Offshore Wind Farm Installation Planning.

Decision-support tool for the analysis of new installation concepts.

Maksym Semenyuk

July 1, 2019

TU Delft Delft University of Technology

SIEMENS Gamesa RENEWABLE ENERGY

Offshore Wind Farm Installation Planning.

Decision-support tool for the analysis of new installation concepts.

Master of Science Thesis

For the degree of Master of Science in Sustainable Energy Technology

at Delft University of Technology
in collaboration with
Siemens Gamesa Renewable Energy
by

Maksym Semenyuk

Committee:

Prof. dr. S. Watson	TU Delft	
Dr. ir. M. Zaayer	TU Delft,	University Supervisor
Ir. M. Reinders	SGRE,	Company Supervisor
Dr. E. M. Lourens	TU Delft	

July 1, 2019

The greatest challenge in life is to make a choice.
The greatest freedom in life is the right to choose.

Make a choice. Follow it.

With great responsibility comes great power.

Acknowledgements

Writing a thesis, as well as completing an entire master programme, takes a little bit more than just knowledge and diligence. It is not a simple mix of planning, learning and working hard. It takes a lot of energy, external assistance and motivation. For me people are the main source of these.

The purpose of this section is to express my appreciation to all people who directly or indirectly facilitated successful completion of this project and my master's as a whole.

I would like to start with my parents. They not only made it possible for me to study at TU Delft but continuously supported me with their advice, belief and love. My parents are probably the biggest motivators I have.

Secondly, I want to acknowledge my supervisors.
Michiel, I admire your supervision skills and intelligence. Your steering during the project was essential to me, and your inputs helped me to take a lot of decisions throughout the research. It was a pleasure for me to work with you seeing that you were always reachable and provided me with a very detailed feedback.
Mark, a big thank you for dedicating time to explain me all the ins and outs of the offshore installation procedure. Your guidance was indispensable when formulating objectives and defining the scope of the research. One phrase that I will remember from you – *when working in innovation department, you have to challenge everything*.

Furthermore, thank you Pim for making this thesis possible. It took a lot of time to arrange it and I appreciate your help in finding for me this topic within SGRE. Thanks to the entire Hague office, it is an incredibly pleasant and friendly environment to work in.

Looking at a larger picture, I have to acknowledge all professors from my master;s programme, as well as two of my internship supervisors. You all contributed to a successful completion of this master. Thanks to my thesis committee for spending time on evaluating my graduate project.

After all, there are two groups of people who are my biggest sources of energy.
My warmest feelings to my friends in Ukraine with whom I have maintained close relationships despite being so far from each other.
And last but not least, my dear Delft friends. Each and every of you brought something special to my life. Some of you inspired me, some of you supported me with advice, some of you helped me, some of you talked to me when I needed it, and all of you had fun together with me. It is a miracle that we all met here and managed to build such a strong friendship among completely different individuals. Like a secure energy system requires different energy sources to operate, and so does our diverse group. I am massively happy to meet you all.

Maksym Semenyuk
Delft, 10.06.2019

Executive Summary

Background and Objectives

According to Global Wind Energy Council, the wind industry has to reach an installation rate of 10 GW per year globally by 2030 in order to comply with the environmental goals (GWEC, 2018). Analysing the cost of offshore wind farm (OWF) development in figure 1, it can be seen that ***a significant part of the total OWF cost is encountered during construction period.***

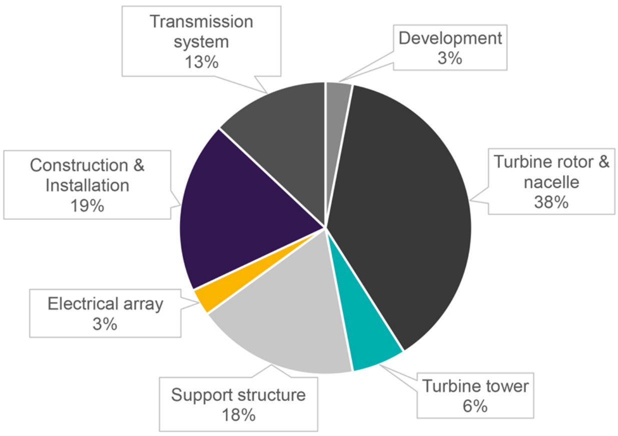


Figure 1: Cost breakdown for a representative OWF in Europe, (IRENA, 2016b)

Furthermore, wind turbine manufacturers, wind farm developers and foundation manufacturers continue introducing new technologies to the market. As an example of recent developments, a new generation of wind turbines with the capacity of 10 MW and more is expected to be installed already in several years. New wind farms are being planned further offshore, which requires using other types of support structures from traditional monopiles. In particular, floaters and large jackets are believed to become an essential feature of new windfarms planned in the emerging offshore wind markets (Equinor, 2018). One of the main issues which designers face when developing new concepts is to assess how they will fit into existing installation practices.

With the above concerns being raised, the motivation behind this thesis project is in the logistical challenges which will inevitably arise in the offshore wind industry in the coming years. ***The main objective is to develop a tool which would allow to assess new offshore wind farm installation concepts and strategies***, in particular their impact on the construction process, as one of the most capital intensive, but at the same time promising large cost reductions in the future (LEANWIND, 2018). Once developed, ***the tool has to facilitate dealing with two main aspects: efficient installation planning and comparison of alternative installation concepts with more conventional ones.*** The tool is intended to support the Department of Innovation at Siemens Gamesa Renewable Energy during the conceptual development of novel support structures, as well as to help project planners in obtaining realistic project timelines accounting for underlying logistics.

Requirements and Conceptual Design

The following functionality requirements to the tool have been identified:

- The tool needs to allow comparing different installation strategies.
- The tool needs to allow comparing different types of turbine substructures.
- The tool needs to be able to deliver a timetable of wind turbine and substructure installation for the wind farms of any size.
- A functionality of advising the user on the better decisions to improve current installation practices needs to be included.
- An ability to analyze the installation at different weather conditions needs to be included.
- The tool needs to allow to investigate different options for the vessel fleet. (Including analysis of different number and combination of vessels, vessels with different capabilities and limitations.)
- The tool needs to produce a detailed representation of the on-site activities performed by the vessels.
- The tool needs to be able to incorporate all relevant precedence requirements between the activities and other OWF installation logic.
- An ability to imitate different assembly strategies needs to be included.
- An onshore port consideration will have an added value.
- A computational time of the tool has to be reasonable, i.e. so as to allow utilization of the tool on a daily basis.
- The outputs of the tool must be targeted at users of various technical background.

After conducting analysis of the industry trends and relevant research, it was decided that the tool has to comprise several blocks which in combination would allow to address the above requirements. A substantial level of functionality has to be accomplished.

For this purpose, ***the tool consists of two major blocks: Simulator and Optimizer***. The former is intended to be able to imitate the process of OWF installation as realistically as possible, allowing a user to specify all the necessary information related to a certain installation approach, wind farm and involved equipment. The simulator was developed based on the existing Python library SimPy that provides a framework for the simulation. The simulator was interfaced via a common office software, Excel. In this way the user is not required to have a good knowledge of programming in order to change inputs and read outputs.

The next major block is an optimizer, which main purpose is to advice its user on the most optimal installation set up, i.e. identifying optimal values of certain parameters that lead to a lower total installation expenditure.

Finally, an ancillary block needed to generate synthetic weather data was added. Seeing that wind speed and wave height are two main weather factors affecting the installation, it is important to account for their variation from year to year. If a decision maker wants to obtain an accurate estimate of the project timeline for a specific location, multiple simulations are needed to achieve statistical accuracy and account for the stochastic weather nature.

The following figure 2 schematically represents the conceptual design of the tool.

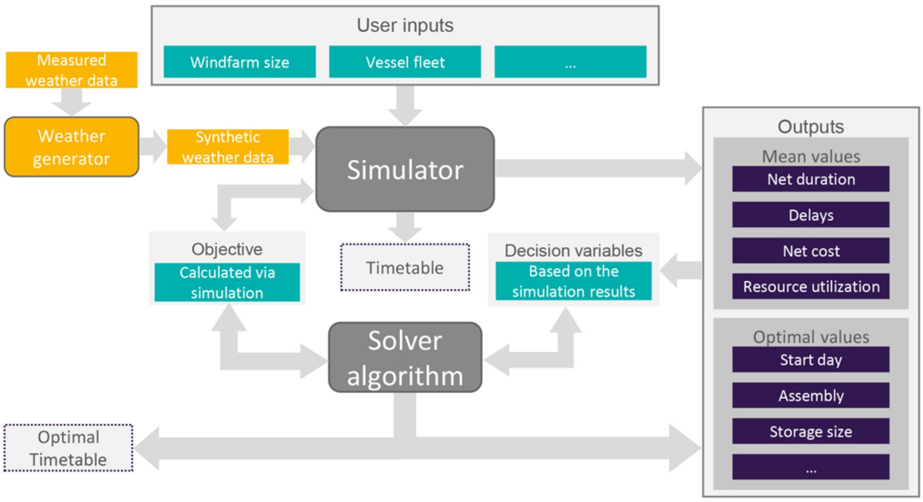


Figure 2: Conceptual design of the developed tool

Weather modelling findings

In the process of development of the tool it was discovered that weather modelling may significantly affect the outcomes of simulation or optimization. Several statistical indicators can be analyzed in order to evaluate the quality of synthetically generated weather data. However, the nature of offshore operations imposes specific requirements. ***The most important parameter for installation planning is a persistence of certain weather state***, where weather state simply means wind and wave conditions below or above certain threshold. A concept of weather window is introduced which describes a period of time with a certain duration, when wind speed and waves are remaining below or above specified limits.

Based on the conducted literature review, Markov Chains were selected as the most suitable approach for weather modelling, which along with average values, distribution probabilities, etc. preserves seasonality of the data and ensures correlation between wind and waves.

The major deficiency of Markov Chain is the fact that the generated weather series were characterized by a high intermittency compared to a measured data. In other words, too many oscillations occurred in synthetic series compared to a real ones. This fact is crucial for the offshore installation. Prior to performing any activity offshore, it has to be assured that the weather conditions will stay belong a certain limit for a certain time.

With the generated series, the outcomes of the simulation would not be realistic as the distribution of weather windows was far from the measured one.

Two steps were proposed that successfully compensated for the volatility of a weather signal generated via Markov Chain approach:

- The first step is applying a moving average to a generated signal, thus essentially making it more smooth and reducing the number of oscillations at high frequency (corresponding to the period of up to three hours). As a result, power spectral density of the generated signal became much more similar to the one of an original measured data.

- The second step was to keep the wind speed and wave height values constant in case a generated state signal contains a sequence of consecutive several hours in the same state (a state is defined by its lower and upper wind or wave limits).

The adopted approach proved to produce the results that are much better than that of the original Markov Chain method. In fact, the probability distribution of different weather windows was imitated with an error below 10%, whilst before it was over 30%.

Simulation findings and Validation Results

The paradigm of Discrete Event Simulation (DES) was selected as the main framework for developing a simulator in this thesis. The main advantage of DES is the fact that between consecutive events no change in the system can occur, and thus system time can be increased in a step-wise manner till the instant of the next event occurrence. This facilitates shorter computational time compared to a continuous simulation. A single simulation of an OWF installation takes approximately 50 seconds.

As a result of such simulation, ***one can obtain such KPIs as total duration, workability, daily progress, durations of loading and installation, percentage of delays per causing factor, total cost and cost per each contributing segment*** (see figure 3). These are conveniently saved in the standard office spreadsheet format so that a deep level of insight is combined with a simplicity of representation.

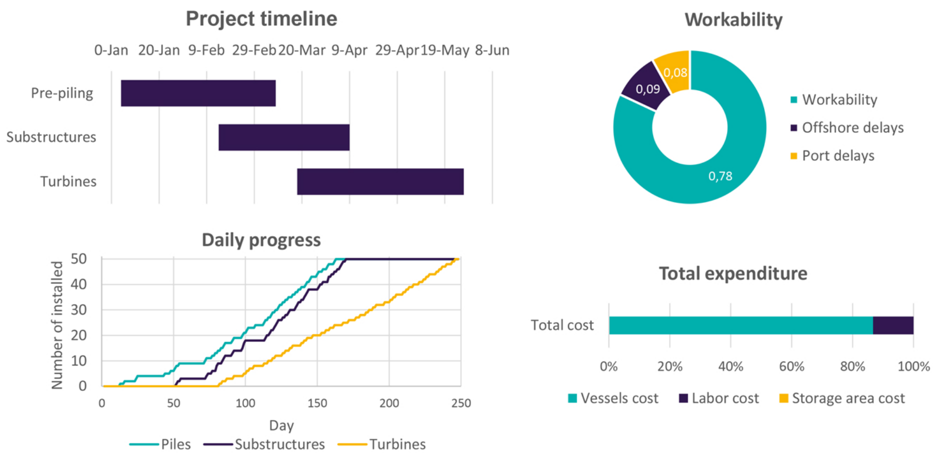


Figure 3: Example of tool outputs

A validation has been performed in order to prove that the results produced by the developed simulator are accurate and realistically reflect installation campaigns. For the purpose of validation an existing installation project was selected for which detailed project records were available. It turned out that the ***simulator successfully imitates installation process with minor deviations occurring due to machinery- and human caused errors*** which are not accounted for. Further, a more accurate results eliminating effect of weather stochasticity were obtained by running multiple simulations with different weather realizations for the same geographic location. This allows to exclude the impact of weather uncertainty and can be used to get a good estimate of how a certain installation campaign would progress in a given geographic location.

Finally, a sensitivity study results were presented which analysed how various installation parameters affect vessel workability, total project duration and duration of different sub-activitites.

Optimization Results

Seeing that a decision maker is often interested in the values of installation parameters that yield the most cost efficient campaign, an optimizer was developed with the total installation cost as an objective function. ***Five optimization variables were selected based on their impact on the total cost and inability to find their optimal values manually. These variables are:***

- start day of pre-piling campaign
- start day of substructure campaign
- start day of turbine campaign
- number of jackets pre-assembly lines onshore
- size of jackets storage area

The type of the problem studied in this thesis is a so-called *black box* optimization. This means, that no direct relation is known between the decision variables and a value of the objective function. Moreover, each evaluation of the objective function takes significant amount of computational time as it requires running around 20 simulations of an OWF installation with given input parameters. Hence, it is not possible to talk about the derivatives as such, which imposes strong limitations on which optimization algorithms can be applied. Consequently, ***the main goal of the optimization study was to compare two types of optimization methods based on how many function evaluations they need in order to find a good solution***. A method which requires a smaller number of function evaluations (NFE) is deemed to be superior since it can save significant amount of time.

Two approaches were compared: direct stochastic metaheuristics Particle Swarm Optimization against model-based deterministic solver based on the Radial Basis Function method. It turned out that ***for a relatively small optimization problem studied in this thesis, model-based approach with a deterministic solver worked better, i.e. required lower NFE in order to find an optimum.***

Contents

Acknowledgements **i**

Executive Summary **iii**

List of Figures **xii**

List of Tables **xv**

Nomenclature **xvii**

1 Introduction **1**
1.1 Industry background 2
1.2 Problem statement & Objectives 3
1.3 Research methodology 4
1.4 Structure of the report 5

2 Conceptual design of the decision support tool for Offshore Wind Farm Installation Planning **7**
2.1 Requirements for envisaged decision support tool 8
2.2 Industry state-of-the-art tools 8
2.3 Research with the focus on Simulation tools 10
2.4 Research combining Discrete Event Simulation and Linear Programming Optimization 12
2.5 Research in Optimization tools with analytic evaluation 14
2.6 Decision support tool for this project 17
2.7 Summary 18

3 Simulation of OWF Installation **21**
3.1 Main elements of DES 22
3.2 Detailed architecture of the developed simulation tool 23
3.3 Fulfillment of the design requirements 26
3.4 Flow-diagram of a simulation run 27
3.5 Summary 30

4 Synthetic Weather Generation **33**
4.1 Need for synthetic weather. Overview of weather modelling approaches. . . . 34
4.2 Markov Chains 35
4.3 Results of the original Markov Chain model 37
4.4 Adjusted Markov chains approach. Comparison of produced weather series. 42

4.5 Summary 45

5 Case Study and Analysis of Simulator Outputs **47**
5.1 Case study 48
5.2 Simulation parameters 48
5.3 Comparison of the simulation results and real OWF installation 49
5.4 Sensitivity analysis and Extreme case 56
5.5 Summary 69

6 Optimization of OWF Installation **71**
6.1 Optimization objectives and prerequisites 72
6.2 Optimization problem 75
6.3 Algorithm. Stochastic metaheuristics and Model-based approach 79
6.4 Results comparison 87
6.5 Summary 95

7 Conclusions **97**
7.1 Fulfillment of the design requirements 98
7.2 Research findings 99
7.3 Recommendations 101

A Case Study **103**

B Optimization details **105**
B.1 Full formulation of the objective function 105
B.2 Optimization case study 105

C Jackets assembly **107**

D Simulation sequence **111**

E Optimization results. Search space exploration. **113**
E.1 PSO 114
E.2 RBF 117

Bibliography **121**

List of Figures

1 Cost breakdown for a representative OWF in Europe, (IRENA, 2016b) . . . v
2 Conceptual design of the developed tool . . . vii
3 Example of tool outputs . . . viii

1.1 Total installed cost breakdown for a representative OWF in Europe (IRENA, 2016b) . . . 2

2.1 ECN Install (Kumar, 2017) . . . 9
2.2 LEANWIND LIVO (LEANWIND, 2018) . . . 10
2.3 Generic structure of simulation-based OWF planner . . . 12
2.4 Upgraded ECN Install architecture (Kumar, 2017) . . . 14
2.5 Concept of planning system (Scholz-Reiter et al., 2010) . . . 15
2.6 Generic structure of OWF installation optimization software . . . 16
2.7 Conceptual design diagram of the tool in this project . . . 18

3.1 Simulator diagram . . . 23
3.2 Example of Simulator outputs . . . 26
3.3 Block-diagram of a single simulation run . . . 28

4.1 Example of wind series generated by Markov Chain approach . . . 38
4.2 Example of wave series generated by Markov Chain approach . . . 38
4.3 CDF curves for original Markov Chain approach . . . 39
4.4 Zoomed weather series for original Markov Chain approach . . . 39
4.5 FFT of synthetic series for original Markov Chain approach . . . 40
4.6 *Good* and *Bad* weather window CDF for original Markov Chain approach . . . 41
4.7 FFT of synthetic series for Markov Chain approach with moving average . . . 42
4.8 Zoomed weather series for adjusted Markov Chain approach . . . 43
4.9 Scatter diagram for adjusted Markov Chain approach . . . 44
4.10 *Good* and *Bad* weather window CDF for the adjusted Markov Chain approach 44

5.1 Project timeline for the simulation with real weather records . . . 50
5.2 Loading duration for the simulation with real weather records . . . 51
5.3 Installation duration for the simulation with real weather records . . . 51
5.4 Convergence of normalized averages against simulations number . . . 52
5.5 Project timeline for simulations with synthetic weather . . . 53
5.6 Average loading duration for simulations with synthetic weather . . . 54
5.7 Average installation duration for simulations with synthetic weather . . . 55
5.8 Start day sensitivity. Durations. . . . 57
5.9 Start day sensitivity. Workability and delays. . . . 57

5.10 Vessel capacity sensitivity. Durations. 58
5.11 Vessel capacity sensitivity. Workability and delays. 59
5.12 Loading sensitivity. 59
5.13 Loading sensitivity. Activity duration sensitivity. Workability and delays. . . . 60
5.14 Loading sensitivity. Operational limits sensitivity. Workability and delays. . . 60
5.15 Travel sensitivity. 61
5.16 Travel sensitivity. Activity duration sensitivity. Workability and delays. . . . 62
5.17 Travel sensitivity. Operational limits sensitivity. Workability and delays. . . . 62
5.18 Installation sensitivity. 63
5.19 Installation sensitivity. Activity duration sensitivity. Workability and delays. . 63
5.20 Installation sensitivity. Operational limits sensitivity. Workability and delays. 64
5.21 Extreme case project timeline 66
5.22 Project timeline for a ”Single parameter change” study 68

6.1 Optimizer diagram 72
6.2 Installation in one and two years 79
6.3 Adopted values for PSO parameters 84
6.4 Gutmann algorithm’s main principle (Gutmann, 2001) 86
6.5 Gantt chart of the optimal solution found by PSO 89
6.6 Gantt chart of the optimal solution found by RBF 89
6.7 Cost spread of the optimal solution found by PSO and RBF 90
6.8 PSO: Swarm evolution 91
6.9 RBF: Search progress 92
6.10 PSO and RBF objective values against NFE 92
6.11 PSO objective function vs starting days d_c, d_s, d_t 94
6.12 RBF objective function vs starting days d_c, d_s, d_t 94

C.1 Jackets assembly line 107
C.2 Jackets assembly in port 108

E.1 PSO objective function vs decision variables 114
E.2 PSO pairwise graphs for campaign start day d_c 115
E.3 PSO pairwise graph for assembly lines number r 115
E.4 PSO pairwise graphs for substructure installation start day d_s 116
E.5 PSO pairwise graphs for turbine installation start day d_t 116
E.6 RBF objective function vs decision variables 117
E.7 RBF pairwise graphs for campaign start day d_c 118
E.8 RBF pairwise graph for assembly lines number r 118
E.9 RBF pairwise graphs for substructure installation start day d_s 119
E.10 RBF pairwise graphs for turbine installation start day d_t 119

List of Tables

3.1 Example of an output timetable 25
3.2 Design Requirements review for Simulator block 26

4.1 State limits for adopted Markov Chain model 36
4.2 Example of Markov Chain transition probability matrix for wave heights corresponding to wind state #8 37
4.3 Summary of weather window counting for original Markov Chain approach 41
4.4 Summary of weather window counting for Markov Chain approach with moving average 42
4.5 Summary of weather window counting for adjusted Markov Chain approach 43

5.1 General information about case study OWF 48
5.2 Aggregated case study input data for a full cycle 48
5.3 Summary of the simulation with real weather records 49
5.4 Extreme case study OWF 65
5.5 Durations of operations for the extreme case 65
5.6 ”Single parameter change” case study OWF 67
5.7 Offshore sequence for ”Single parameter change” case study OWF 67
5.8 Results of a single parameter change 69

6.1 Optimization variables selection 75
6.2 Common radial basis functions 85
6.3 Optimization case study OWF 88
6.4 Optimization results 88

B.1 Optimization case study OWF. Full definition 106

D.1 Case study input data 111

Nomenclature

Latin symbols

a	Harbor size	*number of components*
C	Cost	$Euro$
c_g	Social coefficient	
c_p	Cognitive coefficient	
d	Number of dimensions; Starting day	
$f, g, h(x)$	Functions of variable x	
h	Wave height; step index in Global search	m; -
I	Integer	
L	Lower boundary	
N	Number	
n	Number	
$\boldsymbol{P}_g$	Best position of swarm vector	
$\boldsymbol{P}_i$	Best position of particle i vector	
P	Transition probability matrix	
$p(x)$	Polynomial of x	
p_{ijk}	Element of a transition probability matrix located at i, j, k	
r	Pre-assembly rate; Euclidian distance	*number per day*; -
r_g	Random number for social component	
r_p	Random number for cognitive component	
S	State (wind or wave)	
$s(x)$	Interpolant of x	
t	Time	$hour$ or hr
U	Upper boundary	
$\boldsymbol{V}$	Velocity of particle vector	
$\boldsymbol{X}$	Coordinate of particle vector	
$\mathbb{R}$	Set of Real numbers	
V	Wind speed	m/s
x	Variable	
$\mathbb{Z}$	Set of Non-negative Integer numbers	

Greek symbols

Δ	Difference	

γ	Shape coefficient
κ	Number of steps per optimization cycle
λ	Natural number
Ω	Search space
ω	Inertia coefficient
ϕ	Radial basis function

Abbreviations

AEP	Annual Energy Production
CAPEX	Capital Expenditure
CDF	Cumulative Distribution Function
DES	Discrete Event Simulation
EPC	Engineering Procurement Construction
FFT	Fast Fourier Transform
GA	Genetic Algorithm
ILP	Integer Linear Programming
KPI	Key Performance Indicator
LCoE	Levelized Cost of Electricity
LHS	Latin Hypercube Sampling
LP	Linear Programming
MILP	Multi Integer Linear Programming
MV	Medium Voltage
NFE	Number of Function Evaluations
OPEX	Operational Expenditure
OWF	Offshore Wind Farm
PSO	Particle Swarm Optimization
RBF	Radial Basis Functions
RNA	Rotor-Nacelle Assembly
SA	Simulated Annealing
SGRE	Siemens Gamesa Renewable Energy
SPMT	Self-Propelling Modular Transporter
TP	Transition Piece

Chapter 1

Introduction

The current chapter marks the beginning of the research by formulating the motivation for conducting a study, its objectives and selected approach. Section 1.1 contains an overview of the wind energy industry as of today, with a focus on offshore wind farms installation. Next, section 1.2 gives a complete formulation of the research problem alongside with the objectives which are to be achieved during this thesis project. The selected approach is then described in section 1.3. Finally, the structure of the report is outlined in 1.4.

1.1 Industry background

The worldwide cumulative capacity of installed offshore wind reached 20.8 GW by the end of 2018. Over 18.5 GW (nearly 90% out of it) is grid-connected in Europe as of the the end of 2018 (WindEurope, 2019). Currently, electricity generated from wind accounts for 12% of the net production in Europe and it is expected that by 2050 wind energy could generate equivalent of 36% of total European power generation (in, so-called, Paris-compatible scenario which reflects the targets of 2015 Paris Agreement) (WindEurope, 2018). In order to achieve these ambitious goals, the projection is that the industry will have to reach an installation rate of over 10 GW per year globally up to 2030 (GWEC, 2018). In order to reduce the LCoE of the offshore wind and stimulate investment up to a needed level, innovation has to be steered towards the most capital-intensive aspects of OWF development.

Analyzing the cost of offshore wind farm (OWF) development (Figure 1.1), it can be seen that a significant part of it is due to the expenditures encountered during the construction and installation of the wind farm.

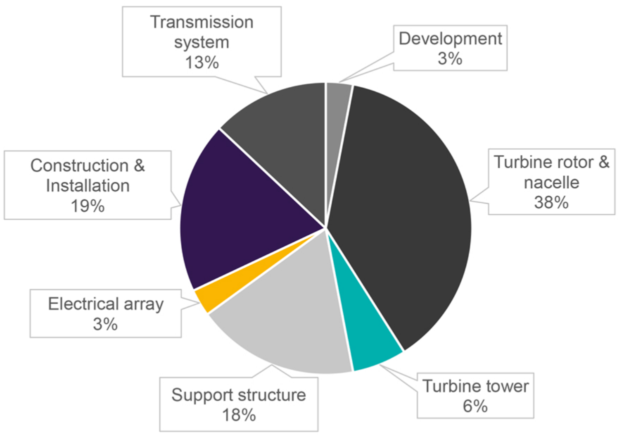


Figure 1.1: Total installed cost breakdown for a representative OWF in Europe (IRENA, 2016b)

The largest cost reduction opportunity (up to 60% of total reduction) to 2025 lays in the construction and installation of offshore wind farms (IRENA, 2016b). In particular. the following measures are proposed:

- Increasing the length of installation windows.
- Using larger vessels for substructure installation.
- Onshore pre-assembly of turbine components.
- Integrated turbine and foundation installation.

Apart from reducing capital expenditure (CAPEX), these measures could reduce the cost of financing due to an earlier commencement of generation (IRENA, 2016a).

Wind turbine manufacturers, site developers and foundation manufacturers keep introducing new technologies to the market. The average capacity of a single turbine installed offshore in 2017 almost reached 6MW, while the average wind farm size approached

500MW (GWEC, 2018). The latest news in wind turbine industry featured even larger machines of 10 MW (Siemens Gamesa, 2018) and 12 MW (GE Renewable Energy, 2018), which are expected to enter the market within the following couple of years.

In 2017 the first floating offshore wind farm Hywind Scotland was commissioned in Europe. The water depth at the site was in a range of 90-120 m. In the near future, up to 7 GW of floating wind farms are technically feasible to be developed by 2030 at the coasts of Japan, France and USA. Among the most active emerging markets, the Asian and the North American are expected to attract large investments in the coming years. There a complexity lays in the fact that very few sites are available for bottom fixed foundations because of great water depth. Hence floating foundations might become a solution (Equinor, 2018).

As the wind farm development moves further offshore, where winds are steadier and stronger, the installation process becomes a subject to harsh met-ocean conditions which limit vessel operability. It is likely that in the deep waters jackets (piled or with suction buckets) or floating foundations will be used in the future. The largest jacket vessel can only accommodate three of these structures, which results in a large amount of travels needed to complete an OWF of several hundreds megawatts. At the same time, turbine size keeps increasing, eliminating the possibility to use today's installation vessels effectively (IRENA, 2016a).

As a result, a complex innovative approach is needed in the areas of support structures, transportation vessels and installation logistics (LEANWIND, 2018). In order to facilitate the progress, the models which would simulate the real life processes need to be designed.

This research is focused on the development of such a tool for a wind turbine manufacturer Siemens Gamesa Renewable Energy (SGRE), and its Department of Offshore Innovation specifically, which is interested in assessing the viability of new support structure concepts, as well as investigation of possible improvements in the installation process.

1.2 Problem statement & Objectives

As described in the previous section, the wind industry moves further offshore, at the same time increasing the size of turbines and foundations. The problem which designers and developers are facing is in assessing the emerging concepts and taking decision on where to steer in-depth investigations. The focus has to be on the developments and ideas which promise substantial cost-reduction in offshore wind as such, and logistics pipeline specifically. Thus, the question to be answered by developers is:

Which new concepts in the area of offshore wind farm construction are the most promising to drive cost-reduction further?

Following a rationale given in the previous sections, the objective of the research is in developing decision support tool for the process of offshore wind farm installation planning. Among the questions which the tool needs to be able to address are:

- What are the consequences of employing certain strategies for installing foundations and turbines?
- What is the duration of OWF installation given its size, location, weather data, contracted vessel fleet, operational limits and installation strategy?
- Which aspects of the OWF installation can become bottlenecks and cause contingencies or delays?

- How does the installation of the novel substructure concepts compare to the installation of conventional ones?

As stated above, the tool is intended to support the Department of Offshore Innovation at SGRE during conceptual development of new support structures, as well as to help project planners with obtaining a realistic project schedule for new OWFs. The possible ways of utilizing the tool should not be limited to the above-mentioned points. Provided a great flexibility is achieved, the tool can serve for other OWF installation logistics-related purposes.

1.3 Research methodology

Based on a preliminary analysis of the industry and research trends (Sharda and Vazquez, 2009), accompanied by the preferences of SGRE, it was decided that the tool has to be implemented in a form of a program combining simulation with optimization (an elaborate comparison of the approaches to the design of decision support tools in OWF installation area and a rationale behind this decision are given in chapter 2). For this purpose, programming language Python will be employed as it contains several libraries created for the development of simulators and for mathematical optimization.

In its classic form, the life-cycle of software development can be represented as a circular process which starts with the analysis of the problem, followed by planning, implementation, testing and maintenance (Ruparelia, 2010). Due to the time limitations of this master thesis project, the process of development will have to be linear as multiple iterations would require significant time resource. Nevertheless, iterations are expected at each stage of development. Additionally, each of the stages cannot be treated separately and it is envisaged that many cross-dependencies will emerge between different stages (e.g. testing may reveal that alterations have to be done in the preliminary design, etc.).

The following activities can be identified as the key steps towards successful completion of the project:

1. **Planning.** Identification of the SGRE requirements towards the tool. Identification of boundary conditions and scope of the research. Formulation of the delivery objectives.

2. **Literature review and industry analysis.** Analysis of existing research and industry trends in the area of planning for offshore wind farm installations.

3. **Conceptual design.** First rough design of the tool which is intended to contain all necessary building blocks without their in-depth structure.

4. **Detailed design.** In-depth design of the tool architecture including flow of information and means of interaction between various blocks.

5. **Implementation.** An actual development of the simulator with the detailed implementation of all processes and inter-dependencies.

6. **Testing and validation.** Validation of the simulation results. This stage is supposed to confirm that the capabilities of the developed tool are consistent with the objectives formulated in 1.2. Additionally, this stage may reveal which stages of installation procedure can be improved and are cornerstones.

7. **Optimization problem formulation**. Formulate optimization problem by selecting objective function, decision variables and algorithm for the solver.

8. **Implementation of Optimizer**. Implement optimizer and connect it with simulator so that the latter is used to supply inputs into an objective function. Analyze results of a typical optimization.

9. **Analysis of the obtained results.** Ex-post analysis of the project outcomes. Draw conclusions and suggest points of attention for further research.

1.4 Structure of the report

The structure of the report is as follows:

Chapter 2 lists the main requirements towards the decision support tool. Based on an overview of existing trends in analysing offshore wind farm installation, a conceptual design of the tool is derived;

Chapter 3 contains an in-depth overview of the developed simulator;

Chapter 4 gives a rationale behind an approach selected to model weather conditions at OWF locations;

Chapter 5 presents the validation studies and results of simulations;

Chapter 6 focuses on the optimization procedure, where the most influential factors of installation process are optimized;

Chapter 7 draws conclusions from the conducted research and suggests the directions for further investigation.

Chapter 2

Conceptual design of the decision support tool for Offshore Wind Farm Installation Planning

This chapter is intended to present and justify a conceptual design of the decision support tool which was selected to address the objectives of this project. In order to drive further investigations, a list of the desired capabilities of envisaged tool is formulated. The rest of this chapter elaborates on the existing approaches for the planning and analysis of the OWF installation. An overview of the industrially- and academia-developed decision support tools is given. Analyzed tools can generally be divided into two categories: *Simulators* and *Optimizers*, where the latter can include simulation or be of a completely analytic nature. Finally, based on the list of requirements and conducted investigation, a high-level architecture of the tool is derived.

Note, that a lot of similar research has been done in the area of offshore wind farm operation and maintenance where multiple approaches for developing decision support tools have been proposed. Nevertheless, since it is not directly related to OWF installation, it will not be covered here.

As a result of the conducted survey, a Discrete Event Simulation (DES) approach was selected as the first and main technique for addressing the objectives of this project. Further, a Multi Integer Linear Programming (MILP) model will be developed to optimize the key variables in the OWF installation.

Section 2.1 contains a list of requirements which the developed tool has to comply with. Section 2.2 provides an outlook on the existing commercial tools for OWF installation planning. Next, section 2.3 describes academia advancement in the area of the simulation of OWF installation. Some scholars have advocated for the combined approach, where DES simulations are employed along with the Linear Programming (LP) optimization methods, and these are reviewed in section 2.4. Additionally, optimization with analytic evaluation and constraint formulation was embraced by several authors to investigate OWF installation. An overview of a few scientific papers in this area is given in section 2.5. In section 2.6 a conceptual design which will be exploited in this thesis is presented and justified. Finally, section 2.7 summarizes the analysis.

2.1 Requirements for envisaged decision support tool

A preliminary analysis of industry and academia trends was conducted. Following, several internal discussions were held with the Offshore Innovation Department of SGRE. As a result, it was decided that the following functionality requirements need to be addressed in the tool developed within this thesis:

- The tool needs to allow comparing different installation strategies.
- The tool needs to allow comparing different types of turbine substructures.
- The tool needs to be able to deliver a timetable of wind turbine and substructure installation for the wind farms of any size.
- A functionality of advising the user on the better decisions to improve current installation practices needs to be included.
- An ability to analyze the installation at different weather conditions needs to be included.
- The tool needs to allow to investigate different options for the vessel fleet. (Including analysis of different number and combination of vessels, vessels with different capabilities and limitations.)
- The tool needs to produce a detailed representation of the on-site activities performed by the vessels.
- The tool needs to be able to incorporate all relevant precedence requirements between the activities and other OWF installation logic.
- An ability to imitate different assembly strategies needs to be included.
- An onshore port consideration will have an added value.
- A computational time of the tool has to be reasonable, i.e. so as to allow utilization of the tool on a daily basis.
- The outputs of the tool must be targeted at users of various technical background.

Several ways to build a tool satisfying these requirements might be possible. Therefore, the following sections of this chapter will present an overview of existing trends in the development of OWF installation planners.

2.2 Industry state-of-the-art tools

As the wind industry keeps growing, so does the commercial appetite of offshore developing companies who face the increasing market pressure whilst competing for a wind capacity planned in the following years (PwC, 2018). On the other side, Engineering Procurement and Construction (EPC) companies try to seize the opportunity of establishing a solid track record by performing OWF installation orders in the most efficient, reliable and cheap way. Developers, who often hire the above-mentioned contractors, would like to be sure that the planning proposed by the installation company is indeed cost-efficient. For this purpose, virtually each involved company aims at developing its own in-house tools. In this situation, it is difficult to deliver a thorough analysis of the industrial advancements as the knowledge remains behind the borders of commercial secrecy. Consequently, only two examples of tools that are not covered in scientific papers are reviewed in this section.

ECN Install

One of the tools which is well-known in the European offshore wind industry is ECN Install. The tool was developed by the Energy Research Centre of the Netherlands in collaboration with Van Oord and Royal IHC, who are major industry players in offshore wind EPC (ECN part of TNO, 2018). The tool allows user to simulate the installation plan in the form of steps, which imitate specific installation activities, each with its own duration and operational limitations. As an input, working patterns of the crew, climate information, vessel and equipment used, harbours and turbine data, etc. are used. By averaging the results of several years of analysis, the total cost and duration of the project, as well as an insight into delays can be obtained (ECN part of TNO, 2015). The tool consists of a planner, a pre-processor, a simulator and a post-processor (Figure 2.1). The planner takes the user inputs about OWF specifications and specific sequences of offshore operations which are to be simulated. The pre-processor produces the schedule, not accounting for any delays. Following, the simulator block imitates the entire installation, generating the planning affected by possible weather delays. In order to incorporate possible weather uncertainties the tool is capable of generating synthetic weather series based on the user input, so that multiple simulations are performed. Finally, the post-processor makes sure that the produced schedule can be represented in a user-friendly way.

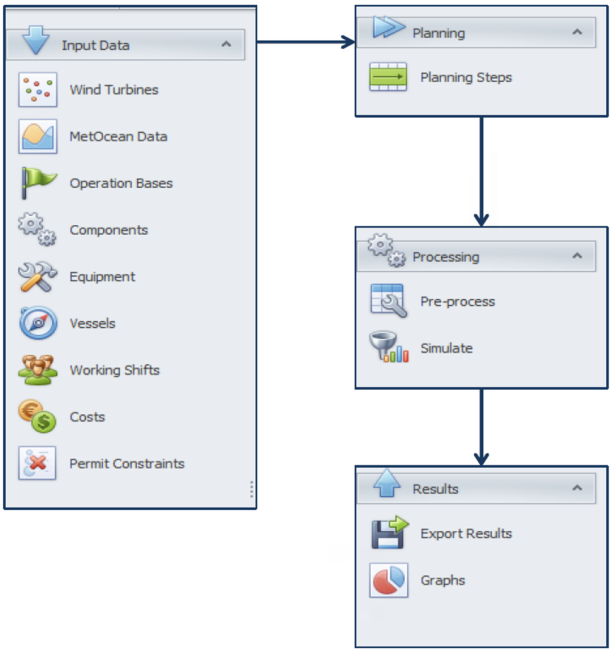


Figure 2.1: ECN Install (Kumar, 2017)

Overall, it can be said that ECN Install incorporates all major factors of the OWF installation processes and allows for a great flexibility which is important for project planners willing to test different installation strategies. The software developers employed a simulation-based approach in order to support project planning. Special attention was payed to a visualisation of obtained results so that a user with any level of insight can comprehend them.

LEANWIND

Another example of a tool, for which information is widely available, is a project planner developed within the EU Horizon 2020 LEANWIND project. It was, actually, a set of holistic optimization tools for various stages of the OWF supply chain that was developed (LEANWIND, 2018). The project developed Installation Vessel Optimizer (LIVO) to explicitly address the question of resource management (Figure 2.2). Moreover, as a part of the financial model developed within the project, an installation model was created. The purpose of the model is to calculate likely cost and duration of the installation. Similar to ECN Install, the module generates the schedule of activities, recording the intermediate activities and their duration. The model includes installation of turbine foundations, turbines, substations with foundations and cables. The tools are supposed to assist major offshore wind stakeholders such as manufacturers, operators and developers.

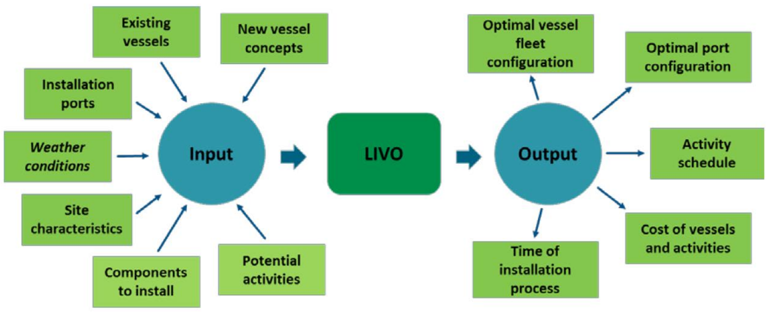


Figure 2.2: LEANWIND LIVO (LEANWIND, 2018)

Unfortunately, both ECN Install and LEANWIND tools are not publicly available and hence no insight into their architecture can be obtained. Nevertheless, they provide a valuable reference for this project. It can be seen that two different approaches are adapted to solve the specific problem. On the one hand, when it comes to obtaining a certain variable value which guarantees the best project performance, the optimization tools are used (such as the one for finding optimal vessel fleet in LIVO). On the other hand, when the schedule of a series of events is to be obtained, simulation is used (likewise in ECN Install). It can be derived that the main blocks, such as input parameters, weather blocks, planners, and post-processors, are crucial to construct the tool which would adequately address the needs of this thesis project.

2.3 Research with the focus on Simulation tools

One of the early works in the fields of offshore installation logistics simulation was published by Lange et al. (2012). The structure of the discrete event simulation (DES) tool contained separate modules for inputs, simulation and evaluation. After entering the data, a first "master plan" is proposed for a user to review. This plan describes production jobs and transportation orders for components. After that, several simulations are being run in order to account for stochastic nature of disturbing factors and weather. According to the paper, simulations performed with such a tool allow for carrying out a comparison of different assembly strategies, logistic strategies at sea (use of feeder vessels) and consolidation of components in a single port.

A series of papers has been published by Barlow et al. (2014a), (2014b) and (2015), where DES simulations have been widely employed in order to identify the most crucial and detrimental factors in offshore wind farm installation. Starting with a case study, investigating the impact of key heavy lift vessel logistical decisions, the simulation tool was first introduced in (Barlow et al., 2014a). The cost and difference in a project duration, as a function of number of vessels and their mobilization day, were studied. Next, an impact of key installation vessel characteristics was investigated in (Barlow et al., 2014b). The authors again focused on the realistic modelling of specifically the installation phase of offshore wind farm logistics. Capacity, average operational transit speed, wave limits for transportation and wave limits for jacking operations were studied and the most optimal values for these variables were obtained with the aid of a simulation tool. Later, in (Barlow et al., 2015), vessel operations most sensitive to the weather delays were identified. The operations undergoing the most weather delays were targeted as the main candidates for innovations.

An overall transport, assembly and installation of wind turbine components were simulated in (Muhabie et al., 2015). The authors underlined the main advantage of DES being the possibility to integrate operating rules of each installation process as well as to simulate the complex interconnection between different actors and resources. Also, it was suggested that the future research should focus on the optimization of the logistics chain. The same authors implemented a DES approach coupled with a metocean model in order to identify possible improvements for the scheduling of the OWF installation in (Tekle Muhabie et al., 2018).

A paper by Vis and Ursavas (2016) is remarkable due to an accurate explanation and a schematic overview of the installation module. The authors incorporated different assembly strategies (single-blade, full rotor and bunny-ear). Finally, the authors point attention to the fact that the decision support tools such as the one developed should be used in order to accurately obtain the most promising solution for a specific project.

A research by Beinke et al. (2017) is unique in its scope as it analyzed the joint use of resources for a simultaneous installation of several offshore wind farms by means of a multi-agent DES tool. A comprehensive network of offshore logistics chain was depicted. The objective of the study was to investigate the degree of resource utilization and an installation time in a typical non-sharing scenario compared to the more collaborative cases. It was concluded that a joint approach can lead to a significant reduction in the total duration for each project, accompanied by a higher utilization rate for each vessel.

As mentioned in (Sharda and Vazquez, 2009) and (Terrazas-Moreno et al., 2012), often a number of decision variables and physical relation between them is too complex to be modelled as an optimization problem. Thus, it has to be tackled by means of multiple simulations where the natural processes do not need to be described in the algebraic terms. This helps to avoid a need of over-simplification when developing a model. As a result, a simulation has been widely favored among researchers. Referring back to the requirements for the tool formulated in 2.1, it can be seen that *simulation* is a powerful approach which allows to address many of them.

A generic structure of a simulation tool for OWF installation can be described by the following Figure 2.3. On this scheme, a *simulator* block is a computer code which contains all the logic needed to realistically mimic the process of OWF installation.

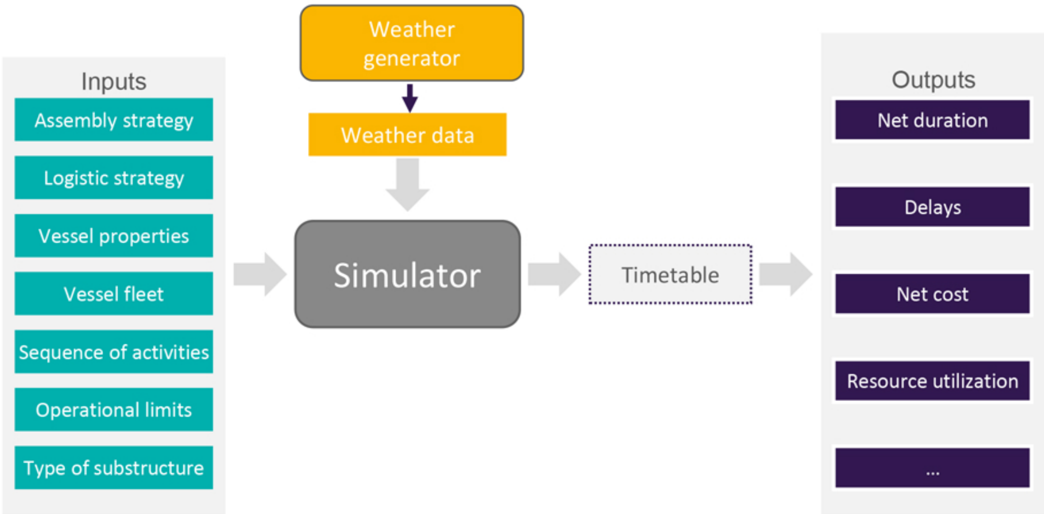


Figure 2.3: Generic structure of simulation-based OWF planner

2.4 Research combining Discrete Event Simulation and Linear Programming Optimization

As it was mentioned in the previous sections 2.2 and 2.3, an essential drawback of simulation approach for planning is its intrinsic inability to advice. In other words, a result is completely determined by a user input and optimal values of the input parameters cannot be computed via simulation. For this reason, several scholars attempted to combine simulation and optimization in order to capitalize on the strong points of each technique. There are several papers not related to OWF logistics, where DES simulation was integrated with the optimization (e.g. as a mean to calculate the values of optimizer's objective function). Examples are (Castro et al., 2011), (Terrazas-Moreno et al., 2012), (Shin et al., 2018). Authors explicitly point out the weaknesses of simulations which lay in the fact that they have fundamentally descriptive function and cannot advise on an optimal planning. Oppositely, as a strong point, the ability to represent complex system interrelations is mentioned.

In the paper by Kerkhove and Vanhoucke (2016) an optimization objective was deliberately assigned to be of a financial nature - maximize an OWF net present value . The base decision variables are *gate times* indicating when a resource for performing certain task becomes available. This research also presented three heuristics to reduce the solution space. It turned out that significant performance gains can be achieved by selecting proper heuristics. Due to its large size, the solution space had to be reduced in several steps: in order to account for precedence relations between various activities, only feasible schedules are selected; then, due to the repetitive nature of installation process solution space is reduced by aggregating activities per month. The model used DES to evaluate the objective function for a specific schedule. The schedule is, however, represented in an abstract way, in terms of high-level encoded variables called *chromosomes*. Depending on the level of abstraction and encoding, different heuristics were applied to obtain a solution. Further, a re-encoding algorithm is used to obtain the final schedule in terms of activities' starting time. It can be seen that the proposed model entails highly abstruse representation of the installation process. Such a model succeeds in producing an optimal schedule but requires additional algorithms to be used. The simulation itself does not have a high level of flexibility and as such is an integral part of the optimization model.

To the best of authors knowledge, only one group of scholars has applied combined MILP and DES in the area of offshore wind farm installation (Barlow et al., 2018).
The difference with the work by Kerkhove and Vanhoucke, is that Barlow et al. developed DES as an independent tool, working with the complete set of realistic sequences, in a way that human can read it and analyze. In contrast, Kerkhove and Vanhoucke used DES with mathematically abstract and encoded *chromosomes*, which were designed specifically for the purpose of their MILP model. The *chromosomes*, despite containing similar information as sequences in Barlow et al., are impossible to analyze without additional encoding and re-encoding algorithms.
This paper by Barlow et al. (2018) is a continuation of an earlier work by Barlow et al. (2015) and Tezcaner Öztürk et al. (2017), where an optimizer from the latter is integrated with a simulator from the former. The paper emphasized a need to develop a mixed-methods approach in order to account for seasonality of operations (simulator) and to be able to explore large decision spaces to identify the optimal scheduling of operations (optimizer). At the same time, the need to run too many simulations (up to 1000) would be eliminated and an optimizer failure to incorporate seasonality would be addressed by simulator. A robust optimization technique was employed to find estimated task start times which minimize total project duration. While each process is potentially subject to uncertainty, the maximum number of uncertainties is defined by user. Three constraint sets were included: precedence constraints, task ready times (time when a vessel is contracted for its first task, user-defined) and task deadlines. For the case study consisting of 120 turbines and 2 vessels, 1000 simulations were performed for each possible starting day, that is to provide statistical accuracy. Next, the case study was solved using a robust optimization model to suggest starting days for the support operations. Finally, authors suggest two types of hybridisation between simulations and optimizations:

- Optimization model applied as a preliminary step to identify the optimal scheduling of the sets of operations (with respect to average yearly weather). Simulation model is then utilized to investigate monthly variations.

- Use simulation model to identify the tasks which are most susceptible to weather delays. This is used later to explicitly define deviating tasks for running robust optimization model.

Interesting is that the developed framework, as claimed in the report, is used by SSE Renewables to support logistical planning of the Beatrice OWF, a 600 MW wind power plant located in the North Sea.

A sophisticated design of a decision support tool was proposed in a master thesis by Kumar (2017), where an additional optimization module was added to the above-mentioned ECN Install (see 2.2). Unlike in the original tool, the new architecture allows to create automatic schedule for the installation based on the particular choice of resource done by optimizer (Figure 2.4). Among the major upgrades which were implemented, an *automated planning block* and an *optimizer* are remarkable. The former prepares a plan based on an installation strategy selected by the user. The latter, as the name suggests, is an essential extension for the ECN Install which allows not only to generate the schedule but ensures that a combination of certain installation parameters is optimal.

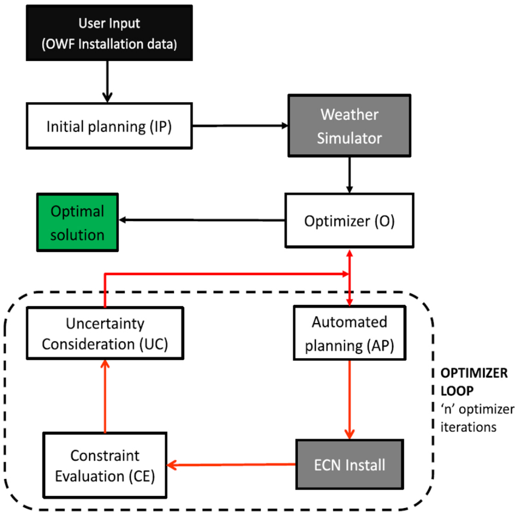


Figure 2.4: Upgraded ECN Install architecture (Kumar, 2017)

The decision variables for the optimization are project start date, vessel division, vessel type and number, harbor and wind turbine pre-assembly strategy. The objective function to be minimized is the total cost of the installation, which is however affected by an *end date penalty function*. This penalty function is deemed to reflect a requirement to complete an installation by a given date. Interesting is that contrary to the papers described in the followings section 2.5, the constraints in this study only reflected the feasibility regions for decision variables, not the precedence requirements. The latter ones were explicitly evaluated in the simulator which significantly simplifies the formulation of the optimization problem and can be sought as a more straightforward approach with a lower level of mathematical abstraction.

Generalizing this section, the main distinguishing feature of the described approach is using an optimization algorithm with integrated simulator, where the latter is responsible for the formulation and evaluation of constraints and objective values.

2.5 Research in Optimization tools with analytic evaluation

In contrast to the optimization tools with integrated simulation (as described in 2.4), it is also possible to formulate an optimization problem completely analytically, i.e. in terms of algebraic equations. This way an objective and constraints are not evaluated via simulation. Several scholars adopted this approach for OWF installation planning.

2.5.1 Finding optimal schedule

Some of the most frequently referenced papers in OWF installation optimization literature are parts of the research by Scholz-Reiter et al.: (2010), and (2011a). A general objective of multi-integer linear programming (MILP) model was to minimize the building time of a wind farm consisting of 12 turbines and employing a single vessel. The model allowed to define several loading sets for a vessel, as well as incorporated three types of weather conditions. The outcome of an optimization is an optimal schedule of vessel activities. Below, figure 2.5 presents schematic picture of the model architecture.

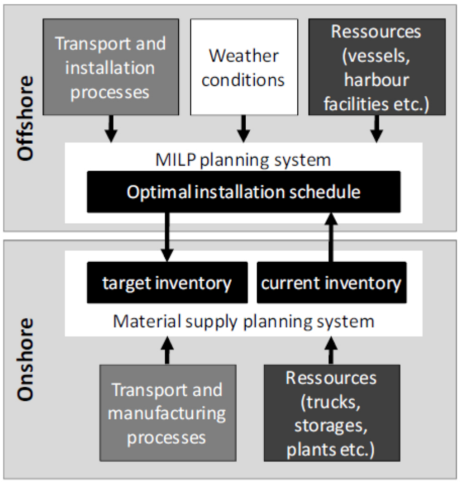


Figure 2.5: Concept of planning system (Scholz-Reiter et al., 2010)

The authors concluded that due to the NP-hardness (the time required for a solution of a problem is a function of inputs size, which is at least bigger than polynomial) of the optimization problem, it is difficult to obtain an exact solution for real-size problems. For this reason, later in (Scholz-Reiter et al., 2011b) a heuristic was developed allowing for scheduling projects of a larger size and for longer time horizons. To the best of author's knowledge, this paper is not publicly available.

The next paper in the topic of OWF installation planning optimization was released by Ait-Alla et al. (2013). The objective of the developed MILP model was to minimize the total cost of vessel chartering during the total planning horizon. The decision variables set by MILP were integer and binary variables indicating how many components have to be processed by each vessel in a given planning period. 30 turbines and 3 scenarios of vessels' combinations were considered. An optimal aggregate plan provided the number of components to be installed by each available vessel to meet the weather forecast.
This research also embedded weather and time limitations into the mathematical model. All precedence-relations of the installation process were expressed as algebraic constraints on decision variables. Therefore, no explicit simulation took place.

A two-stage optimization model was developed by Tezcaner Öztürk et al. (2017), based on the previous research by Barlow et al. (2014a). First stage is a deterministic model resulting in an optimal schedule with the aim of minimizing total duration or cost. The decision variables (start-times for operations) are optimized, subject to precedence relations, task ready times and task deadlines expressed in the form of algebraic constraints. It, however, did not consider an uncertainty in a duration of installation processes and non-operational periods. Second stage is another robust optimization model with a predefined number of parameters which deviate. For both models an algorithm to assign assets to vessels is developed which assigns the next asset to be installed to a vessel that could finalize the task sooner. This algorithm is applied prior to running the optimization, hence all the sequences and streams are known. The final dates are set as the constraints. The author of this thesis was only able to find a manuscript version of this research which was not finished, and therefore no conclusions can be derived.

2.5.2 Finding optimal vessel fleet

A different type of the problem was addressed via analytically-defined optimization in the master thesis by Hansen and Siljan (2017). The objective of the developed tool was to find an optimal vessel fleet which minimizes the charter cost and time span, in contrast to a common scheduling problem. Thus, important parameters included cost of chartering vessels, needed to obtain a value of objective function. A set of decision variables includes information whether: a certain vessel is included in the optimal fleet, a vessel is chartered in a certain moment of time, a vessel is loaded in a certain moment of time, and start- & end-time of operations. An objective function minimized the total cost, plus added a penalty cost implemented to motivate a minimization of total duration. To guarantee the precedence relations between the activities, a precedence matrix is constructed which is later included in one of the installation constraints. Other constraints reflect different installation process logic, such as the fact that a vessel is only able to install components as long as it carries them on board, vessel has to return to port in order to reload, etc. Two models, original and pattern-based, were used in order to speed up computational time. Authors used different methods to generate patterns which significantly reduced solution space and allowed to solve problems for the OWFs of up to 100 turbines. In the end, the authors underlined that a size of the problem can increase exponentially with the vessel deck space, as one of parameters affecting decision variables and solution space. This might cause the problems in finding a solution.

A generic representation of the reviewed optimization tools is given in figure 2.6.

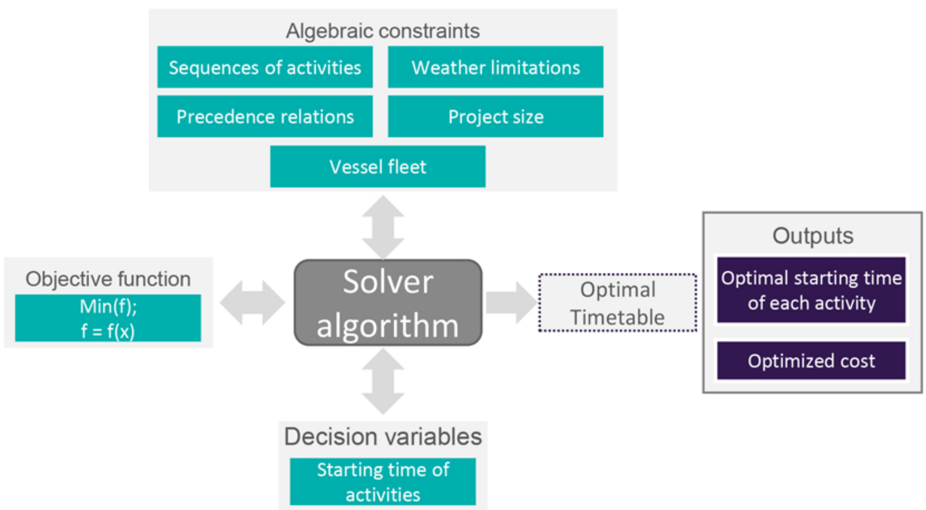


Figure 2.6: Generic structure of OWF installation optimization software

It can be seen that a typical analytic optimization tool contains the following blocks:

- Decision variables - describing the real-world parameters which are optimized (e.g.

starting days)

- Algebraic constraints - mathematical expressions imitating real-world conditions and requirements
- Objective function - algebraic function that maps decision variables into some numerical value (e.g. total duration)
- Solver algorithm - mathematical algorithm for scanning the space of feasible solutions so that the value of objective function is minimized (in the classic formulation)

In the above-presented papers all sequence- and precedence-requirements for the process of installation (e.g. at any given moment the vessel needs to first travel to the site and only then start installation of a component) were reflected in the *analytic* MILP constraints of *optimization* problem and not as a simulation code. Their value was obtained by solver algorithm by means of *analytic evaluation*. Moreover, no *simulation* was developed to calculate the value of an objective function. These facts are essential in understanding the difference between approaches in section 2.4 and this section.

As an outcome of the computations, an optimal schedule is produced. On the one hand, the need to algebraically express properties of the real-world makes such optimization approach less flexible towards testing different installation decisions. Sometimes these algebraic constraints become very cumbersome and hard to analyze. On the other hand, optimization tools without simulators allow to obtain optimal values for certain parameters faster since the objective function is computed as an algebraic equation, rather than as a computer simulation.

2.6 Decision support tool for this project

All of the described tools might have a certain level of limitations in the way they mimic offshore operations, assign activities or sequences to a specific vessels and imitate precedence of certain processes. Albeit it all depends on the research focus and priorities, the more flexible and comprehensive the tool is, the wider is its range of applications and usefulness for the engineers and project planners.

From the list of requirements towards the tool (section 2.1) the reader can see that a substantially high level of flexibility needs to be achieved. Thus, it was decided that the main block of the developed tool has to be a simulator, carrying the functionality to represent an OWF installation process as realistically as possible. Moreover, to improve the computational speed, a discrete event simulation approach is selected. According to DES paradigm, the system may change its state only at precise points of time, when an event happens. DES approach is known for its realistic representation of complex processes (Banks, 1999). At the same time, computations can be done faster since only discrete points of time where an event occurs are relevant, in contrast to continuous simulations where computations are performed at each time step (Matloff, 2008), (Nance, 1993).

The simulator would be developed in a general-purpose programming language with a wide range of extension libraries and packages helping developer to tune a program to a client's needs. For this purpose Python was chosen as it already contains a SimPy library for developing simulators (including DES). This allows to save time and resources needed to develop low-level simulation framework (handling events, time, resources) and to focus on designing the simulator specifically for OWF installation planning.

Further, in order to assist a user in selecting the most promising strategies, an optimizer would be added. An optimizer, however, will have limited functionality focusing only on the variables that turn out to be the most pivotal as a result of multiple simulations. Because an optimizer is inherently less flexible, a well-thought choice has to be taken for defining decision variables and objective function for the optimizer. An approach similar to those described in section 2.4 would be taken to connect two blocks.

A developed tool will consist of two major blocks. Main of them being DES simulator, developed in Python programming language. An available discrete event simulation library SimPy will be employed. Another block will be an optimizer, searching for the best combination of variables which are crucial for the installation according to the results of simulation. As mentioned in literature, several hundreds of simulations may be needed to achieve statistically accurate results. Therefore, an ancillary block responsible for generating artificial weather data needs to be included. Finally, an output has to be visualised so that wide range of potential users can utilize the tool.

A high-level schematic representation of the intended tool design is presented in figure 2.7 below. Note, this diagram only intends to show the envisaged connections between building blocks.

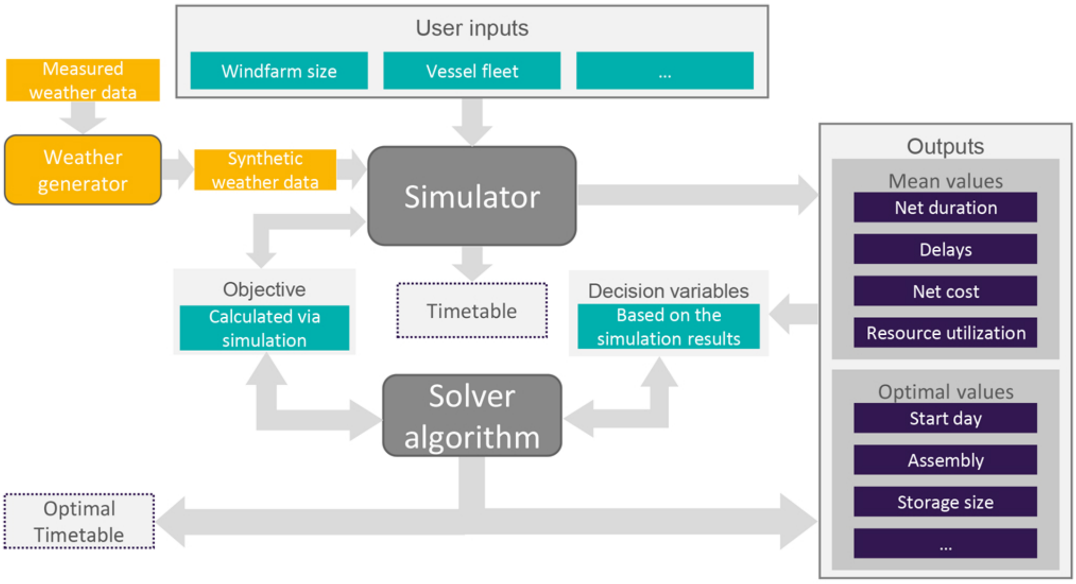


Figure 2.7: Conceptual design diagram of the tool in this project

The following chapter 3 introduces a design of the simulator part, with an accurate representation of all building blocks, while chapter 6 is dedicated to the optimization.

2.7 Summary

In order to satisfy a wide range of industry needs, a developed decision support tool has to possess a high level of flexibility. Multiple installation strategies and processes might be of interest, thus the tool needs to be able to incorporate any of them. A list of requirements was drafted to steer the research.

In order to draw out a conceptual design of a tool an overview of existing OWF installation planners has been done. Both industry and academia has been working on designing the tools to support decision-makers in the OWF installation planning. Not much information is available about industrially developed tools, since these are commercially highly valued in-house developments used in competitive markets. On the other hand, a vast amount of research in academia is available. Two dominant approaches can be distinguished - *optimization* and *simulation*. Whilst developed for the same reason of improving current installation practices, they carry fundamentally different functionality and application scope.

As such, simulators are descriptive tools which can be relatively easily adjusted to the specific needs of a client. A significant level of flexibility and speed of performance can be achieved, allowing to imitate all kinds of installation strategies and approaches. A single simulation cannot advise on the best decision to be taken.

Oppositely, optimization can be utilised with the purpose of finding the best set of variables (e.g. vessel number, starting dates, assembly strategy, etc.) to guarantee an optimal project plan. However, depending on the way the calculations are done (analytically or via simulation), optimization tools might require more time for the calculations. Often, there is a need in additional algorithms to obtain solutions for the OWFs of realistic size. Moreover, optimization tools are less flexible when it comes to describing complicated physical dependencies and precedence requirements. Often, these have to be expressed in terms of algebraic inequalities which makes the tool less straightforward for a perception by an average user.

As an alternative to the above-described methods, the two can be combined to capitalize on the strong points of both. In other words, the simulator can be developed as an independent tool, which does not necessarily need optimizer to be used. Later the optimizer is *"wrapped"* around the simulator and uses it as a mean to compute a value of objective function and constraints. This approach has been selected for the current master thesis. A conceptual design including major building blocks has been presented.

Chapter 3

Simulation of OWF Installation

This chapter is intended to cover the architecture of the developed simulator. First, the reader is introduced to the main definitions and concepts of Discrete Event Simulation in section 3.1. In order to facilitate even better comprehension an example is given. Next, section 3.2 presents a structure of the developed simulation tool describing its blocks and information flow. In section 3.3 it is analyzed how successful Simulator is in fulfilling the requirements posed towards the envisaged tool. Most of the requirements are covered by Simulator, with an exception of advising a user on the best OWF installation parameters. A detailed step-by-step overview of a simulation run can be found in section 3.4. Finally, a summary of the chapter is given in section 3.5.

3.1 Main elements of DES

Before diving deeper into the architecture of the developed tool, it is worth to introduce Discrete Event Simulation and some related notions. While some of these notions are unique and widely-known, others are formulated by the author specifically for the purpose of this thesis, based on the SimPy conventions (SimPy, 2018).

- ***Discrete event simulation*** - an imitation of a real world process or system evolution over time, where a change of state can happen only at the precise moments of time at which events occur (Banks, 1999).
- ***System time*** - a time which has lasted in a simulated environment from the moment of simulation start.
- ***System variables*** - containers of information needed to describe a state of the system at any given moment of time (Nance, 1993).
- ***Event*** - an occurrence which marks a moment of time when a change to a system state takes place (Nance, 1993).
- ***Entity*** - representation of a real-world object, with its attributes. May be dynamic if it moves through the system, or static if it serves other entities (Banks, 1999).
- ***Process*** - an abstraction of a real-world process, characterized by its duration. Beginning of a process is normally triggered as a result of the completion of another process. In the real world, during a process execution some changes would happen to a system. In the simulation world these changes happen instantly at the end of a process. Thus, the end of a process marks an occurrence of an event which changes a system state.
- ***Event queue*** - a set of events which are scheduled to occur as a result of completion of some process. One or several events may be stored in the queue at the same time.

Example

An example in the OWF installation would look as follows:

Discrete event simulation is used to imitate a vessel(***dynamic entity***) loading in a port(***static entity***). The first ***event*** occurring in the system at a ***system time*** t=0 is the vessel entering the port. This event marks the change in the state of the system because now the vessel is located inside of the port. The ***system variable*** describing the number of vessels in port is changed to 1. Next ***process*** is loading of a tower which takes ca.5 hours. If no other ***event*** is scheduled in the ***event queue*** between t=0 and t=5, ***system time*** will directly jump to t=5. At t=5, ***system state*** is changed again, since a tower which used to be in the port, is now loaded to the vessel, so the ***system variables*** which describe number of towers in port and on the vessel deck have changed .

The main characteristic of DES, and at the same time its main advantage, is the fact that between consecutive events no change in the system can occur, and thus system time can be increased in a step-wise manner till the instant of next event occurrence. This facilitates shorter computational time compared to corresponding continuous simulations (Tekle Muhabie et al., 2018). Different paradigms for viewing the world can be implemented via DES. Among the possible variants are activity-oriented paradigm, event-oriented paradigm and process-oriented paradigm (Matloff, 2008). Description of each of them would fall out of this thesis scope. Nevertheless, it is worth to mention that process-oriented paradigm is the one implemented within SimPy framework. A process-oriented paradigm allows to add several threads (i.e. vessels, in studied case) into the simulation world. (Not to be confused with multi-threaded programming which is not implemented in SimPy). Each of them is running in parallel but shares common system variables.

3.2 Detailed architecture of the developed simulation tool

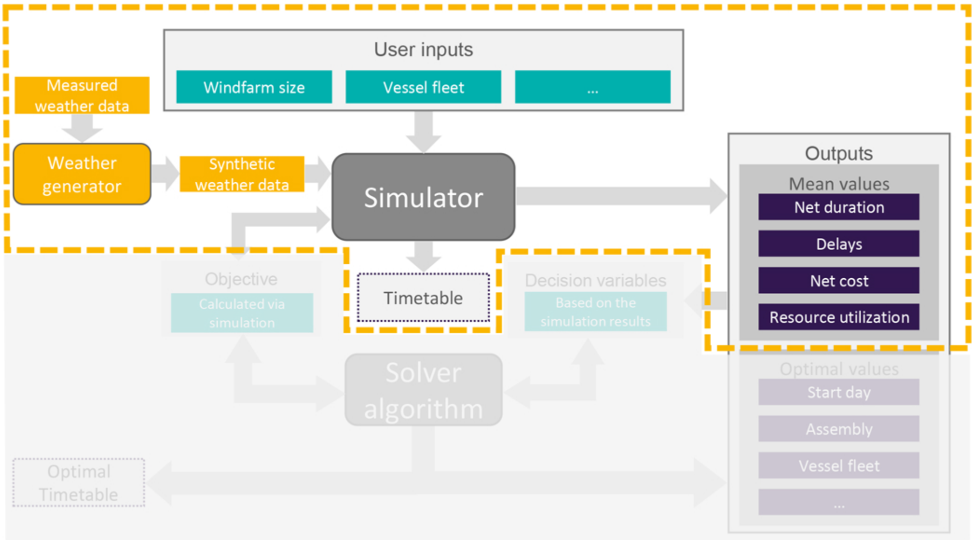


Figure 3.1: Simulator diagram

For the sake of clarity, figure 3.1 repeats figure 2.7 and highlights the simulator part which is the subject of this chapter. The main building-blocks of Simulator are:

- User inputs
- Weather module
- Simulator
- Timetable
- Outputs

A detailed explanation is presented below.

3.2.1 User inputs

User specifies the main parameters of the installation process in a dedicated Excel file, which is read by a Python code. The parameters to be specified are:

- **Size of the wind farm** - number of turbines (and corresponding support structures) to be installed.

- **Starting day** - beginning of the entire campaign counting from the January 1^{st}.
- **Sequence of activities** - a set of installation activities required for specific component (i.e. turbine or substructure) which have to be performed offshore or onshore in port.
- **Min substructure number for turbine installation** - minimum number of substructures which have to be installed for the turbine installation to commence (at least one separate vessel for turbine installation is implied). If a single vessel is contracted for the entire OWF construction, then this parameter is irrelevant.
- **Starting day for turbine campaign** - in case **Min substructure number for turbine installation** is not specified, this parameter will determine the first time a turbine vessel will enter a port to load turbines and begin their installation.
- **Min piles number for jacket installation** - analogically to **Min substructure number for turbine installation**.
- **Starting day of jackets campaign** - analogically to **Starting day for turbine campaign**.
- **Vessel fleet** - number and type of vessels contracted for the project.
- **Vessel characteristics** - properties of each of the contracted vessels (capacity, speed,weather limits).
- **Type of substructure** - monopile, or jacket with piles, or jacket with suction buckets, or floating substructure, etc.
- **Starting day** - the day of a year when the installation begins.
- **Onshore supply time** - time that it takes to deliver new components in the onshore port from the moment they were ordered. For jackets, ordering of components can be disabled and assembly in port will be imitated.
- **Operational limits** - duration in hours, wind and wave limitations for a certain *activity-vessel* pair.
- **Port characteristics** - storage capacity per component, initial stock, vessel capacity of the port.
- **Assembly of jackets** - certain sequence for onshore jackets assembly; number of assembly lines.

Based on the internal discussion within SGRE, it was concluded that these parameters are needed to completely determine the way an installation will be performed. The user may try different combinations of parameters in order to test how various planning choices affect the outcomes of a simulation.

3.2.2 Weather module

The weather module is responsible for supplying needed number of weather realizations so that multiple simulations can be run as described in chapter 2. Sufficient amount of measured weather data is needed to produce realistic synthetic series preserving statistical properties of the original data. In order to generate synthetic weather a Markov chain approach is employed, which is described in a detail in chapter 4. Depending on the envisaged number of simulations to be performed, a corresponding number of weather realizations will be generated. These data series are all different, however preserve all statistical properties of a given location. Therefore, the process of generating weather data has to be performed every time a new location is studied.

3.2.3 DES model

DES model block is an actual simulator where all inputs are merged together to define the system variables of a project. This block runs the discrete event simulation imitating the real-world OWF installation (described in section 3.4). All intermediate events and time when they occur in the simulation environment are documented whilst running in order to generate a timetable. DES block ensures that during the installation all precedence requirements and strategy specifications are followed. Depending on the weather data, a simulation with the same user inputs will result in different timetables.

3.2.4 Timetable

Timetable is the most general type of output which contains a highly detailed time-information about each step of the installation process from a single DES run. An example is presented below in table 3.1. Having this information, different key performance indicators (KPIs) can be calculated. A timetable can optionally be saved in Excel file, so that a user can thoroughly analyse how the installation proceeded. A number of generated timetables (i.e. a number of simulations with different weather realisation) can be defined by user. The higher the number is, the more accurate statistical outcomes can be obtained.

Table 3.1: Example of an output timetable

Abs. time [hr]	Day	Action	Component installed	Number	Vessel name
		...			
2355	98	Arrived to the site			Vessel 1
2355	98	positioning			Vessel 1
2357	98	prepare crane			Vessel 1
2358	98	install monopile			Vessel 1
2361	98	remove crane			Vessel 1
2362	98	prepare crane			Vessel 1
2363	98	hammer monopile			Vessel 1
2366	98	remove crane			Vessel 1
2367	98	Installed	monopile	6	Vessel 1
2367	98	All on-board installed.			Vessel 1
2367	98	anchor up			Vessel 1
2368	98	Start travel back			Vessel 1
		...			

3.2.5 Outputs

Based on the results of multiple simulations, different timetables are obtained. They are processed to derive the values of the most common and representative KPIs:

- **Net duration** - average total duration (in days) of the complete OWF installation.

- **Vessels' workability** - percentage of the total contracted time when each vessel was busy performing some operation.
- **Supply chain delays** - average total duration of delays caused by the onshore supply of components to port.
- **Weather delays** - average total duration of delays caused by the adverse weather conditions. It is possible to identify the distribution of weather delays among all activities.
- **Substructure and turbine installation progress** - the rate of substructure and turbine installations (number per day).
- **Net cost** - average total cost (based on the vessel chartering daily rates, cost of area rent in port and assembly cost) of the installation.
- **Cost per resource** - average cost of installation contributed by a specific segment, e.g.vessels, labor, assembly, etc.

More examples of output information will be given further in chapter 5, dedicated to the validation of Simulator. Figure 3.2 shows some of them.

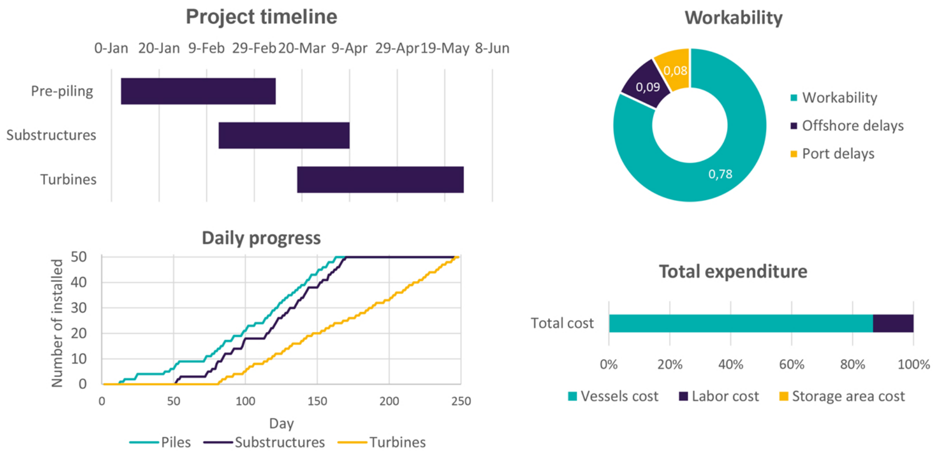


Figure 3.2: Example of Simulator outputs

3.3 Fulfillment of the design requirements

A list of requirements for the envisaged tool was defined in section 2.1. Here an overview of how these requirements are addressed by Simulator is presented.

Table 3.2: Design Requirements review for Simulator block

• Comparing different types of turbine substructures	Addressed by allowing user to select different vessel fleet, different starting days, different offshore activities.
• Comparing different installation strategies	Addressed. The type of substructure is one of the inputs, determining which offshore activities are performed.

• Deliver a timetable	Addressed. The primary outcome is detailed timetable where all performed activities are documented.
• Advising the user on better decisions	Not addressed by Simulator. Has to be addressed by the Optimizer.
• Analyze the installation at different weather conditions	Addressed. included weather module, responsible for generating synthetic weather series.
• Different options for the vessel fleet	Addressed. The user may specify his own parameters for vessels, as well as their number and type.
• Detailed representation of the on-site activities	Addressed. The user has the freedom to specify his own sequence of offshore activities and their operational limits.
• Precedence requirements and other OWF installation logic	Addressed. *"Hard-coded"* in the tool. For instance, jacket can only be installed where piles have already been driven in the seabed, turbine installation requires tower and nacelle to be installed in one go, etc.
• Imitate different assembly strategies	Addressed, offshore installation activities and loading can be determined by the user as an input.
• Onshore port consideration	Addressed as a storage facility with jackets pre-assembly lines (further described in chapter 6).
• Reasonable computational time	Addressed. Computational time for a single simulation is between $\sim$ 30 to 60 sec. In order to account for weather uncertainty, several simulation runs need to be performed. Usually, up to 30 runs with different weather is enough, resulting in $\sim$ 20 min. (Further described in chapter 5 with an example.)
• Outputs targeted at users of various technical background	Addressed, the outputs of the simulation are saved in standard office software. Moreover, several types of graphs can be built.

3.4 Flow-diagram of a simulation run

The following diagram 3.3 is deemed to represent at a medium level of detail how a single OWF installation is simulated by DES block (see 3.2.3). Below-given elaboration is not intended to give an exhaustive description of all rules, decisions, activities and interactions which are implemented in the simulation. Their description would draw reader's attention into technical details and would be onerous to visualize. Instead, only the main steps and decisions are depicted to give an idea of the software capabilities.

Scope of Simulator, as well as of the entire tool, comprises only installation of substructures and turbines. Cabling, installation of scour protection and offshore transformer stations are not included. Onshore - the boundaries of the tool are represented by a harbour facility where pre-assembly takes place.

Note, that some of the weather window checks may be performed differently by different contractors or vessel captains. The following logic has been implemented based on the common practices assumed in SGRE. The software, however, allows to change the duration of weather windows which are checked e.g. when leaving port or moving to the next turbine when operating offshore. All stages of the installation process, as well as intermediate checks were included in order to realistically imitate an installation in real life. Several discussion were held within SGRE to determine the necessary level of detail.

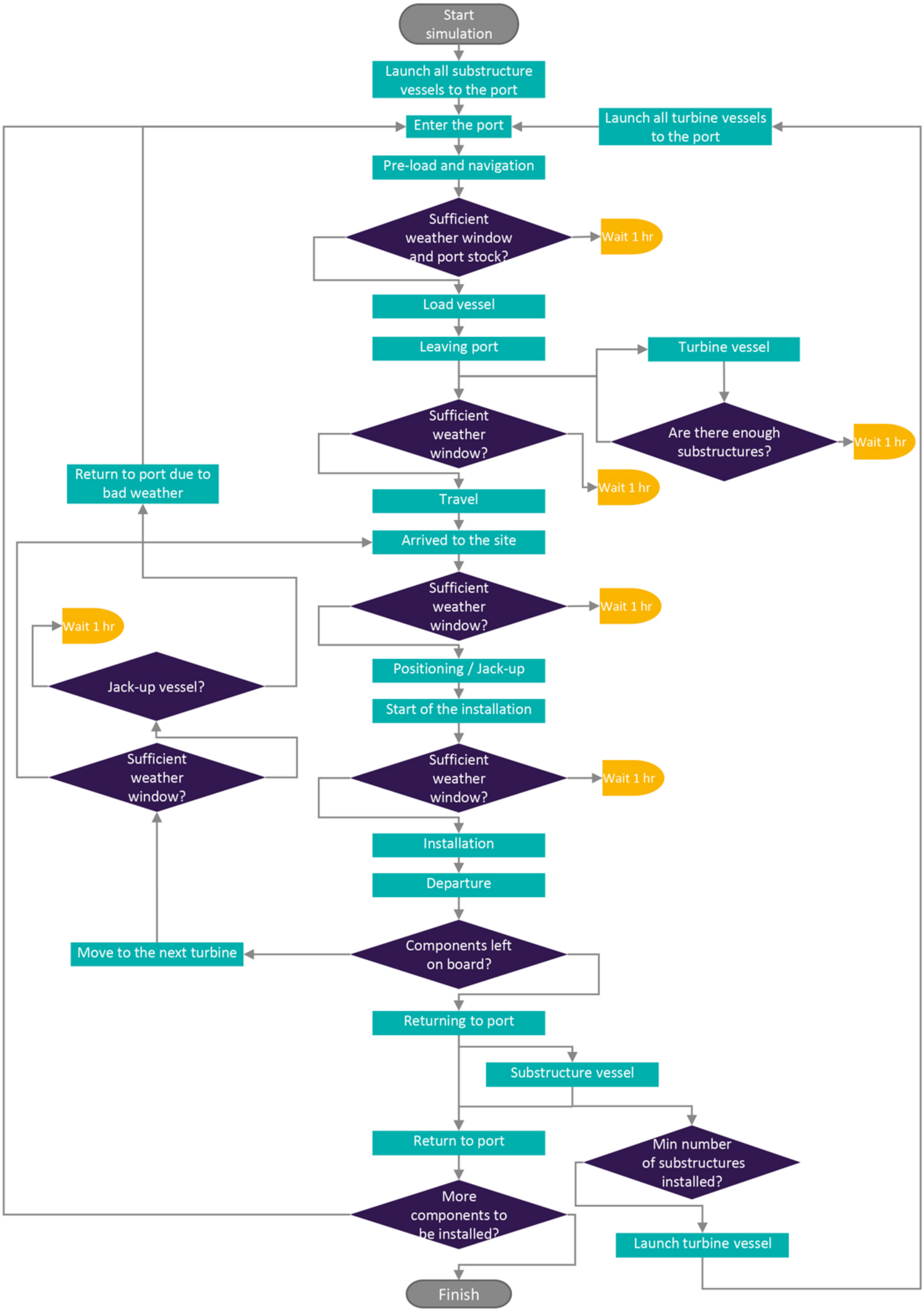


Figure 3.3: Block-diagram of a single simulation run

Loading in port

Construction process begins with launching to port vessels working on substructure installation (***Launching all substructure vessels to port***). Once any vessel enters the port, certain time passes until it begins loading. This time is required for pre-load and navigation operations which do not have any weather limits but need to be performed (***Pre-load and navigation***). Once the vessel has been positioned at the quays, a sufficient weather window for loading of specific component to the specific vessel is ensured. It is also checked whether the needed components are available in port (***Sufficient weather window and port stock?***). Is any of these conditions not satisfied, the vessel will wait, simulation time passes one hour (***Wait 1 hr***), and both checks are performed again. If two conditions are fulfilled, the loading starts (***Load vessel***). When loading is finished, the vessel is ready to leave the port (***Leaving port***).

If the number of components in the port stock falls below certain threshold, new components are ordered and assembled (not visualized in the block-diagram). It is also possible to specify a certain rate of assembly, i.e. every n days new component is assembled in port and is ready to be loaded on a vessel. This way it can be analyzed what is the needed rate of assembly to avoid accumulation of components, and at the same time minimize waiting time of vessels in case components are not yet assembled.

Departure from port

Prior to the departure from port, a captain of floating vessel will assure that within the following 72 hours (period of reliable weather forecast) the weather will not violate transition state limits, so that the vessel can safely sail (***Sufficient weather window?***). Provided this is the case, the following weather check will verify if at least one full installation can be performed and vessel can return to port safely within this 72-hour period. In case with a jack-up vessel it is assumed that once jacked up, the vessel can survive virtually any weather conditions, and therefore the weather check at port will only ensure that a vessel can safely arrive to the location and jack up. After that, if needed, there might be a period of waiting for a weather improvement to actually begin installation. Finally, in case it is turbine installation vessel, it has to be confirmed that there are enough substructures already installed offshore, so that turbine installation can be performed (***Are there enough substructures?***). If not, a turbine vessel will wait in port until there are enough support structures to install all turbines that it currently has on board.

Travel to the site and arrival

Next, the vessel will travel to the site (***Travel***). Once arrived, a jack up vessel will wait for good wave conditions to perform jack-up, whilst floating vessel will position itself next to the spot where a component (substructure or turbine) has to be mounted (***Positioning / Jack-up***). Then, the vessel is ready to begin installation (***Start of the installation***).

Installation on site

An installation process highly differs depending on specific component, and software allows to imitate these differences. For instance, for the installation of jacket with suction buckets, it needs to be ensured that a weather window is sufficient to place jacket on the seabed, pump out water from the buckets and perform grouting in one go, without interruptions. If that is not possible, the vessel needs to place the jacket on the seabed and then wait some time for grouting and pumping out water, the jacket will not be fixed for some period of time. Another example is turbine installation. When mounting a turbine, it has to be ensured that, at least, full tower and nacelle can be mounted in one go. Tower without a nacelle has different natural frequencies and hence different structural response to wind loads. These frequencies do not coincide with those of a complete turbine (with nacelle and blades) which could lead to increased fatigue damage. For the above reasons, the software allows user to specify which operations are needed to perform an installation of a certain component, and what are their weather limits (***Sufficient weather window?***).

Moving to the next location

Once installation has been completed, the vessel is ready to move to the next location if there are more components on board (***Departure***). If not - it will return to port (***Components left on board?***). Prior to moving to the next location, again, it needs to be verified that a vessel can safely move, install and return back to port if weather becomes worth (***Sufficient weather window?***). However, for a jack up, it just needs to be assured, that it can safely jack down, move to the next location and jack up (***Jack up vessel?***). Then it can withstand adverse weather conditions. However, floating vessel will be returned to port if it cannot be guaranteed that it will install another full component and have time to return to port until weather becomes worse. Additionally, after each substructure installation it is checked if the minimum number of installed substructures has been achieved (***Min number of substructures installed?***). If so, the turbine installation campaign can commence (***Launch turbine vessel***).

Return to port

When all components on board are installed, the vessel will return to the port and continue until the OWF is completed.

Feeder barges

A slightly different logic is used when simulating an installation employing the so-called *feeder barges*. It is not depicted on the above block-diagram. In this concept, the main installation vessel is positioned offshore. It does not transport components and does not return to the port unless weather is expected to become worse. Instead, barges are used to transport components from an onshore harbour to the main vessel. Albeit employing additional vessels, tools and crew staff, this concept may result in cost savings. These cost savings would come as a consequence of overall reduction in installation time, seeing that a constant supply of components to the site is ensured with barges. In reality, financial viability of such concept has to be validated per project. The simulator developed in this thesis may assist in such validation, as this set-up can be easily modelled.

3.5 Summary

This section started with an introduction of the discrete event simulation by listing its main building blocks. In order to facilitate a better understanding of how DES works, a brief example from the industry was given. The main rationale behind simulating using DES approach is the increase of computational speed. This advantage in the performance is realized by skipping time moments when no event in the system occurs.

Next, the description of the building blocks of the developed simulator was presented. The structure of Simulator block reflects the requirements which were posed to the planning software in section 2.1. The only requirement which cannot be satisfied by Simulator is advising its user on the best values for installation parameters. It will be addressed by the Optimizer. A large variety of parameters are defined directly by a user and are not *"hard-coded"* into the software. Hence, the desired flexibility is achieved. Furthermore, the results of the simulation are first generated in a very detailed form of full timetable and after that processed to reflect the values of the main KPIs. The major blocks include:

- *User inputs* - values and parameters read from the regular spreadsheet file.
- *Weather module* - responsible for generating synthetic weather series needed to run many simulations to accomplish statistical accuracy.
- *DES model* - main piece of code, where all installation logic is determined.
- *Timetable* - a form of an output where all activities are characterized by their starting moment and duration, and are ordered in chronological sequence.
- *Outputs* - numerical results derived from running multiple simulations which are used to analyze certain installation decisions and choices.

Finally, a generic example of typical OWF installation was supported by a flow-diagram depicting main logic elements of a single simulation run.

Chapter 4

Synthetic Weather Generation

In the process of development of the tool it was discovered that the quality weather modelling itself significantly affects the outcomes of simulation / optimization. As it will be presented in this chapter, multiple approaches exist for weather modelling and have been utilized to develop OWF installation planners. Selection of the best approach could become a research topic itself since the quality of generated synthetic weather series directly affects the results produced by the tool. Therefore, this chapter describes prerequisites for synthetic weather modelling specifically for the purpose of OWF installation planning. Requirements for such weather modelling differ a lot from e.g. modelling weather for annual energy production (AEP) or accumulated fatigue assessment. Initially, the Markov Chain model has been selected due to its property of preserving seasonality and time-dependence of the weather on itself.

Results have shown that the pure Markov Chain model was not capable to reproduce the required weather window distribution and resulted in highly oscillating weather series. With an additional adjustment algorithm persistence of weather series was reproduced much closer to that of the originally measured data. As a result, this upgraded model can be used for a weather generating block of the developed tool.

Section 4.1 highlights the need of generating synthetic weather series for OWF installation decision support tool. A brief overview of different approaches exploited for weather modelling in a relevant research literature will be given. An overview is concluded by selecting a suitable approach for this research. Next, section 4.2 elaborates on the selected Markov Chain approach, highlighting its imperfections. Results of applying this original approach are given in 4.3. An additional algorithm which was used to address these imperfections and produce more realistic weather series is presented in section 4.4. Finally, a summary of weather modelling within this thesis is given in 4.5.

4.1 Need for synthetic weather. Overview of weather modelling approaches.

4.1.1 Need and requirements for weather modelling

A significant impact on the offshore wind farm installation logistics is caused by weather circumstances. In order to account for the stochastic weather nature, many simulations need to be run with the same input parameters and various weather conditions, as it was mentioned in the previous chapters. In this way, obtained results will be statistically representative for a given geographical location with given weather conditions. Consequently, it is of a major importance to be able to create realistic weather series preserving the most important statistical properties of the originally-measured data.

In their paper, dedicated to weather modelling, Nfaoui et al. (2003) identified seven statistical parameters of weather series which can be analyzed to assess the quality of modelling algorithm:

- Mean value
- Variance
- Transition probability matrix
- Probability density
- Energy spectral density
- Auto-correlation function
- Persistence probability

The nature of offshore operations imposes specific requirements towards generated data series. Apart from the common requirement to preserve annual wind speed and wave height distribution, ***the most important parameter is persistence probability of a certain weather state***. Here, a weather state is a combination of wind speed and wave height values. In order to perform a certain offshore activity it needs to be ensured that the operational limits (i.e. maximum wind speed and wave height at which a vessel can perform an activity) will not be violated during the approximate activity duration. Therefore, a so-called *weather window* is introduced. Weather window describes a period of time with a certain duration, when wind speed and wave height are remaining below a specified threshold.

Needless to say that the distribution of wind and wave values, their correlation and seasonal character have to be preserved as well. These requirements are common for weather modelling in most disciplines. However, as noted in the previous paragraph, a requirement to obtain similar persistence probability is quite specific for OWF installation planning. Hence, the remainder of this chapter will focus on it.

4.1.2 Weather modelling approaches

In (Ait-Alla et al., 2013) the arithmetic mean was used to forecast the different weather categories for the next 12 months, based on the historical weather data from the last 50 years. Somewhat similar approach was adopted in (Lütjen and Karimi, 2012), where a parameterizable medium-term weather forecast (only next 10 days) was used. Such models seem to be not enough detailed as a large part of information about weather properties is lost during aggregation. In (Barlow et al., 2014b), (Barlow et al., 2014a), (Barlow et al., 2015) and (Barlow et al., 2018) uncertain weather conditions were modelled through a correlated auto-regression model, which enabled multiple data series to be generated from a hindcast weather series. Autoregression predicts future behaviour of the data-set based on the underlying trends as a data-set changes over time. This model requires removal of the seasonality from data and performs weak when several weather components have to be modelled simultaneously (Kerkhove and Vanhoucke, 2016).

A comparison between deterministic and probabilistic approach for the weather representation was given by (Muhabie et al., 2015). The real historical data directly results in a decision on whether to proceed with the next activity or not, while in a probabilistic approach the decision is taken based on the probability of having sufficient weather. Notable, in order to derive accurate results for the project duration, up to 400 iterations may be needed for the mean output values to converge. In general, the two approaches demonstrated good agreement for the average duration of the installation with the correlation coefficient of 0.7.

The most detailed overview of the various approaches for modelling weather was given in (Kerkhove and Vanhoucke, 2016). Authors stressed that a trade-off exists between a complexity of the model and the ease of its implementation. A model needs to be sophisticated only as far as the needed degree of accuracy is achieved. This accuracy can be measured in terms of the correlation between the statistical properties of produced and original data. Furthermore, the authors acknowledge that it is crucial to preserve a seasonal character of the weather and correlation between wind and waves, even though it may result in additional complexity. For the purpose of their research, the authors adopted a combination of Markov transition probability matrices and Weibull distributions. Authors suggested to use ten wind states in order to achieve a needed accuracy of weather representation and not to burden the solvers. A second-order Markov chain was used to model the transition of the wave state taking into account an already generated wind state so that both correlation with wind and autocorrelation are achieved. Finally, in order to model the wave behavior in a more realistic manner, a persistence probability matrix was introduced for wave state transitions to better capture a lag between change in wind and waves. This means that a rapid change in the wind conditions will not automatically result in the same rapid change in waves, while it would be an outcome of the techniques described so far.

Based on the conducted literature review, it was decided that the Markov Chain model would satisfy the main requirements such as preserving seasonality of data, preserving persistence probabilities and ensuring correlated wind and waves series. A deeper explanation of the Markov Chain model is given in the following section.

4.2 Markov Chains

4.2.1 Markov process

A stochastic process is said to fulfill th hypothesis of first order Markov Chain if each subsequent state depends solely on the value of the previous state (Papaefthymiou and Klöckl, 2008). Thus, it is commonly assumed that wind speed and wave height development in time is such a stochastic process. States are equally spaced by a constant time step. A number of states is limited and each state S^i_{wind} represents a range of wind speed values V, such that each wind speed can be directly related to a state. Each state S^i_{wind}

is characterized by its limits V^i_{min} and V^i_{max}. Hence, the space of wind speed states is a discrete representation of the space of wind speed values. Similar explanation holds for wave height h values.

Transition probability matrix P is an $n \times n$ matrix of elements p_{ij}, where n is a number of states and p_{ij} is a probability that the next state will be j, given that the current state is i. Each row of such matrix (4.1) represents current state and each column represents the following state.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \tag{4.1}$$

4.2.2 Transition probability matrix construction

In order to build such matrices for wind speed and wave height, a sufficiently large amount of data has to be processed. 10 years of historic hourly data series were provided by SGRE for several locations. In order to obtain the probability of transitioning from one state to another, the algorithm first converts the wind and wave signal into their respective state-representations. Next, at each hour it is noticed what is a current state S_i and which state follows S_j. 1 is added to a respective cell in the matrix P. Once the full series were processed, each value in the matrix has to be divided by a total number of samples so that the matrix is normalized.

The number of states for constructing such matrix has to be identified depending on the needs of the project and availability of measured data. Among the reviewed papers, values around 10 were adopted (Kerkhove and Vanhoucke, 2016), (Lange et al., 2012), (Papaefthymiou and Klöckl, 2008), (Engelen, 2015). Provided a large amount of data is available, the number of states can be increased. Having compared obtained results with 8, 10 and 12 states, it was decided that 10 states result in the best representation of original properties, such as mean value and probability distribution. The comparison will not be described here. Table 4.1 presents the limits of states for wind speed V and wave height h in m/s and m, respectively :

Table 4.1: State limits for adopted Markov Chain model

State	Wind	Waves	State	Wind	Waves
1	0 - 2.5	0 - 0.5	6	12.5 - 15	2.5 - 3
2	2.5 - 5	0.5 - 1	7	15 - 17.5	3 - 3.5
3	5 - 7.5	1 - 1.5	8	17.5 - 20	3.5 - 4
4	7.5 - 10	1.5 - 2	9	20 - 24	4 - 4.8
5	10 - 12.5	2 - 2.5	10	24 -	4.8 -

In order to account for the seasonality of weather, a separate Markov matrix can be constructed for each month. Thus, 12 matrices for wind and 12 matrices for waves would be built.

The remaining question is then: *how to ensure correlation between wind and waves?* To make it possible, an approach similar to (Kerkhove and Vanhoucke, 2016) was used. A separate wave matrix is constructed for each wind state within a given month. This essentially means that the number of wind matrices remains the same - 12 (1 for each month), while the number of wave matrices is 120 (10 wind states times 12 month). In order to build these matrices, at each hour it is noticed what is a current wind state S^i_{wind}, current wave state S^i_{wave} and which wave state S^j_{wave} follows. 1 is added to a respective cell in a matrix corresponding to current wind state. In the end each matrix is normalized. Wave matrix element p_{ijk} has to be read as follows: a probability of transitioning from the wave state i to the wave state j, given that the current wind state is k.

Below is an example of such a matrix for wave height for a specific location in the North Sea in February (table 4.2). Cells in the upper row and left column specify wave state limits.

Table 4.2: Example of Markov Chain transition probability matrix for wave heights corresponding to wind state #8

	0-0.5	**0.5-1**	**1-1.5**	**1.5-2**	**2-2.5**	**2.5-3**	**3-3.5**	**3.5-4**	**4-4.8**	**5-5.8**
0-0.5	0	0	0	0	0	0	0	0	0	0
0.5-1	0	0	0	0	0	0	0	0	0	0
1-1.5	0	0	0	0	0	0	0	0	0	0
1.5-2	0	0	0	0.272727	0.727273	0	0	0	0	0
2-2.5	0	0	0	0	0.784483	0.215517	0	0	0	0
2.5-3	0	0	0	0	0.04321	0.876543	0.080247	0	0	0
3-3.5	0	0	0	0	0	0.142857	0.803571	0.053571	0	0
3.5-4	0	0	0	0	0	0	0.090909	0.818182	0.090909	0
4-4.8	0	0	0	0	0	0	0	0.095238	0.857143	0.047619
5-5.8	0	0	0	0	0	0	0	0	0.026316	0.973684

It can be seen that due to the fact that the wave state matrix is presented for the 8th wind state (high winds from 17.5 to 20 m/s), the left upper part of the table (corresponding to low wave heights) is filled with zeros. This means that the wave behavior follows wind speeds and correlation is ensured.

4.2.3 Synthetic weather generation

When all matrices are built, artificial weather series of any length can be generated. First, series are generated in a state space. All transition matrices are substituted by their corresponding cumulative matrices.

Each row of a cumulative matrix describes a discrete cumulative distribution of probabilities for the next transition from a current state. In order to obtain the next transition, a random number is generated uniformly between 0 and 1 at each time step and compared with each cell in a row i, corresponding to a current state i. When the generated number falls between elements j and $j+1$, the following state is determined to be j. The procedure is than repeated but now for row j (Papaefthymiou and Klöckl, 2008).

Generation of wave states is performed after each wind state has been generated and the following wave state is determined from the wave matrix corresponding to a current wind state.

When series of a needed length are generated in wind and wave **state** space, the actual values of a signal are determined by uniformly picking a **value** within the borders of already determined states.

4.3 Results of the original Markov Chain model

The focus of this section is to verify performance of Markov Chain model with correlated wind and waves (12 matrices for wind speed and 120 matrices for wave height) using states' limits as defined in table 4.1. Out of the seven statistical parameters mentioned in section 4.1, special attention will be paid to spectral density and persistence probability along with seasonality and correlation of wind and waves. All other parameters were validated with those of the original series, and were all found to be closely matching. To keep the narration brief, only deficiencies of the model are reviewed and resolved.

An example of weather series for wind and waves is shown in figure 4.1 and figure 4.2, respectively, where both original and synthetic data are depicted for one year starting from January 1^{st}. It can be seen that the **seasonal character** of wind and waves (higher in winter and lower in summer) is preserved.

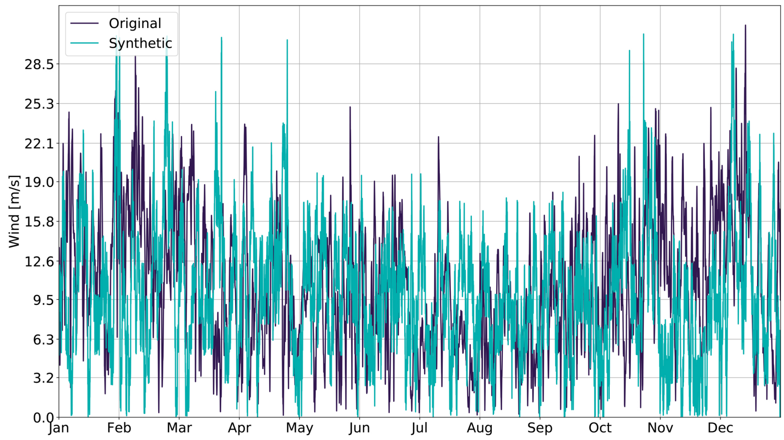


Figure 4.1: Example of wind series generated by Markov Chain approach

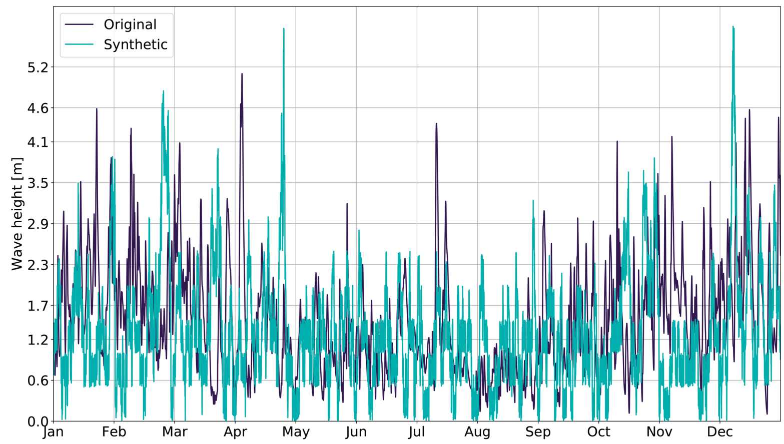
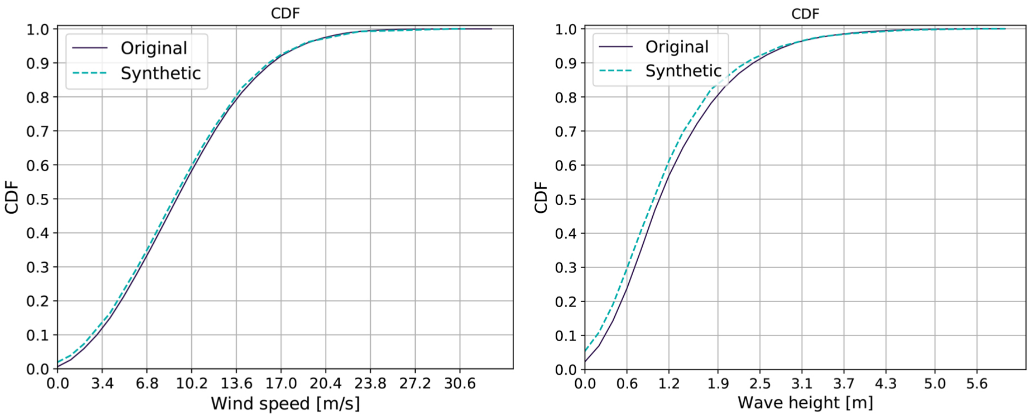


Figure 4.2: Example of wave series generated by Markov Chain approach

Mean wind and wind **variance** of original data were equal to 10.43 and 25.89 m/s, respectively, and those of generated series - 10.21 and 27.37 m/s, respectively. For wave series these values were 1.43 and 0.73 m for original data and 1.33 and 0.77 m for generated. The difference is minor and is negligible for OWF installation planning purposes.

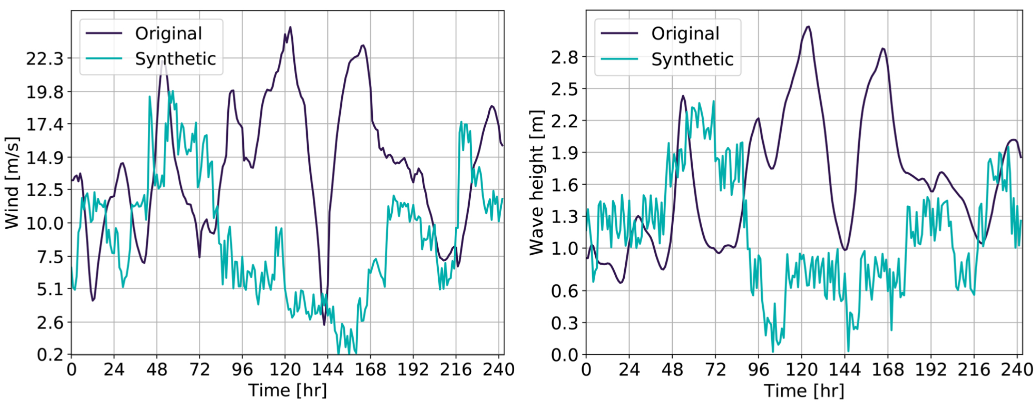
Figure 4.3 contains **cumulative distribution function** (CDF) curves for original and synthetic series. A good agreement between the two is visible.



(a) Wind speed CDF **(b)** Wave height CDF

Figure 4.3: CDF curves for original Markov Chain approach

The following figure 4.4 shows a zoomed version of generated signals. Several observations can be done. First, it is clearly visible that **correlation** is present between wind and wave signals in both original and generated series. This confirms an approach of building a separate wave transition matrix for each wind state.

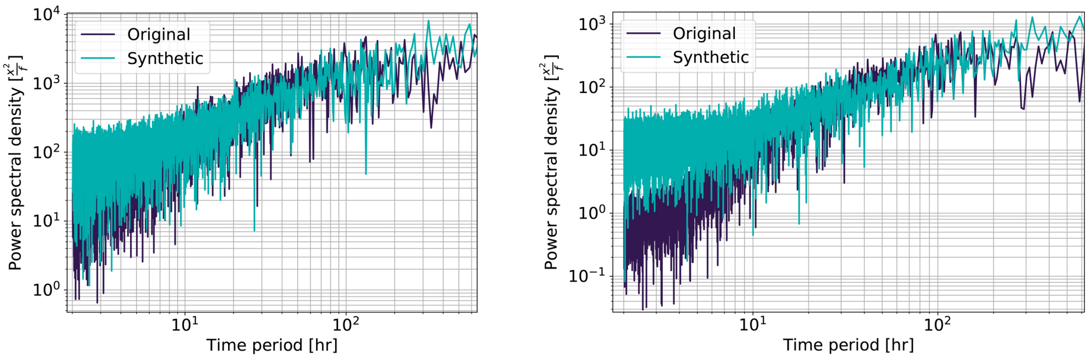


(a) Wind speed **(b)** Wave height

Figure 4.4: Zoomed weather series for original Markov Chain approach

Next, more detailed observations reveal that the generated signal is very intermittent compared to the original one, implying that too many oscillations happen in synthetic series compared with the real ones. This fact is extremely important for OWF installation planning as explained further. In order to investigate this in a detail a spectrum of generated series is analyzed.

Looking at the **power spectral density** (FFT) of the signals in the figure 4.5, it can also be seen that high frequencies, corresponding to short time periods of 2-10 hours, carry much higher power content in the synthetic data. This is in line with the visual impression which arises when looking at the wave signal between 96 and 144 hours in figure 4.4.



(a) Wind speed **(b)** Wave height

Figure 4.5: FFT of synthetic series for original Markov Chain approach

The reason behind this behavior is twofold:

- In reality wind and wave state are determined by more than 1 hour history, while the adopted Markov Chain model only looks one hour behind to determine a current state.

- When generating wind speed and wave height values from already generated state signals, a random number within the state borders is picked uniformly as described in 4.2.3. If several consecutive hours are spend in the same state, an actual value will be changing within the borders of this state each hour (see wave signal between 24 and 48 hours in the figure 4.4 where wave signal is *"stuck"* in state #3 between 1 and 1.5 m).

The combination of these two facts yields very low **persistence** for a generated signal. As described in the first section of this chapter, the main requirement towards synthetic series is to reproduce the same persistence probability as in the original data.

In order to quantitatively estimate this parameter, a pair of CDF curves can be constructed. These curves represent the probability of having a weather window of certain duration with the wind and wave values either violating some predetermined threshold values or not. Thus, X axis represents the length of the weather window, Y axis - probability. Two curves are constructed: one for a probability of being x hours below the threshold (called *Good* on a graph) and one for probability that at least wind or wave signal will remain above the threshold (called *Bad* on a graph).

An example for a threshold of 9 m/s wind speed and 2.5 m wave height, which is typical for performing certain offshore activity, is presented in figure 4.6. It can be seen from the graph that for short durations (low X values) the curves for generated series have much higher probabilities. It implies that they contain more weather windows of short duration compared to a number of weather windows from original series. In particular, for the presented threshold example, the total number of weather windows in 10 years of data was 1507 for original series and 4400 for synthetic. At the same time, the total duration of *"good"* and *"bad"* weather was relatively similar. Yet, a total duration is related to a distribution of wind and wave signal values and their correlation, rather than to their persistence. Thus, total duration cannot be used as a parameter for an estimation of model quality. The results for different thresholds are presented in table 4.3

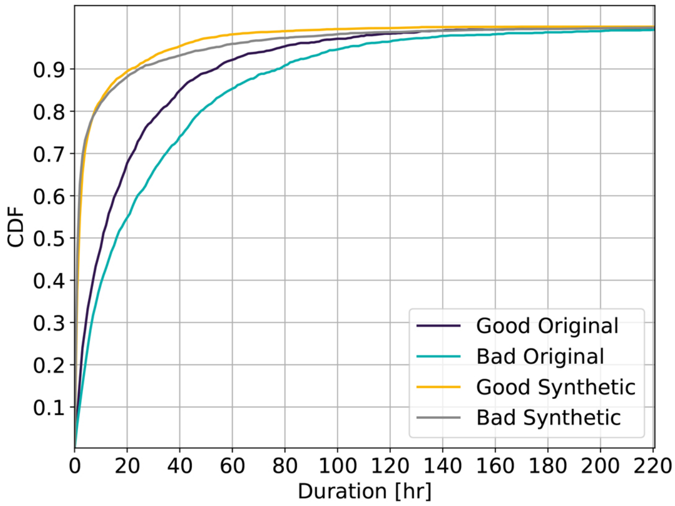


Figure 4.6: *Good* and *Bad* weather window CDF for original Markov Chain approach

Table 4.3: Summary of weather window counting for original Markov Chain approach

Wind threshold [m/s]	6	8	11	14
Wave threshold [m]	0.8	1.2	1.7	2.2
Original - number of windows	746	1170	1365	1200
Original - duration of *"good"*	9704	21082	38775	55119
Original - duration of *"bad"*	69135	57757	40064	23720
Synthetic - number of windows	3240	3371	4034	3282
Synthetic - duration of *"good"*	11335	23198	41837	57577
Synthetic - duration of *"bad"*	67504	55641	37002	21262

It is worth to mention that when these curves are plotted for the weather series in **state** representation (see table 4.1), the curves match almost perfectly. Moreover, when the thresholds for wind and wave are assigned on the border between two states as defined in table 4.1, the difference between a number of windows in the original and synthetic series will not be evident. This confirms a hypothesis that a transition from **state** to **value** space, as described in 4.2.3, is detrimental and results in such a big difference with the original data.

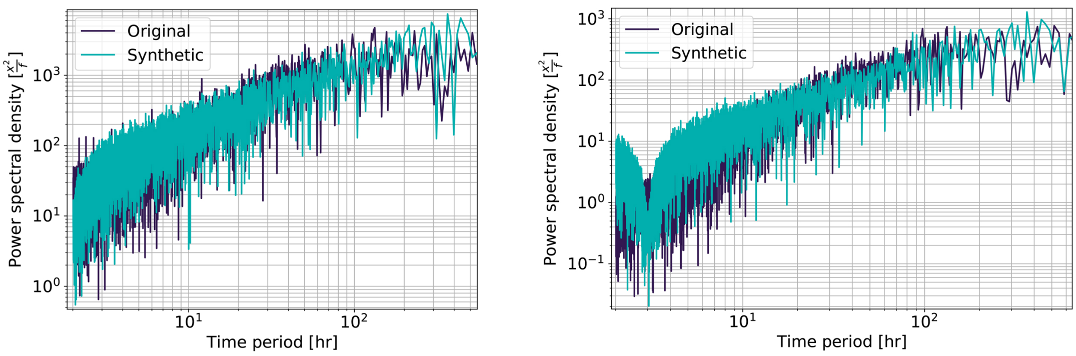
Seeing that a typical duration of offshore operation lies within several hours, it is important that the curves for original and synthetic data match at a low X axis values. This is apparently not the case, and therefore such an approach cannot be used for generating weather series for a decision-support tool for OWF installation planning. This fact was acknowledged in the above-mentioned (Kerkhove and Vanhoucke, 2016) and (Papaefthymiou and Klöckl, 2008) and each group of scholars used their own approach to adopt the original model by introducing persistence probability matrix or automatically adjusting a size of states depending on the specific data set. A more simple and robust approach was, however, used in this thesis as treated in the next section.

4.4 Adjusted Markov chains approach. Comparison of produced weather series.

4.4.1 Step 1: Moving average

The first step is to manually smooth a generated signal by taking its moving average with a small time window, equal to the time periods where many oscillations occur (see FFT graphs in figure 4.5). For this purpose a moving average with a window of 2 hours was used for wind and a moving average with a window of 3 hours (corresponding to a definition of wave state in meteorology) was used for waves.

The resulting FFT is depicted in figure 4.7. A significant improvement, especially for wind signal, can be seen. For waves this is, however, not enough. A large drop that takes place at a period of 3 hours is related to the size of moving average window used for smoothing wave height series.



(a) Wind speed **(b)** Wave height

Figure 4.7: FFT of synthetic series for Markov Chain approach with moving average

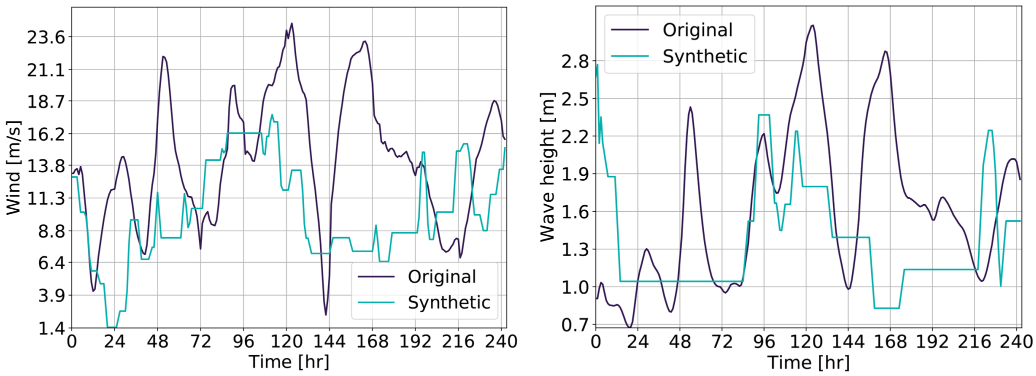
As a result, the following values in table 4.4 are obtained. A notable improvement is achieved compared with the results in table 4.3.

Table 4.4: Summary of weather window counting for Markov Chain approach with moving average

Wind threshold [m/s]	6	8	11	14
Wave threshold [m]	0.8	1.2	1.7	2.2
Original - number of windows	746	1170	1365	1200
Original - duration of *"good"*	9704	21082	38775	55119
Original - duration of *"bad"*	69135	57757	40064	23720
Synthetic - number of windows	1931	1742	2335	1986
Synthetic - duration of *"good"*	12218	21962	39250	55374
Synthetic - duration of *"bad"*	66621	55877	39589	23465

4.4.2 Step 2: Freezing the value in a sequence of several hours in one state

Since this only improves but does not solve the issue of high variability in the generated signal completely, a second adjustment is proposed. As it was identified from the figure 4.4, the roots of the problem lay in the fact that an actual value within the state is picked randomly. When having several consecutive hours in the same state, a high variability is observed, which is unrealistic for both wind and waves. This being confirmed, a second adjustment is to keep the wind speed and wave height values constant in case a generated signal contains a sequence of several consecutive hours in the same state. The value for these subsequent hours is identified based on the randomly generated value for the first hour of a sequence. The rest of the hours in a sequence follow the first hour value. For a better understanding, figure 4.8 depicts zoomed version of generated series. High frequency oscillations from figure 4.4 are eliminated.



(a) Wind speed **(b)** Wave height

Figure 4.8: Zoomed weather series for adjusted Markov Chain approach

Table 4.5 indicates the results obtained employing the second adjustment method. It can be seen that the number of windows for synthetic series is much closer to those of the real weather data. Apart from that, other statistical parameters such as mean value, variance, distribution, seasonal behavior and correlation of wind and waves were preserved and not affected by the introduced adjustments. As an evidence, figure 4.9 presents a scatter plot for adjusted Markov Chain approach.

Table 4.5: Summary of weather window counting for adjusted Markov Chain approach

Wind threshold [m/s]	6	8	11	14
Wave threshold [m]	0.8	1.2	1.7	2.2
Original - number of windows	746	1170	1365	1200
Original - duration of *"good"*	9704	21082	38775	55119
Original - duration of *"bad"*	69135	57757	40064	23720
Synthetic - number of windows	782	1114	1287	1033
Synthetic - duration of *"good"*	11701	23432	42386	57272
Synthetic - duration of *"bad"*	67138	55407	36453	21567

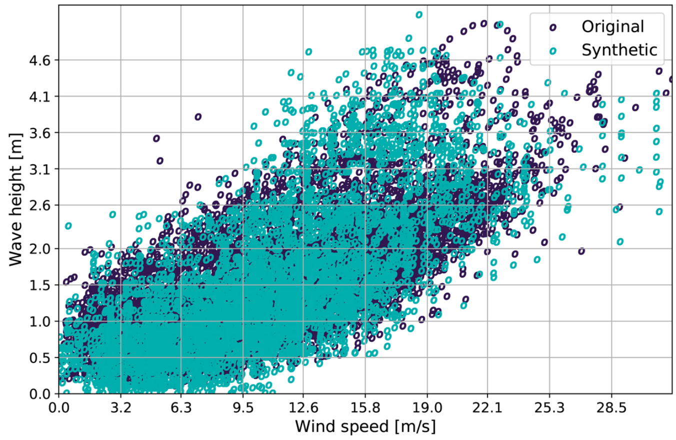


Figure 4.9: Scatter diagram for adjusted Markov Chain approach

Figure 4.10, similar to figure 4.3, presents CDF curves for *"good"* and *"bad"* weather windows for original and synthetic weather (same threshold is used). It can be seen that the difference between related curves barely exceeds 5%, while in the original Markov approach the maximum difference was over 30%.

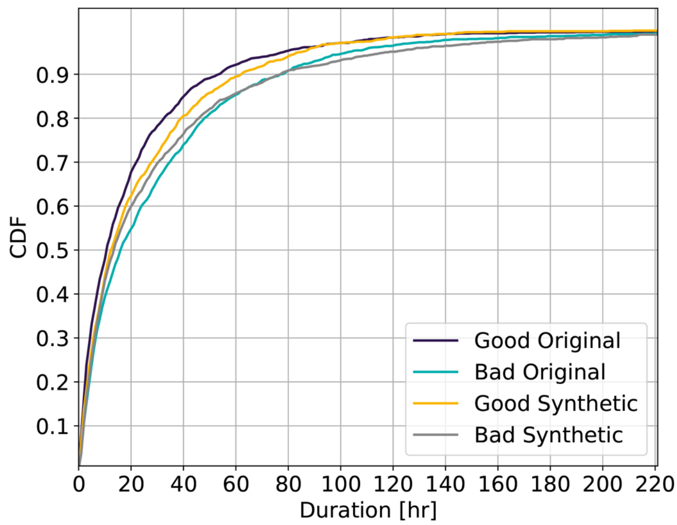


Figure 4.10: *Good* and *Bad* weather window CDF for the adjusted Markov Chain approach

It is important to stress that the proposed adjustment method works only for the purposes of weather modelling within this thesis. The specifics of OWF installation planning are such that weather persistence plays the highest role. The plot of wind and wave series (Fig. 4.8) obtained by employing adjusted Markov Chain approach clearly shows that obtained wind and wave curves hardly resemble real ones. At the same time, the necessity of preserving original persistence of the weather, as a determining parameter for OWF installation planning, completely justifies adopted approach. Author recommends to employ the proposed adjustment algorithms only for the cases where correct statistics of weather windows has to be reproduced, such as in order to investigate OWF installation.

4.5 Summary

In order to develop a tool for OWF installation planning, a synthetic weather generator is needed to be able to run several simulations for the same geographical location. This block is responsible for creating artificial weather series for wind and waves which would carry the same statistical properties as the original weather. The main requirement for the weather generator in OWF installation planning is to preserve weather persistence, correlation of wind and waves and seasonal character of weather.

Based on the conducted literature review, a Markov Chain approach was selected as the most suitable. This approach requires having large series of measured data in order to generate artificial weather series. A key feature of the approach is that it takes into account the weather state in the previous time period in order to generate the following state. This way a generated signal develops based on its historical values. Moreover, the Markov Chain approach allows to correlate wind and waves and reproduce their seasonal behavior.

The only downfall of Markov Chain approach for weather modelling is that it requires large amount of data to construct a model which would take more than one previous value into account. First order Markov Chain only takes into account a single previous value. Another issue is related to the fact that too much randomness is introduced while transferring a generated signal from state space into value space. As a result, generated series are very intermittent and are characterized by greater power spectral density values at high frequencies when compared to the original series.

As a solution, two adjustments were introduced to the original model. A moving average of a generated signal is taken as a first step. This allows to smoothen it and reduce a power spectral density at high frequencies. As a next step, at the stage of transition from state space to value space, the signal is kept constant where several consecutive hours are spent in the same state. In other words, the algorithm notices where a wind or wave signal do not change state for several hours in a row. When transferring from state space into value space, all values are kept the same as a value in the first hour.

As a result of the proposed approach, significant improvement has been achieved. It was proved that synthetic weather generated with the adjusted Markov Chain approach retains the most important properties of the original data. The author points attention to the fact that the proposed adjustment was designed specifically to model weather for OWF installation planning.

Chapter 5

Case Study and Analysis of Simulator Outputs

The goal of this chapter is to validate a performance of Simulator by comparing its results with a real timetable of an installation of an existing OWF. For this purpose, information about the installation of a certain OWF, located in the North Sea, is used. The project data was only available for turbine installation and not for substructures, thus only this stage is addressed. Two types of simulations were run. One using the weather series for the year when the turbines were installed and another consisting of multiple simulations with weather data synthetically generated for this location based on the historical records. All other parameters, including duration of activities, operational limits, etc., were set in such a way to reflect the characteristic values for this OWF.

A range of additional validation studies, including sensitivity analysis and extreme case scenario was performed. As an outcome, it was confirmed that Simulator produces results which are in line with the reality. Minor errors and deviations with the real OWF timeline may occur because only weather uncertainties are incorporated. Apart from that, developed simulation block adequately generates project timelines as it was validated via several extreme case scenarios.

This chapter begins with the description of case study, given in section 5.1. Further, the input values used for the validation are presented in section 5.2. Several descriptive output metrics are compared in section 5.3 in order to validate the results of the simulation with the real timetable for the considered OWF. Some typical outputs of Simulator block will allow the reader to better comprehend its functionality. Sensitivity analysis and extreme case validation are presented in section 5.4. Finally, a summary is given in section 5.5.

5.1 Case study

The following table 5.1 provides information about the OWF installation used in this case study. Due to confidentiality reasons, some of the wind farm data is not given.

Table 5.1: General information about case study OWF

Location	North Sea
Start of installation	2^{nd} of January
Number of turbines	67
Capacity of jack up vessel	4 turbines
Distance to the port of loading	ca. 130 km
Travels	16 x 4 and 1 x 3 turbines

5.2 Simulation parameters

It is first necessary to define the full cycle of installation - from the moment the vessel enters the port for loading, to the moment it arrives back to port from the offshore site to begin the next cycle. This cycle is presented in table 5.2. The inputs for the simulation include operational limits and durations of all sub-activities. Due to the confidentiality reasons it is not possible to provide detailed information on their values. Therefore, only aggregated durations and no operational limits are provided here, still allowing to have a good picture of the process. These aggregated values are obtained as a sum of the duration of underlying sub-activities that are available under appendix D (confidential Appendix A contains operational limits and individual durations).

Table 5.2: Aggregated case study input data for a full cycle

	Activity	**Duration [hr]**
	Navigation	4
Loading	Pre-loading	5
	Load 4 turbines	4x6
	Post-loading	4
	Travel loaded	14
	Jack up (x4)	6
Installation	Install one turbine (x4)	28
	Jack down (x4)	6
	Sail to the next turbine (x3)	3
	Travel back to port	12
Total		**232**

Several assumptions were made:

- It is assumed that travelling to site and travelling back always take constant amount of time, as well as sailing from one turbine to another.
- All durations are fixed based on the average values available from the project records. Stochastic nature of each operations' duration caused by human factor is not accounted for, and would have essential impact (analyzed in 5.4.1).

- All durations were rounded to the nearest integer so that the time resolution is 1 hour due to the resolution of the weather data (see 3.2.3).

- The only disrupting factor is weather conditions. No delays due to the lack of components in port or assets' failure can be experienced.

- Only wind and wave conditions are taken into account. Fog and currents are disregarded in contrast to the real life (explained further).

Furthermore, within the simulation the processes of **Loading** and **Installation** consist of multiple sub-activities. Each of these sub-activities is characterized by its own operational limits and duration, which are incorporated in weather checks within the simulation. Nominal durations that are specified for these activities in table 5.2, are total durations of all sub-activities, assuming that no disruption in their performance took place, thus no additional delays were encountered.

5.3 Comparison of the simulation results and real OWF installation

Three main metrics will be used to compare the results of case study with a real project timeline. In order to validate the accuracy of the developed Simulator the overall project duration, duration of loading in port and duration of installation offshore will be analyzed. For the overall project dynamics, not only the total duration but also the pace of the installation will be qualitatively compared.

First, a simulation with the recorded weather data from the same year when the OWF was built will be performed. This is deemed to be the most straightforward way to validate the accuracy of Simulator. Next, multiple weather realizations will be generated based on 10 years of historical records. For each weather realization a separate simulation will be run. After, aggregated statistical results will be validated against the real project timeline.

5.3.1 Single simulation with real weather records

Table 5.3 summarizes the most indicative results of the simulation.

Table 5.3: Summary of the simulation with real weather records

KPI	Case study OWF	Simulation
Total duration	247 days	230 days
Workability	*Not known*	71%
Average Loading duration	34.21 [hr]	31.71 [hr]
Average Installation duration	40.54 [hr]	41.84 [hr]
Standard deviation of Loading duration	7.4 [hr]	6.3 [hr]
Standard deviation of Installation duration	28.8 [hr]	27.54 [hr]

Note that *workability* is defined as a ratio of the time when certain operation, whether offshore or onshore, was performed to the total rental period of the contracted vessel.

Overall project progress

Figure 5.1 below shows an overall timeline as a result of the simulation with the real weather records.

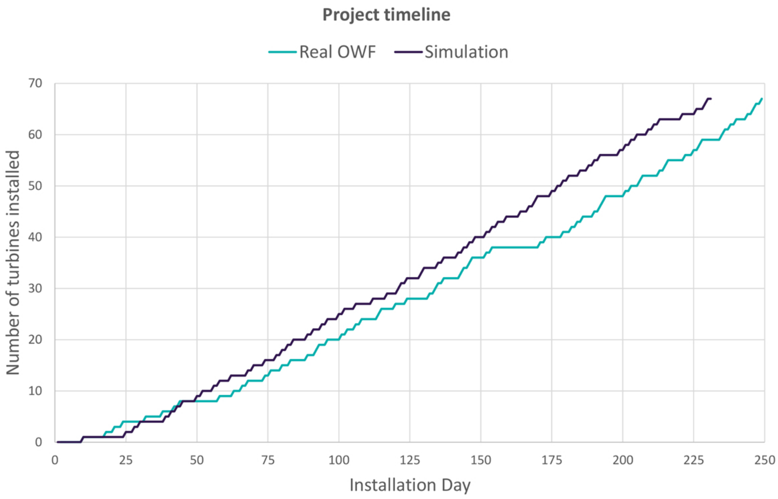


Figure 5.1: Project timeline for the simulation with real weather records

It can be seen that Simulator underestimated the total duration of the project by 17 days (7 %). The real installation was finished on the 4^{th} of September, while simulation completed it by the 19^{th} of August. Generally speaking, two lines are parallel most of the time with several exceptions where significant delays were experienced in the real installation. In other words, the pace of installation was simulated correctly and the end-difference is caused mainly by the big delay experienced in the reference project between days 154 and 170 and smaller delay between days 44 and 57.

Based on the information from the project records, it is known that many long delays were caused by non-weather related factors, e.g. human factor, disruption in the onshore logistics, components breakdown, low tide in port, etc. As a matter of fact, the first large interruption between days 44 and 57 was a result of human-caused delays in port. The biggest disruption between days 154 and 170 happened due to a crane breakdown. Obviously, Simulator does not account for this type of adverse events.

It can be concluded that a total duration produced by Simulator will virtually always be shorter than the one from the real life seeing that there is a range of factors that are not incorporated in the analysis.

Loading in port

Loading and Installation are two main stages for which detailed records are available from the considered OWF. Therefore, it is crucial to understand how their duration can be affected. As an example, one can reasonably assume that each process of loading consists of loading nacelle, blades and tower for each turbine. In case there was adverse weather and loading of any of the above components was delayed, the overall time spent by a vessel in port will grow. Similar holds for the process of offshore installation. If tower installation was delayed because there was not long enough weather window to install both tower and nacelle (as explained in 3.4), the total duration of installation of that particular wind turbine will be higher than the nominal one.

Figure 5.2 below shows the duration of each loading in port.

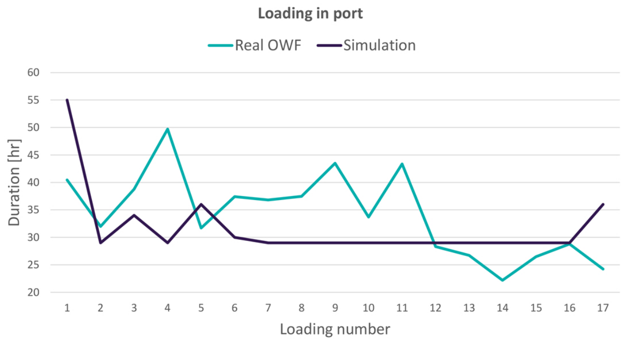


Figure 5.2: Loading duration for the simulation with real weather records

As it was shown in the table 5.3, the average duration of loading was simulated close to the real one, however the difference in the standard deviation reflects the visual impression from comparing two curves. Most loadings within the simulation took the same amount of time, implying that no weather-related delays were experienced. In contrast, the curve representing case study is quite intermittent.

Installation offshore

Figure 5.3 shows the duration of each turbine installation as a result of the simulation with the real weather records.

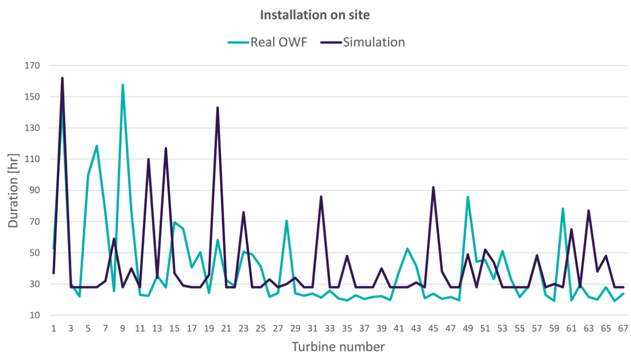


Figure 5.3: Installation duration for the simulation with real weather records

One has to realize that the installation of the same turbine occurred at a different time in the simulation and in the case study. Hence, it would not be correct to judge the quality of the simulation by the difference between duration for the same turbine. What can be compared is a general trend: first turbines took much time to be installed due to more frequent adverse weather conditions in winter, with the improvement towards the summer time. Still, it can often happen that in the real project there was a strong wind for several hours and the vessel had to wait. At the same time, in the simulation vessel would arrive a little bit earlier or later and avoid this time period. In other words, such a simulation is very sensitive and even slight time difference between a simulation and a real case in the beginning of the project may later accumulate and lead to a large discrepancy.

General reflections

Analysing the above presented metrics, it can be said that the outcomes of a simulation are highly sensitive to the exact time moments when a certain activity happens. A delay of a couple of hours in the beginning may yield several days of deviation from the real case by the end of the installation. On the other hand, the pace of the installation is imitated correctly and resembles the one of the real OWF. Simulation should not be regarded as the definitive indication of how the project will evolve, seeing that many delays are just impossible to account for. Such delays include human-caused disruptions, machinery breaks or special weather conditions such as fog. For the above reasoning, a simulation has to be used rather as an indication or a means to compare various installation strategies. Finally, values picked for the durations and operational limits have impact on the produced project timeline. One can expect to obtain more insights by running several simulations with different weather realizations as presented in the next subsection.

5.3.2 Multiple simulations with synthetically generated weather

Convergence of the average results

Figure 5.4 is deemed to show how many simulations with different weather realizations are needed in order to achieve convergence of statistical indicators. In order to plot this graph, mean values of a corresponding indicator were taken from each simulation. Next, it was observed how the average of these means behaves if one keeps increasing the number of simulations. Finally, in order to make it possible to accommodate all curves in one plot, they were normalized with respect to their final value after 50 simulations.

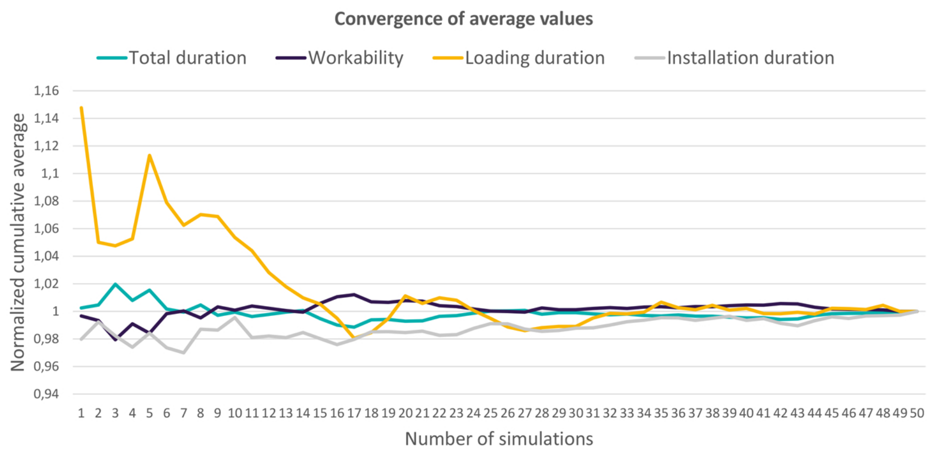


Figure 5.4: Convergence of normalized averages against simulations number

After 30 simulations, the averages vary only within 1% of their nominal value. Hence, the author claims that the developed tool requires around 20 to 30 simulations in order to achieve statistical accuracy for a selected geographic location.

Overall project progress

Figure 5.5 presents a timeline of the project resulting from multiple simulations. All simulated curves fall within the shaded purple area.

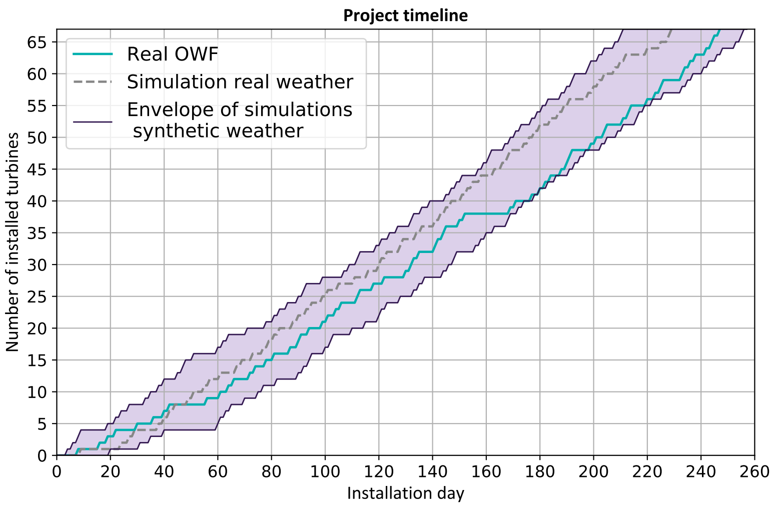


Figure 5.5: Project timeline for simulations with synthetic weather

It can be seen that the majority of simulations produced shorter project duration than 247 days from the real life case. The $p90$ value is 243 days and $p50$ value is 233 days, with only 3 out of 50 simulated timelines having longer duration than 247 days. Such underestimation is absolutely logical as it was explained in the previous subsection 5.3.1. The only disruptive factor incorporated into Simulator is weather conditions, while in the real life multiple factors can cause delays.

A comparison with the real life should be done carefully, since 50 simulations were run but only 1 real installation took place. If the real installation happened in a different year, a different project timeline would arise.

An interesting fact to be observed is how the width of the purple band changes with the period of the year. In the beginning of the installation, January - end of March (Installation days 0 to 80 on the plot), the width varies a lot implying that different weather realizations have large impact on the project progress. Later, from the mid-spring to the mid-summer (Installation days 90 to 180) the width is constant. At this period workability of the vessel is high and weather conditions are good across different simulations. By the end of the installation (Installation days 190 to 250), the envelope broadens, meaning that weather effects are becoming more notable in this period of the year.

Loading in port

Figure 5.6 gives an impression of how the average time spent by the vessel for loading varied. The purple curve shows the average time spent by the vessel in the real project, and the yellow curve is an average over 50 simulations.

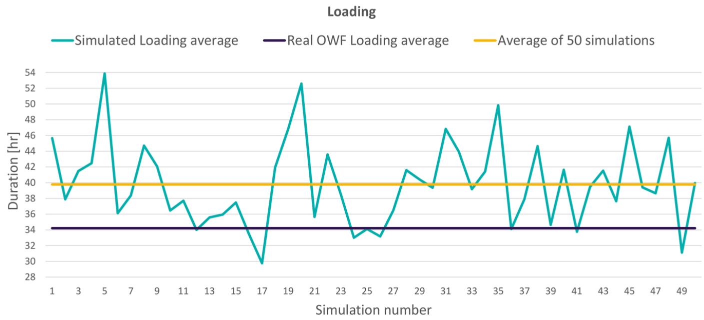


Figure 5.6: Average loading duration for simulations with synthetic weather

Already from the graph in figure 5.4 it can be deducted that the average duration of loading is subject to much higher variations compared to the other three parameters presented. The aquamarine curve in figure 5.6 is characterized by high variability between different simulation runs. The average value produced by Simulator is overestimated by 6 hours. Such results clearly deviate from what could be anticipated. As discussed before, normally one should expect to have underestimated durations for the simulation results. There are three reasons for why it is not the case:

1. **Comparing multiple simulations with a single real case:** as it was mentioned in the analysis of figure 5.5, it is not completely fair to compare results of multiple simulations with a timeline of just one real installation. The project flow recorded during the reference OWF installation was also subject to specific weather conditions in the year of installation. Would it take place one year before or after, different values for the total duration, loading and installation would be recorded. Therefore, results of multiple simulations cannot be used as a prediction of this particular real life project, even though their inputs are based on it. The results have to be interpreted as an indication of what **average values** would look like if one was to construct an OWF in this location with the given inputs.

2. **Simulation inputs:** in contrast to the average durations of each offshore operation available from the real project, for the loading in port only bulk numbers were recorded. No detailed information is available about how much time loading of each component took. Consequently, the input values used for the simulations were based on the common estimates used within SGRE. In fact, they produced good result when running the simulation with the real weather data (see table 5.3). When analyzed deeper, it was discovered that within the project high variability in loading was observed as well, with the minimum time spent in port - 22 hours and maximum - 50 hours. However, the simulation was set up in such a way that even in the best case 29 hours are spent in port, thus 7 more than in the case project. This is why the average produced duration is also shifted by $\sim$ 6 hours.

3. **Low number of loadings:** only 17 loadings take place both in the case study project and simulations. With such a low number of samples a single large delay in port has more impact on the average duration per simulation than an ideal loading without a delay. Some of the worst loading times within the simulations amount to $\sim$ 180 hours, implying that high winds persisted for several days. One might have assumed that the weather in port was not generated correctly. Nevertheless, the opposite was confirmed by checking the CDF curves as in the figure 4.10 for a weather window

with the limits corresponding to blade loading wind limits. When analysing installation times, 67 records are available per simulation, which when averaged produce a much better estimate. Therefore, it is believed that if more loadings happened during the project, a smaller difference would be obtained.

This being said, more attention has to be pointed towards the input values when a low number of loading cycles is performed. At the same time, the inconsistency in simulating loading duration does not jeopardize the functionality of comparing different installation strategies. Albeit A simulated number may deviate from the real projects, a relation between results for two different installation strategies is not affected by this, provided a sufficient number of simulations is done.

Installation offshore

Figure 5.7 has a similar intention to show how the average installation duration varied in the simulations. Again, the yellow line is an average of the aquamarine, while the purple line is an average from case study project.

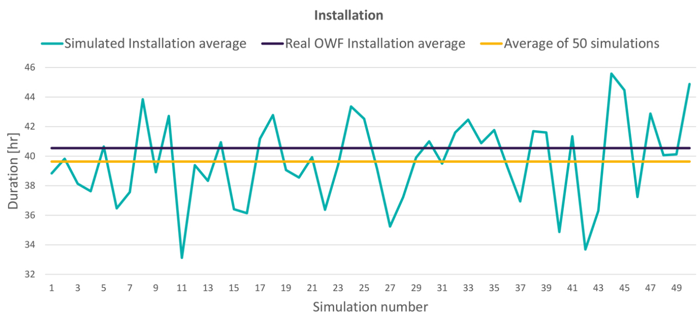


Figure 5.7: Average installation duration for simulations with synthetic weather

A good match between two averages can be observed. This implies that the results of the simulation are accurate for the average installation duration. Average durations and operational limits were available for each separate activity within an installation cycle. Hence, it was not difficult to assign correct values for the simulation inputs.

General reflections

It was observed that when running multiple simulations with synthetically generated weather, obtained results are close to the real values. As in the simulation with the real weather, an overall duration of the project is normally underestimated. For the reference project $p90$ value of 50 simulations gave a good evaluation of the expected duration.

However, one should realise that the comparison of **multiple** simulations was done with a **single** real project, implying that its own progress was also subject to the weather conditions in that specific year. Hence, the values of KPIs obtained from many simulations cannot be treated as an exact indication of what would occur in real life.

Analyzing one level deeper, it can be said that the average duration of loading and installation have different different level of deviation with real values. While the generated installation times are very accurate, the loading figures deviate from the real one. In order to improve the estimate for loading, a better information would be needed with regard to the operational limits and durations for each separate activity within a loading cycle.

Moreover, the results of loading suffer from a low number of samples, i.e. the average loading duration may be highly affected by a single extreme value.

After all, it was concluded that in order to achieve a convergence of average desired KPIs, around 20 to 30 simulations are sufficient. The outcome of 50 simulations showed that an installation of the same OWF, in the same geographic location, with the same parameters may vary up to 45 days (∼17 % of the total duration) depending on the weather conditions. This conclusion once again underlines the necessity of developing this kind of decision support tools.

5.4 Sensitivity analysis and Extreme case

This section provides additional insights into an operation of Simulator. A sensitivity study is done in 5.4.1 to investigate how results of a single simulation can be affected by input parameters. Furthermore, a synthetic extreme case is reviewed in 5.4.2 to confirm the validity of simulation results. Finally, an additional validation for a change in a single input parameter is performed in 5.4.3.

5.4.1 Sensitivity analysis

The objective of this subsection is to familiarize the reader with an impact of alterations in input values. Five installation factors will be considered to study how project outcomes may change depending on a variation in inputs. Each of the following inputs will be varied separately, with the case study parameters from 5.1 and 5.2 as the base scenario:

- Starting day of campaign
- Vessel capacity
- Onshore operations (duration and operational limits)
- Travel (duration and operational limits)
- Offshore operations (duration and operational limits)

The following KPIs will be investigated for each varied parameter:

- Total duration of the OWF installation
- Average duration of loading in port
- Average duration of installation of one turbine offshore
- Workability of the vessel
- Delays during loading - ***Loading delays***
- Delays due to waiting for good weather to depart from the port - ***Departure delays***
- Delays due to waiting for good weather to jack up - ***Jack-up delays***
- Delays due to waiting for good weather to install tower and nacelle in one go - ***Tower+nacelle delays***
- Other delays encountered during installation of one turbine - ***Installation delays***
- Delays due to waiting for good weather to move to the next turbine - ***Move delays***

Starting day of campaign
The starting day of campaign is varied with a step of one month starting from the 2^{nd} of January (base scenario).

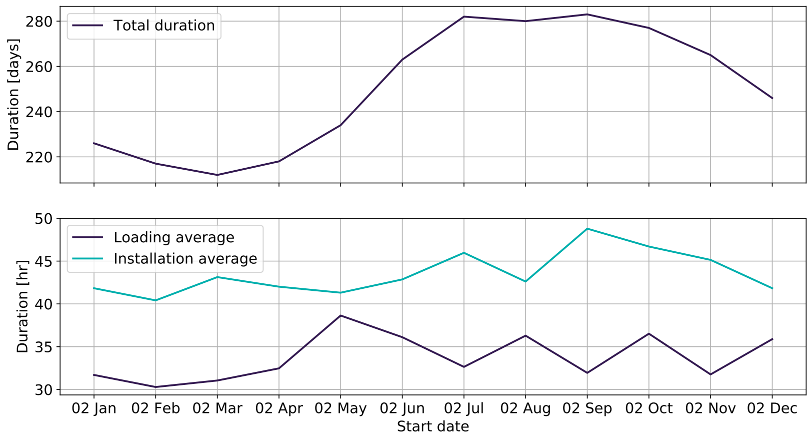


Figure 5.8: Start day sensitivity. Durations.

It can be seen from figure 5.8 that the starting day has a very prominent effect on the overall duration, as it can be expected. The more winter months are in a project time span, the more weather delays are encountered, and consequently the longer is the duration. Nevertheless, looking at an average duration of loading and installation, it is hard to make a straightforward conclusion. Durations of these activities experience accumulated effect of the entire installation procedure. As it was shown in the figures 5.7 and 5.6, the average duration of these activities can vary highly depending on a specific weather window sequence. Despite a general pattern in meteo conditions can be similar, a several hours shift between two projects in the beginning will largely affect the following progress.

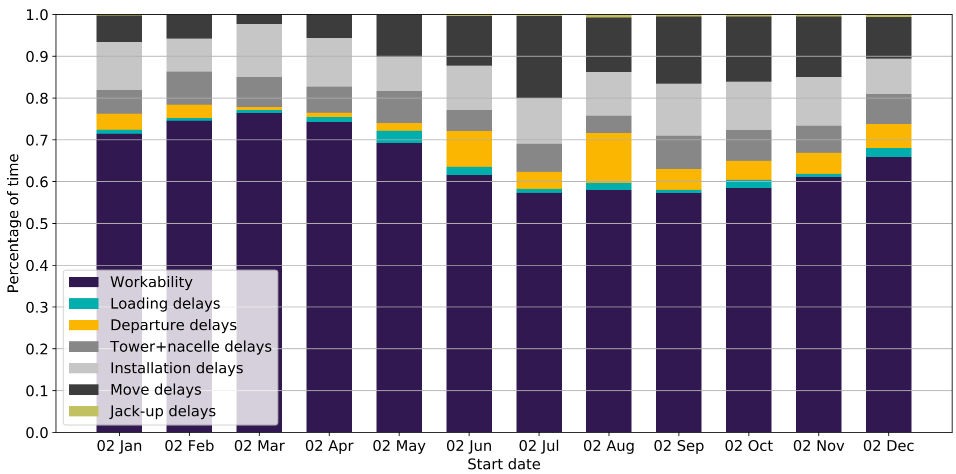


Figure 5.9: Start day sensitivity. Workability and delays.

Figure 5.9 shows that the highest workability can be achieved when starting in early spring, with the lowest to be achieved when starting by the end of summer. ***Loading*** and ***Installation*** delays seem to not depend on the starting day, supporting the above discussion.

At the start dates in July, August and September the weather may affect more either weather checks in port or onsite but the effect on the workability remains the same. The proportion of ***Departure delays*** and ***Move delays*** has exactly opposite behavior compared to the workability. Consequently, the decrease in the workability happens due to the growth in these types of delays. Their operations actually do not have stringent operational limits. However, due to a start in the summer, the major part of campaign runs during autumn and winter months. The effect would be even more prominent for floating vessels, which prior to taking decision of departing from port or moving to the next turbine ensure that at least one turbine can be installed after that. Essentially, these delays would be more affected by the operational limits of installation activities than by their own.

Vessel capacity

The vessel capacity was varied between 2 to 9 turbines on board.

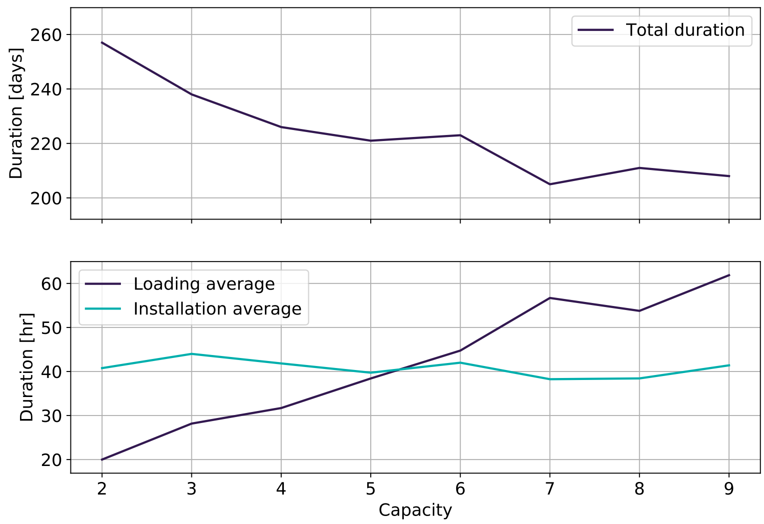


Figure 5.10: Vessel capacity sensitivity. Durations.

Quite remarkable is the shape of the total duration curve in the figure 5.10, as it almost perfectly follows inverse function. It can be noticed that after approximately 7 turbines, the gain in the total duration is not that big and hence further increase in the vessel capacity will not lead to a significant reduction in a campaign duration.

Two remarkable tendencies arise from the figure 5.11. Firstly, the less the vessel capacity is - the more loading cycles (loading cycle means complete vessel loading) it will have to perform, and thus the more delays will be experienced in port prior to departure. Secondly, the more turbines the vessel has on board, the higher number of offshore trips from one turbine to another will have to be done, which results in the increased proportion of ***Move delays*** but less ***Departure delays***.

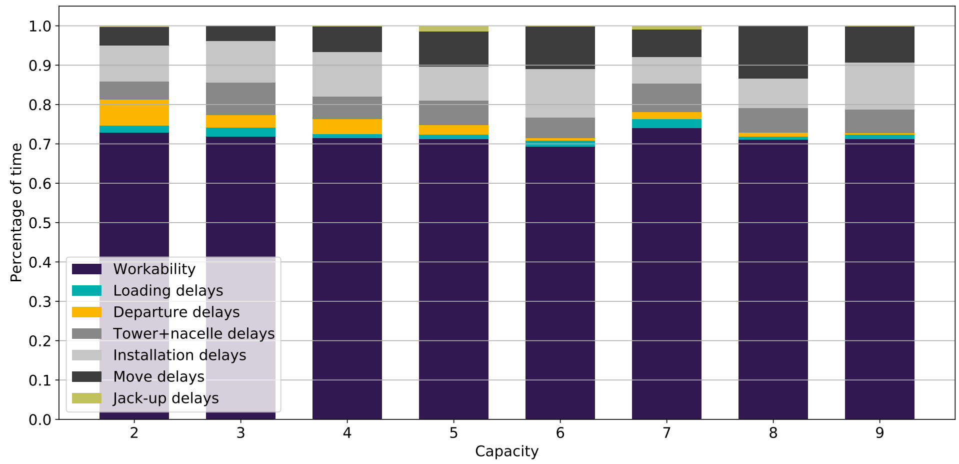
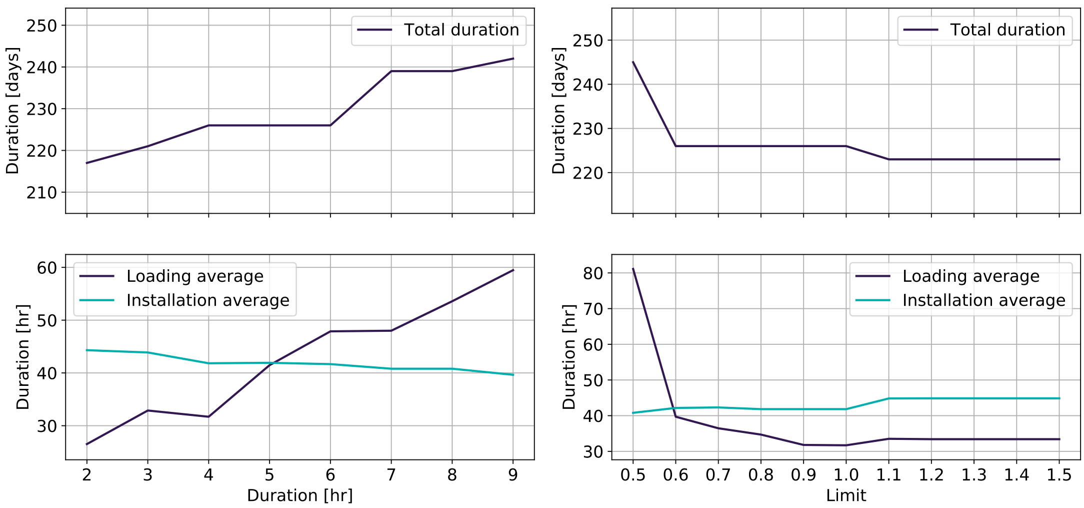


Figure 5.11: Vessel capacity sensitivity. Workability and delays.

Onshore operations

In order to investigate a dependency of the KPIs on the duration of onshore operations (figure 5.12a), the total duration of loading blades and nacelle was varied between 2 and 9 hours (4 hours in the base case). Next, duration was fixed but operational limits of loading blades (base case value is 12 m/s) were multiplied by a factor varying between 0.5 and 1.5 (figure 5.12b).



(a) Activity duration sensitivity. Durations.

(b) Operational limits sensitivity. Durations.

Figure 5.12: Loading sensitivity.

Results of this sensitivity analysis follow anticipated trends, as shown in figure 5.12. The longer the duration of the loading activities is, the higher will be the total duration and average loading duration. More insights can be obtained from the variation of the operational limits, where it is shown that at some point ($\sim$ 0.9 of nominal value) all KPIs stabilize and do not change their value with the following increase in the operational limit. This implies that current practice of loading blades at around 12 m/s (varies per manufacturer and project, but is a good estimate) is already close to an optimum.

Based on the figure 5.13, it can be noted that the increase in the duration of loading activities does not significantly affect an overall workability. The proportion of loading delays moderately increases with the growth of duration.

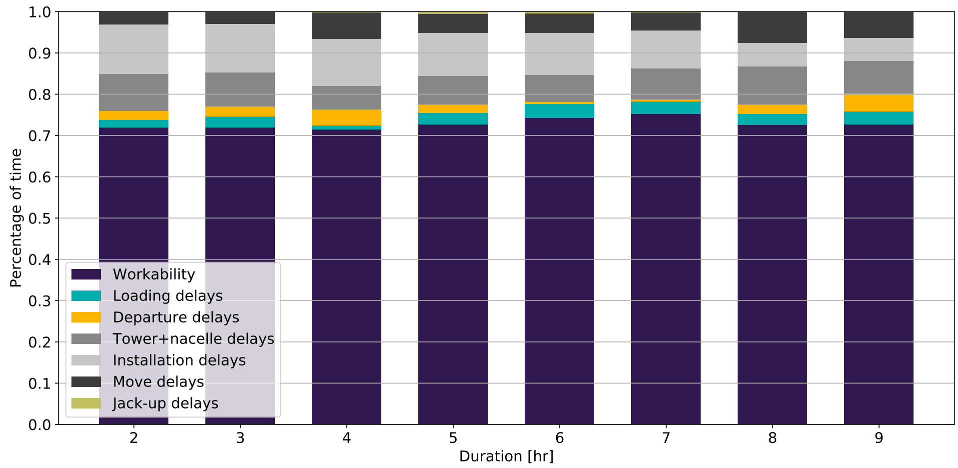


Figure 5.13: Loading sensitivity. Activity duration sensitivity. Workability and delays.

A somewhat more notable effects causes the variation in the operational limits. The percentage of loading delays increases towards lower operational limits as it is shown in figure 5.14.

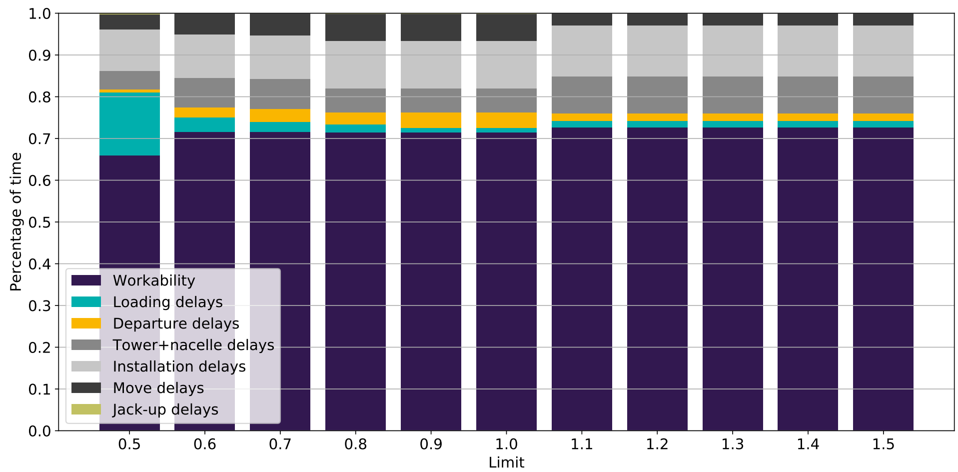
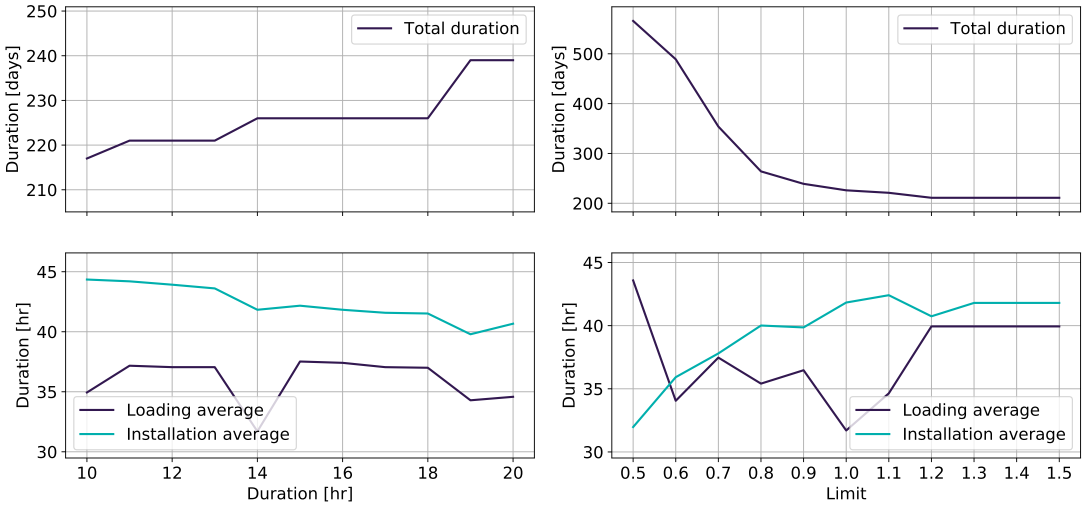


Figure 5.14: Loading sensitivity. Operational limits sensitivity. Workability and delays.

Travel

First, the duration of travel from port to site was varied between 10 and 20 hours (12 in the base case). The duration of travelling from offshore site back to port was always kept 2 hours less, since the vessel is empty and can sail faster. The time it takes to move between the turbines was always kept constant. Second, the operational limits (14 m/s wind speed and 2.5 m wave height) were multiplied by a factor varied between 0.5 and 1.5.

Figure 5.15a shows that the total duration is proportional to the travel duration, as it could be expected, while the variations in the average loading and installation time are not related to this and probably occur due to accumulated effects of the weather on the entire campaign.



(a) Activity duration sensitivity. Durations.

(b) Operational limits sensitivity. Durations.

Figure 5.15: Travel sensitivity.

Similarly to the the outcomes of loading limits sensitivity analysis, the total duration for travel limits sensitivity resembles the shape of the inverse function (figure 5.15b). Again, variations in the duration of loading are quite chaotic and do not show any correlation with the operational limits for loading as expected.

On the other hand, the average duration of installation is lower when operational limits for travel are low. The explanation behind this not obvious phenomenon is the following: 1. Project start is in the middle of winter, thus high values of the average installation duration are due to the installation delays experienced in winter, in the beginning of the project; 2. With low operational limits for travel, the first several trips are significantly postponed in time with respect to the base case, thus happen somewhere in spring; 3. The weather in spring is less harsh so long installation delays are avoided as a trade-off for longer in-port delays. Figure 5.17 below, confirms this rationale (see ***Departure delays***, ***Tower+nacelle delays*** and ***Installation delays***).

From figure 5.16 it can be derived that the duration of travel has no impact on the workability. The only KPI, which experiences a moderate effect of increased travel time, is a proportion of delays prior to departure in port that grew slightly.

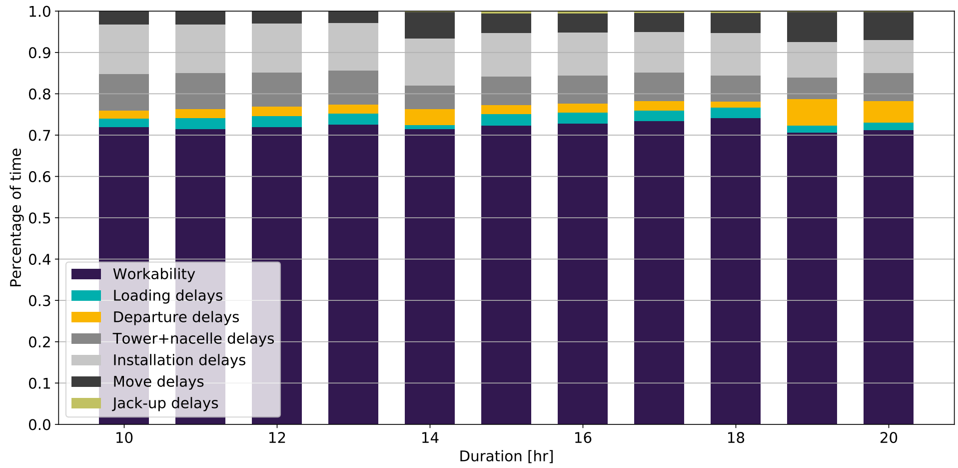


Figure 5.16: Travel sensitivity. Activity duration sensitivity. Workability and delays.

As already discussed above, figure 5.17 gives some insights into how an average duration of the installation may depend on the operational limits of travelling to the site and back.

When operational limits are low, the vast proportion of delays is due to waiting for a better weather when moving between turbines, which takes 14 hours in total, including time for jacking up and down. Seeing that travelling is a relatively long operation which cannot be interrupted, imposing low operational limits for it has a very large effect on the workability, as can be seen on the chart. Finally, increasing the operational limits for travelling higher than 1.2 of the nominal value, does not affect the considered KPIs.

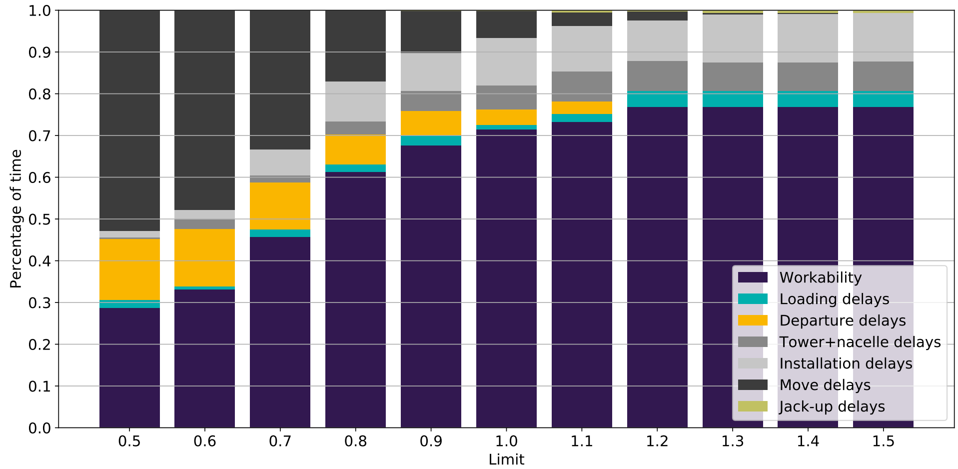
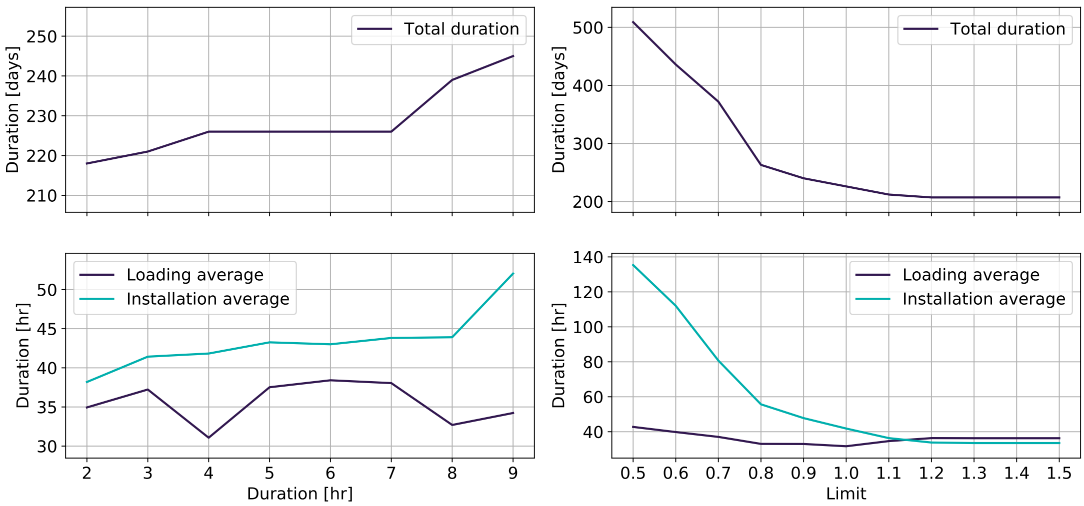


Figure 5.17: Travel sensitivity. Operational limits sensitivity. Workability and delays.

Offshore operations

In order to analyze the impact of offshore operations, their duration and limits were varied. A total time of tower and nacelle installation was modified in a range between 2 and 9 hours, with a nominal total value of 4 hours (figure 5.18a). For the operational limits study, the nominal weather limits for tower, nacelle and blade installation were multiplied by a factor varied between 0.5 and 1.5 (figure 5.18b).



(a) Activity duration sensitivity. Durations. **(b)** Operational limits sensitivity. Durations.

Figure 5.18: Installation sensitivity.

The results of duration variation are in line with what can be anticipated: longer installation duration leads to the overall longer project. Operational limits have already familiar consequence on the behavior of the total duration and average installation duration. Similarly to the variation in the operational limits for travelling and loading, their variation yields almost perfect inversely-proportional dependency curves, which stabilize after ~ 1.1 of the nominal value.

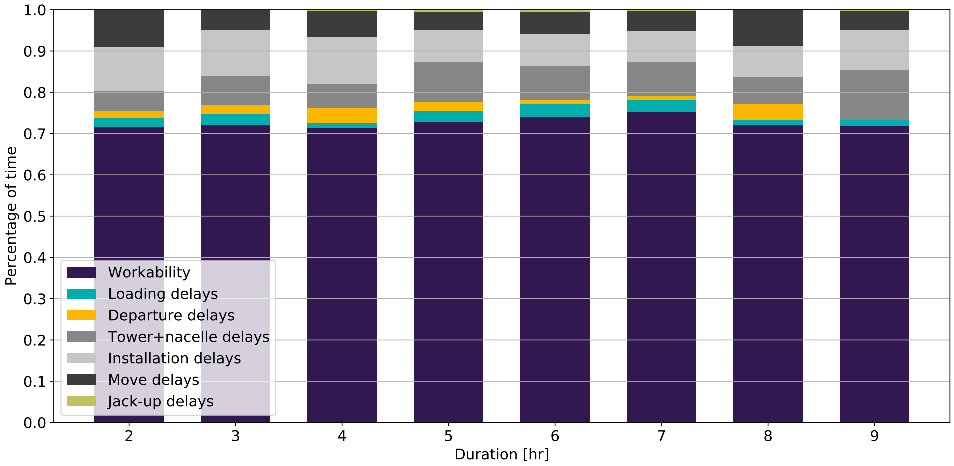


Figure 5.19: Installation sensitivity. Activity duration sensitivity. Workability and delays.

Figure 5.19 shows the impact of the variation in the total duration of tower and nacelle installation. In contrast to the expected growth in the delays associated to the need of installing turbine tower and nacelle without an interruption, the actual increase is very moderate. The reason is probably due to the relatively low proportion of the time these activities take with respect to the overall project duration.

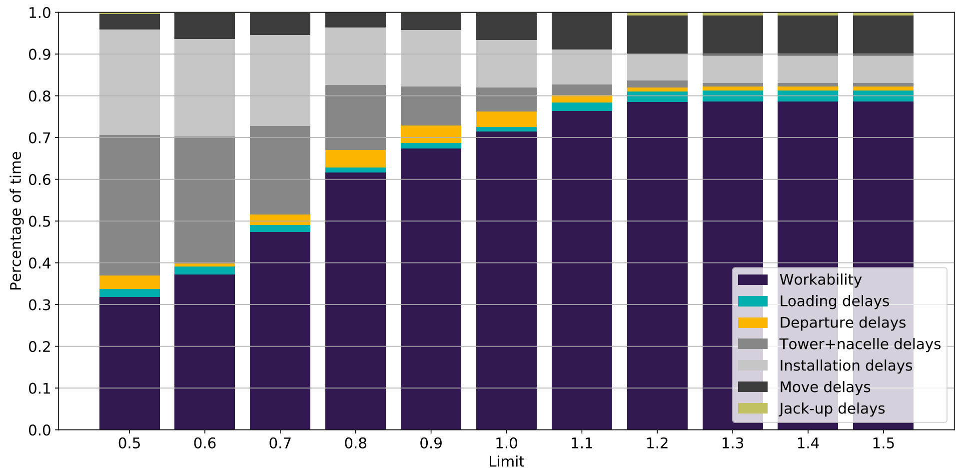


Figure 5.20: Installation sensitivity. Operational limits sensitivity. Workability and delays.

Changing the operational limits for three main operations related to the turbine installation offshore has large prominent implications on the proportion of related delays, as it is presented in figure 5.20. Having very low operational limits leads to exceptionally low workability of $\sim$ 30% at 0.5, and of $\sim$60% at 0.8 of the nominal value, respectively. This effect, however, diminishes after 1.1 of the nominal value. Nevertheless, the growth between workability at 1 and 1.2 of the nominal value is almost 10%, which suggests that it might be beneficial to work on increasing currently used operational limits.

General reflections

With the sensitivity study performed on a standard wind turbine installation campaign, it is concluded that Simulator adequately imitates the process of OWF installation. The obtained results are aligned with what can be expected based on the common knowledge and existing installation practices. Sensitivity study has shown that a change in inputs has logical implications. The validity of the developed block is therefore confirmed.

Based on the sensitivity study, it can be concluded that the main driving factors for the selected KPIs are starting day and operational limits of travelling and offshore operations.

The author suggests that using the average duration of loading and offshore installation should be accompanied by an analysis of their variance and $p90$ value which give information about the distribution of the sample rather than their magnitude. The average durations are a subject to the accumulated overall project progress and often do not reflect the project dynamics, being affected by irregular variations in weather conditions. Furthermore, when analysing operational limits, it was discovered that only some of them might be extended to gain in project time and reduce delays. Generally speaking, the current state of operational limits is close to optimal values seeing that their increase did not reduce a proportion related delays significantly.

Variations in the duration of activities do not have any significant effect on the delays proportion and workability. Only overall duration is affected.

5.4.2 Extreme case

The purpose of the extreme case analysis is to perform an additional validation whether a timetable produced by Simulator is correct. In order to do so, a synthetic case with all operational limits set to sufficiently large numbers is run. Essentially, it completely removes any impact of weather conditions. Consequently, one shall expect that absolutely no delays will be encountered in the process of an OWF installation. Hence, a total duration of the project must be equal to a sum of the durations of all performed operations.

Since all operational limits are set to a sufficiently high number, the impact of weather is eliminated and there is no need to run multiple simulations.

Set up

Below, table 5.4 presents a description of the considered wind farm.

Table 5.4: Extreme case study OWF

Start of installation	t=0
Type of installation	4-leg jackets with piles and turbines
Number of turbines	60
Capacity of floating piles vessel	20 piles
Capacity of floating jacket vessel	4 jackets
Capacity of jack-up turbine vessel	4 turbines

The following operations in table 5.5 have to be performed at each trip for the respecting component (all durations are given in hours):

Table 5.5: Durations of operations for the extreme case

Piles		**Jackets**		**Turbines**	
Activity	Duration	Activity	Duration	Activity	Duration
Navigation	2	Navigation	2	Navigation	2
Backload	5	Backload	5	Backload	5
Load 5 sets of piles	4	Load 4 jackets	8	Load nacelle	2 (x4)
Post-loading	2	Post-loading	2	Load 3 blades	2 (x4)
Travel	10	Travel	10	Load tower	2 (x4)
Position	2	Position	2	Post-loading	2
Pre-load	3	Pre-load	3	Travel	10
Install template	4	Install jacket	6	Jack-up	5
Install 4 piles	12	Grouting	8	Pre-load	6
Prepare hammer	2	Install TP	4	Install tower	3
Hammer 4 piles	4	Departure	1	Install nacelle	4
Remove template	4	Move next	1	Install blades	12
Departure	1	Travel back	8	Complete RNA	4
Move next	1			Energization	5
Travel back	8			Jack down	6
				Move next	1
				Travel back	8
Total per trip	**195**		**134**		**234**
Total campaign	**2340**		**2010**		**3510**

Several remarks can be made to facilitate better understanding of the described installation routines. Each vessel carries several components on board. Thus, activities, which are performed while loading or offshore, have to be multiplied by the capacity of a vessel if different is not specified. After each travel back, loading is performed again. Post-loading is only performed once, when a complete set of components is loaded. After each installed set of piles, each jacket or each turbine, vessel moves to the next turbine as long as it still has components on board.

A single trip for the piles vessel, carrying 20 piles on board, thus enough for 5 jackets, will take 23 hours till the first arrival to the site (first separated section in **Piles** column), five times 32 hours for a complete installation of piles for one jacket (five times second section), four times 1 hour for moving to the next jacket and 8 hours to travel back which results in 195 hours per trip. Accounting that the considered OWF consists of 60 turbines, the total duration of piling should take $60/5 \cdot 195 = 2340$ hours (97.5 days).

For a jacket vessel, time needed to load and arrive on site is 27 (first section in **Jackets** column); then 24 hours are spent per jacket (four times second section); three times 1 hour for moving to the next jacket and 8 hours to travel back. In total this results into 134 hours per trip. For the whole wind farm it amounts to $60/4 \cdot 134 = 2010$ hours for jackets installation (finished by 181.25 day).

Finally, for a turbine vessel it takes 43 hours to load and arrive on site (first section in **Turbines** column); 45 hours to jack up, install complete turbine, commission it and jack down (four times second column); three times 1 hour to move the next turbine and 8 hours to return back. This sums up to 234 hours per trip. Seeing that there are $60/4 = 15$ trips, it should take 3510 hours in total for turbines campaign (finished by 327.5 day).

This being calculated, a total duration of the project should be 7860 hours if substructure installation begins right after piles vessel has returned from its last trip to the port, and turbine vessel starts once jackets vessel has returned from its last trip to the port.

Results

The graph below, in figure 5.21, shows an overall project progress. When comparing the lines it can be noticed that the line representing substructures installation has the steepest slope, piles line - medium, and turbines line - the lowest. This corresponds to a relative order of the above-derived total durations.

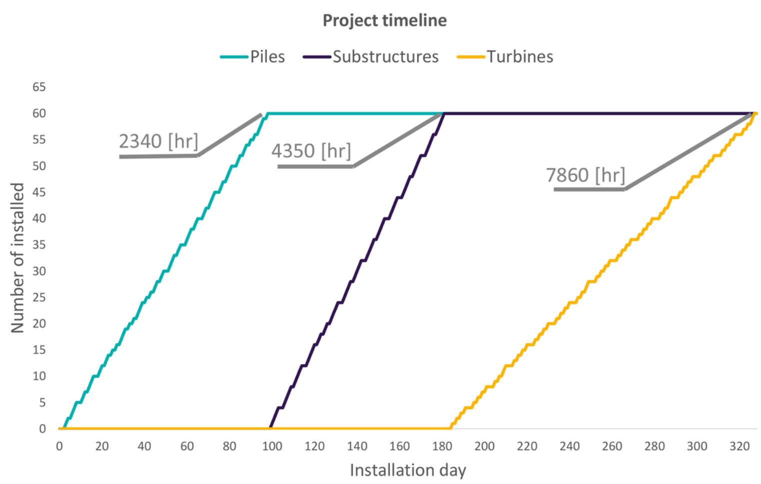


Figure 5.21: Extreme case project timeline

As a result of the simulation, all numbers have been confirmed to be as expected. Each loading and each offshore installation cycle took exactly the same amount of time, implying that no delays have occurred.

5.4.3 Single parameter change

The purpose of this validation is to investigate whether a change of a single operational limit in one simulation will affect the overall timeline as expected.

Set up
Table 5.6 below describes the OWF to be analysed for this purpose.

Table 5.6: "Single parameter change" case study OWF

Location	North Sea
Start of substructure campign	1^{st} of March
Start of turbine campaign	16^{th} of April
Type of installation	monopiles and turbines
Number of turbines	50
Capacity of substructure vessel	4 monopiles
Capacity of turbine vessel	4 turbines
Type of substructure vessel	floating vessel
Type of turbine vessel	jack-up vessel
Distance to site	70 km

For this case study, installation of monopile substructures will be addressed. The following table 5.7 represents typical simplified sequence of offshore operations needed to install monopile.

Table 5.7: Offshore sequence for "Single parameter change" case study OWF

Operation	Duration [hr]	Wind limit [m/s]	Wave limit [m]
Prepare crane	1	N/A	N/A
Upend and install monopile	3	12	1.2
Remove crane	1	15	2.5
Prepare hammer	1	N/A	N/A
Hammer monopile	3	13	2
Remove crane	1	15	2.5

With this set up, a total duration of installation is 5193 hours (216 days), out of which installation of monopiles took 1996 hours and installation of turbines took 4113 hours. Workability of substructure vessel is 55%. This implies that a lot of delays were experienced due to adverse weather conditions. The reason behind such a low number is that the type of substructure vessel is floating vessel. As mentioned in 3.4, it has to be confirmed when leaving the port that during the following 72 hours wind speed and wave height will not exceed transit limits of the vessel. If that is confirmed, the next step to leave the port is to ensure that at least one monopile can be installed. These are strong requirements, and it was further discovered that approximately 40% of delays were encountered due to waiting for a good weather in port or actually on site, with each of these factors having almost equal impact.

In order to investigate, how a change of a single parameter will affect the vessel's workability, the operational limits will be altered for offshore monopile installation. Namely, "Upend and install monopile" wind and wave limits will be increased to 100 m/s and 100 m, thus essentially eliminated. This way one can expect the maximum increase in workability to be $(100 - 55) \cdot 0.4 = 18\%$ resulting in maximum workability limit of 73%. (Operational limit of this operation affects weather checks both in port and offshore, thus 40% of delays could be eliminated at maximum).

However, it would only be possible if the entire offshore sequence consisted only of this operation and there was no need to check prior to leaving the port that there will be 72 hours of good weather to remain in transit mode and assuring that one monopile can be installed. In fact, there are more activities involved offshore and two criteria to leave the port remain present. Hence, the resulting workability has to be lower than 73% but, obviously, higher than previous 55%.

Results

Figure 5.22 represents the project timeline before and after the change. It can be seen that a change of operational limits of just one activity in the offshore sequence of installation can have a tangible impact on the overall project flow, and can lead to significant cost savings due to shorter rental periods and increased utilization of employed resources.

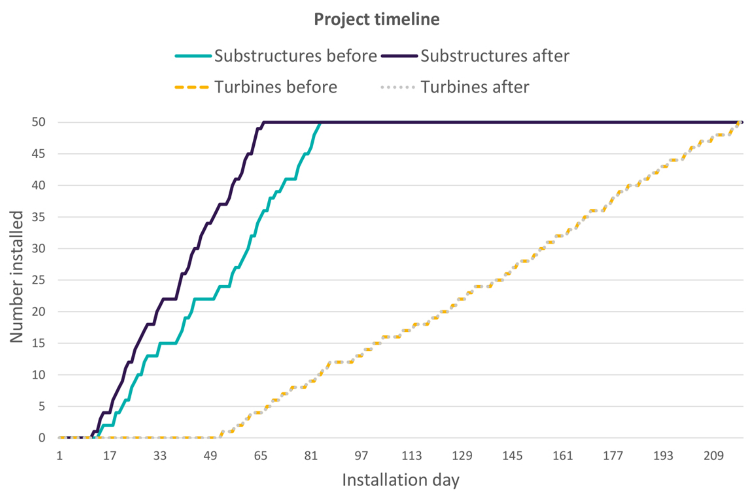


Figure 5.22: Project timeline for a "Single parameter change" study

Table 5.8 summarizes obtained results. The total duration of substructure installation was reduced to 1547 hours from 1996, yielding workability of 68 %. Waiting in port accounted for 18 % of delays, a relatively small reduction due to the fact that there are more than one operation in the offshore sequence that has to be performed for at least one monopile, and there is a requirement to ensure 72 hours of weather not exceeding transit limits. At the same time, waiting on site accounted for 6 % of delays, a significant improvement, compared to an earlier 20 %.

Table 5.8: Results of a single parameter change

KPI	Before	After
Workability of substructure vessel	55 %	68 %
Percentage of delays in port	20 %	18 %
Percentage of delays on site	20 %	6 %
Duration of substructure campaign	1996 [hr]	1547 [hr]

5.5 Summary

This chapter addresses several objectives. A case study was performed in order to validate the accuracy of the developed Simulator against the real life project. It turned out that Simulator adequately imitates the real process of an OWF installation. A slight underestimation of the total project duration may happen due to the fact that only weather-related delays are incorporated into the developed tool. In reality, human caused errors, machinery breaks, etc. may have further impact on the project progress.

Further, multiple simulations were run with synthetically generated weather in order to see how the averaged results of multiple simulations compare to the real life project. It turned out that such a method produces good results that are aligned with the timeline of case study OWF. After around 30 simulations with different weather realizations, obtained $p90$ values of KPIs corresponded to the actual metrics in the real project.

Additionally, a sensitivity analysis was carried. Consequently, another hypothesis arising from the case study was confirmed - average durations of installation offshore and loading in port are highly dependent on the specific weather realizations. These metrics experience an accumulated affect of the entire project. Often a minor change in weather in the beginning of the project yields absolutely different values for these KPIs. The author suggests that analysing average values together with variances and $p90$ values gives better insights, including extra information about a sensitivity to the weather realization.
Another outcome of the sensitivity study is that the start day of a campaign and operational limits have high impact on the project KPIs, while duration of activities and vessel capacity changes result in a more moderate variations in the outcomes.

As an additional check, an extreme case was created and run to verify that Simulator performs correctly. All operational limits were eliminated, implying that zero delays have to be encountered, thus total duration of the project has to be the sum of all underlying activities' duration. This turned out to be as expected.

Finally, another validation was done, where a guess can be made about the expected outcomes prior to simulating. Then, the expectations should be confirmed with the obtained results. Operational limits were changed for a single offshore activity. It was expected that this may only improve workability up to a certain point, based on the available knowledge of how this offshore operation contributes to the total delays. The change in the workability occurred within the anticipated limits. Thus, the validation can be regarded as successful.

Chapter 6

Optimization of OWF Installation

An overview of the optimization problem analyzed in this thesis is given in this chapter. Based on the nature of the problem and properties of the objective function, several requirements were set towards an optimization algorithm. Consequently, a gradient-free optimization tool is designed with optimization parameters including starting day of the entire installation, starting days of substructure and turbine campaigns, onshore port pre-assembly rate and size of onshore harbour. Two different approaches, namely Particle Swarm Optimization (stochastic direct solving algorithm) and model-based deterministic solver based on the Radial Basis Function model (surrogate model-based optimization), are compared. As a result, both algorithms succeeded in finding a good solution. Model-based algorithm, however, required less function evaluations to do so, and therefore might be preferred over metaheuristics when the objective function is computationally expensive.

Section 6.1 lays the foundations of the chapter by analysing prerequisites of the optimization within this thesis. Additionally, a range of requirements towards the Optimizer is imposed. Next, a formulation of the optimization problem is given in section 6.2. Section 6.3 describes two algorithms to be used for optimizing and a rationale behind the choice. Results of applying the optimization are presented and compared for different algorithms in section 6.4. Finally, a summary of this chapter is given in section 6.5.

6.1 Optimization objectives and prerequisites

As discussed in section 2.7 and later reviewed in section 3.3, one of the important requirements towards the tool developed in this thesis is that it has to carry a functionality of advising decision-makers on the optimal values of installation parameters. This feature will be covered by the Optimizer block (see figure 6.1).

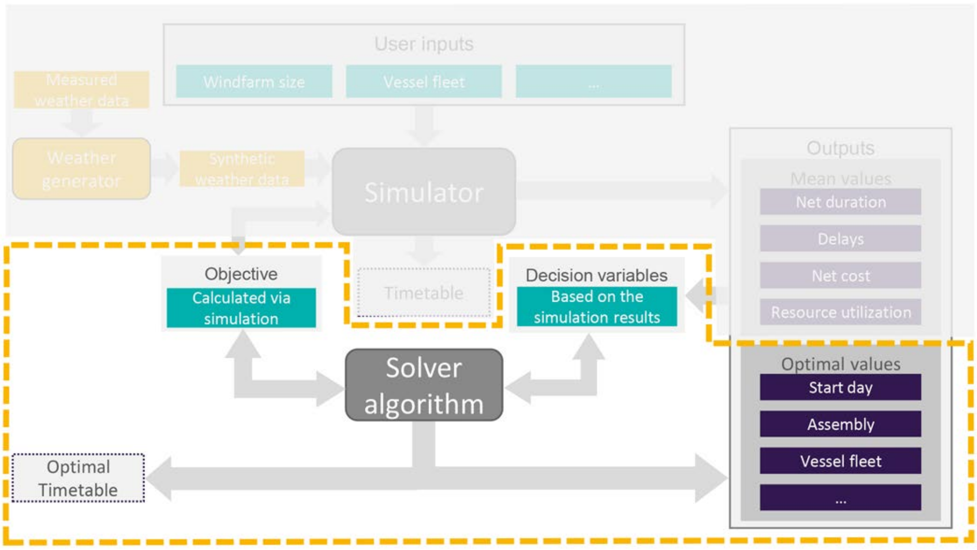


Figure 6.1: Optimizer diagram

In order to formulate an optimization problem, initially it has to be determined what is the nature of an *objective function*, which parameters should be included as *decision variables*, and whether there are *search space limitations*. These three aspects define a typical optimization problem as given in equation 6.1:

$$\begin{gathered} min(f(x_i)) \text{ , s.t.} \\ x_i^{min} \leq x_i \leq x_i^{max} \\ g_j(x_i) \leq 0 \\ h_k(x_i) = 0 \end{gathered} \tag{6.1}$$

In the equation above, $f(x_i)$ is an objective function of n variables to be minimized; x_i - decision variable to be optimized; $x_i^{min} \leq x_i \leq x_i^{max}$ - upper and lower boundaries for variables; $g_j(x_i) \leq 0$ - inequality constraints; $h_k(x_i) = 0$ - equality constraints.

As it was explained in 2.6, the objective function used in this thesis is represented by the earlier-developed Simulator block. As a result of several simulations, an average total cost of an installation can be obtained, hence it will be used as a return value of the objective function.

6.1.1 Solver algorithm prerequisites

Next step is to select a solver algorithm. This selection is based on several considerations related to the mathematical foundations of optimization and properties of the decision variables. Here only the basics are outlined - the actual selected algorithms are discussed further in 6.3.

Type of optimization problem

First of all, it is important to know the type of an optimization problem. The most general classification of optimization problems describes whether the problem is *continuous* or *combinatorial*, i.e. whether all decision variables are continuous real numbers or some of them are discrete. As it often happens, many real-life parameters cannot be represented by continuous numbers. It will be shown later that this is the case with the optimization in this project. Hence, the problem treated in this thesis is combinatorial.

What is more, due to the fact that the objective function simulates the process of an OWF installation, subject to stochastic weather conditions, it is impossible to assign an analytic function describing the relationship between the input variables and values of this function. As a result, it is also impossible to speak of the gradients of such function. Optimization of the functions of such type is called *gradient-free*, *derivative-free* or *black-box* optimization (Rios and Sahinidis, 2012). Technically, black-box optimization may be addressed with gradient-based methods via the algorithms working with finite differences method. Nevertheless, in this project these categories will be used interchangeably.

Computational cost of the objective function evaluation

Another aspect is the computational cost of a single evaluation of the objective function. It was explained that a value of objective function will be obtained as a result of DES simulation within Simulator block. As it was mentioned in the chapters 2 and 5, the result of a **single** simulation run is not enough representative for a specific OWF location. The weather conditions change from year to year and are unknown in advance. Therefore, an average value of several simulations with different weather conditions has to be used for planning to account for possible weather uncertainties.

With this requirement, what is called a value of the objective function is actually an average of multiple simulations runs. The number of required simulation runs for this average to stabilize is usually in the range of $\sim$20, as it was shown in chapter 5, figure 5.4. Therefore, each time Optimizer is evaluating the objective function, 20 simulations will be run. Seeing that a single simulation run takes about $\sim$60 seconds (in the previous chapters different time was mentioned, this depends on the specific installation scenario which is simulated), a total time to obtain a value of the objective function for a given set of parameters is around 20 minutes.

This is deemed to be an expensive computation. In order to keep the optimization time within reasonable limits, it is vital to select a solver algorithm which is capable of finding a good solution with a minimum number of function evaluations. In literature a property called Number of Function Evaluations (NFE) is introduced to compare a performance of different solving algorithms for the same problem.

Nature of decision variables

Potential variables may be classified as *"real"*, *"integer"* or *"categorical"* (exact difference explained further in 6.1.2). Most of the existing optimization algorithms are designed to deal with continuous real variables. They explicitly use the concept of derivative in order to find an optimum. These algorithms are often extended to be applied to integer variables as well. Only a small range of algorithms can also handle categorical variables since the concept of derivative cannot be defined for such variables.

Properties of the objective function
Finally, based on the number of simulation with various parameters it is known that the objective function is likely to have several extremums. Selected solver algorithm has to be able to find a **global** optimum point.

6.1.2 Optimization variables selection

This subsection defines which input parameters will be used as the decision variables for optimization. Their possible values and impact on the objective function are presented further in 6.2. Once the decision variables are selected, it is easy to determine the resulting search space based on the common knowledge of installation practices. The selection is mainly based on the offshore wind industry needs and some mathematical fundamentals.

OWF installation considerations
Some parameters of an OWF installation do not have a wide range of possible options, thus their optimal value can be easily obtained by brute-force trial. This is, for instance, related to pre-assembly strategy where only few options are available (tower pre-assembly, rotor pre-assembly, etc.). Having discussed the range of options with SGRE, it was decided that a decision-maker would prefer to have control over such conceptual parameters. Moreover, when changing, for example, assembly strategy for a rotor, different offshore installation and onshore loading sequences have to be employed. Simulator, which will serve as the objective function, is designed in such a way that these sequences and their operational limitations have to be defined by a user manually via Excel file. For an optimizer to try out different pre-assembly options it would be necessary to know what are the implications of each pre-assembly for the rest of the installation process. This would require creation of a database containing all possible options for the combination of pre-assembly, onshore loading sequence, offshore installation sequence and operational limits.

The same explanation holds for the number of employed vessels. Although it has a large influence on the overall cost, usually the number of considered options is limited and can be easily tested and compared manually via several simulations.

Further, since one of the starting points for this thesis was cost reduction in OWF installation, only the most cost-intensive aspects need to be addressed. It is well-known, that mainly these are vessel and harbour rental costs (Barlow et al., 2015), (Hansen and Siljan, 2017), (Kumar, 2017). This being said, the focus has to be on the parameters which in one or another way affect the length of rental period and total duration of the installation campaign.

Mathematical nature of involved parameters
As mentioned in 6.1.1, variables can be classified as *"real"*, *"integer"* or *"categorical"*. The first type describes variables whose values belong to the range of real numbers. Such variables are continuous and can take any value between minus and plus infinity. The second type describes installation parameters whose possible values form a uniform integer numerical set, such as starting day or vessel capacity. The third, in contrast, are parameters related to specific installation concepts such as vessel fleet or pre-assembly option, neither having a numerical analogue in a real life. For a better understanding, a reader may regard such parameters as those which cannot be sorted based on their value. Categorical values up to a certain extent coincide with those installation parameters which do not have many possible options, however largely affect an entire installation set up. Numerical values,

have less impact on the installation logic and precedence requirements but have a large variety of possible values.

In order to avoid extra function evaluations, it is worth to reduce a search space by limiting the set of decision variables to the ones which are hard to test manually. The set of variables must definitely include those where multiple possible values are available.

Selected variables

Below, table 6.1 presents a list of possible decision variables and their characteristics according to the above-described metrics. In this table, green color represents favorable property for the variable to be selected, red - hindering, and yellow - neutral.

Table 6.1: Optimization variables selection

Variable	**Difficulty to optimize manually**	**Range of options**	**Impact on CAPEX**	**Class**
Starting day	Hard	Large	Large	Integer
Starting day of jackets campaign	Hard	Large	Large	Integer
Starting day of turbine campaign	Hard	Large	Large	Integer
Tower pre-assembly	Low	Small	Medium	Categorical
Rotor pre-assembly	Low	Small	Medium	Categorical
Vessels combination	Medium	Medium	Large	Categorical
Capacity of vessels	Medium	Small	Medium	Integer
Number of vessels	Low	Small	Large	Integer
Onshore harbour size	Hard	Medium	Large	Integer
Onshore pre-assembly rate	Hard	Medium	Large	Integer

It can be seen that in order to keep Optimizer in line with the described rationality and mathematical tractability considerations, a limited set of optimization parameters is selected.

The following five variables will be optimized:

- Starting day of the entire campaign
- Starting day of jackets campaign (with respect to piling)
- Starting day of turbine campaign (with respect to jackets installation)
- Onshore harbour size
- Onshore pre-assembly rate

6.2 Optimization problem

This section gives a formulation of the problem to be analyzed. Further, possible ways to reduce the size of the search space of the problem are explained. Some insights in how the outcomes of an optimization can be affected by search space boundaries are provided.

6.2.1 Optimization problem formulation

The previous section presented a general structure of any optimization problem and identified preconditions for formulating an optimization problem in this thesis. Present section elaborates about the objective function, selected variables and search space for the optimization within the current project.

Decision variables

$$\begin{aligned}
&d_c - \text{starting day of an entire campaign counting from the first day of a year}\\
&d_s - \text{starting day of substructure installation counting from } d_c\\
&d_t - \text{starting day of turbine installation counting from } d_s\\
&a - \text{harbor storage capacity for jackets}\\
&r - \text{number of jackets pre-assembly lines}
\end{aligned} \tag{6.2}$$

Note that the variable d_s is only used when separate vessels are hired for pre-piling and jacket installation. Otherwise, it is not present in the objective function. In case several vessels are hired for the same installation campaign (piles, jackets or turbines), they all start on the same day. Variable a indicates how many fully assembled jackets can be stored in an onshore harbor, thus how much area is needed.

Furthermore it is worth to explain the meaning of the variable r. As discussed in chapter 2, it is beneficial to have some insight into the onshore assembly process. One of the concepts being considered by SGRE is to introduce jackets assembly in the harbour, whereby tubes - parts of jacket lattice structure, and suction buckets arrive to the harbour separately and are assembled right before being transported offshore and installed. Such a concept allows to avoid bottlenecks in the onshore supply chain by eliminating the need to transport large jackets from a manufacturer to the port. In harbour jackets will be assembled from separate parts and upended so that they are ready to be loaded to a vessel waiting for them at the quay side. Appendix C contains schematic representation of the described process.
Note that in the considered problem assembly always starts on the same day as jacket installation campaign. Initially the jackets storage is full, so that substructure vessel does not have to wait for the first jackets to be assembled and can begin right away.

Objective function

$$\begin{gathered}
min(C(d_c, d_s, d_t, a, r)), \text{where}\\
C(d_c, d_s, d_t, a, r) = \sum_i^N C^i_{vessel} + C_{labor} + C_{harbor} + C_{pre-assembly}
\end{gathered} \tag{6.3}$$

It was decided that the objective will be to minimize the total capital expenditure encountered during an entire installation campaign. It will be obtained as an average of the total costs of multiple simulations with different weather realizations. The total cost from a single simulation comprises:

- Cost of N vessels C_{vessel}, a function of d_c, d_s, d_t and duration of corresponding installation campaign (piles, jackets or turbines).
- Cost of labor C_{labor}, a function of total project duration, thus d_c, d_s and d_t. Additional labor costs are experienced due to assembly. These depend on the duration and number of assembly lines r and are only incurred until assembly is finished.

- Cost of harbor rent C_{harbor}, a function of total project duration and required harbor space, thus d_c, d_s, d_t, and a. Costs for storage area for piles, jackets and turbines are incurred during entire duration of corresponding campaign. Additional costs are incurred for the area needed for assembly lines, function of r, but only until assembly is completely finished.

- Cost of equipment and machinery needed for pre-assembly $C_{pre-assembly}$, a function of r. Only incurred until assembly is finished. Fixed costs of assembly and cost per day scale linearly with the number of assembly lines.

The author would like to draw reader's attention to the complexity of interrelation between optimization variables and their effect on the incurred total cost. For instance, when r takes higher value, i.e. more is payed for assembly lines, the jackets are assembled faster, so there are less delays incurred by vessels in port. This reduces cost of vessel rent and makes overall project duration shorter. As soon as the assembly is finished, there are no more costs incurred for it (including no more expenditures for assembly labor, machinery, cranes and area).
Next, interesting effects may happen while varying variable a. The more space is available in port to store jackets - the less delays are experienced by vessels and the sooner the assembly can finish. In case when there is no more space for assembled jackets, assembly process is paused, but the project developer keeps paying for assembly-related assets.

To summarize:

- By saving on the storage area, higher expenditures may be incurred for a prolonged assembly.

- By spending more on assembly, lower costs are incurred for vessels since delays due to the shortage of jackets in port are avoided. Assembly as such is finished faster.

- By having more assembly lines than needed, the urge to put the assembly on pause will arise more frequently, resulting in lower efficiency (percentage of time, when a line is producing).

- Higher assembly rate requires more storage space if one wants to minimize time when the assembly is paused.

- By paying more for a storage area, vessel delays in port are avoided.

- By starting turbine campaign sooner after substructure campaign commencement, the overall project duration is reduced but more delays are experienced in case not enough jackets are installed to put a turbine on.

The above discussion proves that the selected parameters are hard to optimize manually, so the usage of Optimizer is justified for them. Section B.1 of Appendix B contains full formulation of the objective function and facilitates better comprehension of the above discussion.

Search space limits

The starting days are limited by the number of days in a year. The capacity of harbor storage will be varied between 5 and 15, which reflects the size of a typical existing harbor storage site. From the internal discussion it was decided to limit the maximum number of assembly lines r to 3, seeing that each line takes a big part of the area of an entire harbour. It is assumed that the supply of components from an onshore manufacturer to the harbor is automatically adjusted according to an assembly rate in harbor. There are always available components to assemble a jacket and their accumulation is not modelled.

Below are the upper and lower limits on the decision variables.

$$\begin{aligned} 0 \leq d_c \leq 365 \\ 0 \leq d_s \leq 365 \\ 0 \leq d_t \leq 365 \\ 5 \leq a \leq 15 \\ 1 \leq r \leq 3 \end{aligned} \tag{6.4}$$

If extra knowledge is available about the installation, the decision space can be further reduced to limit the range of possible choices and speed up the optimization as explained further. This may however result in sub-optimal solutions and is not recommended.

6.2.2 Further search space reduction

Let's consider an installation of an OWF where pre-piling is needed to install jackets, and a separate vessel is hired for pre-piling. The start day of a campaign (i.e. start day of pre-piling vessel rent) is bounded by 365 as an upper limit. However, it can be reasonably assumed that in reality the installation usually does not start in autumn due to weather conditions, winter holidays, etc. Thus, the upper boundary can be reduced.
Similarly, a starting day of substructure installation (counting from pre-piling campaign start) is bounded between 0 and 365. It is obvious that starting on the same day as pre-piling will result in significant delays due to the fact that there will not be enough piles installed to proceed with jackets installation. Thus, the lower limit can be increased to a reasonable number of days, which a decision-maker expects the pre-piling for the first several jackets will take. The upper bound can be reduced as well as it was explained for the pre-piling start day.
Same logic can be applied to a turbine installation starting day (counting from substructure installation starting day).

Apart from the aforementioned considerations, the following has to be taken into account. As a result of several optimization trials, it was discovered that a near-optimal solution often splits installation of substructures and turbines into two years. In other words, in the beginning of the first year, an installation of piles commences, followed by an installation of jackets. Usually, for a typical OWF of 60 turbines constructed in the North Sea, all substructures would be installed by the middle of the first year autumn. In this case, starting an installation of turbines in the next year might be more beneficial seeing that the period of harsh weather conditions, i.e. October - January would be avoided by the turbine vessel, so less delays are encountered and vessel hiring costs will be lower. Often this is an optimal solution found by Optimizer.

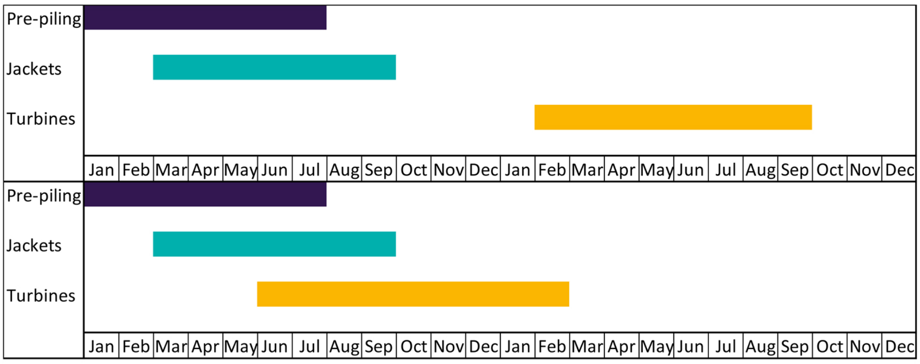


Figure 6.2: Installation in one and two years

Sometimes a decision maker would like to avoid such a situation, being ready to pay more for vessel delays but having the project finished earlier. In order to do so, the upper boundary on the turbine installation starting day has to be further reduced to e.g. 100, so that the installation of turbines starts maximum 100 days after the beginning of substructure installation and there are no breaks in the entire campaign. This is illustrated above, in figure 6.2, with the *"split"* solution in the upper part and *"no breaks"* solution in the lower one.

Such manipulations have impact on the outcomes of an optimization and may result in finding sub-optimal solutions. Therefore, it is important to have a clear understanding of how reduction of search space affects Optimizer output.

6.3 Algorithm. Stochastic metaheuristics and Model-based approach

In this section some starting points for the selection of the most suitable algorithms among the wide range are given. Two algorithms, namely Particle Swarm Optimization and Radial Basis Function, are selected and described.

6.3.1 Algorithm selection

Overview

In section 6.1 it was stated that the algorithm selected for the Optimizer needs to be able to work with a *black box* type of function and to find an optimum with as few evaluations as possible. In such algorithms every iteration is a process of determining the next point to evaluate based on the previously obtained evaluations. In this way, the best algorithm is deemed to be the one that requires the least NFE to reach an optimum (Rios and Sahinidis, 2012). The algorithms can be classified as follows:

- Based on how the search direction is determined:
  - Direct - determine the search direction by computing values of the actual objective function.
  - Model-based - construct and utilize a surrogate model of the objective function to guide the search.

- Based on the search domain coverage:
  - Global - able to refine their search domain arbitrarily.
  - Local - only converge to a local stationary point.
- Based on the need for random search steps:
  - Deterministic - do not need random search steps. Each run results in the same optimum.
  - Stochastic / Random - employ random number generator at some point. Each run may result in a slightly different optimum.

As it was mentioned earlier, local search algorithms are not suitable *a priori* since the objective function has multiple extremums. Further, due to the fact that all decision variables are integers, only a limited set of algorithms is remaining capable to handle combinatorial NP-hard problems (Conn et al., 2009).

Deterministic **exact** algorithms have an advantage of finding a global optimum. However, they are computationally expensive and often require objective function to be convex. Therefore, less demanding stochastic algorithms have to be exploited. These algorithms do not explore an entire search space and only guarantee to find a *good* result. As mentioned in chapter 2, some researchers have applied meta-heuristic techniques in order to guide the search. Often when dealing with noisy black box function, nature-inspired metaheuristics are employed (Collet, 2006), as it was done in (Maljaars, 2017), (Kumar, 2017). These techniques have proved to be able to find a *good* solution without relying on too many evaluations.

On the other hand, Xiao et al. (2018) in their article have compared model-based algorithms with stochastic meta-heuristics and concluded that the choice between these two has to be made based on the size of a problem. Generally speaking, both of them can be applied in derivative-free combinatorial optimization. However, the authors suggested to use model-based algorithms where the size of a search space is relatively small, whilst advocating for stochastic meta-heuristic algorithms where search space is large and response surface has a complicated form.

Selected algorithms

In this thesis stochastic meta-heuristic algorithm, Particle Swarm Optimization (PSO), will be compared with a model-based deterministic algorithm employing the Radial Basis Function (RBF) method proposed by Gutmann (2001). The goal is to investigate how model-based approach performs with respect to a more conventional direct one, specifically for the purposes of computationally demanding derivative-free optimization.

A hypothesis emerging from the literature is that model-based algorithms may turn out to be superior compared to the direct ones in terms of required NFE and an attained fitness value. The rationale for this is that they allow to use deterministic solvers which usually need many function evaluations or rely on derivatives. The latter are hard or impossible to obtain from a computationally heavy black-box function.

For the above reason, often only stochastic meta-heuristic algorithms remain as a last resort for a demanding black-box optimization. These stochastic algorithms, as it was shown in the literature, often require a large amount of NFE to find a good solution. Using meta-heuristic stochastic approach for problems of relatively small size might not be the most efficient way of solving.

That is where additional knowledge of the objective function derived from a surrogate model may be of a great help as it allows to directly obtain approximated function values or its derivatives from the model and to steer the search direction of deterministic algorithms.

Both of the specified algorithms are integrated into Optimizer block of the tool. The next two sections are dedicated to a detailed description of the two methods.

6.3.2 Particle Swarm Optimization

Selection of a stochastic algorithm

PSO algorithm belongs to a family of nature-inspired stochastic global-search meta-heuristic algorithms. Meta-heuristic is a process of guiding the search heuristic to efficiently produce good results. It dynamically balances between exploration and exploitation of the accumulated knowledge about the objective function (Collet, 2006). Several algorithms have been widely applied by researchers. Among them the most notable are Genetic Algorithms (GA), Simulated Annealing (SA) and PSO. Simulated Annealing arguably appears to be slow according to Collet (2006). In addition, in its search it only incorporates information from a single previous evaluation. When selecting an algorithm for the Optimizer it was discovered that generally GA and PSO are the most suitable as a vast amount of information is available on their performance and implementation. Albeit some articles, such as the ones by Fourie and Groenwold (2002), Hassan et al. (2004), compared the two and concluded that PSO might be better in terms of required NFE, it is important to keep in mind *No Free Lunch Theorem*. It states that there is no single algorithm which would perfectly fit for different black-box optimization problems. Therefore, a selection of black-box algorithms is problem dependent. Because a large benchmark comparison is not the purpose of this thesis, PSO algorithm was selected as a well-known example of stochastic algorithms.

For this project an implementation of PSO algorithm in Python by Lee, A. (2015) was used in a direct manner, i.e. with a true objective function. Several adjustments were made such as iteration-dependent algorithm parameters and an ability to handle integer variables. Originally, the algorithm was proposed by Kennedy and Eberhart (1995).

Description of PSO

In its basic form the PSO optimization reflects the social behavior of a swarm of bees. Each bee or *particle* is a vector in the search space of an optimization problem. A number of *particles / individuals* forms a *swarm* which resembles a swarm of bees. At every iteration the objective function is evaluated at each particle's position. After each iteration, each individual adjusts its position based on its current speed, its own previous best position and global best position found within the swarm. In this way particles profit from the past experience of the entire swarm as well as personal discoveries. Eventually, all particles move to a single optimal point and optimization converges. Thus, the algorithm consists of three steps:

1. Generate all particles' position and velocities. Evaluate the objective function in each particle's position. Assign each particle's best position to its current value, and assign the best swarm position to a point corresponding to the best across all calculated values.

2. Update particles' positions and velocities based on the known information. Evaluate the objective function in each particle's position. Update particle's best position if

an improvement in their objective value took place. Update swarm's best position if one of the particles resulted in a better value.

3. Repeat step 2 until one of the termination criteria is met.

The following equations describe how particles' velocity 6.5 and position 6.6 are updated:

$$\boldsymbol{V}_i^{k+1} = \omega \boldsymbol{V}_i^k + c_p r_p \frac{(\boldsymbol{P}_i^k - \boldsymbol{X}_i^k)}{\Delta t} + c_g r_g \frac{(\boldsymbol{P}_g^k - \boldsymbol{X}_i^k)}{\Delta t} \tag{6.5}$$

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{X}_i^k + \Delta t \boldsymbol{V}_i^{k+1} \tag{6.6}$$

Here the following notation is used:
$\boldsymbol{V}_i^k$ - velocity of particle i at iteration k
$\boldsymbol{X}_i^k$ - position of particle i at iteration k
$\boldsymbol{P}_i^k$ - best position of particle i
$\boldsymbol{P}_g^k$ - best position within the swarm at iteration k
ω - inertia coefficient
c_p - cognitive coefficient, a measure of attraction to the personal best position
c_g - social coefficient, a measure of attraction to the collective best position
r_p, r_g - uniformly generated random number between 0 and 1
Δt - time step (assigned to 1 for simplicity)

Looking at equation 6.5, three terms are distinguished. The first term $\omega \boldsymbol{V}_i^k$ is responsible for the inertial movement of a particle; the second term $c_p r_p(\boldsymbol{P}_i^k - \boldsymbol{X}_i^k)$ describes how a particle is attracted by its own best location; third term $c_g r_g(\boldsymbol{P}_g^k - \boldsymbol{X}_i^k)$ describes how a particle is attracted by the public knowledge of best position.

In order to prevent particles from crossing the borders of a search space, the inelastic bouncing is imitated. Moreover, the algorithm was adjusted to work with integer variables - the actual calculated positions are not changed but they are rounded off prior to using them as the inputs of the objective function.

Algorithm tuning

From the above-presented structure of PSO it can be seen that three parameters are used to control performance of the algorithm, mainly ω, c_p and c_g. High cognitive coefficient c_p results in excessive wandering of particles in isolation, while high social coefficient c_g yields premature convergence to what might be a local minimum. There are no general guidelines on how to select the values for these parameters, however it is proved that dynamically assigned iteration-dependent values allow for a better control over an optimization (Laskari et al., 2002), (Maljaars, 2017).

Furthermore, a choice has to be made with regard to a number of particles. Too many particles would result in a high NFE at each iteration and consequently high computational cost. On the other hand, too low number of particles hinders the convergence rate and requires much more iterations to thoroughly probe an entire solution space (which eventually also results into high NFE). In literature, a number of particles is usually selected with respect to the number of search space dimensions d and varies between $3d$ to $10d$ (Laskari et al., 2002), (Fourie and Groenwold, 2002). Again, the final choice is problem dependent and can only be determined by trial and comparison.

Another decision is related to a number of iterations n_i. Ideally, this should not be limited if one wants to reach a global optimum. An optimization has to be stopped when the improvement of the best swarm value between two consecutive iterations is less than some predefined number. However, this results in a large computational time which is often undesired. Within this thesis, the main termination criterion will be related to a number of function evaluations, i.e. independent of iterations number.

Finally, important is a choice of the starting points. Albeit a conventional option is to assign each coordinate of a particle randomly, one might end up with some points clustered, which might lead to a premature convergence in a local extremum. This can be especially crucial when the number of particles is low with regard to a problem dimensionality. In this case one should favor samples which are well distributed over each coordinate, i.e. are not too close to each other. A standard random uniform sampling may not be the best for this purpose and some more advanced techniques may be used to explore a solution space more diversely. Latin Hypercube Sampling (LHS) aims to spread the sample points more evenly across the space (Chrisman, 2014). The underlying principle of LHS is that a square grid is a Latin square if and only if each row and column contains only one sample. It can be easily generalised for n dimensions. This method will be used in this thesis to initialize particles positions.

Adopted settings

As it was mentioned before, there cannot be any universal guidelines with regard to the values of algorithm parameters. Their selection is problem-specific and should to a possible extent reflect behavior of the objective function (smoothness and presence of local extremums). Within this thesis they have been chosen based on the following general logic.

Inertia weight ω is set at higher levels in the beginning of optimization to facilitate search space exploration and higher rate of random travelling. It imitates higher impact of personal velocity rather than attraction to the potential optimum. Later, with the increase in iteration number, particles become less driven by their own current velocity and are more directed to the optimum.

Cognitive coefficient c_g will follow similar pattern as the inertia weight so that in the beginning of the optimization more attention is being paid to a thorough exploration of several local optima. Each particle will probe its own sector of the search space. In the following iterations all particles will be moving towards a global swarm optimum and cognitive coefficient will not play a noticeable role.

Social coefficient c_p is set to be low in the beginning of the search, and increases towards the end for exactly the same reasoning as inertia and cognitive coefficient. It almost does not affect the search dynamics in the first iterations and gradually takes over in the following iterations. In the last two iterations, social coefficient significantly dominates over inertia weight and cognitive coefficient and ensures that all particles converge to the vicinity of what is deemed to be the global optimum.

Accounting for the above reasoning the following parameter settings were used:

- The number of particles will be set to three times the dimension of the problem, thus 15. This was done in order to allow for more iterations keeping the computational time reduced.

- The maximum NFE is set to 150, thus the number of iterations is 10. It is a relatively low number. One can say that even though coefficients ω, c_p and c_g are adjusted in such a way that particles will converge, it does not however mean that a true global optimum will be found. Convergence is a consequence of parameters' values rather than of finding an optimum. This was done in order to reduce the computational time of the problem, possibly compromising the quality of a final solution.

- Figure 6.3 shows the values of ω, c_p and c_g for each iteration. There is an additional multiplication term r_p and r_g for cognitive and social coefficients, respectively. This term uniformly takes values between 0 and 1, i.e. 0.5 on average. In order to make the average value of cognitive and social terms to be comparable with the inertia, their maximum value is set to be 1.5. The coefficients apply starting from the second iteration, velocities for the first iteration are generated randomly. At the sixth iteration, cognitive and social coefficients have similar impact, their average value over all particles is 0.5, while inertia coefficient is slightly higher – 0.6.

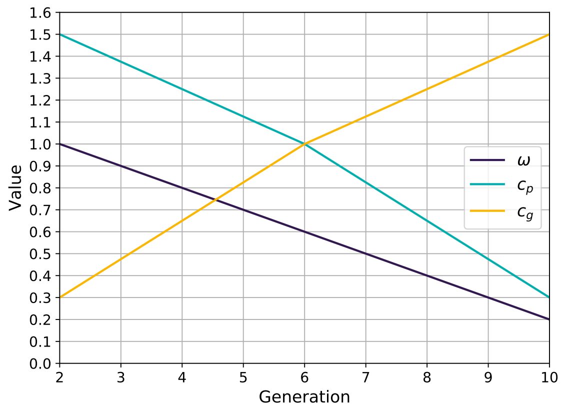


Figure 6.3: Adopted values for PSO parameters

6.3.3 Radial Basis Function model-based method

Selection of a modelling method

At the root of RBF model-based method lies the concept of *surrogate-modelling (response-surface modelling)* often used for an optimization of computationally expensive functions. A model is used to approximate the behavior of an objective function. Albeit being less accurate, a surrogate model is cheaper to evaluate and allows to use deterministic solver algorithms relying on the first and second derivatives. For black-box optimization with a computationally expensive objective function this is impossible to do with direct algorithms.

RBF method is related to a family of interpolating functional methods, which produce functions that pass through the sampled responses. Usually, these methods consist of: a class of basis functions, procedure for sampling, fitting criterion and some procedure to combine them all. A good overview of different model-based algorithms is given in Conn et al. (2009). This paper, together with Rios and Sahinidis (2012), served as a major source for comparing various optimization techniques.

RBF-based methods have reported to produce good results on a various set of problems (Holmström et al., 2008), and specifically with highly nonlinear objective functions (Fang and Horstemeyer, 2005). Even when compared with other popular techniques for surrogate modelling, such as *regression splines, kriging* and *polynomial regression*, RBF models appear to be superior, in particular for small and scarce sample sets (Jin et al., 2001). Nevertheless, researchers often stress that the performance of the surrogate models strongly depends on the sampling technique and less on the basis function or fitting criterion which are employed (Rios and Sahinidis, 2012), (Conn et al., 2009).

In this thesis RBF model was used, seeing that: a limited NFE is sought, the objective function is highly nonlinear. Originally, the algorithm was proposed by (Gutmann, 2001). The implementation of RBF algorithm in Python by (Costa and Nannicini, 2018) will be used. No adjustments were made in this thesis, seeing that the purpose was not to develop own method but rather to compare a performance of two families of algorithms - direct stochastic (PSO) against model-based deterministic (RBF).

Description of RBF method

The following description is solely based on (Costa and Nannicini, 2018) and (Gutmann, 2001). Only main steps will be explained to keep the narrative short. Interested reader is encouraged to take a look at the underlying papers which contain a thorough overview of the algorithm.

Let $\Omega = [x^L; x^U] \subset \mathbb{R}^d$ and $\Omega_I = \Omega \cap (\mathbb{Z}^d)$, where: x^L - lower bounds on decision variables x, x^U - upper bounds, d - number of problem dimensions (all variables are integers in the studied problem). Given distinct points $x_1, ..., x_k \in \Omega$, an RBF interpolant s_k is defined as:

$$s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\| x - x_i \|) + p(x), \tag{6.7}$$

where ϕ: $\mathbb{R}_+ \to \mathbb{R}$, $\lambda_1, ..., \lambda_k \in \mathbb{R}$ and p is a polynomial of degree n, which is typically picked to be exactly d_{min} (minimum degree to guarantee the existence of interpolant depending on the form of functions ϕ). Table 6.2 contains the most common RBF functions $\phi(r)$ and related d_{min}. The parameter $\gamma > 0$ is used to control the shape of a function and is normally set to 1. The values of λ_i and $p(x)$ are determined by solving linear system of equations which will not be presented here.

Table 6.2: Common radial basis functions

ϕ		d_{min}
r	(linear)	0
r^3	(cubic)	1
$\sqrt{r^2 - \gamma^2}$	(multiquadratic)	0
$r^2 log(r)$	(thin plate spline)	1

A general representation of the algorithm is:

1. Initial step: Choose independent points $x_1, ..., x_{d+1} \in \Omega_I$ using an initialization strategy (LHS by default). Set $k \leftarrow d + 1$.

2. Iteration step:

   - Compute the RBF interpolant s_k based on the points $x_1, ..., x_k$ (include points selected at the Initial step and points found at the previous Iteration steps).
   - Choose between *exploration* and *exploitation*.
     - At steps $h \in \{0, ..., \kappa\}$ (Global search) choose an expected minimum value $f_k^\star = s_k^{min} - (1 - h/\kappa)^2 (f_{max} - s_k^{min})$, where κ is a predefined number of steps per optimization cycle (6 by default); s_k^{min} is the minimum value

of the interpolant, found by a solver; f_{max} is the maximum value of the objective function obtained so far. This is an *exploration* phase which aims at improving the current model at the unknown part of search domain.

  – At step $h = \kappa$ (Local search) choose $f_k^\star = s_k^{min}$. This is an *exploitation* phase trying to find the best objective value based on a current interpolant.

- The assumption of Gutmann algorithm is that a likely location for the point y, where interpolant s is equal to the target minimum value of the actual objective function, is the one that minimizes the curvature of s. Essentially it means that the algorithm looks for the *"least bumpy"* interpolant. Thus, at this step algorithm finds point x_{k+1} where the most smooth interpolant would attain the value $f_k^\star$.
- Compute the actual objective function f at the point x_{k+1}.
- Set $k \leftarrow k+1$. If the prescribed optimization criteria has been reached - stop the search. Otherwise repeat Iteration step. The search will completely restart with different initial points if no improvement in the objective value has been achieved during the past 5 cycles.

In order to facilitate a better understanding of the fundamental principle of Gutmann algorithm, the following figure 6.4 shows what is meant by finding the *"least bumpy"* function with the smallest curvature.

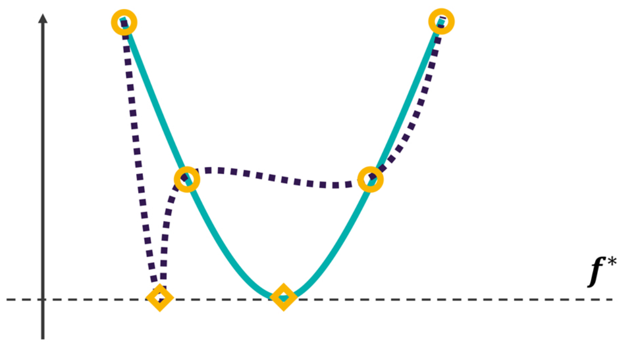


Figure 6.4: Gutmann algorithm's main principle (Gutmann, 2001)

Let's assume that values of the objective function are known in four points indicated by orange circles. The Gutmann algorithm suggests that a more likely point (orange diamond) to minimize the interpolant with value f^* is located at a solid aquamarine line, rather than a dotted purple one, because the resulting interpolant is less *"bumpy"*.

Several secondary modifications of the Gutmann algorithm have been added by the authors of implementation, as described in Costa and Nannicini (2018). The overall algorithm was not affected; hence they will not be reviewed here. The only modification, worth to mention, is an automatic selection of RBF basis function. An additional process of cross-validation is performed at each cycle of selecting f^*, which uses a part of the available function values to fit a model, and tests its quality on the remaining part. This is done to build the most precise surrogate model.

Modelling method tuning

The main adjustments which have effect on performance of the algorithm were done by the authors of the employed Python implementation. These are: 1) Dynamically selecting f_{max} in the Global search of each Iteration step; 2) Restriction of global minimization search box.

1. In Global search f_{max} is replaced with dynamically chosen value of $f(x_i)$. From the second step onward, progressively lower values of $f(x_i)$ are picked in order to stabilize search and avoid large difference between the expected target value and the interpolant minimum. This essentially means that the further the algorithm proceeds in search - the less likely is that it will find a much better value, and only small improvements are expected.

2. It was confirmed by researchers that in Global search high values of h do not necessarily imply that relatively local search is being performed. Authors claim that it might significantly affect convergence on problems where global minimum is located in a steep valley. Therefore, it is proposed to progressively reduce the search space for x_{k+1} around the best known solution during Global search steps. At the beginning of every Global search the search domain coincides with Ω_I but becomes smaller as h increases, and eventually converges to a small vicinity around the best known solution.

These adjustments were used by default in this project, seeing that the authors have done an extensive study and proved their contribution.

Solver algorithm

It is important to realize that the selected RBF method does not only serve as an intermediate step needed to construct a surrogate model. RBF method also prescribes how the search should proceed. For this purpose at each Iteration step an optimum of the interpolant is evaluated by a solver. Within the employed Python RBF library the use is made of an open-source Bonmin solver which uses exact- for convex and heuristic branch-and-bound global algorithm for non-convex objective functions (Belotti et al., 2019).

This is where the compromise lies – surrogate-based algorithms allow employing an exact solver for convex- or deterministic heuristics for non-convex functions, both of them relying on many evaluations of the objective function. At the same time, the function used by such deterministic solvers in this case is only an approximation of the computationally demanding true objective function, thus it is by definition less accurate. Hence, it is a trade-off between the ability to quickly evaluate the surrogate of the objective function and accuracy of the obtained values. In contrast, one can use accurate true objective function, at the cost of a significant increase in the time needed for an optimization, thus being limited by direct stochastic meta-heuristics.

6.4 Results comparison

As it was mentioned in section 6.1.1, the focus in this thesis will be on the NFE as the main metrics used to compare two algorithms. First, optimal solutions found by the optimization will be analyzed. Then, NFE and pace of optimization progress will be presented. Finally, the algorithms will be compared based on how they explore the search space. Note that in the rest of this chapter the algorithms will be addressed as PSO and RBF, even though RBF is a method to build surrogate model rather than a solving algorithm.

Table 6.3 below gives a brief formulation of the optimization case study. Appendix B.2 contains a complete definition.

Table 6.3: Optimization case study OWF

General parameters	
Location	North Sea
Type of an installation	4-leg jackets with pre-piling and turbines
Number of turbines	50
Capacity of piles vessel	20 piles
Capacity of substructure vessel	4 jackets
Capacity of turbine vessel	6 turbines
Distance to site	70 km
Decision variables	
Campaign start day d_c	0-365
Substructure campaign start day d_s	0-365
Turbine campaign start day d_t	0-365
Assembly lines number r	1-3
Jackets storage area a	5-15

Note that substructure start day d_s is counted from the project start date d_c, and turbine start day d_t is counted from substructure start day d_s. Hence, these variables show the difference between the starting days of campaigns rather then the actual start day value.

6.4.1 Overview

Table 6.4 presents an overview of the optimization outcomes. Note that the value of the objective function of two algorithms was normalized with respect to the lowest one, in this case RBF.

Table 6.4: Optimization results

Criterium	PSO	RBF
Normalized cost found	1.03	1
Optimal values [d_c, d_s, d_t, r, a]	[55, 54, 13, 2, 10]	[69, 46, 0, 2, 15]
NFE to find optimum	142	86
NFE to find optimum +5%	120	81
Evaluations per ”iteration/cycle”	15	6
Number of ”iterations/cycles”	10	25
Computational time	~60 [hr]	~60 [hr]

One can see that even with relatively different values of decision variables, the found objective values are quite close. The general tendency is that an optimal beginning of the project should be couple months after the 1^{st} of January, with about 1.5 month pause until the beginning of the substructure campaign (and assembly). Results for the duration of the needed pause between substructure campaign and turbine campaign differ by two weeks. In fact the optimum obtained by RBF suggests that they should start simultaneously. This is counter-intuitive because the turbine vessel might experience delays in the beginning waiting for the first substructures to be installed. Both approaches agreed on the fact that two assembly lines are optimal. The size of the jacket storage area is also very different.

The optimal solution was obtained by RBF after 86 evaluations of the actual objective function, with a solution within 5% margin of it obtained already after 81 evaluations. PSO performed worse and optimum was found after 142 evaluations with the first value within 5% margin being obtained after 120 evaluations. The total computational time of around 60 hours didn't allow to try out different algorithm settings or to conduct a more extensive study. The pace of optimization progress is analyzed deeper in 6.4.3.

6.4.2 Optimal solution

PSO

As a result of the optimization employing PSO algorithm, the starting day of campaign is taking place on the 24^{th} of February (day 55); the start of substructure campaign is 54 days later, i.e. on the 18^{th} of April; turbines installation start day is 13 days after that, i.e. on the 1^{st} of May; the optimal number of assembly lines is two and the capacity of the jackets storage area is 10. Pre-piling campaign in this case finishes on the 19^{th} of June, assembly - on the 12^{th} of July, substructure campaign - on the 18^{th} of July and the whole project - on the 30^{th} of October.

Figure 6.5 presents a timeline which corresponds to these parameters. In order to build this timeline, 20 simulations were done and average duration of each campaign were taken.

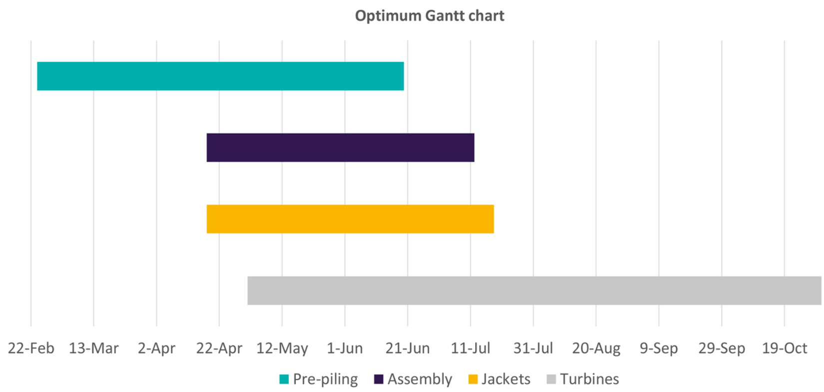


Figure 6.5: Gantt chart of the optimal solution found by PSO

RBF

Figure 6.6 presents a corresponding Gantt chart.

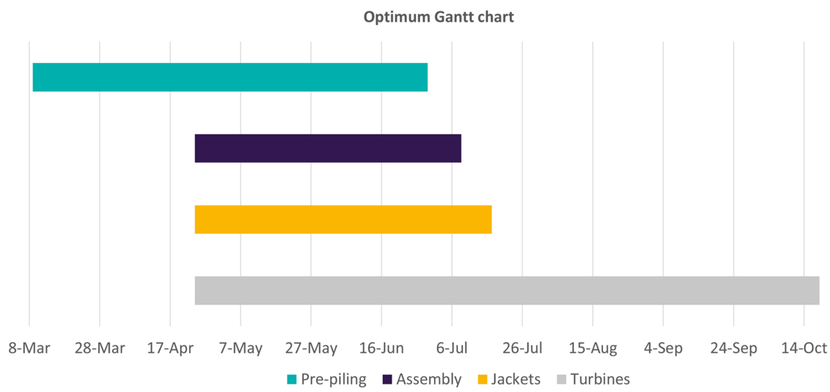


Figure 6.6: Gantt chart of the optimal solution found by RBF

As a result of the optimization based on the RBF surrogate modelling method, the starting day of campaign is 9^{th} of March (day 69); the start of substructure campaign is 46 days after, i.e. on the 24^{th} of April; turbines start the same day; the optimal number of assembly lines is 2 and the capacity of the jackets storage area is 15. Pre-piling campaign in this case finishes on the 29^{th} of June, assembly - on the 8^{th} of July, substructure campaign - on the 17^{th} of July and the whole project - on the 18^{th} of October.

One can reasonably assume that the obtained solution could be improved by starting turbine campaign even later. Currently, looking at the chart it can be seen that the duration of this campaign is much longer than that of pre-piling and substructure campaigns. Generally speaking, this differs per project. The weather conditions play a big role: substructure campaign is happening during the summer, while turbine campaign is extended to the autumn months. Thus, it would not be correct to always expect a certain difference in the total durations.

Nevertheless, it was tested how the change of d_t, i.e. the difference between turbine campaign start day and substructure one, will affect the result. It was changed from 0 to 18 and again 20 simulations were run. It turned out that duration of the turbine installation campaign remained the same while the objective value became worse. The total cost increased due to the contribution of extended period of paying for labor. Hence, the assumption that later turbine start would facilitate better outcomes is not correct. Most probably, the moment that the first substructure is installed, the turbine campaign can commence without further delays due to a lack of installed jackets.

Figure 6.7 shows how the expenditures averaged over 20 simulations were spread across different segments.

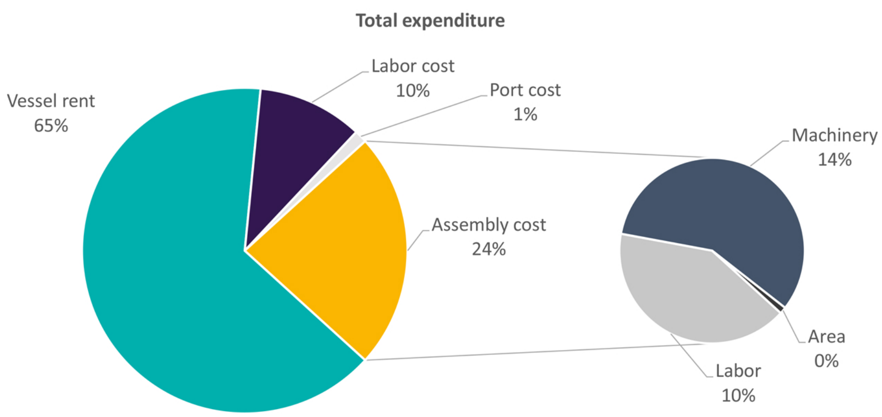


Figure 6.7: Cost spread of the optimal solution found by PSO and RBF

Two biggest contributors to the total costs are vessel rent and assembly cost. This is in agreement to what was explained when selecting the variables for optimization. Indeed, when optimized, these two factors may result in substantial cost reduction. Seeing that the contribution of vessel rent is more than two times larger, the optimization procedure will normally compromise larger assembly cost due to extra assembly lines in favor of reduced vessel delays in port.

The cost spread across different segments is absolutely identical between PSO and RBF solutions, thus presented only once.

6.4.3 NFE comparison

PSO

Figure 6.8 presents the search history for the PSO algorithm, showing how the obtained values of the objective function evolved from one iteration to another. All values are normalized to the optimum obtained by PSO.

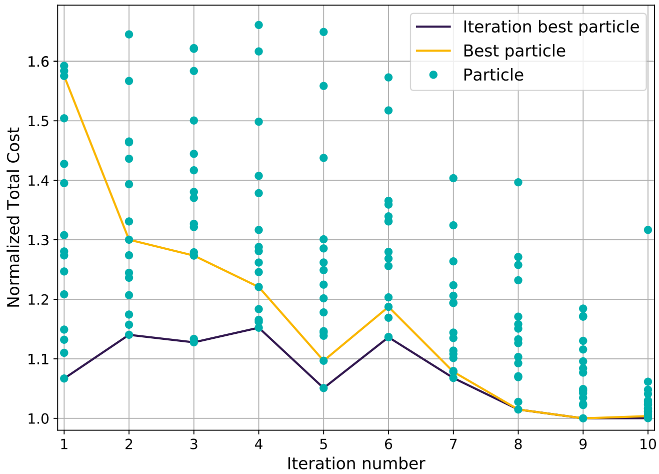


Figure 6.8: PSO: Swarm evolution

If analyzed deep, this graph is a reflection of figure 6.3, where PSO parameters were defined. One can see how in the beginning of the search the swarm was spread across the search space, and objective values varied a lot. Starting from the sixth iteration, the swarm begins its convergence to the optimum. Occasionally, a relatively good value was obtained already in the first iteration, however it was checked and decision variables were far from their final optimal position, thus it was only a locally good point.

By selecting the parameters of PSO algorithm properly, one can successfully avoid being trapped around local optimums in the beginning of the search. These areas are studied thoroughly by each particle but then the preference is given to a globally best point. In other terms, low social coefficient allows to make every individual in the swarm independent in the beginning of the search.

Thus, if the number of iterations is defined in advance, PSO algorithm allows to manually control the dynamics of search based on the simple principles explained in 6.3.2. At the same time, it is hard to predict which optimum would be found would the number of the iterations be larger. Accounting that the current optimization took around 60 hours, this was left out of this study.

RBF

Analogically, figure 6.9 shows the dynamics of RBF search. All values are normalized to the optimum obtained by RBF.

It is interesting to see how differently the two algorithms direct their search. PSO has a clear converging pattern beginning after the first half of the search. From the 6^{th} iteration onward, the best value in the swarm improves or at least remains the same. In contrast, RBF-based algorithm has done several restarts (every 5 cycles, i.e. every 30 evaluations) as mentioned in 6.3.3. Hence, it is difficult to say that the algorithm is converging as such, as each restart is essentially a completely new search with a new surrogate. Due to the sophisticated decision process of selecting between *exploration* and *exploitation*, the values of the objective function are not necessarily always improving.

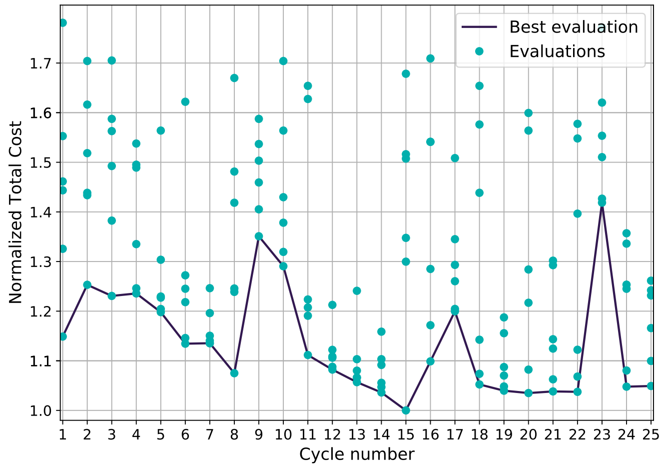


Figure 6.9: RBF: Search progress

PSO against RBF. NFE-based performance.

In figure 6.10 the reader can see how the two methods proceeded in their search. Since most of the improvement between the objective values happens from one iteration/cycle to another and not within them, it is hard to analyse the dynamics just based on the plot of objective function value against the number of evaluation. For this purpose an average was taken from the values obtained at each PSO iteration (every 15 evaluations) and RBF cycle (every 6 evaluations) and polynomials were fitted to both curves so that the reader can better see the trends. Note that all values are normalized to the optimum obtained by RBF method.

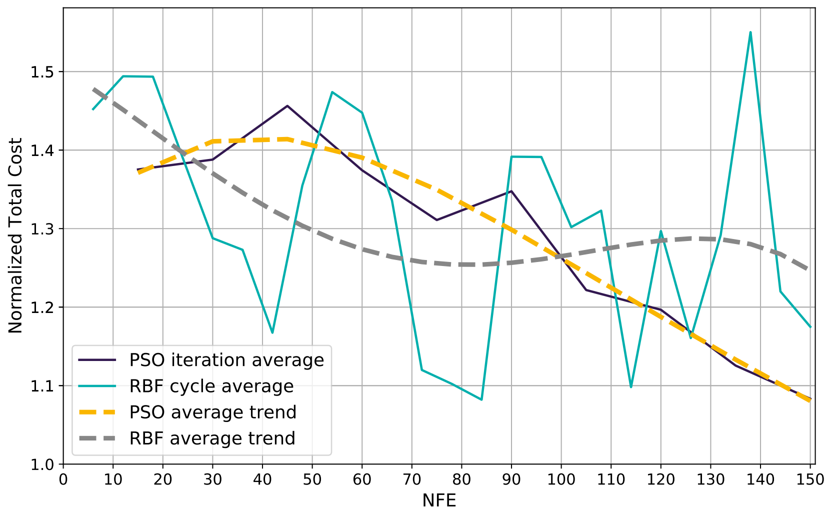


Figure 6.10: PSO and RBF objective values against NFE

Analyzing the graph, the first notable tendency is that in the beginning of the search (NFE from 15 to 40) the PSO trend line is growing. This means that the quality of solutions gets worse and particles are spread over entire search space, as it was actually intended by selecting the values of PSO parameters (see figure 6.3 and explanation to it). The first three iterations (NFE from 0 to 45) correspond to a global search with each particle being relatively independent from the entire swarm. After that the swarm begins its movement to what is deemed to be optimum.

Oppositely, RBF method trend line is descending from the very beginning and already after ∼40 evaluations, or at the 8^{th} cycle, the average is within 20% margin of the optimum and the best obtained value is within 10% margin of the optimum. This can also be seen from figure 6.9. Of course, later due to the restarts of the search, this tendency changes, and the trend line even grows.

The presence of the surrogate model allows to direct the search towards an optimum faster. With the first 40 NFE, while PSO individuals are still exploring the search space, RBF method already constructs a surrogate model, which albeit not being perfectly accurate, contains much more information of the objective function which is not yet available to the PSO algorithm. An obvious recommendation would be: if the goal is to quickly find a relatively good solution, model-based (RBF method) algorithm might be a better option compared to a metaheuristic one (PSO).

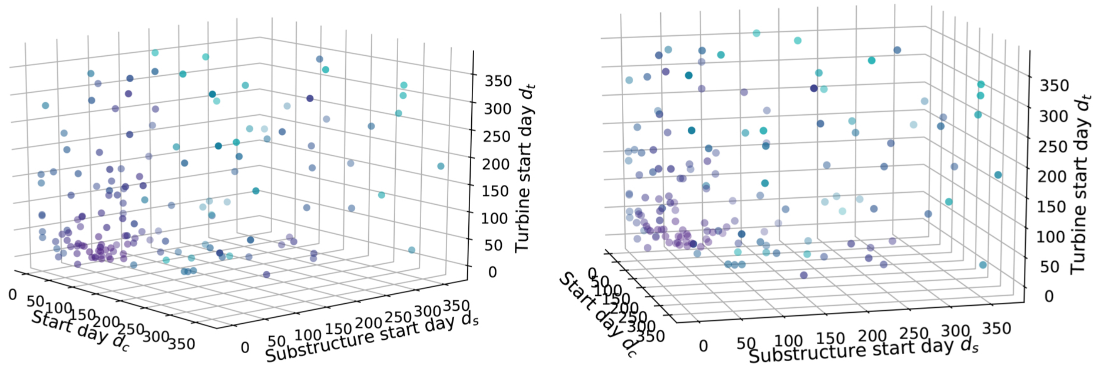
However, as mentioned in the beginning of this chapter, a *No free-lunch theorem* claims that what worked for one optimization problem may not necessarily be the case for another. In this thesis, a hypothesis of Xiao et al. (2018) was confirmed. Metaheuristic algorithm performed worse compared to a model-based one, for a problem with low number of decision variables with a relatively small search space. On the other hand, it could also be that if the objective function was even more noisy, RBF method would fail due to an inability to construct a good surrogate. This is just another assumption which cannot be validated directly.

With the above being said, the conclusion is that whenever possible one should try to investigate how different families of optimization algorithms perform. As it was shown here, the popular stochastic metaheuristics is indeed robust in finding a good solution. Nonetheless, surrogate model-based deterministic algorithm was capable of obtaining a good solution quickly. This might be crucial when the objective function is computationally heavy as it is in this thesis, and therefore low NFE is appreciated. If an evaluation of the objective is not expensive, then metaheuristics might be preferred over other options since it is usually simple in implementing, quite intuitive to understand and can be tuned easily.

6.4.4 Search space exploration

PSO

First, figure 6.11 focuses on the starting days, since these variables have a wider range of possible options. It is interesting to track how PSO algorithm probes their value domain. Two views of the same data are presented to facilitate better comprehension of 3-dimensional plots. The color of particles reflects the value of the objective function: more purple particles correspond to lower objective function values, while aquamarine particles correspond to a higher one.



(a) (b)

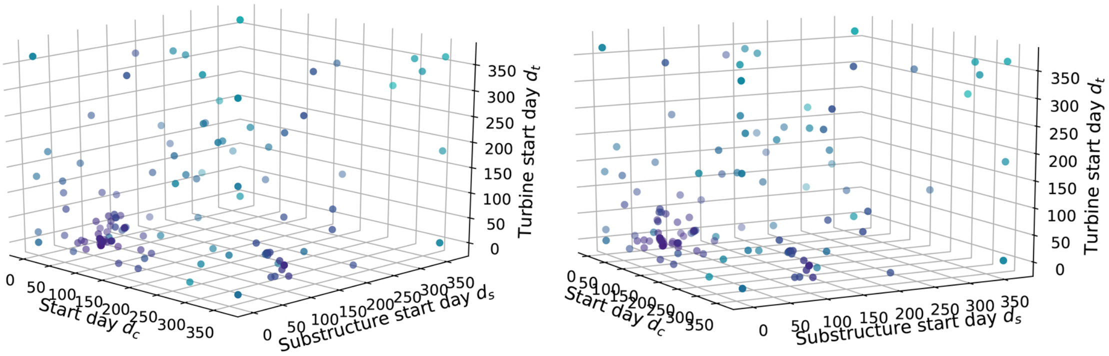
Figure 6.11: PSO objective function vs starting days d_c, d_s, d_t

It can be seen that the search space has been explored well, implying that different regions have been visited by the swarm. Next, these graphs one more time confirm that the objective function has many local optima. Multiple purple particles can be found far from the final optimum (vicinity of [d_c, d_s, d_t] = [55, 54, 13] with a high concentration of purple dots). At the same time, several aquamarine particles are very close to an optimum, implying that it is located in a *"steep valley"*.

Additionally, Appendix E contains multiple plots of objective value as a function of each variable, individually or pairwise, both for PSO and RBF. The reader is encouraged to have a look as the further discussion will be partly based on the results presented in Appendix.

RBF

Figure 6.12 presents how the search space of start days was explored by RBF algorithm.



(a) (b)

Figure 6.12: RBF objective function vs starting days d_c, d_s, d_t

Not obvious from the first impression, but particles are not covering the search space as uniformly as in the figure 6.11. This is in line with the fact that additional knowledge obtained from the surrogate model steers the search towards an optimum quicker than in

PSO search. In other words, PSO particles inevitably go through all intermediate points on their way from an initial to a final point. RBF method acts differently and does dedicated evaluations near available optimum (Local search or *exploitation* step, see 6.3.3). The difference between search space coverage is also visible when comparing figures E.2, E.3, E.4, E.5 for PSO with corresponding E.7, E.8, E.9, E.10 for RBF.

From the graphs in Appendix E, it is visible that albeit the plots for start days have similar pattern, the plots of assembly lines number and jackets storage area are quite different. While PSO algorithm found 10 jackets as an optimal size of the storage, according to RBF this is the worst option. Once again it raises an important point that decision variables in this problem are highly interdependent. It is impossible to conclude that a given size of the storage area is the best no matter which value the other variables will take.

The set of obtained optimal values has to be treated as one entity. Only in combination with other variables proposed by Optimizer a storage size of 10 is good. Would the other parameters have different values, the optimal size will also change.

6.5 Summary

This chapter presented the optimization study conducted in this project. As a first step, five parameters were determined for the optimization: start day of pre-piling campaign, substructure campaign start day, turbine campaign start day, number of jackets pre-assembly lines in harbor and jackets storage size in a harbor. These variables were chosen based on their impact on the total cost of the project and difficulty to optimize manually.

Secondly, an overview of the existing optimization algorithms was done. Specifically, focus was on the algorithms which can cope with computationally heavy integer black box optimization. *No free-lunch theorem* says that there is no single optimization algorithm that would be a good method for any optimization problem. In order to investigate which algorithms perform better for the optimization problem in this thesis, two algorithms were selected based on the literature study. The first is direct stochastic metaheuristics algorithm called Particle Swarm Optimization. The algorithm uses the actual objective function, contains stochastic component in its search, and is inspired by nature processes. The second algorithm is surrogate model-based deterministic approach based on the Radial Basis Function method for building surrogate models. This algorithms exploits surrogate model of the actual objective function in order to steer its search. The surrogate model is obtained by fitting polynomials through already sampled points of the objective function.

A hypothesis from the literature, claiming that model-based algorithms may appear superior compared to metaheuristic ones in case of relatively small optimization problems, was confirmed. Indeed, model-based algorithm was able to find an optimum with much less function evaluations than a metaheuristic one. When the objective function is computationally intensive, it might be beneficial to be able to obtain a good solution with a limited number of evaluations.

Some insights into the performance of algorithms was obtained by analysing how the search space was explored and how the algorithms proceeded from one iteration to another. It was concluded, that PSO algorithm explores search space more thoroughly if appropriately tuned. It is also possible to adjust algorithm in such a way that local optima found in the beginning of the search will be avoided, and no preliminary convergence will happen. This is advantageous when dealing with highly fluctuating objective functions.
For the RBF-based algorithm a use was made of a surrogate model which quickly provides extra information about the objective function behavior and allows to use this information for steering a search into what is deemed to be an optimum.

Chapter 7

Conclusions

This final chapter gives an outlook on the conducted research. The objective to develop a decision-support tool for the offshore wind farm installation planning, as it was formulated in the beginning of this report, was achieved. The developed tool comprises two fundamental blocks, namely a simulator and an optimizer. In order to tackle weather uncertainty and its effect on the installation timeline, a separate ancillary weather generator was developed. Consequently, Simulator is able to produce realistic project timeline depending on the geographic location and time of the year. A big advantage of Simulator is its high level of flexibility towards user-defined inputs, i.e. a variety of installation parameters can be adjusted in order to imitate an installation close to reality. Furthermore, in order to assist a decision-maker with taking the most cost efficient decisions in the process of installation planning, Optimizer was implemented. It was applied to a future oriented scenario where OWF installation and partial substructure assembly in port were combined. As a result of the optimization, the most optimal combination of values for the installation start day, start of each separate campaign (pre-piling, substructures and turbines), onshore jackets storage area and a number of jackets pre-assembly lines is obtained. This being said, the tool also incorporated some insights into onshore port logistics.

A recap of the formulated design requirements and their accomplishments is given in 7.1. The most important research findings from each phase of the study are summarized in 7.2. Recommendations for the future investigations are given in 7.3.

7.1 Fulfillment of the design requirements

The primary goal of this research project was to develop a decision-support tool for offshore wind farm installation planning. Below is a generalised overview of the requirements towards the tool which were formulated by the Innovation Department of SGRE. A full list and an intermediate review are presented in 2.1 and 3.3, respectively.

1.Inputs and scope of the tool

In what concerns inputs and scope of the tool, the focus was on the offshore installation of support structures and turbines with some limited insight into onshore port logistics. Indeed, the majority of input parameters are related to the installation of the above components, allowing to analyse an installation of the wind farms of any size, with different type of substructures and various vessels. The user is free to specify vessel operational limits and durations of each operation so that project-specific settings can be used. Moreover, the operations themselves are largely defined by user and only several main steps are *"hard-coded"* to take care of the general order of installation processes.

2.Weather considerations

Weather conditions is the major factor determining whether a vessel is allowed to perform an operation offshore or a delay in the project will be encountered. Every single delay induces large additional costs due to the high rates of vessel rent and other project expenditures. The total capital expenditure related to OWF installation is very sensitive to the overall project duration. Therefore, it is desirable to plan the project taking into account weather specifics, in such a way that the delays are minimized. A record of the past wind speed and wave height measurements is needed. With the help of this data, one can analyze which periods of the year are more favorable to conduct a project and which should be avoided. For this purpose, a weather generating block was added into the tool. This block generates multiple data series based on past measurements and allows to obtain statistically valid predictions on how meteorologic circumstances may affect the installation campaign in a certain geographical location at a certain time of the year.

3.Flexibility of the tool

Flexibility was one of the major requirements, seeing that the tool must be able to handle installation strategies which have not been realised yet and are conceptual as such. Essentially, it was needed to create an additional interface where a user would be able to adjust an installation process as much as possible to imitate specific installation strategies. Hence, it was important to minimize the amount of *"hard-coded"* installation logic and allow user to specify it externally. An independent input file was used, where a user is free to specify his own sequence of offshore installation activities. Consequently, for the program code it does not make a difference which particular activity is being performed. Any type of substructure, with its own corresponding installation procedure, can be simulated. The tool only ensures that the whole cycle and each separate activity can be performed without violating corresponding weather limits. Furthermore, a range of parameters was added to control onshore loading (e.g. loading several components in parallel, etc.).

4.Outputs of the tool

Several different metrics and KPIs can be used in order to assess the attractiveness of a certain installation strategy. Various factors are important when looking at the overall expected project progress. Hence, it was important to make it possible to extract from the tool complete information about the project flow. A detailed project timeline, daily progress, workability recordings, durations of each loading and installation are just some of the possible output statistics. All outputs are saved in Excel files so that they can be analysed with standard office software. These are automatically generated after each simulation and virtually all needed indicators can be derived out of this statistics. It can be concluded that two sub-goals were accomplished in what concerns the tool outputs - deep level of insight and simplicity of representation.

5.Performance of the tool

As far as the performance of the tool is concerned, mainly the computational time is implied. Provided a single run can be done quickly, multiple iterations can be done by a decision-maker to test different settings and select the best one. Considering Simulator block, this objective was achieved - a single simulation takes 40-60 seconds. Due to the fact that sometimes multiple simulations are needed to obtain statistically representative outcomes for a certain location, up to 20 simulations may need to be run which results in 20 minutes of computational time. It is a good value considering how much information can be derived. On the other hand, the Optimizer block of the tool is much more demanding as its computational time is at least two orders higher. When using Particle Swarm Optimization algorithm, up to $\sim$120 runs are needed resulting in more than two days of running. If Radial Basis Function model-based solver is used, then around 70 runs are needed and that is considered a large improvement. Furthermore, the accuracy can be slightly compromised against computational time reducing the duration even further - 50 second per simulation times 15 simulations for statistical validity and 70 optimization iterations resulting in $\sim$15 hours to obtain an optimum.
Last but not least, it is worth to mention that the tool was validated with the real OWF installation project performed by SGRE and produced results similar to the actual timeline.

7.2 Research findings

1.Conceptual design of the tool

Once the list of requirements was formulated, a literature review has been conducted in order to create a conceptual design (see Fig. 2.7) of the envisaged tool. In this project a relatively universal design was proposed taking into account previous research and state-of-the-art developments in the industry. It was decided to split the tool into two major blocks, namely Simulator and Optimizer. The former carries the functionality of quickly producing estimated project timetable based on the combination of user inputs and weather data for a specific location. The latter one advises a user on a the most crucial parameters of the installation so that the total cost of the project is reduced. It was not necessary to define installation problem separately for optimization, because Optimizer was designed as a shell around Simulator. Optimizer effectively uses Simulator to calculate an objective function value.

It turned out that special attention has to be payed to the weather modelling, seeing that meteorological conditions may vary significantly from year to year and therefore several simulations may need to be run for the same location in order to obtain statistically accurate results. This required the development of a separate weather modelling block, because often only a limited number of past records is available, and there is a need to generate synthetic weather series.

2.Discrete Event Simulation

Discrete Event Simulation proved its reputation of being a very suitable paradigm for modelling complex logistic processes and running simulations within a relatively short period of time. This is an established technique with an abundance of sources available. The author recommends opting for DES when dynamic systems with complex interrelations, multiple working agents and resources are being modelled.

Important is that events in such system occur only at discrete moments of time, and no change happens in the system between two consecutive events. In this manner, one does not need to employ computationally demanding continuous simulation and can significantly reduce computational time. Within this thesis DES was the main paradigm used for Simulator block, which completely determined its high efficiency and short computational time ($\sim$50 sec for a single simulation of a complete offshore wind farm installation).

3.Synthetic Weather generation. Markov Chains.

As already mentioned, there is a need to generate multiple synthetic weather series in order to obtain statistically valid results from multiple simulations for a given location. Among common statistical parameters characterising weather data, persistence of weather states was identified as the most crucial for offshore operations planning. The reason is that a vessel can only perform offshore provided that sufficiently long weather window is ensured. In other words, it can be expected that within the approximate duration of the operation, the weather conditions will not exceed vessel's operational limits.

Markov Chains were adopted for the purpose of weather modelling. It was, however, discovered that first-order Markov Chains do not preserve all needed statistical characteristics of original weather series (see 4.1). In particular, the generated signal was characterised by a very intermittent behavior, i.e. very unstable compared to real measurements. Hence, the persistence of weather conditions was not imitated properly.

A two-step algorithm was proposed to adjust the original Markov Chain method. As a first step, moving average of the generated signal is taken in order to smoothen the high-frequency oscillations in a generated wind and wave series. As a second step, the value of the wind speed or wave height signal is kept constant where a generated state signal contains a sequence of several hours in the same state (see 4.4). These adjustments significantly improved generated series and allowed to use them for the simulations.

Apart from that, Markov Chains proved to be a good methodology for weather modelling when a seasonal character of wind and waves, as well as a correlation between the two, have to be preserved.

4.Sensitivity to inputs

While running multiple simulations it became clear that the outcomes of a simulation are highly dependent on the inputs and specific weather realization.

Due to the the fact that every simulated activity depends on the past progress of the entire installation campaign, a small change in the timeline at the start of the project may accumulate to the end and result in significant differences with an alternative realization. Specific weather conditions may introduce a minor delay at the beginning, which will shift entire project dynamics and yield quite different outcomes. Therefore, the author suggests to pay attention to values of the input parameters, especially of the operational limits and to a smaller extent of the durations of operations. Preferably, in order to deal with the stochastic effect of weather, it is recommended to run multiple simulations, with the weather realizations for different years, and analyze the obtained statistical KPIs – average values, variance, $p - 90$, etc.

Furthermore, when analyzing the average duration of loading in port or installation on site, one has to take into account how many of these procedures took place during the project. The number of loading cycles is usually relatively low, therefore a large deviation in the duration of just one cycle will significantly affect the overall average.

5.Optimization. Computationally-heavy Black-box functions.
An optimization problem studied in this thesis employed total installation cost as the objective function. Five decision variables have been identified due to their impact on the total cost and inability to find their optimal values manually. These variables are: start day of pre-piling campaign, start day of substructure campaign, start day of turbine campaign, number of jackets pre-assembly lines onshore and size of jackets storage area.

The type of the studied problem is a so-called *black box* optimization. This means that no direct relation is known between the decision variables and a value of the objective function. Hence, it is not possible to talk about the derivatives and even finite differences methods cannot be used due to a large amount of time needed to obtain function values. The fact that an optimization is *gradient-free* imposed strong limitations on applicable algorithms. Consequently, the main goal of the optimization study was to compare two types of optimization methods based on how many function evaluations they need in order to find a good solution. A method which requires a smaller number of function evaluations (NFE) was deemed to be superior since it can save significant amount of time.

Two approaches were compared: direct stochastic metaheuristics Particle Swarm Optimization (PSO) against model-based deterministic solver based on the Radial Basis Function (RBF) method. The former is a well-established technique for dealing with this type of optimization problems, while the latter was proposed as a more efficient alternative based on the literature review. It turned out that for a relatively small optimization problem studied in this thesis, model-based approach with a deterministic solver worked better, i.e. required lower NFE in order to find an optimum.

Summarizing the outcomes, the author suggests that using model-based optimization algorithms might perform better compared to more traditional stochastic metaheuristics for a certain range of problems. On the other hand, selection of an algorithm is problem-dependent and cannot be done without prior testing and comparison. Albeit nature-inspired metaheuristics, such as the one reviewed in this project, are simple to understand and implement and can be easily tuned, more sophisticated model-based techniques may allow to reduce computational time.

7.3 Recommendations

Throughout the research several points of attention were identified. These aspects proved to give extra insight or may lead to new findings.

1.Scope
In what relates to the scope of the project the main added value is related to adding onshore port processes into the overall flow. The onshore logistics was considered up to a point of onshore storage and jackets pre-assembly. It was shown that assembly has an essential impact on the total cost and may affect the timeline. When onshore campaigns are added into the simulator scope, they can bring extra value for an optimal planning and possibly identify areas for cost reduction.

2.Weather modelling

Considering weather modelling, Markov Chains approach proved to be good with the added adjustments. It succeeded in reproducing weather series which resemble the actual records and preserve the major statistical properties of real weather. Nevertheless, if many years of historical records are available, one could look into the second-order Markov Chains which are expected to produce results matching closer with the reality.

3.Simulation

The following aspects are related to the set up of simulation:

- Constant duration of activities - within this thesis all operations were assumed to have a constant duration. This is not the case in the reality and often some arbitrary errors may lead to a different time spent on performing a certain action. It is suggested to introduce stochasticity in the durations of simulated activities. This leads to the following point.
- Time resolution of inputs - all activities within this thesis had a precision of one hour. The reason behind such a decision is the resolution of employed weather data. One can, however, use interpolation between available weather points and increase resolution to e.g. 30 minutes which would already allow to use better estimates of the durations accompanied with adding stochastic variability in them.
- Non-weather related delays - as it was mentioned in the validation study, in the real project a significant proportion of the delays is encountered due to the human-caused errors or machinery breakdown. It would be beneficial to introduce these concepts into the simulation, for example, based on the frequency of their occurrence known from the project records.

4.Optimization

Finally, some recommendations can be given in what concerns optimization of OWF installation:

- Metaheuristics algorithm - the selected PSO algorithm succeeded in finding optimum. The choice was done based on the fact that in the reviewed literature PSO performed well more often than other metaheuristics.
- PSO tuning - in order to keep the computational time limited, the number of generations was fixed to 10 and the number of particles to 3 times the number of problem dimensions. Other values may yield better results. On the other hand, a justified selection of values for inertia, cognitive and social coefficients can positively affect how quickly the swarm will converge. In this thesis, a relatively straightforward logic was followed which aimed at avoiding premature convergence to what can be a local optimum. The algorithm performed as intended and author recommends using iteration-dependent values for PSO parameters.
- Surrogate model - RBF method was selected based on the literature review. The selected model-based approach not only defined the family of functions to construct a surrogate, but also rules for steering search (based on (Gutmann, 2001)). Consequently, it was almost two times more efficient than the metaheuristic PSO. A different outcome might have been obtained if another search procedure was used.

Appendix A

Case Study

Appendix B

Optimization details

B.1 Full formulation of the objective function

Below is the full formulation of the objective function 6.3, which aims at obtaining total cost of an entire installation campaign as a result of a single simulation.

$$
\begin{gathered}
min(C(d_c, d_s, d_t, a, r)), \text{where} \\
C(d_c, d_s, d_t, a, r) = \sum_i^N C^i_{vessel} + C_{labor} + C_{harbor} + C_{pre-assembly} = \\
C_{\text{piles vessel per day}} \cdot c_p + C_{\text{jackets vessel per day}} \cdot c_j + C_{\text{turbines vessel per day}} \cdot c_t + \\
C_{\text{piles storage area}} \cdot c_p + C_{\text{jackets storage area per jacket}} \cdot a \cdot c_j + C_{\text{turbines storage area}} \cdot c_t + \\
r \cdot (C_{\text{jackets assembly fixed per line}} + C_{\text{assembly labor, machinery}} \cdot d_a) + \\
C_{\text{quay access piles vessel}} \cdot c_p + C_{\text{quay access jackets vessel}} \cdot c_j + C_{\text{quay access turbines vessel}} \cdot c_t + \\
C_{\text{labor, overheads, etc.}} \cdot d_p
\end{gathered}
\tag{B.1}
$$

where:
c_p - piles campaign duration,
c_j - jackets campaign duration,
c_t - turbines campaign duration,
a - harbor jackets storage capacity,
r - number of assembly lines,
d_a - assembly duration,
d_p - overall project duration

B.2 Optimization case study

Below, table B.1 has the full information about the OWF considered in optimization case study.

Table B.1: Optimization case study OWF. Full definition

General parameters	
Location	North Sea
Type of an installation	jackets with pre-piling and turbines
Number of turbines	50
Capacity of piles vessel	20 piles
Capacity of substructure vessel	4 jackets
Capacity of turbine vessel	6 turbines
Distance to site	70 km
Decision variables	
Campaign start day d_c	0-365
Substructure campaign start day d_s	0-365
Turbine start day d_t	0-365
Assembly lines number r	1-3
Jackets storage area a	5 - 15
Cost related parameters	
Turbine storage area (15 turbines)	150 000 m^2
Piles storage area	960 m^2
Jackets storage area (per jacket)	1764 m^2
Jackets assembly area (per line)	50 000 m^2
Labor number	25
Assembly labor number (per line)	32
Cranes 750 tones (per line)	2
Cranes 400 tones (per line)	2
SPMT (per line)	1
Quay length (substructure vessel)	216 m
Quay length (turbine vessel)	135 m
Quay length (piles vessel)	115 m

Appendix C

Jackets assembly

The main driver behind researching jackets pre-assembly options is to know how a jacket can be split up in modules so that standardization and cost savings are achieved. The reality is that a typical size of a jacket can reach up to 80 m in height and around 35 m in width, with the weight of about 900 tonnes. Furthermore, welding a jacket may take days and weeks to be done. As it can be seen the process is both time- and resource-consuming. When it comes to floating structures, their weight can often reach 5000 tonnes.Transporting such a large structures onshore causes major bottlenecks in the supply chain and sometimes is not even possible. Thus, even bigger need for the port assembly is in place. The drivers are to:

- Enable on-site assembly
- Enable delivery by road
- Limit welding operations in favor of bolting

Figure C.1 schematically presents how jacket assembly in port could be done.

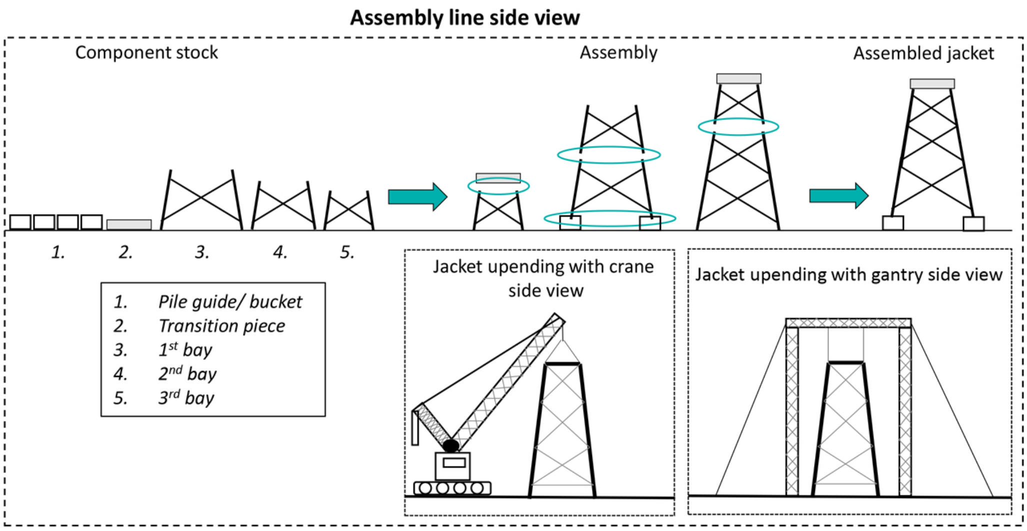


Figure C.1: Jackets assembly line

Offshore Innovation Department of SGRE has been investigating possible options to resolve existing issues. The following elaboration is mainly based on the internal discussions within SGRE and the assembly concept as described by Nielsen et al. (2018).

In this concept the assembly process consists of several major steps. Generally, the process could be split into operations requiring sophisticated and precise technologies, and less demanding operations. Some parts of a jacket unavoidably have to be welded, while others can be bolted. The idea would be to perform welding and coating in a controlled environment at the manufacturer site, while bolting can be done in port. A typical jacket would consist of 3 to 4 bays depending on its size, transition piece, and suction buckets or pile guides. All of these are manufactured elsewhere and then transported to a harbor for a final assembly via bolting.

In order to simulate an assembly process in the harbor, the total duration of the assembly is assumed to be 3.5 days per jacket per assembly line. This imitates an assembly of a standardized jacket designed for a water depth of $\sim 40m$. Such jacket usually consists of 3-4 bays stacked upon each other. Interfaces are bolted to facilitate faster build-up.

The most cost-intensive machinery includes two large cranes (lifting capacity of ca. 750 tonnes) needed to lift assembled jackets. Alternatively, a so-called gantry – a temporary portal frame for upending jackets, can be used. In this case two 750-tonnes cranes are still needed to build-up the gantry. Moreover, two smaller cranes (lifting capacity of ca. 400 tonnes) are constantly used for building up parts of a jacket. Self-propelling modular transporters (SPMTs) are used to move a jacket from one assembly area to another, and eventually to the quay side. Once assembled, jackets are loaded to the vessel by the vessel crane. Figure C.2 schematically shows how such an assembly in port can be arranged.

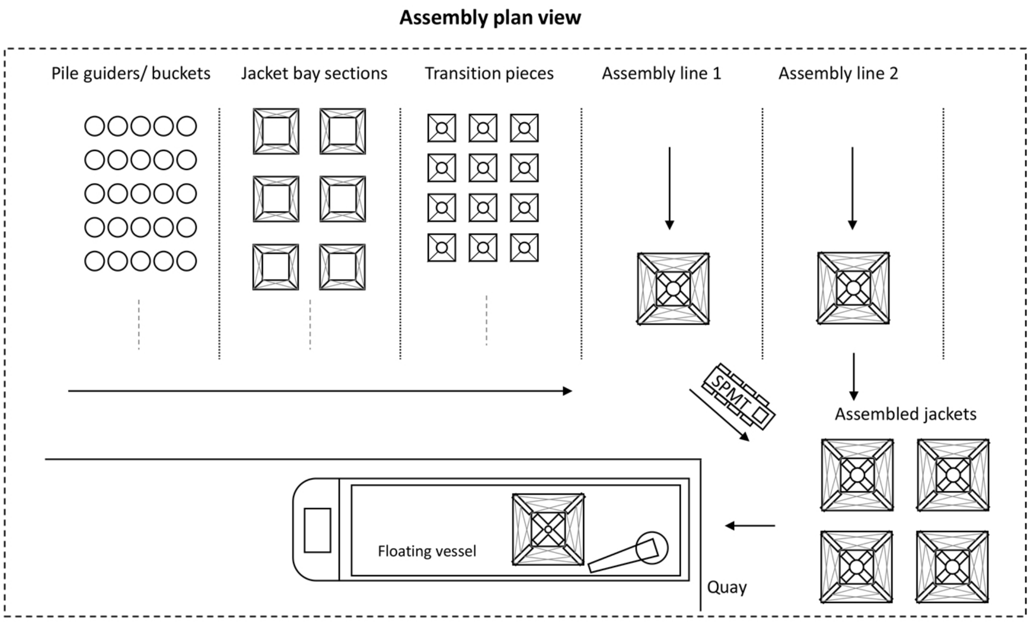


Figure C.2: Jackets assembly in port

An entire assembly process is represented in a very generic way consisting of 21 operations, each with a duration of 4 hours (yielding 3.5 days in total) and wind limit of 14 m/s, reflecting the fact that most of the operations involve lifting. Below is a list of resources and equipment per one assembly line:

- 50000 m^2 of area in port
- 32 workers including crane operators, technicians, supervisors, etc.
- 2 Cranes with the lifting capacity of $\sim$750 tonnes to lift large bays and full jacket.
- Alternatively, a gantry can be used for the same purposes
- 2 Cranes with the lifting capacity of $\sim$ 400 tonnes to lift smaller components.
- 1 or 2 18-axel SPMTs to move assembled parts from one station to another

Appendix D

Simulation sequence

Table D.1: Case study input data

	Activity	Duration [hr]
	Navigation in harbor	4
Loading	Backload	5
Loading	Load nacelle Load 3 blades Load tower	6
Loading	Repeat for 4 turbines	-
	Post-loading	4
	Travel loaded	14
	Jack up	6
Installation	Pre-installation Prepare crane Install tower Remove crane Prepare crane Install nacelle Remove crane Prepare crane Install blade Turn hub Repeat previous 2 steps Install blade Remove crane RNA commissioning Cable termination Energization MV cable commissioning	28
	Jack down	6
	Sail to the next turbine	3
	Repeat for all on board	-
	Travel back to port	12

Appendix E

Optimization results. Search space exploration.

E.1 PSO

Figure E.1 presents plots of objective function values for each decision variable.

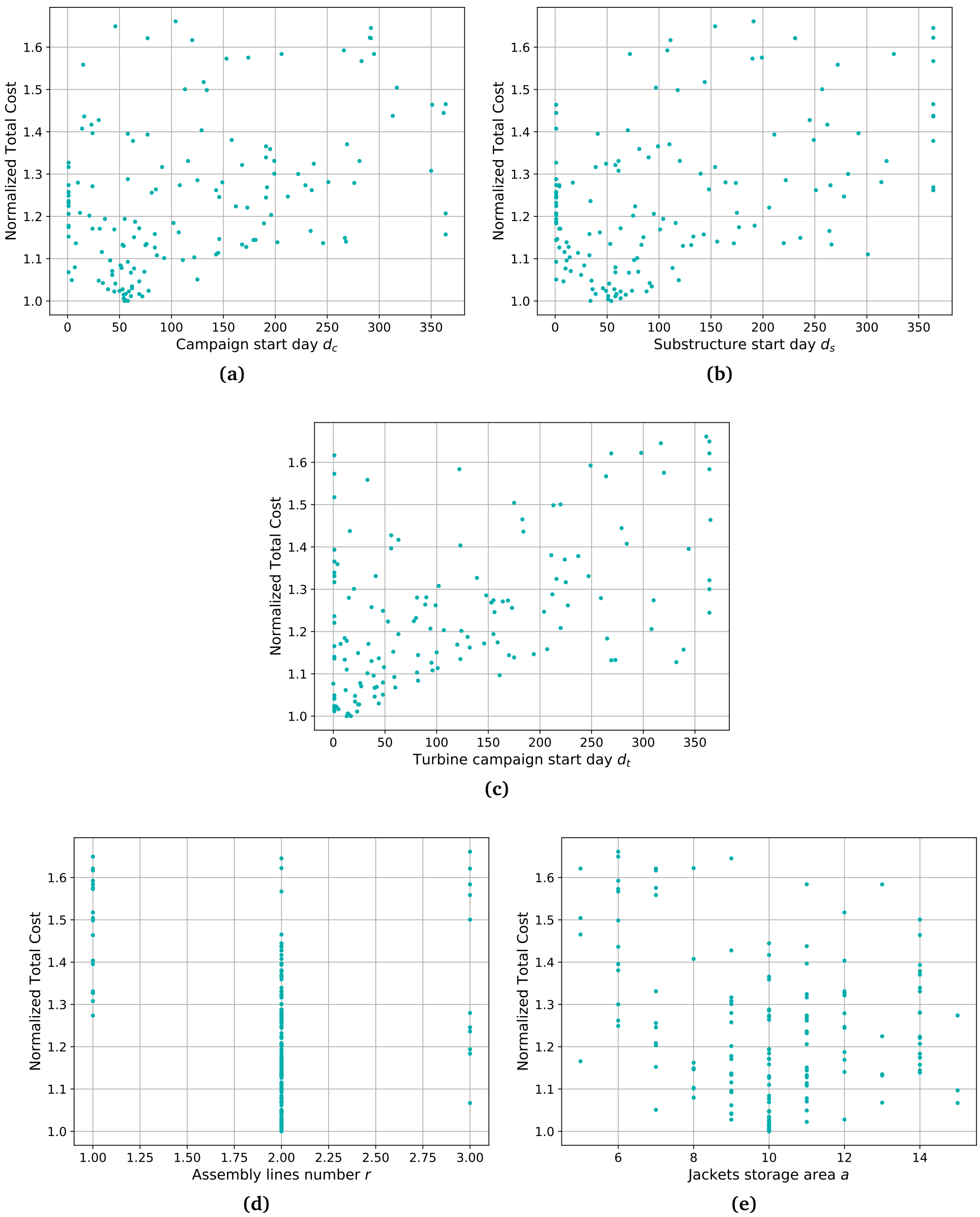


Figure E.1: PSO objective function vs decision variables

Below pairwise graphs are presented for each possible combination of decision variables. The color of particles reflects the objective value: purple means lower value, closer to optimum, and aquamarine means higher value, far from optimum.

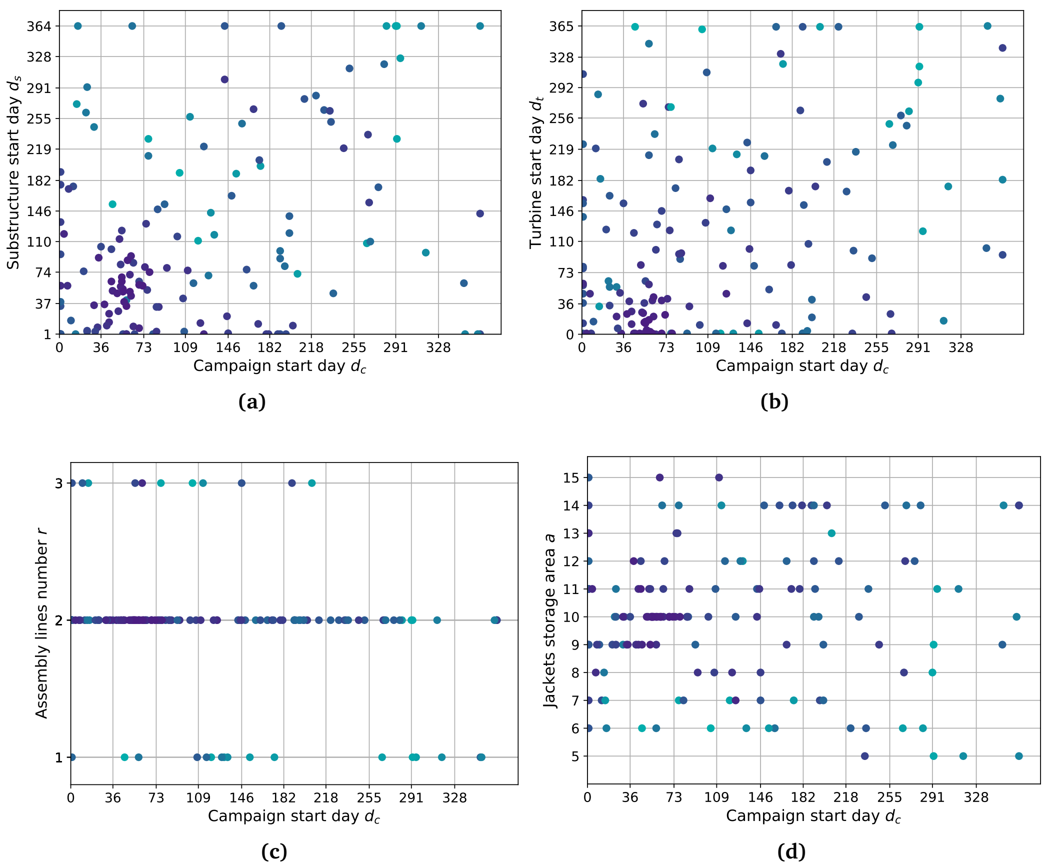


Figure E.2: PSO pairwise graphs for campaign start day d_c

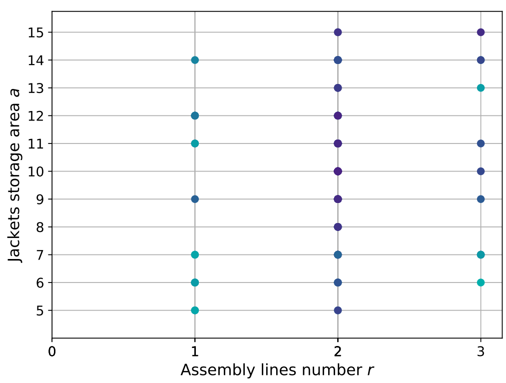
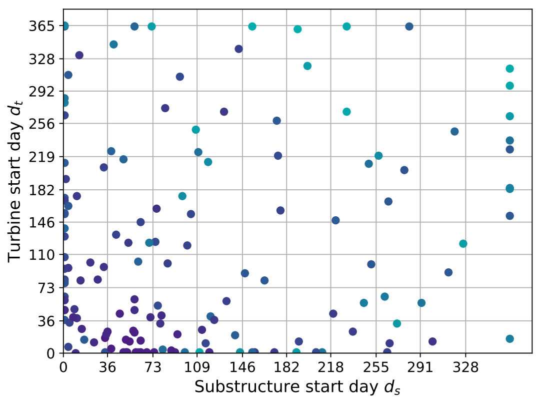
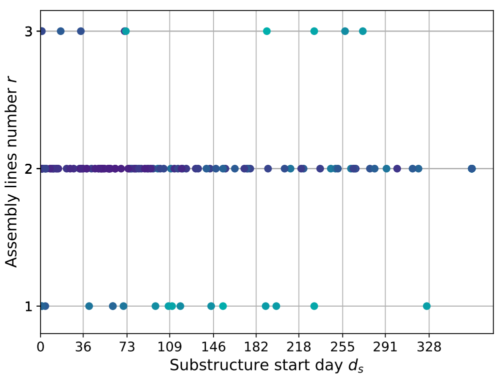


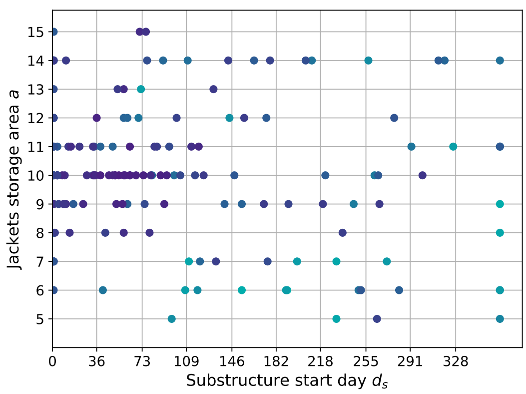
Figure E.3: PSO pairwise graph for assembly lines number r



(a)

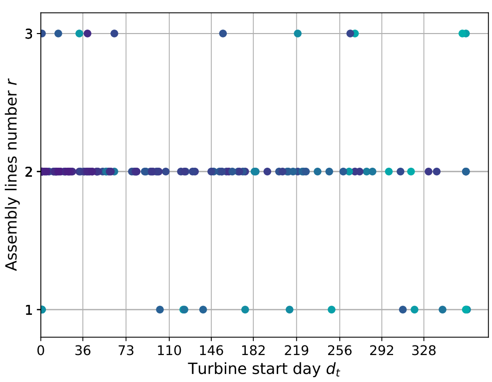


(b)

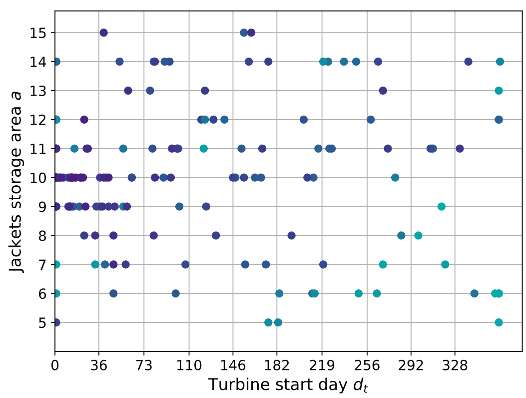


(c)

Figure E.4: PSO pairwise graphs for substructure installation start day d_s



(a)

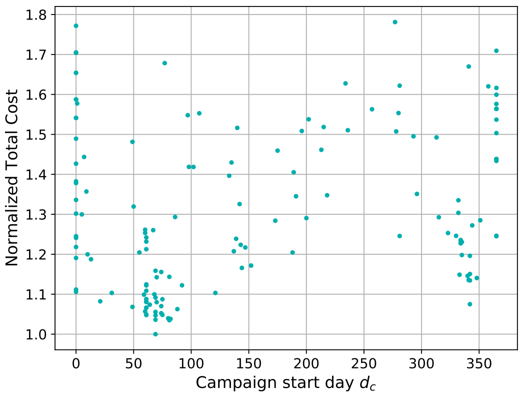


(b)

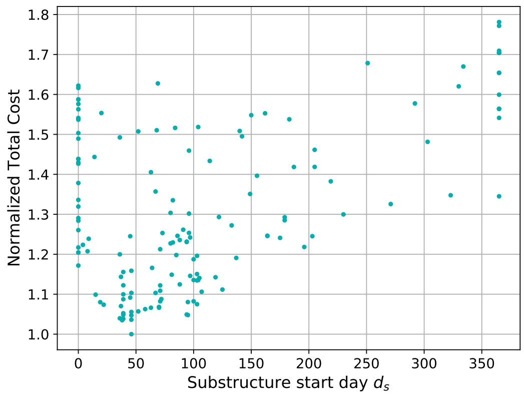
Figure E.5: PSO pairwise graphs for turbine installation start day d_t

E.2 RBF

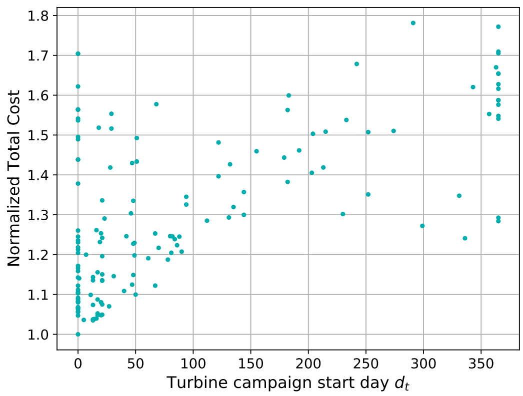
Figure E.6, similarly to E.1, shows plots of objective value as a function of each decision variable for RBF search.



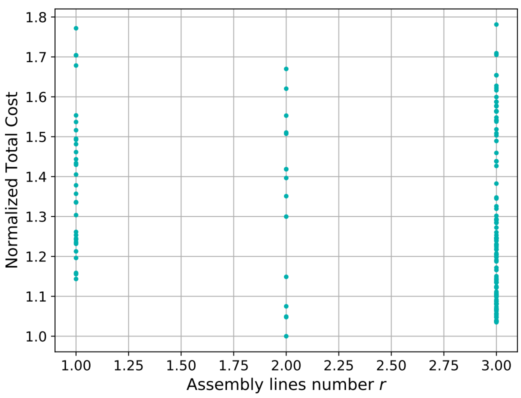
(a)



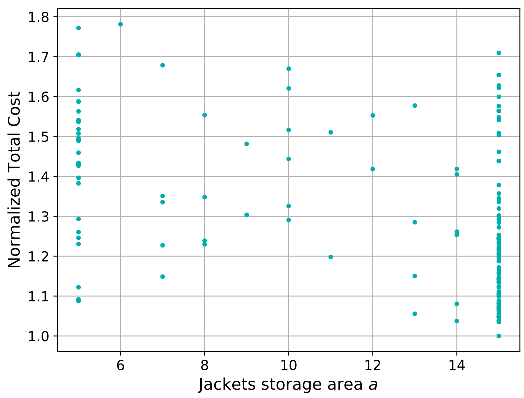
(b)



(c)



(d)



(e)

Figure E.6: RBF objective function vs decision variables

The rest of the figures are analogue to those presented for PSO search.

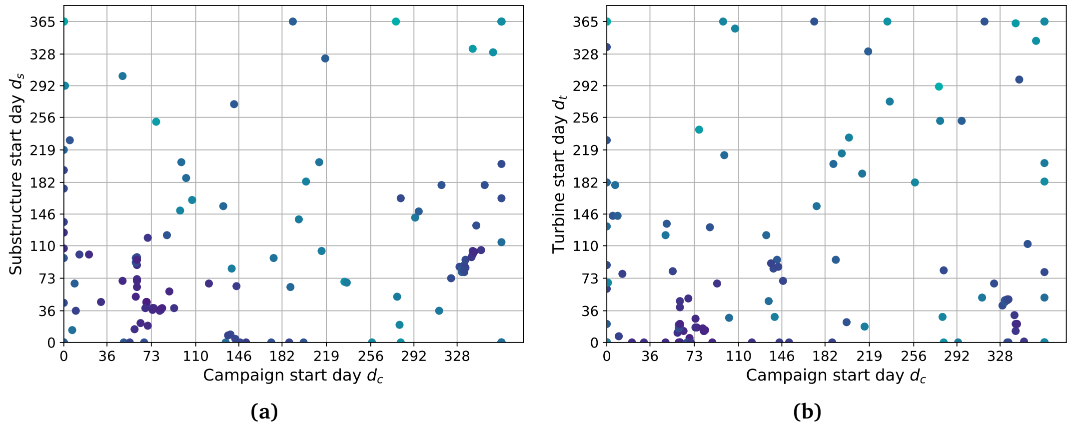


Figure E.7: RBF pairwise graphs for campaign start day d_c

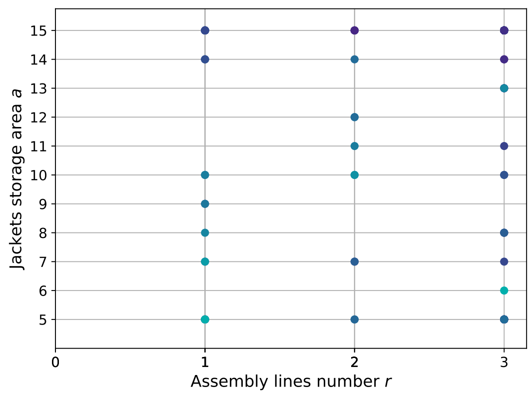
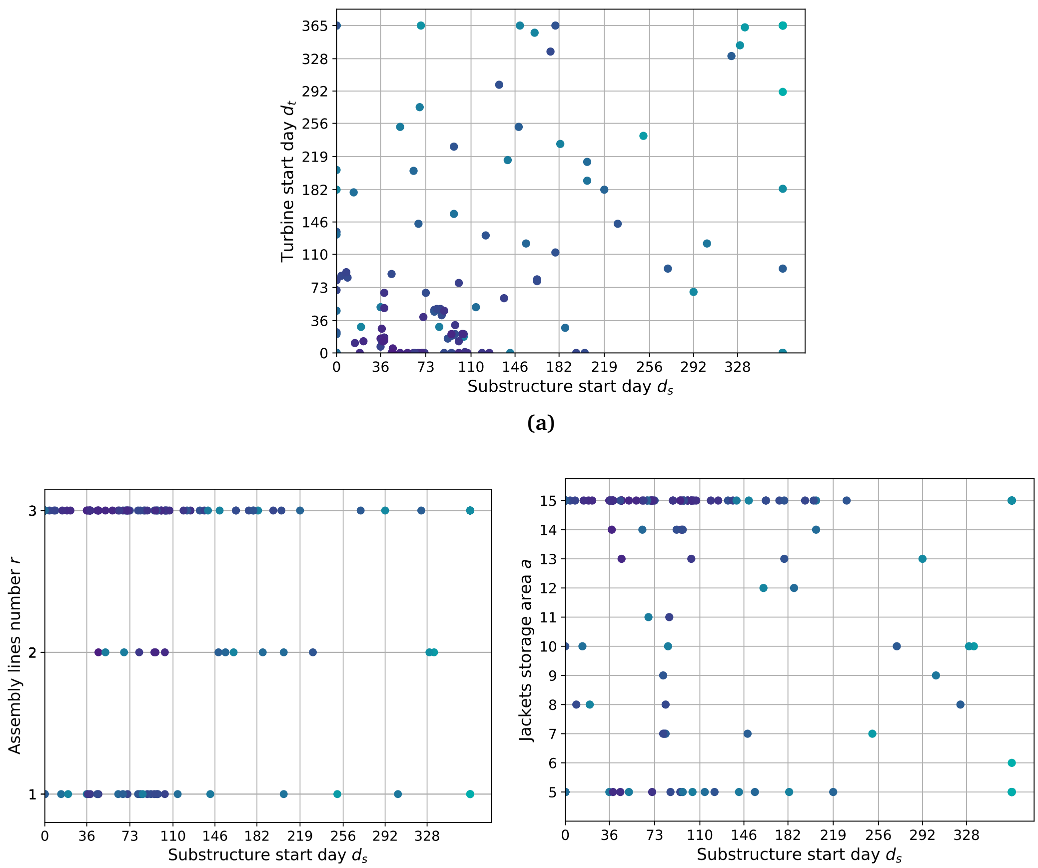


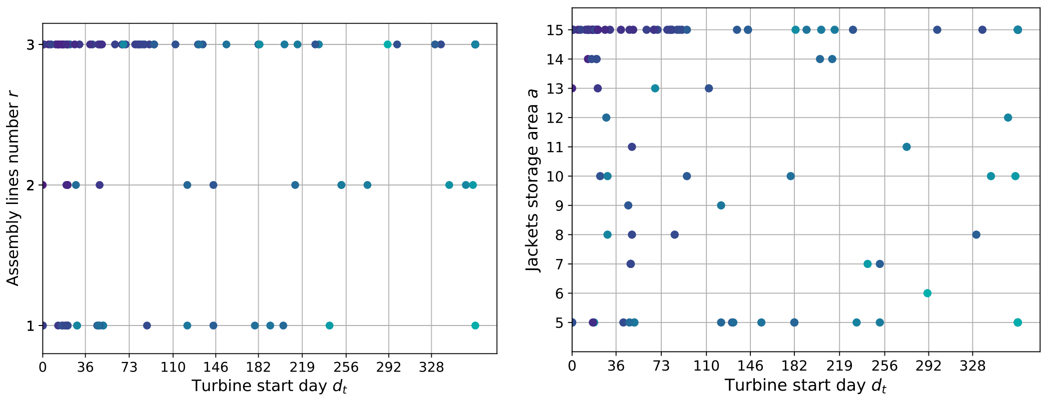
Figure E.8: RBF pairwise graph for assembly lines number r



(a)

(b) (c)

Figure E.9: RBF pairwise graphs for substructure installation start day d_s



(a) (b)

Figure E.10: RBF pairwise graphs for turbine installation start day d_t

Bibliography

Ait-Alla, A., Quandt, M., and Lütjen, M. (2013). Simulation-based aggregate installation planning of offshore wind farms. *International Journal Of Energy*, 7(2):23–30.

Banks, J. (1999). Discrete event simulation. *Resonance*, 18:78–86.

Barlow, E., Tezcaner Öztürk, D., Day, S., Boulougouris, E., Revie, M., and Akartunali, K. (2014a). A Support Tool for Assessing the Risks of Heavy Lift Vessel Logistics in the Installation of Offshore Wind Farms.

Barlow, E., Tezcaner Öztürk, D., Day, S., Boulougouris, E., Revie, M., and Akartunali, K. (2014b). An Assessment of Vessel Characteristics for the Installation of Offshore Wind Farms.

Barlow, E., Tezcaner Öztürk, D., Revie, M., Akartunalö, K., Day, A. H., and Boulougouris, E. (2015). Exploring the impact of innovative developments to the installation process for an offshore wind farm. *Ocean Engineering*, 109:623–634.

Barlow, E., Tezcaner Öztürk, D., Revie, M., Akartunalö, K., Day, A. H., and Boulougouris, E. (2018). A mixed-method optimisation and simulation framework for supporting logistical decisions during offshore wind farm installations. *European Journal of Operational Research*, 264(3):894–906.

Beinke, T., Ait Alla, A., and Freitag, M. (2017). Resource Sharing in the Logistics of the Offshore Wind Farm Installation Process based on a Simulation Study. *International Journal of e-Navigation and Maritime Economy*, 7:42–54.

Belotti, P., Bonami, P., D'Ambrosio, C., Forrest, J., Ladanyi, L., Laird, C., Lee, J., Margot, F., Vigerske, S., and Waechter, A. (2019). Bonmin (Basic Open-source Nonlinear Mixed Integer programming). https://projects.coin-or.org/Bonmin. Accessed: 2019-04-22.

Castro, P. M., Aguirre, A. M., Zeballos, L. J., , and Mendez, C. A. (2011). Hybrid Mathematical Programming Discrete-Event Simulation Approach for Large-Scale Scheduling Problems. *Industrial and Engineering Chemistry Research*.

Chrisman, L. (2014). Latin Hypercube vs. Monte Carlo Sampling. http://www.lumina.com/blog/latin-hypercube-vs.-monte-carlo-sampling. Accessed: 2019-04-04.

Collet, P. (2006). Stochastic Optimization Algorithms.

Conn, A., Scheinberg, K., and Vicente, L. (2009). Introduction to Derivative-Free optimization.

Costa, A. and Nannicini, G. (2018). RBFOpt: an open-source library for black-box optimization with costly function evaluations. Mathematical Programming Computation. http://www.optimization-online.org/DB_HTML/2014/09/4538.html. Accessed: 2019-04-02.

ECN part of TNO (2015). Commercial Proof of Innovative Offshore Wind Installation Concepts using ECN Install Tool.

ECN part of TNO (2018). Modeling the Installation of Offshore Wind Farms. https://www.ecn.nl/publications/PdfFetch.aspx?nr=ECN-L--16-018. Accessed: 2019-02-05.

Engelen, C. (2015). The nonlinear effect of combining uncertainties on the energy yield of an offshore wind farm. Master's thesis, TU Delft.

Equinor (2018). The market outlook for floating offshore wind. https://www.equinor.com/en/what-we-do/hywind-where-the-wind-takes-us/the-market-outlook-for-floating-offshore-wind.html. Accessed: 2019-02-01.

Fang, H. and Horstemeyer, M. (2005). Global response approximation with radial basis functions.

Fourie, P. and Groenwold, A. (2002). The particle swarm optimization algorithm in size and shape optimization.

GE Renewable Energy (2018). HALIADE-X OFFSHORE WIND TURBINE PLATFORM. https://www.ge.com/renewableenergy/wind-energy/turbines/haliade-x-offshore-turbine. Accessed: 2019-02-01.

Gutmann, H. (2001). A Radial Basis Function Method for Global Optimization.

GWEC (2018). GWEC Global Wind Report.

Hansen, K. and Siljan, O. M. (2017). Optimizing the vessel fleet used to install an offshore wind farm. Master's thesis, NTNU.

Hassan, R., Cohanim, B., and de Weck, O. (2004). A Comparison of Particle Swarm Optimization and the Genetic Algorithm.

Holmström, K., Quittineh, N.-H., and Edvall, M. (2008). An Adaptive Radial Basis Algorithm (ARBF) for Expensive Black-Box Mixed-Integer Constrained Global Optimization.

IRENA (2016a). Innovation Outlook Offshore Wind.

IRENA (2016b). Power to Change.

Jin, R., Chen, W., and Simpson, T. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization.

Kerkhove, L. and Vanhoucke, M. (2016). Optimised scheduling for weather sensitive offshore construction projects.

Kumar, V. (2017). Optimization of offshore wind farm installation procedure with a targeted finish date. Master's thesis, TU Delft.

Lange, K., Rinne, A., and Haasis, H.-D. (2012). Planning Maritime Logistics Concepts for Offshore Wind Farms: A Newly Developed Decision Support System. 18:142–158.

Laskari, E., Parsopoulos, K., and Vrahatis, M. (2002). Particle Swarm Optimization for Integer Programming.

LEANWIND (2018). Driving Cost Reductions in Offshore Wind. The LEANWIND project final publication.

Lee, A. (2015). PYSWARM - Particle swarm optimization (PSO) with constraint support. https://pypi.org/project/pyswarm/. Accessed: 2019-04-02.

Lütjen, M. and Karimi, H. R. (2012). Approach of a Port Inventory Control System for the Offshore Installation of Wind Turbines. *Isope*, 4(January 2012):502–508.

Maljaars, N. (2017). Support structure optimization. on the use of load estimations for time efficient optimization of monopile support structures of offshore wind turbines. Master's thesis, TU Delft.

Matloff, N. (2008). Introduction to Discrete-Event Simulation and the SimPy Language.

Muhabie, Y. T., Caprace, J.-D., Petcu, C., and Rigo, P. (2015). Improving the Installation of Offshore Wind Farms by the use of Discrete Event Simulation. *World Maritime Technology Conference (WMTC), At ,Providence, RI, USA*, pages 1–10.

Nance, R. E. (1993). A history of discrete event simulation programming languages.

Nfaoui, H., Essiarab, H., and Sayigh, A. (2003). A stochastic Markov chain model for simulating wind speed time series at Tangiers, Morocco.

Nielsen, A., Krog, D., Reilev, J., Serup, M., and Poulsen, S. (2018). Development of a Second-Generation Product and Production Setup for Modular Jacket Offshore Foundations (Confidential). https://projects.coin-or.org/Bonmin.

Papaefthymiou, G. and Klöckl, B. (2008). MCMC for Wind Power Simulation.

PwC (2018). Unlocking Europe's offshore wind potential. Moving towards a subsidy free industry.

Rios, L. and Sahinidis, N. (2012). Derivative-free optimization: a review of algorithms and comparison of software implementations.

Ruparelia, N. B. (2010). Software Development Lifecycle Models.

Scholz-Reiter, B., Heger, J., Lütjen, M., and Schweizer, A. (2010). Planning and control of logistics for offshore wind farms.

Scholz-Reiter, B., Heger, J., Lütjen, M., and Schweizer, A. (2011a). A MILP for Installation Scheduling of Offshore WindFarms. 5(7):371–378.

Scholz-Reiter, B., Heger, J., Lütjen, M., and Schweizer, A. (2011b). Towards a Heuristic for Scheduling Offshore Installation Processes.

Sharda, B. and Vazquez, A. (2009). Evaluating Capacity and Expansion Opportunities at Tank Farm: a Decision Support System using Discrete Event Simulation.

Shin, S. Y., Brun, Y., Balasubramanian, H., Henneman, P. L., and Osterweil, L. J. (2018). Discrete-Event Simulation and Integer Linear Programming for Constraint-Aware Resource Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9):1578–1593.

Siemens Gamesa (2018). Siemens Gamesa launches 10 MW offshore wind turbine. https://www.siemensgamesa.com/en-int/newsroom/2019/01/new-siemens-gamesa-10-mw-offshore-wind-turbine-sg-10-0-193-dd. Accessed: 2019-02-01.

SimPy (2018). SimPy API reference. https://simpy.readthedocs.io/en/latest/api_reference/index.html. Accessed: 2019-02-08.

Tekle Muhabie, Y., Rigo, P., Cepeda, M., de Almeida D'Agosto, M., and Caprace, J. D. (2018). A discrete-event simulation approach to evaluate the effect of stochastic parameters on offshore wind farms assembly strategies. 149(December 2017):279–290.

Terrazas-Moreno, S., Grossmann, I., and Wassick, J. (2012). A Mixed-Integer Linear Programming Model for Optimizing the Scheduling and Assignment of Tank Farm Operations.

Tezcaner Öztürk, D., Barlow, E., Akartunali, K., Revie, M., Day, A., and Boulougouris, E. (2017). Robust Rolling Horizon Optimisation Model for Offshore Wind Farm Installation Logistics.

Vis, I. F. and Ursavas, E. (2016). Assessment approaches to logistics for offshore wind energy installation. *Sustainable Energy Technologies and Assessments*, 14:80–91.

WindEurope (2018). Breaking new ground. Wind Energy and the Electrification of Europe's Energy System.

WindEurope (2019). Offshore Wind in Europe. Key trends and statistics 2018.

Xiao, H., Pei, W., Z., D., Kong, L., and Wang, D. (2018). Application and Comparison of Metaheuristic and New Metamodel Based Global Optimization Methods to the Optimal Operation of Active Distribution Networks.

TU Delft Delft University of Technology

SIEMENS Gamesa RENEWABLE ENERGY