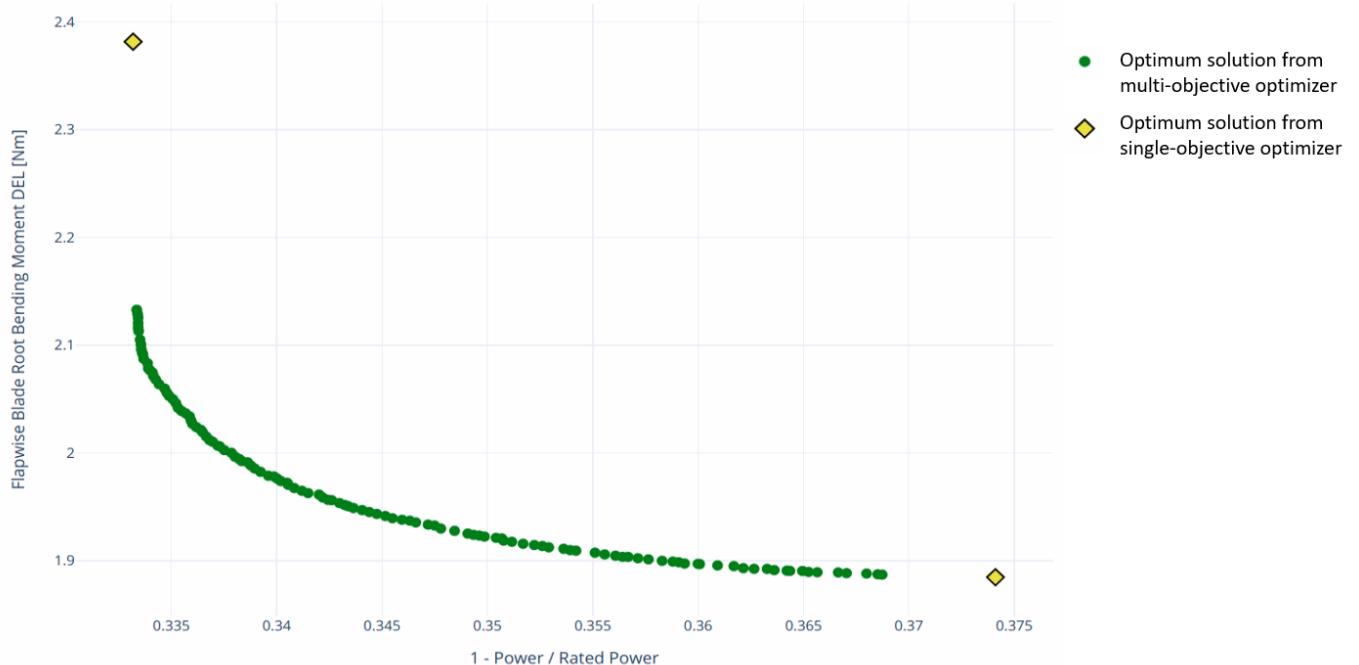


Figure 4.3 Comparison of the Pareto front created using the MOO program (green points) with the single-objective optimizers for power maximization and loads minimization (yellow diamonds), in 12 turbine wind farm using yaw misalignment of the turbines as the decision variables, 12m/s wind speed, TI of 5% and a horizontal distance of 600m for all cases.

4.2 Analyzing Pareto Fronts and its Effect on the Wind Farm

At this point, the aim of this project has been accomplished in creating a fully functional multi-objective wind farm optimization program for AWC cases, with yield maximization and load minimization objectives when performing induction control or wake steering control. It is finally possible to analyze the Pareto-optimal fronts that the optimizer creates for curtailment and yaw misalignment cases and therefore observe the influence it exerts in the power and loads of wind turbines in the results.

An example can be observed in Figure 4.4. In the image a comparison is shown between the Pareto Frontier solutions using induction control methods in black, and a front of solutions using wake steering control methods in red. Also, the yellow diamond point indicates the solution without using any curtailment or yaw misalignment. Each of these individual solutions from the Pareto fronts being a particular wind farm of 12 turbines in this case, with applied AWC techniques in each as shown in Figures 4.5 and 4.6. First of all, it can already be seen that just by applying a

small amount of curtailment or yaw misalignment, there is an initial substantial amount of benefit in load reduction, especially for wake steering control where the yellow solution is even dominated by the rest of the solutions.

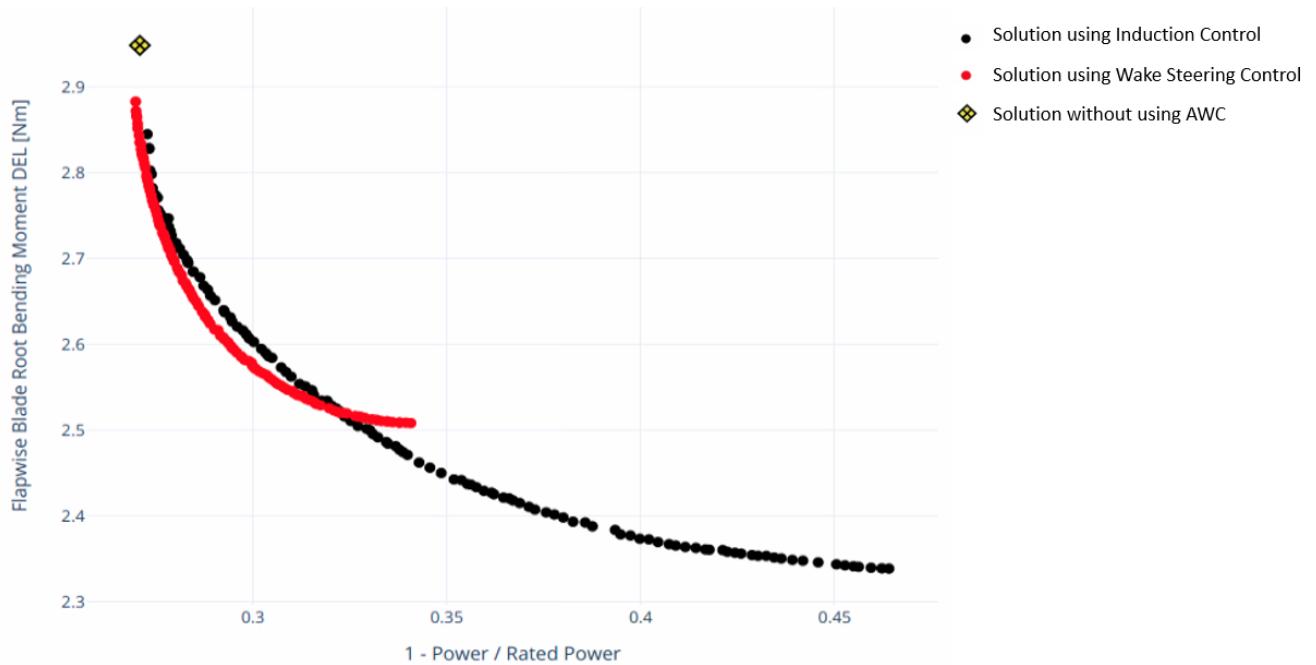


Figure 4.4 Comparison of the Pareto Frontiers created ,using the MOO program, when induction control methods are being used (black points) or when wake steering control methods are used (red points) in a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.

The trade-offs offered by the yaw wake steering control for this particular conditions seems more beneficial, if one does not intends to reduce too much power production. The percentage of trade-off benefit offered by the Pareto Front consists of a DEL 13% - Power 7%, meaning that going from the best power maximization scenario (top-left) to the best loads minimization scenario (bottom-right), one will reduce DEL by 13% by a cost of reducing power production by 7% also, which if interested in minimizing loads may in beginning sound as a good deal.

But in reality, this trade-off relationship is not linear, actually the convexity of the shape makes choosing intermediate points of the Pareto Front more beneficial. At this point is where the role of the Decision Maker comes into place (see Section 2.2.4) at choosing the optimal solution for his indicated case, always being able to use some of the "A-Priori" techniques mentioned if necessary to guide the process of selection before the optimization has begun.

The trade-offs offered by the induction control methods in contrast offer more beneficial solutions if there is a stronger interest in reducing the DEL, offering a percentage of trade-off benefit of DEL 17% - Power 19%, which again when taking into account the wake steering case and non-linearity of the solutions, the Decision Maker must decide if it is in its benefit for the particular wind farms in case.

Each individual solution from the Pareto fronts represents a wind farm with a value for total power production and flapwise DEL. Different solutions in the PF, indicate different combinations of curtailment or yaw misalignment for each individual turbine in the wind farm, as each affect the wake in a different manner.

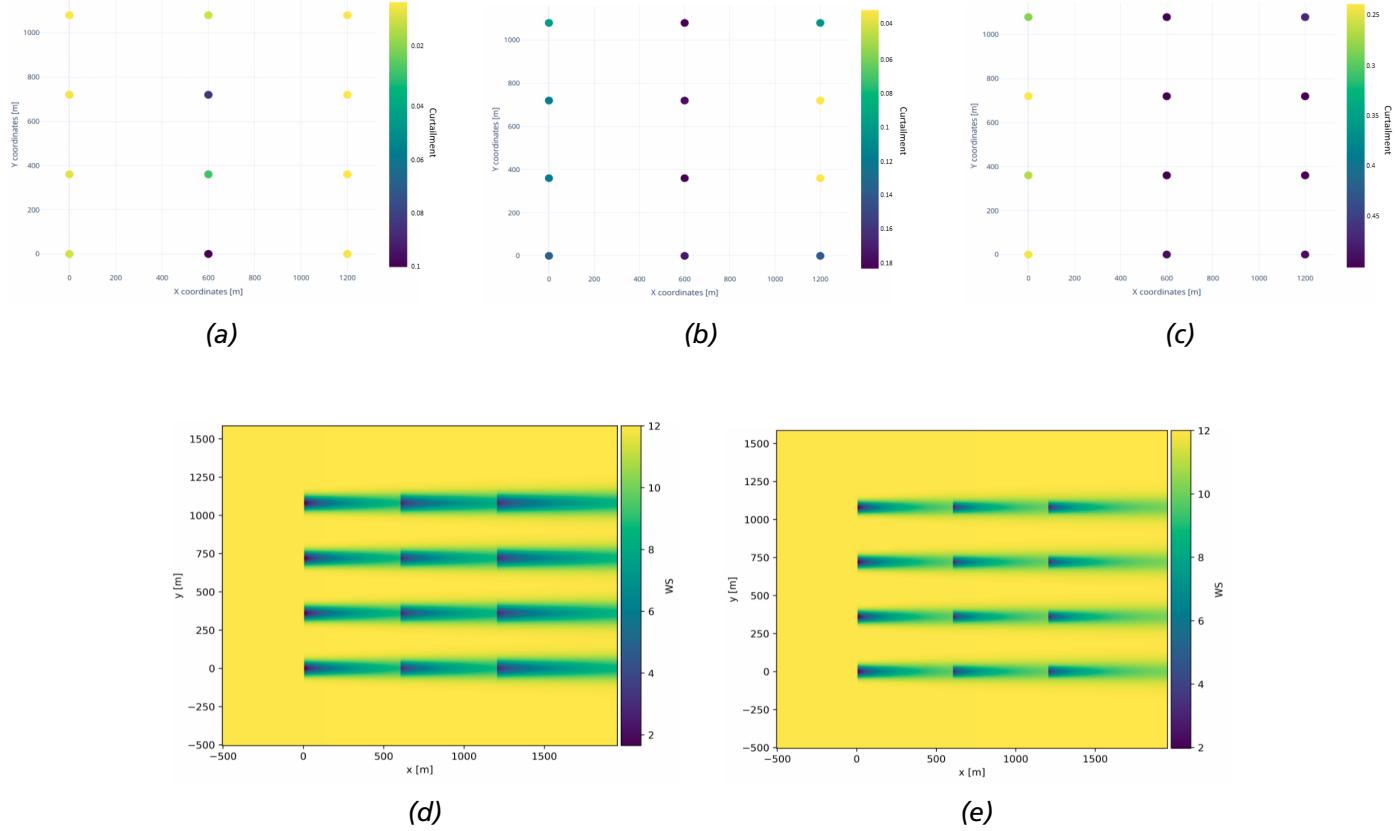


Figure 4.5 Figure showing individual wind farm solutions from the Pareto front solutions of induction control in Figure 4.4 for the cases of a) the trade-off solution offering maximum power production and its wake (d), c) the trade-off solution offering minimum DEL and its wake (e), and b) a case in between offering a balance between both objectives. For a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.

For the case of induction control methods, the trade-off solutions offering maximum power production as seen in Figure 4.5a have very little amount of curtailment in general, which is coherent with the fact that they are more close in the objective space to the uncurtailed case. As observed in the layout figure, turbines placed in the middle column mainly suffered some noticeable amount of curtailment, and this was enough to reduce more drastically the DEL in the wind farm. But these yield maximization results are particular to these conditions, in this case, using an initial wind speed of 12m/s and TI of 5%.

When a factor like changing the initial wind speed changes, the yield maximization may change entirely, as it can be seen in Figure 4.7, where the 9m/s scenario has a higher curtailment in the first two columns, while the 15m/s scenario has much lower curtailment in general in a non-defined way. For the solutions of the Pareto front offering minimum flapwise DEL, the curtailment as can be seen in 4.5c is much greater (up to around 35% reduction of power production in some turbines), and the reduction of this DEL is much greater in the last two columns. Finally, an intermediate case offering better trade-offs for both can be found in Figure 4.5b. The shape and characteristics of the wake can also be observed for the solution with maximum power production (Figure 4.5d) and minimum DEL (Figure 4.5e). It can clearly be seen how the wakes in the first one are much stronger (higher wind speed reduction) than in the second case, where the curtailment effect can be appreciated better.

For the case of wake steering control, there is this same amount of diversity in solutions if not more. In Figure 4.6 to start the solutions offering better power production values have lower yaw misalignment angles (again, making sense as it is near the unyawed case in the objective space), especially low in the last column of turbines, while for the solutions that offer low DEL the yawing angles are much higher, in the permitted limits of near 30° and with higher misalignment in the first column in this case. As commented before, the variations when changing some initial conditions like wind or TI may change the distribution, space and amount of loads in each turbine in a considerable manner, although generally the top-left side of the Pareto front will provide low yaw misalignment values in the turbines and viceversa in the case of the bottom-right side of the PF. It is interesting to observe the shapes of the wakes (Figures 4.6d and 4.6e) especially for the wake steering control, as one can observe how the flow from the turbines is redirected, and the effect it has in the next turbines. More on this will be commented in the discussion (Chapter 5).

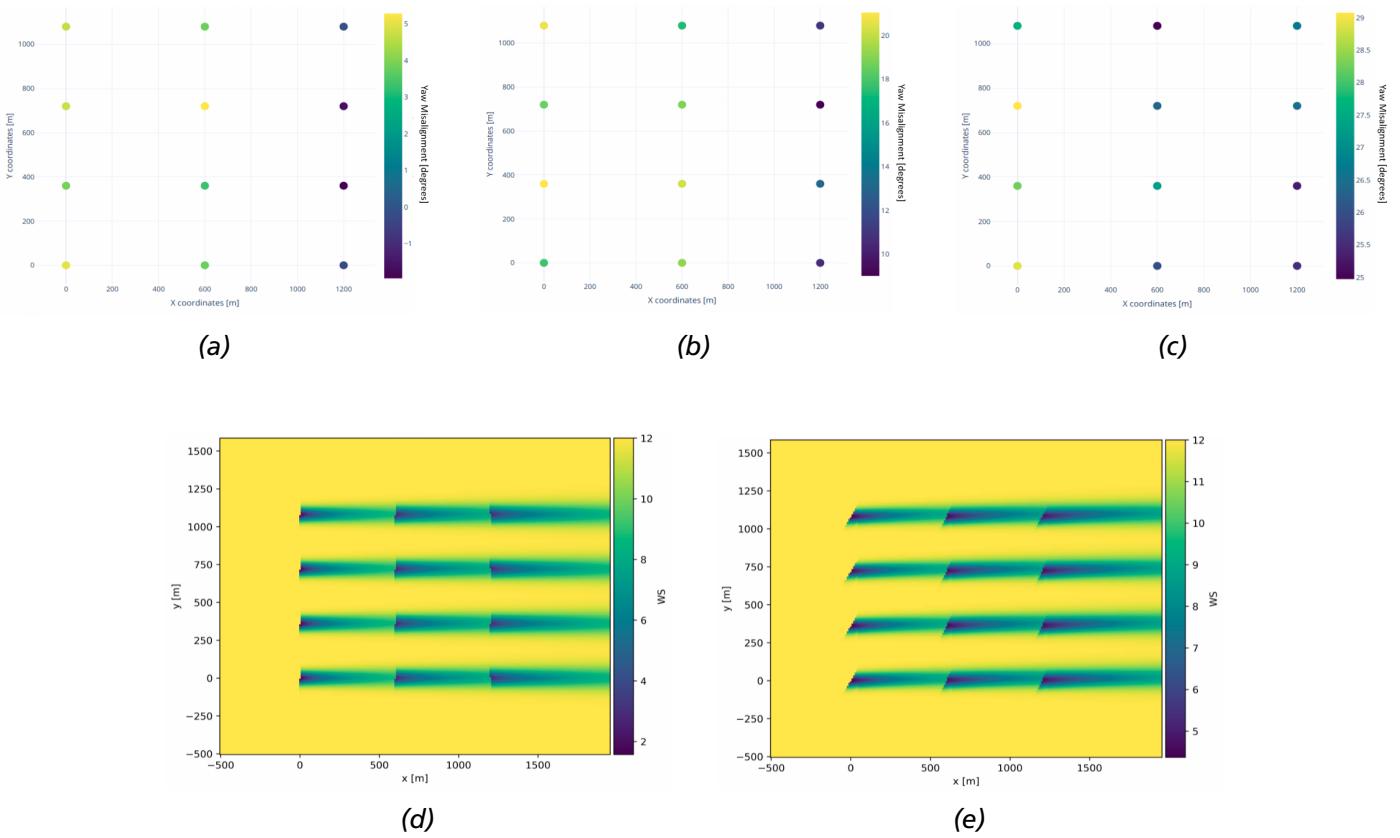


Figure 4.6 Figure showing individual wind farm solutions from the Pareto front solutions of wake steering control in Figure 4.4 for the cases of a) the trade-off solution offering maximum power production and its wake (d), c) the trade-off solution offering minimum DEL and its wake (e), and b) a case in between offering a balance between both objectives. For a 12 turbine wind farm using an initial wind speed of 12m/s, TI of 10% and a horizontal distance of 600m for all cases.

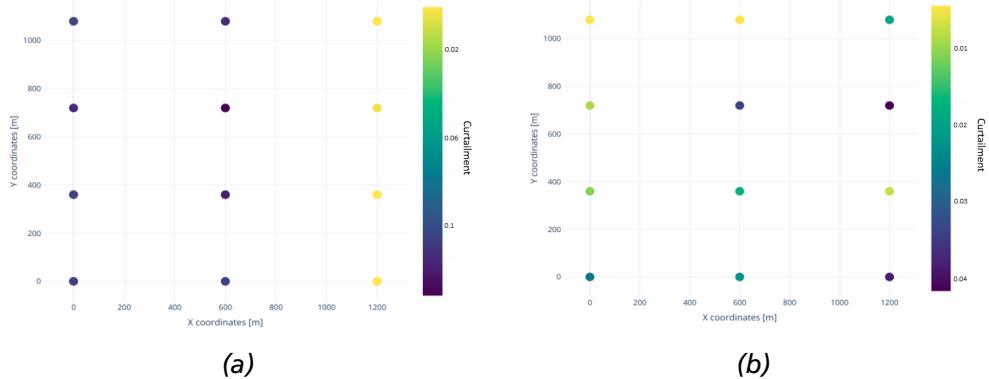


Figure 4.7 Figure showing solutions from a Pareto Front that offer maximum power production (top-left) using the same properties from Figure 4.4 for the wind farm but changing only the initial wind speed to a) 9m/s and b) 15m/s

4.3 Comparing Multiple Pareto Front Scenario

As mentioned in the last section, changes in wind farm conditions like initial wind speed, TI, turbines location relative to each other and within the farm, within many others, may drastically change the Pareto fronts shape and location, which may then change the preference for the type of solutions inside the Pareto front that the Decision Maker would be interested in.

Using the MOO program developed in this paper, a variety of different scenarios have been computed following the methodology and properties explained in Section 3. The termination criteria for the optimization process of these cases had a high tolerance, around the 1000 generations in most cases using 120 population (60 offsprings). The final Pareto optimal fronts for each these scenarios are shown in Figures 4.8, 4.9 and 4.10.

The results are shown in a similar manner to the concept in Figure 4.4, where selected some specific wind farm conditions, the PFs for the solutions of induction control and wake steering control, together with the single solution where no AWC technique was applied, are shown in the figure. The difference in these cases is that now there is no single specific wind farm conditions, but multiple at the same time, therefore creating a different number of PFs of different geometries and positions. Each of these elements (induction control solutions, wake steering control solutions and the solution without utilizing AWC) are grouped in the same color code, having also curtailed solutions as triangles, wake steering control solutions as circles and the individual solution unyawed and uncurtailed as diamond shaped.

Each figure corresponds to a different fixed initial wind speed condition (9m/s, 12m/s and 15m/s), and inside each figure there are four different group cases, for two different kind of initial TI conditions (5% and 10%) and for two different horizontal distances between the turbines turbines in each column (360m and 600m).

Some of the main characteristic features, that will be later discussed in the Chapter 5, is that the coloured grouping sets of PFs, when increased their TI, this increases the DEL and slightly the power production on the turbines. When increased their wind speed, this also increases more significantly the power production, but also increasing the loads. The varying horizontal distance and the different types of AWC used are more case dependent and will be considered in the next chapter.

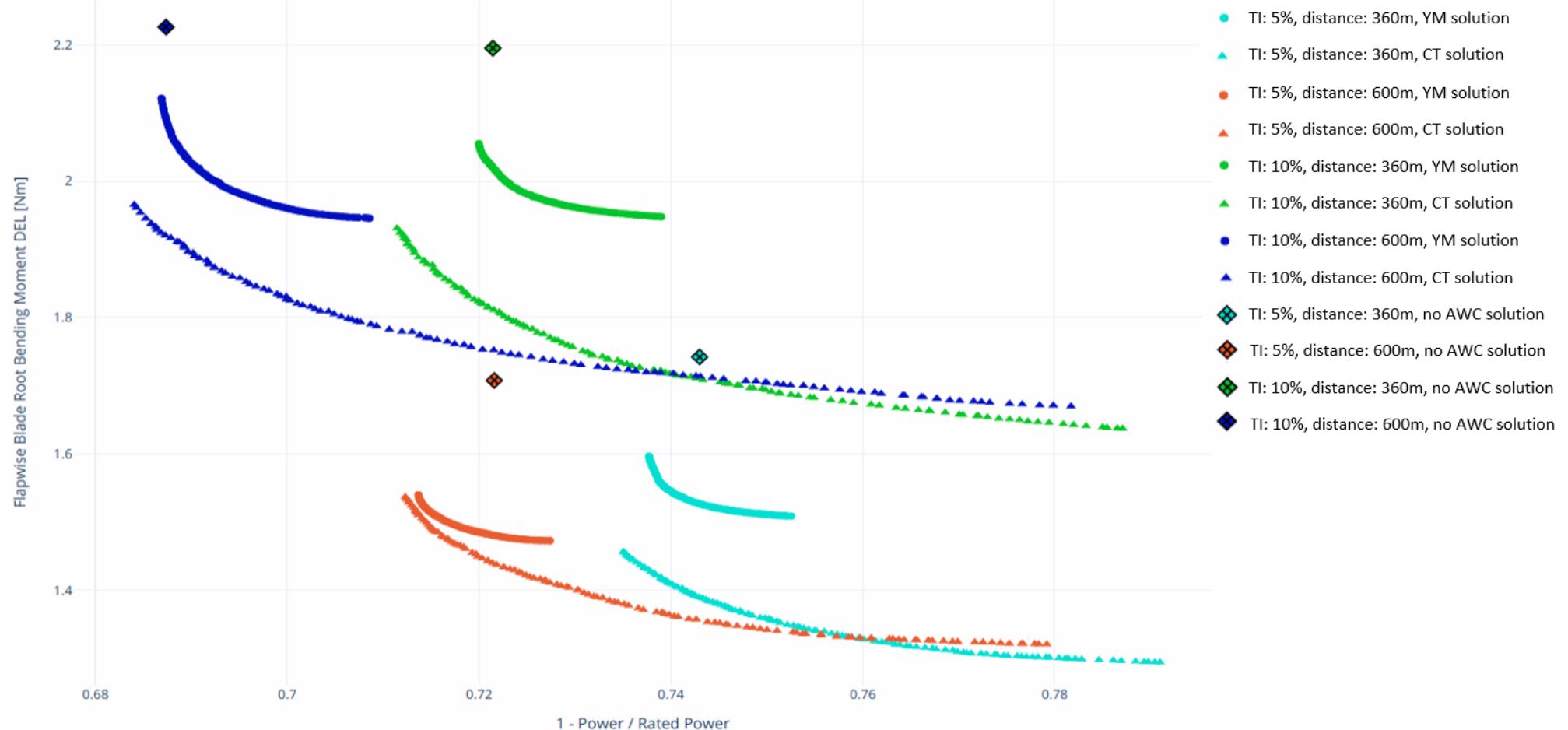


Figure 4.8 Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 9m/s. Solutions where no AWC method was applied are also shown (diamond).

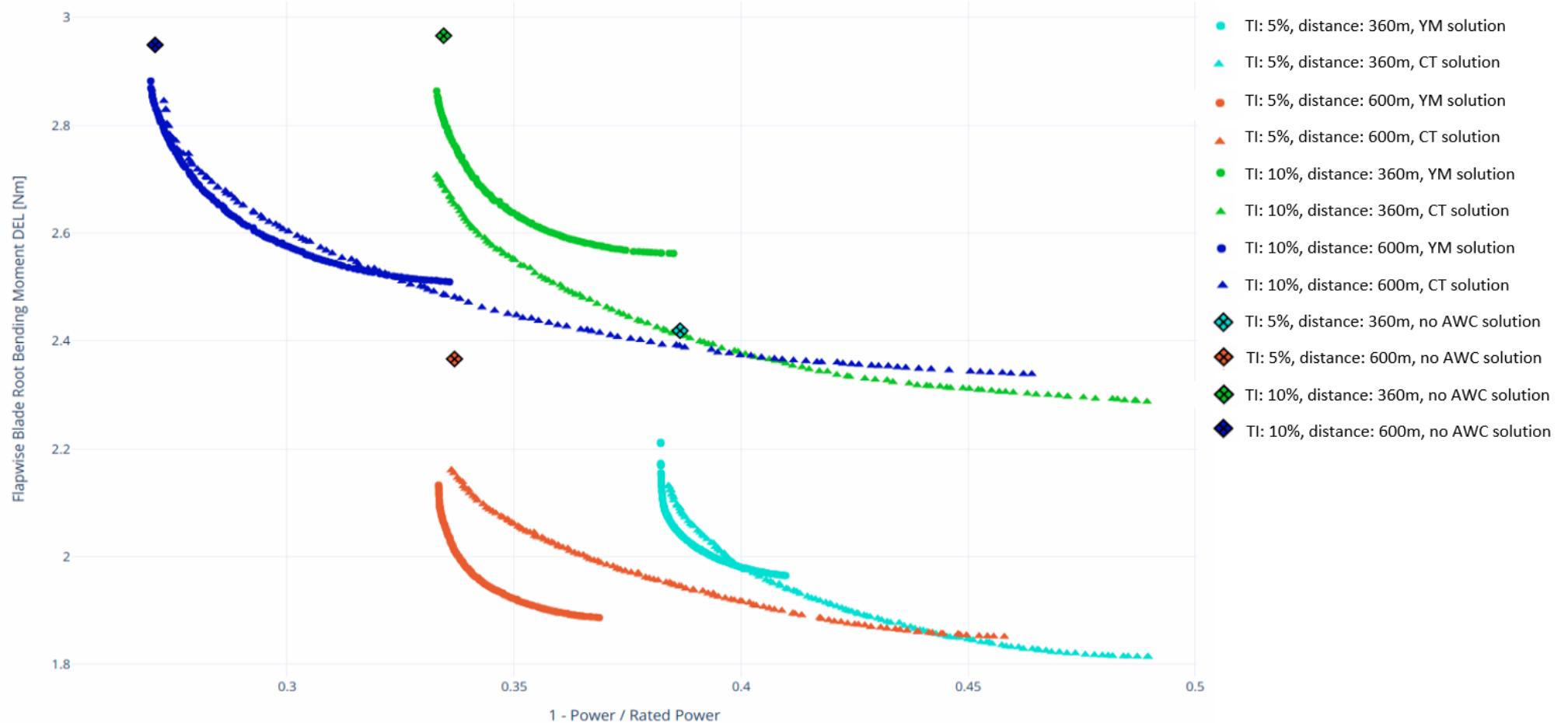


Figure 4.9 Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 12m/s. Solutions where no AWC method was applied are also shown (diamond).

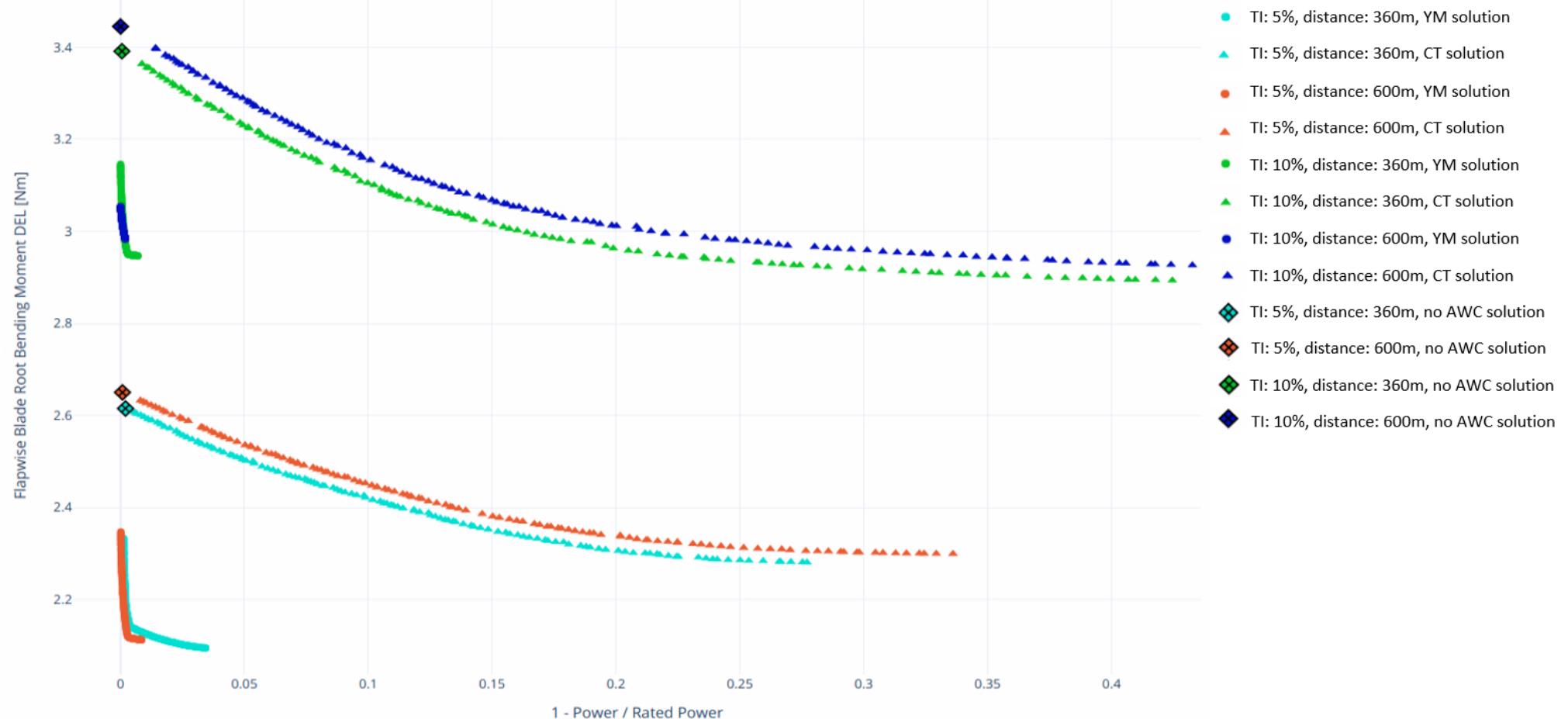


Figure 4.10 Multiple Pareto fronts, created using the MOO program developed, employing techniques of induction control (triangles), where turbines are individually curtailed (CT), and wake steering (circles) where turbines are yawed (YM). Four different group cases, with varying initial TI conditions (5% and 10%) and horizontal distance between the turbines (360m and 600m) for a fixed initial wind speed of 15m/s. Solutions where no AWC method was applied are also shown (diamond).

5 Discussion and conclusions

Active wake control and wind farm optimization using multiple objectives are powerful tools in the wind energy industry. The results of this paper have shown the birth of a multi-objective optimizer program developed for AWC techniques, i.e., using curtailment or yaw misalignment of the turbines to achieve the goal of yield maximization and load minimization. These results have shown that the optimizer developed, not only correctly solves the wake calculations of a wind farm, but it also optimizes these results in a front of solutions, and that these in practice arrive or approximate the optimal Pareto Frontier of best solutions, and also that the solutions are diversified enough to achieve the best trade-offs (principles of convergence and diversity).

Also from the results, it has been shown what is the behaviour and characteristics of Pareto fronts and their significance, together with some results from each individual wind farm solution and how is each turbine affected differently in each solution scenario.

Lastly, multiple Pareto front solutions generated with the optimizer were compared in Figures 4.8, 4.9, and 4.10 and further explained. From the results in these figures there are many valuable interpretations for discussion. Firstly, perhaps the least interesting, is the effect of horizontal distance between turbines. When this one is increased (360m to 600m in the mentioned figures) it seems to have a different effect in the wake steering case compared with the induction control case. Taking into account Figures 4.8 for 9m/s and 4.9 for 12m/s only (as in high wind speeds the power curve gives full power even when in wake and this changes the behaviour) for the cases where induction control was applied, the PFs from wind farms with increased distance between turbines will give better trade-off's when you are interested in gaining power more than decreasing loads. On the other hand, when searching for trade-off's where you are more interested in decreasing loads than gaining power, a wind farm with shorter distance between turbines will give you better solutions.

For the cases where wake steering methods are applied there is a different behaviour, where wind farms with higher distance between its turbines will always achieve better solutions, as it is the same case when there is no AWC method applied. The most interesting relationship from modifying the horizontal distance is that, as the distance between turbines increases, it seems to benefit the wake steering control scenarios more than it does the induction control ones. This could be because the higher distance allows for the wake of the turbine to divert more from its original path avoiding in greater part the next turbine. But the danger of diverting the wake, is that it may be strongly dependent on the scenario, as diverting the wake by yawing, may bring the wake to a different turbine. Nevertheless, the results are promising for high distance wake steering control wind farm scenarios.

More intriguing from these figures, is the influence that different initial wind speed conditions have in each group scenarios, especially when comparing the AWC methods. For the lower speeds (9m/s in Figure 4.8), from the results it seems that applying induction control methods will always bring better solutions than when using wake steering control methods, i.e., the Pareto fronts from curtailment cases dominate entirely the ones from yaw misalignment cases. All the more, for these wind speeds, curtailment methods not only have a big potential to decrease DEL, but also to even increase the power production using curtailment, which as will be seen later when talking about the other higher initial wind speed conditions (12m/s and 15m/s) the solutions which increase the power production will be for curtailment nonexistent or will just increase power by a tiny fraction (although still decreasing DEL). These results discussed above apply for all different TI (5% and 10%) and distance changing (360m and 600m) cases for 9m/s initial wind speeds.

For the case of 12m/s initial wind speed conditions, this pattern begins to change, and induction control Pareto fronts do not all dominate entirely the wake steering control fronts anymore. As seen

in Figure 4.9, the different relationship between these AWC methods depends now more strongly in the TI and the distance between turbines. It can be observed already the difficulty the solutions have to increase power production compared with the 9m/s case.

Finally, there is the group case in Figure 4.10 where the initial wind speed conditions are 15m/s. Above 12 m/s, all turbines will operate at full power, meaning that a reduction in wind speed in the 15 m/s case through the wake will not make any difference. For this reason, there is no power gain obtained by AWC methods when compared with the case without AWC in this figure. But how the loads are affected might indeed be of great interest. In the Pareto fronts obtained by using induction control method, the Pareto front offers very poor trade-offs, where for every gain in DEL reduction, there is a loss in power production of about thrice the size of the gain. But the application of wake steering control techniques can here be of great benefit for the wind farm, as although there is no power production gain, there is a reduction of around 10% (depending on the case) in the results, while virtually keeping the same amount of power production. This case does not offer a great trade-off of solutions per se, as mainly just a couple of solution from these yaw misalignment scenarios are really useful. Nevertheless, by applying single-objective optimization for DEL, bad solutions can be achieved where power production could be highly reduced at very low gain in DEL minimization, therefore indicating the value MOO solutions for these higher initial wind speed cases has.

By observing the same figures, there seems to be a general characteristic geometry of the Pareto optimal solutions for each type of AWC method when compared to each other. While the curvature is somewhat similar, for both cases, the solutions obtained from curtailed wind farm turbines expand twice as longer as the wake steering cases in the power range space (x-axis in figures), while in the DEL range this phenomenon does not happen, varying from case to case (although slightly a higher range for curtailment still in most cases). This makes the curvature of the wake steering PF cases more steep in comparison. This effect means that reducing DEL with the curtailment technique (by choosing solutions in the Pareto front) has a bigger cost in power production as a norm, while when trying to reduce DEL with yaw misalignment technique it does not suppose such a high cost in power reduction. This effect is not decisive in choosing one type or the other of AWC methods, as their relative position to each other varies too much depending of other factors. But it may create cases like in Figure 4.9 for TI of 10% and 600m distance, and TI of 5% and 360m distance, where both methods could be used simultaneously depending of the Decisions Maker choice of strongly decreasing loads or increasing power production.

The initial TI has also a strong influence in wake steering control scenarios. Comparison of all cases (i.e., 5% TI cases versus 10%), indicate that lower initial TIs favour more using wake steering control methods. This improvement is found in all Figures 4.8, 4.9, and 4.10, but more clearly is appreciated in Figure 4.9 where the difference in initial TI causes the Pareto fronts to stop dominating the one obtained from curtailment (in 600m case) when TI is lower, or to be completely dominated by the curtailment PF (in 360m case) when the TI is higher. This could indicate that higher TIs affect yawed turbines more negatively, or that yawed turbines cannot reduce TI downstream as effectively. It could also mean that there is a relative benefit in using curtailment methods instead when initial TI conditions are higher, but more results and investigation are needed for conclusions.

Regarding now the actual individual curtailment and yaw misalignment of each turbine in a specific wind farm, as explained in the comparison of Figures 4.5, 4.6 and 4.7, this behaviour appears to change significantly depending of the initial conditions of wind speed, TI and many other factors, where the the only common behaviour consists in an increase of curtailment and yaw misalignment as the soluitons in the PF go from high power production values (solutions in top-left) to low DEL values (solutions in bottom-right). In the case of induction control cases, according to ECN studies [28] curtailment is applied only to the first row of turbines in a wind farm, as there is too much

of an increase in TI after the first turbine, which leads also to higher wake recovery. But from the results it does not seem to follow the same principle for the MOO optimizer developed. This could be a consequence of the number of wind farms in the results being too small, and not accounting for more turbines in a row. Or it could be something to examine more closely in the future.

But in the cases where wake steering control methods are being applied, there is a behaviour from the optimizer to select always positive angles when performing the optimization in the low DEL values (solutions in bottom-right of the PF). Being the range of the yaw misalignment permitted as a decision variable of between -30° and 30°, it is odd that only positive values are being used in the observed cases, and not a combination of both, specially at solutions favoring low DEL values. At solutions in the top-left of the PFs in yaw misalignment cases, there is some negative angles that may be found in usually less than 20% of the total number of turbines. Also, during the optimization of the problem, negative angles are selected inside of the populations and kept for a short period of generation where they are gradually being replaced by only positive angles. This results are not fully understood, in the way of knowing if it is the fault of the optimizer itself or of how the problem was constructed, or if otherwise there is some grounds for this behaviour, as for example in Table 4.2 where for the case of a 3 turbines case using yaw misalignment, the flapwise DEL has a directionality. In any case more results would need to be provided for confirmation.

A general behaviour observed from the Pareto Fronts, is that AWC methods will, as a general rule, have higher potential to decrease DEL and in higher amounts, than the potential that they have to increase power production. Also, it seems that applying even a small amount of AWC technique of any type, will for almost the entirety of cases bring an advantage in either of the objective. Concluding from the results in this paper that, using any type of AWC will grant better (or at least equal) results, for any of the objectives of power and loads, in almost the entirety of cases. These results therefore advocate for the use of AWC techniques in general.

Regarding the developed MOO optimizer, although it converges and diverges towards solutions in the objective space with satisfactory results, there are some points that should be discussed. Firstly, as shown more clearly in Figure 4.3, the solutions from the optimizer do not reach to the entirety of the trade-offs solutions possible, i.e., the solutions from the single objective optimizer, although using a really high number of generations one approximates it more and more. This could be due to the fact that the solutions the NSGA-II algorithms tries to find are already in there optimum trade-off range. Arriving to the optimum solutions from the single optimizer brings almost tiny amounts gain in one of the objectives when compared with the other. Therefore it could be that the repulsiveness of these solutions makes it harder for the new populations to want to get to these values, and just through rand mutations these values may be finally at some point reached.

Also, when plotting all the population solutions from the entire optimization process, like in Figure 4.2b, there is cases where older population points appear to dominate points in the final Pareto front. These could a bug or error somewhere in the program, or part of the behaviour in some cases of the algorithm NSGA-II, but in any case it is of important interest to assure the best possible trade-offs are always offered.

There are at the moment many directions that the optimizer and analysis of these Pareto fronts could take. Regarding the DEL objective, there are many other types of DEL that could be also further investigated and used as objectives in the near future, especially the edgewise blade root bending moment DEL and the combined tower bending moments DEL. This could be specially done also by increasing the number of objectives so to compare the effect of two of these loads with power production, or the three entirely as objectives to compare its effects. Also using "A-Priori" methods (like the R-NSGA-II) could be done to reach more specific parts of the Pareto front, maybe solving the problem just mentioned in Figure 4.3, where the solutions from the optimizer do not

reach to the entirety of the trade-offs solutions possible, even though it may leave out more efficient solutions outside of the indicated reference points. Figures 4.8, 4.9, and 4.10, also suggest that further investigation regarding the comparison of curtailment technique cases with yaw misalignment technique cases could be very beneficial, and furthermore the combination of both methods together at certain wind speeds and TI and other factors may also seem very promising.

To conclude, a multi-objective wind farm optimizer has been successfully implemented and analyzed, and the testing of this one has verified that it produces satisfactory results using AWC methods to influence the power and the loads in the turbines. The final comparative results of multiple induction control and wake steering control problems run by the optimizer indicate that both methods have high potential in increasing the power production and reducing the loads on the turbines.

6 References

- [1] H.-J. Wagner and J. Mathur, *Introduction to Wind Energy Systems: Basics, Technology and Operation*. Springer-Verlag Berlin Heidelberg, Jan. 2018, ISBN: 978-3-319-68803-9. DOI: 10.1007/978-3-319-68804-6.
- [2] P.-E. Réthoré, "Thrust and wake of a wind turbine: Relationship and measurements," *Technical University of Denmark (DTU)*, pp. 32–44, 2006.
- [3] M. Fischetti and M. Fraccaro, "Machine learning meets mathematical optimization to predict the optimal production of offshore wind parks," *Computers & Operations Research*, vol. 106, Apr. 2018. DOI: 10.1016/j.cor.2018.04.006.
- [4] S. Kanev, F. Savenije, and W. Engels, "Active wake control: An approach to optimize the lifetime operation of wind farms," *Wind Energy*, vol. 21, no. 7, pp. 488–501, 2018.
- [5] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, "Wind energy handbook," in, 2nd. Wiley, Apr. 2011, ISBN: 9780470846063. DOI: 10.1002/0470846062.ch9.
- [6] P. Lynn, *Onshore and Offshore Wind Energy: An Introduction*. Wiley, 2012, ISBN: 9780470976081.
- [7] M. Ragheb and A. M. Ragheb, "Wind turbines theory - the betz equation and optimal rotor tip speed ratio," *Fundamental and Advanced Topics in Wind Power, Rupp Carriveau, InTechOpen*, Jul. 2011. DOI: 10.5772/21398.
- [8] A. N. Vicente, "Validation of wind turbine wake models," Ph.D. dissertation, Universidade de Coimbra, 2018.
- [9] N. Simisioglou, "Wind power wake modelling: Development and application of an actuator disc method for industrial utilization.," Ph.D. dissertation, Acta Universitatis Upsaliensis, 2018.
- [10] N. O. Jensen, "A note on wind generator interaction," *Risø National Laboratory*, no. 2411, 1983.
- [11] M. Bastankhah and F. Porté-Agel, "A new analytical model for wind-turbine wakes," *Renewable energy*, vol. 70, pp. 116–123, 2014.
- [12] ——, "Experimental and theoretical study of wind turbine wakes in yawed conditions," *Journal of Fluid Mechanics*, vol. 806, pp. 506–541, 2016.
- [13] I Katic, J. Højstrup, and N. O. Jensen, "A simple model for cluster efficiency," in *European wind energy association conference and exhibition*, vol. 1, 1986, pp. 407–410.
- [14] N. G. Mortensen, D. N. Heathfield, L. Myllerup, L. Landberg, O. Rathmann, I. Troen, and E. L. Petersen, "Getting started with wasp 9," *Risø-I-2571 (EN). Risø National Laboratory, Technical University of Denmark, Roskilde*, 2007.
- [15] A. Peña, P.-E. Réthoré, and M. P. van der Laan, "On the application of the jensen wake model using a turbulence-dependent wake decay coefficient: The sexbierum case," *Wind Energy*, vol. 19, no. 4, pp. 763–776, 2016. DOI: <https://doi.org/10.1002/we.1863>.
- [16] J. Choi and M. Shan, "Advancement of jensen (park) wake model," in *Proceedings of the European Wind Energy Conference and Exhibition*, sn, 2013, pp. 1–8.
- [17] T. Göçmen, P. van der Laan, P.-E. Réthoré, A. P. Diaz, G. C. Larsen, and S. Ott, "Wind turbine wake models developed at the technical university of denmark: A review," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 752–769, 2016, ISSN: 1364-0321. DOI: <https://doi.org/10.1016/j.rser.2016.01.113>.
- [18] S. Krohn and G. J. van Bussel, "Wind energy online reader," *Delft University of Technology*, Aug. 2021. [Online]. Available: <http://mstudioblackboard.tudelft.nl/duwind/Windenergyonlinereader/index.htm>.

- [19] M. A. Miner, "Cumulative damage in fatigue," 1945.
- [20] H. J. Sutherland, "On the fatigue analysis of wind turbines," 1999.
- [21] G Hayman, "Mlife theory manual for version 1.00," *National Renewable Energy Laboratory (NREL)*, 2012.
- [22] I. Abdallah, K. Tatsis, and E. Chatzi, "Fatigue assessment of a wind turbine blade when output from multiple aero-elastic simulators are available," *Procedia Engineering*, vol. 199, pp. 3170–3175, 2017.
- [23] A. Staino and B. Basu, "Emerging trends in vibration control of wind turbines: A focus on a dual control strategy," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2035, p. 20140069, 2015.
- [24] M. H. Hansen, "Aeroelastic instability problems for wind turbines," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 10, no. 6, pp. 551–577, 2007.
- [25] K Thomsen, J. T. Petersen, E Nim, S Øye, and B Petersen, "A method for determination of damping for edgewise blade vibrations," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 3, no. 4, pp. 233–246, 2000.
- [26] G. P. Corten, K. Lindenburg, and P. Schaak, *Assembly of energy flow collectors, such as windpark, and method of operation*, US Patent 7,299,627, Nov. 2007.
- [27] G. P. Corten and P. Schaak, *Method and installation for extracting energy from a flowing fluid*, US Patent 7,357,622, Apr. 2008.
- [28] S. K. Kanev and F. Savenije, *Active wake control: loads trends*. ECN Petten, 2015.
- [29] P. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Simulation comparison of wake mitigation control strategies for a two-turbine case," *Wind Energy*, vol. 18, no. 12, pp. 2135–2143, 2015.
- [30] P. A. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Evaluating techniques for redirecting turbine wakes using sowfa," *Renewable Energy*, vol. 70, pp. 211–218, 2014.
- [31] M. Nagy, Y. Mansour, and S. Abdelmohsen, "Multi-objective optimization methods as a decision making strategy," *International Journal of Engineering and Technical Research*, Mar. 2020.
- [32] A. Abraham and L. Jain, "Evolutionary multiobjective optimization," in *Advanced Information and Knowledge Processing (AI&KP)*, Springer, 2005, pp. 1–6.
- [33] K. Deb, "Multi-objective optimisation using evolutionary algorithms: An introduction," in *Multi-objective evolutionary optimisation for product design and manufacturing*, Springer, 2011, pp. 3–34.
- [34] ———, *Optimization for innovation in research and practice - multiobjective optimization [lecture]*, GIAN - Ministry of Human Resource Development, Government of India, IIT Kharagpur, Jun. 2016.
- [35] M. T. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: Fundamentals and evolutionary methods," *Natural computing*, vol. 17, no. 3, pp. 585–609, 2018.
- [36] G. R. Zavala, A. J. Nebro, F. Luna, and C. A. C. Coello, "A survey of multi-objective meta-heuristics applied to structural optimization," *Structural and Multidisciplinary Optimization*, vol. 49, no. 4, pp. 537–558, 2014.
- [37] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of nsga-ii for optimizing machining process parameters," *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011, CEIS 2011, ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2011.08.745>.

- [38] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5.
- [39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [40] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. Prasath, "Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach," *Information*, vol. 10, no. 12, p. 390, 2019.
- [41] G. C. Pereira, M. M. de Oliveira, and N. F. Ebecken, "Genetic optimization of artificial neural networks to forecast viriplankton abundance from cytometric data," 2013.
- [42] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *GECCO '06*, 2006.
- [43] C. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension-sweep algorithm for the hypervolume indicator," *2006 IEEE International Conference on Evolutionary Computation*, pp. 1157–1163, 2006.
- [44] S. Garcia and C. T. Trinh, "Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem for novel biocatalysis," *Processes*, vol. 7, no. 6, p. 361, 2019.
- [45] J. Schmidt, L. Vollmer, and M. Dörenkämper, *Flappy -farm layout program in python [computer software]*, Fraunhofer IWES, 2019. [Online]. Available: <https://gitlab.cc-asn.fraunhofer.de/iwes-cfsd/windsite-assessment/flappy/-/wikis/home>.
- [46] NREL, *Floris. version 2.4*, 2021. [Online]. Available: <https://github.com/NREL/floris>.
- [47] J. Schmidt, *Webinar: Flappy v0.3*, Fraunhofer IWES [PowerPoint slides], 2020.
- [48] F. Biscani and D. Izzo, "A parallel global multiobjective framework for optimization: Pagmo," *Journal of Open Source Software*, vol. 5, no. 53, p. 2338, 2020. DOI: 10.21105/joss.02338. [Online]. Available: <https://doi.org/10.21105/joss.02338>.
- [49] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.
- [50] A. Crespo and J. Hernández, "Turbulence characteristics in wind-turbine wakes," *J Wind Eng Ind Aerod*, vol. 61, no. 1, pp. 71–85, 1996, ISSN: 0167-6105. DOI: [https://doi.org/10.1016/0167-6105\(95\)00033-X](https://doi.org/10.1016/0167-6105(95)00033-X). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/016761059500033X>.
- [51] D. R. Munroe, "Genetic programming: The ratio of crossover to mutation as a function of time," 2004.

Appendix A Optimization Procedure Example Case

The following text shows an example of the optimization procedure undergone by the multi-objective optimizer created using curtailment as decision variables and with the objectives of power maximization and DEL minimization. This short case contains only six individual in the population (three of the elite), three decision variables (curtailed turbines) and has a termination criteria of only three generations for exemplification.

Wind Farm and Ambient State description:

```
-----
Number of turbines      : 3 (in horizontal row)
Distance between turbines : 360 m
Wind direction          : 270°
Wind speed              : 13 m/s
TI                      : 0.05
Air density              : 1.225 kg/m^3
```

Decision variables description:

id	decision variable type	turbine	min	max	initial value
0	Curtailment	0	0	0.5	0.01
1	Curtailment	1	0	0.5	0.01
2	Curtailment	2	0	0.5	0.01

Optimization problem description:

```
-----
Nº decision variables : 3
Nº of Objectives       : 2
Objectives              : (1): Power maximization (2): DEL minimization
Nº of Constraints       : 0
Lower Boundary           : (0, 0, 0)
Upper Boundary           : (0.5, 0.5, 0.5)

Algorithm                : NSGA-II
Termination criteria     : Generations = 3
Population                : 6
Elitism                   : 3
```

Population with initial set of solutions (sampling):

CURTAILMENT VALUES PER TURBINE (GENE)			
1st	2nd	3rd	
turbine	turbine	turbine	
0.291489	0.13983775	0.49994281	1st individual
0.34883371	0.42662205	0.4538307	2nd individual
0.40686989	0.32721964	0.30161626	3rd individual
0.23059163	0.29040274	0.15739025	4th individual
0.39777388	0.06094128	0.4863062	5th individual
0.16476624	0.2913476	0.22065509	6th individual

Flappy calculations of the objective funtions in the entire population:

POWER RATIO	FLAPWISE DEL	
0.39508856	2.01958317	1st individual
0.44096394	1.9758253	2nd individual
0.41108297	1.98765501	3rd individual
0.34189483	2.00724175	4th individual
0.39811525	2.07136194	5th individual
0.34306909	2.00854189	6th individual

Calculation of 1st Generation

(Follows the pseudo-code procedure from Section 2.2.3,
which is done internally by PYMOO)

Resulting new population after optmization of first generation
(only 3 individuals due to elitism principle using 3 elite individuals):

CURT T0	CURT T1	CURT T2
0.23160269	0.29040274	0.15645376
0.35306659	0.42195921	0.4538307
0.16375519	0.2913476	0.22159158

Objective function calculation:

POWER RATIO	FLAPWISE DEL
0.34188111	2.0073745
0.44116372	1.97627471
0.34306195	2.00872582

 Calculation of 2nd Generation

Resulting new population:

CURT T0	CURT T1	CURT T2
0.23191812	0.28737281	0.4538307
0.34750722	0.42965194	0.15739025
0.23160269	0.41340016	0.15645376

Objective function calculation:

POWER RATIO	FLAPWISE DEL
0.396285	1.98742488
0.38438531	1.98759041
0.35564671	1.99109946

 Calculation of 3rd Generation (Final)

Final Pareto Front solutions and its decision variables:

PARETO FRONT VALUES

CURT T0	CURT T1	CURT T2	POWER RATIO	FLAPWISE DEL
0.348833	0.426622	0.453830	-->	0.44096394 1.9758253
0.230591	0.290402	0.157390	-->	0.34189483 2.00724175
0.231602	0.290402	0.156453	-->	0.34188111 2.0073745
0.231918	0.287372	0.453830	-->	0.396285 1.98742488
0.347507	0.429651	0.157390	-->	0.38438531 1.98759041

(Only 5 individuals from the 6 members of the population where selected, as the 6th member is not part of the Pareto Frontier)

Appendix B Testing Wind Farm Calculation Scenarios

In Section 3.3 a program is developed in order to test and evaluate the multi-objective optimizers, which performs only the wind farm calculations without optimization. Some of the result figures are shown here, and their behaviour is explained further in the results of Chapter 4 (Note that the colour of the scatter points in the following figures indicates change in the same way as the z-axis variable does, thereby the color bar is redundant and therefore omitted). For all cases, the wind direction comes from the right (270) and the air density considered is 1.225kg/m^3 . The dependent values shown for all cases arise from the turbine at the upmost right location (therefore receiving the wake from the turbines located at its left) which is in all cases uncurtailed and unyawed.

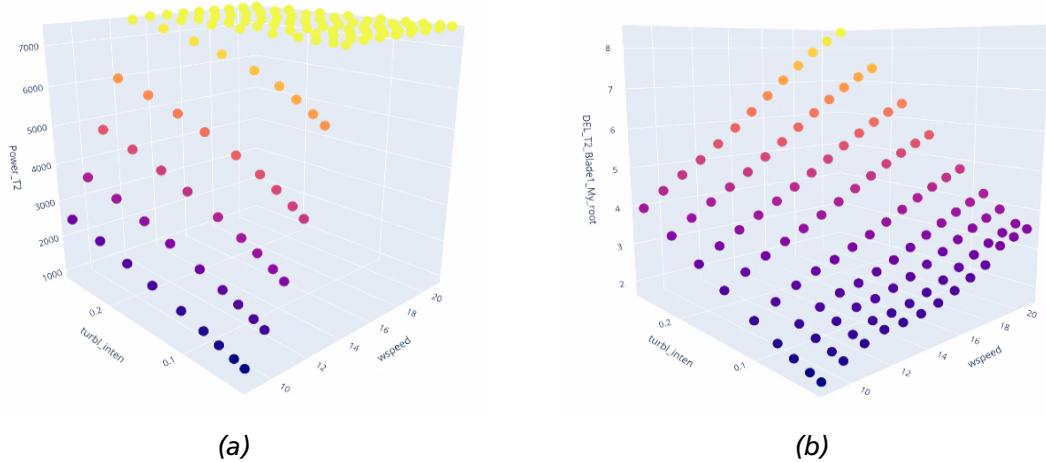


Figure B1 Test showing wind farm calculations for multiple case scenarios using as independent variables wind speed and TI, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other.

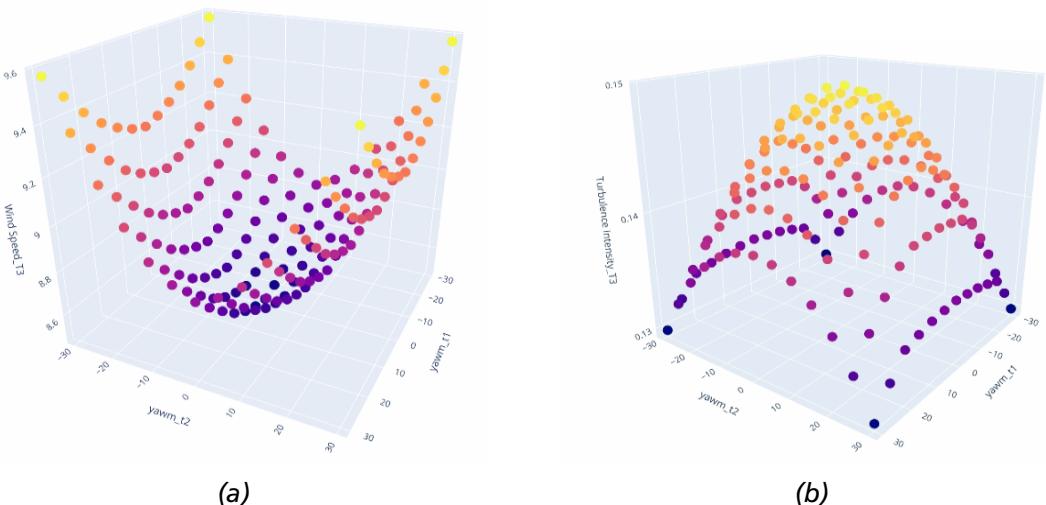


Figure B2 Test showing wind farm calculations for multiple case scenarios using as independent variables the yaw misalignment of the first and second turbine at the leftmost, and showing the resulting a) wind speed and b) TI for a 3 turbine case at a 600m distance from each other, initial 12m/s wind speed and an initial TI of 5%.

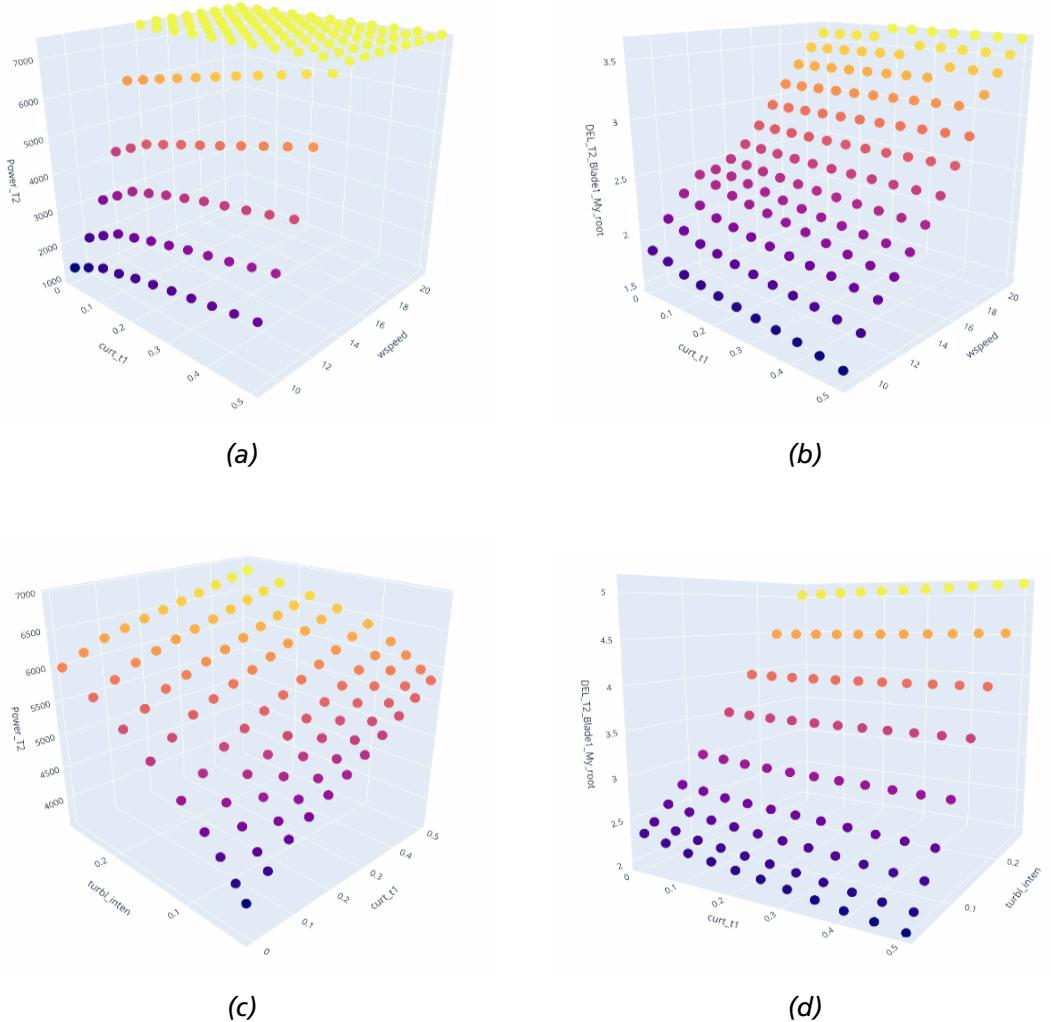


Figure B3 Test showing wind farm calculations for multiple case scenarios using as independent variables (Top) curtailment of leftmost turbine and wind speed, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other and a TI of 5%. And using as independent variables (Bottom) curtailment of leftmost turbine and TI, and showing the resulting c) power and d) Flapwise DEL with same conditions and wind speed of 12m/s.

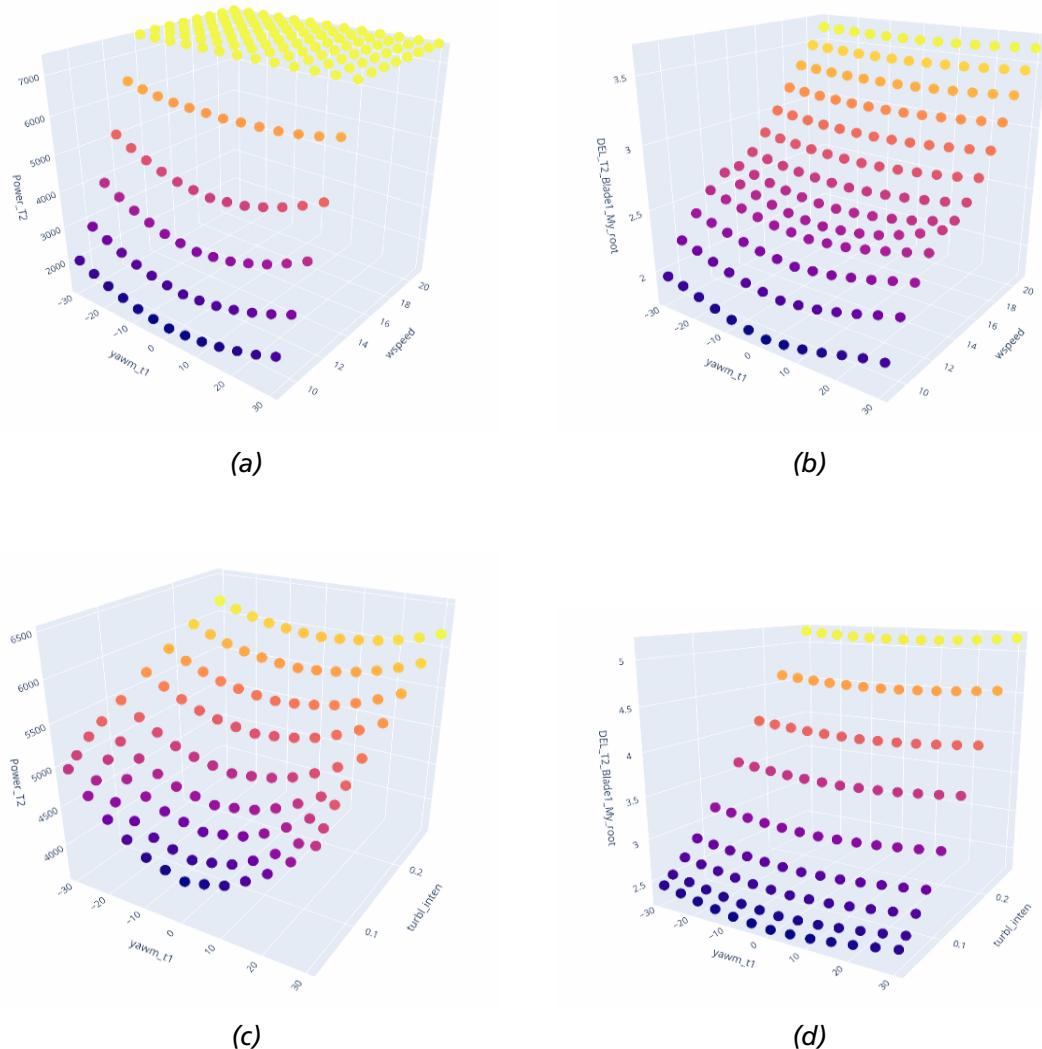


Figure B4 Test showing wind farm calculations for multiple case scenarios using as independent variables (Top) yaw misalignment of leftmost turbine and wind speed, and showing the resulting a) power and b) Flapwise DEL for a 2 turbine case at a 600m distance from each other and a TI of 5%. And using as independent variables (Bottom) yaw misalignment of leftmost turbine and TI, and showing the resulting c) power and d) Flapwise DEL with same conditions and wind speed of 12m/s.

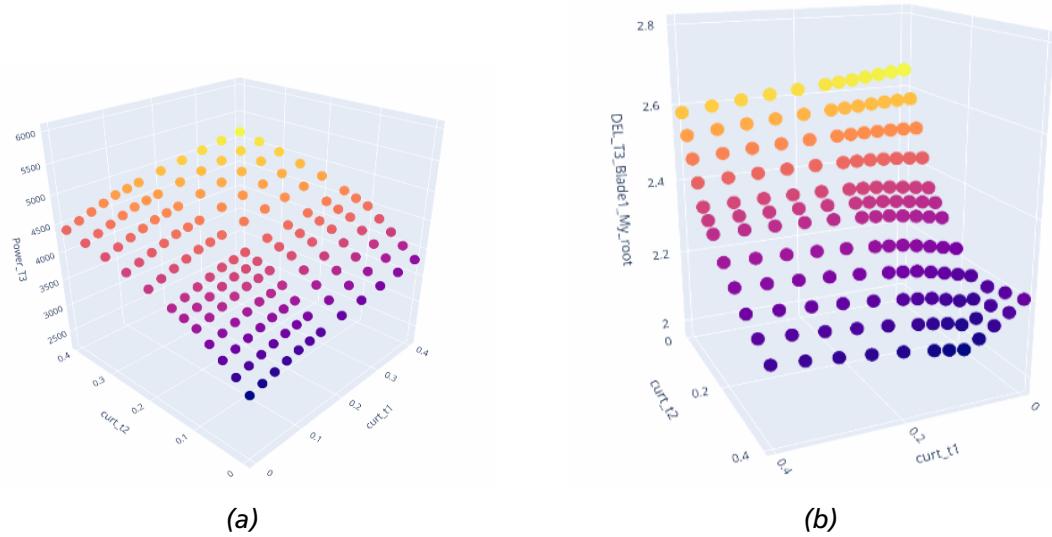


Figure B5 Test showing wind farm calculations for multiple case scenarios using as independent variables the curtailment of the first and second turbine at the leftmost, and showing the resulting a) power and b) Flapwise DEL for a 3 turbine case at a 600m distance from each other, 12m/s wind speed and TI of 5%

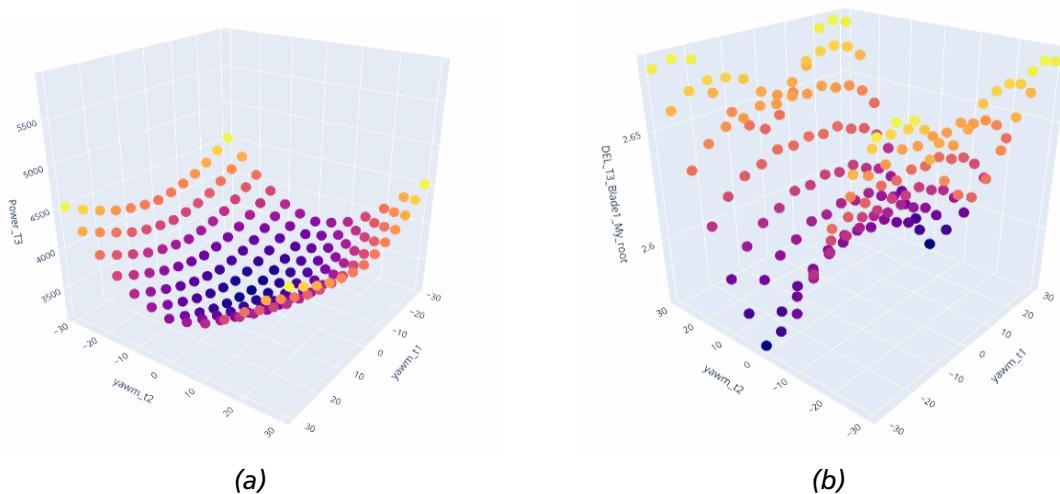


Figure B6 Test showing wind farm calculations for multiple case scenarios using as independent variables the yaw misalignment of the first and second turbine at the leftmost, and showing the resulting a) power and b) Flapwise DEL for a 3 turbine case at a 600m distance from each other, 12m/s wind speed and TI of 5%

Report sheet

1. ISBN or ISSN n.a.	2. Report type (Thesis or internship) Bachelor Thesis
3. Title Wind Farm Optimization: Multi-Objective Problems of Induction Control and Wake Steering Control	
4. Candidate [Name(s), First name(s)] Alberto Jimenez Haro	5. Degree name Engineering Physics
	6. Submission date 12.10.2021
	7. Publication type Non-printed
8. Fraunhofer Institute(s) (Name, Address) Fraunhofer Institute for Wind Energy Systems IWES Küpkersweg 70, 26129 Oldenburg	9. Supervisor Fhl Dr. Jonas Schmidt
	10. Supervisor University Prof. Dr. Martin Kühn
	11. Page No. 77
12. Academic Institution (Name, Address) Carl von Ossietzky Universität Oldenburg Ammerländer Heerstraße 114-118 26129 Oldenburg	13. References 51
	14. Tables 4
	15. Figures 27
16. Additional delivery None	
17. Submitted to (Name, Place, Date) Carl von Ossietzky Universität Oldenburg - Academic Examination Office	
18. Abstract Strategies for mitigating wake effects on turbines are called Active Wake Control strategies and have become very popular in the last years. This paper aims to implement and evaluate multi-objective wind farm optimization problems with the objectives of load minimization and yield maximization, and analyze the Pareto optimal solutions. The optimization problems are solved by using induction control and wake steering control as decision variables together with simulation provided by the software MoWiT. The results finally indicate that the multi-objective optimizer does calculate problems of AWC, converging to the optimal solutions and diverging towards these solutions in a satisfactory manner. The comparison results show multiple Pareto fronts for both AWC types, which seem to produce solutions which almost always dominate the solution for the base case, i.e., without using AWC strategies, concluding that both methods have positive potential towards increasing the power production and reducing the loads on the turbines of wind farms.	
19. Keywords Active Wake Control, Wind Farm Optimization, Multi-Objective Optimization, Induction Control, Wake Steering Control.	
20. Publisher n.a.	21. Price n.a.

Declaration of Academic Integrity

I hereby confirm that this thesis is entirely my own work. I confirm that no part has been copied from either a book or any other source - including the internet - except where such sections are clearly shown as quotations and the sources have been correctly identified within the text or in the list of references. Moreover I confirm that I have taken notice of the 'Leitlinien guter wissenschaftlicher Praxis' of the University of Oldenburg.

Alberto Jimenez Haro, October 12, 2021

Signature,

A handwritten signature in black ink, appearing to read "ALBERTO JIMENEZ HARO". The signature is somewhat stylized and includes a small flourish at the end.