A guideline for selecting MDAO workflows with an application in offshore wind energy

Sanchez Perez Moreno, Sebastian

DOI
10.4233/uuid:ea1b4101-0e55-4abe-9539-ae5d81cf9f65

Publication date
2019

Document Version
Final published version

Citation (APA)
Sanchez Perez Moreno, S. (2019). *A guideline for selecting MDAO workflows with an application in offshore wind energy*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:ea1b4101-0e55-4abe-9539-ae5d81cf9f65

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A GUIDELINE FOR SELECTING MDAO WORKFLOWS WITH AN APPLICATION IN OFFSHORE WIND ENERGY

A GUIDELINE FOR SELECTING MDAO WORKFLOWS WITH AN APPLICATION IN OFFSHORE WIND ENERGY

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus, Prof. dr. ir. T. H. J. J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
donderdag 14 maart 2019 om 12:30 uur

door

Sebastian SANCHEZ PEREZ-MORENO

Master of Science in New and Renewable Energy,
Durham University, United Kingdom,
geboren te Mexico City, Mexico.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. G. J. W. van Bussel	Technische Universiteit Delft, promotor
Dr. ir. M. B. Zaaijer	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. ir. K. I. Aardal	Technische Universiteit Delft
Prof. dr. P. Palensky	Technische Universiteit Delft
Prof. dr. P. W. Cheng	Universität Stuttgart, Duitsland
Dr. ir. G. La Rocca	Technische Universiteit Delft
Dr. K. Dykes	National Renewable Energy Laboratory, Verenigde Staten van Amerika

Keywords: Offshore wind farm design, systems engineering, MDAO workflows

ISBN 978-94-6366-138-6

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

A Luciana, Bruno, Camilo y Canek

ACKNOWLEDGEMENTS

Three factors can make or break the passion put in a PhD research: the supervisors, the topic and the place. I am one of the very lucky few who did not have to sacrifice either of them. Although I absolutely love the topic of my research and lived in the beautiful city of Delft, my deepest gratitude is to have been supervised by Michiel Zaaijer and Gerard van Bussel, and to have been channelled to them by Gijs van Kuik.

Michiel, you are by far one of the most generous people I know. I am grateful beyond words for the amount of hours, wisdom and constructive criticism you brought not only onto this research, but also into my personal development. I will never forget the quality of our discussions, and your triggering questions that always gave me complementary and valuable viewpoints. What I learnt from you goes beyond what is written in this book and will stay with me for life.

Thank you Gerard for the guidance, advice and for helping me keep an overview of my overall research objectives, which are usually easy to forget. I also deeply appreciate your encouragement for travelling to conferences and courses.

Gijs, it was thanks to you that I sneaked into this wonderful research group, where I have made friends for life.

I am personally thankful to Sylvia, whose selfless and friendly eternal support keeps the office from collapsing. Thank you for making our life so much easier.

My research was strongly influenced by my colleagues at the International Energy Agency Wind Task 37 for Systems Engineering. I appreciate so much having been invited to be part of this small group of MDAO enthusiasts, who put in their time and effort for the advancement of our field. Katherine, Karl, Mike, Pietro, Pierre and Frederik, thank you for the great collaboration and joyful dinners after our day-long meetings. Katherine thanks for hosting me at NREL, a memorable trip that taught me so much.

I also thank the MSc students that I co-supervised or used my tools, for their trust: Ioannis, Marthijn, Tanuj, Vishal, Raffaello and Mihir.

Our research group has given me life-lasting friendships and purely good memories, I will miss all of you. Navi, I treasure our mezcal nights and sharing amazing discussions, thank you. Vinit, you have made the office a happier place, you are a true friend. Daniel, thank you for the deep and inspiring conversations involving Pingüino Rodriguez. Rene, thank you for the uncontrollable laughs. Bruce, it is a pleasure to party with your light and warm spirit. Ashim, your noble heart gave me memorable moments. Juan, thank you for caring so much. Zi, I loved being your roommate, may life give you only sweet surprises. Ricardo, sharing a beer with you has always been a joy. Wei, thank you for the charming times together. Simon, you showed me that hierarchy is worth little outside the office, thanks for being a great leader, I wish you all the best. Sebastian, thank you for showing me the true meaning of work and leading our fun card game. Mikko, thank you for keeping it real, I wish you pure success. Carlos, thank you for trusting me with your MOOC and giving me an opportunity, I will never forget it. Erik, thank you for your

sincere interest in my work, I enjoy so much working with you. Bedassa, thank you for making our office a better place. Delphine, thank you for smiling all the time. Mark, thank you so much for your disinterested translation. To Etana, Lars, Giuseppe, Lorenzo, Maarten, Ye, Linda, Ruud, Dick, Axelle, Nando, Wim, Sharif, Chihoon, Julia, Christopher, Gael, Ming, Leandro, Jing, Ashwin, Jaco, Laurent, Tom, Carlos, Qingqing, Ehsan, Livia, Fra, Dani, Roland, thank you for the great group atmosphere and I wish you the very best.

Beyond our research group, I am thankful for the talented and outstanding friends I met at this university: Imco, Günther, Javi, Inés, Megan, Thomas, Tony, Nando, Tom, Sebastian, Selma, Victoria, Filip, Roberto, Johan, Marsil, Dadui, Hermes, Fabricio, and Toño.

My peace of mind was strongly supported by my friends at the Latin American students association LATITUD. Edgard, Pablo, Juan Carlos, Juan Pablo, Manu, Toño, Aldo, Diana, Mateo, Meli, Libardo, Memo, Silvana, Jorge, Jorge, León, Marice, Victor, Emilio, Constantino, Tania, Adonis, Ana y Jorge. Learning our cultural differences lead to countless nights of pure laughter. Thank you for spreading our cultures beyond borders.

My football team AC Brancaleone deserves a spot for all the goalless nights. Thank you for your commitment and team spirit.

A good life in the Netherlands has been a pillar for finishing my PhD. And the people who made my life in this country so happy includes Anna, Nora, Alan and Henning; Jeannette, Thijs, Nick, Alex, Mariana, Andreas, Nilo, Karina, Ramón, Martina, Esther, Natalia and Jaime.

Special thanks go to my friends and family who came to stay at our small apartment in Delft. You made it feel closer to home.

My family deserves a special place in this list for continuously offering me words of encouragement. My mother Gisele for her friendship, love, advice and being the source of all of me. My father Alberto for loving me and being the most wonderful person. My sister Luciana and brothers Bruno, Camilo and Canek, for our intrinsic and infinite affective bond. I love you all so much. Thank you Vania for being here, you are a true star in my life.

The final word is saved for the highlight of my life, Frida. Every single day of my PhD research was a happy one solely for getting back home and being with you. Your neverending support, love, understanding, patience, friendship, advice and encouragement filled me with strength and passion to finish this work. While everyone in this list is part of my past and my present, you are also my future, and for that I am the most grateful man on Earth.

Gracias!

SUMMARY

A system is a set of interconnected components whose individual behaviour and interactions determine the overall performance of the set. Wind farms are amongst the most complex systems deployed worldwide, based on their uncertainty, heterogeneity and complexity. Moreover, many technical and social disciplines may simultaneously describe the performance of a complex system such as wind farms.

Currently, the components of offshore wind farms are to a large degree designed sequentially. The lack of knowledge about how some design decisions affect other aspects of the plant leads to suboptimal designs and higher costs of energy. Furthermore, design automation is not fully exploited in the offshore wind industry. Multidisciplinary Design Analysis and Optimisation, commonly shortened to MDAO, is a systems engineering technique that allows to computationally explore many more designs than is manually possible.

MDAO consists of a workflow where a set of computational tools are coupled to simulate the entire system. The coupling of the tools is called the analysis block. Additionally, by including drivers that control how and when each tool is executed (e.g. optimisation algorithm), the workflow can fulfil a certain functionality. The functionality of the workflow is defined by a use case. Use case is the term that describes a particular domain problem that can be solved with an MDAO workflow.

Complications arise when its acknowledged that computational tools of varying levels of fidelity and different driver algorithms may be used to solve the same use case. Several MDAO workflows may thus be built for the same purpose.

Hence, the objective of this work is to develop a systematic and objective methodology for selecting the best-performing model fidelities and driver algorithms of an MDAO workflow in the domain of offshore wind farms.

A tool was developed in this work with two layers of functionality. The outer layer of the tool requests the models and driver algorithms to be coupled from the workflow designer, instantiates and, if necessary, executes the MDAO workflow. The inner layer is an MDAO workflow that requests from the wind farm designer the site conditions and fixed design parameters, and its output is an improved design of the offshore wind farm.

This tool on its own can bring value to education and industry. Exploring the interactions between design choices of different sub-components of the wind farm leads to better understanding of the complex dynamics that take place in the design and operation of an offshore wind farm.

This work further proposes a guideline to systematically, quantitatively and objectively simplify the decision making process to a more tractable problem when hundreds of thousands of MDAO workflows can be built with a set of tools and drivers.

The guideline is broken down into two phases. In phase 1, the model fidelities of the analysis block are optimised using the Multiobjective Particle Swarm Optimisation for Categorical Variables (MOPSOC) algorithm. The output of this algorithm is the Pareto

front of analysis blocks across any number of predefined and possibly conflicting criteria. Phase 2 is a multi-criteria decision making process, where a few combinations of best-performing analysis blocks and driver algorithms are coupled and scored against multiple criteria.

The guideline helps select the set of solutions that efficiently solve trade-offs between criteria and provides good compromises, and enables the designer take a lower risk by choosing an MDAO workflow that maximises certain criteria while sacrificing the least on others.

The guideline proposed in this work is a novel mechanism to evaluate, compare, and rank different MDAO workflows. There are no known formal methodologies for this purpose in the published literature.

A workflow designer benefits most from this guideline if the outcome of the use case is sufficiently critical to a research or development project, or if the MDAO workflows selected are expected to become the workhorse throughout a project.

At the last stage of the guideline, as with any other multi-criteria decision analysis problem, the decision maker must take a more active role, supported by expertise, experience, data and the outcome of this guideline, to choose a final single or few MDAO workflows that comply with the predefined requirements.

The guideline avoids the workflow designer from making catastrophic decisions based on intuition while also avoiding becoming overwhelmed by the great amount of analyses and information involved in choosing the best-performing MDAO workflows. The greatest advantage of adhering to this guideline is that in a short time, workflow designers can largely reduce the amount of alternatives to choose from. Using one of the selected MDAO workflows for solving a use case can impact the total budget allocated to a design optimisation or a what-if analysis campaign. The benefits can also cover the development phase of an offshore wind plant, by providing a more accurate analysis block for making financial, logistical or manufacturing decisions. Other positive impacts relate to practical issues such as reducing the costs of computational hardware and software.

Conversely, the drawbacks of following this guideline are that the workflow designer has to commit a good amount of effort into defining the criteria that will govern the selection process and into enabling the automatic generation of analysis blocks and MDAO workflows. In addition, the MOPSOC algorithm is of a stochastic nature.

The MDAO community benefits from this research as the ever-present trade-off between the sophistication and cost of multidisciplinary analysis and optimisation workflows continues to be overlooked. In essence, this guideline enables more efficient, cheaper and optimal system design processes.

Finally, the guideline is instantiated and validated using a multidisciplinary design optimisation problem, where the layout, electrical infrastructure and foundations of an offshore wind farm are simultaneously optimised.

SAMENVATTING

Een systeem is een set van gekoppelde componenten. De kenmerken van de individuele componenten en de interacties tussen de componenten zijn bepalend voor het functioneren van de volledige set. Windparken behoren tot de meest complexe systemen in de wereld vanwege onzekerheden, heterogeniteit en complexiteit. Bovendien zijn er veel technische en sociale aspecten gemoeid bij een complex systeem zoals een windpark.

Momenteel worden de componenten van offshore windparken veelal los van elkaar ontworpen. Het gebrek aan kennis over hoe ontwerpkeuzes voor een component een ander component beïnvloeden leidt tot suboptimale ontwerpen en hogere energiekosten. De offshore windindustrie maakt weinig gebruik van technieken voor het automatiseren van het ontwerp. Multidisciplinaire ontwerpanalyse en -optimalisatie (MDAO) is een techniek waarmee veel meer verschillende ontwerpen kunnen worden geëvalueerd dan handmatig mogelijk is.

MDAO bestaat uit een workflow waarin rekentools onderling zijn gekoppeld om het hele systeem door te rekenen. De gekoppelde tools vormen samen het analyseblok. Tevens onderdeel van de workflow is een algoritme die regelt hoe en wanneer elk tool wordt aangeroepen (bijvoorbeeld een optimalisatie-algoritme). Het algoritme zorgt ervoor dat de workflow een bepaalde functionaliteit vervuld. De functionaliteit van de workflow wordt bepaald door de use-case. De use-case beschrijft het specifieke probleem dat kan worden opgelost met een MDAO worklow. Meerdere algoritmes en rekentools met verschillende precisie kunnen worden gebruikt voor dezelfde use-case. Hierdoor kunnen verschillende varianten van MDAO-workflows worden samengesteld voor hetzelfde doel.

Het doel van dit werk is om een systematische en objectieve methodiek te ontwikkelen voor het selecteren van de best presterende rekenmodellen en algoritmen voor een MDAO-workflow in het domein van offshore windparken.

Een tool is ontwikkeld bestaande uit twee lagen aan functionalitiet. De buitenste laag van de tool vraagt de workflow-ontwerper om de te koppelen modellen en algoritmen, maakt de MDAO-worklow aan en, indien nodig, voert deze uit. De binnenste laag is een MDAO-workflow welke de wind condities en vaste ontwerpparameters nodig heeft als input van de ontwerper van het windpark. De output is een verbeterd ontwerp van het offshore windpark.

Deze tool kan van waarde zijn voor onderwijs en industrie. Het onderzoeken van de interacties tussen ontwerpkeuzes van verschillende subcomponenten van het windpark leidt tot meer kennis van de complexe dynamiek in het ontwerp en de werking van een offshore windpark.

Dit werk stelt verder een richtlijn voor om het besluitvormingsproces systematisch, kwantitatief en objectief te reduceren tot een beter hanteerbaar probleem wanneer honderdduizenden MDAO-workflows kunnen worden samengesteld uit een set tools en algoritmes.

De richtlijn is onderverdeeld in twee fasen. In fase 1 worden combinaties van rekenmodellen van het analyseblok geselecteerd met behulp van het Multiobjective Particle Swarm Optimization for Categorical Variables algoritme (MOPSOC). De output van dit algoritme is een Pareto-front van analyseblokken over een willekeurig aantal vooraf gedefinieerde en mogelijk tegenstrijdige criteria. Fase 2 is een besluitvormingsproces met meerdere criteria, waarbij een paar combinaties van de best presterende analyseblokken en algoritmes worden gekoppeld en op meerdere criteria worden beoordeeld.

Uit de richtlijn volgt de set aan oplossingen die op efficiënte wijze criteria afweegt en tot goede compromissen komt. De MDAO workflows waaruit de ontwerper kan kiezen zijn ieder optimaal bevonden bij een bepaalde afweging en de keuze heeft hierdoor een lager risico.

De richtlijn die in dit werk wordt voorgesteld is een nieuwe manier om verschillende MDAO-workflows te evalueren, vergelijken en rangschikken. Er zijn geen formele methodieken bekend voor dit doel in de literatuur.

Een workflow-ontwerper kan het meest uit deze richtlijn halen wanneer de uitkomst van de use case voldoende kritiek is voor een onderzoeks- of ontwikkelingsproject, of als wordt verwacht dat de MDAO-workflows de werkpaarden worden gedurende een project.

In de laatste fase van de richtlijn moet de beslissingsnemer, net als met elk ander beslissingsanalyse-probleem met meerdere criteria, een actievere rol aannemen, met behulp van expertise, ervaring, gegevens en de uitkomst van deze richtlijn, om één of enkele MDAO-workflows te kiezen die voldoen aan de vooraf vastgestelde eisen.

De richtlijn voorkomt dat de workflow-ontwerper foute beslissingen neemt op basis van intuïtie. Bovendien wordt voorkomen dat de ontwerper overweldigd raakt door de vele analyses en informatie die komen kijken bij het kiezen van de best presterende MDAO-workflows. Het grootste voordeel van het volgen van deze richtlijn is dat in een korte tijd workflow-ontwerpers het aantal alternatieven om uit te kiezen kunnen beperken. Het gebruiken van één van de geselecteerde MDAO-workflows voor het oplossen van een use case kan van invloed zijn op het totale budget dat is toegewezen aan een ontwerpoptimalisatie of een what-if analyse. De richtlijn komt ook ten goede aan de ontwikkelingsfase van een offshore windmolenpark, door een nauwkeuriger analyseblok te bieden voor het nemen van financiële, logistieke of productiebeslissingen. Andere positieve effecten hebben betrekking op praktische zaken zoals het verminderen van de kosten van computer hardware en software.

Een nadeel van het volgen van deze richtlijn is dat het vaststellen van de criteria, die bepalend zijn voor het selectieproces, veel inspanning vereist van de workflow-ontwerper. Net als het mogelijk maken van het automatisch genereren van analyseblokken en MDAO-workflows. Bovendien is het MOPSOC-algoritme stochastisch.

De MDAO-gemeenschap profiteert van dit onderzoek omdat de wisselwerking tussen de verfijning en de kosten van MDAO-workflows nog steeds over het hoofd wordt gezien. In wezen maakt deze richtlijn efficiëntere, goedkopere en optimale systeemontwerpprocessen mogelijk.

Tot slot wordt de richtlijn aangemaakt en gevalideerd met behulp van een multidisciplinair ontwerpoptimalisatieprobleem, waarbij de lay-out, de elektrische infrastructuur en de fundamenten van een offshore windpark gelijktijdig worden geoptimaliseerd.

CONTENTS

NOMENCLATURE

Acronyms

AEP	Annual energy production
A1	Ainslie 1D wake model
A2	Ainslie 2D wake model
BEM	Blade element momentum theory
BOS	Balance of station
C	Dummy model
CAPEX	Capital expenditure
DE	Differential evolution
DECOM	Decommissioning costs
DPD	Discrete probability distribution
DR	Danish recommendation wake-added TI model
DWA	Dynamic weighted aggregation
EW	Esau-Williams heuristic
F	Frandsen wake-added TI model
F2	Frandsen 2 wake-added TI model
J	Jensen wake model
L	Larsen wake model
LCOE	Levelised cost of energy
MAX	Maximum deficit wake merge model
MCDA	Multicriteria decision analysis
MDAO	Multidisciplinary Design Analysis and Optimisation
MOPSOC	Multiobjective particle swarm optimisation for categorical variables
MULT	Multiplied deficits wake merge model
O&M	Operations and maintenance
OPEX	Operational expenditure
POS	Planar Open Savings heuristic
PSO	Particle swarm optimisation
Q	Quarton wake-added TI model
QB	QBlade wind turbine tool
RSS	Root sum square wake merge model
SUM	Summed deficits wake merge model
TI	Turbulence intensity
TP	TeamPlay design tool
WS	WindSim wind turbine tool
XDSM	Extended design structure matrix

Symbols

C_k	Score of criterion k
$\hat{C}_k$	Normalised score of criterion k
D_{rotor}	Diameter of the turbine rotor
D_{tower}	Diameter of the turbine tower
D_{TP}	Diameter of the transition piece
H	Water depth
I	Set of input parameters to a model
n	Economic lifetime of the project
N_s	MOPSOC number of DPD samples
N_S	Number of offshore substations
N_T	Number of turbines
OS	Open-sourceness of a model
$\mathbf{P}_i^l$	PSO local best solution
$\mathbf{P}^g$	PSO global best solution
Px_{ij}	MOPSOC probability of position x_i taking on the value y_j
Pv_{ij}	MOPSOC probability of velocity v_i taking on the value y_j
r	Real interest rate
r	Random number
s	MOPSOC sample solution
t	Year into the lifetime of the project
T_i	Wind turbine i
Δu	Wind speed deficit
$\mathbf{v}_i^j$	MOPSOC velocity of particle i at time step j
$\mathbf{x}_i^j$	MOPSOC position of particle i at time step j
W	MDAO workflow
w^{in}	PSO inertia weight
w^g	PSO social weight
w^l	PSO cognitive weight
X	Design variable
$\mathbf{y}$	MOPSOC vector of categorical values

Greek symbols

δ	PSO turbulence parameter
ϵ	MOPSOC probability-scaling parameter
ν	MOPSOC DPD of a particle's velocity
χ	MOPSOC DPD of a particle's position

Introduction

***Design**: To decide upon the looks and function of something before it's made.*

Oxford English Dictionary

This research began with one goal: to improve the use of computational tools during the design process of offshore wind farms.

This chapter begins with the benefits from making renewable energy cheaper, and goes on to touch upon the current state of wind energy and presents methods for designing wind energy systems, the importance of optimisation for design, and ultimately, what it means to design wind farms using Multidisciplinary Design Analysis and Optimisation (MDAO) techniques.

By further introducing the concepts of model fidelity and optimisation algorithms, this chapter presents the value of using the right tools for the right job.

Finally, a description of the concrete research objective is given: to devise a methodology for selecting the best-performing model fidelities and optimisation algorithms of an MDAO workflow in the domain of offshore wind farms.

1

1.1. THE WORLD NEEDS COST-COMPETITIVE WIND ENERGY

1.1.1. GLOBAL WARMING

PLANET EARTH AND HUMAN LIVES are being dangerously threatened by global warming. 2017 was the hottest year on record with no occurrence of the climate phenomenon El Niño, and 17 of the 18 hottest years have occurred since the year 2000 [1]. In 2010, Hansen et al. concluded that since the late 1970s, there has been a trend of 0.15 - 0.20 °C increase in the global temperature per decade [2]. Until 2017 that trend has further worsened, as the temperature increased a further 0.2 °C in only 8 years. Figure 1.1 shows that both CO_2 equivalent (CO_2e) emissions and global mean surface have gradually increased since the 1960s. Furthermore, annual average temperatures may drop or increase from year to year, though the 5-year average temperature is rising steadily.

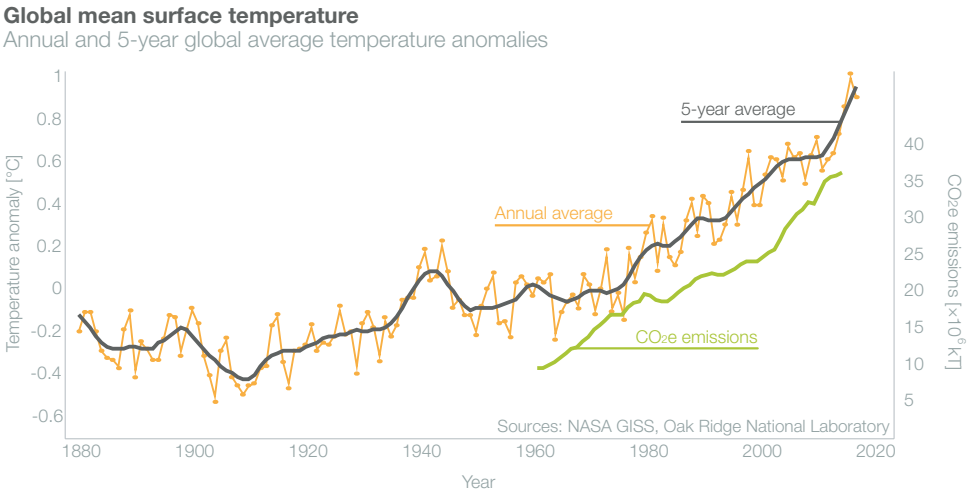


Figure 1.1: Global mean surface temperature anomalies compared to the average in the period 1951-1980. The increase is more evident in the 5-year average (modified from [2]).

The worrying fact is that global warming is associated with an increase in weather-related catastrophes (see Fig. 1.2), the rise of the sea level, and the spread of tropical diseases to geographic regions further north of the Equator. The combination of these phenomena is what puts human life and Earth's biodiversity at risk. Three of these phenomena are explained in more detail.

First, storms and floods are becoming more frequent and more violent, due to higher evaporation rates. Likewise, prolonged droughts and heatwaves result in water scarcity, desertification and hunger behind.

Second, 600 million people live within 10 meters of sea level [4], and could face new hazards due to a rise in sea level provoked by global warming. Mass migration is a phenomenon no country is prepared for and could lead to more geopolitical instabilities.

Third, all living beings could be subject to diseases both existing and potentially new ones. The spread of tropical diseases such as malaria and dengue fever is influenced by climate, and epidemics could lead to human health crises and collapse of livestock

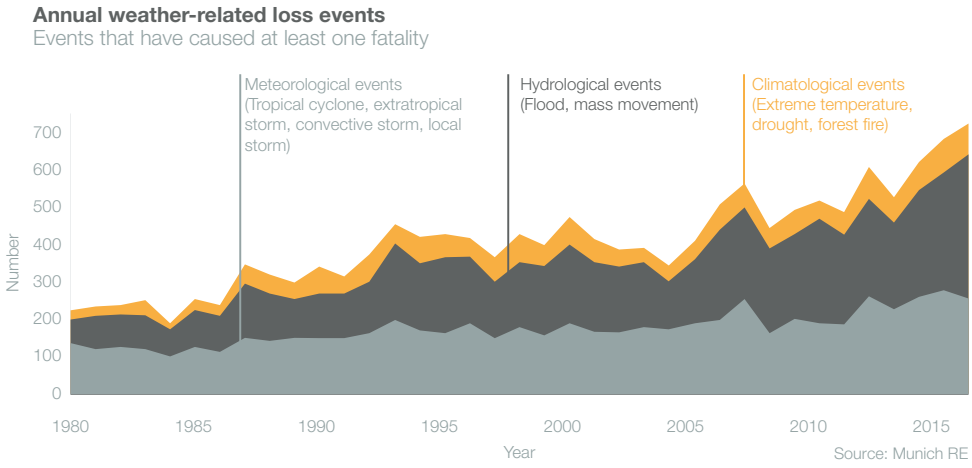


Figure 1.2: Weather-related loss events worldwide (modified from a report by Muniche RE[3]).

industry and plant life[5].

In short, if left unaddressed, global warming has the potential to lead to a total devastation of human society.

1.1.2. HUMAN IMPACT ON GLOBAL WARMING

The concentration of greenhouse gases are now at unprecedented levels in at least the last 800,000 years, and their emission has increased since the industrial era[6]. The Intergovernmental Panel on Climate Change (IPCC) has stated that "it is extremely likely that more than half of the observed increase in global average surface temperature was caused by the anthropogenic increase in greenhouse gases concentrations"[6]. Figure 1.1 shows the increase in global anthropogenic CO_2 equivalent annual emissions since 1960.

The burning of fossil fuels and industrial processes have contributed to 78% of the CO_2 equivalent in the atmosphere. Since 2010, roughly 50% of greenhouse gas emissions came from transport and production of electricity and heat alone[6].

1.1.3. RENEWABLE ENERGIES AS MITIGATION FACTORS

Most renewable energy sources have much lower lifecycle greenhouse emissions than fossil fuels[6]. The right hand side of Fig. 1.3 compares the carbon dioxide equivalent lifecycle emissions of different energy sources. Wind energy has a clear advantage in this regard. In fact, according to the Global Wind Energy Council, 20% of the pledged reductions of CO_2 equivalent emissions by industrialised countries for 2020 can be met by wind energy alone.

1.1.4. THE COST OF WIND ENERGY

A common system level performance indicator for comparing offshore wind plant projects is the levelised cost of energy (LCOE)[8–10], as it aggregates the levelised system performance and costs in a single metric. LCOE is calculated as:

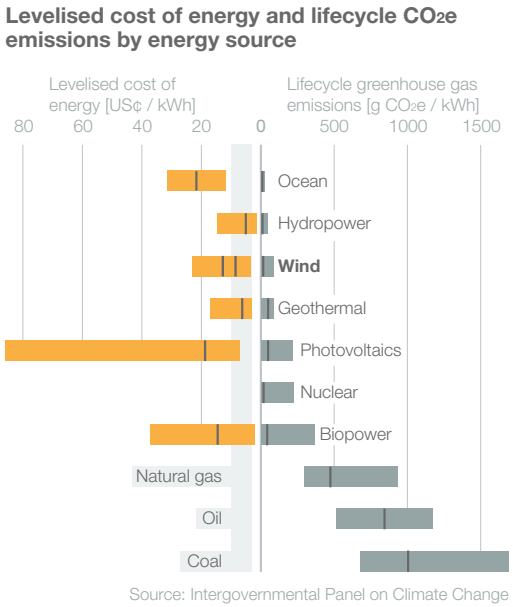


Figure 1.3: Levelised cost of energy and lifecycle CO_2e emissions of different energy sources. Non-renewables and their LCOE are shaded in light gray (modified from Edenhofer et al.[7]).

$$LCOE = \frac{CAPEX + \sum_{t=1}^{n} OPEX_t (1+r)^{-t} + DECOM(1+r)^{-n}}{\sum_{t=1}^{n} effAEP_t (1+r)^{-t}}, \tag{1.1}$$

where $CAPEX$ is the capital expenditure in year $t = 0$, $OPEX_t$ is the operational expenditure, including maintenance costs, in year t, $DECOM$ are the decommissioning costs, $effAEP_t$ is the effective annual energy production of the plant in year t after electrical and availability losses, n is the economic lifetime of the project in years, and r is the real interest rate.

The left hand side of Fig. 1.3 shows that the levelised cost per unit of electricity converted (LCOE) from renewable sources tends to be higher than from non-renewable sources.

Since wind turbines do not incur fuel expenses, their operation is cheap. However, higher capital costs is what makes the LCOE of wind energy typically higher than their non-renewable counterparts.

One of the goals set by governments, industry and academia is to reduce the LCOE of wind energy. As an example, European wind farm developers DONG (now Ørsted) and E.ON set out to reduce the LCOE of offshore wind farms from 160 €/MWh in 2012 to less than 90 €/MWh in 2020[11], and Ørsted claims they will reach an LCOE of 78 €/MWh in the Borssele 1 and 2 offshore wind farms, due to go live in 2020[12]. Also, Vattenfall's Kriegers Flak project recently set a record LCOE forecast of 40 €/MWh[13].

1.1.5. HOW TO REDUCE THE COST OF WIND ENERGY

The long sought LCOE reduction has its origins in innovations on many fronts. Figure 1.4 shows the sensitivity of LCOE to major concepts that govern the performance and cost of an offshore wind plant.

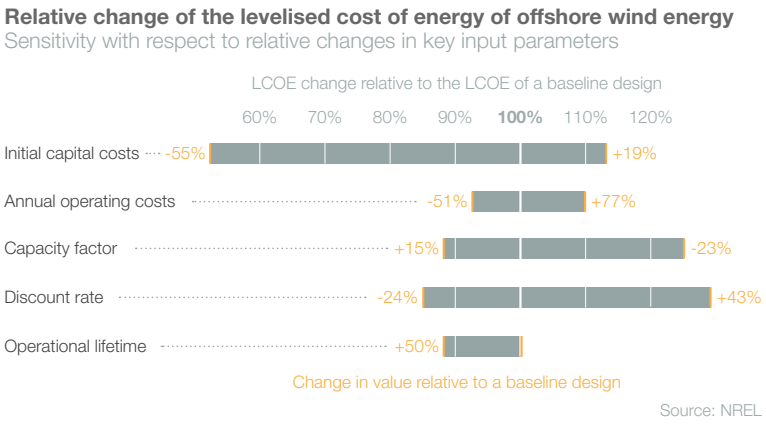


Figure 1.4: Sensitivity of the levelised cost of energy of offshore wind energy with respect to changes in key input parameters, compared to a baseline design (modified from Tegen et al. [14]).

The keys to achieving LCOE reductions on the initial capital costs are the optimisation and streamlining of processes for installation, the economies of scales due to the ever growing nameplate capacity of wind turbines, the maturation of the entire supply-chain, and the technological innovations in most of the physical components of the wind plant. In addition, optimisation of the operations, maintenance and decommissioning logistics can lead to reductions in the annual operating costs [11,13,14]. Furthermore, the adaptation of some components and processes to specific sites and projects also helps take advantage of the local conditions and thus improve the capacity factor and costs, compared to off-the-shelf components. The same is valid for the wind farm layout, since maximising the array efficiency improves the capacity factor and costs. The discount rate is influenced by the risk the investor is willing to assume. Higher risk is rewarded with higher returns, since greater losses are also possible. As a consequence, reducing uncertainties in the performance and cost helps reduce the discount rate.

Table 1.1 shows concrete examples of research and development programmes in offshore wind energy that could translate into a reduction of the cost of energy. Tegen et. al. drew these potential improvements [14] with further elaboration.

1.1.6. REDUCING COSTS BY DESIGN

As mentioned earlier, the optimisation of procedures and physical components for a specific site has the potential to reduce LCOE. This is achieved by improving the performance of components and the efficiency of procedures, and by reducing their associated costs. An example is the aeroelastic optimisation of a wind turbine blade. The shape of aerofoils at each section of the blade can be optimised to improve the overall aerodynamic performance of the blade, which in turn may translate into a higher energy yield.

Table 1.1: Concrete examples of innovations and their implications for reducing overall costs or improving the energy yield of offshore wind plants (modified from Tegen et al. [14]).

R&D area	Potential changes	Expected impact
Turbine scaling	Larger rotors, taller towers, higher nameplate capacity, primarily enabled by advanced controls. Manufacturing efficiency and quality assurance improvements.	Component and machine economies of scale. Fewer trips from port to installation site. Fewer foundations and maintenance trips per unit of installed capacity. Downward pressure on production, installation and O&M costs.
Offshore specific turbine designs	Explicit design for marine installations (i.e., port based assembly and industry specific installation vessels) and operating conditions.	Minimise work at sea while increasing ease of maintenance and accessibility from offshore vessels. Maximise the value of simplified sea transport.
Foundation and support structures	Incremental modifications to existing technology. Development and maturation of technology for deepwater installations.	Minimise foundation costs through mass production, increased standardisation and design refinement. Reduce time to install foundation infrastructure.
Installation techniques and vessels	Mission specific installation vessels and enhanced installation techniques.	Increased installation efficiency, reduced weather risk, lower installation costs.
Grid interconnection infrastructure	Serial production of HV cable, improved DC conversion technology. Enhanced frequency and voltage control, fault ride-through capacity, broader operative ranges.	Reduced cost for grid interconnection, improved wind farm power quality and grid service capacity.
O&M strategy	Enhanced condition-monitoring technology and design-specific improvements. Improved operations strategies.	Real-time, condition monitoring of turbine operating characteristics. Increased availability and more efficient O&M maintenance planning.
Resource assessment	Turbine mounted real-time assessment technology (e.g., LIDAR) linked to advanced controls systems. Enhanced array impacts modeling and turbine siting capacity.	Increased energy capture while reducing fatigue loads, allows for slimmer design margins and reduced component masses; increased plant performance

Furthermore, the materials used to manufacture the blade and their layout may also be optimised to improve the structural dynamics or reduce the weight and therefore, costs. However, the optimisation of individual components may or may not reduce the LCOE depending on trade-offs elsewhere in the system.

To optimise the design of any given component, a computational model needs to be developed and calibrated, to search the design space for improved solutions. It may be infeasible, costly or time-consuming to manufacture and test a real size version of the component at each design iteration. Returning to the previous example, software that predicts the energy yield of a particular wind turbine design can be run thousands of times by an optimisation algorithm to find better-performing designs faster.

1.2. DESIGN OF OFFSHORE WIND FARMS

This section gives an impression of what present and upcoming wind farms look like, and how they are currently designed.

1.2.1. WIND FARMS ARE COMPLEX SYSTEMS

A system may be seen as a set of interconnected components whose individual behaviour and interactions determine the overall performance of the set.

Wind farms are amongst the most complex systems deployed worldwide, based on their uncertainty, heterogeneity and complexity[15]. The power output and costs of wind farms are uncertain as they rely on many stochastic parameters and imperfect models. Wind farms are heterogeneous in the sense that they all look and perform differently, and the design of every new project will face particular constraints perhaps unseen in other wind plants. Moreover, many technical and social disciplines may simultaneously describe the performance of a complex system such as a wind farm, with coupled interactions across subcomponents and disciplines.

A complete description of a wind farm must include the behaviour of the atmosphere and water body (for the offshore case), the air flow inside the wind farm, the terrain in which it is installed, the energy production, collection and transmission to shore, the loads exerted on the turbines and support structures, plant control, balance of plant construction and assembly including foundation structures, the operation and maintenance strategies, the electrical infrastructure and operation, finances and electricity markets, as well as environmental and societal impacts[16]. Figure 1.5 shows a sketch of all these disciplines and components.

Additionally, the level of complexity of a single wind turbine in and of itself is very large, compounding the overall complexity of the system, since disciplines such as aerodynamics, structural dynamics, materials engineering, power and loads control, cost modelling and electromagnetism interact significantly[18].

To give an idea of the scale of offshore wind farms that will become operational this decade, Fig. 1.6 illustrates some facts about the Hornsea Project One. This wind farm is being developed by Ørsted in the North Sea off the eastern coast of the United Kingdom and will become the largest offshore wind farm ever built.

1.2.2. TRADITIONAL DESIGN METHODS

Offshore wind farms are grid-scale projects backed by consortia of multinational companies that typically have governments, electrical system operators, financial institutions and consumers as stakeholders. This structure already hints at the complex decision making process to satisfy the requirements of all parties involved. The partitioned nature of the industry is most evident during the design process, in which technical engineering companies are responsible for different components of the system with limited communication between them[19]. Consequently, the components of offshore wind farms are to a large degree designed sequentially. An example of this engineering practice is for a team to first design the wind turbines independent of the site location. Then layout designers fix the positions of the turbines in such a way that turbines interact through wakes as little as possible with each other, and thus to avoid energy losses to the greatest extent. Once the layout has been decided, then another team goes on to design the support structures for the water depths and soil conditions at which every wind turbine must be installed. Simultaneously, another team designs the topology of the power collection system that must pass through every wind turbine, with the goal of reducing the overall cable length.

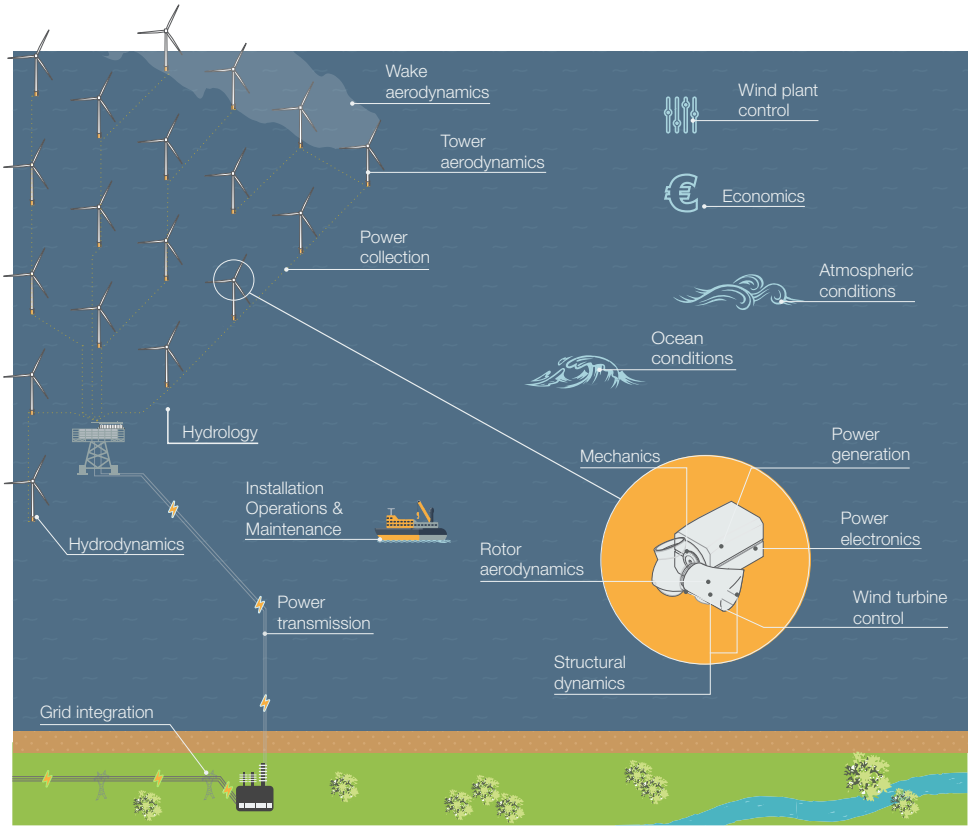


Figure 1.5: Sketch of some components and disciplines that govern an offshore wind farm (graphic elements modified from Bos [17]).

Furthermore, while experience and expertise are and always will be key to making better design decisions and making engineering processes more efficient, design automation is not fully exploited in the offshore wind industry. Design automation is the practice of building a computer simulation model and repeatedly exploring the design space in search of a better design. Design automation augments the innate abilities of the human designer to more thoroughly explore the design space and discover non-intuitive designs that may actually be superior to what the designer could come up with through a manual development process. Some facts that lead to suspect a lack of design automation are that existing offshore wind farms are arranged into regular grid-like layouts and the reported topology of infield collection systems are using longer cables than necessary [20]. Thus, there seems to be little exploration of irregular layouts or cable topology optimisation in utility-scale offshore wind farms. Nonetheless, these decisions might result from other unknown design constraints.

A recurring theme in this thesis is the fact that sequential design is actually detrimental to the overall system cost and performance, and that the lack of automation in

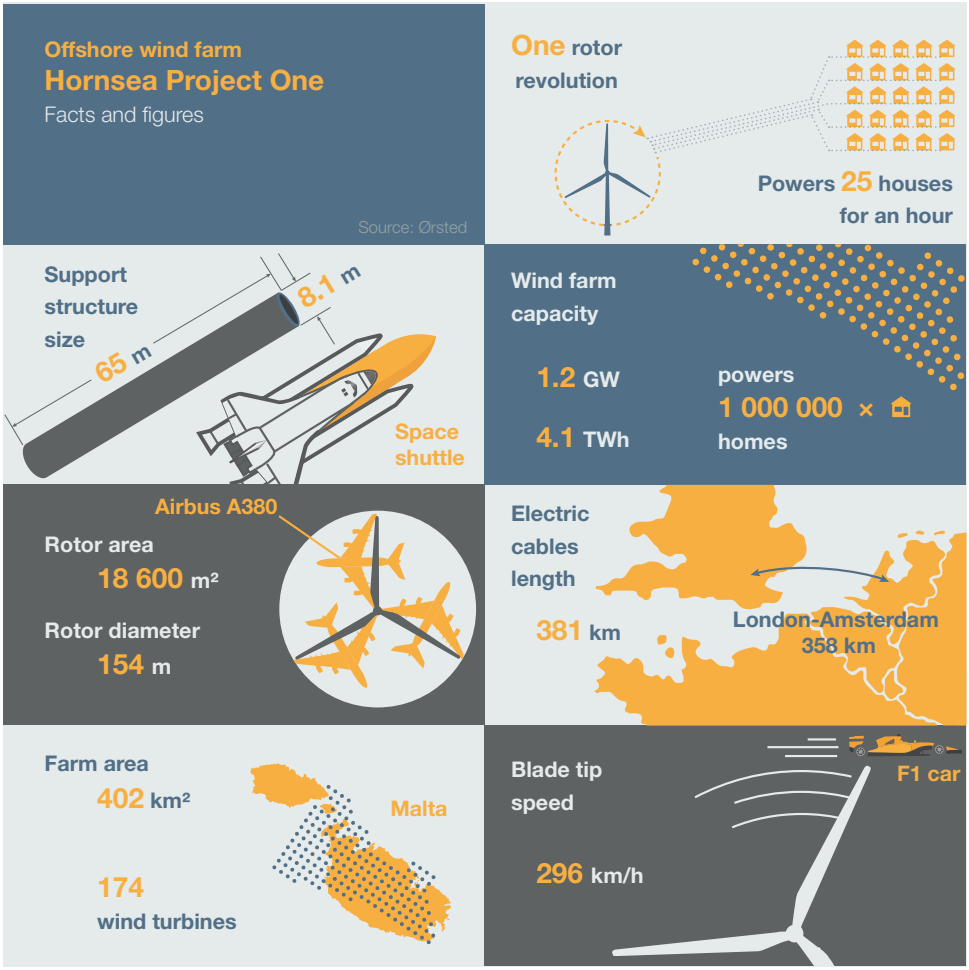


Figure 1.6: Infographic on Ørsted's Hornsea Project One offshore wind farm.

the design process is a missed opportunity for improving the design.

1.2.3. SYSTEMS ENGINEERING FOR WIND ENERGY

Systems engineering is a well established branch of engineering that tackles holistic system design. In contrast to traditional design methods, systems engineering considers the contributions of every component and the advantages or disadvantages of their mutual interactions for the system's performance and cost[21].

Due to their complexity, wind plants qualify as prime beneficiaries of systems engineering methods. The lack of knowledge about how some design decisions affect other aspects of the plant leads to suboptimal designs and higher cost of energy[15]. One common example of a missed opportunity for reducing the levelised cost of energy (LCOE) of an offshore wind farm is to optimise the layout without robust consideration for plant

balance of system costs— including the electrical collection system or the cost of the support structure with varying water depths, among others.

The application of systems engineering relies on a number of methods, of which this work is restricted to one: Multidisciplinary Design Analysis and Optimisation, commonly shortened to MDAO or MDO.

Multidisciplinary Design, Analysis and Optimisation is a technique that deals with the interactions between different components and disciplines of a system. The exploration of interactions is achieved by integrating information obtained from physics-based models, measurements and experts elicitation that belong to a wide range of disciplines, using numerical models. This technique allows the analysis not only of the behaviour of every component and individual discipline, but also of their coupled behaviour[22].

Wind energy researchers, project designers, developers or policy makers will benefit from applying MDAO to solve a myriad of problems. Examples include the optimisation of the LCOE with respect to the types of underwater foundations installed within the wind farm; uncertainty quantification of wind turbine fatigue loads; assessment of the impact of a new generator technology on the performance of the entire wind plant; sensitivity analysis of LCOE with respect to a financial design variable, and design certification with respect to multiple cases[23,24].

An extensive review of MDAO applied to wind turbines was done by Caboni in his work on multidisciplinary robust optimisation[25]. The seminal paper by Dykes et al. explores works in MDAO applied to both wind turbines and wind farms, and sets the foundations of what would later become the MDAO workflow WISDEM (Wind-Plant Integrated System Design and Engineering Model)[15]. Dykes et al. stated that most research was being done on singular components or disciplines, and thus concluded that there were huge opportunities for researching and developing MDAO in the domain of wind energy. Similar observations and conclusions were made by Zaaijer in the context of support structure design[26]. Another MDAO workflow for wind plant layout optimisation is TOPFARM[27], which accounts for electrical infrastructure, foundation costs, fatigue degradation, O&M costs and AEP to maximise the finance balance, and makes use of a gradient-free optimiser for searching the global optimum and then a gradient based optimiser for refining the optimal solution.

More examples of MDAO for wind energy applications are the work by Ashuri et al.[28], where an offshore wind turbine is optimised using multiple disciplines; the rotor nacelle assembly comprehensive design tool by Zaaijer[10]; a multi-level wind turbine design approach that makes use of metamodels by Maki et al.[29]; and the work by Fleming et al.[30], demonstrating that coupling two disciplines (in this case control and wake modeling) decreases the cost of energy more than sequentially optimising layout and control strategy. Another comprehensive wind turbine design workflow is CpMAX[31], which successfully couples high fidelity aerodynamic, structural and control models with nested optimisation algorithms. All of these works report a system level performance improvement through the use of multidisciplinary design, analysis and optimisation. This suggestion is further supported by one conclusion drawn from a review of approaches for wind farm design[8]: “New holistic models are required to improve the wind farm performance modelling and its optimisation. ... optimisation frameworks must encompass all the design variables during the micro-siting process, since current existing approaches have limited

the number of design variables and their degrees of freedom."

1.3. INTRODUCTION TO MDAO

MDAO developed within the aerospace industry, due to the strong influences between diverse disciplines that impact the performance of aircraft. Later, MDAO went on to be successfully applied in the automotive, naval and civil engineering industries, among others[22].

MDAO consists of a workflow where a set of computational tools that represent different components and disciplines are coupled to simulate the entire system. The coupled tools are called an **analysis block**. With this technique, valuable analyses that assist the decision making process during the design of the system can be performed. Additionally, by including **drivers** that control how and when each tool is executed, the workflow can fulfil a certain functionality. The functionality of the workflow is defined by a **use case**. In this context, use case is the term that describes a particular domain problem that can be solved with an MDAO workflow, such as the optimisation of the annual energy yield of a wind farm with respect to its layout.

Figure 1.7 depicts a simplified diagram of an MDAO workflow: an analysis block composed of two computational tools coupled to a driver.

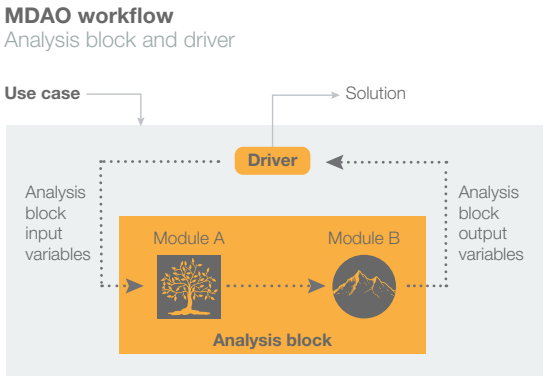


Figure 1.7: Simplified diagram of an MDAO workflow with an analysis block composed of two modules and a driver.

Drivers that are commonly used in an MDAO workflow serve different purposes, *e.g.* to run design of experiments (DOE) or sensitivity analyses, to perform uncertainty quantification (UQ) or to implement optimisation algorithms. DOE drivers systematically analyse a subset of the design space to predict the performance of untested designs. System level uncertainty is quantified by propagating input uncertainties, and thus typically requires the analyses to be performed many times. Finally, optimisation algorithms attempt to find the optimal system design that maximises its performance by smartly exploring the design space.

1

1.4. THE THREE CORE ASPECTS OF MDAO WORKFLOWS

The creation of MDAO workflows has three dimensions of complexity[16], all of which play a key role in their performance: system scope, architecture and model fidelity/driver algorithm. Each of these concepts is described next.

1.4.1. SYSTEM SCOPE

First, the scope of the system that must be included (depicted in Fig. 1.8) needs to be defined before instantiating an MDAO workflow. The reason for this is that not all components and disciplines influence one another with the same strength. Two particular examples of use cases with different system scope in the field of wind energy are the following: in the optimisation of the layout of an offshore wind farm, a workflow will have to include, among others, the calculation of wake losses and cable lengths. On the contrary, in a sensitivity analysis of LCOE with respect to the type of foundation, one does not need to re-analyse the performance or cost of the electrical collection system, as the interaction between them is negligible.

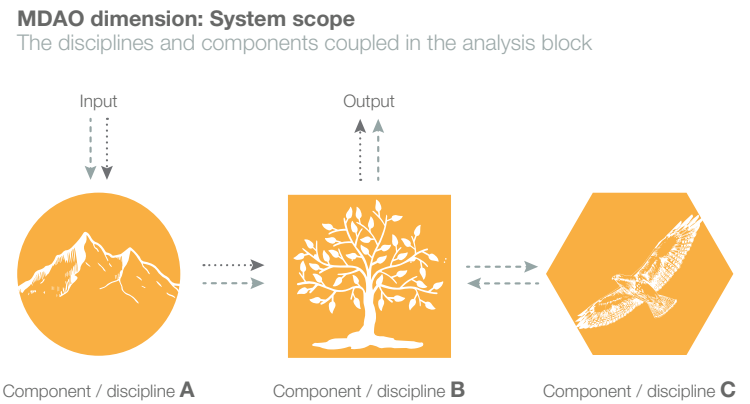


Figure 1.8: Depiction of two workflows with different system scope. One (with dotted arrows) includes components or disciplines A and B, while the other (with dashed arrows) includes A, B and C.

One work that has addressed the complex evaluation of system scope for offshore wind farms is the OWFgraph knowledge database[32]. The goal of this database is to map the real world composed of components and their specifications and behaviour, to the virtual world where they are simulated by models and specified by variables. One strength of this database is that it enables the tracing of dependencies of the performance and cost of any given component of the system, and therefore to establish a causal map that can inform the scope of a system simulation.

1.4.2. MODEL FIDELITY/DRIVER ALGORITHM

Second, the consideration of the fidelity of the models coupled (depicted in Fig. 1.9 with the levels of detail of a tree) is key to ensure that results are representative of reality and to avoid the unnecessary waste of resources. Particular use cases will require simpler or more sophisticated models included in the workflow. Similarly, the consideration of the

algorithm used to drive the analysis block is important to get useful results out of the MDAO workflow.

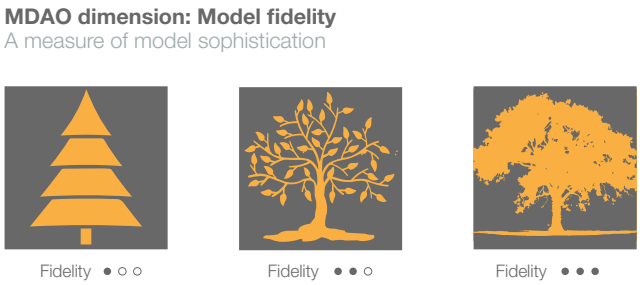


Figure 1.9: Depiction of three levels of model fidelity. The level of sophistication of computational tools is represented by the quality of the representation of a tree.

Two absurd scenarios exemplify the previous statement[23]. The first is a wind farm layout optimisation using the solution of the full Navier-Stokes equation with fully resolved wind turbine blades. The second is the assessment of the 90-percentile of the levelised cost of the energy converted by a wind plant for making an investment decision using lookup tables made with empirical models. In both cases, the choice of tools is rather poor since their best attributes are not fully exploited. In the first scenario, the optimisation would be prohibitively expensive and time consuming due to the consideration of irrelevant details. In the second scenario, the uncertainty resulting from such an unsophisticated model would be unacceptably high for the purpose of financial investment and lead to unreliable decisions. Nevertheless, this modelling approach could be beneficial to the early stage design of the wind plant layout. In other words, the usefulness of any given model fidelity depends on its use case.

With regards to literature, the aspect of model fidelity has been under scrutiny only recently. Some publications touch upon the possibilities of creating multi-fidelity workflows by combining the outputs of sophisticated physics and simple engineering models[27], the exploration of surrogate modelling and model reduction techniques[33], and sensitivity analyses with varying model fidelities[34].

Regarding the choice of driver algorithm, a good example is the wide range of alternatives for optimisers. Some gradient-based algorithms are better suited to continuous, smooth functions with a single global optimum, whereas gradient-free optimisers outperform the former with multi-modal, discrete or discontinuous functions.

1.4.3. MDAO ARCHITECTURE

Third, the architecture of the workflow (depicted with solid and dashed arrows in Fig. 1.10) is also partly responsible for the performance of the MDAO workflow. MDAO architecture is concerned with the coupling between tools and drivers and between tools mutually.

For example, the selection of MDAO architecture has direct impact on the performance of MDAO workflows for wind turbine blade optimisation. The tight coupling between aerodynamics and structural dynamics calls for both modules to depend on each

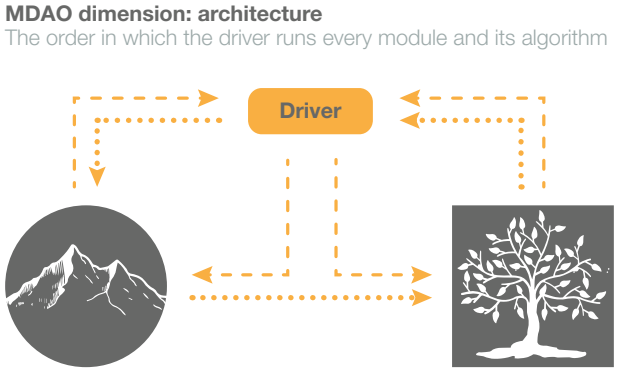


Figure 1.10: Depiction of two workflows with different MDAO architecture. The dashed arrows show the driver calling each discipline individually, whereas the dotted arrows show the driver calling the entire analysis at every iteration.

other. That is to say that there is a feedback loop between both disciplines. While on the one hand the avoidance of aerodynamic drag drives the external shape of the blade to be as slender as possible, on the other hand the internal structure requires thicker blades to withstand aerodynamic loads. Consequently, at every step of the optimisation of the blade's external geometry and internal structure, an iterative solver needs to converge both disciplinary modules to a single design. One MDAO architecture (Multidisciplinary Feasible) would couple the optimisation to the iterative loop between the modules, while another architecture would only run every module once at each step (Individual Discipline Feasible), and include the constraint that the external geometry and internal structure of the blade be feasible until convergence to the optimal design.

The possibility of nesting driver algorithms at different levels of the analysis block, increases the complexity of the problem of choosing the most useful combination and architecture.

Most of the current research on the implementation of MDAO lies on the architecture axis. Specifically, much scrutiny surrounds the subject of MDAO architecture, due to its capacity to improve the speed of an optimisation problem by decoupling the disciplines and let the driver take charge of the coupling instead. Indeed, researchers have developed several architectures, each with its own strengths. The strengths include the reduced time of the optimisation process, the easiness to implement, the ability to yield feasible results at every iteration, and the ability to parallelise the execution of the tools.

Several publications describe and compare different architectures using common variables and reference problems for different engineering fields [22,35–40].

1.5. THE CASE FOR MDAO WORKFLOW SELECTION

In light of the aforementioned three dimensions of complexity of MDAO, it becomes clearer that multiple different workflows can be used for solving the same use case, albeit they will all perform differently.

In particular, it is acknowledged that computational tools of varying levels of fidelity

may be used to simulate and analyse the same wind plant component[24]. Since multiple tools are coupled in an MDAO analysis block, there may be more than one possible combination, *i.e.* different MDAO analysis blocks may exist for the same purpose.

At present, MDAO users usually provide qualitative reasons for the selection of model fidelity and driver algorithms, if at all. Typical arguments are that tools are selected for being the fastest, highest fidelity, in-house built or the only ones available.

It is argued that while their choice can yield acceptable results, MDAO users are missing out on the possibility of improving the performance of their MDAO workflow by not exploring the coupling of other tools with different levels of fidelity. Likewise, some MDAO users choose driver algorithms often based on intuition, without testing its performance and comparing between alternatives[24]. However, MDAO for wind energy applications is a relatively new research space, where there is much to be gained by experimenting with different levels of system scope, model fidelity and MDAO architectures. As the field matures, there will be opportunities for more comparative work and develop recommended practices.

It is not surprising then that some MDAO analysis blocks will outperform others for the same use case.

The previous analysis reveals a problem in the application of MDAO, relevant in the field of wind energy:

There currently exists no systematic and objective methodology for selecting the best-performing MDAO workflow for a given use case.

1.6. RESEARCH OBJECTIVE

Given the problem statement, it is now possible to describe the objective of this research. To define an attainable objective, one of the three dimensions of complexity of MDAO workflows are explored in this work: model fidelity and driver algorithm. The system scope dimension and MDAO architecture are fixed, as the input-output connections between modules are pre-established for all combinations of model fidelities and driver algorithms. One consequence of defining a single use case for this research is that the system scope is fixed. In addition, the MDAO architecture has been a studied aspect of MDAO and it is thus out of the scope of this work. The use case behind this research is the multidisciplinary optimisation of offshore wind farms.

Develop a systematic and objective methodology for selecting the best-performing model fidelities and driver algorithms of an MDAO workflow in the domain of offshore wind farms.

1.7. RESEARCH TASKS

The approach to tackle the research objective is broken down into tasks. Tasks represent the necessary sequential steps envisioned to lead to the achievement of the research objective.

Task 1: Build a tool that allows the creation of multiple MDAO workflow where models of varying levels of fidelity and driver algorithms can be swapped in a plug-and-play

fashion.

A methodology for selecting the best-performing model fidelities coupled in an MDAO workflow requires a tool that can automatically coupled different tools. This tool shall programmatically access a database of models, connect their input and output variables to other models, and execute the MDAO workflow. Such a tool does not exist at present, and the first task is therefore devoted to build it. This task is broken down into five subtasks.

Task 1.1: Define a use case in the field of wind energy.

As mentioned earlier, every use case requires a different MDAO workflow. To make the objective attainable, only one use case is chosen to drive the evaluation and selection of MDAO workflows: the multidisciplinary optimisation of an offshore wind farm. In this task, the specific formulation of the optimisation problem is defined.

Task 1.2: Define the requirements of the tool.

The tool must fulfil what is required of it. This task defined the exact expectations and requirements of the tool.

Task 1.3: Make a conceptual model of the tool.

With the requirements laid out, this task deals with the definition and design of the tool.

Task 1.4: Instantiate the tool.

Once the conceptual model of the tool exists, it can then be implemented.

Task 1.5: Validate and verify the tool.

This task is meant to increase the confidence of the user in the model fidelities available, in the analysis block of one MDAO workflow enabled by the tool, and in the optimisation results using that analysis block and one optimisation algorithm.

Task 2: Develop a guideline for evaluating, comparing and ranking MDAO workflows for a given use case in the field of offshore wind farms.

In this task a methodology for selecting the best-performing combination of model fidelities and optimisation algorithms is developed. The methodology or guideline must provide mechanism to evaluate, compare, and rank different MDAO workflows. This task is further broken down into four subtasks.

Task 2.1: Define the requirements.

The guideline for MDAO workflow selection is meant to comply with functional requirements defined in this task.

Task 2.2: Define a process for determining the governing criteria that evaluate performance.

A methodology for evaluating the performance of every MDAO workflow is the deliverable of this task.

Task 2.3: Define comparison mechanisms.
Once the performance of two or more MDAO workflows is known, then a methodology for comparing them is needed.

Task 2.4: Define ranking mechanisms.
Due to the multiple possible MDAO workflows enabled by having several model fidelities at every module, a methodology is needed to rank all the workflows, using the information found with the methodologies defined in Tasks 2.2 and 2.3. The list of the best and worst performing workflows can be extracted from the resulting ranking.

Task 3: Instantiate the guideline.
The tool built in Task 1 is used to test the guideline by enabling the automatic instantiation of different MDAO workflows in the domain of offshore wind farm design. The model fidelities and optimisation algorithms implemented in the tool are used to provide an example of the implementation of the guideline defined in Task 2. The outcome of this task is the best-performing set of models and optimisation algorithms for the multidisciplinary design of offshore wind farms.

Task 4: Validate and verify the guideline.
To increase the confidence in the guideline and test whether the guideline fulfils the research objective, a validation study of the guideline ensues in this task.

1.8. OUTLINE

Every chapter in this thesis is dedicated to one of the major tasks presented above. Chapter 2 reports the entire process that led to the instantiation of an MDAO workflow for offshore wind energy. Then, Chapter 3 describes the core and main contribution of this research, the guideline for evaluating, comparing and ranking multiple MDAO workflows. Subsequently, Chapter 4 provides an example of the application of the guideline. Chapter 5 reports the validation and verification studies of the guideline. Finally, Chapter 6 draws conclusions and comments on the generalisation and outlook of this research.

Figure 1.11 provides a visualisation of the distribution of information in the chapters of this document.

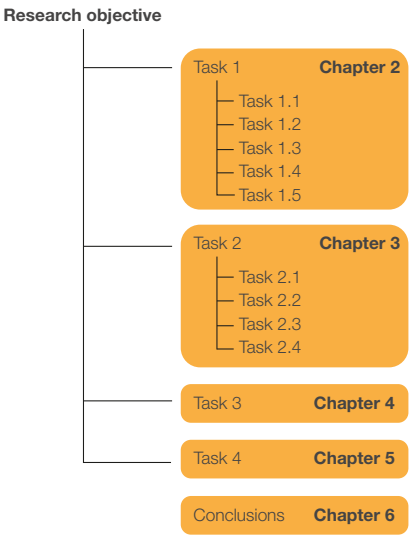


Figure 1.11: Outline of the chapters of this thesis.

Tool for MDAO workflow design

Perfection is achieved,
not when there is nothing more to add,
but when there is nothing left to take away.

Antoine de Saint-Exupéry

Everything should be made
as simple as possible, but not simpler.

Albert Einstein

A tool is built for instantiating MDAO workflows with models of varying level of fidelity and different optimisation algorithms, for testing the guideline for MDAO workflow selection.

The conceptual modelling, implementation, validation and verification of the tool is presented. The use case behind creation of the tool is a multidisciplinary offshore wind farm design optimisation to minimise LCOE with respect to the layout, the design of the electrical infrastructure and the design of the support structures.

The models included with the tool are verified with implementations previously reported in the literature, whereas the analysis block is validated by means of sensitivity analyses and a cost-breakdown comparison.

The optimisation algorithm is validated by comparing the sample evaluated by the optimiser against a random sample of the design space; and by analysing the optimised layouts of a utility-scale offshore wind farm using three distinct objective functions.

2.1. INTRODUCTION

As mentioned in §1.7, the guideline for MDAO workflow selection can be better tested with a tool that instantiates different MDAO workflows by allowing models with varying levels of fidelity and driver algorithms to be exchanged in a plug-and-play fashion.

The Framework for Unified Systems Engineering and Design of Wind Plants (FUSED-Wind) [41], made by the National Renewable Energy Laboratory (NREL) and the Technical University of Denmark (DTU), is currently the only existing tool that enables the instantiation of MDAO workflows with interchangeable models for wind farm design. However, FUSED-Wind could not be used for the purpose of this work because it agglomerates the balance of station costs in a single module, a feature that is less flexible to accommodate the cost models of particular system components.

The further desire for the analysis block of the tool to include specific disciplines and input-output connections between them, drove the development of a new tool from the ground up. The tool is developed for a wind plant multidisciplinary optimisation use case.

2.2. OVERVIEW AND TERMINOLOGY

The tool for MDAO workflow instantiation, referred to in this chapter as *the tool*, comprises two layers of functionality. The outer layer is meant for users (referred to as *workflow designers*) interested in designing MDAO workflows according to pre-established criteria. The inner layer is meant for users interested in designing offshore wind farms (referred to as *wind farm designers*). Each layer of the tool is sketched in Fig. 2.1. Both layers interact when the outer layer communicates to the inner layer the models and driver algorithm that are to be coupled. The instantiation of an MDAO workflow by the outer layer is a necessary prior step for the inner layer to be executable.

Workflow designers require information from the wind farm designers. In particular, the workflow designer instantiates the best-performing MDAO workflow for a given use case, defined previously by the wind farm designer.

The outer layer of the tool requests from the workflow designer the models and driver algorithms to be coupled, instantiates and, if necessary, executes the MDAO workflow. The outer layer is meant to support the evaluation of the performance of different MDAO workflows by the workflow designer.

The inner layer is an MDAO workflow that requests from the wind farm designer the site conditions and fixed design parameters, and its output is the solution to the use case.

The tool is built to fulfil certain requirements, and these are listed below per layer.

Outer layer: This layer must allow the workflow designer to interchange the models that analyse every system component and the optimisation algorithm, and instantiate and execute the MDAO workflow for workflow-evaluation purposes. Fig. 2.2 shows a diagram of this requirement.

Inner layer: This layer is a means for testing the functionality of the outer layer, and thus it does not have to be readily suitable for deployment in industry practice. Therefore, it must have the ability to capture realistic changes in the LCOE with respect to

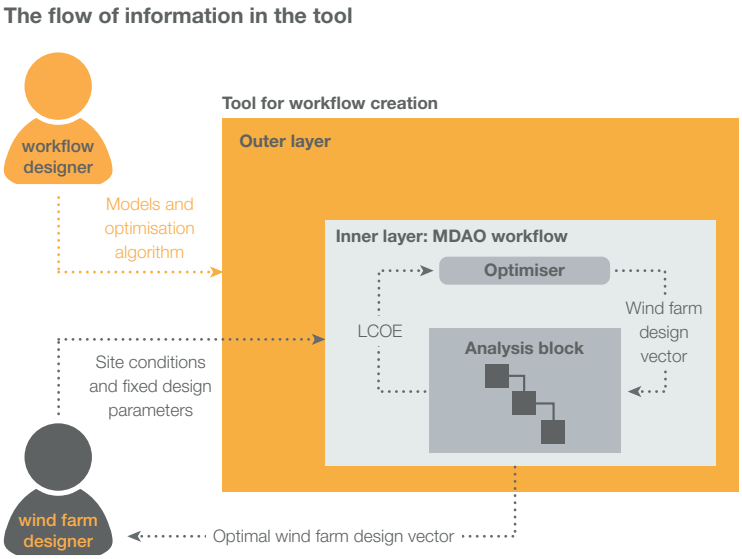


Figure 2.1: Sketch of the layers of the tool, the users of each layer, and their inputs and outputs.

variations in the the design variables of the farm layout, electrical infrastructure and support structures, rather than the ability to yield realistic accurate absolute numbers. These realistic trends should then drive an optimisation algorithm towards the minimum levelised cost of energy (LCOE) of an offshore wind plant.

To avoid confusion, a list with the terminology used throughout this work is listed below:

1. **Discipline:** Technical or scientific branch of knowledge that governs one aspect of a phenomenon or component. Example: wake aerodynamics.

2. **Component:** Specialised piece of equipment or hardware of the system (offshore wind plant). Example: electrical infrastructure.

3. **Module:** Separable analysis of the MDAO analysis block. Modules may analyse the performance or cost of an entire component of the system, or analyse a component according to a single technical discipline. Example: analysis of the array efficiency.

4. **Model:** Computational tool that implements a mathematical model for analysing a component of the system or discipline. Models may have different levels of fidelity. Example: a specific implementation of the Jensen wake model.

The following sections describe the resulting tool.

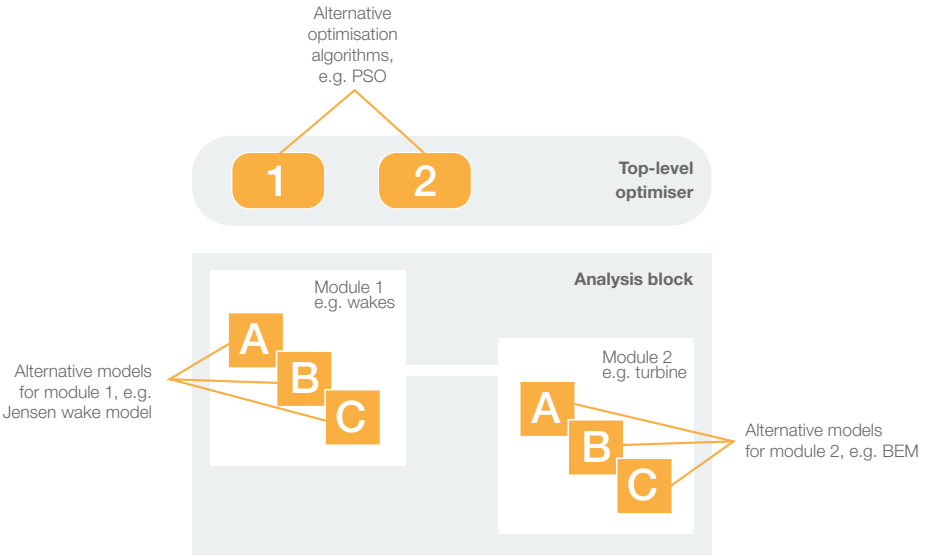


Figure 2.2: Different models can be used at every module. Likewise, different optimisation algorithms can be coupled to the analysis block.

2.3. CONCEPTUAL DESIGN OF THE TOOL

The outer layer of the tool consists of a function that communicates the list of models and optimisation algorithms to be plugged into an *empty* MDAO workflow in the inner layer made specifically for the wind farm designer's use case. The term *empty* is meant to reflect that the MDAO workflow is initially void of any models that perform the analyses. However, the inputs and outputs of modules and the connections between them are included.

The ability to interchangeably accommodate user-defined models and optimisation algorithms is best served by modularity. The analysis block of the MDAO workflows built with the tool has clearly defined boundaries between independent modules that wrap individual models and provide the input/output connections between them.

The ability of the inner layer to minimise the LCOE of a wind farm design is met with a top level optimiser. Its input is the LCOE and its output is a multidisciplinary design vector. The optimisation problem is formulated as follows:

minimise:	LCOE
with respect to:	wind farm layout,
	design of the electrical infrastructure,
	design of the support structures.

As mentioned in §1.6, a single use case defines the inputs, outputs and connections of the modules of the *empty* MDAO workflow. This use case is the following:

What is the optimal wind farm layout, number of wind turbines, design of the electrical infrastructure and design of the support structures that jointly minimise the levelised cost of energy at a given wind site, in a preliminary design phase?

The inner layer includes a descriptive analysis that estimates the LCOE of a wind plant design at a given site. This analysis forms the analysis block of the empty MDAO workflow, and comprises multiple modules coupled via their inputs and outputs.

As mentioned earlier in §1.4, the complexity in creating an MDAO workflow covers three dimensions (see Figs. 1.8, 1.9 and 1.10): system scope, model fidelity/driver algorithm, and MDAO architecture [16]. Figure 2.3 shows an abstraction of the three-dimensional space that the tool allows the workflow designer to explore.

The MDAO space that can be explored by the tool.

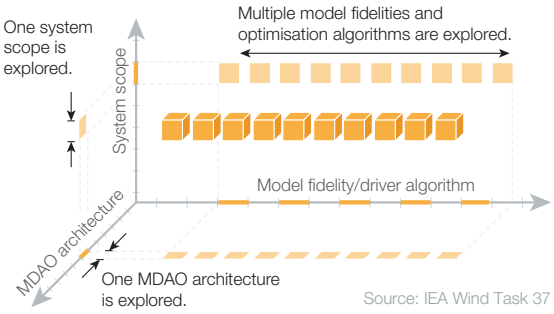


Figure 2.3: Abstract sketch of the three-dimensional space that can be explored by the workflow designer using the tool.

The description of the MDAO workflow along each of these dimensions is presented next.

2.3.1. SYSTEM SCOPE

The system scope of the analysis block is fixed. The choice for system scope is a consequence of the use case set by the wind farm designer, and is described below.

Although the consideration of the social and environmental aspects of wind energy are of utmost importance in its development, the MDAO workflows created by the tool focus exclusively on technical disciplines.

Figure 2.4 illustrates the six major cost contributions to the LCOE of a large offshore wind plant.

In spite of there being many components that impact the LCOE, the support structures and electrical infrastructure are the most dependent cost-wise on the wind farm layout. Additionally, wind plants are built to a large extent with off-the-shelf wind turbines, and manufacturing processes and facilities are not deemed to tailor wind turbine designs for individual wind energy projects or sites. For this reason, a single wind turbine is used to design the balance of station (BOS).

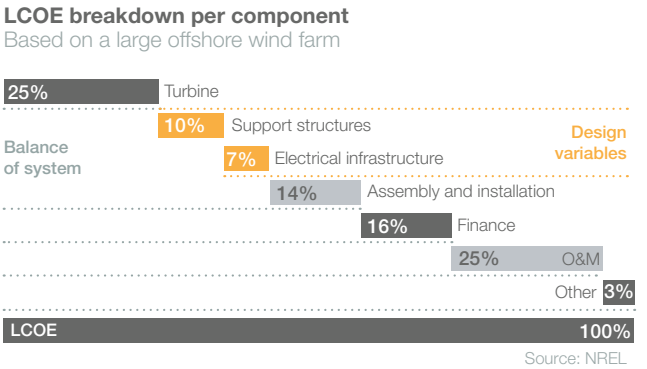


Figure 2.4: Contributions to LCOE per subcomponent (modified from [9]).

The design of the assembly, installation, operations and maintenance strategies and logistics are out of the scope of the use case. However, there are reports where these have partially been co-optimised with the rest of the BOS [10].

Moreover, the additional costs incurred from financial considerations are typically set by shareholders and not designed. The effect of design decisions on finance is neglected in this work.

The breakdown in Fig. 2.4 shows that the design of the electrical infrastructure and the design of the support structures affect 17% of the LCOE. In addition, looking at Eq. 1.1, it is expected that increasing the AEP leads to a decrease in the LCOE as well. This is not apparent in the LCOE breakdown shown above, because the AEP has been fixed.

The opportunities for maximising AEP and minimising the costs of the electrical infrastructure and the support structures are further explained next.

AEP The AEP of a wind plant is a function of the wind conditions at the site, and the power that can be extracted by the wind turbines. Whereas the former is an external, unmodifiable condition, the latter can be improved by the wind farm designer.

The power extracted by a wind farm is strongly influenced by the relative positions of the wind turbines: the wind farm layout. Since wind turbines generate a wake behind them where the wind has less kinetic energy, turbines placed inside the wakes of upstream turbines generate less power than those situated outside the wakes. These array power losses can be minimised if the layout is smartly designed taking into consideration the wind conditions.

The layout is perhaps the single most studied design variable of a wind farm. The optimisation of a wind farm layout to maximise power production, while satisfying multiple constraints is a complex task, and is one of the reasons why it has attracted much research. Layout optimisation alone can return improvements in AEP of up to 15% depending on the baseline design [42].

Electrical infrastructure The electrical infrastructure of a wind plant comprises the collection and transmission to shore systems, and their combined cost of procurement and installation amounts to 9-24% of the capital expenditure (CAPEX)[9,43,44]. The collection system is responsible for gathering the electrical energy generated by the individual turbines at the offshore substations, where the voltage is increased for long distance transmission, using transfomers. The transmission system transfers the electrical energy from the offshore substation to an onshore substation connected to the local grid. Multiple studies have addressed the optimal configuration of the collection system[20,45,46]. Other studies have analysed the cost of alternating and direct current transmission-to-shore systems[45,47]. Costs can be cut by optimising the collection cable topology, cable ratings, transformer ratio, substation foundation, voltage ratings and installation logistics. Important is to note that besides cost, electrical power losses should also be minimised by design.

The wind farm layout strongly influences the cost of the electrical infrastructure through cable length.

Support structures Installation and procurement of support structures contribute on average to 14-22% of the CAPEX[9,43,44]. While several types of foundations exist, roughly 90% of all offshore wind farms use monopiles. Costs of monopiles can be reduced by designing them specifically for the local water depth and metocean and soil conditions at every wind turbine location[10,48], and by optimising the logistics in their installation and transportation. Similarly to the electrical infrastructure, the wind farm layout determines to some extent the cost of the support structures due to the local bathymetry.

Assumptions are made for simplifying the analysis block of the MDAO workflow. The estimation of the LCOE of an offshore wind farm does not include a detailed assessment of manufacturing costs, onshore transportation, installation, operations, maintenance and decommissioning costs and logistics, degradation and failures of components, fatigue of support structures, electrical power losses, wind turbine loads and unsteady aero- and hydrodynamics, as well as behaviour of the energy market, risk and asset management. These limitations impact the accuracy of LCOE, and potential couplings involving these disciplines are ignored.

2.3.2. MDAO ARCHITECTURE

The architecture of the MDAO workflow is fixed, as this dimension is outside the scope of the research objective of this work (see §1.6).

Figure 2.5 shows the MDAO architecture chosen for the MDAO workflow of the tool.

Stemming from the fact that AEP, cost of the electrical infrastructure and cost of the support structure are all competing values to improve the LCOE, a closer look at how they depend on the design variables informs the choice of MDAO architecture.

The design variables of the electrical infrastructure that define the cable topology or offshore substations are assumed to be local design variables, as they mainly affect the cost of that component, and have little influence on either AEP or the cost of the support structures. In contrast, the remaining design variable, the wind farm layout, is global, because it affects AEP and support structures costs, too.

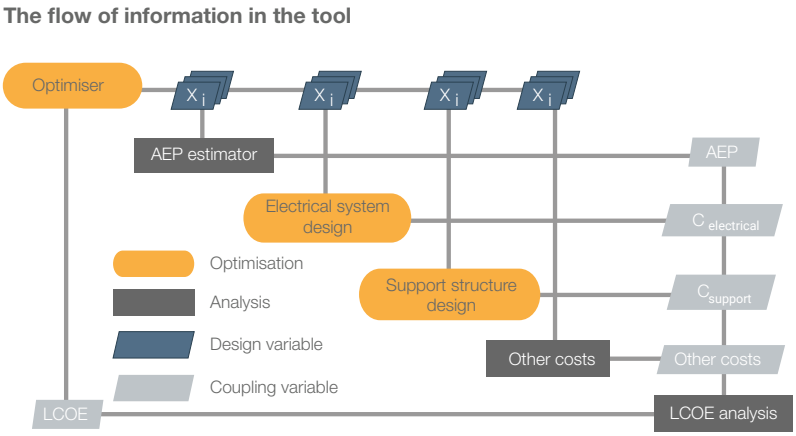


Figure 2.5: MDAO architecture of the tool.

Similarly, except tower height, which is fixed, the variables that describe the support structures, such as the geometry of the monopile and the turbine tower, penetration depth and scour protection, are local design variables, as they have no effect on the cost of the electrical infrastructure or AEP. The layout, once again, is the only global design variable of this component.

AEP is only affected by the wind farm layout, as the design of the wind turbine is outside the scope of the use case.

To summarise, the wind farm layout (shown in Fig. 2.5) as connecting variable X_i) is the only global design variable common to all three modules, whereas the rest of the design variables only affect a single component. This separation of design variables informs that a top level optimiser can drive the layout to minimise the LCOE, while the electrical infrastructure and support structures are driven by a nested optimiser at every iteration of the top level optimiser. As identified earlier in §2.3.1, there are other costs that contribute to the LCOE, but they are assumed to be independent of the design variables.

2.3.3. MODEL FIDELITY/DRIVER ALGORITHM

The level of fidelity of the models and algorithm of the driver is the only dimension explored in this research. However, the model fidelities investigated are constrained, as the input-output connections are fixed by the system scope, and thus some model fidelities are not readily compatible with the MDAO workflow. This limited subset of model fidelities can be used to demonstrate the process of selecting the best-performing MDAO workflows.

Wind plants will typically undergo a preliminary- and a detailed-design phase. During the first phase certain design decisions are made based on feasibility studies relying mostly on unsophisticated computational tools. Design variables such as number of wind turbines, layout, topology of infield cables, number of offshore substations, wind turbine power rating, etc. are decided and fixed to a certain degree at this stage, as they

have influence on power production and overall costs.

In contrast, during the second phase the previously fixed design variables are revisited with more sophisticated computational tools to reduce the uncertainty in their performance and cost. The other sets of design variables that have less influence on power production or costs are decided upon in the detailed design phase, and are instead driven by manufacturing or physical constraints. Furthermore, the second phase consists of the detailed redesign of all system components under new constraints. For example, the design of the support structures for load-bearing capacity, wind farm layout considering unexploded ordnances or existing infrastructure, as well as the design of installation, maintenance and transportation logistics.

Since the use case explicitly relates to a preliminary design stage, the constraints of the detailed design phase are neglected.

The constraints presented above suggest that the models compatible with the MDAO workflow must be fast, as multiple scenarios are to be analysed in this design stage. However, speed will usually imply a poorer accuracy. This loss of accuracy is justified as final investment decisions are not made until after the preliminary design stage. The set of tools that comply with this requirement are engineering models that either make assumptions to simplify the physics, or make use of empirical rules. In addition, faster models are typically deterministic. Although the quantification of uncertainty using probabilistic models is valuable for robust optimisation, it is not the norm at preliminary design stages and is outside the scope of this work.

2.4. EXTENDED DESIGN STRUCTURE MATRIX

Figure 2.5 is expanded to include all the modules of the MDAO workflow and the connections between them. This information is formalised into an extended design structure matrix (XDSM) [22]. Figure 2.6 shows the XDSM of the MDAO workflow for the optimisation of the layout, electrical infrastructure and support structures. The local input parameters to every module are omitted from the XDSM. The modules and connections of the XDSM are explained in the following sections.

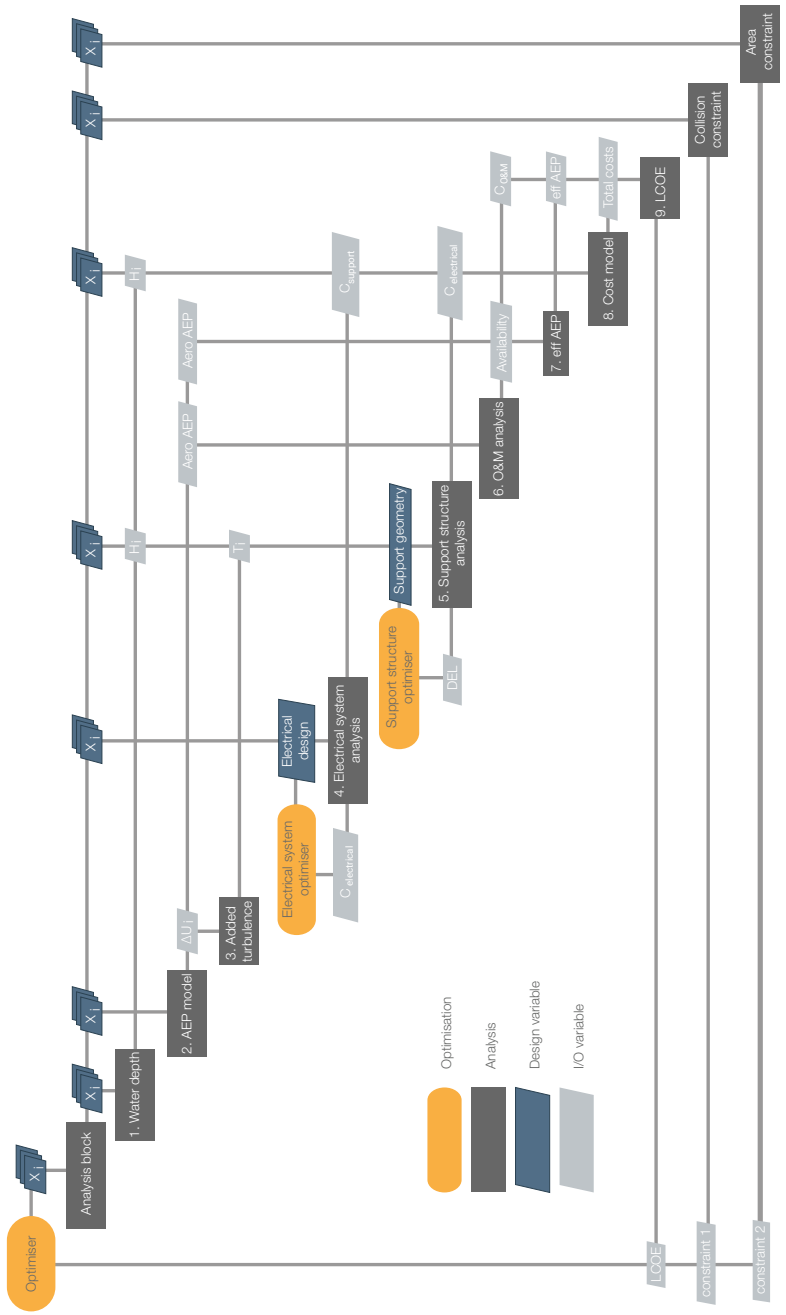


Figure 2.6: XDSM of the MDAO workflow in the tool.

2.5. MODULES

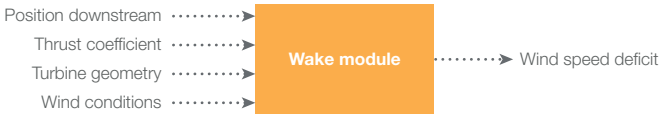
The modules coupled into the MDAO workflow of the tool to estimate the LCOE of an offshore wind plant are listed below and explained from physical principles and referred to the XDSM in Fig. 2.6 by their numbers. Section 2.7 describes the actual models or tools provided for every module.

1. **Water depth:** This module requests a bathymetry file where water depth is defined for a list of x and y coordinates, and outputs the interpolated water depth at any other position.

2. **AEP model:** The module to estimate AEP is composed of four sub-modules: wind turbine performance, wake wind speed deficit, wake merging and an energy convolution module. Each of these is further explained next.

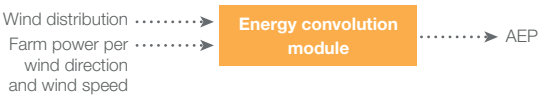
   (a) **Wind turbine performance:** Modelling the conversion of the kinetic energy in the wind into electrical energy requires a wind turbine simulation module that requests the local wind conditions and outputs electrical power converted and aerodynamic thrust.

   (b) **Wake wind speed deficit:** Because wind turbines extract energy from the upstream flow, the wind is on average slowed down in the downstream wake of the turbine, affecting turbines that lie in its wake. The linear momentum the turbine extracts from the wind is a function of the thrust force exerted by the wind on the turbine. The dimensionless thrust coefficient is thus used to characterise the wind turbine. The wind turbine rotor diameter affects the diameter of the wake. Lastly, the ambient turbulence intensity influences the mixing of the wake with the freestream unperturbed flow. Most engineering wake models require a subset of the upstream wind speed, thrust coefficient of the turbine, turbulence intensity, rotor diameter and downstream location as input. Their output is the wind velocity deficit in the wake at the downstream location.

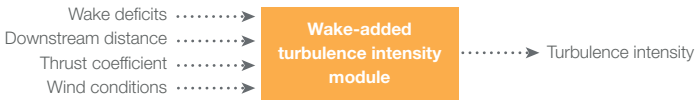


   (c) **Wake merging:** As multiple wakes may converge at the same place in space, their deficits are mathematically combined using a wake merging model. Its input is the multiple wind speed deficits and the output is the local wind speed at that location.

(d) **Energy convolution:** The annual energy production is obtained by combining the information in the windrose and Weibull wind speed distributions for a particular site, with the total power produced per wind direction and wind speed.



3. **Wake-added turbulence:** The added turbulence intensity in the wake is calculated by another module. The input are the wake deficits that all turbines induce in each other, as calculated by the wake model, and the output is the maximum local turbulence intensity at every wind turbine. The wake deficits are used to determine the closest turbine which is assumed to be the only one contributing an added turbulence intensity[49].



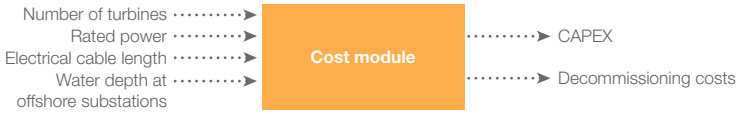
4. **Electrical system design:** The cost of the electrical infrastructure is a function of the plant layout because turbines spaced further apart demand longer electrical collection cables. A module is integrated that minimises the cost of the electrical collection system for a given layout with respect to its topology. The inputs to this module are the number and position of offshore transformer substations, the capacities and costs of up to three electrical collection cable types, and the wind farm layout.

5. **Support structure design:** The water depth at every turbine location determines the cost of the support structure. A module is integrated that minimises the cost of the support structure for a given water depth and local wind turbulence. The inputs to this module are the local metocean conditions, geometry of the wind turbine and water depth. This workflow considers only bottom-fixed monopiles.

6. **O&M analysis:** The cost of operations and maintenance is calculated in this work as directly proportional to the nominal power rating of the wind plant. Besides cost, this module outputs the availability.

7. **Effective AEP:** The AEP calculated so far does not account for electrical losses in the electrical infrastructure nor availability issues due to failures or planned maintenance works. These efficiencies are factored in in this module, and are inputs to the analysis block.

8. **Cost model:** Additional integrated modules calculate the cost of engineering, management, procurement, installation, and decommissioning of the entire wind plant. Some of the costs are assumed constant whereas others are a function of the nominal capacity of the plant.



9. **LCOE:** A last component aggregates the effective AEP, all the costs, the economic lifetime of the project and the annual real interest rate into the LCOE using Eq. 1.1.

2.6. I/O CONNECTIONS BETWEEN MODULES

The connections between the modules of the analysis block presented above are listed in this section.

1. Wind farm layout to AEP module. The wake models require the relative positions between the turbines, which are imposed by the layout. The layout is the output of the top-level optimiser.

2. Wind farm layout to electrical collection topology optimiser module. The topology of the electrical collection cables is optimised for the given position of the wind turbines.

3. Wind farm layout to support structure sizing module. Since the geometry of the support structure is a function of water depth, it needs to be given at every position of the layout.

4. Wind speed deficits matrix from AEP model to wake added turbulence intensity module. The wind speed deficits inform which is the upstream turbine affecting each downstream turbine the most. Only the wake of this specific upstream turbine contributes to added turbulence intensity.

5. Turbulence intensity (TI) from wake added TI module to the support structure sizing module. TI is used to account for fatigue damage due to turbulence in the wind.

6. AEP to O&M. The simple O&M module estimates annual costs from AEP before wake losses.

7. AEP from AEP module to the effective AEP module. This connection ensures the AEP is combined with the electrical efficiencies and availability.

8. Costs of support structures, electrical infrastructure from their respective modules to a cost-aggregating module.

9. Costs and AEP from their respective modules to determine the LCOE in the LCOE module.

10. LCOE from the LCOE module to the top level optimiser. This connection closes the MDAO loop by providing the objective function evaluation to the optimiser.

2.7. MODELS AND OPTIMISATION ALGORITHMS INCLUDED

The following sections provide a list of the models available at every module and a comparison of their output.

2.7.1. AEP MODEL

The AEP model used integrates the power produced by the farm for a sample of wind speeds and wind directions. The number of samples of the Weibull distribution of wind speeds can be adjusted by the wind farm designer to change this model's fidelity. Similarly, the wind farm designer can select the number of wind directions to be analysed for power production. Probabilities are obtained from the wind rose provided.

2.7.2. WIND TURBINE POWER AND THRUST MODELS

Since the wake models adapted in the MDAO workflow simulate the steady state of the plant, the following rotor aerodynamics simulation tools were used to create power and

thrust coefficient look-up tables as functions of upstream wind speed at hub height. Figure 2.7 shows the power curve of the NREL 5 MW reference wind turbine[50] calculated with each of these tools, and Fig. 2.8 shows the thrust coefficient curves.

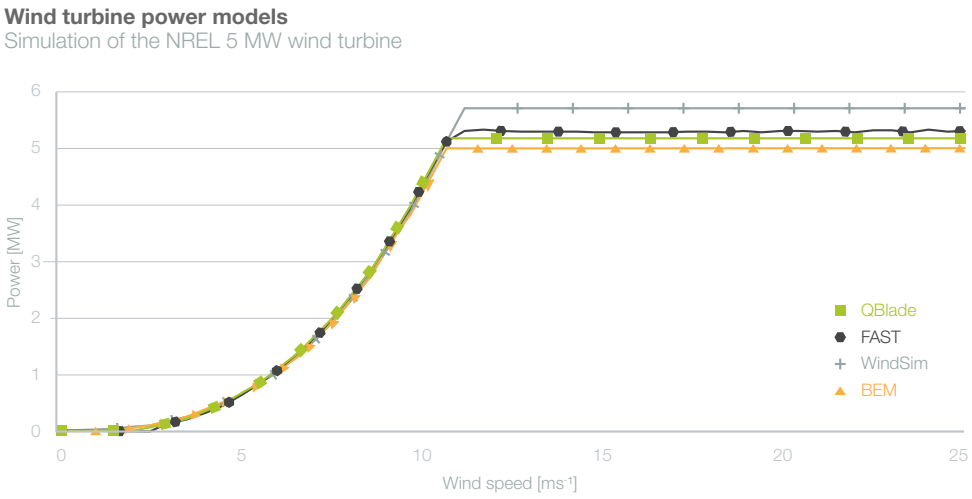


Figure 2.7: Plot of the power curves made with different turbine rotor simulation tools.

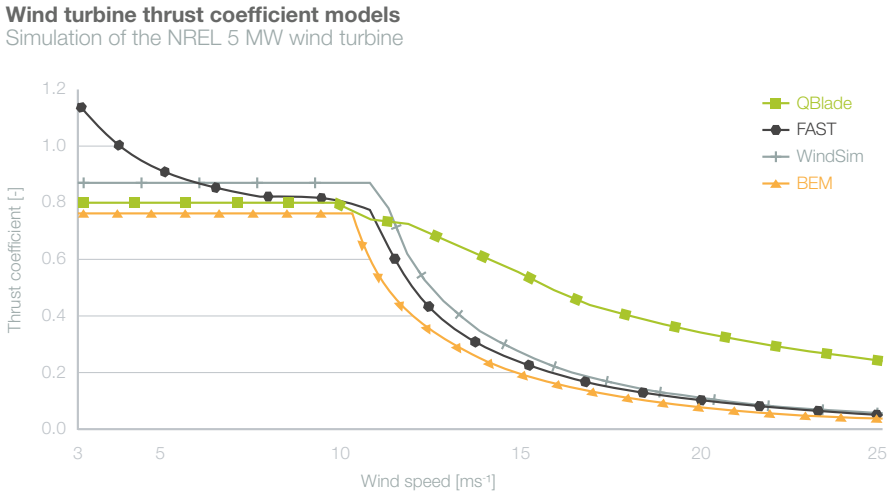


Figure 2.8: Plot of the thrust coefficient curves made with different turbine rotor simulation tools.

FAST

FAST is an open source time-based aero-hydro-servo-elastic wind turbine simulator developed by NREL[51]. The aerodynamic calculation of power and thrust are done with

an implementation of the Blade Element Momentum (BEM) model that includes three-dimensional and tip correction factors. The turbine is coupled to a monopile in an offshore environment. To create lookup tables of power and thrust coefficient, FAST is run with a steady uniform inflow field for 300 s to avoid any transient responses and reach a steady power output.

QBLADE

TU Berlin's QBlade calculates the aerodynamic performance of a rotor using a time-based lifting line model[52]. The lookup table is generated by running QBlade with steady uniform wind fields for 300 s to ensure the full development of the wake and a steady operation.

WINDSIM

A simple BEM model and a seven-degrees-of-freedom dynamic module that evaluates forces at equilibrium are included in the WindSim tool developed by TU Delft[53]. The lookup table is generated by running WindSim at the operating wind speeds.

BEM

This model is simple in that it only takes into consideration the aerodynamic modelling of the rotor by means of a BEM model. The implementation corresponds to that reported by Tanmay[54], following the model description by Manwell et al.[55].

2.7.3. WAKE MODELS

Four engineering wake models are available in the MDAO workflow. A downstream wind velocity field of a single wind turbine according to the different wake models is plotted in Fig. 2.9.

JENSEN

The Jensen empirical wake model[56] is the most commonly used for wind plant design due to its simplicity. It is based on the law of conservation of mass, and assumes a linear expansion of the wake radius and uniform wind speed in the radial direction. Figure 2.9 shows the downstream field of a wake predicted with the Jensen model and a wake decay factor of 0.04 suitable for offshore applications.

LARSEN

The Larsen wake model is an analytical model obtained from Prandtl's turbulent boundary layer equations. These equations are an asymptotic version of the Navier-Stokes equations for large Reynolds numbers[57]. Figure 2.9 shows the downstream field of a wake predicted with the Larsen model. The finite radius of the wake is given by an analytical expression.

AINSLIE 2D EDDY VISCOSITY

The Ainslie or eddy viscosity wake model[58] is a field model that results from solving a thin shear layer approximation of the Navier-Stokes equations, and neglecting viscous terms, together with a Reynolds stress definition of the turbulence viscosity, and the

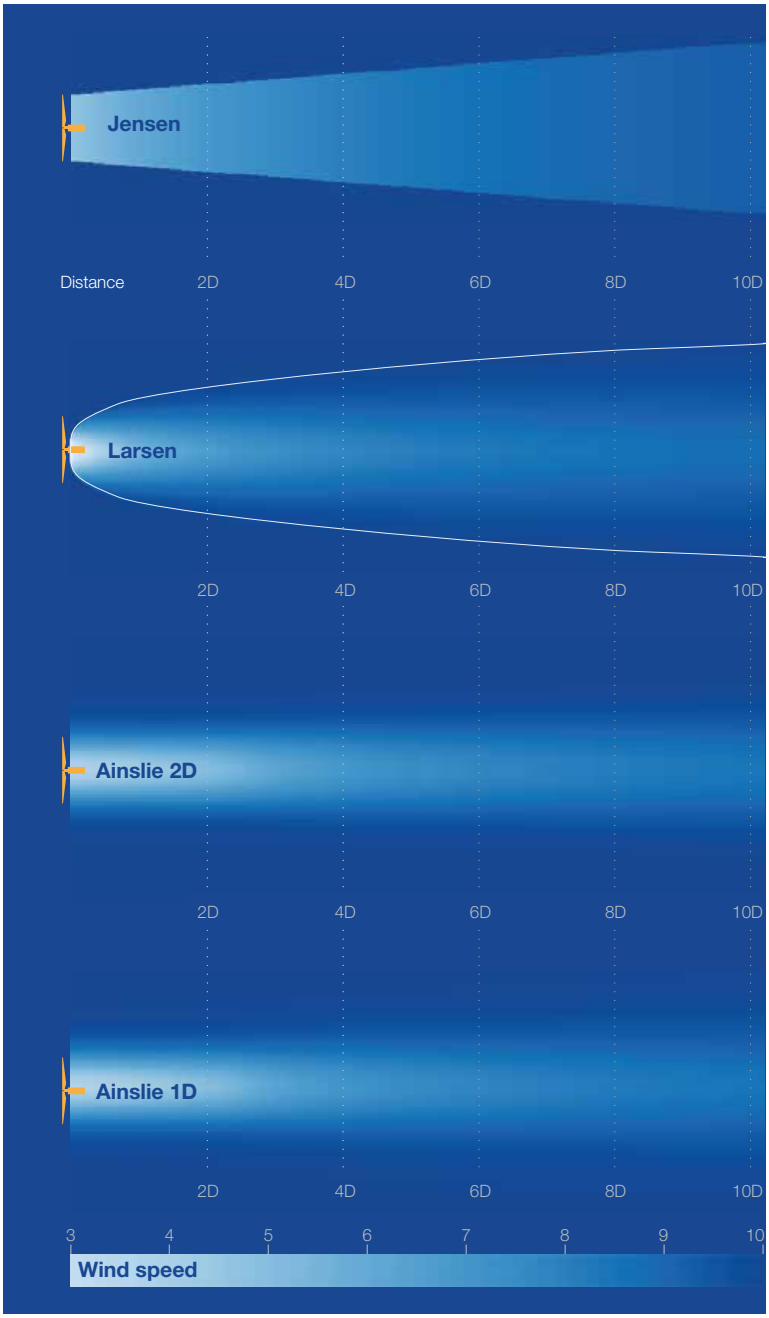


Figure 2.9: Average wind velocity in the downstream field of a wind turbine according to different wake models. The turbine rotor has a diameter of 126 m, a thrust coefficient of 0.79, the freestream wind velocity is 10 ms^{-1} and the ambient turbulence intensity is 8%.

continuity equation. Figure 2.9 shows the downstream field of a wake predicted with the Ainslie 2D model. The axial and radial directions are discretised into 100 elements. To improve the model's efficiency, the wind speed in the radial direction is solved only up to a distance equal to twice the rotor diameter, beyond which freestream wind speed is assumed. The differential equations are solved with an Euler forward numerical scheme.

AINSLIE 1D EDDY VISCOSITY

Anderson[59] reduced the order of the Ainslie model to one dimension, solving only the wake centreline velocity, by noting that the velocity in the radial direction is self-similar in the axial direction and follows a Gaussian profile, which was determined empirically. Figure 2.9 shows the downstream field of a wake predicted with the Ainslie 1D model.

2.7.4. WAKE MERGING MODELS

Four mathematical expressions for merging wake velocity deficits are included in the tool[60–62]. These are listed next:

Root sum square	$\Delta u_i^2 = \sum_j \Delta u_{ij}^2,$
Maximum	$\Delta u_i = max(\Delta u_{ij}),$
Summed	$\Delta u_i = \sum_j \Delta u_{ij},$
Multiplied	$\Delta u_i = \prod_j \Delta u_{ij},$

where Δu_i is the wind speed deficit with respect to the freestream velocity at turbine i, and Δu_{ij} is the deficit at turbine i due to the upstream turbine j.

2.7.5. WAKE ADDED TURBULENCE MODELS

The MDAO tool includes five models to calculate the wake added turbulence intensity at a downstream turbine due only to the upstream turbine that is closest in distance to it[49]. Figure 2.10 shows a plot of the local turbulence intensity (TI) as a function of the normalised downstream distance, calculated with all the models described below.

LARSEN

The Larsen model[63] provides an empirical expression to characterise the added wake turbulence, as a function of spacing and the thrust coefficient of the upstream turbine.

DANISH RECOMMENDATION

The Danish recommendation model[64] is an empirical model in which the added turbulence intensity in the wake results from multiplying two correction factors related to wind speed and wind farm configuration[61]. This model is a basic model that ignores the thrust coefficient of the upstream turbine, and only considers spacing and upstream wind speed.

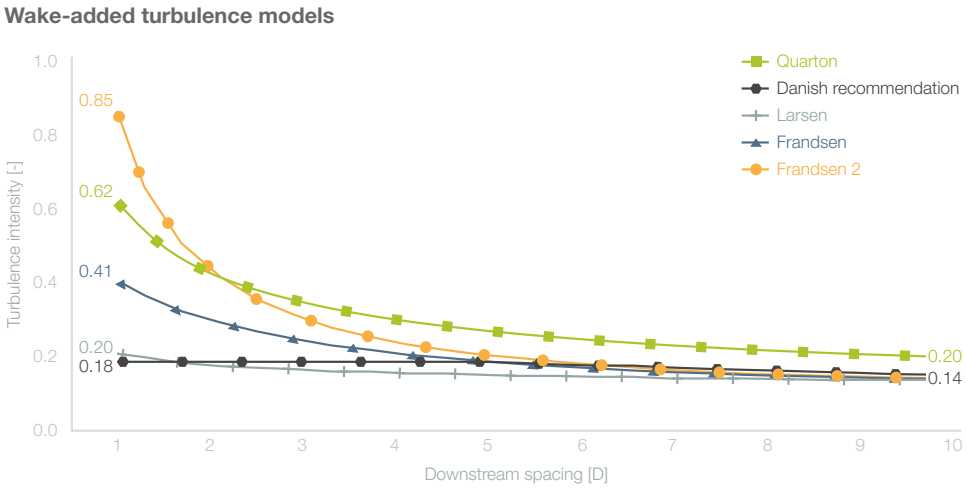


Figure 2.10: Plot of the total turbulence intensity as a function of downstream distance using different wake-added turbulence models.

QUARTON

The Quarton model[65] is derived from measurements, and it attempts to model the added turbulence in the near- and far-wake. It is a function of the ambient turbulence intensity, the spacing, the thrust coefficient, number of blades and the tip speed ratio of the upstream wind turbine.

FRANDSEN

The Frandsen model[66,67] provides two empirical expressions to characterise the added wake turbulence, as a function of spacing, upstream wind speed, ambient turbulence intensity and the thrust coefficient of the upstream turbine. It has one expression for closely spaced turbines in the crosswind direction, and another for large spacings in all directions.

FRANDSEN 2

A modified expression for the Frandsen model[68] is used in the European Wind Turbine Standards[69]. This model is a function of the spacing, ambient turbulence intensity and the thrust coefficient of the upstream turbine. The output of this model diverges at spacings smaller than one rotor diameter.

2.7.6. ELECTRICAL COLLECTION

Two methods for designing the topology of the electrical collection cables are included in the tool. One utilises the Esau-Williams heuristic to generate a branched topology, whereas the other is the Planar Open Savings heuristic that generates a radial topology. In both algorithms up to three cable capacities can be used and cable crossings are forbidden. The topologies and cable lengths of a wind farm with 74 turbines and three offshore substations are shown in Fig. 2.11. While the electrical cable lengths are similar

in this case, the Esau-Williams heuristic yields lower cable lengths than the Planar Open Savings heuristic when a single offshore substation is centrally located. The inputs to both of these heuristics are listed in Table 2.1.

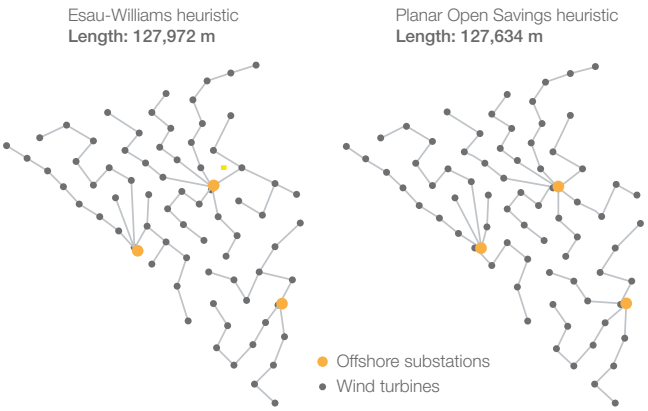


Figure 2.11: Electrical collection cable topologies made with the Esau-Williams heuristic and the Planar Open Savings heuristic. The total lengths of both topologies are also shown.

Table 2.1: Inputs to the Esau-Williams and Planar Open Savings heuristics for designing the electrical collection system.

Variable	Unit
Number of turbines per cable type	[-]
Cost per cable type (incl. installation)	[€ m^{-1}]
Number of offshore substations	[-]
Coordinates of offshore substations	[(m, m)]
Coordinates of onshore substation	[(m, m)]
Layout	[(m, m)]

ESAU-WILLIAMS HEURISTIC

The Esau-Williams heuristic[20,70] optimises the topology of the electrical collection cables. It starts from a star topology whose centre is the offshore substations and at each iteration routes are merged if there is a saving in cable cost. The resulting optimal cable topology is branched.

PLANAR OPEN SAVINGS

The Planar Open Savings (POS) heuristic[20,71] is a greedy algorithm that connects the turbines to the substation in strings that form a star or radial topology. The term greedy is used because every string is made by merging two routes that maximise the cost savings from the current solution. In the initial solution every turbine is connected directly to the substation using a single line.

2.7.7. SUPPORT STRUCTURES

TEAMPLAY

The support structure sizing model of TeamPlay[10] comprises an analysis of aero- and hydrodynamic loads for a given monopile and tower geometry. Hydrodynamic loads are calculated using the Morison equation while aerodynamic loads are calculated by analytically integrating the drag over a cylinder. A root-finding algorithm then drives the geometry of the entire support structure to equalise the stresses to the critical yield values. Additionally, this model sizes the rocks of the scour protection layers to ensure the lateral loads are equal to the lateral bearing capacity of the soil without a scour hole. TeamPlay includes an empirical cost model for the support structure as a function of its mass and the volume of the scour protection. Only ultimate limit states are calculated for parked and operation load cases. Fatigue is neglected in the calculation and is compensated instead by a safety factor. The inputs of this model are listed in Table 2.2.

2.7.8. COST MODEL

The cost model implemented in the tool is divided into four modules: the procurement and installation of support structures, the procurement of the electrical collection cables, the O&M annual costs and all the other associated costs. The first two are embedded into the support structures and electrical infrastructure models explained above. The last two are explained next.

2.7.9. OPERATIONS AND MAINTENANCE

The average annual costs of O&M per MW of power capacity installed at realised offshore wind plants (135,000 €/MW[9]) is used to estimate the total annual O&M costs. This empirical model is thus only a function of the plant's nominal capacity, and is not influenced by array or electrical losses. One of the outputs of this model, availability, is kept equal to a constant percentage (98%), representative of current large offshore wind plants at the North Sea[72].

TEAMPLAY

The cost model of TeamPlay[10] is an empirical model that yields the cost of engineering, management, insurance, procurement, installation and decommissioning of the balance of station. The costs are a function of the number of turbines, number of substations and length of the electrical collection cables. The inputs of this model are listed in Table 2.3.

2.7.10. OPTIMISATION ALGORITHMS

Since some of the design variables in the problem formulation are discrete (see §4.2.1), only gradient-free optimisation algorithms are explored in this work. Furthermore, algorithms known for converging to local maxima (e.g. hill-climbing) are also not explored in this work. This is a consequence of the multimodality of the layout optimisation problem[73]. Feasibility is enforced at every iteration in all the following optimisation algorithms.

Table 2.2: Inputs to the TeamPlay support structure sizing model.

Component	Variable	Unit
Site	Reference height for wind speed	[m]
	Wind shear exponent	[-]
	Highest astronomical tide	[m]
	Lowest astronomical tide	[m]
	Storm surge positive	[m]
	Storm surge negative	[m]
	50-year extreme significant wave height	[m]
	1-year extreme significant wave height	[m]
	50-year extreme depth-averaged current	[m s^{-1}]
	50-year extreme wave-current angle	[°]
	Water density	[kg m^{-3}]
	50th percentile soil sieve size	[m]
	90th percentile soil sieve size	[m]
	Wave friction angle	[°]
	Submerged unit weight	[N m^{-3}]
	Ambient turbulence intensity	[-]
	Water depth	[m]
Wind turbine	Rotor radius	[m]
	Hub height	[m]
	Rotor solidity	[-]
	Drag coefficient rotor idle	[-]
	Drag coefficient of nacelle	[-]
	Front area of nacelle	[m^{-2}]
	Maximum thrust	[N]
	Distance from yaw to hub height	[m]
	Rotor mass	[kg]
	Rotor mass eccentricity	[m]
	Yaw diameter	[m]
	Wind speed at maximum thrust	[m s^{-1}]

PARTICLE SWARM OPTIMISATION

The Particle Swarm Optimisation (PSO) algorithm[74,75] mimics a swarm of particles exploring the design space, where each particle represents a design solution. Each particle broadcasts information to the rest of the swarm about the best solution (lowest fitness value) it has found at every iteration. Each particle is attracted at every time step to its

Table 2.3: Inputs to the TeamPlay cost model.

Component	Variable	Unit
Wind turbine	Rated power	[W]
	Rotor radius	[m]
	RNA purchase price	[€]
	Warranty percentage of RNA purchase price	[%]
	Hub height	[m]
Wind farm	Distance to grid	[m]
	Distance to harbour	[m]
	Onshore transport distance	[m]
	Grid frequency	[Hz]
	Electrical transmission voltage	[V]
	Grid coupling point voltage	[V]
	Number of substations	[-]
	Number of turbines	[-]
	Layout	[(m,m)]
Finance	Interest rate	[-]
	Operational lifetime	[years]
	Management percentage of CAPEX	[%]

own best-known solution and to the swarm's best-known solution. Since PSO makes no assumptions about the underlying function, it is particularly fit for approximating the global minimum of multimodal functions. The layout and position of the offshore substations are continuous design variables, whereas the number of turbines, number of substations and electrical cable capacities are discrete. Discrete variables are dealt with by considering them as continuous coordinates and rounding up their values to the nearest integer when evaluating their fitness value, effectively creating a stepwise multivariate function.

GENETIC ALGORITHM

Genetic optimisers are algorithms based on natural evolution. They recreate the mechanism by which the fittest set of individuals of a certain species survive and mate more often than those unfit, rendering thus better offspring and improving the average fitness at every generation[76]. The genetic optimiser used in this work creates an initial set of random individuals, evaluates their fitness (cost) function, and then selects the top 20% of the population and 5% of the remaining individuals to increase the genetic pool and thus the variability[77]. If only a few individuals were kept, the algorithm would more rapidly converge to a local optimum, without a wider exploration of the solution space. The individuals kept then mate randomly to produce offspring which include a random

percentage of one parent's genes, and the rest from the other. Furthermore, each offspring is mutated randomly and at a random gene, with a predefined mutation probability of 1%. Genetic algorithms have been used extensively for wind turbine blade design and wind farm layout optimisation [8,78]. The convergence of this family of algorithms is slower than PSO since evolution is a rather slow mechanism by which species improve their average fitness. A limitation of genetic algorithms is that the best final solution is typically strongly dependent on the initial population, so higher variability of the genetic pool improves the exploration of the design space.

DIFFERENTIAL EVOLUTION

Another population-based evolutionary algorithm is the differential evolution (DE) optimiser [79]. DE, in essence, adds the weighted difference between two randomly-selected individuals to a third individual. If the resulting mutant individual has better fitness than a predetermined population member, then the later is replaced element-wise by the mutant with a probability of 80%. The weight of the differential is typically set to 0.9. Out of several strategies, the random selection strategy is pursued in this work. DE is designed to handle multimodal functions and consistently converge to the global optimum.

SIMULATED ANNEALING

Simulated annealing (SA) is an optimisation algorithm based on the field of statistical mechanics [80]. SA emulates the process by which the internal energy of a crystal configuration is reduced by slowly decreasing the temperature, finding ground states of matter, where atoms are in equilibrium. In this analogy, the cost function of the optimisation problem plays the role of the energy. Small random displacements of the atoms (or design variables) are used to explore the design space, and when a new solution has lower energy than before, that solution is accepted. Instead, if the energy of the new solution is higher, then the new solution is accepted with a probability that decreases exponentially with temperature. In optimisation, the temperature decreases from high values where the material melts, to zero where the system freezes and no changes are further allowed. SA is therefore, a probabilistic search technique.

2.8. INPUTS AND OUTPUTS OF THE TOOL

The inputs of the tool are divided into two layers. The inputs to the outer layer are the models to be plugged in at every module, and the top-level optimisation algorithm. The wake models included in the AEP estimation module can be further tweaked by setting the number of wind directions and wind speeds that are to be sampled for wake analysis.

The set of inputs to the inner layer comprises the site conditions where the offshore wind farm is to be installed, fixed design parameters of the turbine and the plant, as well as finance- and market-related variables.

The output of the analysis block of the MDAO workflow is the LCOE of the given wind farm design, which is an input to the top-level optimiser.

The output of the MDAO workflow to the wind farm designer is the optimised wind farm design and its performance. However, the workflow designers may require other outputs to evaluate the performance of the MDAO workflow itself. The performance of the workflow is the information needed by the workflow designer to iterate on the model

fidelities and optimisation algorithms in the search for the set of best-performing MDAO workflows.

2.9. IMPLEMENTATION

To facilitate modularity, the Python programming language[81] is selected for its support for object oriented programming, wide availability of libraries and easiness to learn and implement.

NASA's open source framework for MDAO, OpenMDAO[82,83], is used to link the separate modular analyses and manage the data flow of the workflow. OpenMDAO is useful to speed up the creation and modification of input-output connections between modules, and to provide a standard canonical way to declare modules with their inputs and outputs accessible by others. OpenMDAO is also written in the Python programming language.

The outer layer of the tool includes an interface that enables the workflow designer to select the models to be plugged into the OpenMDAO framework.

An OpenMDAO workflow is instantiated with the set of models and number of sampling points of the windrose and Weibull distributions.

The code is available for download at:
`https://github.com/sebasanper/WINDOW_openMDAO`.

2.10. VERIFICATION AND VALIDATION

This section is devoted to validate and verify the tool for MDAO workflow instantiation. The analysis block of the workflow is verified and validated first, and then the optimisation algorithm is validated.

2.10.1. VERIFICATION OF THE ANALYSIS BLOCK

The verification of a computational model must answer the question: Is the model being implemented correctly? In other words, does the analysis block do what it's supposed to?

First, to verify whether the analysis block is implemented correctly, only the study of the AEP model is reported. The verification of the cost models and sizing tools is reported in their original sources, and is not necessary here as the original implementations are used in this tool.

The power production of one turbine at the offshore wind farm Horns Rev I is assessed at every wind direction with the same freestream wind speed, and compared to the values of another implementation reported in literature. Figure 2.12 shows the layout of the Horns Rev I wind farm and the reference turbine used for verification of the AEP module. Figure 2.13 displays the normalised power of the reference turbine predicted by the AEP module with different wake models, as well as the power estimated by the implementation and measured power reported by van Luvanee[84]. This study is used to first verify the correct implementation of the wake models, as all wake models yield similar results to the van Luvanee implementation. Secondly, this study validates the implementation too, as the results also fall within one standard deviation of the measured power.

The windrose is made with a constant and equal wind speed at all wind directions. The difference between the Ainslie 1D and 2D models is negligible. The power peaks and troughs coincide in wind direction for all wake models, and with the measured data.

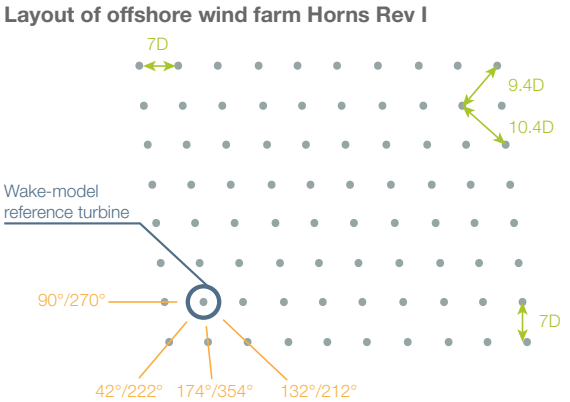


Figure 2.12: Layout, spacings and orientation of the offshore wind farm Horns Rev I.

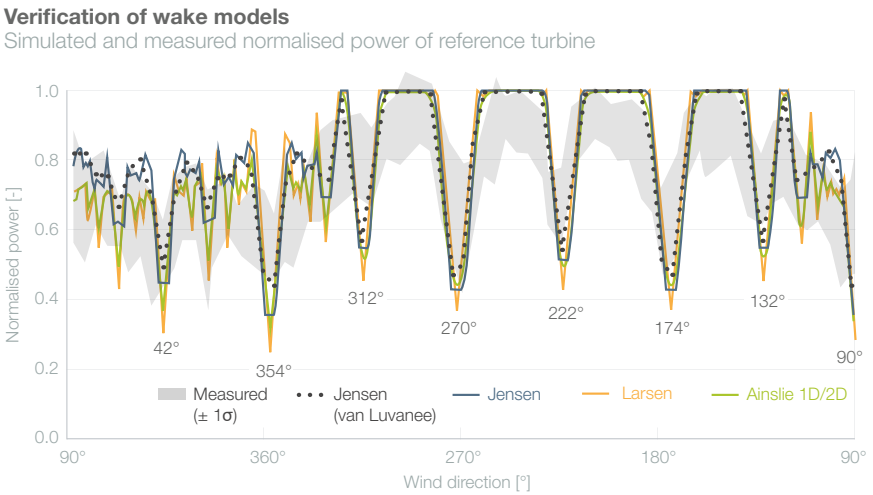


Figure 2.13: Plot of the normalised power of the reference turbine per wind direction with constant wind speed.

This study shows that the AEP model with the Jensen, Larsen, Ainslie 2D and Ainslie 1D wake models are correctly implemented, since the power of the reference turbine matches well with a previously verified implementation. The reference turbine is located inside the wind farm, and is thus subjected to wake effects from several wind directions. Since the choice of reference turbine is made arbitrarily, it is concluded that the AEP model is well implemented. Together with all cost models previously verified, it is concluded that the analysis block is well implemented.

2.10.2. VALIDATION OF THE ANALYSIS BLOCK

A reality referent is an abstraction of reality that a model intends to replicate. In practice, reality is too complex and uncertain to fully be described, and instead, a representation of reality is used. This representation—or referent—is limited to the scope of the application at hand[85].

The validation of a tool answers the question: Is the reality referent being modelled correctly? In other words, does the analysis block yield results that resemble the reality referent? In this work, the reality referent for component costs, AEP and LCOE is their expected behaviour reasoned from physical principles. Also, the reality referent for the CAPEX breakdown is a set of values previously reported in literature.

As stated in §2.2, the MDAO workflow is meant to minimise the LCOE with respect to design variables of multiple wind farm subcomponents. A model is only validated with respect to its application. It can not be assumed that a model valid for one purpose is also valid for another[86]. Therefore, the following validation study does not include the purpose of using the MDAO workflow as an offshore wind farm design tool, but as a model that captures and exploits trade-offs between the design of several system components.

Instead of validating every aspect of a model and striving to measure absolute model accuracy, one should aim at improving the confidence in the model[86]. The aim, thus, is to increase the wind farm designer's confidence in the analysis block. Because even real world data is inaccurate, nothing can represent reality perfectly[86].

The LCOE of an offshore wind farm is an elusive figure because it can only be precisely known after the project has reached the end of its lifetime. Moreover, the LCOE of an industrial project is typically confidential. For these reasons, and the fact that LCOE is strongly affected by financial interests, risk management and unexpected costs, none of which is considered in the MDAO workflow, validating the absolute LCOE value returned by the analysis block of the analysis block is not a fruitful exercise.

Doing sensitivity studies of the output with respect to its input helps increase the confidence in a model[86]. The following sensitivity studies consist in the change of a single input variable and keeping all others constant, and recording certain outputs. The goal of these studies is to increase the confidence of the wind farm designer on the ability of the inner layer of the tool to guide the optimiser towards the optimal layout.

The first sensitivity analysis assesses the responses of the normalised LCOE, AEP and costs of the electrical collection cables to changes in the power density of the site. Power density is decreased by linearly increasing the length of the sides of a squared site and fixing the number of turbines within the site to 15. For every power density, a new layout is made with a greedy algorithm that places a new turbine as far as possible from all the previous. A flat seabed is considered to avoid any influence from water depth in the layouts. Figure 2.14 shows the normalised values of the outputs. AEP increases with decreasing power density as wake effects tend to disappear, eventually limiting AEP to a constant value. On the other hand, the total cost of the electrical collection cables increases as turbines are spaced farther apart. In consequence, at first, LCOE starts to decrease due to higher energy yield, until the increase in costs of the electrical cables have a greater impact than AEP. It is concluded that the analysis block correctly captures the trends of the costs of the electrical collection cables, AEP and LCOE. As a result, the

optimiser will be driven to place turbines closer together to minimise electrical cable costs, but also to space turbines farther apart to maximise AEP. The optimiser will have a trade-off to solve to minimise LCOE.

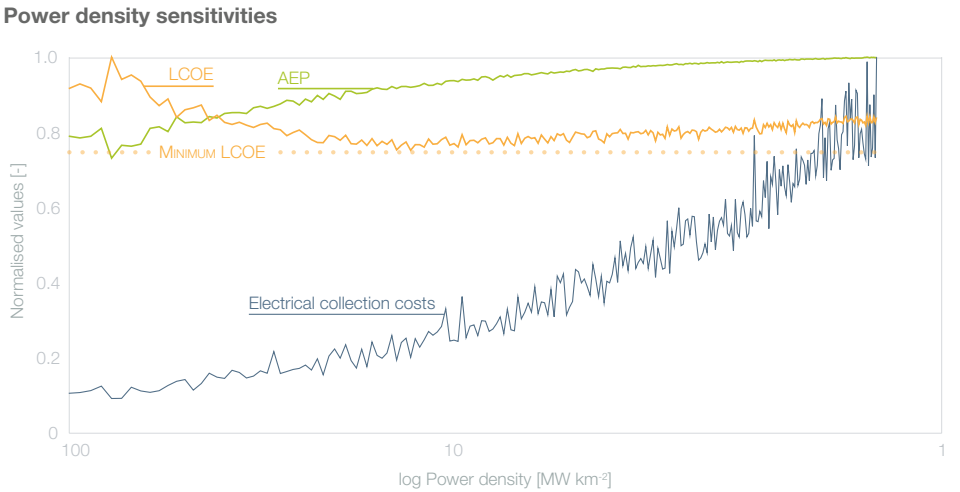


Figure 2.14: Plot of the normalised LCOE, AEP and cost of the electrical collection cables as functions of the power density of a wind farm.

Second, the responses of the normalised LCOE and cost of the support structures to water depth are analysed. This is done by analysing a single layout with varying depths of a flat seabed, so all turbines are affected identically. The results are shown in Fig. 2.15. The cost of the foundations and the LCOE increase at deeper waters, because the support structures need additional mass to support the higher bending moments, have longer monopiles and deeper penetration depths. The analysis block captures a negative impact of greater water depth on LCOE. Thus, the analysis block is able to correctly capture the impact of the local bathymetry on the optimal layout.

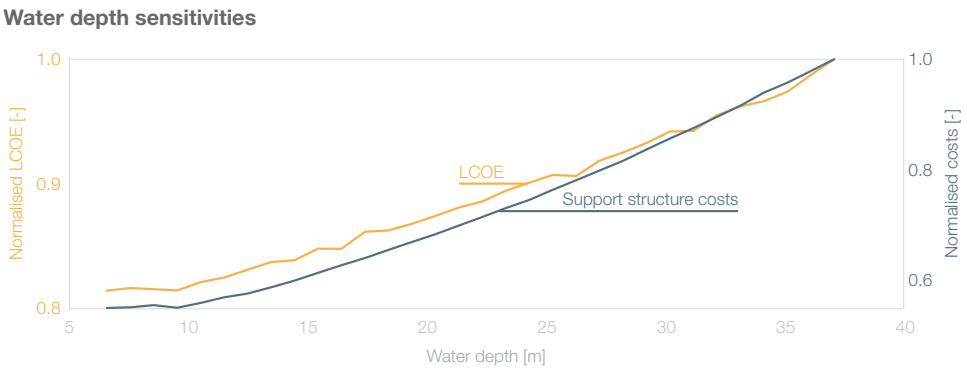


Figure 2.15: Plot of the normalised LCOE and cost of the support structures as functions of the depth of a flat seabed.

Third, the changes in the normalised LCOE, AEP and costs of the electrical collection cables with respect to the number of turbines in a fixed area with a flat seabed are assessed. For every number of turbines, a new layout is made with a greedy algorithm that places a new turbine as far as possible from the previous ones. Figure 2.16 shows that both the AEP and the costs of the electrical cables increase with increasing number of turbines. A higher AEP results from adding operational wind turbines, and AEP is expected to plateau with more turbines due to stronger wake effects. Longer electrical cables to connect more turbines results in ever-increasing costs. LCOE is seen to decrease at first due to the increase in AEP until the increase in cost of the electrical cables start dominating the increase in AEP. The analysis block again provides the optimiser the opportunity to solve a trade-off between packing more turbines in a fixed area to increase AEP and reducing the number of turbines to reduce the costs of the electrical cables.

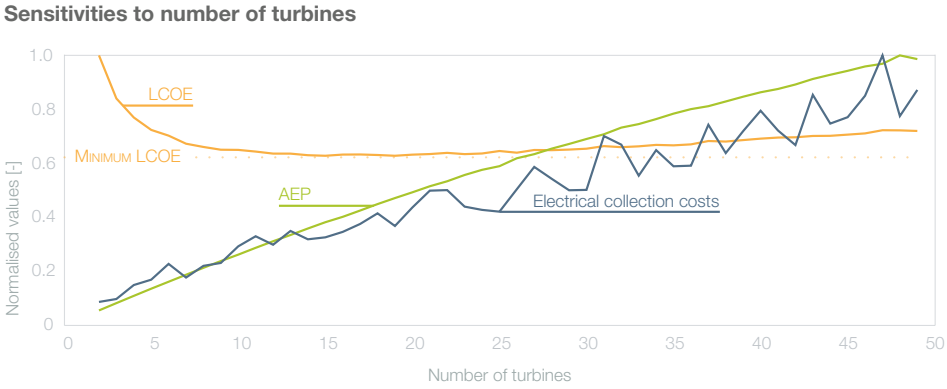


Figure 2.16: Plot of the normalised LCOE, AEP and cost of the electrical collection cables as functions of the number of turbines in a fixed area.

Fourth, the normalised LCOE and costs of the electrical collection system of a wind plant are analysed when one offshore transformer substation is located at different places inside a squared site. The seabed is an inclined plane that rises with the positive horizontal direction. Fig. 2.17 shows the positions of the wind turbines with blue dots, the optimal position of the offshore substation with white circles, and heatmaps of the costs and the LCOE. If only the costs of the electrical collection costs are considered, the optimal position of the substation is that which minimises its distance to the turbines. When LCOE is considered, it can be seen that water depth (in the form of additional costs of installation and procurement of the foundation of the substation) affects the optimal positioning of the substation too. The analysis block is able to capture the effect of water depth and length of the electrical collection cables on the optimal positioning of the offshore substation.

Fifth, to demonstrate that the analysis block captures the effect of modifying the capacities of up to three different electrical collection cables on the LCOE, the normalised LCOE of all possible combinations between 0 and 7 turbines is assessed. The position of the offshore substation and the layout are fixed. The results are shown in Fig. 2.18. The blocks, rows and columns tell the number of turbines allowed per cable string, and

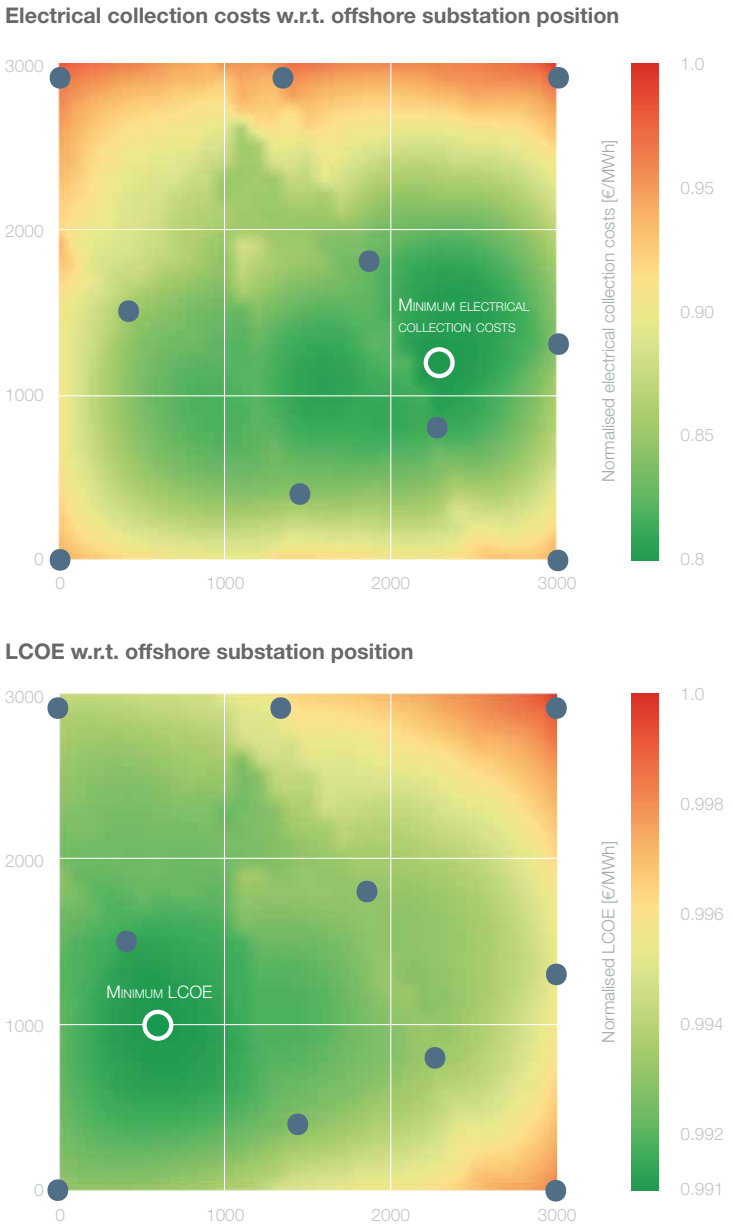


Figure 2.17: Heatmaps of the normalised LCOE and cost of the electrical collection system with respect to the position of an offshore transformer substation inside the wind farm area. Water depth increases linearly in the horizontal direction.

the LCOE is represented with a colour scale. In this case study, the electrical collection system with cable types that support the electrical current of up to 3, 6 and 7 wind turbi-

nes respectively yields the lowest LCOE. This study shows that the analysis block predicts that the use of more than one cable type leads to a decrease in costs and LCOE, as short branches of the cable topology can be made of cheaper, lower capacity cable types. Key to note is that the optimal combination of cable types is not a trivial solution and can thus be optimised.

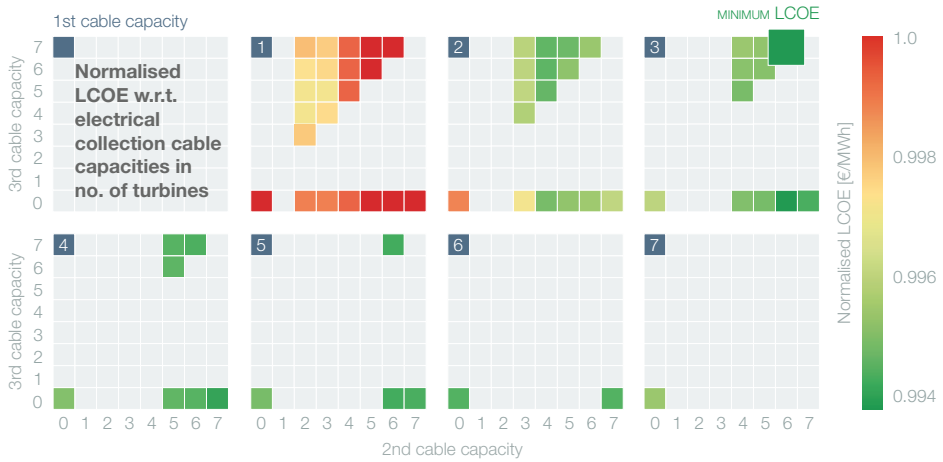


Figure 2.18: Plot of the normalised LCOE of a wind farm with three cable capacities expressed in number of turbines per string.

Finally, to validate that the cost models of the analysis block realistically represent the CAPEX distribution of a utility-scale offshore wind farm in the North Sea, a comparison is made between the cost distributions modelled and published in literature. Figure 2.19 shows the CAPEX distribution of the offshore reference wind plant of the IEA Wind Task 37 [73]. Table 2.4 summarises the CAPEX distribution of the cost model and those published by NREL [9], IRENA [44] and the compilation made by Crabtree et al. [43]. The CAPEX distribution calculated by the analysis block is deemed to be within the bounds reported in literature, and thus correctly captures the relative importance of designing the different system components.

2.10.3. VALIDATION OF THE OPTIMISATION ALGORITHMS

To validate whether the PSO algorithm coupled in the MDAO workflow optimises LCOE better than by random sampling the design space, LCOE is minimised with respect to the wind farm layout, number of turbines, number and positions of the offshore transformer substations and the ratings of the collection electrical cables. The percentiles of LCOE as sampled by the optimiser and a random sample of the same size is shown in Fig. 2.20. These results suggest that the PSO samples the design space at points closer to the optimum than a random sample.

To validate if the PSO algorithm is reaching sensible optimal solutions, three layout optimisations are carried out with different objective functions: AEP, electrical collection costs, and support structure costs.

The case study is performed for the reference wind plant of the IEA Wind Task 37.

CAPEX distribution calculated with the MDAO analysis block

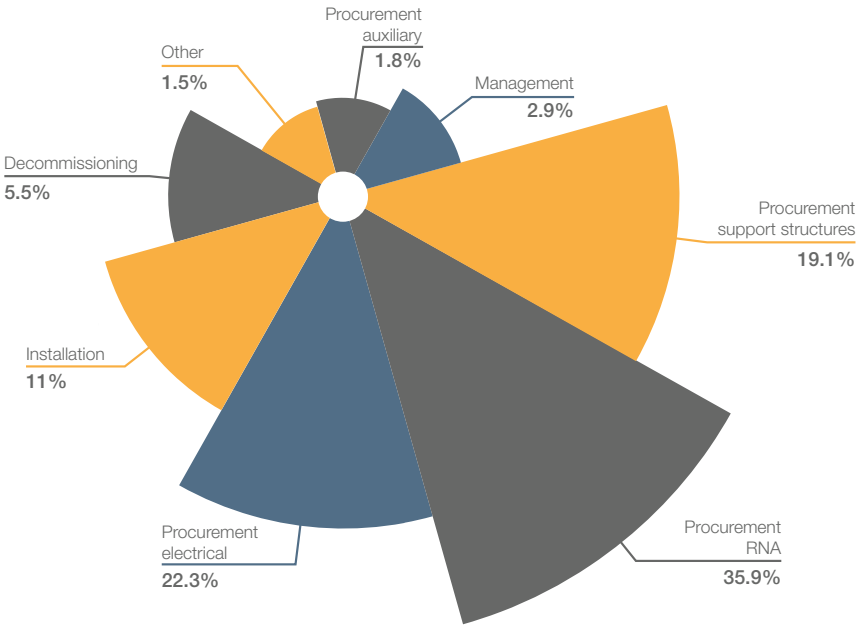


Figure 2.19: CAPEX breakdown of a large offshore wind farm in the North Sea, calculated with the MDAO analysis block's cost models.

Table 2.4: CAPEX breakdown of a large offshore wind farm in the North Sea calculated with the MDAO workflow and reported in literature. Note that columns do not sum to 100% as some costs may belong to more than one concept.

Component	MDAO analysis block [%]	NREL [%]	IRENA [%]	Crabtree et al. [%]
WT procurement	36	33	44	36-41
WT procurement and installation	41	52	57	41-56
Balance of station	48	47	43	40-59
BOS installation	7	19	13	20
Electrical infrastructure	22	9	17	14-24
Support structures	19	14	16	18-22

The site is the Borssele wind energy area off the coast of the Netherlands, with 74 10 MW wind turbines installed [73]. A minimal inter-turbine spacing of 2D is enforced in all three cases. Each optimisation run consists of 24 particles in the swarm and the termination

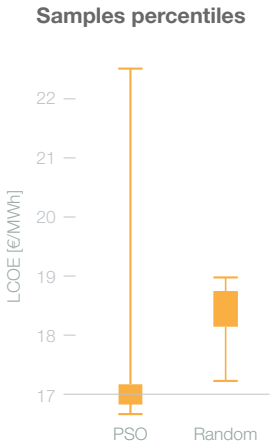


Figure 2.20: Percentiles of LCOE as sampled by the PSO optimisation algorithm and a random sample of the same size.

criterion is a number of time steps equal to 300. In total, the analysis block is called 7200 times.

Figure 2.21 shows the three resulting layouts of this validation activity. The AEP-driven layout increases the inter turbine spacing, and places half of the turbines on the site's boundaries to minimise wake effects. The layout found with electrical costs as objective function pulls all the turbines as close as possible to the offshore transformer, to minimise the length of the electrical collection cables. The last layout driven by the cost of the foundations places most of the turbines on the shallowest regions of the site. Table 2.5 shows the improvement of the three optimised objective functions and LCOE compared to a baseline design made with a greedy algorithm that spaces turbines as far as possible from each other[73]. The large spacings between the turbines of the baseline design is the reason why AEP sees the least improvement in the optimisation, while electrical cable costs sees the highest. The LCOE actually increases when the objective functions are the cost of the electrical system and the cost of the support structures. LCOE decreases more, however, when minimising the LCOE than when maximising AEP. It is concluded that if the MDAO workflow is able to drive the layout towards optimality for different objective functions, then it is also able to find the trade-off between the three objective functions when minimising the LCOE.

2

Validation of PSO algorithm
Layouts optimised with different objective functions

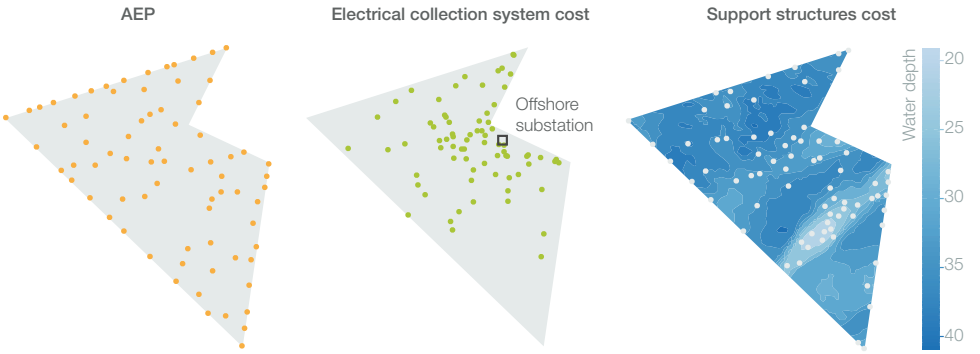


Figure 2.21: Optimised wind farm layout designs according to three objective functions: AEP, cost of the electrical collection cables, and cost of the support structures.

Table 2.5: Improvement in the objective functions of the optimised layouts with respect to a baseline design.

Objective function	Improvement of the optimised design	LCOE improvement of the optimised design
AEP	1.96%	1.03%
Electrical collection cables cost	51.98%	-7.36%
Support structures cost	3.42%	-0.02%
LCOE	—	1.12%

3 Guideline for MDAO workflow selection

Decision makers can satisfice
by finding optimum solutions for a simplified world,
or by finding satisfactory solutions for a more realistic world.

Herbert A. Simon

A guideline for selecting the most useful models and optimisation algorithms for an MDAO workflow for a predefined use case is presented.

The guideline is separated into two phases. In the first phase the most useful analysis blocks are found, and in the second these are coupled to multiple driver algorithms to select the overall most useful MDAO workflows. Both phases are further subdivided into guidelines for the evaluation, comparison and ranking of alternatives.

The selection process is treated as a multiple criteria decision analysis, where criteria are defined, scored and aggregated to detect the best-performing alternatives.

3.1. INTRODUCTION

Section 1.5 lays the case for selecting the most useful MDAO workflows for a given use case. The guideline presented in this chapter responds to that need, formally expressed in the description of Task 2 in §1.7:

> Develop a guideline for evaluating, comparing and ranking MDAO workflows for a given use case in the field of offshore wind farms.

It is emphasised that the differences in the MDAO workflows treated by this guideline lie in the fidelity of the models coupled, as well as the choice for driver algorithm.

Ideally, MDAO workflows should yield results of the highest quality, in an inexpensive way. In practice, however, these two objectives can not be met simultaneously. Therefore, the evaluation of the performance of an MDAO workflow should judge how good its results are and the cost of achieving them. For practical reasons, the quality of the results by an MDAO workflow is traced back to the quality of the results of the analysis block and the quality of the results provided by the driver. Analogously, the cost of the entire workflow is divided into the cost or use of resources of the analysis block and those of the driver.

Key in this work is to recognise that the behaviour of a system cannot be predicted by aggregating the behaviour of its components, but it will also be a function of their interactions. This remark leads us to suspect, likewise, that the realism of an analysis block cannot be estimated by only measuring the realism of its isolated constituent models. Instead, the realism of the analysis block as a whole should be assessed. Evaluating the realism of the analysis block by comparing its output with measured data is a more informative activity than evaluating the realism of the individual models coupled into it.

The following is an example of a two-way interaction between two phenomena in a wind farm that translates into a key coupling that affects the evaluation of the realism of two coupled models that simulate both phenomena. The total energy harvested by a wind farm is a function of the local wind speed experienced by every wind turbine, which in turn is a function of the wake effects. Equally important, the energy converted by every wind turbine is a function of the fraction of time that they are operational, also known as their availability. Consequently, in order to estimate the total electrical energy converted by a wind farm realistically, both the wake effects and the availability of the turbines—among others—should be taken into consideration. Consider an analysis block that includes the simulation of both phenomena to estimate the energy converted by a wind plant. Suppose, furthermore, that for the purpose of benchmarking the accuracy of both simulations independently, the measurement of the electricity converted by a real wind farm is available. The measurements consist of a time series of the local wind speed experienced by every turbine, the time it is operational and the total energy produced by the wind farm. The total energy produced by the wind plant will naturally be determined by the wake effects and availabilities. However, it will also be determined by two interactions: first, the wake behind every turbine is responsible for higher induced loads at the downstream wind turbines, which lead to higher failure rates and thus, lower availability. Second, a non-operational wind turbine due to a failure or

planned maintenance will lead to a change in the farm wake effects. Hence, besides the validity of each model, the validity of the coupled analysis block needs to be ensured.

Notwithstanding, it is acknowledged that currently the most common way of validating models that simulate individual system components is by isolating the component in an experimental set-up that avoids complex interactions, even if it is not representative of the real environment.

Because the usefulness of an MDAO workflow relies on several, often conflicting criteria (that collectively quantify cost and quality of the results), the selection of MDAO workflows is treated as a Multiple Criteria Decision Analysis (MCDA) problem. MCDA helps a decision maker resolve the trade-offs between criteria [87].

This chapter contains a description of the requirements that the guideline must fulfil. Then, the conceptual design of the guideline and a detailed description of the steps and activities recommended for selecting the best-performing MDAO workflows is presented.

3.2. REQUIREMENTS OF THE GUIDELINE

The guideline will be designed to comply with certain functional and non-functional requirements. The only functional requirement is the ability to semi-autonomously rank the alternative MDAO workflows with respect to their overall utility for a predefined use case. The non-functional requirements are that the guideline is:

- **concrete**, with the goal of it being understandable to workflow designers and unambiguous, to avoid personal interpretations;
- **flexible**, so it can accommodate the specific requirements of workflow designers and adapt to their use case and set of MDAO workflows;
- **objective**, so the opinion of the workflow designer has minimal impact on the choice of models and driver algorithm to be coupled and thus providing the guideline with higher credibility;
- **simple**, for wide deployment, accessibility and acceptability.

3.3. CONCEPTUAL DESIGN

Evaluating the utility or usefulness of an MDAO workflow is in the broadest sense composed of two parts: scoring the performance of MDAO workflows according to multiple criteria, and aggregating the scores with an MCDA method to yield an overall utility. This approach should make the guideline meet the requirement of being objective. Accordingly, the guideline includes rules to evaluate, compare and rank the alternative MDAO workflows.

However, the criteria may be conflicting, and thus, in this guideline, the concept of *solving* the trade-offs between criteria is interpreted as finding the set of non-dominated solutions or Pareto front, explained next.

Given two analysis blocks, W_1 and W_2, the standard definition of dominance is used to compare them: W_1 dominates W_2 if it satisfies the following two conditions:

1. $C_k(W_1) \leq C_k(W_2) \quad \forall k \in (1,\ldots,n)$,
2. $\exists k \in (1,\ldots,n) \ni C_k(W_1) < C_k(W_2)$,

where n is the number of criteria and $C_k(W_l)$ is the score of the l-th analysis block with respect to the k-th criterion.

When an alternative is not dominated by any of the other alternatives, it is said to be non-dominated. Due to the conflicting nature of the criteria, usually several alternatives are non-dominated, and the set of non-dominated alternatives form what is called the Pareto front of a design space. Solutions in the Pareto front solve the trade-offs between competing criteria and it is then up to the designer to select a subset of solutions for further exploration by assigning higher preference to certain criteria.

Because certain criteria require the execution of the MDAO workflows—some of them multiple times—, attempting to choose the workflow in its entirety at once is intractable due to the high computational burden of scoring the metrics that are proposed for the judgement of workflows. In consequence, the guideline for selecting the most useful MDAO workflows is divided in two sequential phases: selecting first the most useful analysis blocks and then the most useful driver algorithms. By dividing the guideline in two phases, all driver-analysis interactions are neglected. This is considered a reasonable assumption because the analysis block simulates a part of reality while the driver is a mathematical artificial technique that determines which design solutions are evaluated. The performance of the driver therefore relates directly to the nature of reality and only indirectly to the nature of the analysis block by which reality is represented.

MCDA problems can typically be classified into two groups: those with a finite number of alternatives explicitly defined, and those with an infinite number of unknown solutions or a very large number with discrete variables. The problem of analysis block selection of Phase 1 belongs to the second group due to the large number of permutations that may result from having access to many models and the workflow containing many modules.

There are three categories of MCDA techniques based on their underlying theory: outranking methods, utility function-based methods, and multiobjective programming support[88]. The first two categories are better suited for MCDA problems with a finite number of alternatives explicitly defined. Multiobjective programming, the paradigm followed in this work, is meant to support problems with a very large number of alternatives with discrete variables.

Due to the large amount of different analysis blocks, it is foreseen that it will be intractable to score all the alternatives in a reasonable amount of time. Therefore, it will be necessary to sample the space of alternatives in a *smarter* way using multiobjective programming. In contrast, the usually low number of alternative driver algorithms should be solved with an outranking or utility function-based MCDA method, where all the alternatives are scored.

Consequently, phase 1 is formulated as a multiobjective optimisation problem that yields the Pareto front of the most useful analysis blocks, while phase 2 is a multicrite-

ria decision problem that yields the most useful driver algorithms by means of applying MCDA techniques. Combined, both phases yield the most useful MDAO workflows. Figure 3.1 sketches the outline of the guideline.

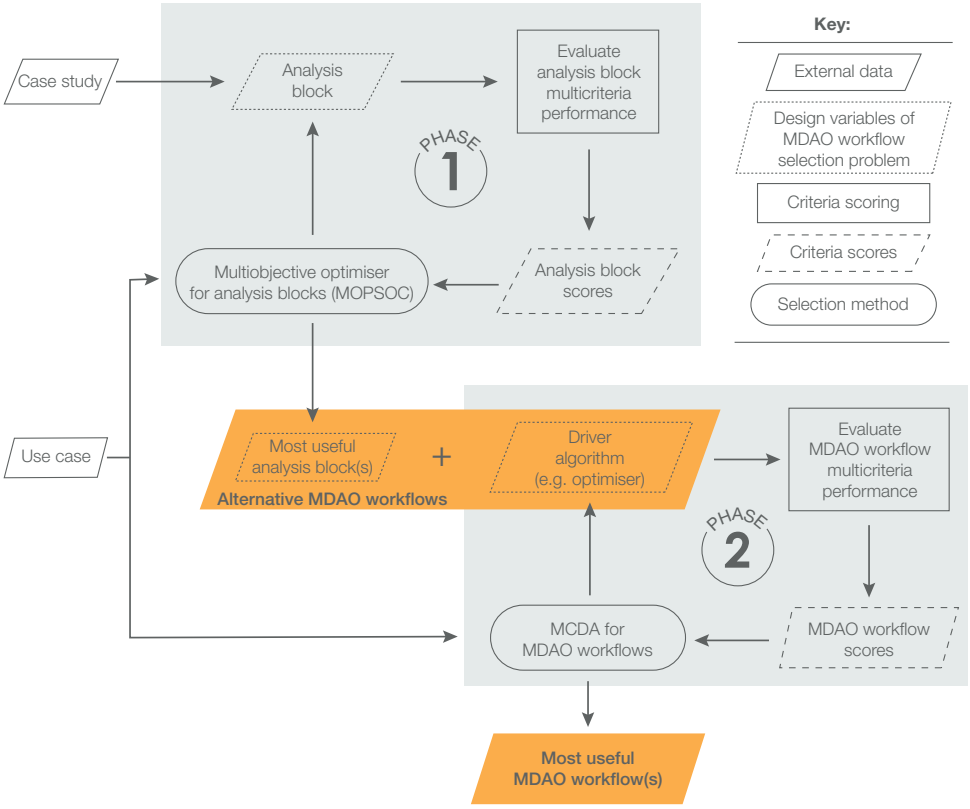


Figure 3.1: Diagram of the phases and procedures described in the guideline for MDAO workflow selection.

To generate the alternatives that will be subject to evaluation and ranking, feasible MDAO workflows have to be instantiated with the permutations of the models and driver algorithms available. At this point, all unfeasible workflows (which cannot be connected due to input-output variables inconsistencies) are discarded.

Figure 3.2 sketches what is meant by workflow feasibility. If models $A1$ and $A2$ are available for module A, and models $B1$ and $B2$ for module B, the four possible analysis blocks that can result are $A1-B1$, $A1-B2$, $A2-B1$, $A2-B2$. Of these, workflow $A1-B2$ can be discarded, as $B2$ requires inputs that cannot be obtained from the outputs of $A1$. There are then three feasible MDAO workflows from which to select the most useful (workflows 1, 2 and 4).

Having set the feasible alternatives from which to select the best MDAO workflows, the workflow designer can define further constraints on the workflows to reduce the size

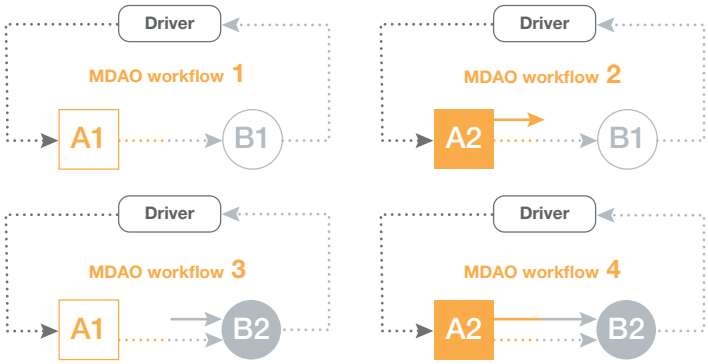


Figure 3.2: Diagram of four possible MDAO workflows where two modules can be filled by two tools each. MDAO workflow 3 is the only non-feasible alternative.

of the set, such as the requirement that models provide analytic derivatives or be programmed in a specific language.

The following sections describe, in agreement with the outline, the selection of analysis blocks first and then the selection of driver algorithms.

3.4. ANALYSIS BLOCK SELECTION

To get from a set of analysis blocks to a subset of the best-performing alternatives (a Pareto set), the guideline proposes to perform the following activities:

1. **Define the criteria** and metrics that judge the utility of the analysis blocks.
2. **Formulate the multiobjective optimisation problem** in terms of the objective functions and constraints.
3. **Run the multiobjective optimiser** to to find an approximation of the Pareto front of alternatives.
4. **Reduce** the cardinality of the Pareto set, if judged necessary by the workflow designer.

The following sections describe these activities in more detail.

3.4.1. DEFINITION OF THE CRITERIA

In MCDA, alternatives are evaluated with the purpose of comparing them. This means that the evaluation of the usefulness of an analysis block is useless if there is a single alternative. Instead, when multiple alternatives are present, there are qualitative aspects in which several alternatives stand out. In this work, these aspects are called objectives and sub-objectives, which are, in turn, measured with quantitative criteria. Because the alternatives are meant to be useful, criteria typically correspond with tangible (sub-)objectives they must comply with.

Examples of criteria that may be of relevance to the evaluation of an MDAO analysis block include: accuracy, precision, repeatability, detail, range, resolution, sensitivity, execution time, convergence, parallelism, feasibility, robustness, presence of analytical derivatives, availability, integrability, interoperability, causality, consistency, programming complexity, numerical stability, temporality, dependency, accountability, augmentability, communicativeness, completeness, conciseness, device-independence, efficiency, legibility, self-containedness, self-descriptiveness, structuredness and open-sourceness[85,89–91]. The exact interpretation of these criteria is still a matter of debate.

To shortlist the criteria used to evaluate the feasible analysis blocks, this guideline proposes to combine two methods.

The first method is a top-down approach where the following question (**Q**) is answered (**A**) recursively:

> What makes an MDAO analysis block [*(sub-)objective here*]?

An example of this process is:

- **Q**: What makes an analysis block useful?
  - **A.1**: its practicality to the use case and realism.
    - **Q**: What makes an analysis block practical?
      - **A.1**: its ability to achieve a solution fast and use resources efficiently.
  - **A.2**: its suitability to represent reality.

and so on.

The second method, in contrast is a bottom-up approach, where the meaningful differences between the alternatives are listed and then structured to higher level objectives. For example, if some analysis blocks are entirely made of open-source models, and some are not, this difference can then be take into account by an open-sourceness criterion, in turn responding to an objective that favours cost-effective solutions. Therefore, this method is informed by doing static and dynamic analysis of the different analysis blocks to discover potential differences and how meaningful they are. Static analysis, for evaluating criteria such as open-sourceness, do not need to execute the workflow and assess its structure, while dynamic analyses, to evaluate criteria such as execution time, do need to execute the workflow.

In this way a criteria tree is progressively built where each branch is a sub-objective and the branches at the lowest level are the criteria to be measured. Every use case might have different sub-objectives, e.g. optimisation would have high optimality as one of its objectives, whereas a design certification use case would care for low uncertainty of the outputs.

An example of a criteria tree for judging analysis blocks is provided in Fig. 3.3.

The difficult task is, however, to identify the relevant and most useful criteria for each specific use case and set of available alternatives. The reason why the set of alternatives may impact the criteria tree development process is that in practice, the alternatives will

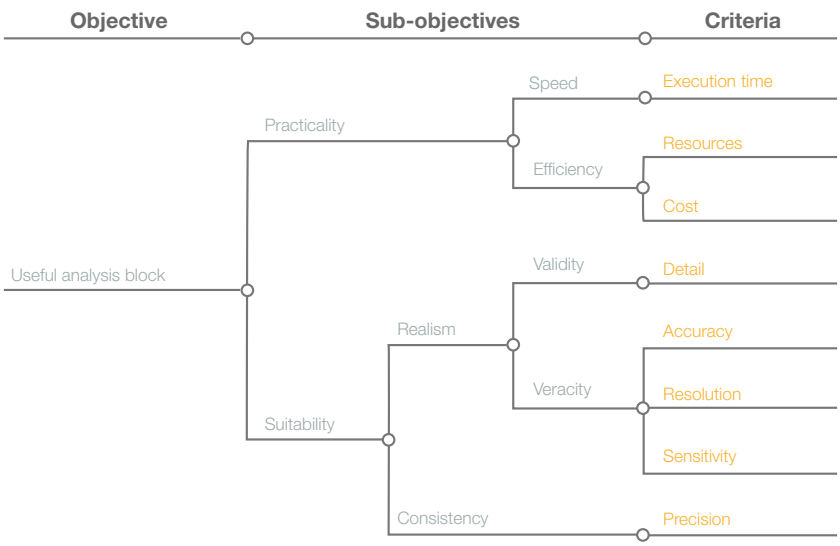


Figure 3.3: Example of a criteria tree for the multiple criteria decision analysis of MDAO analysis blocks.

share certain attributes and differ in others. It is their differences that need to be detected by the criteria.

Keeney and Raiffa[92,93] state that the list of criteria with which MCDA shall be performed must comply with five attributes: completeness, operability, decomposability, non-redundancy and size. The meaning of these five attributes is explained below, along with a discussion of how they are addressed in the guideline.

Completeness addresses the adequacy of the list of criteria to meet the overall objective (in this case usefulness) and if its sub-objectives cover all areas of concern related to the performance of an analysis block. A test for completeness entails logical deductive and inductive reasoning for proving that no gaps are left by the chosen objectives and criteria. The goal of this test is to have the list fully describe the utility of each alternative. By making a criteria tree with the methodology proposed in this guideline, gaps are more easily identified and dealt with from the early inception of the list.

Operability means that criteria must be meaningfully used in MCDA, have metrics that make (sub-)objectives measurable, be understandable and pragmatic, and be useful for making decisions. Measurable criteria should be a monotonic metric that produces values or probability distributions that reflect the decision maker's preference between alternatives. The requirement for monotony on the metrics guarantees transitiveness between the decision maker's preferences of alternatives. In addition, criteria are responsible for advocating for a particular alternative, so they should represent differences in the alternatives. If all available alternatives have the same negative or positive attribute, then a criterion quantifying that attribute will not comply with the operability attribute. Operability is addressed in this guideline by means of argumentation. The MCDA is based on the premise that the predefined metrics explain the variability in the performance of the alternatives and thus, the operability of the metrics can be deducted

from analysing the scores of the alternatives throughout the MCDA.

Decomposability refers to the capacity of a list of criteria to be arranged in the form of a tree. A decomposable list of criteria allows the decision making problem to be disaggregated into smaller problems, since criteria will fit a hierarchy. Furthermore, decomposing a problem guarantees that criteria can be measured one at a time and they should have no explicit dependencies on other criteria. The list-building approach proposed in this guideline also guarantees decomposability.

Non-redundancy in the list of criteria strives to avoid double counting any effect. Criteria must be pairwise independent. There exist a number of correlation measures that determine the degree of independence between the scores of any two criteria, and a correlation matrix helps determine whether some criteria can be discarded or combined into one. It is stressed the fact that non-redundancy is tested for the list of criteria that govern the selection problem at hand in combination with the tested MDAO workflows, and does not attempt to elevate the overlap between criteria to an absolute truth.

Size refers to the number of criteria and should be kept as small as possible. Following Occam's razor, there is great value in avoiding unnecessary complexity.

3.4.2. FORMULATION OF THE MULTIOBJECTIVE OPTIMISATION PROBLEM

Once the criteria and their metrics to evaluate the performance of an analysis block are defined, the scores of the alternative analysis blocks with respect to these individual criteria can be jointly optimised with a multiobjective optimisation algorithm.

The problem formulation of the optimisation problem is stated as:

$$\begin{aligned} &\underset{W}{\text{minimise}} && C_i(W) \quad i \in (1,\dots,n), \\ &\text{subject to:} && C_j \leq R_j \quad j \in (1,\dots,m) \end{aligned}$$

where it is assumed that the criteria to be minimised (C_i) and the criteria to be constrained (C_j) are subsets of all the criteria. These need not be mutually exclusive. R stands for the criteria scores that the workflow designer sets as constraints, and W is the categorical vector that defines the models implemented in the analysis block.

The choice of which criteria to use as objective functions or constraints is left for the workflow designer to make.

The use case can provide significant hints towards making this decision, e.g. an optimisation may benefit from having low execution time of the analysis block as one of the objective functions, and keeping its accuracy as a constraint, as the overall goal of the optimisation is not to reduce the error with respect to a referent, but to capture the trends in the behaviour of the system performance with respect to the design variables.

The following analysis is meant to provide insight into the differences of the Pareto front of analysis blocks when the optimisation problem is formulated differently.

In the extreme case where all criteria are constrained and none are minimised, the first feasible alternative found that satisfies all constraints will be the solution, as there is no global best solution in the MOPSOC algorithm.

When one criterion is minimised and the rest are treated as constraints, the multiobjective optimiser will move the swarm along a single dimension as long as the solutions

are feasible. A single solution will be returned by the algorithm. Figure 3.4 shows the solution of this case in two and three dimensions.

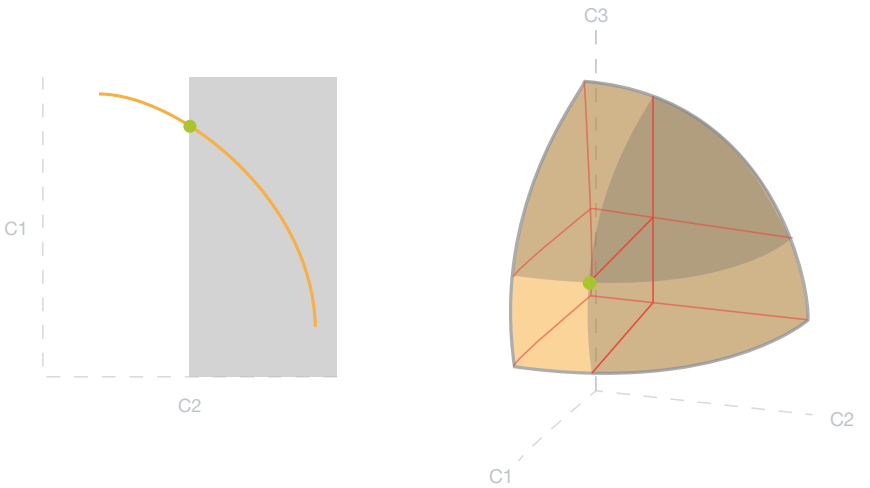


Figure 3.4: The green dot shows the solution of a multiobjective optimiser when the criterion C1 is treated as the single objective function and the rest as constraints, in two and three dimensions.

Figure 3.5 shows the 2-dimensional Pareto front yielded by the multiobjective optimiser if two criteria are jointly minimised and the rest are constrained.

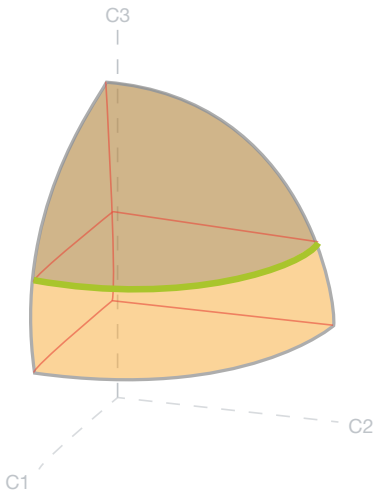


Figure 3.5: The green line shows the solution of a multiobjective optimiser when the criteria C1 and C2 are treated as objective functions and C3 is constrained.

In general, if there are m criteria, of which n are minimised and $m-n$ are constrained, the Pareto front of analysis block will have n dimensions.

Lastly, when all criteria are jointly minimised, the multiobjective optimiser will provide an approximation of the entire Pareto front of the alternatives.

3.4.3. EXECUTION OF THE MULTIOBJECTIVE OPTIMISER

This section describes the procedure by which the Pareto front of the alternatives is found.

While the Pareto front be can found by scoring and comparing all alternatives pairwise, this becomes unfeasible when the analysis blocks have several modules and there are several models available to each module.

The proposal in this guideline is to approximate the Pareto front by means of a combinatorial multi-objective optimisation algorithm. This algorithm works by minimising all criteria, so lower scores in metrics are expected to represent a higher preference of the workflow designer.

The design variables of the optimisation formulation at hand are categorical by nature (module A, module B, etc.), while the objective functions are continuous (the scores of the analysis blocks with respect to the metrics of the criteria). As a consequence, it is necessary to use an optimiser for categorical variables.

Although a genetic algorithm for this type of problems exists[94], it was deemed valuable to develop a new algorithm based on the particle swarm optimisation (PSO) algorithm[95]. PSO is a family of nature-inspired algorithms, where a swarm of particles traverses the design space, where every particle is influenced by a combination of its individual cognition and the collective behaviour of the swarm[75]. PSO algorithms converge faster than genetic based algorithms as the latter rely on long-term evolution while the former aggregates the short-term social knowledge of the swarm[96], and their exploration capabilities for finding the global optimum are enhanced by randomly scaling the velocity vectors and by adding a turbulence variable—sometimes referred to as craziness.

The new Multiobjective Particle Swarm Optimisation algorithm for Categorical Variables (MOPSOC)[95] proposed here uses probability distribution functions as design variables instead, and the Pareto front (even if non-convex) is approximated by using dynamic weight aggregation and an archive of non-dominated solutions. The next section describes the MOPSOC algorithm in detail.

The output of the MOPSOC algorithm is an approximation of the set of analysis blocks that dominate all others, across multiple criteria. As explained before, this is the approximated Pareto front. These analysis blocks are candidates to be included in the most useful and thus optimal MDAO workflows.

MOPSOC

Particle Swarm Optimisation (PSO)[75] is a heuristic algorithm originally developed for approximating the global optimum of functions of continuous variables. In nature, a swarm of bees or a flock of birds collectively explores its surroundings for a source of food that can provide for the entire colony. This mechanism is the inspiration for PSO, as every individual particle will fly about in the design space remembering the minimum

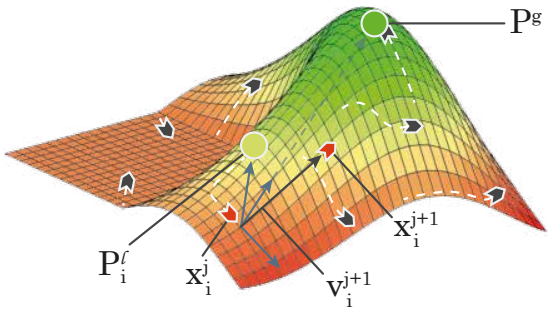


Figure 3.6: Particle Swarm Optimisation. The red particle's velocity is the linear combination of three velocity components driven by inertia, the global best solution and its own local best solution.

(or maximum) value of the underlying function it has encountered so far. This point in the search space is called the local best $\mathbf{P}_i^l$, and every particle will have its own. Meanwhile, particles broadcast their local best solution to all other particles, so that they are all aware of the best global solution $\mathbf{P}^g$ found by all particles together.

The process that updates the velocity of particle i at time step j, $\mathbf{v}_i^j$, is sketched in Fig. 3.6. Particles are attracted from their current position $\mathbf{x}_i^j$ at every time step to $\mathbf{P}_i^l$ and $\mathbf{P}^g$ with cognition and social weights w^l and w^g, respectively. Both the cognition and social velocity vector are multiplied by random numbers r_1 and r_2 in $[0,1]$. Particles also keep their velocity from the previous time step, with a weight w^{in} that represents their inertia. The sum of the three resulting velocity vectors yields the particle's velocity with which its new position $\mathbf{x}_i^{j+1}$ is calculated by using a time step equal to one ($\Delta T = 1$). Finally, a random mutation in $[0,1]$, called turbulence (δ), is added to the velocity with probability P_δ, leading to the following equations of motion:

$$\begin{aligned} \mathbf{v}_i^{j+1} &= w^{in}\mathbf{v}_i^j + r_1 w^g(\mathbf{P}^g - \mathbf{x}_i^j) + r_2 w^l(\mathbf{P}_i^l - \mathbf{x}_j^j) + \delta, \\ \mathbf{x}_i^{j+1} &= \mathbf{x}_j^j + \mathbf{v}_i^{j+1}\Delta T. \end{aligned} \tag{3.1}$$

The ratio between the inertia, cognitive and social weights can drive the swarm to converge prematurely to a local optimum or to converge very slowly to a single global optimum or to multiple local optima. However, there have been meta-optimisation studies that recommend particular sets of weights[97].

This section describes MOPSOC (Multiobjective Particle Swarm Optimisation for Categorical variables), a hybrid PSO algorithm that leverages and slightly modifies two existing mechanisms: the adaptation of PSO to single objective functions of categorical variables by Strasser et. al.[98], and the multiobjective Pareto-front-finding algorithm for functions of continuous variables by Jin et al.[99].

Concerning previous related work, there is one alternative optimisation algorithm that could potentially handle functions of categorical variables, namely genetic algorithms (GA), in particular the NSGAII algorithm[94]. However, as published, NSGAII is

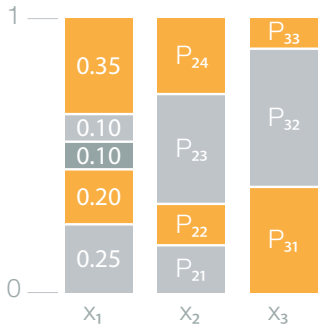


Figure 3.7: Diagram of the discrete probability distribution of three categorical variables. Variable x_1 admits five possible values, x_2 four and x_3 three.

validated and tested with functions of continuous variables and the encoding and decoding operators would need to be adapted for categorical variables. Beyond genetic algorithms, there exist a few multiobjective Particle Swarm Optimisation algorithms for mixed-integer problems[100], but they use continuous operators for discrete ordinal values and then assign the new particles the closest discrete value. This notion of distance is meaningless for unorderable categorical variables.

The capabilities of MOPSOC to execute multiobjective optimisation and explore nominal design spaces are described in the following sections.

Categorical variables feature A key characteristic of categorical variables is that their ordering is a spurious and misleading exercise. Generally, it cannot be stated that an apple is two or three times an orange, so they should not be assigned integer values but rather a symbolic, otherwise meaningless code. Therefore, MOPSOC ignores any information regarding the possible order of the alternative values of a categorical variable.

As described with more detail by Strasser et al.[98], the particle swarm lives in a space of discrete probability distributions (DPD). Every dimension of categorical data of the original problem is now described by a DPD that expresses the probability of that dimension taking on each of the possible values of that variable. To illustrate the new search space, let $\boldsymbol{\chi}_i$ be the DPD of the i-th entry of the position vector $\boldsymbol{\chi} = (\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \ldots, \boldsymbol{\chi}_m)$, then:

$$\boldsymbol{\chi}_i = \{Px_{i1}, Px_{i2}, \ldots, Px_{in}\},$$

where $Px_{ij} = Px(x_i = y_j)$ is the probability that variable x_i takes on the nominal value y_j ($\boldsymbol{y}$ is the vector of categorical alternatives for variable x_i), and $n = n(i)$ is the number of possible values of the variable x_i. Figure 3.7 shows the DPD of three categorical variables. This formulation increases the number of dimensions of the swarm to $\sum_i n(i)$. Note that:

$$\sum_{j=1}^{n} Px_{ij} = 1.$$

In an analogous manner, $\boldsymbol{v}_i = \{Pv_{i1}, Pv_{i2}, \ldots, Pv_{in}\}$ is the velocity in the probability distribution sub-space of the i-th dimension of the velocity vector $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m)$.

The global and local best positions are expressed as $\boldsymbol{\chi}^g$ and $\boldsymbol{\chi}^l_k$, where subscript k is for the k-th particle of the swarm.

By definition, $\boldsymbol{\chi}$ and $\boldsymbol{v}$ belong to a continuous space bounded by 0 and 1, and thus the subtraction and summation operators needed for PSO (Eq. 3.1) can be applied albeit with slight modifications.

Subtraction and summation in Eq. 3.1 operate component-wise on the probability distributions. After a particle's position is updated, if one of its new probabilities is less than zero or greater than one, it is reassigned to the nearest boundary and the DPD is renormalised to ensure the sum remains equal to one.

Since particles represent probability distributions, the solutions to be evaluated by the fitness function are obtained by sampling the DPDs a user-defined N_s times and then taking the mode of the values of the samples. On one hand, when the DPDs are sampled once ($N_s = 1$), similar distributions have a higher probability of having radically different fitnesses. On the other hand, when $N_s \to \infty$, similar distributions tend to yield the exact same solutions and have the same fitnesses, thus increasing the risk of prematurely converging at a local minimum.

Furthermore, when the global (and local) best solution is assigned a new distribution, it is ensured that sampling $\boldsymbol{\chi}^g$ (and $\boldsymbol{\chi}^l_k$) returns a solution similar to the sample $s = (s_1, s_2, \ldots, s_m)$ produced by sampling the new distribution. This is done because the fitness value is valid for s, and not for $\boldsymbol{\chi}$. The mechanism to do so is the following:

$$Px^g_{ij} = \begin{cases} \epsilon \times Px_{ij} & \text{for } y_j \neq s_j \\ Px_{ij} + \sum_{l \neq j} (1-\epsilon) \times Px_{il} & \text{for } y_j = s_j \end{cases}.$$

In other words, the probability of the values of the global best distribution that are not equal to the mode of the samples are scaled by a user-defined factor ϵ, and the value that corresponds to the mode absorbs the reductions in the probabilities of the other values. This is done equivalently for the local best solutions.

Additionally, MOPSOC includes the option to add turbulence to particles with probability P_δ, which adds randomness to the search and may increase the exploration capabilities of particles at the cost of slower convergence.

Multiobjective feature The second core feature of MOPSOC is its ability to find the set of non-dominated solutions with respect to multiple criteria.

This section describes the concept of an archive that stores the Pareto front at every time step of the optimisation algorithm.

MOPSOC uses the Dynamic Weighted Aggregation (DWA) technique to approximate the Pareto front[99]. A weighted sum of the n criteria C_k and weights a_k is expressed as:

$$\sum_{i=1}^{n} a_k C_k.$$

Aggregating the criteria into a single objective function using the weighted sum leads the optimiser to converge to a single non-dominated solution. However, according to Jin et al.[99], a gradual change in the weights forces the optimiser to move along the Pareto front whether it is convex or concave. It is important to note that it should be ensured

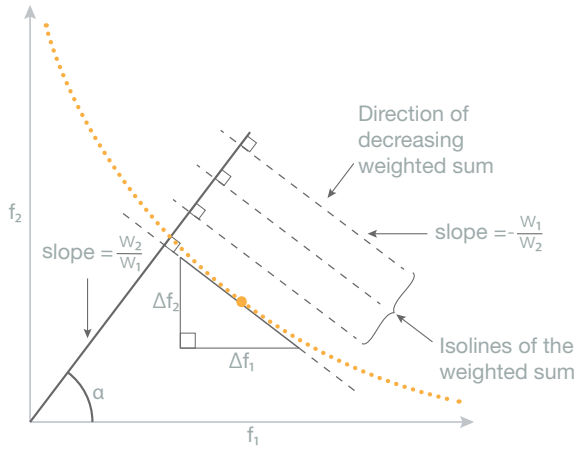


Figure 3.8: Diagram of the Pareto front, the isolines of the weighted sum and the origin of the α angle that characterises the ratio between the weights.

that the metrics of all the criteria cover the same range, for example by normalising them. This lets the weights represent only the ratio between the preferences of the criteria and avoids the weighted sum to be influenced by their different scales.

Constraints are dealt with in MOPSOC by adding penalties to the weighted sums of infeasible solutions.

Since the global optimum of the swarm moves slowly along the Pareto front, an archive is kept and maintained at every time step to store the non-dominated solutions found so far. The archive has a user-defined size and at every time step it is evaluated whether particles in the swarm dominate solutions in the archive and replace them accordingly. To avoid non-dominated solutions that are too similar in the archive, a minimum Euclidean distance is enforced with respect to all solutions in the archive to be admitted into it.

Dynamic Weighted Aggregation (DWA) relies on the generation of weights at every iteration. This algorithm includes a novel approach to generate the weights of the DWA, regardless of the number of objective functions. The concept is exemplified first in two dimensions, and then extended to higher-dimensional spaces.

The slope of the isolines ($\Delta f_2/\Delta f_1$) of the weighted sum is equal to the ratio $-w1/w2$ [101], as shown in Fig. 3.8.

Taking into account that the solution of the weighted sum problem is always Pareto optimal, without loss of generality, the big yellow circle in Fig. 3.8 will represent the solution for a given set of weights. This point lies in the intersection of the Pareto front with the line tangent to the curve, which is parallel to the isolines of the weighted sum. The direction normal to the tangent is characterised by the angle α equal to the arctangent of the weights ratio:

$$\alpha = tan^{-1}(w_2/w_1).$$

Therefore, if α is regularly sampled from the range bounded by the angles 0° and 90°,

and $w_1 = 1$ is fixed as the reference weight, then the set of weights for w_2 can be obtained with:

$$w_2 = tan(\alpha).$$

Because the tangent function diverges at 90°, the sample cannot include this value, but can be arbitrarily close.

In particular, at the lower extreme of the Pareto front, the weighted sum function is horizontal, and has its minimum at the minimum of the objective function f_2. This line has a gradient of zero ($-1/w_2 = 0$), which results when $w_2 \to \infty$. At the opposite upper extreme, when the vertical weighted sum function has a gradient of infinity, w_2 is equal to zero. Halfway through the range, at an angle of 45°, $w_2 = w_1 = 1$ and both functions have thus the same importance on the weighted sum.

In problems with n objective functions, this concept holds. The normal to the $n-1$-dimensional isosurfaces of the weighted sum is the vector of weights [101]. The search has to fill the space defined by $[0°, 90°)_1 \times [0°, 90°)_2 \times \ldots \times [0°, 90°)_{n-1}$, since one reference weight can be made equal to 1. Ideally, traversing the angles space should not have abrupt changes to let the particle swarm traverse the Pareto front smoothly.

It is furthermore suggested, when possible, to traverse the angle(s) space more than once to reduce hysteresis effects.

Algorithm MOPSOC is condensed in Algorithm 1. It is broken down into five sections: initialisation of particles and archive (lines 2-4), fitness evaluation (lines 9-12), DWA (lines 15-16), PSO operators (lines 19-20) and management of archive of non-dominated solutions (lines 23-33).

3.4.4. REDUCTION OF THE PARETO FRONT

The set of non-dominated solutions found by MOPSOC might be too large to be carried onto the next phase. It is also possible for the workflow designer to get rid of certain solutions by setting new constraints *a posteriori* on the scores of the criteria.

The most common approach for reducing the size of the Pareto front, is to rank the non-dominated alternatives based on their distances either to the utopia point or the anti-ideal points [102]. The distance can be calculate using different norms, most notably the 1, 2 and ∞-norms. This method belongs to the family of methods that find the *knee* regions. The *knee* of the Pareto front is where a small improvement along one criterion leads to a large deterioration in at least one other criterion [103]. *Knee* regions are usually the most interesting for a designer with no articulated preferences.

Furthermore, the utopia point can be defined as the non-existing alternative that scores the lowest possible values in all criteria, or the non-existing alternative that scores all the lowest values found within the Pareto front. Similarly, the anti-ideal point scores the worst possible values across all criteria. It is necessary to normalise the values of the criteria for finding their distances to utopia points. This method can be combined with the preferences of the designer by weighting the differences in values along every criterion in the overall distance norm.

Another approach to rank the alternatives, independent of the ranges of the values of the criteria, is to average the rankings of the solutions along every individual criterion [104].

Algorithm 1 Multiobjective PSO for Categorical Variables

Require: archive size N_a, number of samples N_s, number of time steps T_{max}, number of particles, w^{in}, w^g, w^l, ϵ, P_δ
Ensure: Archive of non-dominated solutions

Randomise positions and velocities $\boldsymbol{\chi}$, $\boldsymbol{\nu}$
Set global and local best fitnesses at infinity
Create empty archive

for $T = 0$ to T_{max} **do**
Old swarm ← swarm
–Evaluate fitnesses–
  for $n = 0$ to N_s **do**
    Sample the positions of the swarm
Calculate mode of samples of the swarm
Evaluate criteria at the modes of the swarm

–Dynamic Weighted Aggregation–
Generate DWA weights for current time step
Sum weighted criteria

-PSO operators–
Update local and global best solutions using weighted sums, ϵ and P_δ
Calculate new velocities and positions of the swarm using w^{in}, w^g, w^l

–Archive management–
  for particle p_i in swarm **do**
    if p_i dominates a particle in old swarm **and** p_i is not dominated by any solution in the archive **and** p_i is not similar to any solution in the archive
      if archive size ≤ N_a
        add p_i to the archive
      else if p_i dominates any solution a in the archive
        replace a with p_i
      else if any solution a_1 in the archive dominates another solution a_2
        replace a_2 with p_i
  for a_1, a_2 in the archive **do**
    if a_1 dominates a_2
      remove a_2

This approach can be combined with designer's preferences by weighting the individual rankings per criterion.

The k-modes algorithm is an extension of the well-known k-means algorithm for finding representative solutions of clusters [105], but for categorical variables. This method does not rank solutions, but simply reduces every cluster of alternatives into a single

solution.

Another method for reducing the cardinality of the Pareto front is to define objectives that quantify the utility of the non-dominated alternatives, such as how low their scores are, or how many criteria are minimised for each alternative. The problem then is to find the alternative that jointly minimises the metrics[106]. Depending on the problem, the solution may be a single alternative, or a set of non-dominated alternatives.

This activity is suggested by the guideline for cases where there are too many solutions in the original Pareto front found with the multiobjective optimiser. The method chosen for that purpose is left for the workflow designer to decide. It is suggested to use multiple approaches to compare the alternatives favoured by each.

3.5. DRIVER ALGORITHM SELECTION

The set of analysis blocks deemed the most useful for the predefined use case in phase 1 are now coupled to a top-level driver, as per the MDAO architecture defined in §2.3.2. In phase 2, thus, driver algorithms are selected by evaluating full MDAO workflows (see Fig. 3.1), where the alternatives result from permuting the available driver algorithms and the set of non-dominated analysis blocks.

In contrast to phase 1, only a few alternatives are ranked in phase 2. The consequence is that the selection process differs to phase 1.

The guideline for phase 2 is divided into the following activities:

1. **Define the criteria** and metrics that judge the utility of the driver algorithms.
2. **Score** all the alternative MDAO workflows.
3. **Sorting** the alternatives to find the Pareto front.
4. **Reduce** the cardinality of the Pareto set, if judged necessary by the workflow designer.

The next sections elaborate on the methods and definitions used in every activity.

3.5.1. DEFINITION OF THE CRITERIA

The process for evaluating MDAO workflows continues by establishing the criteria and metrics with which the usefulness of a driver is assessed.

Examples of criteria for the evaluation of MDAO drivers include precision, repeatability, convergence, sensitivity, execution time, optimality, parallelism, feasibility, robustness, integrability, programming complexity, numerical stability, efficiency, legibility and open-sourceness.

The process for making the list of criteria is covered in §3.4.1 for the analysis block, and it applies identically for making a criteria tree for evaluating a driver algorithm. An example of a criteria tree for evaluating an optimisation algorithm is provided in Fig. 3.9.

3.5.2. SCORING THE ALTERNATIVES

Once the tree of governing criteria has been chosen, all alternatives can be scored against them. Due to the low number of driver algorithms and analysis blocks, it is expected that

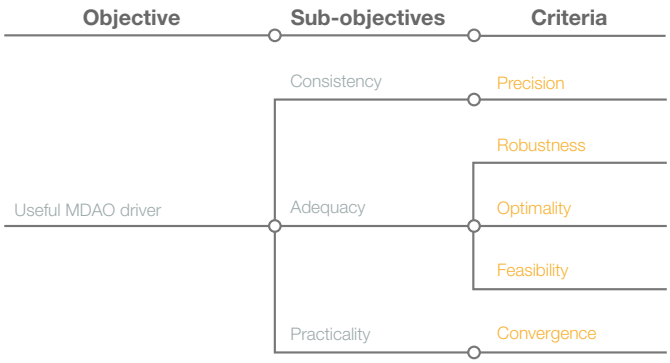


Figure 3.9: Example of a criteria tree for the multiple criteria decision analysis of MDAO driver algorithms.

this activity can be executed in a reasonable amount of time. The scores per alternative per criterion should be stored for sorting the alternatives later.

3.5.3. SORTING THE ALTERNATIVES

The concept of non-dominance for comparing alternatives, introduced in §3.4.2, applies to phase 2 as well. Alternative drivers are thus compared by means of detecting if they dominate each other in the sense of Pareto.

Provided that the number of alternatives in phase 2 is expected to be in the order of tens, the entire set of alternatives may be evaluated without incurring in extreme costs or use of resources. Therefore, it is suggested to score all alternatives with respect to all criteria and then use the ϵ-non-dominated sorting algorithm [107]. This algorithm sorts all alternatives and provides the non-dominated set of alternatives, or the Pareto front of the complete set.

3.5.4. REDUCTION OF THE PARETO FRONT

The process for reducing the cardinality of the Pareto front in phase 2 is identical to phase 1. Some techniques are recommended and discussed in §3.4.4, and the choice is left to the workflow designer.

4

Implementation of the guideline

The sweet spot between *paralysis by analysis* and going *extinct by instinct.*

The guideline described in the previous chapter is put to the test in this chapter. An example of an offshore wind farmmultidisciplinary design optimisation case study and a set of alternativeMDAO workflows are used to implement the guideline.

In the first phase, four criteria, accuracy, execution time, detail and open-sourceness, are co-optimised with respect to the models and number of sampling points of the analysis block. A Pareto front of analysis blocks results and is further reduced in size by introducing ranking techniques.

The resulting six best-performing analysis blocks are then coupled to five optimisation algorithms in the second phase. This set of 30 MDAO workflows is ranked with respect to the optimality, precision, number of function calls, robustness of the design and improvement criteria, and the Pareto set is found. The set is further reduced and the solutions ranked by their utility to the use case.

This chapter demonstrates how to instantiate the guideline and identifies unforeseen obstacles, for which suggestions are given to overcome them.

4.1. INTRODUCTION

The guideline for selecting the set of best-performing MDAO workflows presented in chapter 3 describes the activities that a workflow designer should follow.

This chapter reports one implementation of the guideline. It is important to showcase what every step of the guideline requires to reach a solution that is useful to the wind farm designer. This is also a first exercise to critically assess the utility of the guideline.

The multidisciplinary design optimisation of an offshore wind farm is the use case that drives the search for the most useful MDAO workflow.

The set of alternative MDAO workflows to select from are built using the tool described in chapter 2, with its *empty* MDAO workflow, and its available models and optimisation algorithms. A total of 3,317,760 different feasible MDAO workflows are available to select from, made of 663,552 analysis blocks and 5 optimisation algorithms.

The implementation of the guideline follows, with sections devoted to every activity suggested by the guideline and in the same order.

Finally, the entire process of instantiating the guideline is discussed, putting forward the challenges that may be encountered by a workflow designer, a few suggestions to solve them.

4.2. USE CASE

The use case presented in §2.3 is also chosen for implementing the guideline for selecting the best-performing MDAO workflows. It is repeated here to define the context of this chapter:

> **What is the optimal wind farm layout, number of wind turbines, design of the electrical infrastructure and design of the support structures that jointly minimise the levelised cost of energy at a given wind site, in a preliminary design phase?**

Sensitivity analyses of the LCOE with respect to the design of all the components and sub-components considered in the use case were presented in §2.10.2. The analyses show an associated LCOE curve with a non-trivial minimum, which justifies the use of the MDAO workflow presented in chapter 2 for solving this use case.

4.2.1. OPTIMISATION FORMULATION

The optimisation problem is formulated with LCOE as the objective function, the design variables are listed in Table 4.1, and the constraints are listed in Table 4.2.

The number of wind turbines is allowed to vary between 2 and 50. The upper bound results in a very large power density of 28 W m^{-2}, unseen in offshore wind farms, and helps maintain acceptable analysis times. The wind farm layout is constrained to keep all turbines inside a square with its bottom left corner placed at the origin. The layout is constrained to maintain a minimum separation between adjacent turbines of two rotor diameters (D_{rotor}) to avoid collisions, avoid large fatigue loading, and remain within the operable domain of the wake models.

Concerning the electrical collection system, the cabe topology is expressed as a list of connections between two turbines or between a turbine and a substation. Cables are

Table 4.1: Design variables per component

Component	Design variables (number)	Bounds	Unit
Layout	Number of turbines N_T (1)	[2, 50]	[-]
	Coordinates (2 N_T)	[(0, 3000), (0, 3000)]	[(m, m)]
Electrical collection	List of cable links (N_T)	—	[-]
	Cable capacities (3)	[(1, 9), (0, 9), (0, 9)]	[turbines]
	Number of substations N_S (1)	[1, N_T]	[-]
	Position of the substations (2 N_S)	[(0, 3000), (0, 3000)]	[(m, m)]
Support structures	Monopile diameter (N_T)	[(0,1)]	[(m)]
	Monopile penetration depth (N_T)	[(0,1)]	[(m)]
	Tower wall thicknesses (50 N_T)	[(0, 0.5 D_{tower})]	[(m)]
	Transition piece wall thickness (N_T)	[(0, 0.5 D_{TP})]	[(m)]
	Scour protection d_{50} (N_T)	[(0, H)]	[(m)]

Table 4.2: Constraints per component

Component	Constraints (number)
Layout	$distance(T_i, T_j) \geq 2D_{rotor} \quad \forall \; i \neq j \quad (N_T^2)$
Electrical collection	No cable crossings (N_T)
Support structures	Combined stress on monopile ≤ Critical stress (N_T) Overturning moment of monopile ≤ Soil lateral bearing capacity (N_T) Shear stress on scour protection ≤ Critical stress (N_T)

not allowed to cross each other as they are trenched into the seabed[20]. Up to three cable capacities are allowed to be used in the topology, and the optimiser is free to choose them. Cable capacity is bounded by 9 turbines, corresponding with the highest cable capacity available to the optimiser. Furthermore, the number and location of the offshore substations are optimised as well. The top-level optimiser is responsible for driving the electrical collection cable types, number and location of the substations, whereas a nested optimiser drives the topology of the electrical collection cables for every layout (see the XDSM in Fig. 2.6).

The design variables of the support structure and scour protection are bounded by physical constraints. The tower wall thicknesses can not be greater than half the diameter of the tower ($0.5D_{tower}$) and the thickness of the transition piece can not be greater than half its diameter ($0.5D_{TP}$). Water depth (H) is the maximum size the scour protection rocks can have. Other geometrical parameters are found using knowledge-based

rules[10]. The support structures are designed by matching the maximum stresses of the monopiles and towers, and the maximum lateral load on the soil under extreme loads, to their critical values. The support structures are optimised in their entirety by a nested optimiser, as per the XDSM in Fig. 2.6.

4.2.2. CASE STUDY

This section describes the case study of the offshore wind farm to be optimised. It includes the site conditions and the fixed design parameters.

Because the goal of implementing the guideline is to understand and interpret its results, an artificial case study is developed to gain control of the site conditions and fixed design parameters. The case study is complex enough to provide the MDAO workflow the possibility to capture the multiple trade-offs between disciplines and components.

The site corresponds to a square area whose side length is 3 km. If 70 MW of nominal power is installed, this area would have a power density of 7.8 W m^{-2}, a value representative of recent commercial developments in the North Sea.

The site is located 60 km away from the onshore electrical substation. The operational lifetime of the wind farm is 25 years at an interest rate of 0.075%. The electrical collection system operates at 33 kV, and the transmission line has a nominal voltage of 220 kV.

The bathymetry is designed as a Gaussian hill. This shape is expected to attract turbines and the offshore substation towards the centre to minimise installation costs, whereas the wake interactions will push the turbines towards the boundaries.

The wind rose is also artificially created to have a uniform wind direction distribution with the same Weibull wind speed distribution at all wind directions (A=8.15, k=2.11). Ambient turbulent intensity is assumed constant and equal to 8%.

Figure 4.1 shows the bathymetry at the site of this case study. The chosen bathymetry and windrose increase the interpretability of the optimised designs.

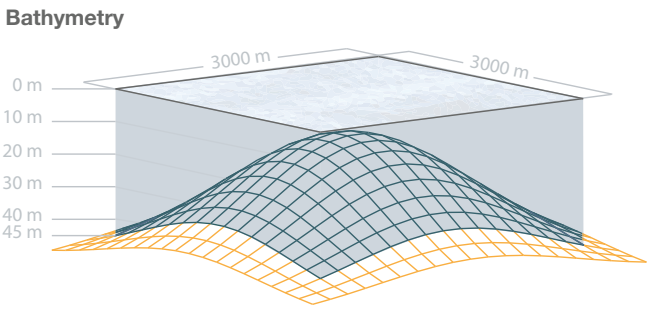


Figure 4.1: Gaussian hill bathymetry at the site of the case study.

One wind turbine type is to be installed on site, corresponding to the NREL 5 MW reference wind turbine[50]. This turbine is chosen due to the wide availability of its design parameters and input files that describe this turbine for different simulation software packages.

4.3. ALTERNATIVE MDAO WORKFLOWS

This section reports which are the alternative MDAO workflows considered in the selection process. The alternatives are divided into analysis blocks and optimisation algorithms.

4.3.1. ANALYSIS BLOCKS

Table 4.3 lists the available models and the numbers of sampling points of the distributions of the wind speed and direction. The samples are taken at regular intervals instead of randomly. The table also lists whether each model is supposed to be open-source or not, and a short identifier. Most models are described in §2.7, but a few other dummy models are added solely to test the guideline. The dummy models are built to yield little to no variation in their outputs. The Ainslie 2D wake model is used as the referent to score the performance of the other wake models, and is deemed too slow for wind farm optimisation. This model is therefore not included in the selection process.

The alternative analysis blocks that can be built with the available models and numbers of sampling points result from the permutation of all the models per module, as shown in Fig. 4.2. A total of 663,552 different analysis blocks can be built. This large amount of alternatives stresses the need to sample the models sub-space smartly.

4.3.2. OPTIMISATION ALGORITHMS

Four optimisation algorithms and one random-sampling algorithm can be plugged into the top-level driver as described in §2.7.10. These are all gradient-free methods that can optimise continuous and integer design variables. The random-sampling technique is a dummy method, included to test the guideline.

4.4. ANALYSIS BLOCK SELECTION

As expressed previously in §3.3, the guideline suggests starting with the selection of the set of best-performing analysis blocks. This section follows the first phase of the guideline described in §3.4 and reports the results of all the suggested activities in the same order: defining the criteria, formulating the multiobjective optimisation problem, executing the optimiser and reducing the Pareto front.

4.4.1. DEFINITION OF THE CRITERIA

This section gives an account of the implementation of the activities suggested in §3.4.1 of the guideline.

The choices of criteria that will be used to compare the features and the performance of analysis blocks are the result of a top-down approach, in which higher order objectives are broken down into sub-objectives, followed by a bottom-up approach in which the differences and similarities of the analysis blocks are identified and grouped into higher-level objectives.

In the top-down approach, the process of building the criteria tree starts by defining the top-level objective of the analysis blocks: the desire that it is useful for solving the use case. An analysis block is expected to be useful for wind farm optimisation if it simulates reality well, and if it uses as few resources as possible. A model simulates reality well if it

Table 4.3: Available models (and their ID) to every module and number of sampling points. Open-sourceness is a fictitious attribute.

Module	Models	Open-source	ID
Wake deficit	Jensen	Yes	J
	Larsen	No	L
	Ainslie 1D	Yes	A1
	Constant deficit	Yes	C
Wake merge	Root sum square	Yes	RSS
	Maximum	Yes	MAX
	Multiplied	Yes	MUL
	Summed	Yes	SUM
Wake-added turbulence	Larsen	Yes	L
	Danish recommendation	Yes	DR
	Quarton	No	Q
	Frandsen	No	F
	Frandsen 2	Yes	F2
	Constant effective turbulence intensity	Yes	C
Electrical collection design	Esau-Williams	No	EW
	Planar Open Savings	Yes	POS
	N_T-dependent cables length and cost	Yes	C
Support structure sizing	TeamPlay	Yes	TP
	Constant support structure cost	Yes	C
Wind turbine power and thrust	FAST	Yes	FAST
	QBlade	No	QB
	WindSim	Yes	WS
	BEM	Yes	BEM
	Step function power and thrust coefficient	Yes	C1
	Constant power and thrust coefficient	Yes	C2

Distribution	Number of sampling points
Weibull	2, 3, ..., 25
Windrose	12, 18, 24, 30, 36, 45, 60, 90

yields accurate and precise results. Resources that an analysis block can consume are of a storage, memory, monetary, labour or timely nature. Accurate results should capture absolute output values close to a referent, and their sensitivities with respect to changes in the input should match the behaviour of a referent.

A bottom-up approach looks at the alternative analysis blocks, their attributes and characteristics that make them distinguishable. The differences between the models

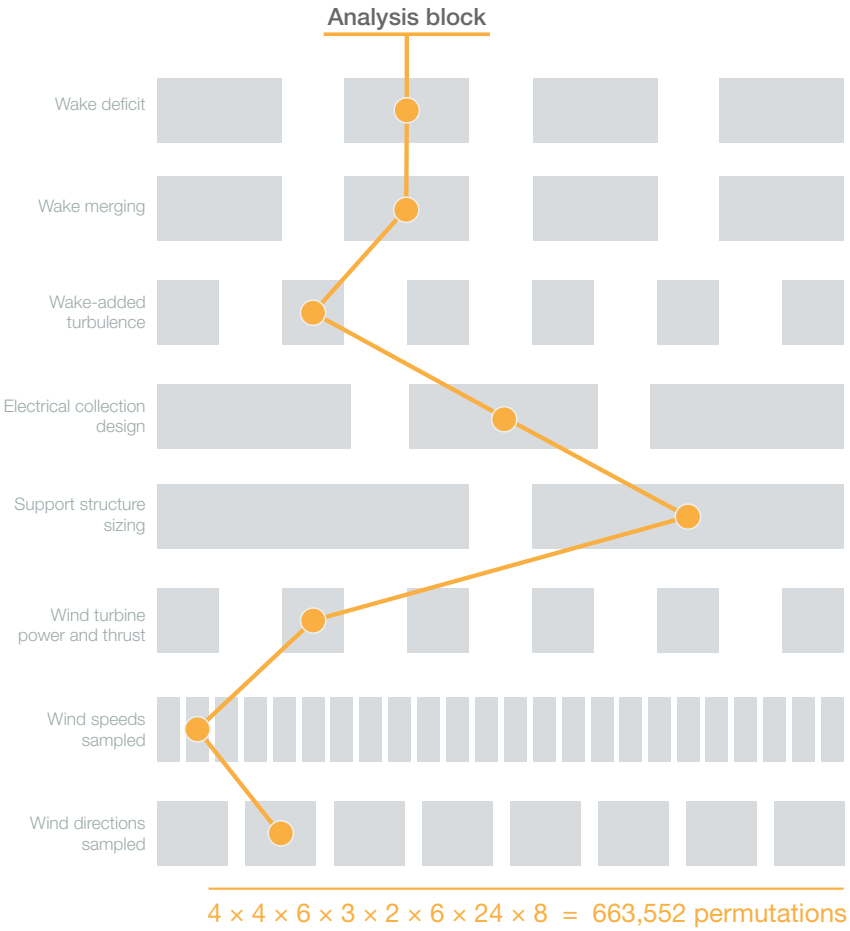


Figure 4.2: There are multiple permutations of models to build different-performing analysis blocks.

presented in §2.7 will certainly lead to different output values, i.e. they will have their own associated accuracy. Similarly, analysis blocks have an associated resolution and sensitivity. Resolution measures the minimum change in the input that corresponds to a change in the output while sensitivity captures the magnitude of the change of the output with respect to changes in the input. Moreover, precision captures the variation in the output values calculated with the exact same inputs. In addition, models will have different execution times and memory use depending on their level of complexity or number of sampling points. A noticeable feature of the models is that they all request a different number of inputs, represented by a criterion called detail[85]. Lastly, all models have been assigned a fictitious binary value that represents their open-sourceness. This attribute has been assigned arbitrarily to test the capabilities of the guideline.

Altogether, accuracy, precision, use of resources, sensitivity, resolution, detail, and open-sourceness, are the criteria found relevant in this selection process.

The tentative list of criteria found above is evaluated against the completeness, operability, decomposability, non-redundancy, and size attributes.

Completeness: The optimisation of the design of a wind farm benefits from an analysis block that captures the effect of the variation of the design variables on the system level performance, and that does so in an acceptable amount of time, as it will be run many times over. The top-down approach yields a complete list of criteria that fulfil these two objectives. Missing criteria, such as the existence of analytical gradients or programming language, are irrelevant for the use case at hand, where only gradient-free optimisation algorithms are coupled in the workflows, and all the models are coded or wrapped in the same programming language.

Operability: Accuracy, execution time, open-sourceness and detail are the only operable criteria, as they can be measured in a short time and have the power to distinguish between alternatives. Accuracy can be measured by comparing the estimated LCOE of a design with a referent LCOE value. Execution time can simply be measured with a stopwatch. The open-sourceness of an analysis block can be assigned the percentage of open-source models coupled into it. Finally, by counting the number of inputs to an analysis block, the detail criterion expresses to a first degree the level of complexity of the physics modelled.

On the other hand, the rest of the criteria are deemed inoperable. The fact that all the models are deterministic, renders the precision criterion useless. The resolution criterion is also regarded as a useless criterion, since the resolution of the inputs and outputs of most models is limited only by machine precision. The sensitivity criterion is considered too expensive to be measured since the behaviour of LCOE with respect to changes in all the design variables is needed. The memory consumption of all models is low compared to the memory of a desktop computer, and thus this criterion is not meaningful as it will hardly point out distinction between different analysis blocks. A similar reasoning yields storage and monetary cost as inoperable criteria.

Decomposability: The criteria tree shown in Fig. 4.3 tells that the problem can be decomposed into measuring how well it simulates reality (realism) and how many resources it consumes (practicality). Realism can be also decomposed into accuracy and detail. Likewise, practicality can be decomposed into speed and efficiency.

Non-redundancy: It is crucial to avoid the list of criteria from counting any effect two or more times. For that purpose, Fig. 4.4 plots the values of the four operable criteria pairwise, for a random sample of 6000 analysis blocks, using normalised metrics defined later. It can be seen that no salient correlations between criteria exist, which is evidence to support the non-redundancy of the list.

Size: The list of operable criteria has, conveniently, a number of criteria (four) that the MOPSOC algorithm can deal with in an acceptable number of iterations. The number of dimensions of the MCDA make the problem tractable.

With regards to the metrics, some criteria will need to have a referent defined, to indicate how well the analysis block represents it, e.g. an output value should match a reference value determined by another method, either a measurement or simulation, while other criteria will only have a desired minimum or maximum value to achieve. The

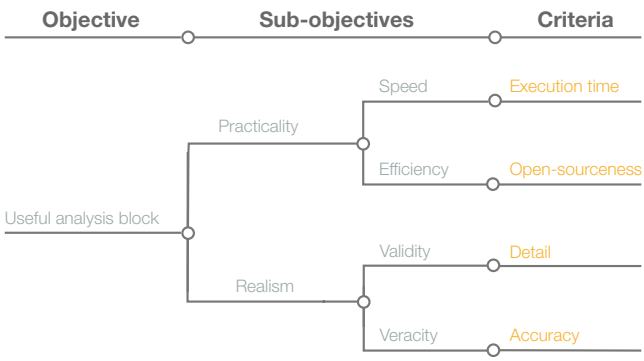


Figure 4.3: Criteria tree to evaluate the analysis block.

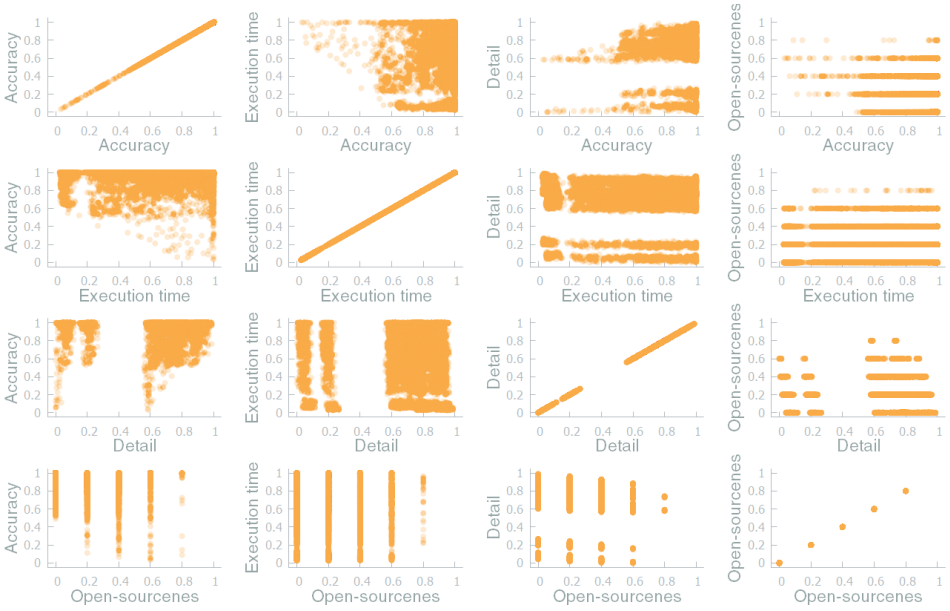


Figure 4.4: Correlation plots of a random sample of analysis blocks.

metrics are designed to be minimised by the optimiser, so lower values should represent better alternatives. The metrics of the final list of criteria are described below.

Accuracy: To measure how accurately a given analysis block simulates a reality referent, this metric (C_{acc}) calculates the root of the sum of the squares of the absolute difference between the estimated ($LCOE_e$) and referent LCOE ($LCOE_R$) of five random design vectors. These designs are drawn once and used to score all the analysis blocks. The referent LCOEs of the designs are estimated with the analysis block that couples the most sophisticated models: Ainslie 2D wake model, RSS wake merge model, Quarton turbulence model, Esau-Williams heuristic for cable topology, TeamPlay support struc-

ture design tool, FAST aeroelastic wind turbine simulation, 26 sampling points of the Weibull distribution, and 360 1° wind direction sectors. Lower values of C_{acc} represent more accurate analysis blocks. Mathematically, this metric is expressed as:

$$C_{acc} = \sqrt{\sum_{i=1}^{5} (\mid LCOE_{R_i} - LCOE_{e_i} \mid)^2}. \tag{4.1}$$

Execution time: The time it takes to run the analysis block for all of the five designs mentioned above is recorded. The metric (C_{time}) corresponds to the measured time divided by 5, to represent an average execution time per design. The additional time incurred in building the wind turbine look-up tables for power and thrust curves is not included in this criterion for simplicity purposes. Lower values of C_{time} represent faster analysis blocks. This metric reads:

$$C_{time} = \frac{1}{5} \sum_{i=1}^{5} T_i. \tag{4.2}$$

Detail: The cardinality of the set of input parameters that the models coupled in an analysis block request is a proxy for the level of detail of the physics simulated. This metric counts the cumulative input parameters (I) of the models and it is normalised with respect to the analysis block that couples the models with the most input parameters (I_{max}). Only five modules (wake, turbulence, electrical design, turbine and support design) have exchangeable models with varying values of detail. Lower values of this metric (C_{det}) represent alternatives with higher level of detail. Its expression is:

$$C_{det} = 1.0 - \frac{\mid I \mid}{\mid I_{max} \mid}. \tag{4.3}$$

Open-sourceness: This metric (C_{os}) measures the fraction of open-source models that are coupled into an analysis block. Since the open-sourceness of individual models ($OS(M_i)$) are counted with a binary variable, the total open-sourceness of the analysis block can only take its value from a set discrete values. Only five modules (wake, turbulence, electrical design, turbine and support design) have exchangeable models with varying values of open-sourceness. Lower values of this metric represent analysis blocks with higher open-sourceness. The metric is the following:

$$C_{os} = 1.0 - \frac{\sum_{i=1}^{5} OS(M_i)}{5}. \tag{4.4}$$

4.4.2. FORMULATION OF THE MULTIOBJECTIVE OPTIMISATION PROBLEM

This section describes the implementation of the activities described in §3.4.2 of the guideline.

A multiobjective optimisation will be performed to approximate the Pareto front of the analysis blocks with respect to the criteria and metrics defined above.

The Pareto front found by minimising some criteria while constraining the values of the rest of the criteria is a lower dimensional subset of the Pareto front found by optimising all the criteria without constraints. The only difference in applying constraints on

the values of the objective functions *a priori* and *a posteriori* is that the former is a faster approach, as the search space is reduced and convergence is achieved faster.

At one extreme, placing constraints on the maximum values of all the criteria would lead to a focalised search near the section of the Pareto front that satisfies the constraints. At the other extreme, when no constraints are defined, the optimisation yields the entire Pareto front.

In this case study, no constraints are enforced *a priori* and instead the values of all the criteria are optimised. Formally, this formulation is written as:

$$\underset{W}{\text{minimise}} \qquad C_{acc}(W), C_{time}(W), C_{det}(W), C_{os}(W),$$

where W is the vector that defines the models coupled in the analysis block and the number of sampling points of the wind speed and direction probability distributions.

4.4.3. EXECUTION OF THE MULTIOBJECTIVE OPTIMISER

This section reports the implementation of the activities suggested by the guideline in §3.4.3.

Using the set of feasible alternative analysis blocks, the tree of criteria and the formulation of the multiobjective optimisation problem, the Pareto front of the alternatives is approximated using the MOPSOC algorithm.

MOPSOC compares analysis blocks by assigning them a weighted sum of the criteria, and the weights are dynamically changed to have the swarm follow the Pareto front.

The weights for the weighted sum should only represent the relative importance between criteria. Hence, the criteria are normalised prior to the multiobjective optimisation to avoid their scales from influencing the weighted sum. The normalisation procedure is only done for the accuracy ($\hat{C}_{acc}$) and execution time ($\hat{C}_{time}$) criteria, as the metrics for detail and open-sourceness are already normalised. The normalised criteria depend on user-defined maximum acceptable values for accuracy($C_{acc_{max}}$) and execution time ($C_{time_{max}}$), beyond which their metrics tend to 1. The hyperbolic tangent is one example of a continuous differentiable function with range [0, 1) in the domain of the non-negative real numbers. The functions used in this work are:

$$\hat{C}_{acc} = tanh(C_{acc}), \tag{4.5}$$

$$\hat{C}_{time} = tanh(C_{time}), \tag{4.6}$$

The execution time beyond which a metric is close to 1, results from the total optimisation budget. Suppose the wind farm design optimisation needs to last at most 24 hours. If an optimisation algorithm with 300 iterations and 20 individuals calls the analysis block 6000 times, the maximum time allowed to be spent per execution of the analysis block is 14.4 seconds.

Analogously, the accuracy of the analysis block is designed to be at its worst if it exceeds or falls short of the LCOE estimated by the referent analysis block by 30%. The normalisation functions are modified accordingly, and result in the following equations:

$$\hat{C}_{acc} = tanh(0.08C_{acc}), \tag{4.7}$$

$$\hat{C}_{time} = tanh(0.04 C_{time}). \tag{4.8}$$

MOPSOC is run with 20 particles in the swarm, as recommended in literature [75,108]. The optimisation consists of 343 iterations, to sample a three-dimensional weight-space at 7 regular intervals along all axes.

Figure 4.5 shows the level diagrams representation of the Pareto front of alternative analysis blocks found with MOPSOC. The level diagrams is a technique for visualising n-dimensional Pareto fronts [109], and consists of making several plots where each of them shows the values of a criterion or a design variable against the distance to the utopia point of all the solutions in the Pareto front. This technique has two useful features: the solutions can be ranked according to their proximity to the utopia point, and solutions have the same y-coordinate across all the plots. The distance can be calculated based on different norms. In this example 12 plots are shown: 4 for the criteria and 8 for the design variables, using the 2-norm Euclidean distance of the normalised criteria. The criteria values on the x-axis appear in their original scale.

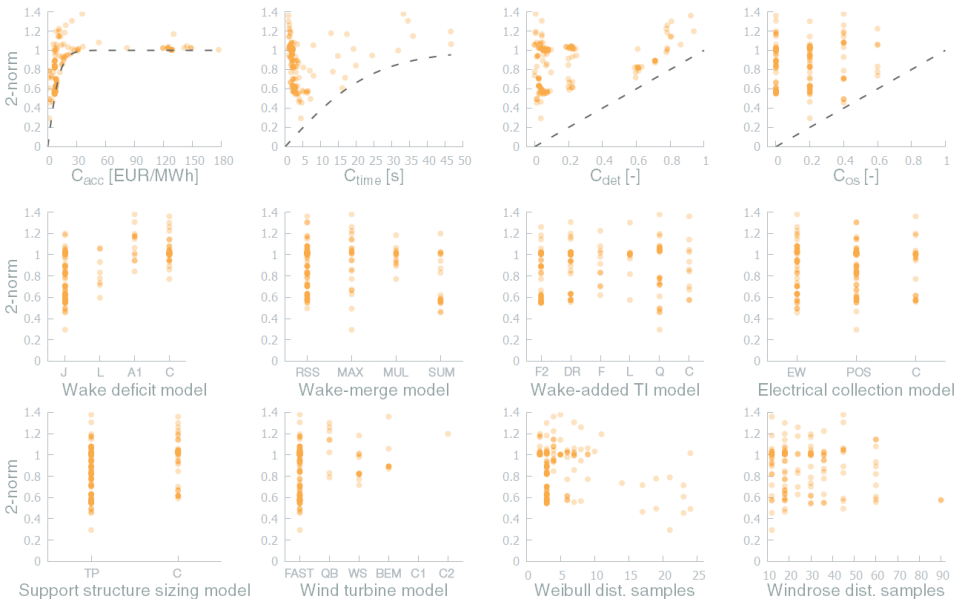


Figure 4.5: Level diagrams of the Pareto front using the 2-norm distance to the utopia point. The dashed lines define the value of each criterion when all the other criteria are zero.

Several conclusions can be drawn from the level diagrams. First of all, all the analysis blocks lie within a 0.3 and 1.37 2-norm distance to the utopia point. Of these, several have a 2-norm lower than 0.6, which represent solutions with low scores across all the criteria.

By inspecting the first level diagram for the accuracy criterion, it is observed that the solutions that have an error greater than ~30 €/MWh score nearly zero in all the other

criteria. There is a big cluster of solutions with $6.2 < C_{acc} < 9.4$, of which several score nearly zero in all other criteria. Below an error of 6.2 €/MWh there are solutions with a 2-norm distance between 0.3 and 0.8, meaning that very accurate analysis block may quickly score worse in the other criteria. It is noteworthy that no solution has an error lower than 0.92 €/MWh.

The execution time of the solutions with the lowest 2-norm distance is between 2 and 4 seconds, an indication that both faster and slower models quickly lose value with respect to the rest of the criteria. In particular, faster analysis blocks tend to be less accurate and have less level of detail. It can be seen that the other three criteria barely ever are valued close to zero simultaneously, as all solutions have are well above the line defined by the execution time metric. Moreover, most of the analysis blocks in the Pareto front take less than 4 seconds to execute.

There is a gap in the level of detail between 0.24 and 0.6. All the solutions with $C_{det} < 0.24$ couple the more accurate FAST wind turbine model, and many of them are closer to the utopia point than all the analysis blocks without FAST because all the wind turbine models are represented by look-up tables and have the same execution time. Similarly, the distance of the solutions to the line defined by the detail metric indicates that the other three criteria do not score values near zero simultaneously.

With respect to the open-sourceness criterion, most of the solutions closer to the utopia point are between 60 and 100% open-source ($C_{os} \leq 0.4$), and none have 20 or 0% open-sourceness. The solutions closest to the utopia point have an open-sourceness value of 0.2 and 0.4. The lowest 2-norm is achieved by an analysis block with $C_{os} = 0.2$, followed by analysis blocks with $C_{os} = 0.4$. Therefore, high open-sourceness helps achieve lower 2-norm distances. However, solutions with all levels of open-sourceness can also have high 2-norm distances, a result of assigning this attribute to models arbitrarily without correlation to other criteria.

Regarding the design variables, most of the solutions of the Pareto front, including those closest to the utopia point, couple the Jensen wake model. This model has the best compromise between accuracy and execution time. The Larsen model is negatively affected by fictitiously not being open-source. The constant dummy wake model is the least accurate, and this is well reflected by the higher 2-norm distance of the blocks that include it. Similarly, the Ainslie 1D wake model is too slow to be closer to the utopia point than the other models.

The wake-merging method most commonly found in the Pareto front is the root sum square, and the least found is the multiply method. The root-sum-square, maximum and summed methods appear in the solutions with the lowest 2-norm distances. The solutions that couple the multiply method always have a 2-norm distance greater than 0.8, so this model seems to negatively influence the overall utility of the analysis blocks. A single analysis block of the Pareto front has a 2-norm lower than 0.4 and it couples the maximum wake-merging method.

All the turbulent models appear in the non-dominated solutions, but the Frandsen 2, Danish Recommendation, and Quarton models appear the most times. The Quarton model, although fictitiously not open-source, appears in the solutions closest to the utopia point. This can be traced to its accuracy, since it is the model used in the referent analysis block. The fact that all models appear with low and high 2-norm distances, can

be explained by the low sensitivity of the LCOE with respect to the added turbulence intensity in the wake.

Interestingly, all three models for designing the electrical cable topology appear in the analysis blocks that are farthest and closest to the utopia point. This is an indication that the cost of the electrical infrastructure affects the LCOE much less than other costs or the AEP, and that the variation in the output of the three models is small in terms of the LCOE. Therefore, those analysis blocks that yield a very inaccurate cost of the electrical infrastructure are fast and may still be accurate in the LCOE.

The support-structure sizing model TeamPlay appears in the alternatives with the lowest distance to the utopia point. This is because it is the most accurate and more detailed of the two. Nevertheless, the dummy constant support structure cost model can appear in good analysis blocks, too. Although the TeamPlay model can also appear in the solutions farthest from the utopia point, it clearly provides an advantage in terms of the 2-norm distance.

The analysis blocks that couple the FAST wind turbine model are clearly favoured. This models appear in the solutions closest to the utopia point, whereas the other models appear less often and in solutions farther from the utopia point. The dummy step and constant output models, are too inaccurate to be worth coupling them to the analysis block. FAST is very much favoured because it is coupled in the referent analysis block for the accuracy metric and also has the highest level of detail. At the same time, FAST has no negative implications because look-up tables are used as surrogates for all the models and do not incur additional execution time.

The solutions with the lowest 2-norm distance sample the Weibull distribution between 15 and 25 times. This is an expected result, since the referent analysis block for the accuracy criterion samples the Weibull distribution in 26 points. However, most solutions, whether close or far from the utopia point, sample the distribution in less than 10 points. More sampling points results in a higher execution time and the improvement in accuracy is seemingly not enough to counteract the loss in time.

On the contrary, the number of wind directions sampled from the windrose distribution seems to have very little effect on the overall utility of the analysis block. A lower number favours execution time at the expense of accuracy, and vice-versa. The irregularity of the five wind farm layouts used to measure the accuracy criterion, show that wake losses are insensitive to wind direction, as the wind direction sample size barely influences the accuracy of the analysis block.

While the 2-norm distance is helpful to see the geometrical distance to the utopia point, the ∞-norm distance is better suited to see the trade-offs. Figure 4.6 shows the level diagrams based on this norm.

The ∞-norm-based level diagrams show that analysis blocks with C_{time} lower than 5 seconds, C_{det} lower than 0.2, and C_{oS} lower than 0.2 rapidly deteriorate due to other criteria. The fact that the ranges where all the criteria have an ∞-norm closer to zero, also include many solutions with large ∞-norm distances, is an indication that the Pareto front is non-convex. This is further supported by the range of the 2-norm distances in Fig. 4.5. All the solutions in the Pareto front lie between the hypersphere of radius 0.3 and the hypersphere of radius 1.4. Because the hypersphere of radius 1 is non-convex, the

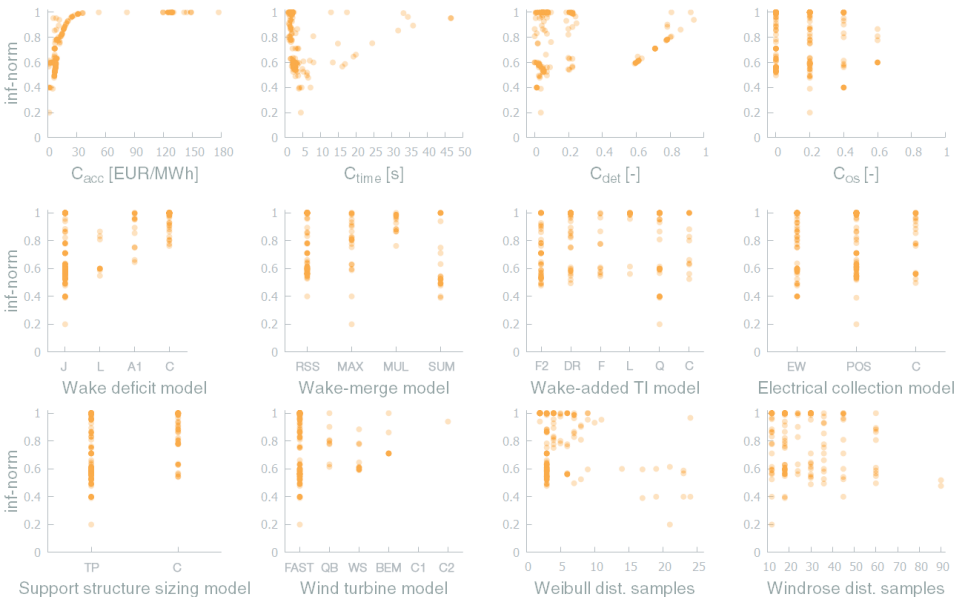


Figure 4.6: Level diagrams of the Pareto front using the ∞-norm distance to the utopia point.

solutions above the unit hypersphere correspond to non-convex regions of the front[109].

In addition, by colouring the ∞-norm-based level diagrams based on the wake model coupled in Fig. 4.7, it can be seen that the choice has the highest impact on the accuracy and execution time criteria. These two criteria roughly cluster analysis blocks with the same wind wake model together. In particular, the Jensen and Ainslie 1D wake models are coupled in the most accurate analysis blocks, followed by the Larsen and the dummy constant wake deficit model. Furthermore, The Jensen and dummy models are in the fastest analysis blocks, followed by the Larsen model and with the Ainslie 1D model being by far the slowest. Also, the most detailed analysis blocks are closer to the utopia point when coupling the Jensen wake model, due to its good performance in accuracy and execution time. The summed wake-merging method is almost exclusively coupled to the Jensen wake model, and the Jensen model is never coupled to the multiply wake-merging method. Because the dummy constant-wake deficit model is independent of the ambient wind speed, the MOPSOC finds that there is no gain in sampling the Weibull distribution in more than 10 wind speeds, whereas the Jensen and Larsen models benefit the most from increasing the number of sampling points. The Ainslie 1D model is too slow when run with many wind speed sampling points. The Jensen model is closest to the utopia point when sampling the windrose at 30 or more wind directions. Finally, the Ainslie 1D wake model is never run at less than 30 wind directions, unlike the rest of the models. Therefore, the Ainslie 1D model is most useful when more time is spent sampling the windrose than the Weibull distribution.

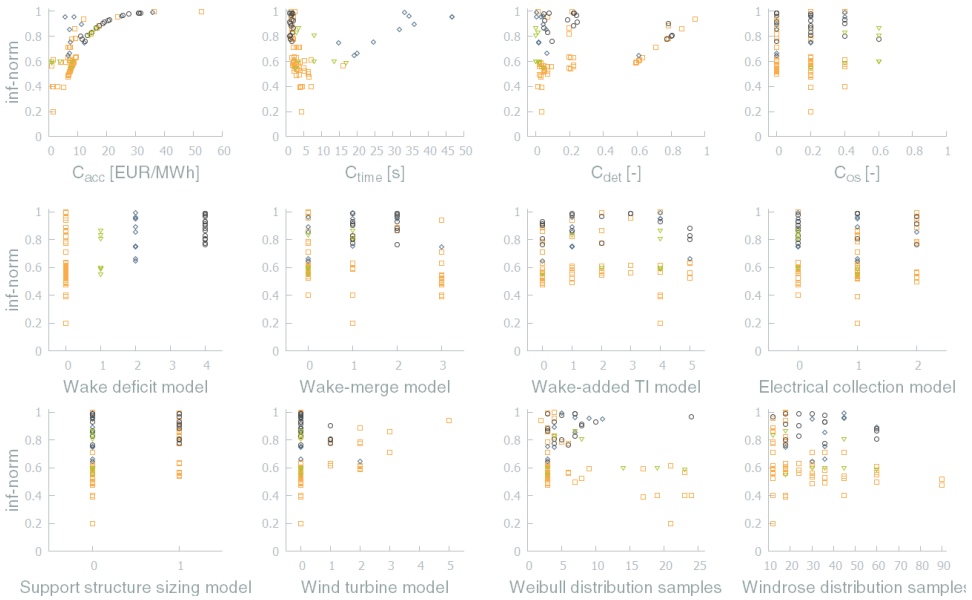


Figure 4.7: Level diagrams of the Pareto front using the ∞-norm distance to the utopia point. Colour represents the wake model coupled in the analysis blocks. Use the wake-deficit model as legend.

Other relevant observations provided by level diagrams not shown are the following.

Accuracy is higher when more points of the Weibull distribution are sampled, but at the cost of execution time.

Almost all the analysis blocks that sample the Weibull distribution at more than 15 points, couple the Quarton wake-added turbulence model. This artefact results from the fact that both choices appear in the most accurate analysis blocks, which, like the referent analysis block for accuracy, also couple the TeamPlay support structure sizing tool.

The dummy constant support structure cost model appears in the fastest analysis blocks, while TeamPlay appears in the most accurate. After the FAST wind turbine model, TeamPlay is the model with the largest number of inputs, and this is reflected in the detail criterion. The analysis blocks that couple TeamPlay either score ~0.1 or ~0.6 in C_{det}, whereas those that couple the dummy model score either ~0.2 or ~0.9, depending on whether the FAST model is present or not.

Noteworthy is the fact that the analysis blocks that do not couple the FAST wind turbine model, are closer to the utopia point when coupling the TeamPlay support structure sizing tool. This could indicate that the impact of the choice of support structure cost model on level of detail and accuracy outweighs the combined impacts of the choice of wind turbine model. And given that besides FAST all the other wind turbine models appear in analysis blocks that score higher than 0.6 in the detail criterion, it can be stated that it is the accuracy criterion which determines that the support structure model is more impactful.

4.4.4. REDUCTION OF THE PARETO FRONT

This section exemplifies the activities of the guideline suggested in §3.4.4.

If the workflow designer has absolutely no preferences on the criteria or design variables, then two approaches—amongst many others—to select the set of analysis blocks that will be carried onto phase 2 are to rank the solutions in the Pareto front with respect to their distance to the utopia point, and by averaging their rankings when sorted per criterion.

Table 4.4 shows the 3 closest solutions to the utopia point using the 2-norm distance.

Table 4.4: Models and number of sampling points in the top 3 solutions using the 2-norm distance to the utopia point.

Module	First place	Second place	Third place
Wake deficit	J	J	A1
Wake merge	MAX	SUM	RSS
Wake-added turbulence	Q	Q	Q
Electrical collection design	POS	EW	POS
Support structure sizing	TP	TP	TP
Wind turbine power and thrust	FAST	FAST	FAST
Weibull sampling points	21	23	23
Windrose sampling points	12	12	30

Table 4.5 shows the 3 closest solutions to the utopia point using the 1-norm distance.

Table 4.5: Models and number of sampling points in the top 3 solutions using the 1-norm distance to the utopia point.

Module	First place	Second place	Third place
Wake deficit	J	J	J
Wake merge	MAX	RSS	RSS
Wake-added turbulence	Q	DR	F2
Electrical collection design	POS	POS	POS
Support structure sizing	TP	TP	TP
Wind turbine power and thrust	FAST	FAST	FAST
Weibull sampling points	21	3	3
Windrose sampling points	12	18	36

Table 4.6 shows the 3 closest solutions to the utopia point using the ∞-norm distance.

Table 4.7 shows the 3 most consistent solutions according to the average ranking by their 1-norm, 2-norm and ∞-norm distances to the utopia point.

Finally, Table 4.8 shows the top 3 ranked analysis blocks when averaging their rankings with respect to every criterion. The advantage of this method with respect to the previous four, is that it disregards the utopia point and the scales of the criteria.

Table 4.6: Models and number of sampling points in the top 3 solutions using the ∞-norm distance to the utopia point.

Module	First place	Second place	Third place
Wake deficit	J	J	J
Wake merge	MAX	SUM	SUM
Wake-added turbulence	Q	Q	Q
Electrical collection design	POS	POS	EW
Support structure sizing	TP	TP	TP
Wind turbine power and thrust	FAST	FAST	FAST
Weibull sampling points	21	17	23
Windrose sampling points	12	18	12

Table 4.7: Models and number of sampling points in the 3 most consistent solutions according to the average ranking by their 1-norm, 2-norm and ∞-norm distances to the utopia point.

Module	First place	Second place	Third place
Wake deficit	J	J	J
Wake merge	MAX	SUM	RSS
Wake-added turbulence	Q	Q	Q
Electrical collection design	POS	EW	EW
Support structure sizing	TP	TP	TP
Wind turbine power and thrust	FAST	FAST	FAST
Weibull sampling points	21	23	19
Windrose sampling points	12	12	45

Table 4.8: Models and number of sampling points in the top 3 analysis blocks using the average ranking method.

Module	First place	Second place	Third place
Wake deficit	J	J	J
Wake merge	SUM	SUM	RSS
Wake-added turbulence	F2	DR	DR
Electrical collection design	EW	EW	POS
Support structure sizing	TP	TP	TP
Wind turbine power and thrust	FAST	FAST	FAST
Weibull sampling points	3	3	3
Windrose sampling points	30	36	18

In the 15 best solutions found with all methods there are 8 unique analysis blocks. The top 1st and 2nd places—in total six analysis blocks—are selected to be coupled to the alternative optimisation algorithms in phase 2. These are the 1st analysis block of Table 4.4 (ID: 1A), the 2nd of Table 4.5 (ID: 2B), the 2nd of Table 4.6 (ID: 2C), the 2nd of

Table 4.7 (ID: 2D) and the 1st and 2nd of Table 4.8 (IDs: 1E, 2E).

If a workflow designer does want to include preferences, one example of a procedure to do so *a posteriori*, is to filter the solutions in the Pareto front that have an accuracy $C_{acc} < 10$, are 100% open-source, and do not couple the FAST wind turbine model. The blue dots in the level diagrams in Fig. 4.8 show the solutions that fulfill these constraints, and the black square is the solution with the lowest ∞-norm distance to the utopia point. This solution happens to be the same with the Euclidean distance.

Figure 4.8: Level diagrams of the Pareto front using the ∞-norm distance to the utopia point. The filtered solutions are shown with blue dots, and the closest one to the utopia point is marked with a black square.

The preferred analysis block that incorporates the designer's preferences couples the Jensen wake model, root-sum-square wake-merging model, Danish Recommendation wake-added turbulence model, the POS electrical cable design tool, the support structure sizing tool TeamPlay, the WindSim turbine model, and samples the Weibull distribution at 3 wind speeds, and the wind rose 60 times every 6 degrees.

Because this analysis blocks includes subjective preferences added by the workflow designer, it is not mixed with the previous six selected for the second phase of the implementation.

4.5. OPTIMISATION ALGORITHM SELECTION

This section applies the second phase of the guideline presented in §3.5 to the use case defined in §4.2. The objective is to select the set of best-performing MDAO workflows by coupling five alternative optimisation algorithms to the set of the most useful analysis blocks found in phase 1 (§4.4.4). The steps to be taken in this phase are reported in

the order they are suggested in the guideline: defining the criteria to evaluate the performance of the MDAO workflows, scoring all the alternatives against all criteria, sorting the alternatives to find the set of non-dominated workflows and reducing, if necessary, the size of the Pareto front.

4.5.1. DEFINITION OF THE CRITERIA

As with the criteria to evaluate the analysis blocks, a top-down approach is followed first to find the criteria with which different optimisation algorithms can be evaluated.

The usefulness of an optimisation algorithm for wind farm design relies on two broad aspects: its adequacy to find good designs, and its ability to do it efficiently using the least possible amount of resources. Design solutions are deemed to be good if their performance is close to the optimal, if they are robust design solutions insensitive to small changes in their design variables, and if they are feasible. On the other hand, an efficient optimiser will find the solutions with the least amount of effort and have the ability to consistently reach the same optima when the algorithm is run multiple times (precision). Effort can be classified as computational expenses, human labour or monetary costs.

Because the global optima are not known *a priori*, other objectives are needed to quantify the quality of the resulting designs. One of them is the overall improvement between the final design and the initial designs explored, and another is the ability to find all the global optima, if there is more than one. In wind farm layout optimisation, symmetries in the layout and site conditions may lead to several different design vectors with the same performance.

By looking at the alternative optimisation algorithms available, it is possible to discard a few criteria from those mentioned above. All the optimisation algorithms enforce feasibility at every iteration, which ensures the final design is always feasible, rendering this criterion useless for ranking purposes. Similarly, all the alternative algorithms are only capable of producing a single optimal result, and so number of optima found is not a good criterion. Additionally, effort is only differentiated in the number of function calls needed to achieve convergence. Hence, monetary, time and human labour expenses are not considered.

The initial tentative list of criteria to use for evaluating the performance of the optimisation algorithms is: optimality, robustness, number of function calls to convergence, overall improvement, and precision. The criteria tree is shown in Fig. 4.9.

Therefore, the tentative list of criteria is evaluated against the completeness, operability, decomposability, non-redundancy, and size attributes.

Completeness: The list of criteria captures the differences in the resulting designs, the process that leads to finding the optimum, and the resources needed to find the optimum. The list is complete to evaluate the set of alternatives at hand, e.g. if there were other feasibility-enforcing techniques present, a measure of the feasibility of the resulting design would be necessary for the list to be complete.

Operability: Optimality is typically the first point of comparison between optimisers and its equal to the LCOE of the optimal design calculated with the most accurate analysis block. Robustness is measurable using the variation of the objective function when

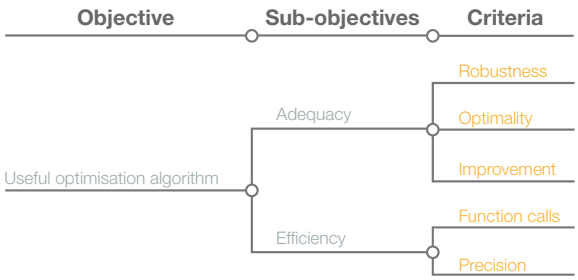


Figure 4.9: Criteria tree to evaluate the optimisation algorithms.

the design variables sample the vicinity of the optimal solution, and it is an operable criterion if a small number of points are sampled. Number of function calls to find the optimal design is easily found after the optimisation has finished and its history is revised. Improvement is the difference between the objective function evaluated at the initial design and at the last iteration. It is key to be aware that a good value of improvement might be due to a good value of optimality or a bad performance of the initial designs, and vice-versa. Recognising this undesired behaviour will lead to a careful treatment in the later stages of the MDAO workflow selection process where the preferences of the workflow designer play a determinant role. Finally, precision measures the spread of the objective function across multiple identical runs. Because the optimisation algorithms are non-deterministic heuristics, precision is a meaningful criterion. All the criteria are measurable and expected to capture differences between the alternatives.

Decomposability: By design, the tree of criteria built with the top-down approach (Fig. 4.9) is decomposable.

Non-redundancy: To verify the level of redundancy between the criteria, it is appropriate to make a correlation plot of their values and show their Pearson correlation coefficients. This plot and the Pearson coefficients are shown in Fig. 4.10.

Most of the criteria have low pairwise correlation coefficients which is a evidence of their non-redundancy. However, it is critical to focus on the Pearson coefficients of robustness with precision and improvement.

Indeed, improvement and robustness have the highest correlation ($r = -0.75$), followed by robustness and precision ($r = 0.63$). Since improvement and precision do not have a direct causal relationship and are not strongly correlated to each other, the common cause for their correlation with robustness is the low sensitivity of the analysis block. A flat LCOE response surface due to the poor accuracy of the analysis block can be responsible for a higher precision value, instead of due to the virtues of the optimisation algorithm. Evidently, the lack of sensitivity of an inaccurate analysis block is also present at the optimal design. A good value in the robustness criterion, in this case, can mean poor accuracy too, rather than a robust design. An inaccurate LCOE function can have two features that impact its sensitivity. On the one hand, the correlation between improvement and robustness can be a consequence of the flatness of an inaccurate LCOE function. On the other hand, the correlation between precision and robustness can be a consequence of the smoothness of an inaccurate LCOE function, because the optimi-

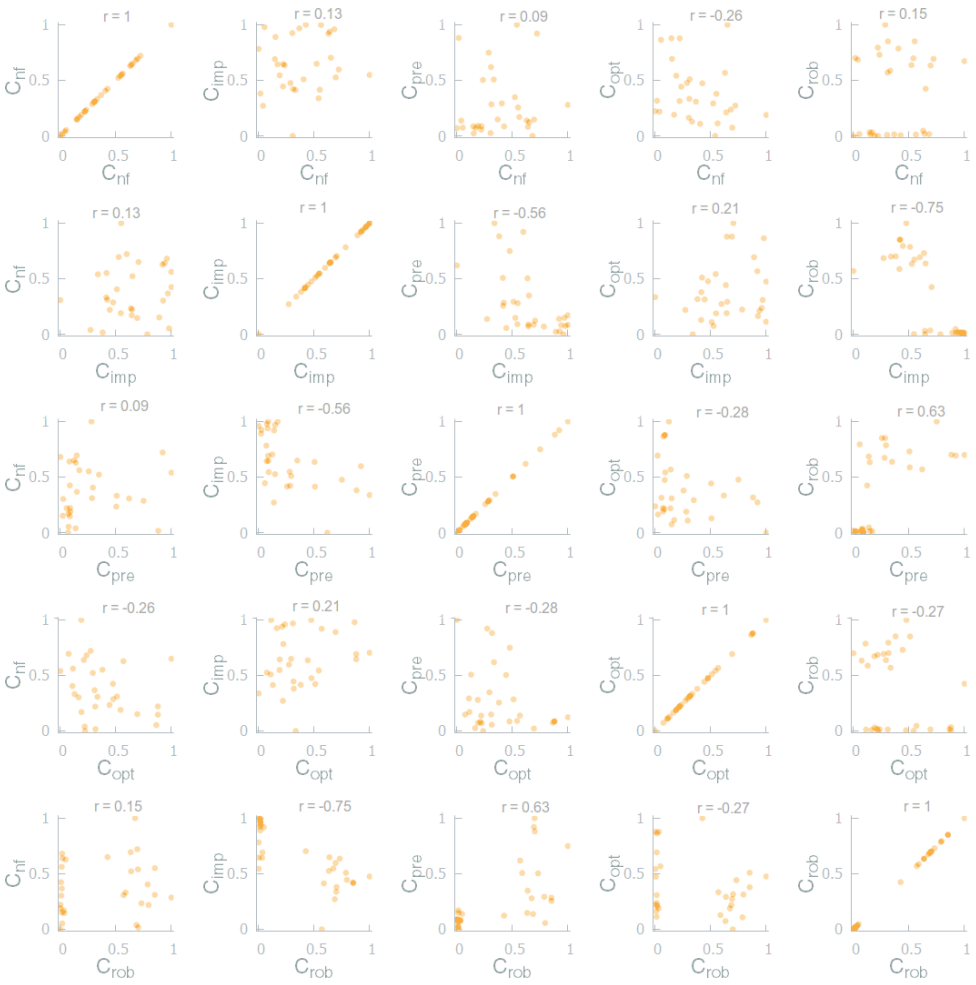


Figure 4.10: Correlation plot of the values of the criteria for all alternatives.

ser may be steadily moving towards the optimum, and once there, it happens that the function is locally flat.

Notwithstanding, correlation does not always imply causality and therefore, redundancy. Because the correlated criteria result from non-redundant aspects of the MDAO workflow, these are all kept.

In addition, improvement is defined by the distance between the performance of the best initial design of the population and the performance of the optimal design found. Therefore, improvement must be, at least partially, redundant with the optimality criterion. By acknowledging this and the fact that they both enable different points of view of the evaluation of MDAO workflows, they will both be kept for the following activities. Because the presence of an alternative in the Pareto set of non-dominated solutions should not be affected by the redundancy of the metrics, these correlations will be dealt with in

the final stage of the implementation, where the solutions in the Pareto set are reduced and ranked.

Size: Executing the MCDA with five criteria to evaluate and rank 20 MDAO workflows is considered a feasible problem to solve.

The metrics of the criteria are shown next. Note that because the Pareto front will be found with a sorting algorithm, these need not be normalised at this stage. The stopping criterion of all algorithms is when the LCOE has not improved in the last 1,000 iterations. The tolerance for improvement in this work is zero.

Optimality: This metric (C_{opt}) is simply the LCOE of the optimal design, calculated with the most accurate analysis block defined previously in §4.4.1. Each MDAO workflow is run three times and the best value is recorded. The metric reads:

$$C_{opt} = LCOE(\hat{x}). \tag{4.9}$$

Robustness: To measure this criterion the design variables of the optimal solution ($\hat{x}$) are perturbed five times according to certain rules and the average of the absolute difference between the LCOEs of the perturbed design vectors and the LCOE of the optimal design vector is recorded. The rules to calculate the perturbation Δx are the following: wind turbines and offshore substations positions are moved randomly by up to one rotor diameter, number of turbines is changed by randomly subtracting or adding one turbine, number of offshore substations is kept equal and electrical cable capacities are modified randomly by $\pm$ 1. The metric C_{rob} is expressed as:

$$C_{rob} = \frac{\sum_{i=1}^{5} \mid LCOE(\hat{x}) - LCOE(\hat{x} + \Delta x) \mid}{5}. \tag{4.10}$$

Number of function calls to convergence: This metric (C_{nf}) is the number of times the analysis block has been called when the stopping criterion is met. Because every run is limited to calling the analysis block 6000 times, if the optimiser has not satisfied the stopping criterion at the end, the metric will take the value 6,000. This metric is measured three times and averaged across the runs. It is written as:

$$C_{nf} = N_{calls} \text{ s. t. } LCOE^{n} = LCOE^{n-1000}. \tag{4.11}$$

Improvement: The metric is defined as the relative difference between the LCOE of the final optimal design and the best LCOE found at the first iteration ($LCOE_1$). This metric (C_{imp}) is measured three times and averaged across the runs. Its mathematical expression is:

$$C_{imp} = \frac{LCOE_1 - LCOE(\hat{x})}{LCOE_1}. \tag{4.12}$$

Precision: This metric (C_{pre}) is the standard deviation of the optimal LCOE values found across three runs. It reads:

$$C_{pre} = \sigma(LCOE(\hat{x})_i). \tag{4.13}$$

4.5.2. SCORING THE ALTERNATIVES

Five optimisation algorithms: Genetic (GA), Differential Evolution (DEA), Particle Swarm Optimisation (PSO), Simulated Annealing (SA) and a random search (RS), are coupled to the six analysis blocks found in phase 1: 1A, 2B, 2C, 2D, 1E, and 2E. In total, there are 30 MDAO workflows to be compared.

The metrics are normalised with respect to the range of values across all alternatives, so the alternatives in the Pareto front can be ranked using the distance to the utopia point. Zero is therefore assigned to the best performing MDAO workflow and 1 to the worst.

Figure 4.11 shows the scores across all criteria of all the MDAO workflows with a colour range, with alternatives ordered by their optimisation algorithm and by their analysis block.

To interpret the results, it is useful to mention the differences between the analysis blocks. These are listed in Table 4.9.

Table 4.9: Differences between the models and number of sampling points of the analysis blocks.

Analysis block ID	Wake-merge	Wake-added turbulence	Electrical collection design	Wind speeds sampled	Wind directions sampled
1A	MAX	Q	POS	21	12
2B	RSS	DR	POS	3	18
2C	SUM	Q	POS	17	18
2D	SUM	Q	EW	23	12
1E	SUM	F2	EW	3	30
2E	SUM	DR	EW	3	36

A few patterns are recognisable on the left-hand side of Fig. 4.11. First of all, the SA and RS algorithms converge with the least number of function calls. These two optimisers move about the design space randomly, and so as they advance, the probability of finding a better solution becomes smaller, while the probability of moving into infeasible designs is very large. In this criterion, the GA has the worst score, because the only way an individual can become infeasible is by mutation, which has a probability of 1%. Let us remember that feasibility is enforced by randomly reinitialising the vector of an infeasible design, and thus infeasibility leads to a more random search. Therefore, most of the solutions remain feasible, and the crossover operator provides new feasible and improved solutions for longer. Remember, the RS algorithm is a dummy method that is not properly used for optimisation, so the fact that it performs well in one criterion, does not mean it is a promising solution, nor that the function-calls criterion is reliable for assessing optimisation algorithms.

The DEA and GA algorithms seem to achieve the worst improvements. This is an indication that although the GA continuously improves the solution, the improvements are small. The DEA algorithm also incurs many feasibility violations, a probable cause for the small improvements seen.

The RS and DEA algorithms score the highest in the precision metric. There is little variation in the LCOE of the optimal designs under identical conditions. However, these

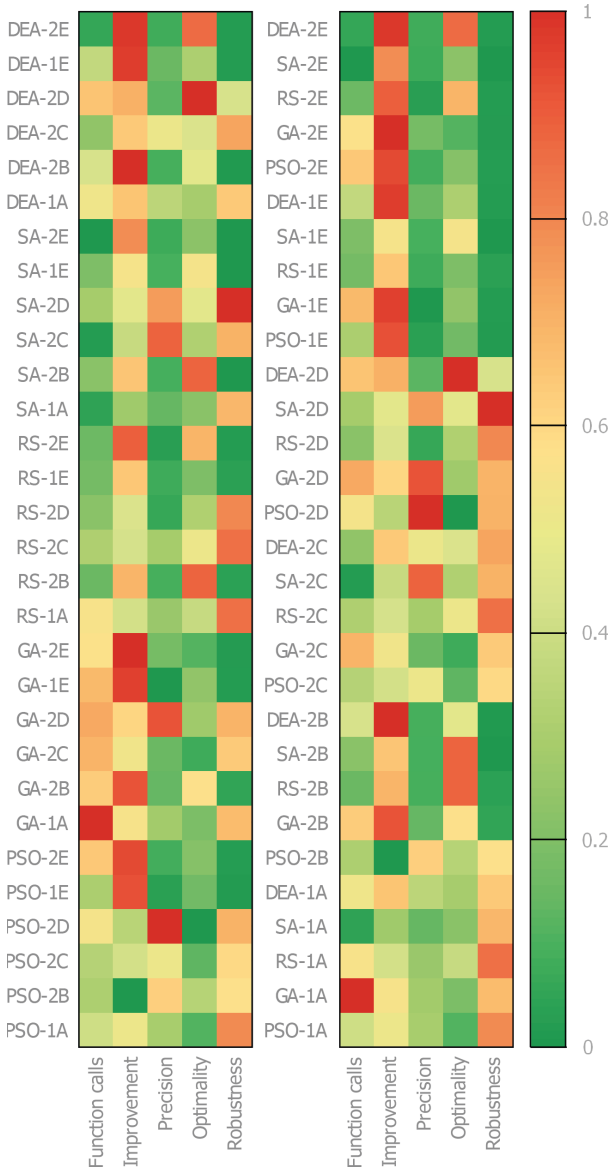


Figure 4.11: Heatmap of the scores of all the MDAO workflows across all the criteria. The left hand heatmap organises the workflow by their optimisation algorithm whereas the right hand heatmap organises them by their analysis block. Green represents a better criterion value, while red represents a worse one.

two algorithms have the most randomness, and the optimal values they consistently find is driven by the shape of the LCOE hypersurface. The LCOE function seems to be a very flat hypersurface with narrow optimality troughs.

With regards to optimality, it is the better-established PSO and GA algorithms that

score the highest. Every generation of the GA algorithm is composed of 120 individuals. It can be stated that by having a large population, GA covers a large sub-space of the design space, getting a head start with respect to other algorithms. The PSO algorithm is well known for its capabilities of exploring large areas of the design space and consistently moving towards the best solution found by all the particles in the swarm. The LCOE function is also highly multimodal, which makes all the algorithms converge to local optima.

Looking at the right-hand side of Fig. 4.11, it can be noted that analysis blocks 1E, 2E and 2B are responsible for the low improvement of all the optimisation algorithms coupled to them. These three analysis blocks sample the Weibull distribution at only 3 wind speeds and their associated LCOE errors are 6.69, 6.77 and 7.97 €/MWh, whereas the remaining three alternatives have lower values of 1.53, 5.15 and 1.47 €/MWh. This strongly limits the sensitivity of AEP with respect to the wind farm layout, and so the optimiser benefits less from manoeuvring the layout.

As expected from the correlation coefficients shown above, the robustness follows a completely opposite pattern of the improvement. Analysis blocks 1E, 2E and 2B score the highest values in robustness.

4.5.3. SORTING THE ALTERNATIVES

To deal with the pairwise correlations, the robustness criterion is dropped. This avoids double counting inherent characteristics of the underlying LCOE function of the analysis blocks during the decision-making process. The objective of this phase is to select the best-performing MDAO workflows altogether, and not only optimisation algorithms.

Hence, using the four other criteria for optimisation algorithms, the set of non-dominated solutions can be found using the ϵ-sorting algorithm [107]. Of the 30 original alternatives, 16 are non-dominated, and are listed in Table 4.10. The optimisation algorithm that appears the most times is PSO and the most frequent analysis block is 2E. The DEA algorithm does not belong to the Pareto front at all.

Table 4.10: Combinations of analysis blocks and optimisation algorithms that belong to the Pareto front.

Optimiser / Analysis block	Particle swarm optimisation (PSO)	Genetic algorithm (GA)	Random search (RS)	Simulated annealing (SA)	Differential evolution algorithm (DEA)
1A	Yes	—	—	Yes	—
2B	Yes	—	Yes	—	—
2C	Yes	Yes	—	Yes	—
2D	Yes	—	Yes	—	—
1E	—	Yes	Yes	Yes	—
2E	Yes	Yes	Yes	Yes	—

4.5.4. REDUCTION OF THE PARETO FRONT

If the number of MDAO workflows in the Pareto front is still too large for the workflow designer to make a decision, the concepts of the minimum distance to the utopia point

and average ranking can be used again to reduce the number of alternatives and alleviate the problem.

In §4.5.2, it was found that the inaccuracy of analysis blocks 1E, 2E and 2B was not counteracted by any other criterion and performed well under the criteria used in phase 1. It was also discussed that the inclusion of a sensitivity criterion would have filtered out these alternatives. Alternatively, a constraint on the accuracy of the selected analysis blocks could have also avoided carrying these three to the second phase, but it was decided to reduce the Pareto front without any subjectivity from the workflow designer.

As it has become evident, analysis blocks 1E, 2E and 2B are not fit for wind farm layout optimisation. Sampling the Weibull distribution at 3 wind speeds leads to very inaccurate analysis blocks which negatively influence the optimisation problem.

Therefore, in this case study, the nine MDAO workflows that couple analysis blocks 1E, 2E and 2B are discarded. It is acknowledged that this should have been an outcome of the guideline, and it proves that more than one iteration of the implementation can help detect issues like this.

Figure 4.12 shows the level diagrams of the Pareto front with the average ranking on the vertical axis. Optimisation algorithms are represented by colour, and the analysis blocks by marker symbol. The discarded alternatives are shown, nevertheless, with gray dots.

In this phase, to exemplify a real decision analysis done by a workflow designer, preferences are assigned to the criteria and the top ranked alternatives are discussed.

If the workflow designer is interested in the MDAO workflows that provide the designs with the lowest LCOE, then good options would include PSO-2D, GA-2C and PSO-2C. To achieve lower LCOEs, these MDAO workflows have to call the analysis block more times, and would also have to be run several times to compensate for their low precision.

All solutions fall within a short range of the improvement metric, so it is no longer an informative criterion.

Focusing on the number of function calls, this criterion benefits the simulated annealing (SA) optimisation algorithm coupled to the 1A and 2C analysis blocks, at the cost of other criteria.

If the workflow designer is after precision, then the GA-2C, SA-1A are the best alternatives. In this regard, the dummy random search algorithm has the highest score, but at a cost in optimality.

It is concluded that from the original 30 alternative MDAO workflows, 9 are discarded outside of the guideline by the workflow designer due to a shortcoming of the implementation, and 7 non-dominated options can cater to different relative preferences between the criteria.

4.6. RESULT OF THE IMPLEMENTATION

Within the limits imposed by the chosen criteria, there is evidence that an MDAO workflow for the multidisciplinary design optimisation of offshore wind farms would better serve its utility if it is composed of the following: the Jensen wake model, the sum wake-merging model, the Quarton wake-added turbulence model, the Esau-Williams electrical collection design heuristic, the TeamPlay support structure sizing tool, the surrogate wind turbine power and thrust coefficient look-up tables made with FAST, sample the

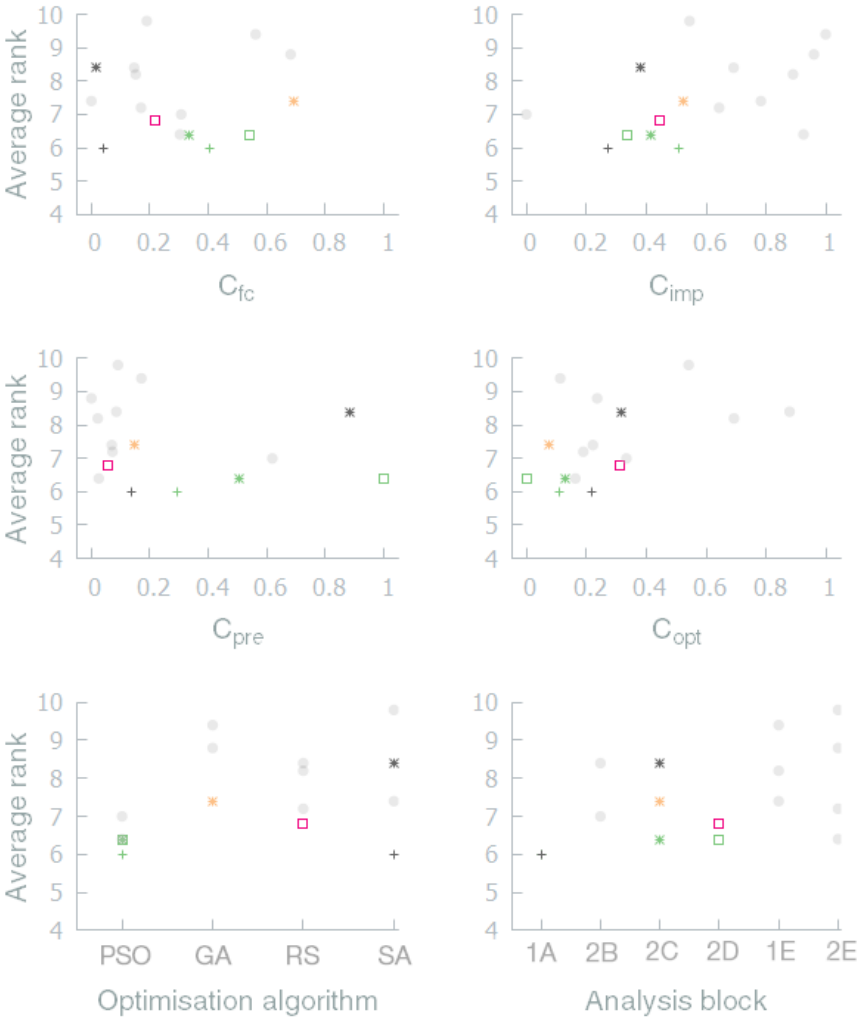


Figure 4.12: Level diagrams of the Pareto front of MDAO workflows where the vertical axis represents the averaged rankings per criterion. Colours represent the optimisation algorithm coupled, and the shape of the markers represents the analysis block coupled.

Weibull wind speed distribution at over 20 points, and sample the windrose between 12 and 18 wind direction sectors.

Moreover, because optimality is typically the most sought-after virtue of optimisation algorithms, the particle swarm optimisation and genetic algorithms satisfy this desire better than the simulated annealing and differential evolutionary algorithms, and without sacrificing much in the improvement and precision criteria.

4.7. DISCUSSION

This case exemplifies that the guideline can be implemented and the MCDA problem solved in a matter of days when the alternative MDAO workflows can be programmatically instantiated and run.

The objective of the guideline is to systematically, quantitatively and objectively simplify the decision to a more tractable problem. The guideline selects the set of solutions that efficiently solve the trade-offs and provide good compromises, and enables the designer take a lower risk by choosing an alternative from that set that maximises certain criteria while sacrificing the least on others.

The assumptions, strengths, weaknesses and challenges of the guideline, together with suggestions to deal with them are discussed next.

Guideline The guideline is most useful when the use case is detailed enough to capture what are the desired treats from the MDAO workflow.

Similarly, a workflow designer would benefit most from this guideline if the outcome of the use case is sufficiently critical to a research or development project, to justify the computational cost incurred in implementing it for selecting a set of best-performing MDAO workflows. Likewise, the MDAO workflows selected should be expected to be the workhorse throughout the project, to outweigh the cost of selecting it by using this guideline. In other words, this guideline is not meant to select an MDAO workflow that will be used only once.

In addition, only if the complete set of alternative number of analysis blocks can not be evaluated in a reasonable amount of time, then this guideline has a reason to exist. Running the multiobjective optimisation would be an excessive effort if all the alternatives could be scored against all the criteria.

At the last stage of the guideline, as with any other multi-criteria decision analysis problem, the decision maker must take a more active role, supported by expertise, experience, data and the outcome of this guideline, to choose a final single or few MDAO workflows that comply with the predefined requirements.

The MCDA suggested by the guideline can be revisited and iterated upon as many times as the workflow designer needs to feel confident on the results.

The guideline avoids the workflow designer from making catastrophic decisions based on intuition while also avoiding becoming overwhelmed by the great amount of analyses and information involved in choosing the optimal MDAO workflow.

Phase 1: analysis block selection One assumption of the first phase is that all the available analysis blocks are feasible. This means that all connections between different models can be made programmatically, and that no inputs are left unconnected.

Definition of the criteria The selection of the criteria and play the most dominating role of the entire guideline. Because an ill-posed problem leads to a useless, untrustworthy solution, the criteria have to represent the preferences of the workflow designer and have distinguishing power between the alternatives.

One of the roles of the guideline is to help detect the criteria that provide more meaningful solutions to the problem of choosing an MDAO workflow. In this implementation, a critical criterion measuring the sensitivity of the analysis blocks was missing, and the repercussion of this decision was that the optimisation algorithms were negatively affected by some underlying unrealistically flat LCOE functions.

Dealing with redundant criteria is a challenge. The Pearson correlation coefficient establishes the strength of linear correlations only. Correlation plots should reveal any other strong non-linear correlation. Although the Pareto front of alternatives should not be affected by redundant criteria, their ranking is biased towards alternatives that outperform the others on correlated criteria.

The guideline does not suggest ways to define the metrics of the criteria. Similarly to the criteria themselves, ill-defined metrics can break the utility of the alternatives selected by the guideline. It is key, however, that these metrics are normalised, so the weighted sum reflects only the relative importance between criteria and not their magnitudes. The MOPSOC algorithm, in particular, works by minimising the metrics, so non-decreasing functions should be used, where higher values represent worse solutions along that criterion.

Iterating on the application of the guideline also helps refine the criteria tree and metrics that lead to a truly useful solution.

Formulation of the multiobjective optimisation problem The guideline does not suggest a method to formulate the multiobjective optimisation. However, the choice of objective functions and constraints does have an impact on the results. A constrained optimisation places more effort to search in the feasible regions of the Pareto front, but the workflow designer has to make *a priori* decisions on the thresholds of some criteria, which is a difficult task.

MOPSOC algorithm The higher the dimension of the criteria space, the more iterations the MOPSOC algorithm will need to traverse the Pareto front with enough granularity. The size attribute of the criteria tree should help limit the number of criteria used, and to discard beforehand those deemed less relevant.

Naturally, the MOPSOC algorithm is sampling the analysis block space, so the workflow designer should temper expectations of the speed of the algorithm when sampling computationally expensive alternatives.

While the MOPSOC algorithm guides the search towards the unknown true Pareto front of the universe of analysis blocks, its probabilistic nature implies that its output will be different at every execution. However, the non-dominated alternatives provided by different executions should perform similarly.

Reduction of the Pareto front The approximated Pareto front might include hundreds of alternatives, although this limit can be preset by the workflow designer. In any case, while a large Pareto set is still a great improvement to the initial universe of alternatives, it might be overwhelming to choose a smaller set of solutions from it.

The guideline does not recommend a way to rank the alternatives in the Pareto front. This implementation shows a few methods, amongst many other existing ones.

If the decision-maker has no preferences, then ranking the solutions by their distance to the utopia point yields the solutions for which the loss along all criteria is minimal. This distance, however, can be negatively affected by the normalisation procedure of the metrics. Alternatively, ranking the alternatives per criterion is independent of the magnitude of their values, and averaging the rankings is an indication of the consistency of the strength of a particular analysis block.

The guideline aims at providing a set of solutions that objectively dominate the rest. Nevertheless, it is acknowledged that frequently, the decision-maker will have preferences over the criteria. These can be introduced in the form of a weighted distance to the utopia point, or a weighted average of the rankings per criterion.

5

Validation of the guideline

All models are wrong but some are useful.

George Box[87]

A theoretical guideline for MDAO workflow selection has been proposed and implemented once to demonstrate the practical aspects associated to it.

This chapter now reports the validation study of the guideline. Its goal is to increase the confidence on the usefulness of the guideline to select better-performingMDAO workflows.

Validation is enforced through logical argumentation for the unbiased nature of each of the activities suggested in the guideline, and controlled experiments that prove the correct functioning of the MOPSOC and ε*-non-dominated sorting algorithms under different scenarios.*

5.1. INTRODUCTION

The purpose of this chapter is to increase the confidence of workflow designers in the guideline. The validation of the guideline is done by progressively building trust on the activities suggested. Trust is represented in this case by proving that the activities are unbiased and are reliable. The activities should therefore provide meaningful results consistently across different scenarios [110].

The validation aims to answer the question: Does the guideline help the workflow designer make better decisions and build better MDAO workflows? The aim is thus to prove that the guideline is a good method for selecting MDAO workflows, and not that it is the best method.

Three methods are used to validate different parts of the guideline: informed argumentation, experimental set-up, and comparing with other selection methods [10]. A valid informed argument should include three propositions: the claim, the data or evidence, and the warrant. The claim is the disputed conclusion of the argument, e.g. "An apple a day keeps the doctor away", the data or evidence sustains the claim, e.g. "Apples are a source of healthy nutrients and its consumption can prevent strokes and lowers LDL cholesterol", and the warrant acts as a bridge between the data and the claim, e.g. "Fewer strokes and low LDL cholesterol lead to a reduction in medical attention and prescriptions" [111]. An experimental set-up is analogous to laboratory tests, where conditions are controlled and the effect of every parameter can be studied in near isolation. Finally, the results of the guideline are compared with the outcome of other informal methods that workflow designers might use in the absence of a formal approach for MDAO workflow selection.

The claim of the validation study is the following:

The guideline is useful for aiding the selection of the set of best-performing MDAO workflows.

The next sections report the evidence gathered, and its implication on the validity of the guideline is discussed.

This validation is only concerned with the activities of the guideline where there is a concrete suggestion for the workflow designer to follow.

Three levels of the guideline are assessed for its validation, from top to bottom: the guideline in its entirety, phases 1 and 2 in isolation, and the activities defined in each phase. Following Fig. 5.1, **V** refers to the validation process of the overall guideline in §5.2 by informed argumentation. **V1** and **V2** indicate the validation processes of phase 1 in §5.3 and phase 2 in §5.4, respectively, where their validity with respect to the set of alternatives is argued, and phase 1 is compared with other analysis block selection methods. **V11** and **V21** denote the validation by informed argumentation of the activities to define the criteria, presented together in §5.3.2. **V13** stands for the validation process of the MOPSOC algorithm using experimental set-ups in §5.3.3. **V22** is the process to validate the activity of sorting the MDAO workflows in phase 2 using an experimental set-up, described in §5.4.2. Finally, the activities for formulating the multiobjective optimisation of phase 1 and for reducing and ranking the solutions in the Pareto front

of phases 1 and 2 are not validated, as the workflow designer is free to choose the appropriate techniques. The impact of the choices are discussed, however, in §3.4.2 and §3.4.4.

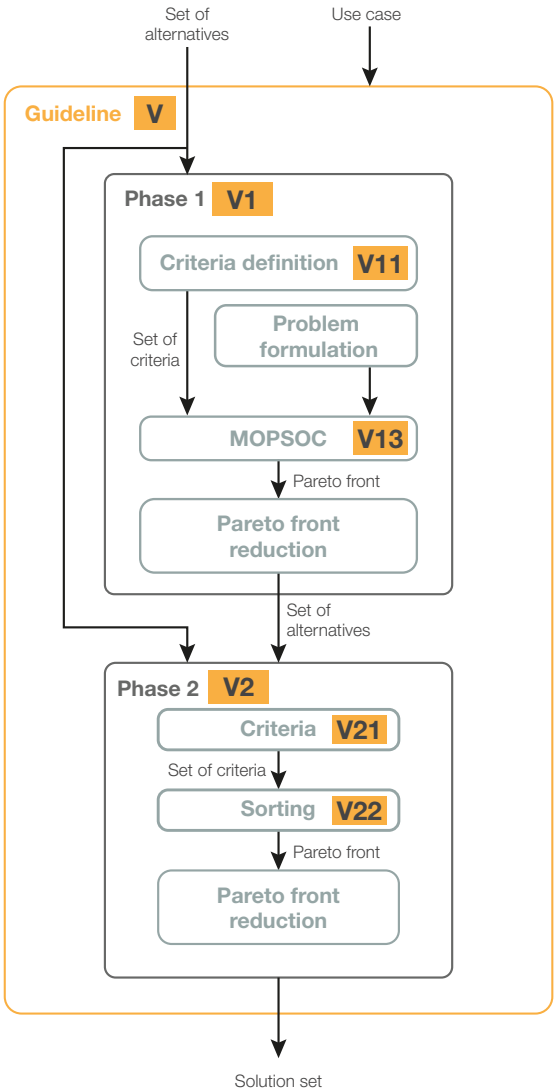


Figure 5.1: Outline of the validation study.

5.2. VALIDATION OF THE OVERALL GUIDELINE

This section presents the validation activity of the entire guideline, marked with **V** in Fig. 5.1, discussing the concepts behind it and the influence of the use case.

"If humans were always successful in instinctively making optimal decisions on their own, there would be no need for formal decision theory" [112]. This note by Thurston et. al., implies that workflow designers, in their human condition, are not always successful in building optimal MDAO workflows, and a formal decision theory for that purpose is necessary.

While a formal approach could overwhelm the workflow designer and lead to a stall in the decision-making process, the guideline is meant to support a semi-automated decision analysis. This approach reduces the burden of analysing all alternatives, and provides the workflow designer a much smaller set of solutions to choose from and information about the performance of these workflows.

The guideline uses the multiobjective programming MCDA paradigm for its ability to approximate the set of non-dominated solutions or Pareto front. All the optimal solutions that satisfy any combination of preferences of the workflow designer are elements of the Pareto front. The guideline proposed in this work avoids the need for subjective preferences *a priori* that affect the information made available to the designer.

Theoretically, the guideline has therefore the ability to yield all the best-performing solutions. In practice, however, all multiobjective programming techniques only approximate the Pareto front. This is due to time limitations and the stochastic nature of the optimisation algorithm.

The fulfilment of the requirements of the guideline presented in §3.2 is discussed next.

The guideline is deemed to be **concrete** because there is no ambiguity in the description of the activities. The activities must be followed in a predefined order and none of their outputs are left to the interpretation of the workflow designer. The outputs of the suggested activities are: a single list of useful criteria, a single multiobjective optimisation formulation, a set of non-dominated alternatives and a ranked subset of the Pareto front of MDAO workflows.

The guideline is **flexible** because it allows the workflow designer to have an active role in the selection process. The criteria are not predefined, but the workflow designer is given a framework for developing a list of criteria for different use cases and sets of alternatives. Additionally, the formulation of the multiobjective optimisation is also not fixed, and it may instead cater to the preferences of the decision maker. Finally, the workflow designer is allowed to choose freely the technique for reducing the Pareto front.

The guideline is **objective** because it makes no assumptions about the preferences of the designer on the criteria. It simply looks for non-dominated solutions and presents them to the designer to apply preferences at a very late stage, and with much fewer alternatives. The MOPSOC algorithm is the only known multiobjective optimisation algorithm for categorical variables, which justifies its inclusion in the guideline. The ϵ-non-dominated sorting algorithm finds the exact Pareto front of any given set, with a user-defined tolerance for the maximum distance between elements across all criteria.

Finally, the guideline is **simple** because the steps are clearly defined, there is no information withheld from the workflow designer for maximum transparency, and it comprises a sequence of eight activities that always yield a solution. All the steps in the implementation of chapter 4 are fully described, and they all follow the exact instructions

of the activities of the guideline in chapter 3. There is a one-to-one correspondence between the implementation and the guideline.

The concepts of the guideline that need a critical reflection regarding their effectiveness are treated below.

This guideline proposes the separation of the selection of MDAO workflows into two phases: the selection of analysis block and the selection of driver algorithm. It is claimed that the Pareto front of all the possible MDAO workflows contains the Pareto front of the MDAO workflow found by separating the selection into two phases. The separation of phases is equivalent to finding the Pareto front of MDAO workflows with constraints on the performance of its analysis block. The drawback of the separation of the phases is that the entire Pareto front of MDAO workflows is not found. Nonetheless, the analysis block is the most influential part of an MDAO workflow. A bad driver algorithm coupled to a good analysis block yield much more useful results than a bad analysis block coupled to a good driver algorithm. Furthermore, in chapter 4 roughly 6,000 analysis blocks were evaluated by the MOPSOC algorithm. Coupling each of these to a driver algorithm and evaluating the performance of the resulting MDAO workflow would take an unreasonably large amount of time due to the iterative nature of the optimisation. The separation of phases enables the benefits of implementing the guideline to outweigh its cost.

Other problems may arise from the definition of the criteria using separated phases. If the criteria to evaluate the analysis block are redundant with the criteria used for evaluating the driver algorithms, the analysis blocks chosen may bias the selection of the driver algorithm.

The feasibility of implementing the guideline is not always guaranteed. The MOPSOC algorithm requires the automatic generation of analysis blocks, a task that can be difficult to achieve and impose restrictions in the set of alternative analysis blocks without full access to the programmatic instantiation of models with varying parameters.

5.2.1. INFLUENCE OF THE USE CASE

The use case driving the design or selection of MDAO workflows influences only the activities for defining the criteria and for reducing the Pareto front by applying the preferences of the designer, if any. The rest of the guideline is blind to the use case.

Any use case would likely require MDAO workflows to be evaluated along several criteria. Although different use cases might have the same sub-objectives, they may also have very different requirements, which could translate into different criteria. For example, it could be argued that an investor stakeholder would have no preference over the execution time of the analysis block, as there are qualities of the MDAO workflow of much higher priority. In other words, the criterion execution time is not meaningful to the workflow designer, and thus has no power to discern between alternatives. On the contrary, any certification analysis would have a computational budget allocated to it, making execution time a criterion of interest to the workflow designer.

Similarly, the use case may have implications on the preferences of the designer, included at the last step of the selection process. While chapter 4 exemplified a case with no preferences on the criteria, the workflow designer may have non-zero preferences on

some criteria. The workflow designer will typically shift his preferences according to the use case. For example, the execution time of the analysis block will attract more interest relative to other criteria the more times the analysis block is needed to be run.

The guideline is thus able to handle different use cases by design. It is a task of the workflow designer to capture the requirements of the use case, and use them to drive the instantiation of the guideline, to eventually reach solutions that satisfy the requirements.

5.3. PHASE 1: ANALYSIS BLOCK SELECTION

The validation of phase 1 with respect to the set of alternatives, shown as **V1** in Fig. 5.1, is treated here. Then the activities of phase 1 are validated one by one, followed by a comparison of the selected analysis block using phase 1 of the guideline with other selection methods, and an assessment of the sensitivity of the output of phase 1 with respect to the criteria used.

5.3.1. INFLUENCE OF THE SET OF ALTERNATIVES

This section addresses whether phase 1 works for all kind of models and analysis blocks.

Notwithstanding the dependency of the performance of the MOPSOC on the metrics of the criteria, the greatest showstopper for the ability of MOPSOC of providing meaningful results is the execution time of the alternative analysis blocks.

Knowing in advance that the MOPSOC algorithm samples analysis blocks in the order of thousands, it would not be worth following this procedure for selecting an analysis block that will be used for less time than it takes selecting it.

With the computational power currently available to offshore wind farm developers and researchers it is infeasible to run MOPSOC with high-order physics models that require a high-density mesh, such as computational fluid dynamics or finite element methods. Moreover, analysis blocks that involve several solving cyclic-connections with non linear iterative solvers can also be prohibitively expensive to optimise with MOPSOC.

However, it is hypothesised that beyond the guideline proposed in this work, any MCDA method would struggle to select a set of better-performing analysis blocks from a universe of alternatives that couple very computationally expensive tools. While this hypothesis does not increase the value of the guideline, it highlights the need to continue the development of MCDA theories for selecting analysis blocks with high levels of sophistication. One possible line of research is the use of surrogate models as stand-ins for sophisticated high-order physics modelling tools.

It is therefore affirmed that the set of alternatives should have no implications on the validity of the guideline, but there may be practical limitations regarding execution time or the ability to automatically generate the analysis blocks.

5.3.2. DEFINITION OF THE CRITERIA

The validation activities **V11** and **V21** in Fig. 5.1 are discussed together here.

This guideline is based on the premise that the MDAO workflow used for solving a use case will be evaluated based on multiple criteria. It is better to choose (or design) the MDAO workflow by looking at all the criteria since its inception, instead of designing it for a single or few criteria and then judging its performance against another set of

criteria[112].

This is a critical activity, as the result of the entire guideline will be defined by the criteria. Analogous to how the solution of an optimisation problem is as good as the objective function, good MCDA with bad criteria will lead to meaningless solutions. Workflow designers need to solve the right problem.

The guideline assumes that a compromise between certain criteria exists, otherwise the criteria only reinforce the fact that a single solution is the absolute best alternative. In that case the problem becomes unidimensional and MCDA theory is no longer necessary. The guideline further assumes that criteria are not preference-complementary, i.e. that the value of one criterion must not depend on the value of another[110].

A multicriteria evaluation function can include criteria and binary constraints[112]. Binary constraints express whether the designer is willing to pay to exceed a certain value of a given performance indicator. Criteria differ from binary constraints in that a range of acceptability exists, and the designer has a preference over the range of the criterion, i.e. that more or less of that criterion is better.

Inspiration for criteria in the domain of computer simulations can be found in papers that attempt to quantify simulation fidelity[85,89]. Simulation utility is a trade-off between the (relevant) information acquired and the cost of acquiring it. Falkenhainer states the trade-off as having sufficient fidelity to answer a query and contain as little extraneous detail as possible to avoid wasting computational resources[113], while Radhakrishnan declares the usefulness of a simulation if it predicts reality accurately at a small cost[114]. Instead of using the concept of cost, Provan says model complexity can be the overarching sub-objective counteracting accuracy[115]. Harmon suggests criteria that evaluate the actual federation of simulations, in terms of their connections and simulated representations, as done in MDAO[90].

Regarding the metrics, non-linear evaluation functions are not trivial to determine and the process can be very time-consuming[112]. However, it is important for MCDA that the metrics are monotonically increasing, for transitivity and to correspond to the designer's preferences.

The scores of some criteria will depend on the environment, and for repeatability this should be taken into account when formulating the criteria and metrics. For example, the execution time of the analysis block will depend on the computational power available.

The guideline is blind to the criteria selected, and therefore works regardless of the criteria and metrics. As stated above, the guideline will undoubtedly yield useless answers, if the criteria or their metrics are meaningless.

5.3.3. MOPSOC ALGORITHM

This section reports the validation of the MOPSOC algorithm, shown as **V13** in Fig. 5.1.

The MOPSOC algorithm is validated using three sources of evidence. The first is the evaluation of the performance of the algorithm with multiobjective optimisation test functions. The second source of evidence is the comparison of the distribution of the Euclidean distances of the Pareto front found with the MOPSOC algorithm, with that of the Pareto front of a random sample. Third, the MOPSOC algorithm is run for the use

case defined in chapter 4 with different optimisation formulations, and the results are discussed.

TEST FUNCTIONS

The MOPSOC algorithm is partly validated with continuous test functions of which the Pareto front is known. The functions are adapted by discretising the search domain and randomising the order of the sample points.

The following multiobjective test functions have the same notation: f_i for the m objectives to be minimised, subject to h_j constraints:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\ f_i(x_1, x_2, \ldots, x_n), \\ &\text{subject to } h_j(x_1, x_2, \ldots, x_n) \leq 0. \end{aligned}$$

The results of the tests will be shown in graphs (Figs. 5.2-5.7), which show the Pareto front of the test functions with a yellow line and their approximation found by MOPSOC with a black line or dots. Furthermore, the dotted areas show the mapped search space of the objective functions found by randomly sampling 100,000 points in the function's original continuous domain. The dark gray areas show the feasible regions that fall within the constraint functions. For clarity, the objective functions spaces that are well-delimited are shaded with a dotted pattern.

All functions were optimised using 20 swarm particles and 300 time steps, which require 6,000 function evaluations. Each dimension of **x** is discretised into 200 intervals of equal length.

The next sections show the results for the following test functions:

1. Binh-Korn function
2. Chankong-Haines function
3. ZDT3 function
4. Poloni function
5. Fonseca-Fleming function
6. Viennet function

Binh-Korn function The Binh-Korn function [116] is chosen to prove the ability of MOPSOC to approximate a convex Pareto front of a constrained function (see Fig. 5.2). This function reads:

$$\begin{aligned} f_1(\boldsymbol{x}) &= 4x_1^2 + 4x_2^2, \\ f_2(\boldsymbol{x}) &= (x_1 - 5)^2 + (x_2 - 5)^2, \\ h_1(\boldsymbol{x}) &= (x_1 - 5)^2 + x_2^2 - 25 \leq 0, \\ h_2(\boldsymbol{x}) &= -(x_1 - 8)^2 + (x_2 + 3)^2 + 7.7 \leq 0, \end{aligned}$$

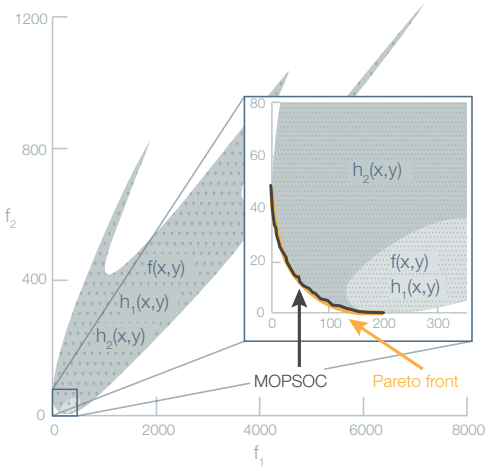


Figure 5.2: Pareto front of the Binh-Korn function approximated by MOPSOC.

and the search domain is:

$$0 \leq x_1 \leq 5,$$
$$0 \leq x_2 \leq 3.$$

The MOPSOC algorithm is able to approximate the Pareto front of the feasible regions. MOPSOC manages to find the narrow feasible region at the bottom.

Chankong-Haimes function The Chankong-Haimes test function[117] is also selected to show the ability of MOPSOC to approximate the convex Pareto front of a constrained function (see Fig. 5.3). This function reads:

$$\begin{aligned} f_1(\boldsymbol{x}) &= 2 + (x_1 - 2)^2 + (x_2 - 1)^2, \\ f_2(\boldsymbol{x}) &= 9x_1 - (x_2 - 1)^2, \\ h_1(\boldsymbol{x}) &= x_1^2 + x_2^2 \leq 225, \\ h_2(\boldsymbol{x}) &= x_1 - 3x_2 + 10 \leq 0, \end{aligned}$$

and the search domain is:

$$-20 \leq x_1, x_2 \leq 20.$$

MOPSOC is able to closely approximate the true Pareto front of the feasible region.

ZDT3 function The Zitzler-Deb-Thiele 3 (ZDT3) test function[118] is commonly used to benchmark multiobjective optimisation algorithms. The ZDT3 function is expressed by:

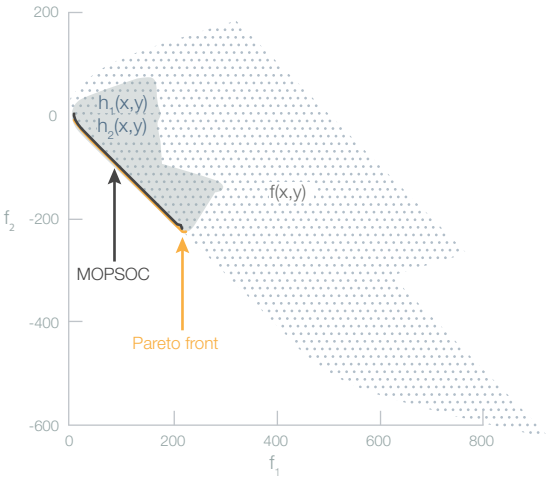


Figure 5.3: Pareto front of the Chankong-Haimes function approximated by MOPSOC.

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= x_1, \\
f_2(\boldsymbol{x}) &= g_1(\boldsymbol{x}) g_2(f_1(\boldsymbol{x}), g_1(\boldsymbol{x})), \\
\text{where:} & \\
g_1(\boldsymbol{x}) &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i, \\
g_2(\boldsymbol{x}) &= 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g_1(\boldsymbol{x})}} - \left(\frac{f_1(\boldsymbol{x})}{g_1(\boldsymbol{x})} \right) \sin(10\pi f_1(\boldsymbol{x})),
\end{aligned}
$$

and the search domain is:

$$0 \leq x_i \leq 1, \quad i \in [1, 2, \ldots, 7].$$

In this work, the dimension of the design vector, n, has been set equal to 7, instead of the more common $n = 30$, which would make MOPSOC's swarm particles live in a 200×30-dimensional space and converge extremely slowly. In spite of the reductions of the number of dimensions to make MOPSOC tractable to solve the ZDT3 test function, 7 dimensions is representative of many practical categorical functions. The random sampling of function ZDT3 (Fig. 5.4) shows that the density is much higher about $f_2 \simeq 4$ than in the vicinity of the Pareto front, and that MOPSOC is able to separate from the denser region and approximate the Pareto front better than by random sampling the function.

Poloni function The Poloni function [119] is used to prove the ability of MOPSOC to approximate a discontinuous Pareto front (see Fig. 5.5). This function is given by:

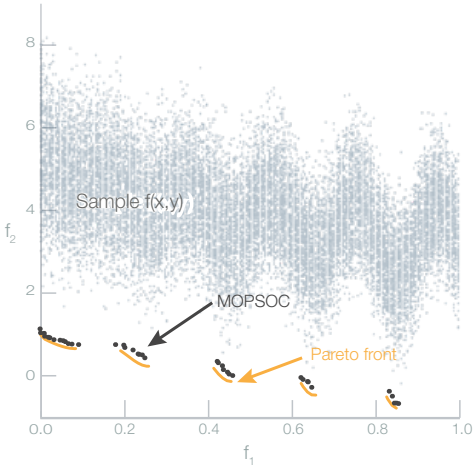


Figure 5.4: Pareto front of the ZDT3 function approximated by MOPSOC.

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= [1 + (A_1 - B_1(\boldsymbol{x}))^2 + (A_2 - B_2(\boldsymbol{x}))^2], \\
f_2(\boldsymbol{x}) &= (x_1 + 3)^2 + (x_2 + 1)^2, \\
\text{where:} & \\
A_1 &= 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2), \\
A_2 &= 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2), \\
B_1(\boldsymbol{x}) &= 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2), \\
B_2(\boldsymbol{x}) &= 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2),
\end{aligned}
$$

and the search domain is:

$$-\pi \leq x_1, x_2 \leq \pi.$$

MOPSOC approximates the true Pareto front of the function, even though it is discontinuous.

Fonseca-Fleming function The Fonseca-Fleming test function[120] is selected to show the ability of MOPSOC to approximate a concave Pareto front. The Fonseca-Fleming function is given by:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right], \\
f_2(\boldsymbol{x}) &= 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right],
\end{aligned}
$$

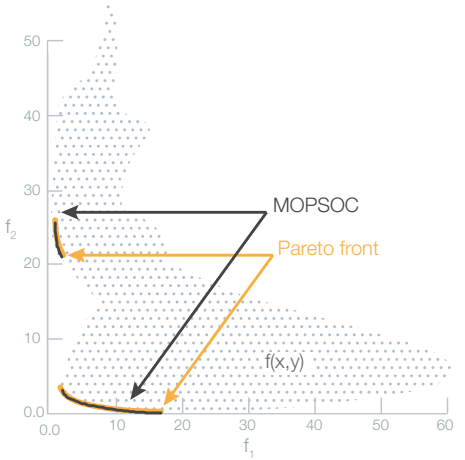


Figure 5.5: Pareto front of the Poloni function approximated by MOPSOC.

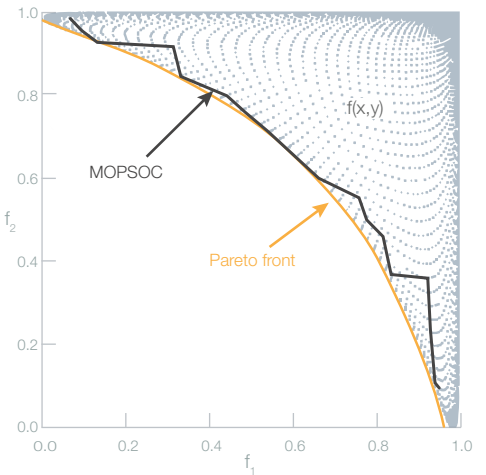


Figure 5.6: Pareto front of the Fonseca-Fleming function approximated by MOPSOC.

where the dimension of the design vector, n, is set to 3. The search domain is:

$$-4 \leq x_1, x_2, x_3 \leq 4.$$

It is worth noting in Fig. 5.6 that the Fonseca-Fleming function has a higher density at higher values of f_1 and f_2, and yet MOPSOC is able to move away and towards the Pareto front.

Viennet function The Viennet test function [121] is chosen to show the ability of MOPSOC to approximate a three dimensional Pareto front (see Fig. 5.7). The Viennet set of functions read:

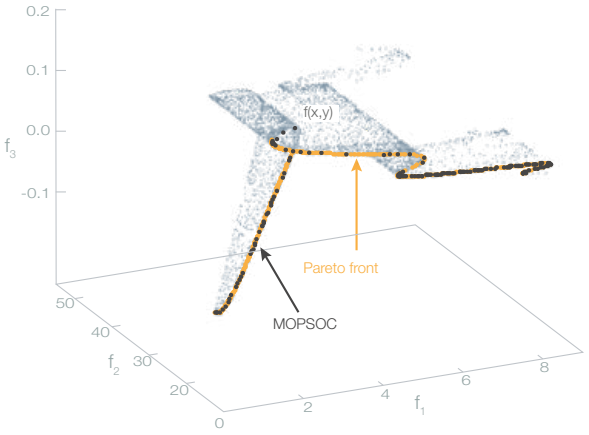


Figure 5.7: Pareto front of the Viennet function approximated by MOPSOC.

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= 0.5\left(x_1^2 + x_2^2\right) + \sin\left(x_1^2 + x_2^2\right), \\
f_2(\boldsymbol{x}) &= \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15, \\
f_3(\boldsymbol{x}) &= \frac{1}{x_1^2 + x_2^2 + 1} - 1.1\exp\left(-x_1^2 - x_2^2\right),
\end{aligned}
$$

and the search domain is:

$$-3 \leq x_1, x_2 \leq 3.$$

The Pareto front approximated by MOPSOC is a subset of the true Pareto front. The density of the non-dominated solutions, however, is not uniform. There is no mechanism in the MOPSOC algorithm to avoid clusters of non-dominated solutions.

The validation functions used in this study show that MOPSOC approximates Pareto fronts closely using only 6,000 function evaluations. In comparison, the ZDT3 function in its original continuous form is recommended to be optimised with at most 25,000 function calls[94].

RANDOM SAMPLE OF ANALYSIS BLOCKS

In this section the Pareto front approximated by the MOPSOC algorithm is compared to the Pareto front of a random sample of analysis blocks.

Figure 5.8 shows the minimum, first, second and third quartiles, and the maximum of the Euclidean distances of the analysis blocks to the utopia point, for the approximated Pareto front found with MOPSOC comprising 118 analysis blocks, for the Pareto front

of a random sample of the same size as the number of workflows evaluated by MOPSOC, and for the entire random sample. The utopia point is defined as the point where $C_{acc} = C_{time} = C_{det} = C_{os} = 0$. The distance is calculated using the normalised versions of the criteria. The median Euclidean distance of the Pareto front of six thousand randomly sampled analysis blocks is 0.9992, whereas the median of the set of non-dominated analysis block found with MOPSOC is 0.9464. The minimum distance found is 0.6061 for the random sample while MOPSOC is able to find a solution with a distance of 0.2947. This is an indication that MOPSOC is able to break away from the dense regions in the criteria space and move towards the true Pareto front of the universe of analysis blocks. The approximated Pareto front contains 119 analysis blocks, and the Pareto front of the random sample contains 105.

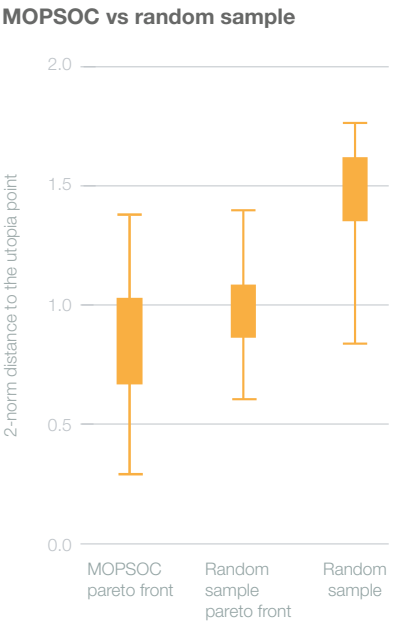


Figure 5.8: Minimum, quartiles and maximum of the Euclidean distances to the utopia point of the analysis blocks in the approximated Pareto front found with MOPSOC, of a random sample of 6,000 analysis blocks and the Pareto front of the sample.

OPTIMISATION FORMULATION

The goal of this section is to assess the robustness of MOPSOC for producing meaningful results under different optimisation problem formulations. Besides the formulation used in chapter 4 where all the four criteria are treated as objective functions, one and two objective functions are used here with three and two constraints, respectively.

One objective function Accuracy is the criterion to be minimised by the MOPSOC algorithm while complying with constraints on the execution time, detail and open-

sourceness criteria. Formally, the optimisation formulation is the following:

$$\begin{aligned} &\underset{W}{\text{minimise}} && \hat{C}_{acc}(W) \\ &\text{subject to:} && \hat{C}_{time}(W) < 0.2, \\ & && C_{det}(W) < 0.2, \\ & && C_{os}(W) < 0.2. \end{aligned}$$

Formulations with a single objective function produce a single optimal analysis block. In this case the solution has an accuracy metric of 0.07 (0.92€/MWh), 4.21 s of execution time (metric equal to 0.16), a detail of 0.03 and an open-sourceness of 0.2.

The resulting analysis block slightly dominates the Pareto fronts found with more objective functions. The MOPSOC algorithm is more efficient with less objective functions. This is partly due to the reduced weight-sampling frequency since there are a limited number of iterations and a fixed size of the swarm.

Two objective functions Accuracy and execution time are minimised in this formulation, with constraints in place for limiting detail and open-sourceness. This formulation is formally expressed as:

$$\begin{aligned} &\underset{W}{\text{minimise}} && \hat{C}_{acc}(W), \hat{C}_{time}(W) \\ &\text{subject to:} && C_{det}(W) < 0.2, \\ & && C_{os}(W) < 0.2. \end{aligned}$$

Figure 5.9 shows that by reformulating the original problem, the new Pareto fronts are farther from the utopia point than the unconstrained problem chapter 4. In addition, compared to the formulation reported in §4.4.2 where all four criteria are jointly minimised (projected in Fig. 5.9 onto the accuracy-execution time plane), the resulting Pareto front is a 2D subset of the full 4D Pareto front.

This exercise helps validate the robustness of the MOPSOC algorithm with respect to changes in the formulation of the optimisation problem. The approximated Pareto fronts comply with the expectations that the constrained optimisation will limit the lower values that execution time and accuracy can achieve and always produce feasible results, and that the unconstrained optimisations produce similar results. Nevertheless, the constraint-handling technique of the MOPSOC algorithm can be improved to allow greater exploration of the infeasible regions.

5.3.4. INFLUENCE OF THE CRITERIA

This section describes the validation of phase 1 of the guideline by solving the use case defined in chapter 4 with different sets of criteria and discussing the impact these have on the resulting Pareto fronts.

The MOPSOC algorithm is run with a single criterion, and certain combinations of two and three criteria. All the following where run with the same conditions as the implementation of chapter 4: 300 iterations with 20 particles in the swarm.

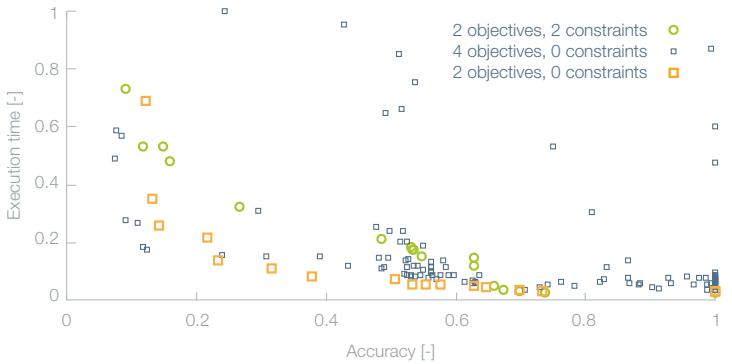


Figure 5.9: Scatter plot of the Pareto front when the execution time and detail criteria are jointly minimised. Three cases are considered: all four criteria minimised, accuracy and time minimised unconstrained, and accuracy and time minimised with constraints on detail and open-sourceness.

Accuracy only By optimising only accuracy and disregarding execution time, detail and open-sourceness, an error of 1.84% or 0.54 €/MWh is achieved. The optimal analysis block is very similar to the analysis block used as reference for the accuracy metric, coupling the most sophisticated models and a high number of sampling points. Furthermore, the accuracy of the analysis block found with MOPSOC is compared to the accuracies of 6000 randomly-sampled analysis blocks in Fig. 5.10. Two random samples are slightly more accurate, but MOPSOC breaks away from the denser regions of accuracy after only 17 iterations.

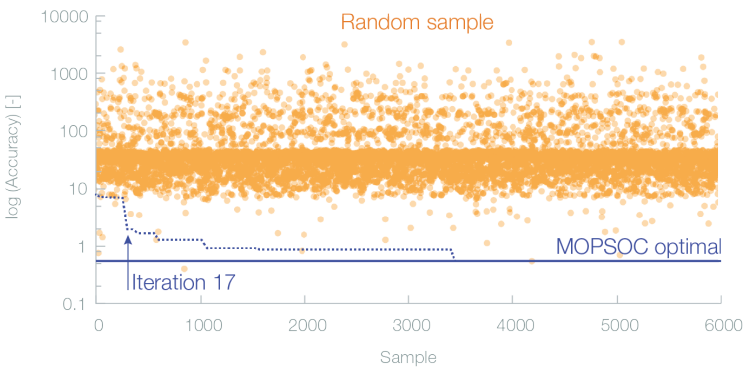


Figure 5.10: Accuracy of 6000 randomly-sampled analysis blocks (orange circles) and of the optimal analysis block found with MOPSOC (blue line). The dotted line shows the history of the accuracy of the best solution found by the MOPSOC algorithm.

Time only MOPSOC finds an alternative with 0.65 s of execution time when this is the sole criterion to be minimised. This analysis block couples all the very basic dummy models, with the lowest possible number of sampling points. The FAST wind turbine model is the only non-dummy model coupled, because it has the same execution time

as the dummy model, as it is a look-up table made from FAST simulations.

Detail only The metric for detail can only have discrete values and only five modules have models that contribute to the detail criterion. The other modules only have available models with the same level of detail. Only 768 workflows (0.11% of the possible analysis blocks) have the best detail value of zero. One of these is found by MOPSOC.

Open-sourceness only 27% of the possible analysis blocks have the best open-sourceness value of zero. It is thus relatively easy to find a best-scoring analysis block, and MOPSOC manages to do that.

Accuracy and time The Pareto front resulting from running the MOPSOC algorithm with the accuracy and execution time as objective functions shows that these are two competing criteria. The wake model and the number of sampling points are the most consequential choices for the accuracy and execution time of analysis blocks. Figure 5.11 shows that the position of the analysis blocks on the Pareto front approximated by the MOPSOC algorithm largely depends on the wake model coupled and the number of sampling points, as expected.

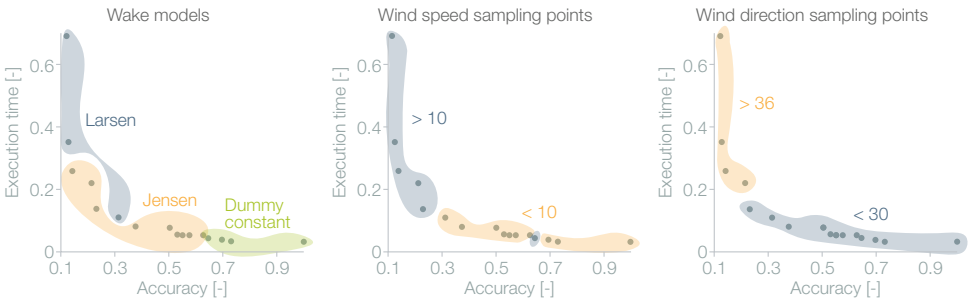


Figure 5.11: Scatter plot of the Pareto front when the accuracy and execution time criteria are jointly minimised. Colour represents the wake model coupled and number of sampling points.

Time and detail Detail depends largely on the presence of FAST and TeamPlay models. Neither of these have great implications on the execution time of the analysis block, and can be expected to appear in this Pareto front. Detail and time are competing criteria, where the sole responsible for analysis blocks belonging to either side of the Pareto front is the model for the support structure sizing module as seen in Fig. 5.12. All the analysis blocks in the Pareto front couple the FAST wind turbine model, since it increases the level of detail without affecting time. Because accuracy is not an objective nor a constraint, the dummy constant wake model appears often to decrease execution time without incurring in a loss of detail. The same happens with the low number of wind speed and wind direction sampling points.

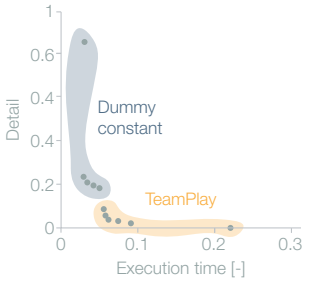


Figure 5.12: Scatter plot of the Pareto front when the execution time and detail criteria are jointly minimised. Colour represents the support structure sizing model.

Accuracy and detail Because the reference for accuracy is calculated using one of the most detailed and sophisticated analysis blocks, accuracy and detail are not competing criteria. The MOPSOC algorithm captures this behaviour as evidenced by the fact that this Pareto front only has two analysis blocks that are very close to each other and have high accuracy (errors of 0.95 € /MWh) and high level of detail (0.006). The difference between the analysis blocks in the Pareto front lies in the wake model. The Larsen model is slightly more detailed than the Ainslie 1D model, but marginally less accurate.

Accuracy, time and detail Analysis blocks with a higher level of detail, should have higher accuracy due to their sophistication, at the cost of higher execution times. Figure 5.13 shows the 2D projection of the three-dimensional Pareto front onto the plane defined by the normalised metrics for accuracy and execution time. The plot shows the isolines of the detail metric. Precisely as expected, the contour lines progressively reach more accurate and computationally expensive alternatives as the metric for detail decreases.

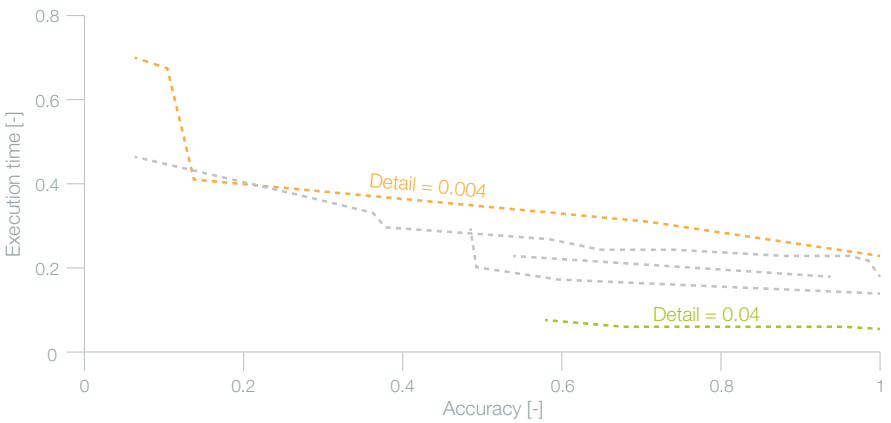


Figure 5.13: Two-dimensional plot of the three-dimensional Pareto front when the accuracy, execution time and detail criteria are jointly minimised. Contours of detail are shown in colour.

Criteria are the most defining elements in MCDA. This study has demonstrated that the MOPSOC algorithm, and consequently phase 1, yield entirely different solutions when different criteria are used. In particular, obtaining a 2D Pareto front using 2 objective functions provides much more information than running twice a single-objective optimisation. In that light, it is seemingly beneficial to use all the criteria and obtain as much information as possible, as phase 1 and the MOPSOC are perfectly able to handle them. It should be reminded, nonetheless, that the list of criteria should comply with the size attribute, to avoid placing much burden on the optimisation. The results of this exercise were expected and no non-trivial information was obtained from this particular set of alternatives, which helps validate the effectiveness of phase 1.

5.3.5. COMPARISON WITH OTHER SELECTION METHODS

Because a case could be made for quickly picking a sub-optimal analysis block, the weight of the potential benefits of implementing phase 1 of the guideline and enacting the solution against its cost is emphasised in this section. In particular, the implementation of phase 1 of the guideline takes a couple of days to execute, and the analysis block selected by the guideline in §4.4.4 has an error of 2.76 €/MWh, 2.27 s of execution time, is one of the most detailed workflows (0.01) and has a score of 0.4 in open-sourceness.

Figure 5.14 shows the performance of one analysis block found with the guideline in chapter 4 (Method G) and the analysis blocks selected with seven other approaches that a workflow designer may use in the absence of a formal decision theory. These methods are described next.

Method A: first satisfactory solution This approach consists of selecting a solution that satisfies predefined constraints on all the criteria. To implement it, analysis blocks are randomly sampled and scored until a satisfactory solution is found. The constraints are $\hat{C}_{acc} < 0.5$, $\hat{C}_{time} < 0.25$, $C_{det} < 0.2$ and $C_{os} < 0.2$.

The workflow selected in an exemplary execution of this approach couples the Jensen wake model, summed deficits wake-merging model, Larsen wake-added turbulence model, Esau-Williams electrical collection design tool, dummy constant support structure cost model, FAST wind turbine model, and samples 3 wind speeds and 36 wind directions.

Method B: greedy for accuracy Analysis blocks can be built without knowing their performance a priori. In this method, design choices are made by the workflow designer with the aim of increasing for accuracy, while keeping an eye on execution time, open-sourceness and detail. This procedure would function as follows.

For accuracy, a seasoned workflow designer would keep the Ainslie 1D wake model together with a large number of sampling points for wind speed, avoid the inaccurate multiplied wake-merging model and couple the RSS model, and the Quarton wake-added turbulence model. For acceptable execution times, the number of wind directions would be capped at 12 wind directions. Accuracy is more sensitive to the number of wind speed sampling points than to the number of wind directions sampling points. Open-sourceness can be increased by coupling the POS electrical collection design heuristic.

Finally, the TeamPlay support structure sizing tool and the FAST wind turbine model would be used to increase the level of detail of the analysis block.

This analysis block has indeed a high accuracy, detail and open-sourceness, at the cost of the execution time.

Method C: greedy for execution time This method follows the procedure of method B, though prioritising execution time and being aware of the rest of the criteria. The implementation of this method would encourage the workflow designer to avoid the time-consuming Ainslie 1D wake model and use a low number of wind speed sampling points. Then accuracy would be better served by increasing the number of wind directions sampled to the maximum possible, coupling the Quarton, RSS, FAST and TeamPlay models, which do not significantly impact execution time. Open sourceness would require not to couple the QBlade wind turbine model, the Larsen wake model, the Frandsen and Larsen wake-added TI models and the Esau-Williams heuristic. The Jensen model and POS heuristic are chosen instead.

This analysis block has low execution time, detail and open-sourceness at the cost of accuracy.

Method D: minimum execution time This approach sees the workflow designer choose the analysis block that minimises execution time and disregards the rest of the criteria entirely. Evidently, without any optimisation algorithm, the global optimal analysis block cannot be known. In this case, the optimal is approximated by the best-performing analysis block of a set of 6000 randomly sampled alternatives.

The resulting analysis block reduces the sampling points to the minimum possible and couples the Jensen, RSS, Larsen wake-added TI, POS, TeamPlay and BEM models. This choice has naturally a positive effect on execution time at a huge cost on accuracy and detail. Open-sourceness is at its lowest value too, as closed-source models were chosen in between the most and least sophisticated models.

Method E: maximum accuracy This method complies with the objective of the workflow designer to maximise accuracy and disregard the rest of the criteria.

The solution is merely the analysis block closest to the alternative used as reference for the accuracy metric. This solution couples the Ainslie 1D, RSS, Quarton, Esau-Williams, TeamPlay, FAST models and samples 25 wind speeds and 90 wind directions. As expected, the wake analysis is very expensive with this analysis block. Detail is low due to the great sophistication of the models coupled and open-sourceness is 60%.

Method F: maximum detail Because detail is a criterion that takes only discrete values, there are several analysis blocks with the highest level of detail. These solutions are picked from a random sample of 6000 analysis blocks and their range of accuracies and execution times are shown in Fig. 5.14.

All these analysis block couple the Larsen, Quarton, Esau-Williams, TeamPlay and FAST models, and they all have a high open-sourceness score of 0.6. As shown also in Fig. 5.13, accuracy and time will typically not reach low or high values simultaneously.

Method H: maximum open-sourceness This method is analogous to method F, in that many solutions have an open-sourceness of 100%. Figure 5.14 shows the range of values of the rest of the criteria for these analysis blocks.

None of the solutions couple the closed-source models Larsen, Frandsen, Quarton, Esau-Williams and QBlade models. Detail, execution time and accuracy will typically not reach their lowest values simultaneously.

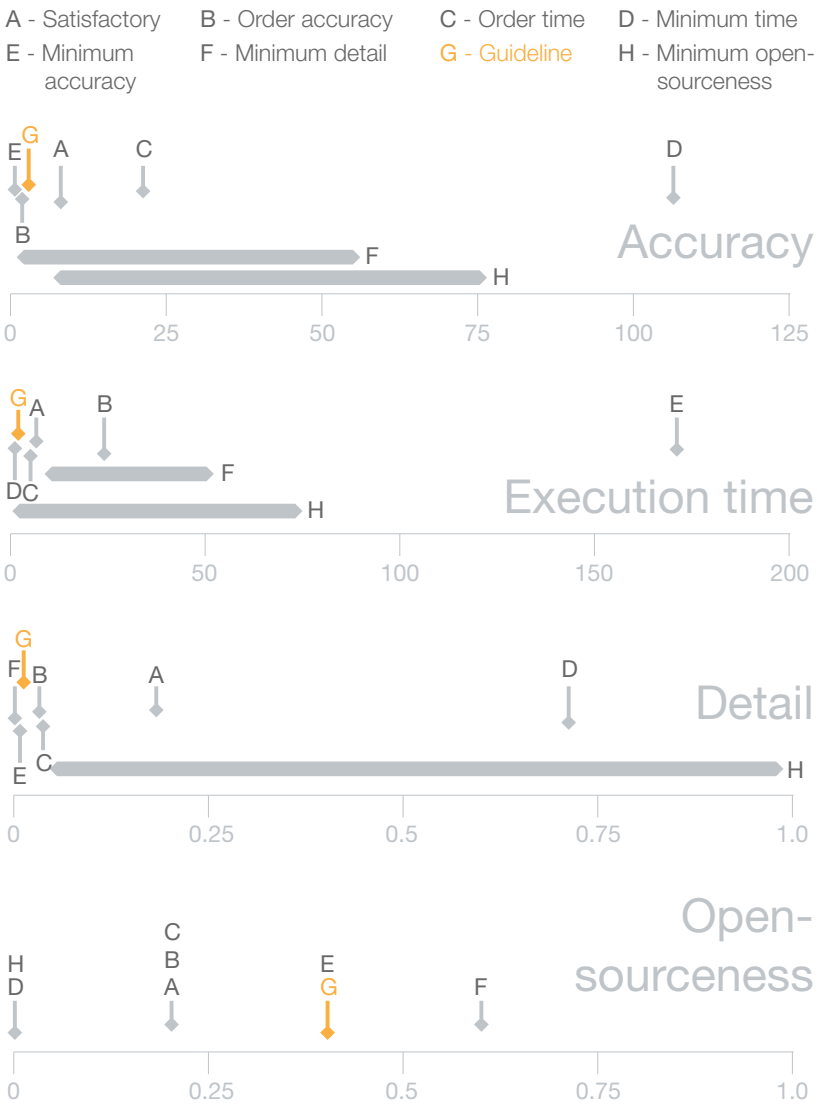


Figure 5.14: Performance of analysis blocks selected with the guideline in chapter 4 and using other methods.

The analysis block selected by the guideline (method G in Fig. 5.14) can be seen to be

the only solution that jointly minimises accuracy, time and detail, at the cost of open-sourceness. Other solutions, such as those found with methods A and B also tend to score low in all criteria. However, they do not consistently have better scores than the analysis block found with the guideline. Methods A, B and C try to cater for all the criteria, while methods D, E, F and H cater for a single criterion.

Therefore, methods B and C can more directly be compared with the guideline, as the solution marked with G is chosen under no preferences on the criteria. The Euclidean distances to the utopia point of the solutions found using methods A, B, C are all higher than that of the solution of the guideline. In ascending order, these distances are: G: 0.46, A: 0.66, B: 0.78, C: 0.97. Complementary, the average ranking of the analysis blocks along the four criteria are: G: 2.83, B: 3.41, C: 3.91, A: 4.41. In both analysis, the solution of the guideline comes out on top of the four methods. Additionally, the solution of the guideline is non-dominated in the sense of Pareto with respect to the solutions of other methods.

The analysis blocks found with methods D, E, F and H help to evaluate the trade-offs that the guideline solves. The birth of the guideline is rooted at the notion of avoiding the design or selection of MDAO workflows based on a single criterion, and instead providing a means to find a compromise between criteria. Methods D, E, F and H fail to acknowledge that the analysis blocks will be judged along several criteria.

Admittedly, this analysis uses the criteria and metrics defined in the guideline itself to compare workflows selected with other methods. However, this comparison is deemed inevitable due to a lack of standards and methodologies for MDAO workflow evaluation. In addition, the guideline outperforms the other methods using various ranking methods.

More interesting, however, is to discuss when the improved performance of solutions found with the guideline outweigh the extra efforts compared to other methods. It can be safely assumed that a minimum level of accuracy is needed in all use cases. For this reason, the guideline is only compared to methods A, B and E, and the solutions of the rest of the methods are deemed much more inaccurate than the solution obtained with the guideline and never worth implementing. Method A can be claimed to be a good enough analysis block worth implementing if the workflow designer is willing to use a less accurate alternative, though not by much, and also less detailed, in exchange for a gain in open-sourceness. Method B's greatest disadvantage with respect to the guideline is the slightly higher execution time of its solution. A use case with multiple iterations of the analysis block that can not afford 25 seconds per call, would benefit from the solution of the guideline. Conversely, if 25 s per iteration is deemed acceptable, then method B would provide a workflow with less effort than using the guideline. Finally, the solution of method E has the highest execution time of all the methods, and the highest accuracy. Certain use cases related to certification and sensitivity analysis for making investment decisions would benefit from using this workflow instead of the solution of the guideline, due to the higher accuracy needed and the low number of iterations of the analysis block required. For use cases that require many iterations, then the guideline should provide a much faster option with minimum losses in accuracy.

5.4. PHASE 2: DRIVER ALGORITHM SELECTION

The validation of phase 2 with respect to the set of alternatives, shown as **V2** in Fig. 5.1, is treated here. Of all the activities suggested in the second phase of the guideline (§3.5, only the alternative-sorting algorithm to find the Pareto front needs to be validated. The other activity for defining the list of criteria is identical in phase 1, and its validations is treated in §5.3.2.

5.4.1. INFLUENCE OF THE SET OF ALTERNATIVES

This section addresses whether phase 2 works for all kind of models and driver algorithms.

With respect to the selection of driver algorithms, the second phase of the guideline is an objective approach that reduces the number of alternatives to the size of the set of non-dominated analysis blocks times the number of driver alternatives.

The burden of scoring all the alternatives depends on the number of MDAO workflows considered by the workflow designer.

Furthermore, the activities of phase 2 provide a deterministic process. When the scores of all the alternatives across all the criteria have been measured, the resulting Pareto front is always the same. In other words, the guideline is blind to the intrinsic nature of the alternatives once the scores are known.

Naturally, as with any other MCDA method, the criteria tree should have discerning power over the alternatives.

5.4.2. SORTING THE MDAO WORKFLOWS

To demonstrate the capabilities of the ϵ-non-dominated sorting algorithm in the MDAO workflow selection process of phase 2, a set of 6000 randomly sampled analysis blocks is used with different criteria to find different Pareto fronts. Figure 5.15 plots the random sample of analysis blocks (orange dots) and their non-dominated set (purple dots). Three combinations of two criteria and one combination of three criteria are shown.

The sorting algorithm manages to find the exact subset of non-dominated analysis blocks. This exercise is evidence of the general validity of this algorithm.

5.5. DISCUSSION

Informed argumentation, controlled laboratory experiments and simulated field experiments have been used to increase the confidence of workflow designers on the guideline. These are sources of evidence that point towards the original claim of this validation study: that the guideline is useful for selecting MDAO workflows. The guideline relies on validated theories that have stood the test of time, and which have remained essential to the operational research community. These theories are core elements of the MCDA approach taken by the guideline, which favours the concept of Pareto dominance over utility functions. Furthermore, validated theories support the activities suggested to define the list of criteria, to rank the solutions in the Pareto front and to find the Pareto front of MDAO workflows in phase 2.

The MOPSOC algorithm is the novel element that can be most contested, and much of the focus has been put on delivering examples of its proper response to varying defi-

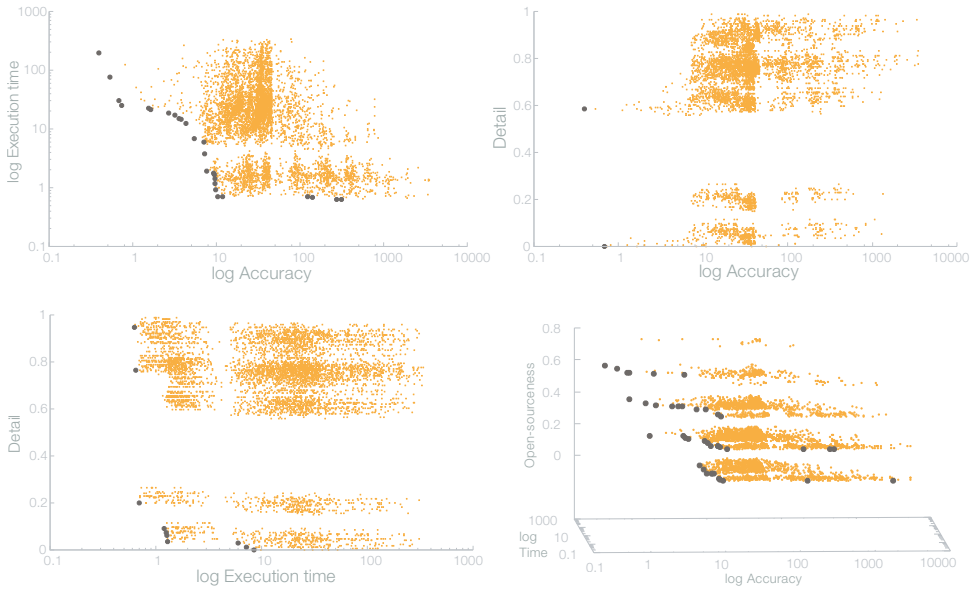


Figure 5.15: Scatter plots of a random sample of analysis blocks and their Pareto front found with the ϵ-non-dominated sorting algorithm. Four combinations of criteria are shown.

nitions of the criteria and optimisation formulation, and to underlying functions.

The guideline is an asset when there is a need to trade-off attributes of the MDAO workflow and when the choice of MDAO workflow can have serious repercussions on finance and/or safety. In this cases, the benefits of implementing a solution found with the guideline should outweigh the effort of implementing the guideline and the time devoted to it.

On the contrary, the guideline is less useful for selecting MDAO workflows when there are few alternatives, or when many alternatives couple very time-consuming models. It is also considered an overly complicated approach when a *good enough* MDAO workflow is needed for a simple job without much repercussions.

Key is to note that the criteria and their metrics used to judge MDAO workflows, as well as the non-deterministic nature of the MOPSOC algorithm will inevitable incur in assumptions and measurement errors. These errors translate into an underlying uncertainty in the main outcome of the guideline: the ranking of the alternatives. It is therefore suggested to experiment with the choices made throughout an implementation of the guideline to raise awareness of their implications on the output and to increase the confidence in the chosen solution.

6 Conclusions

'Tis better to have loved and lost
than never to have loved at all.

Alfred Tennyson

This chapter brings this book to an end by recapitulating the research done as well as the main contributions and findings.

Additionally, the compliance of the initial research objective is discussed.

Furthermore, assumptions and limitations of this research, recommendations for future work, the generalisation of the contributions and their implications for society are examined.

6.1. INTRODUCTION

Defined in §1.6, the objective of this research was to develop a systematic and objective methodology for selecting the best-performing model fidelities and driver algorithms of an MDAO workflow for the domain of offshore wind farms.

A methodology for selecting different MDAO workflows greatly benefits from the ability to automate the creation and evaluation of alternatives. The first part of this work has entailed, therefore, the design and development of a tool that instantiates MDAO workflows for offshore wind farm design optimisation. The instantiations differ from each other in the models that are coupled in the analysis block that evaluates the LCOE of an offshore wind plant, and in the algorithm used to drive the optimisation of its design. The tool enables workflow designers to choose the models and optimisation algorithm coupled, instantiate the MDAO workflow by making the connections between modules and between the analysis block and the driver, and execute the MDAO workflow.

In the second part of this work, a guideline for selecting the set of best-performing MDAO workflows has been developed, shown how to instantiate it, and validated. The guideline is based on the concept of multicriteria decision analysis, where several, conflicting criteria are used to evaluate, compare and rank multiple alternatives. In particular, the notion of non-dominance in the sense of Pareto is used to compare the alternatives. The guideline is divided into two phases. In the first phase a set of analysis blocks is selected, and in the second these are coupled to driver algorithms to select a set of MDAO workflows.

Both phases of the guideline start by suggesting a method for defining and assessing the quality of the list of criteria against which the alternatives will be judged.

Because an analysis block comprises multiple modules, each of which may be instantiated with multiple different models, the number of analysis blocks that can be built is too large to evaluate all of them. In light of that, the first phase of the guideline treats the problem of finding the set of non-dominated analysis blocks as a multiobjective optimisation problem. For that purpose, a novel algorithm called MOPSOC (Multiobjective Particle Swarm Optimisation for Categorical Variables) has been developed, which approximates the Pareto front of a function of categorical variables, such as the choice of models coupled in the analysis block. The guideline includes information about the consequences of using different formulations of the optimisation problem.

Depending on the number of criteria, the Pareto front approximated by the MOPSOC algorithm can include in the order of a hundred solutions. To further reduce this number to a more tractable decision problem, the guideline suggests a couple of techniques for ranking non-dominated solutions with or without preferences over the criteria.

The second phase of the guideline assumes that in the order of ten MDAO workflows can be built by coupling the analysis blocks selected in the first phase with a small number of driver algorithms. It is further assumed that it is reasonable to score all the alternatives against all the criteria. The ϵ-non-dominated sorting algorithm is then used to extract the true Pareto front of the set of alternatives. This new set can be further re-

duced using the techniques mentioned above.

The guideline is subsequently instantiated to select a set of MDAO workflows for the multidisciplinary optimisation of an offshore wind farm built in the North Sea. Of over 650,000 different analysis blocks, phase 1 of the guideline is able to recommend six alternatives, in a case where the workflow designer has no preferences over the criteria. The six analysis block are then coupled to five optimisation algorithms to build 30 MDAO workflows. The second phase of the guideline then reduces this list to 16 non-dominated MDAO workflows, and then suggests which one to choose based on the workflow designer's preferences.

To validate the guideline, informed argumentation, controlled laboratory experiments and simulated field experiments are drawn upon to produce evidence that supports its unbiased and reliable performance. In particular, the MOPSOC algorithm is validated with a set of multiobjective test functions, by comparing the distribution of the distances to the utopia point of the approximated Pareto front, with the Pareto front of a random sample of analysis blocks, and by analysing its performance under a diverse set of optimisation formulations. The results of phase 1 of the guideline are compared with the analysis blocks obtained with other selection methods, and the influence of the set of criteria on the results is also assessed. The validity of the ϵ-non-dominated sorting algorithm is proven by visualising the Pareto front of a random sample of analysis blocks yielded by the algorithm.

The following sections delve deeper into the conclusions drawn about the tool built for instantiating MDAO workflows, and the guideline for selecting MDAO workflows.

6.2. TOOL FOR MDAO WORKFLOW INSTANTIATION

There are several aspects in which this tool is unique. Regarding its ability to instantiate MDAO workflows, no other tool is known to provide a set of model fidelities to choose from and to programmatically build and execute different analysis blocks. Without this tool, it would have been impossible to optimise the choice of models and number of sampling points using the MOPSOC algorithm.

Regarding the analysis block, it is useful for the optimisation of offshore wind farms with respect to LCOE. Although the validation study presented in §2.10 was done with a single set of models and number of sampling points, it showed that the sensitivities of the approximated LCOE with respect to multiple design variables have the correct signs. This increases the wind farm designer's confidence in the tool for optimisation purposes. Optimisation algorithms can exploit these sensitivities to drive the design towards a minimum LCOE.

The XDSM of the analysis block is deemed correct and in line with similar existing tools. The correctness of the XDSM of the analysis block can be assessed by discussing the completeness of the system scope and the existing connections between modules. By definition, simulations are abstractions of reality and assumptions have to be made.

The extent and validity of these assumptions define how well reality is represented. The XDSM of this tool represents all phases of the development of an offshore wind farm: from installation to operation to decommissioning. However, for simplicity and lack of access to modelling data, not all aspects of these phases are treated in the tool. For example, the impact of the logistics of installation and decommissioning procedures, and manufacturing costs of diverse components on the LCOE is ignored. Concerning the connections between modules, hard-coding these limits the choices of design variables and the capability to host high fidelity models. For example, the wind turbine can not currently be jointly designed with the wind farm as the tool has no connections between the parameters of the turbine and the rest of the farm-level modules, nor between the cost of the wind turbine to the LCOE module. In addition, the impact of the placement of the wind turbines on the fatigue of the blades and support structures is also not considered. Physics-based models with a higher degree of sophistication lack support for connecting their inputs and outputs. However, the modularity of this tool ensures that with proper continued development, any model fidelity can eventually be added to the pool of models available. At its current stage, on average, this tool matches the fidelity of other wind farm design tools publicly available.

The development of this tool can continue in multiple directions. Work has already began on the expansion of the XDSM to support the joint optimisation of the wind turbines with the design of the wind farm. In addition to LCOE, the tool is able to support other system level performance indicators with slight modifications. A recommended step is the addition of analytical gradients to the models. NASA's OpenMDAO framework, upon which this tool is built, is meant to facilitate the propagation of model's derivatives throughout an analysis block. Taking advantage of this feature would allow the rapid calculation of the full gradient of the LCOE, and explore faster gradient-based optimisation methods with multiple starts. However, these methods need modification to handle discrete design variables.

The tool was validated for an optimisation use case. Nevertheless, the use of the analysis block with other drivers, such as uncertainty quantification and robust optimisation, is not only possible but encouraged.

Furthermore, this tool is envisioned to eventually be able to read files that describe use cases by means of an XDSM, and automatically implement the connections between the necessary modules to create analysis blocks, and make the entire MDAO workflow by coupling the desired driver algorithm.

Lastly, this tool can bring value to education and industry. Exploring the interactions between design choices of different sub-components of the wind farm leads to better understanding of the complex dynamics that take place in the design and operation of the system. After proper calibration of the cost models included in the tool using field-measured data, the tool has the potential to become the industry workhorse for early-stage design explorations. Its open-sourceness ensures collective development and wide availability, in addition to helping spread and promote the adoption of systems engi-

neering and MDAO design philosophies.

6.3. GUIDELINE FOR MDAO WORKFLOW SELECTION

The guideline proposed in this work is a novel mechanism to evaluate, compare, and rank different MDAO workflows. There are no known formal methodologies for this purpose in the published literature.

The validity of the guideline concluded in chapter 5 applies to two concepts: the proven unbiased set of resulting MDAO workflows, and the ability to provide the workflow designer with a more tractable problem that requires a small effort compared to choosing from the entire set of alternatives. The validation study of the guideline was blind to the use case, arguing the usefulness of the guideline for multiple use cases.

The greatest advantage of adhering to this guideline is that in a short time, workflow designers can largely reduce the amount of alternatives to choose from. Using one of the selected MDAO workflows for solving a use case can impact the total budget allocated to a design optimisation or a what-if analysis campaign. The benefits can also cover the development phase of an offshore wind plant, by providing a more accurate analysis block for making financial, logistical or manufacturing decisions. Other positive impacts relate to practical issues such as reducing the costs of computational hardware and software.

Conversely, the drawbacks of following this guideline are that the workflow designer has to commit a good amount of effort into defining the criteria that will govern the selection process and into enabling the automatic generation of analysis blocks and MDAO workflows. In addition, the MOPSOC algorithm is of a stochastic nature. The analysis blocks that comprise the Pareto front will differ at every equal execution of the optimisation. However, the overall conclusions about the nature of the models to be coupled should be more enlightening than the exact analysis block yielded by the algorithm.

Key is to be aware that the hardest limitation for implementing this guideline is its infeasibility of handling highly sophisticated models or XDSMs with multiple cyclic-connections to be solved iteratively, both of which may lead to long execution times. However, this limitation could be partly mitigated by using surrogate models.

Noteworthy is the fact that the guideline might be considered too complicated for simple use cases or when a *good enough* solving philosophy provides useful results to the workflow designer.

One further achievement of this work is the encapsulation of the description of the MDAO workflow selection problem, which has lead to the compartmentalisation of the research. In particular, suggestions for further research can be broken down into the following units.

- **Criteria**:
  - (Sub-)objectives: It is imperative that the MCDA problem is properly defined to find a meaningful solution. A good problem definition depends mostly

on using the right criteria. A more thorough treatment of the list of criteria that differentiate MDAO workflows is in order. However, the criteria are to an extent determined by the differences between the alternatives, and naturally, strongly dependent on the use case.

  - Metrics: Another focal point for future research is the principle behind the quantitative measurement of criteria. The elusiveness of the term *fidelity* extends to the quantification of individual attributes of a simulation, a prime example being accuracy. In engineering design, models are evaluated in a domain for which they have not been validated. Therefore, workflow designers can only gain more confidence on the accuracy of the analysis block by doing more evaluations, which is expensive.

- **Optimisation formulation**: Although less critical than the criteria, the analysis block optimisation formulation influences the usefulness of the solution. Further research can lead to suggestions on how to decide which criteria should remain objective functions and which should become constraints.

- **MOPSOC**: The MOPSOC algorithm has been tested and validated. Its performance in the context of the guideline is acceptable. However, more effort can be put to investigate the effect of the parameters of the algorithm on the optimal solution. Parameter meta-optimisation is a common follow-up study on most novel algorithms. Additionally, the treatment of the constraints can be improved from the penalty functions currently used. In general, this algorithm can be optimised for a more efficient performance. Any optimisation problem would greatly benefit from increasing the number of iterations and better sampling the design space, at little additional cost.

- **Pareto front reduction**: The techniques suggested for reducing the cardinality of the Pareto front work well. However, there are several other methods, especially for including the preferences of the workflow designer on the criteria. It is recommended to expand the evaluation of other techniques.

Although industry would greatly benefit from the inclusion of this guideline for MDAO workflow selection, expectations in the short term must be tempered. There is currently limited availability for a wide range of model fidelities that simulate multiple disciplines that can be programmatically coupled into a fixed XDSM.

Despite the guideline having been instantiated and validated in the context of offshore wind energy, its deployment can be generalised to any other knowledge domain or industry. The core concepts of the guideline belong to the field of operational research, and as such, may be drawn upon by any scientific and engineering field.

The MDAO community benefits from this research as the ever-present trade-off between the sophistication and cost of multidisciplinary analysis and optimisation workflows continues to be overlooked. This guideline is expected to increase awareness of the need to improve the design and selection of MDAO workflows. In essence, this guideline

enables more efficient, cheaper and optimal design processes. These gains should translate into a cost reduction in the development of future offshore wind farms by solving use cases more efficiently, and to the reduction of the CAPEX, OPEX and LCOE of wind farms by implementing more accurate models and more efficient optimisation algorithms.

To conclude, the tool for generating MDAO workflows for offshore wind energy and the guideline are the first of their kind. This work attempts to make MDAO more visible and accessible to the wind energy community, and is meant to encourage the wider deployment of systems engineering in the field of wind energy.

REFERENCES

[1] G. A. Schmidt and D. Arndt, "Annual global analysis for 2017," January 2018.

[2] J. Hansen, R. Ruedy, M. Sato, and K. Lo, "Global surface temperature change," *Reviews of Geophysics*, vol. 48, no. 4, 2010.

[3] M. RE, "Relevant weather-related loss events worldwide 1980-2016," tech. rep., Munich RE, 2018.

[4] G. McGranahan, D. Balk, and B. Anderson, "The rising tide: assessing the risks of climate change and human settlements in low elevation coastal zones," *Environment and urbanization*, vol. 19, no. 1, pp. 17–37, 2007.

[5] N. Watts, M. Amann, S. Ayeb-Karlsson, K. Belesova, T. Bouley, M. Boykoff, P. Byass, W. Cai, D. Campbell-Lendrum, J. Chambers, P. M. Cox, M. Daly, N. Dasandi, M. Davies, M. Depledge, A. Depoux, P. Dominguez-Salas, P. Drummond, P. Ekins, A. Flahault, H. Frumkin, L. Georgeson, M. Ghanei, D. Grace, H. Graham, R. Grojsman, A. Haines, I. Hamilton, S. Hartinger, A. Johnson, I. Kelman, G. Kiesewetter, D. Kniveton, L. Liang, M. Lott, R. Lowe, G. Mace, M. Odhiambo Sewe, M. Maslin, S. Mikhaylov, J. Milner, A. M. Latifi, M. Moradi-Lakeh, K. Morrissey, K. Murray, T. Neville, M. Nilsson, T. Oreszczyn, F. Owfi, D. Pencheon, S. Pye, M. Rabbaniha, E. Robinson, J. Rocklöv, S. Schütte, J. Shumake-Guillemot, R. Steinbach, M. Tabatabaei, N. Wheeler, P. Wilkinson, P. Gong, H. Montgomery, and A. Costello, "The lancet countdown on health and climate change: from 25 years of inaction to a global transformation for public health," *The Lancet*, vol. 391, no. 10120, pp. 581–630, 2018.

[6] L. Meyer, S. Brinkman, L. van Kesteren, N. Leprince-Ringuet, and F. van Boxmeer, "IPCC, 2014: Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change," tech. rep., Intergovernmental Panel on Climate Change, 2014.

[7] O. Edenhofer, R. Pichs-Madruga, Y. Sokona, K. Seyboth, S. Kadner, T. Zwickel, P. Eickemeier, G. Hansen, S. Schlömer, C. von Stechow, *et al.*, *Renewable energy sources and climate change mitigation: Special report of the intergovernmental panel on climate change*. Cambridge University Press, 2011.

[8] J. F. Herbert-Acero, O. Probst, P.-E. Réthoré, G. C. Larsen, and K. K. Castillo-Villar, "A review of methodological approaches for the design and optimization of wind farms," *Energies*, vol. 7, no. 11, pp. 6930–7016, 2014.

[9] T. Stehly, D. Heimiller, and G. Scott, "2016 cost of wind energy review," tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO., 2017.

[10] M. B. Zaaijer, *Great expectations for offshore wind turbines: Emulation of wind farm design to anticipate their value for customers*. PhD thesis, Delft University of Technology, 2013.

[11] INNWIND.EU, "Lcoe reduction for the next generation offshore wind turbines," tech. rep., INNWIND.EU, 2017.

[12] Ørsted, "Wind technology website." `https://orsted.com/en/Our-business/Wind-Power/Wind-technology`, Feb. 2017. Accessed: 11-02-2018.

[13] Leanwind, "Driving cost reductions in offshore wind," tech. rep., Leanwind, 2017.

[14] S. Tegen, M. Hand, B. Maples, E. Lantz, P. Schwabe, and A. Smith, "2010 cost of wind energy review," tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO., 2012.

[15] K. Dykes, R. Meadows, F. Felker, P. Graf, M. Hand, M. Lunacek, J. Michalakes, P. Moriarty, W. Musial, and P. Veers, *Applications of systems engineering to the research, design, and development of wind energy systems*. National Renewable Energy Laboratory, 2011.

[16] K. Dykes, P. Rethore, F. Zahle, and K. Merz, "IEA Wind Task 37 Final Proposal. Wind Energy Systems Engineering: Integrated RD&D.," tech. rep., International Energy Agency, 2015.

[17] R. Bos, *Extreme gusts and their role in wind turbine design*. PhD thesis, TU Delft, 2017.

[18] C. L. Bottasso, F. Campagnolo, and A. Croce, "Multi-disciplinary constrained optimization of wind turbines," *Multibody System Dynamics*, vol. 27, no. 1, pp. 21–53, 2012.

[19] G. A. M. van Kuik, J. Peinke, R. Nijssen, D. Lekou, J. Mann, J. N. Sørensen, C. Ferreira, J. W. van Wingerden, D. Schlipf, P. Gebraad, *et al.*, "Long-term research challenges in wind energy – a research agenda by the european academy of wind energy," *Wind Energy Science*, vol. 1, no. 1, pp. 1–39, 2016.

[20] G. Katsouris, "Infield cable topology optimization of offshore wind farms," Master's thesis, TU Delft University of Technology, 2015.

[21] NASA, "Systems engineering handbook," tech. rep., National Aeronautics and Space Administration, 2007.

[22] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[23] S. Sanchez Perez-Moreno and M. B. Zaaijer, "How to select MDAO workflows," in *2018 AIAA/ ASCE/ AHS/ ASC Structures, Structural Dynamics, and Materials Conference*, p. 0654, 2018.

[24] S. Sanchez Perez-Moreno, M. B. Zaaijer, C. L. Bottasso, K. Dykes, K. O. Merz, P. E. Réthoré, and F. Zahle, "Roadmap to the multidisciplinary design analysis and optimisation of wind energy systems," in *Journal of Physics: Conference Series*, vol. 753, p. 062011, IOP Publishing, 2016.

[25] M. Caboni, *Probabilistic design optimization of horizontal axis wind turbine rotors*. PhD thesis, University of Glasgow, 2016.

[26] M. Zaaijer, "Review of knowledge development for the design of offshore wind energy technology," *Wind Energy*, vol. 12, no. 5, pp. 411–430, 2009.

[27] P.-E. Réthoré, P. Fuglsang, G. C. Larsen, T. Buhl, T. J. Larsen, and H. A. Madsen, "Topfarm: Multi-fidelity optimization of wind farms," *Wind Energy*, vol. 17, no. 12, pp. 1797–1816, 2014.

[28] T. Ashuri, M. B. Zaaijer, J. R. Martins, G. J. van Bussel, and G. A. van Kuik, "Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy," *Renewable Energy*, vol. 68, pp. 893–905, 2014.

[29] K. Maki, R. Sbragio, and N. Vlahopoulos, "System design of a wind turbine using a multi-level optimization approach," *Renewable Energy*, vol. 43, pp. 101–110, 2012.

[30] P. A. Fleming, A. Ning, P. M. Gebraad, and K. Dykes, "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, 2015.

[31] P. Bortolotti, C. L. Bottasso, and A. Croce, "Combined preliminary–detailed design of wind turbines," *Wind Energy Science*, vol. 1, no. 1, pp. 71–88, 2016.

[32] E. Quaeghebeur, S. Sanchez Perez-Moreno, and M. B. Zaaijer, "OWFgraph: A graph database for the offshore wind farm domain." Manuscript in preparation, 2019.

[33] T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang, "Approximation methods in multidisciplinary analysis and optimization: a panel discussion," *Structural and multidisciplinary optimization*, vol. 27, no. 5, pp. 302–313, 2004.

[34] K. Dykes, A. Ning, R. King, P. Graf, G. Scott, and P. Veers, "Sensitivity analysis of wind plant performance to key turbine design parameters: A systems engineering approach," in *32nd ASME Wind Energy Symposium, National Harbor, Maryland*, 2014.

[35] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette, "A survey of multidisciplinary design optimization methods in launch vehicle design," *Structural and Multidisciplinary Optimization*, vol. 45, no. 5, pp. 619–642, 2012.

[36] R. J. Balling and J. Sobieszczanski-Sobieski, "Optimization of coupled systems-A critical overview of approaches," *AIAA journal*, vol. 34, no. 1, pp. 6–17, 1996.

[37] E. J. Cramer, J. Dennis, Jr, P. D. Frank, R. M. Lewis, and G. R. Shubin, "Problem formulation for multidisciplinary optimization," *SIAM Journal on Optimization*, vol. 4, no. 4, pp. 754–776, 1994.

[38] O. de Weck, J. Agte, J. Sobieski, P. Arendsen, A. Morris, and M. Spieck, "State-of-the-art and future trends in multidisciplinary design optimization," in *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA*, vol. 1905, 2007.

[39] N. M. Alexandrov and R. M. Lewis, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," *AIAA journal*, vol. 40, no. 2, pp. 301–309, 2002.

[40] N. M. Alexandrov and S. Kodiyalam, "Initial results of an mdo method evaluation study," *AIAA paper*, vol. 4884, p. 1998, 1998.

[41] F. Zahle, P. Réthoré, P. Graf, K. Dykes, and A. Ning, "Fused-wind v0. 1.0," *ZENODO. doi*, vol. 10, 2015.

[42] S. Chowdhury, J. Zhang, A. Messac, and L. Castillo, "Unrestricted wind farm layout optimization (uwflo): Investigating key factors influencing the maximum power generation," *Renewable Energy*, vol. 38, no. 1, pp. 16–30, 2012.

[43] C. J. Crabtree, D. Zappalá, and S. I. Hogg, "Wind energy: Uk experiences and offshore operational challenges," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 229, no. 7, pp. 727–746, 2015.

[44] International Renewable Energy Agency, *Renewable energy technologies: cost analysis series. Volume 1: Power Sector. Wind Energy*, 2012.

[45] S. Fragoso Rodrigues, *A Multi-Objective Optimization Framework for the Design of Offshore Wind farms.* PhD thesis, TU Delft, 2016.

[46] M. Zhao, Z. Chen, and F. Blaabjerg, "Optimisation of electrical system for offshore wind farms via genetic algorithm," *IET Renewable Power Generation*, vol. 3, no. 2, pp. 205–216, 2009.

[47] N. B. Negra, J. Todorovic, and T. Ackermann, "Loss evaluation of hvac and hvdc transmission solutions for large offshore wind farms," *Electric power systems research*, vol. 76, no. 11, pp. 916–927, 2006.

[48] R. Haghi, T. Ashuri, P. L. van der Valk, and D. P. Molenaar, "Integrated multidisciplinary constrained optimization of offshore support structures," in *Journal of Physics: Conference Series*, vol. 555, p. 012046, IOP Publishing, 2014.

[49] S. Frandsen and K. Thomsen, "Change in fatigue and extreme loading when moving wind farms offshore," *Wind Engineering*, pp. 197–214, 1997.

[50] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-mw reference wind turbine for offshore system development," tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO., 2009.

[51] J. Jonkman, "The new modularization framework for the fast wind turbine CAE tool," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 202, 2013.

[52] D. Marten, J. Wendler, G. Pechlivanoglou, C. Nayeri, and C. Paschereit, "QBLADE: an open source tool for design and simulation of horizontal and vertical axis wind turbines," *International Journal of Emerging Technologies and Advanced Engineering*, vol. 3, no. 3, pp. 264–269, 2013.

[53] D. van Dommelen, *Python's wind turbine design package manual.* Delft University of Technology, September 2013.

[54] T. Tanmay, "Multi-disciplinary optimization of rotor nacelle assemblies for offshore wind farms: An agile systems engineering approach," Master's thesis, 2018.

[55] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application.* John Wiley & Sons, 2010.

[56] N. O. Jensen, "A note on wind generator interaction," tech. rep., Riso National Laboratory, 1983.

[57] G. C. Larsen, "A simple wake calculation procedure," tech. rep., Riso National Laboratory, 1988.

[58] J. F. Ainslie, "Calculating the flowfield in the wake of wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, no. 1, pp. 213–224, 1988.

[59] M. Anderson, "Simplified Solution to the Eddy-Viscosity Wake Model," *Document prepared by Renewable Energy Systems Ltd ("RES")*, 2009.

[60] E. Machefaux, G. C. Larsen, and J. M. Leon, "Engineering models for merging wakes in wind farm optimization applications," in *Journal of Physics: Conference Series*, vol. 625, p. 012037, IOP Publishing, 2015.

[61] D. J. Renkema, "Validation of wind turbine wake models," Master's thesis, Delft University of Technology, 2007.

[62] K. Gunn, C. Stock-Williams, M. Burke, R. Willden, C. Vogel, W. Hunter, T. Stallard, N. Robinson, and S. Schmidt, "Limitations to the validity of single wake superposition in wind farm yield assessment," in *Journal of Physics: Conference Series*, vol. 749, p. 012003, IOP Publishing, 2016.

[63] G. C. Larsen, J. Højstrup, and H. A. Madsen, "Wind fields in wakes," 1996.

[64] Danish Energy Agency, *Recommendation for the fulfillment of the requirements found in the technical criteria*, 1992.

[65] D. Quarton and J. Ainslie, "Turbulence in wind turbine wakes," *Wind Engineering*, pp. 15–23, 1990.

[66] S. Frandsen and M. L. Thøgersen, "Integrated fatigue loading for wind turbines in wind farms by combining ambient turbulence and wakes," *Wind Engineering*, pp. 327–339, 1999.

[67] S. T. Frandsen, "Turbulence and turbulence-generated structural loading in wind turbine clusters," 2007.

[68] Duckworth and Barthelmie, "Investigation and validation of wind turbine wake models," *Wind Engineering*, vol. 32, no. 5, pp. 459–475, 2008.

[69] I. Carlén and G. Schepers, "European wind turbine standards 2. part 1 sub a: Wind farms-wind field and turbine loading," in *European Wind Turbine Standards 2. Draft*, EUREC-Agency, 1998.

[70] L. R. Esau and K. C. Williams, "On teleprocessing system design, part ii: A method for approximating the optimal network," *IBM Systems Journal*, vol. 5, no. 3, pp. 142–147, 1966.

[71] J. Bauer and J. Lysgaard, "The offshore wind farm array cable layout problem: a planar open vehicle routing problem," *Journal of the Operational Research Society*, vol. 66, no. 3, pp. 360–368, 2015.

[72] G. Van Bussel and M. Zaaijer, "Reliability, availability and maintenance aspects of large-scale offshore wind farms, a concepts study," in *Proceedings of MAREC*, vol. 2001, 2001.

[73] S. Sanchez Perez-Moreno, K. Dykes, K. O. Merz, and M. B. Zaaijer, "Multidisciplinary design analysis and optimisation of a reference offshore wind plant," in *Journal of Physics: Conference Series*, vol. 1037, p. 042004, IOP Publishing, 2018.

[74] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of Machine Learning* (C. Sammut and G. Webb, eds.), pp. 760–766, Springer US, 2010.

[75] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, vol. 1, pp. 39–43, New York, NY, 1995.

[76] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[77] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*. John Wiley & Sons, 2004.

[78] A. Chehouri, R. Younes, A. Ilinca, and J. Perron, "Review of performance optimization techniques applied to wind turbines," *Applied Energy*, vol. 142, pp. 361–388, 2015.

[79] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[80] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[81] G. Rossum, "Python reference manual," tech. rep., Amsterdam, The Netherlands, The Netherlands, 1995.

[82] J. S. Gray, K. T. Moore, and B. A. Naylor, "Openmdao: An open-source framework for multidisciplinary analysis and optimization," in *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, AIAA, AIAA-2010-9101*, (Fort Worth, Texas), AIAA, August 2010.

[83] J. Gray, T. A. Hearn, K. T. Moore, J. T. Hwang, J. R. Martins, and A. Ning, "Automatic evaluation of multidisciplinary derivatives using a graph-based problem formulation in openmdao," in *Proceedings of the 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2014.

[84] D. R. VanLuvanee, "Investigation of observed and modelled wake effects at Horns Rev using WindPRO," Master's thesis, Technical University of Denmark, 2006.

[85] Z. C. Roza, *Simulation fidelity theory and practice*. PhD thesis, Technische Universiteit Delft, 2005.

[86] S. Robinson, "Simulation model verification and validation: increasing the users' confidence," in *Proceedings of the 29th conference on Winter simulation*, pp. 53–59, IEEE Computer Society, 1997.

[87] V. Belton and T. Stewart, *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer US, 2012.

[88] H. Polatidis, D. A. Haralambopoulos, G. Munda, and R. Vreeker, "Selecting an appropriate multi-criteria decision analysis technique for renewable energy planning," *Energy Sources, Part B*, vol. 1, no. 2, pp. 181–193, 2006.

[89] D. Gross, "Report from the Fidelity Definition and Metrics Implementation Study Group (FDM-ISG). Orlando, Florida: Simulation Interoperability Standards Organization (SISO); 1999 March. Report No," tech. rep., 99S-SIW-167, 1999.

[90] S. Y. Harmon and S. Youngblood, "Leveraging fidelity to achieve substantive interoperability," in *Proceedings of the 2001 Spring Simulation Conference*, pp. 25–30, 2001.

[91] D. K. Pace, "Fidelity, Resolution, Accuracy, and Uncertainty," in *Modeling and Simulation in the Systems Engineering Life Cycle*, pp. 29–37, Springer, 2015.

[92] R. L. Keeney and H. Raiffa, *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.

[93] R. L. Keeney and H. Raiffa, "Decision with multiple objectives," 1976.

[94] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," in *International Conference on Parallel Problem Solving From Nature*, pp. 849–858, Springer, 2000.

[95] S. Sanchez Perez-Moreno and M. B. Zaaijer, "Multiobjective particle swarm optimisation for categorical variables." Manuscript submitted, 2018.

[96] R. Hassan, B. Cohanim, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2005.

[97] S. Helwig, *Particle Swarms for Constrained Optimization Partikelschw arme f ur Optimierungsprobleme mit Nebenbedingungen.* PhD thesis, University Erlangen-Nuernberg, 2010.

[98] S. Strasser, R. Goodman, J. Sheppard, and S. Butcher, "A new discrete particle swarm optimization algorithm," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pp. 53–60, ACM, 2016.

[99] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 1042–1049, Morgan Kaufmann Publishers Inc., 2001.

[100] W. Tong, S. Chowdhury, and A. Messac, "A multi-objective mixed-discrete particle swarm optimization with multi-domain diversity preservation," *Structural and Multidisciplinary Optimization*, vol. 53, no. 3, pp. 471–488, 2016.

[101] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[102] C. Kao, "Weight determination for consistently ranking alternatives in multiple criteria decision analysis," *Applied Mathematical Modelling*, vol. 34, no. 7, pp. 1779–1787, 2010.

[103] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *International conference on parallel problem solving from nature*, pp. 722–731, Springer, 2004.

[104] P. J. Bentley and J. P. Wakefield, "Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms," in *Soft computing in engineering design and manufacturing*, pp. 231–240, Springer, 1998.

[105] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[106] I. Cabezas and M. Trujillo, "A method for reducing the cardinality of the pareto front," in *Iberoamerican Congress on Pattern Recognition*, pp. 829–836, Springer, 2012.

[107] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 263–282, 2002.

[108] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*. John Wiley & Sons, 2006.

[109] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez, "A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908–3924, 2008.

[110] A. Garcia-Hernandez, "A note on the validity and reliability of multi-criteria decision analysis for the benefit–risk assessment of medicines," *Drug safety*, vol. 38, no. 11, pp. 1049–1057, 2015.

[111] S. E. Toulmin, *The uses of argument*. Cambridge university press, 2003.

[112] D. L. Thurston, J. V. Carnahan, and T. Liu, "Optimization of design utility," *Journal of Mechanical Design*, vol. 116, no. 3, pp. 801–808, 1994.

[113] B. Falkenhainer and K. D. Forbus, "Cmpositional modeling: finding the right model for the job," *Artificial intelligence*, vol. 51, no. 1, pp. 95–143, 1991.

[114] R. Radhakrishnan and D. A. McAdams, "A methodology for model selection in engineering design," *Journal of mechanical design*, vol. 127, no. 3, pp. 378–387, 2005.

[115] G. M. Provan and A. Feldman, "A framework for assessing diagnostics model fidelity.," in *DX@ Safeprocess*, pp. 127–134, 2015.

[116] T. T. Binh and U. Korn, "Mobes: A multiobjective evolution strategy for constrained optimization problems," in *The Third International Conference on Genetic Algorithms (Mendel 97)*, vol. 25, p. 27, 1997.

[117] V. Chankong and Y. Y. Haimes, "Multiobjective decision making: Theory and methodology," in *North-Holland Series in System Science and Engineering*, vol. 8, Elsevier Science Publishing Co New York NY, 1983.

[118] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[119] C. Poloni, G. Mosetti, and S. Contessi, "Multi objective optimization by gas: Application to system and component design," in *ECCOMAS'96: Computational Methods in Applied Sciences' 96*, pp. 1–7, John Wiley & Sons, Ltd, 1996.

[120] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary computation*, vol. 3, no. 1, pp. 1–16, 1995.

[121] R. Viennet, C. Fonteix, and I. Marc, "Multicriteria optimization using a genetic algorithm for determining a pareto set," *International Journal of Systems Science*, vol. 27, no. 2, pp. 255–260, 1996.

CURRICULUM VITÆ

Sebastian SANCHEZ PEREZ-MORENO

27-04-1988 Born in Mexico City, Mexico.

EDUCATION

2002–2005 High School
Instituto Carlos Gracida
Oaxaca, Mexico

2006–2013 Undergraduate in Physics
Universidad Nacional Autónoma de México
Mexico City, Mexico
Thesis: Numerical Solution of the Swift-Hohenberg Equation

2012–2013 Writer and developer of the Physics MOOC *Saber UNAM*
Universidad Nacional Autónoma de México
Mexico City, Mexico

2013–2014 MSc. New and Renewable Energy
Durham University
Durham, United Kingdom
Thesis: The Aerodynamic Damping of Large Wind Turbine Blades during Fatigue Tests

2015–2019 PhD Researcher
Delft University of Technology
Delft, the Netherlands
Thesis: A guideline for selecting MDAO workflows with an application in offshore wind energy
Promotors: Prof. dr. G. van Bussel and Dr. ir. M. Zaaijer

2019 Developer of the MOOC *The future of wind energy*
Delft University of Technology
Delft, the Netherlands

List of Publications

1. E. Quaeghebeur, S. Sanchez Perez-Moreno, and M. B. Zaaijer, "OWFgraph: A graph database for the offshore wind farm domain." Manuscript in preparation, 2019

2. S. Sanchez Perez-Moreno and M. B. Zaaijer, "Multiobjective particle swarm optimisation for categorical variables." Manuscript submitted, 2018

3. S. Sanchez Perez-Moreno, K. Dykes, K. O. Merz, and M. B. Zaaijer, "Multidisciplinary design analysis and optimisation of a reference offshore wind plant," in *Journal of Physics: Conference Series*, vol. 1037, p. 042004, IOP Publishing, 2018

4. S. Sanchez Perez-Moreno and M. B. Zaaijer, "How to select MDAO workflows," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 0654, 2018

5. K. Dykes, S. Sanchez Perez-Moreno, F. Zahle, A. Ning, K. Merz, M. McWilliam, P. Bortolotti, and M. B. Zaaijer, "IEA Wind Task 37: systems modeling framework and ontology for wind turbines and plants," in *EAWE Wind energy science conference Copenhagen*, 2017

6. E. R. G. Quaeghebeur, S. Sanchez Perez Moreno, and M. B. Zaaijer, "OWFgraph: A graph database for the offshore wind farm domain," in *EAWE Wind Energy Science Conference Copenhagen*, 2017

7. M. P. C. Bontekoning, S. Sanchez Perez-Moreno, B. C.Ummels, and M. B. Zaaijer, "Analysis of the reduced wake effect for available wind power calculation during curtailment," in *Journal of Physics: Conference Series*, vol. 854, p. 012004, IOP Publishing, 2017

8. S. Sanchez Perez-Moreno, M. B. Zaaijer, C. L. Bottasso, K. Dykes, K. O. Merz, P. E. Réthoré, and F. Zahle, "Roadmap to the multidisciplinary design analysis and optimisation of wind energy systems," in *Journal of Physics: Conference Series*, vol. 753, p. 062011, IOP Publishing, 2016

9. S. Sanchez Perez Moreno and M. Zaaijer, "Model fidelity selection in the multidisciplinary optimisation of offshore wind farms," in *EAWE PhD seminar Stuttgart*, 2015

10. S. Sanchez Perez-Moreno, S. R. Chavarría, and G. R. Chavarría, "Numerical solution of the Swift–Hohenberg equation," in *Experimental and Computational Fluid Mechanics*, pp. 409–416, Springer, Cham, 2014