Simulation and Optimization of Offshore Renewable Energy Arrays for Minimal Life-Cycle Costs

Second Year Report

Robin Kuipers

Supervisors:

Kerem Akartunali

Euan Barlow

University of Strathclyde

Strathclyde Business School

Glasgow, Scotland

October, 2020

Abstract

This report aims to give a detailed overview of the research progress I made between January 2020 and September 2020, which is the time between the completion of the first yearly review and the start of the second yearly review of this PhD project into logistical decisions regarding offshore wind farms. It will quickly recap the progress made in the first year, and then detail the progress made in the second year, which primarily focused on developing and implementing models for optimization problems related to topic. The report also discusses future work, and what the next steps are.

Contents

1   Introduction

For the past two years, I have been researching logistics related to offshore windfarm projects; the installation, maintenance, and decommissioning of windfarms in various seas and oceans, primarily the North Sea. The installation and decommissioning projects (at the start and end of the windfarm's lifespan respectively) can often take up to several years, and the lifespan of such farms is usually between 20 and 30 years, over which maintenance has to be done. For these projects, expensive vessels have to be used, the rent of which is often upwards of £100.000 per day (Barlow et al. 2014), hence a project with multiple vessels over many months can cost upwards of £100 million (Kaiser & Snyder 2010). Therefore, even small improvements to the schedules can save significant amounts of money.

The complexity with these logistics comes from various factors, the first being the severe impact that weather conditions can have on when operational tasks can be completed. These projects take place on open sea, where weather can often be rougher than on land. In addition, the high-tech vessels are performing operations on large industrial constructions, hence there is a limited range of allowed wind speeds and wave heights. Another factor which can further limit possible schedules is the inflexibility involved in vessel chartering (renting), since vessels of the required caliber cannot be chartered on short notice, making adaptive in-the-moment scheduling is impossible. This means we will have to make decisions significantly in advance, and run the risk of chartering vessels in periods where they might not be able to complete their tasks due to the weather conditions.

This report will detail the research progress I have made since the completion of my first yearly review, which (due to scheduling issues and delays) was only completed in January 2020. Further on in this section, I will recap the specific parts of this problem that I am looking at (Section 1.1), and which research questions I initially posed (Section 1.2). Then Section 2 through Section 5 will talk about the work I have been doing since the last

review. Respectively, these sections are about the gaps in my own knowledge that I closed, the models I have developed, how I have implemented them, my efforts to acquire data, and all additional activities I have taken part in. Finally, Section 7 will discuss the next steps for this project.

1.1 Problem description

This research looks at offshore windfarms (OWFs) and the scheduling decisions made over the entire life-cycle of such an OWF. I will focus on the type of windfarms currently being build in the North Sea (Renewable-UK 2017, Barlow et al. 2018). These windfarms generally consist of up to 150 Wind Turbine Generators (WTGs) and the farms we focus on are located in roughly 50+ kilometers off the coast. The specific duration of each phase of the life-cycle depends on the specific windfarm, but to give the reader some idea of the timescales involved: A typical life-cycle would consist of roughly 2-3 yeas of installation, 20-30 years in which the WTGs generate energy and need to be maintained, and then another 2-3 years in which the turbines are decommissioned. These turbines are large structures that will need to be transported and installed by specialized and expensive vessels, leading to large costs over these years.

In the literature these three phases are commonly treated separately. Both the installation and decommission phases have similar structures; a fixed set of tasks that needs to be completed as quickly or as profitably as possible. Focusing on cost over speed might lead to a different schedule, as some decisions might slow down the overall project but lead to individual turbines being operational earlier, from which point it can start generating energy and income. Commonly there will be contractual or legal deadlines on installation and decommission, setting dates at which the OWF should be fully or partially operational, or completely decommissioned. A key difference between the installation and decommission phase are that in the installation phase minimizing cost will help reach the deadlines; ideally the installation is

completed as soon as possible, as income will be generated by each completed turbine. For decommission this is not the case; if decommission starts later turbines will generate income for longer, but it will be harder to reach the deadlines.

The maintenance phase has an entirely different structure from the other two phases. There will be a small set of fixed tasks, since there are legal requirements on minimum amount of maintenance that needs to be done, but this will not be the bulk of the work performed during this phase. A maintenance strategy will need to be formed that determines when turbines will be visited. These visits can be at predetermined moments, or after a turbine fails, or sometimes even right before a turbine is predicted to fail (based on sensor data). An optimal strategy will often use a combination of these types of visits in order to minimize cost, accounting for both the cost of repairs and the amount of income missed due to failures and downtime during maintenance. An additional difference during the maintenance phase is that tasks will often be smaller than the tasks in the other phases. During installation and decommission entire turbines need to be transported and worked on, often requiring very specific and expensive vessels. Conversely during maintenance many smaller repairs can often be done by simply sending engineers over and all you need is a crew transport vessel. This is not the case for every maintenance task as larger repairs and replacements will sometimes need to happen, but it will still hold for many small tasks.

Each of these three phases is subject to stochastic elements, the weather conditions being a major factor to take into account. In addition to the weather possibly restricting which tasks can be performed at any given time, it plays another factor during the maintenance phase. Strong winds also increase the energy output of the turbines, making it extra beneficial to have all turbines up and running before periods of strong winds. Other than the weather, task durations have an inherently uncertain factor, and turbine failures are also difficult to predict. All this together means that uncertain factors can have an immense impact on the overall costs of the project. For

this reason robustness is often a metric to be taken into account as well. It might be in the operators best interest to work a less intensive schedule if it reduces the chances of large delays.

1.2 Research questions

In this PhD project my goal is to look at scheduling during the entire life-cycle of an OWF. As far as we are aware, this has not been done in the literature, which indicated there may be optimizations and new insights to be found here. Therefore my primary research question is:

Question. *Can considering the entirety of the life-cycle of an Offshore Wind Farm, and how each of the phases interact, improve logistical decision making on these projects?*

It is clear that a primary reason for splitting the project into phases is that the problem becomes more manageable, and there is generally no need to make all scheduling decisions at the start of the project. There is no point in scheduling the entire decommission phase at the time of installation, as over the 20-30 year lifespan of the windfarm the available vessels will likely change. However, treating the phases entirely separately misses the interactions between the phases. This interaction exists within the real world, and if the literature ignores it this creates a divide between academia and the real world. For that reason I want to investigate these interactions.

The first type of interaction takes place when two phases are active at the same time; after the first turbines have completed installation they potentially need maintenance, while installation continues on the rest of the turbines. The reverse effect takes place when decommission starts, as it does not start at the same time for every turbine. During this time, the phases share resources and could potentially hinder each other (when the same port is used for different vessels servicing different phases). On the other hand, if attention is payed this sharing of resources could be beneficial. Some installation tasks require smaller vessels that can also be used for maintenance

operations, and some maintenance operations use larger vessels more commonly used for installation tasks. Therefore paying attention to this overlap period could both help reduce obstacles and create new benefits from this interaction.

The second type of interaction is the long-term effect scheduling decisions might have. If the installation is looked at in isolation a schedule might be produced in which the completion time of the first and the last turbine is years apart; this might have an affect on their wear and chance to fail over the course of the maintenance phase. This in turn might also lead to those first-installed turbines being decommissioned first as well. Since decisions made during the installation phase might still influence events long after installation is complete, these long term effects might also influence the decision in the first place. For that reason this interaction can be looked at from two perspectives; early decisions that are influenced by their long-term effects, and later decisions that are influenced by decisions made in earlier phases.

These interactions bring me to the sub-questions of my research:

Sub-Question 1. *Can considering how phases in the life-cycle of a windfarm overlap and share resources improve logistical decision making on these projects?*

Sub-Question 2. *Can simulating the entire life-cycle of a windfarm provide useful data to base logistical decisions on in the later phases of these projects?*

Sub-Question 3. *Can considering the long-term effects of logistical decisions early on in the life-cycle of a windfarm improve these decisions?*

2 Closing learning gaps

During my first yearly review it was remarked that most of my reading up to that point focused on either methodology or installation projects, and there were gaps concerning the other two phases, primarily maintenance. Over the past months I have worked to close these gaps, and I consistently updated the literature review as I gained knowledge. This section will show the highlights of that updated literature review, and I will give some closing thoughts on these new insights at the end.

2.1 The maintenance phase

In (Shafiee 2015) the different topics that fall under maintenance are divided into three timescales (echelons):

1. Strategic: Long term, over the lifespan of an OWF

   - Wind farm design for reliability
   - Location/Capacity of maintenance accommodations
   - Maintenance strategy selection
   - Outsourcing decisions

2. Tactical: Medium term, between 1 and 5 years

   - Spare parts management
   - Maintenance support organization
   - Purchasing/leasing decisions

3. Operational: Short term, day to day

   - Maintenance scheduling
   - Routing of maintenance vessels
   - Performance measurement

Decisions within the strategic echelon are found to have the biggest impact on the costs of operations, which makes sense as they affect the largest amount of time. However each of these echelons include important decisions that will have to be made for the maintenance of the sites. Besides providing this framework to categorize research, they also highlight which areas appear most thoroughly researched, and identify some unexplored areas. Potentially due to their large impact, the topics within the strategic echelon have received most attention, while the topics in the operational echelon received least. The writers indicate they these shorter term areas might hold unexplored potential for future research.

For my research the most relevant topics are Maintenance strategy selection and maintenance scheduling. These are categorized in opposite echelons, but considering my research focus this makes sense. I am looking at reducing lifetime costs, which clearly falls in the strategic echelon. Specific ideas related to this are considering the long-term impact of decisions and considering historical data from the windfarm. However in order to reduce these costs I am also looking at specific stretches of time within the lifetime, such as when the maintenance phase overlaps with the installation or decommission phases, and specific small-scale decisions made within these stretches of time. This focus clearly falls within the operational echelon, which means different aspects of my research fall within different echelons.

In (Dinwoodie et al. 2012) an attempt is made to analyse the availability of OWFs. It is observed that the availability of offshore farms is much lower than that of onshore farms; currently offshore availability is around 80%, while onshore this is 97%. Various future scenarios are simulated to investigate which measures are likely to improve availability. It is found that increasing vessel operability to allow a higher significant wave height will greatly increase availablity, but realistically it will remain significantly below onshore availability. It is posited this can only be improved with improvement of the components to lower their failure rates, but an 8-fold improvement is required to achieve the same level of availability as onshore.

In another paper (Dinwoodie & McMillan 2014) the same researchers focus on comparing operational strategies. They consider 4 approaches:

- Annual charter, where maintenance operations take place at predetermined times

- Fix on fail, where repairs are made as soon as any failures are detected

- Batch repair, where repairs are done after a fixed number of failures is detected

- Purchase, where a vessel is purchased rather than rented

In their research they find the latter three strategies to be close together in costs while the Annual charter strategy lags behind. Which of these three strategies performs best depends on scenario specifics such as fluctuations in vessel costs, the value of energy, and specific failure rates. Each of the strategies also has inherent characteristics that the operator might have preferences in, such as the purchase strategy having much higher costs up front, but lower operational costs.

An issue with this paper is that mixing of strategies is only very briefly discussed, while this is often beneficial and commonly discussed in the literature. For example, in batch repair, if the batch size is set to 5 failures and for an extended period of time there are 4 failures, it might be beneficial to repair it if this situation goes on for long enough. One way to do this is to mix the annual charter strategy with the batch repair strategy, having a set time to do maintenance even if the required number of failures is not reached. Since this combination of preventive and corrective maintenance is a common strategy within the literature its absence in this paper is surprising.

In (Dinwoodie et al. 2015) this same group of researchers proposes a reference case for maintenance models. Their idea is for various maintenance models to optimise a constructed scenario a fictional windfarm, as a benchmark to compare these models. This allows us to see which models perform

better in various scenarios, as they constructed a base case and a set of variants. In the paper they compare four models, and compare various metrics of the results. For availability of the windfarms all results are extremely close, apart from three scenarios. It is explained that these differences result from previously unknown assumptions in the different models, such as whether parallel maintenance tasks are possible on the same turbine, and the moment at which tasks should be assigned to vessels.

This uncovering of assumptions is a strong benefit of using a standardised reference case to verify a model. Seeing that a new model performs similarly to other models apart from certain scenarios can give insight into how this new model handles these scenarios in ways that might otherwise have gone unnoticed, as demonstrated in this paper.

In (Besnard et al. 2011) a stochasitc optimisation model is constructed for short-term maintenance planning. The primary goal is to utilize times when energy production is expected to be low for maintenance, in order to minimise production losses. An interesting aspect of this model is the build-in uncertainty of weather forecasts. The stochastic element of the model is realized by having a small set of scenarios, and the average cost over all scenarios is minimised. The first time step is assumed to have the correct forecast, hence the idea behind the model is for the optimisation to be recomputed regularly. A rolling horizon approach such as this seems like a good method to deal with these uncertain forecasts, although it can be computationally expensive to recalculate regularly.

Another optimisation model focused on maintenance is given in (Stålhane et al. 2019), where the problem of fleet composition is investigated through a two-stage stochastic programming model. The problem is decomposed using a variant of a Dantzig-Wolfe decomposition. In the first stage the set of vessels to be chartered for each active base is decided, while in the second stage the maintenance tasks are divided over the available vessels. The first stage has an objective function minimising costs, which has one term

representing the expected costs from the second stage. This cost depends on the stochastic variables in the second stage (failures, energy generated based on weather conditions, availability of vessels based on weather conditions). In order to solve it, a set of scenarios is generated, each of which represent a realisation of this set of variables. The weighted average costs of all scenarios (weighted on their probability) is used to represent the expected costs, after which the first stage of the problem can be solved.

This sort of decomposition is potentially useful for my models, to handle the stochastic aspects of the problem. In this paper the model is verified by comparing it to reference cases constructed in (Dinwoodie et al. 2015) which was discussed earlier in this section. The computations are shown to run in feasible time, and when the fleet composition is fixed it is shown to give comparable results to what is expected.

2.2   Other new knowledge

While there is not much research relating specifically to decommission projects, some recent work has still been done in this field. In (Irawan et al. 2019) an attempt to optimise the decommissioning is made using an Integer Linear Program. The model created focuses on minimising total costs, and takes many factors into account, such as both offshore and onshore logistics, components that can be sold and components that can be recycled. This broadness means the model is more accurate, but it also means they were not able to solve the exact problem. For this reason they tried various relaxations, and proposed a matheuristic method in which some integer constraints are relaxed. They use this method to create feasible and well-performing solutions in a short time.

In the last two decades the the installation time has generally decreased, as analysed in (Lacal-Arántegui et al. 2018). This work looks at the installation times of the foundations of a windfarm, the installation times of the turbines, and the overall installation times. It observes a 22% time decrease

per turbine between the periods of 2000 to 2003, and 2016 to 2017. This improvement is more impressive when you consider the distance from the shore has increased in that time. However, the true scale of the improvement only becomes clear when you look at the time per megawatt; this reveals a 71% time reduction within this timeframe. This is a very substantial reduction, and one that shows the potential of large windfarms with many turbines that are located far off the shore. The specific causes of this improvement are not investigated much, so it is unclear whether this improvement mainly comes from the components used, or the logistics of installation, or some other cause. Investigating this is noted as future research.

An overview of the state of robust optimisation is given in (Gabrel et al. 2014). This study looks at 130 papers, 45 PhD dissertations, and selected other works published between 2007 and 2013 which discussed robust optimisation. The sheer amount of research done reflects the strength of this method. Both the theory and applications of robust optimisation are discussed. The applications include classical logistics problems such as inventory management and scheduling, problems related to finance and revenue management, queueing networks, energy systems and public good (decisions that will benefit the general public). This shows the wide variety of possible applications that RO methods can have. They conclude by highlighting four key developments in RO between 2007 and 2013; (1) the extensive amount of research regarding robustifying stochastic optimization, (2) a link between uncertainty sets and risk theory allows for a connection with decision sciences, (3) new results in the area of sequential decision-making and multi-stage models, and (4) new areas of applications.

2.3   Closing thoughts

As is hopefully evident from this section, I have worked hard to improve my knowledge where it was lacking before. A key insight I gained from the maintenance papers discussed is how different this phase is, not only in the

work that need to be done, but also from a modeling perspective. When scheduling the installation phase which will take roughly 3 years you could (roughly) plan which tasks are performed in which weeks, for the entire project. And each week the project will get closer to completion. With the maintenance phase spanning 20-30 years this approach is infeasible, and there is no one goal that you progressively move closer to. Instead it has a more cyclic nature where you aim to mitigate the degradation of the windfarm over time, and the tasks that need to be performed cannot be known *a priory*. Therefore smaller timescales, such as rolling horizons covering weeks rather than years, are much more appropriate for this phase. This is an insight that needs to be considered in a model aiming to combine this phase with the other phases.

3   Developing models

In the past year I have started developing optimisation models for various scheduling decisions that come up during an OWFs life-cycle. I developed the theory of these models, and implemented and tested them with simple test-cases. This implementation will be discussed in Section 4 and this section will talk about the models and the process of creating and improving them.

I started with very simple models that had a small scope and many assumptions, and progressively added complexity while relaxing the assumptions. This provided me with practice and insights in optimization modeling, a technique I had previously extensively read about but not used myself on any large scale. Therefore this start with simple models was needed, and this approach has made me learn a lot about the nuances in creating these models.

All models I created are Mixed-Integer Linear Programming (MILP) models, meaning the variables can both be continuous or integer variables, and every variable has a linear relation with the objective and the constraints (i.e. no variables are multiplied by other variables). This is the type of models most commonly used in the scheduling literature I have come across, indicating they are generally well-suited for these this type of problem.

It is also worth noting that each of the models shown in this section represents a stage in my development as a researcher and as someone designing model. For that reason the models shown, especially the early ones, can likely be refined further. Since this report is about my work and progress I opted to not to go back to my older models and refine them extensively with the experience I gained since making them originally.

3.1   Deterministic models

The first models I worked on were three simple models, one for each of the three phases. My idea behind this was to create a basis for each of the phases, which I could then combine into models where multiple phases are

considered. To give the reader an idea of how these models work I will show some of the models I created.

3.1.1 Installation model

I first started work on an installation model. This was an iterative development process where the first handful of models had logical contractions or were infeasible for different reasons, and the first feasible models turned out to be missing some crucial restrictions and gave models that would be impossible in practice. The first model that gave potentially possible results, albeit under heavy assumptions, is shown here:

$$\max_{\substack{O_p, N_{r,p} \in \mathbb{Z}^* \\ s_{a,i,t} \in \{0,1\}}} \sum_{p \in P} [O_p \cdot v_p - \sum_{r \in R} N_{r,p} \cdot C_{r,p}] \tag{1}$$

subject to:

$$s_{a,i,t} \leq s_{a,i,(t+1)} \qquad\qquad \forall a \in A, \forall i \in I, \forall t \in T \tag{2}$$

$$s_{a,i,(\omega_{i,t_N})} = 1 \qquad\qquad \forall a \in A, \forall i \in I \tag{3}$$

$$s_{a,j,t} \leq s_{a,i,(\omega_{i,t})} \qquad\qquad \forall a \in A, \forall (i,j) \in IP, \forall t \in T \tag{4}$$

$$N_{r,p} \geq \sum_{a \in A} \sum_{i \in I} (\rho_{i,r} \cdot (s_{a,i,t} - s_{a,i,(\omega_{i,t})})) \qquad \forall r \in R, \forall p \in P, \forall t \in T_p \tag{5}$$

$$O_p = \sum_{a \in A} s_{a,i_N,(\omega_{i_N,t_p})} \qquad\qquad \forall p \in P \tag{6}$$

$$N_{r,p} \leq m_{r,p} \qquad\qquad \forall r \in R, \forall p \in P \tag{7}$$

In the problem for which this model is made there is a set of assets A that each have a set of I tasks that need to be completed within the timeframe. There are two timescales used; periods p which is the timescale at which turbines can be brought online and vessels can be chartered (a month within my artificial test cases) and timesteps t at which tasks are performed (an hour in my test cases). These two different timescales help incorporate the variety of events captured by this model; weather conditions and tasks being completed happens on the scale of hours, but a vessel chartered is chartered for a minimum duration (for example, one month).

There are three types of decision variables in this model:

- O_p: integer variable representing how many turbines are online in a given period p

- $N_{r,p}$: integer variable representing how many resources of type r are used in period p

- $s_{a,i,t}$: binary variable which takes a value of 1 if task i on asset a has start at (or before) time t

The primary decisions that need to be made are the start times of the tasks (the first time steps for each asset a and task i for which $s_{a,i,t}$ takes the value 1); from this the times the turbines come online and the resources required follow directly.

Equation (1) is the objective; the goal is maximizing the total value of the project. Therefore, per period, we add up the energy value generated ($O_p \cdot v_p$ where v_p is the value of energy that a single turbine generates in that timeframe) and subtract the costs of vessels used ($N_{r,p} \cdot C_{r,p}$ where $C_{r,p}$ is the cost of that type of vessel). It is assumed the values $C_{r,p}$ and v_p are deterministic and given as inputs. If a day-rate for vessels or MWh and energy market rates are given, the values per period can be calculated from that.

Equation (2) enforces that if a task is started it remains started for every later timestep. Equation 3 uses the time indicator $\omega_{i,t}$ which treated as a parameter; it indicates the timestep at which task i should have been started, for it to have finished at time t. It is defined as

$$\omega_{i,t} = \max\{\tilde{t} | \sum_{x=\tilde{t}}^{t} \psi_{i,x} \geq d_i, \tilde{t} \in T\}$$

where $\psi_{i,t}$ is a binary parameter which is 1 if the weather allows task i to be performed at time t, and d_i is the duration of task i. In this it is assumed that every task can be started and then halted if the weather turns bad, and continued the moment the weather allows this again (during this entire

18

timeframe the vessels are unavailable to perform other tasks). Since this is a highly simplified model, the weather and duration are assumed to be perfectly predicted, hence $\omega_{i,t}$ is a deterministic value. In equation (3) it uses the value t_N which is the final timestep, hence this equation enforces that every task is started early enough for it to be finished by the final timestep.

Equation (4) enforces a specific task order; IP is the set of pairs of tasks (i, j) which require task i to be finished before task j is started. In equation (5) the number of vessels needed in a specific period is calculated; $\rho_{i,r}$ is the amount of resources of type r task i needs. This value is counted once for every task started, but every task that finished is subtracted again. This equation uses T_p which is simply the set of timesteps within period p. In Equation (6) i_N is the final installation task, and t_p is the deadline (in timesteps) for a turbine to be finished if it is to be active in period p (this can be the first timestep of p, but could also be chosen to be earlier if the operator wishes to do so). Equation (6) counts how many assets have finished installation (i.e. completed task i_N) in time to be active in period p (before time step t_p); the term ω_{i_N, t_p} is the timestep at which the final installation task should have been started for it to be completed by the deadline t_p. Finally equation (7) puts a maximum on the number of vessels of a specific type used in a given period, where $m_r p$ is an input parameter. Without this the model might decide to charter 100 heavy lift vessels in the first period and work on all turbines simultaneously, which is unrealistic in practice.

The above model can be used to optimize the installation project. The approach of using binary variables representing decisions for each time step comes from the literature, for example (Besnard et al. 2011, Irawan et al. 2019). It is assumed that the precise weather is known for the entire project (which can often span years), assumes no delay in tasks other than that based on weather, and assumes every turbine is active at full strength without any failures from the moment it is completed. This said, creating this model was valuable practice for me, as the steps it took me to get to this model taught me more aspects to pay attention to.

3.1.2 Maintenance model

The maintenance model is very similar in structure; it assumes that an asset will fail if it is not maintained for some predetermined time. Under these assumptions the maintenance models had a set amount of minimum maintenance tasks that should be performed, and a set of optional tasks that can be performed to increase uptime:

$$\max_{\substack{O_t, N_{r,p} \in \mathbb{Z}^* \\ s_{a,i,t}, b_{a,t} \in \{0,1\}}} \sum_{p \in P} [DIS^p (\sum_{t \in T_p} (O_t \cdot v_t) - \sum_{r \in R} (N_{r,p} \cdot C_{r,p}))] \tag{8}$$

subject to:

$$s_{a,i,t} \leq s_{a,i,(t+1)} \qquad\qquad \forall a \in A, \forall i \in M, \forall t \in T \tag{9}$$

$$s_{a,i,(\omega_{i,t_N})} = 1 \qquad\qquad \forall a \in A, \forall i \in M^M \tag{10}$$

$$N_{r,p} \geq \sum_{a \in A} \sum_{i \in M} (\rho_{i,r} \cdot (s_{a,i,t} - s_{a,i,(\omega i,t)})) \quad \forall r \in R, \forall p \in P, \forall t \in T_p \tag{11}$$

$$b_{a,t} > \sum_{i \in M} [s_{a,i,(\omega_{i,(t-\lambda_a)})} - s_{a,i,(\omega_{i,t})}] \qquad \forall a \in A, \forall t \in T \tag{12}$$

$$O_t = |A| - \sum_{a \in A} b_{a,t} \qquad\qquad \forall t \in T \tag{13}$$

$$N_{r,p} \leq m_{r,p} \qquad\qquad \forall r \in R, \forall p \in P \tag{14}$$

The objective function is similar to that of the installation model, with two changes. It adds a discount factor DIS which can be used to discount income and costs later in the project, if the operator would wish to do so. Additionally the terms regarding online turbines and the energy value now depend on the timestep rather than period, as during maintenance they can be brought online or offline more often and on smaller timescales.

Equation (9) is identical to equation (2) from the installation model, other than that the set of tasks is now called M rather than I. This set M is made up of two distinct subsets: M^M and M^O, which respectively contain the mandatory and optional tasks. The idea behind this is M^M contains some minimum required maintenance, for example if the maintenance project spans 20 years and the operator wants at least one routine inspection to be

done every year, M^M would contain 20 inspection tasks, and M^O contains any other tasks. Looking back at this approach, it is clear that the model would not guarantee these mandatory tasks to be equally spread out. The way failures are modeled (which I will discuss later) helps to mitigate that, but I expect an operator would prefer to have some guarantee that these 20 inspection tasks do not all happen in the first 5 years.

Equation (10) enforces all mandatory tasks to be finished by the end of the considered timespan, and equation (11) is identical to (5) from the previous model.

A new variable is introduced in (12); $b_{a,t}$ is a binary variable which takes a value of 1 if asset a is broken at time t. We have a parameter λ_a which indicates the amount of time an asset can go without maintenance before failing. In the equation we add up all maintenance tasks that have finished more than λ_a timesteps ago, and subtract all tasks that have finished now. If these amounts are equal, $b_{a,t}$ is set at 1.

Finally equation (13) determines how many assets are online by subtracting the amount of broken assets from the total number, and equation (14) is identical to (7).

Based on the way this model handles failures and the set of tasks available, we could reframe the problem and say it only handles planned maintenance, and not responsive maintenance. Planned maintenance is all the maintenance that happens at predetermined moments, while responsive maintenance is that which happens in response to failures or condition-monitoring data. This distinction may seem similar to the distinction between preventive and corrective maintenance made in some papers in Section 2, but that distinction focuses on whether the maintenance happens before or after failure, which is a different metric since maintenance that happens based on condition-monitoring before any failure occurs is preventive but not planned.

If we reframe the problem in this way, λ_a is a predetermined amount of time after maintenance after which the chance of a failure exceeds some threshold; from this moment the turbines energy is not longer included in the

21

objective function as the chance of failure is deemed too high. M^M would still represent some minimum amount of maintenance, and $M^M \cup M^O$ represents the maximum amount of planned maintenance. With this reframing the model could be used to schedule all planned maintenance, while a separate model and strategy would have to be determined for responsive maintenance.

3.1.3 Decommission model

The initial decommission model is nearly identical to the previously shown installation model:

$$\max_{\substack{N_{r,p}\in\mathbb{Z}^* \\ s_{a,i,t}\in\{0,1\}}} \sum_{p\in P}[DIS^p(\sum_{t\in T_p}\sum_{a\in A}((1-s_{a,i_0,t})\cdot v_t) - \sum_{r\in R}(N_{r,p}\cdot C_{r,p}))] \qquad (15)$$

subject to:

$$s_{a,i,t} \leq s_{a,i,(t+1)} \qquad\qquad \forall a \in A, \forall i \in I, \forall t \in T \quad (16)$$

$$s_{a,i,(\omega_{i,t_N})} = 1 \qquad\qquad \forall a \in A, \forall i \in I \quad (17)$$

$$s_{a,j,t} \leq s_{a,i,(\omega_{i,t})} \qquad\qquad \forall a \in A, \forall(i,j) \in IP, \forall t \in T \quad (18)$$

$$N_{r,p} \geq \sum_{a\in A}\sum_{i\in I}(\rho_{i,r}\cdot(s_{a,i,t} - s_{a,i,(\omega_{i,t})})) \qquad \forall r \in R, \forall p \in P, \forall t \in T_p \quad (19)$$

$$N_{r,p} \leq m_{r,p} \qquad\qquad \forall r \in R, \forall p \in P \quad (20)$$

The only difference is the absence of the O_p variables and constaints related to it, and the objective function. It adds a discount factor DIS, as in the maintenance model, and sees if decommission has started at any particular time with variable $s_{a,i_0,t}$, where i_0 is the first decommission task (after which a turbine is offline).

3.1.4 Combined model

After creating these models I combined them into one model spanning the entire life-cycle, planning every single task (under the same deterministic assumptions).

22

$$\max_{\substack{N_{r,p} \in \mathbb{Z}^* \\ s_{a,i,t}, o_{a,t} \in \{0,1\}}} \sum_{p \in P} [DIS^p (\sum_{t \in T_p} \sum_{a \in A} (o_{a,t} \cdot v_t) - \sum_{r \in R} (N_{r,p} \cdot C_{r,p}))] \qquad (21)$$

subject to:

$$s_{a,i,t} \leq s_{a,i,(t+1)} \qquad \qquad \forall a \in A, \forall i \in \mathcal{I}, \forall t \in T \quad (22)$$

$$s_{a,i,(\omega_{i,t_N})} = 1 \qquad \qquad \forall a \in A, \forall i \in \mathcal{I} - M^O \quad (23)$$

$$s_{a,i_0^D,t} - 1 \leq s_{a,i,(\omega_{i,t})} - s_{a,i,t} \qquad \forall a \in A, \forall i \in \mathcal{I} - D, \forall t \in T \quad (24)$$

$$s_{a,j,t} \leq s_{a,i,(\omega_{i,t})} \qquad \qquad \forall a \in A, \forall (i,j) \in IP, \forall t \in T \quad (25)$$

$$N_{r,p} \geq \sum_{a \in A} \sum_{i \in \mathcal{I}} (\rho_{i,r} \cdot (s_{a,i,t} - s_{a,i,(\omega_{i,t})})) \qquad \forall r \in R, \forall p \in P, \forall t \in T_p \quad (26)$$

$$N_{r,p} \leq m_{r,p} \qquad \qquad \forall r \in R, \forall p \in P \quad (27)$$

$$o_{a,t} \leq s_{a,i_N^I,(\omega_{i_N^I,t})} - s_{a,i_0^D,t} \qquad \forall a \in A, \forall t \in T \quad (28)$$

$$o_{a,t} \leq \sum_{i \in M \cup \{i_N^I\}} (s_{a,i,(\omega_{i,t})} - s_{a,i,(\omega_{i,(t-\lambda_a)})}) \qquad \forall a \in A, \forall t \in T \quad (29)$$

$$o_{a,t} \leq 1 + s_{a,i,(\omega_{i,t})} - s_{a,i,t} \qquad \forall a \in A, \forall i \in M, \forall t \in T \quad (30)$$

For the most part this model is a combination of the previously shown models. It uses new variables $o_{a,t}$, which is a binary variable per asset and timestep indicating whether that asset is online at the given time. In order for it to be online that asset needs to be fully installed and not started decommission (equation (28)), have had recent maintenance (equation (29)) and no maintenance on at asset can be happening at this time (equation (30)). The value of these variables follows directly from the scheduling decisions, but I introduced the auxiliary variable for clarity. In these equations i_N^I indicates the final installation task and i_0^D indicates the first decommission task. In the objective function this new variable is used to calculate the total value of energy generated. Equation (24) ensures only decommission tasks are allowed to be performed after decommissioning has started. All other equations are taken directly from the installation model, with the only difference that the set $\mathcal{I}$ is the set of all tasks and D is a subset containing all decommission tasks.

This model still assumes deterministic weather which limits its practical usability, but even if this limitation is acceptable it will still be computationally challenging to find an optimal solution for a schedule spanning two or three decades, which would be the common duration of the full lifespan for which the model is designed. To illustrate, take a windfarm with 80 assets, each of which need 40 tasks to be scheduled over the course of its lifespan. If this lifespan is 20 years and the timesteps are chosen to be 1 hour, this results in 175.000 timesteps, and over half a billion $s_{a,i,t}$ variables, which would be difficult to solve to optimality. Additionally the uses of scheduling every single task over the course of 20 years are limited.

However, these three phases tend to be treated completely separately within the literature, which also has strong drawbacks. A goal of this research is to find a middle ground that balances the drawbacks of both approaches, and the above model is a step in that process. Even if the direct practical uses of this model are limited, it was still a good exercise for me to work on. I would not have some of the insights above if I had not created a (simplified) model spanning the entire life-cycle, and experimenting with the size of model (in number of variables and constraints) gave me a better idea of how quickly models such as this can grow in size.

3.2 Stochastic model

After working my way up to the previously discussed life-cycle spanning model, I aimed to relax the assumptions in it, first and foremost the assumption that every aspect is deterministic. This proved difficult for various reasons. I started with the model shown in Section 3.1.4, which is a fairly complicated model already. Since I do not have much experience with Stochastic Programming models, starting with a complicated model such as this was challenging. The model also spans a large timescale, but still has a high fidelity (if a timestep is still assumed to be an hour).

In the models shown before the parameter $\omega_{i,t}$ was used to incorporate weather circumstances, and this parameter was used as a variable index.

This approach is not well suited for Stochastic Programming, as this $\omega_{i,t}$ would be subject to uncertainty, which means the variables involved are also subject to uncertainty. I tried to handle this issue by introducing scenarios, a technique used in the literature (Besnard et al. 2011, de Matos & Finardi 2012). Each scenario represents a realization of all uncertain factors, and the model would be optimized over a set of (selected) scenarios. In this example a new index would be introduced to $\omega_{i,t}$, say $\omega_{i,t,s}$, where s is the scenario index. However this approach is more suited to shorter timescales than the ones I was using; only a handful of scenarios are needed to capture the possible weather conditions over the course of a week or a month, but over the course of 20 years many scenarios are needed to make the set representative of the possible options. Therefore I opted to scale down and introduce stochastic elements in a smaller model first.

For this I chose to focus on the maintenance phase, introducing stochastic failures and energy values:

$$\max_{\substack{N_{r,p}\in\mathbb{Z}^* \\ s^P_{a,i,t},s^R_{a,i,t,\sigma},o_{a,t,\sigma}\in\{0,1\}}} \sum_{\sigma\in S}\sum_{p\in P}[DIS^p(\sum_{t\in T_p}\sum_{a\in A}(o_{a,t,\sigma}\cdot v_{t,\sigma})-\sum_{r\in R}(N_{r,p}\cdot C_{r,p}))] \tag{31}$$

subject to:

$$s^P_{a,i,t} \leq s^P_{a,i,(t+1)} \qquad\qquad \forall a\in A, \forall i\in M^P, \forall t\in T \tag{32}$$

$$s^R_{a,i,t,\sigma} \leq s^R_{a,i,(t+1),\sigma} \qquad\qquad \forall a\in A, \forall i\in M^R, \forall t\in T, \forall \sigma\in S \tag{33}$$

$$s^P_{a,i,(\omega_{i.t_N})} = 1 \qquad\qquad \forall a\in A, \forall i\in M^P \tag{34}$$

$$N_{r,p} \geq \begin{cases} \sum_{a\in A}(\sum_{i\in M^P}(\rho_{i,r}\cdot(s^P_{a,i,t}-s^P_{a,i,(\omega_{i,t})})) \\ +\sum_{i\in M^R}(\rho_{i,r}\cdot(s^R_{a,i,t,\sigma}-s^R_{a,i,(\omega_{i,t}),\sigma}))) \end{cases} \qquad \forall r\in R, \forall p\in P, \forall t\in T_p, \forall\sigma\in S \tag{35}$$

$$N_{r,p} \leq m_{r,p} \qquad\qquad \forall r\in R, \forall p\in P \tag{36}$$

$$o_{a,t,\sigma} \leq \begin{cases} \sum_{i\in M^P}(s^P_{a,i,(\omega_{i,t})}-s^P_{a,i,(\omega_{i,(t-\lambda_{a,i,\sigma})})}) \\ +\sum_{i\in M^R}(s^R_{a,i,(\omega_{i,t}),\sigma}-s^R_{a,i,(\omega_{i,(t-\lambda_{a,i,\sigma})})\sigma}) \end{cases} \qquad \forall a\in A, \forall t\in T, \forall\sigma\in S \tag{37}$$

$$s^R_{a,i,t}-s^R_{a,i,(t-1)} \leq \begin{cases} 1-\frac{1}{L}\cdot(\sum_{j\in M^P}(s^P_{a,j,(\omega_{j,t})}-s^P_{a,j,(\omega_{j,(t-\lambda_{a,j,\sigma})})})) \\ +\sum_{j\in M^R}(s^R_{a,j,(\omega_{j,t}),\sigma}-s^R_{a,j,(\omega_{j,(t-\lambda_{a,j,\sigma})})\sigma})) \end{cases} \qquad \forall a\in A, \forall i\in M^R, \forall t\in T, \forall\sigma\in S \tag{38}$$

$$o_{a,t,\sigma} \leq 1+s^P_{a,i,(\omega_{i,t})}-s^P_{a,i,t} \qquad\qquad \forall a\in A, \forall i\in M^P, \forall t\in T, \forall\sigma\in S \tag{39}$$

$$o_{a,t,\sigma} \leq 1+s^R_{a,i,(\omega_{i,t}),\sigma}-s^R_{a,i,t,\sigma} \qquad\qquad \forall a\in A, \forall i\in M^R, \forall t\in T, \forall\sigma\in S \tag{40}$$

This model is still based on the previous models, and it introduces a set of scenarios S, and every scenario $\sigma\in S$ is a realization of the stochastic failures and energy values. What this model does is split the $s_{a,i,t}$ variables from the previous models in two groups, $s^P_{a,i,t}$ and $s^R_{a,i,t,\sigma}$, which represent planned and responsive maintenance tasks respectively. The $s^P_{a,i,t}$ variables function exactly like before, and $s^R_{a,i,t,\sigma}$ do the same but depend on a scenario. In the same way the variables $o_{a,t,\sigma}$ now depend on the scenario. In this way, the failure state $o_{a,t,\sigma}$ and responsive tasks $s^R_{a,i,t,\sigma}$ are solved for every scenario separately, but the planned tasks $s^P_{a,i,t}$ and vessel charters $N_{r,p}$ have to be planned in advance and are the same for every scenario.

The only really new equation is (38); it is meant to ensure that responsive tasks only happen after a failure. The left hand side takes a value of 1 only

if that the corresponding task started at timestep t, and the right hand side is 1 unless a task has recently been completed on the asset. How recently is determined by $\lambda_{a,i,\sigma}$ which is the time after task i is completed at which asset a fails in scenario σ. These variables for all tasks are summed up in the same way as was done in equation (29), where the sum is equal to the number of tasks completed recently enough. The L in this equation is a large number, at least $|M^P \cup M^R|$, to ensure the right hand side never goes negative.

Equations (32) and (33) are the same as equation (22), this time split for the planned and responsive tasks respectively; the same happened with equations (39) and (40) which both come from (30). Equations (34) and (36) are equal to (23) and (27) respectively. In equations (35) and (37) the task summation is split in two parts for the two types of task, but apart from that they are directly taken from (26) and (29) respectively.

The strength of this model is that it can be used to plan decisions that need to be made in advance (which vessels to charter and when to schedule certain routine tasks) while considering a set of uncertain scenarios. However it does still assume complete *a priori* knowledge of each of these uncertain factors, as well as the weather. Since the focus lies on the decisions universal to all scenarios, rather than any specific scenario, this might not pose an issue for some uses. The model could potentially be improved by letting the weather conditions (encompassed in $\omega_{i,t}$) also depend on the scenario. The reason I have not done that (yet) is that while developing these models, I have also been implementing at the same time. This is discussed in detail in Section 4. The implementation of the above model already had issues being solved on my computer with a very small test case, and I wanted to improve that before adding more complexity.

3.3 Multilevel model

The above model grew in complexity with every iteration, and since I had reached a point where the implementation could not even solve a very simple

test case in a number of hours (on my personal computer) I attempted a different approach. In the literature multilevel models are common, and splitting the problem into multiple subproblems can reduce the time it takes to solve. My idea behind this model was one level in which large scale decisions are made, such as for which tasks to complete and which vessels to charter for every month. Then a lower level can assign these tasks to specific vessels in some order.

At the time of writing this model exists in a preliminary form, but it has not been completely tested yet, and there may still be improvements to be made. There are still some aspects of the model that I have not fully decided on, and in this report I will try to explain those aspects. The higher level model I came up with is shown here:

$$\min_{\substack{N_{y,m,\sigma},P_m,R_{m,\sigma}\in\mathbb{Z}^* \\ \gamma_\sigma\in\mathbb{R}_{\geq0} \\ h_{m,\sigma}\in\{0,1\}}} \frac{1}{|S|}\cdot\sum_{\sigma\in S}(\sum_{m\in M}[\sum_{y\in Y}(N_{y,m,\sigma}\cdot c_{y,m})+P_m\cdot d_P\cdot e_m^H+ \\ \sum_{m'=0}^{m}(f_{m',\sigma}-R_{m',\sigma})\cdot e_m^H\cdot H_m]-u\cdot\gamma_\sigma) \tag{41}$$

subject to:

$$l_y\cdot N_{y,m,\sigma}\geq\begin{cases}P_m\cdot d_y^P+R_{m,\sigma}\cdot d_y^R \\ +\gamma_\sigma-h_{m,\sigma}\cdot L\end{cases} \qquad \forall\sigma\in S,\forall m\in M,\forall y\in Y \tag{42}$$

$$1\geq h_{y,m,\sigma}+\frac{P_m+R_{m,\sigma}}{L} \qquad \forall\sigma\in S,\forall m\in M \tag{43}$$

$$\sum_{m'=0}^{m-1}f_{m',\sigma}\geq\sum_{m'=0}^{m}R_{m',\sigma} \qquad \forall\sigma\in S,\forall m\in M \tag{44}$$

$$\sum_{m\in M}P_m\geq A \tag{45}$$

Keep in mind that where every previously shown model used similar notation, this model was started from scratch, and uses a lot of new notation.

There are three primary types of decision variables used in this model:

- $N_{y,m,\sigma}$: The amount of resources (vessels) of type y chartered in month m in scenario σ

28

- P_m: The amount of planned maintenance tasks in month m

- $R_{m,\sigma}$: The amount of responsive maintenance tasks in month m in scenario σ

For the purposes of this model it is assumed there is only one type of planned task, and one type of responsive task. This means that the durations and required vessels are uniform. This assumption could be relaxed, but as my goal with this model was to try out a new, easier to solve, approach before complicating it more.

There are two other types variables in the model, γ_σ and $h_{m,\sigma}$. The latter is simply a set of helper variables to make the constraints work linearly which will be explained later, but the former type of variable represents a buffer in the capacity. In this model a set of chartered vessels and a set of tasks is decided for each month. Each vessel has a capacity, such as the amount of hours it can work in a month, and each task has a duration. There will usually be months where the full capacity is not used, but this can be good for robustness. These spare hours can be used in case of a delay, and we call it a buffer. For a scenario, γ_σ is the minimum buffer over all months within that scenario, and it is included in the objective function to encourage a higher buffer. This should lead to tasks being spread over the months, and will ideally create a robust schedule in which no vessel is busy every hour in a given month.

The objective function (41) uses the following parameters:

- $c_{y,m}$: The cost of chartering vessel type y in month m

- d_P: The duration of a planned task

- e_m^H: The average value of energy generated by a single turbine in an hour, in a given month m

- $f_{m,\sigma}$: The amount of failures in month m and scenario σ

- H_m: The the amount of hours in month m

- u: A parameter that sets the value per hour of buffer

The parameter u can be adjusted based on the operators wishes; the higher it is, the higher buffers are prioritized over minimizing expected costs. An appropriate value for a given situation will likely have to be found through experiments.

In the objective function an average is taken over all scenarios, which counts the costs made through vessel charter $((N_{y,m,\sigma} \cdot c_{y,m}))$, value of energy missed due to downtime during planned maintenance $(P_m \cdot d_P \cdot e_m^H)$, value of energy missed due to failures that are not yet repaired $(\sum_{m'=0}^{m}(f_{m',\sigma} - R_{m',\sigma}) \cdot e_m^H \cdot H_m)$, and the discount for the size of the buffer $(-u \cdot \gamma_\sigma)$. These are approximations, as energy values can fluctuate per hour and an average is used, and failures are assumed to cost a full month worth of energy. Since a failure can occur at any point in the month, and be repaired at any point in a later month, this seems like a reasonable assumption for an approximation. This model focuses on high-level decisions, so approximations of the cost are sufficient.

Equation (42) is the capacity constraint; l_y is the capacity (available hours) of a single vessel, and d_y^P and d_y^R are the durations for which planned and responsive tasks require a vessel of type y. The variable $h_{m,\sigma}$ is binary and can take the value of 1 if and only if no tasks are planned in month m in scenario σ, which is enforced by the constraint in equation (43). In both of these equations L is a large number. This extra variable is required in case there are months in which no tasks are planned and no vessels are chartered; in this situation the left hand side of equation (42) is equal to 0, and there could be no buffer without this extra term.

In equation (44) it is enforced that there can never be more repairs than there were failures at the end of the previous month, and in equation (45) a minimum amount of planned tasks is set to be carried out over the planning horizon. If A is set as the amount of assets, every asset would be maintained once within this horizon.

This above model can be used to plan high-level decisions, in a similar

way the model in Section 3.2 can. However since the level of detail is much lower, the resulting problem will be smaller and faster to solve. The drawback may also be reduced accuracy, but in order to determine whether that is the case computational experiments will have to be done.

When a solution to the above model has been found, there is one remaining problem; for any given month, which vessel will do which tasks in what order. In order to solve this the following lower-level model can be solved for every month:

$$\min_{\substack{s_i \in \mathbb{R}_{\geq 0} \\ a_{v,i,j} \in \{0,1\}}} \sum_{i \in I} c_i \cdot \left(s_i + \max_{y \in Y}(s_{y,i} + d_{y,i})\right) \tag{46}$$

subject to:

$$\sum_{i \in I} a_{v,i,j} \leq 1 \qquad\qquad \forall v \in V, \forall j \in J \tag{47}$$

$$\sum_{i \in I} a_{v,i,j} \leq \sum_{i \in I} a_{v,i,(j-1)} \qquad\qquad \forall v \in V, \forall j \in J - \{0\} \tag{48}$$

$$\rho_{y,i} \leq \sum_{v \in V_y} \sum_{j \in J} a_{v,i,j} \qquad\qquad \forall y \in Y, \forall i \in I \tag{49}$$

$$s_i + \max_{y \in Y}(s_{y,i} + d_{y,i}) \leq T \qquad\qquad \forall i \in I \tag{50}$$

$$\begin{aligned} L \cdot (a_{v,i,j} + a_{v,i',(j-1)}) + \\ d_{y,i'} \cdot a_{v,i',(j-1)} - 2L \end{aligned} \leq s_i + s_{y,i} - s_{i'} - s_{i',y} \qquad \begin{aligned} &\forall y \in Y, \forall v \in V_y, \\ &\forall i, i' \in I, \forall j \in J - \{0\} \end{aligned} \tag{51}$$

This model has two types of decision variables:

- s_i: The starting time of task i

- $a_{v,i,j}$: A binary variable indicating whether vessel v has task i as its j-th task

Through deciding these variables every task will be assigned to a vessel in a specific order, and each task will have a starting time. In addition to the set of vessel types Y we had in the high-level model we now also have a set of specific vessels V, which can be divided into distinct sets V_y based on

vessel type. For every task, the amount of vessels of type y needed is called $\rho_{y,i}$. In addition to the starting time s_i that every task has, there may be a delay before the vessel(s) of type y are required, this delay is called $s_{y,i}$. For example, if a task needs a Crew Transfer Vessel from the start but a Heavy Lift Vessel after 3 time units, it would have $s_{CTV,i} = 0$ and $s_{HLV,i} = 3$. It is assumed that if a task requires multiple vessels of the same type, each of these is needed from the same time, and for the entire duration $d_{y,i}$ without breaks. Finally every task has a cost per time unit c_i in which it is not completed; for reparation tasks this would be equal to the value of energy generated per time unit, while for routine checkups this cost could be 0 (which would mean these tasks will generally be done last).

The objective is to minimize the cost over all tasks. The term $\max_{y \in Y}(s_{y,i} + d_{y,i}$ is a constant which indicated the total duration of the task, and it could be removed without affecting the solution. However it is included to have the objective value better reflect the actual costs of the schedule.

Equation (47) ensures every vessel has at most 1 task assigned as its j-th task, for every j, and equation (48) enforces a $(j-1)$-th task to be assigned if a j-th task is assigned. Equation (49) makes sure every task has sufficient vessels assigned to it, and (50) uses the same constant term the objective function does to ensure every task is finished before the finish time T.

Equation (51) is used to ensure the starting times of consecutive tasks are far enough apart. L is again a large number, which ensures that the left hand side is a large negative number unless $a_{v,i,j} + a_{v,i',(j-1)} = 2$, i.e. task i' is the $(j-1)$-th task assigned to vessel v and task i is its j-th task. If this is the case the L terms cancel out and left hand side becomes equation to the duration for which $v \in V_y$ is needed. The right hand side is equal to the time at which the vessel is needed at task i $(s_i + s_{y,i})$ from which its starting time at task i' $(s_{i'} + s_{y,i'})$ is subtracted.

This is a fairly straightforward model (compared to the other models shown) as it is meant to only schedule task in a short timespan (a month) and many impactful decisions have already been made in the high-level model.

The largest drawback of this model is the speed at which it grows based on the number of tasks. As $|J|$ is the maximum amount of tasks that can be assigned to a single vessel, $|J|$ will likely have to be equal to $|I|$. This would mean the amount of $a_{v,i,j}$ variables grows quadratically with the amount of tasks, and the amount of constraints resulting from equation (51) grows cubically with the number of tasks. While cubic growth in the number of tasks is a weakness, the model size is independent of amount of timesteps. The previously shown models had discrete timesteps and binary variables for each timestep, which resulted in the amount of variables and constraints growing linearly with timesteps. As there will generally be orders of magnitudes more timesteps than tasks, removing this linear growth mitigates the effect of the cubic growth in the number of tasks. This is further helped by the robustness measure in the high-level model; that model can be tuned to try and spread out tasks over months, which will reduce the amount of tasks in a single month.

The model could potentially be expanded by adding a more sophisticated cost function (which takes into account downtime costs for routine maintenance tasks), making durations stochastic, or by adding a robustness measure (a buffer variable similar to γ_σ in the high level model can be added to equation (51)). As I have not yet fully implemented this multilevel model these improvements might happen later on in my research, depending on the initial implementation results. These results are also required to properly reflect on these two levels of the multilevel model; potentially a stronger link can be used between the two levels but at this point in time I am unable to make any definitive statement on this.

4 Implementation

In addition to the developing the models discussed in the previous section, I also implemented the models and solved them for small test cases. This part of the process was highly valuable in my development as a modeler, and many of the design decisions that lead to the creation of the previously shown models were informed by insights and results gained from the implementation. In this section I will describe the chronological progress made through implementation and highlight the obstacles and what I learned in dealing with them. In addition to this section, my full current code is available as an appendix.

4.1 Getting an initial implementation

Before this project I had only used simple graphical interfaces for solvers to solve simple exercises from introductory courses to optimization. This means I had very little experience really working with solvers, and no experience doing so through programming. I decided for this project to work in C++ as I am familiar with the language (more than Java or Python for example) and most solvers I looked at can be called through a C++ library. By the time I started this phase of the project I was forced to work from home because of the Covid-19 pandemic, which means I could not use the university computers and had to install any solver I wished to use on my own personal computer. This caused some startup problems getting a good solver to work properly. I started off using free solvers, Google's OR-tools at first (*OR-Tools* 2015). I had issues getting this to work on my PC and I kept having issues with the library, so I switched to another free solver, COIN-OR (*Computational Infrastructure for Operations Research (COIN-OR)* 2004). This also had some issues, and after I asked my supervisors for advice they helped me get a license for the FICO Xpress optimization suite (*FICO Xpress Optimization* 1983). This was the first commercially available LP and MIP solver, and its current version is still ranked among the top solvers, together with Cplex

and Gurobi. This solver was much easier to get to work than the ones I previously used, and I could finally get to actually solving models.

I started with the Xpress tutorial and from there slowly changed the tutorial model to the installation phase model I had at the time, a precursor to the model shown in Section 3.1.1. I had made a very simplistic test case for this, with only 3 turbines, 2 tasks per turbine, and 2 vessels to complete them, with artificial durations for each task to simply test whether the model gave the results I expected. Whenever this was not the case, it gave me insights in the model; for example at first I did not have a limit on the amount of vessels that could be rented in a period, and regardless of how many turbines there were every turbine would be installed in the first period, as the amount of vessels available was unlimited and there was a benefit to installing all turbines as fast as possible. This lead to unrealistic amounts of vessels rented in the first period, and none after; hence I introduced the limit seen in equation (7).

4.2 Increasing efficiency

I kept on expanding the implementation of the model while adapting my models based on new findings or insights, and eventually got to the combined model from Section 3.1.4. While the implementation seemed to work with very small test cases, I was unable to find solutions for any artificial test cases spanning several years within 8+ hours of running. As an inexperienced modeler it was hard to determine where the issue lay, and I saw four possible areas that could limit the speed:

- The model itself could be inefficient

- The implementation could be the problem

- My PC might simply not be powerful enough (it has a 3.20 GHz Intel i5-4570 CPU)

- The size of the test case might simply be unrealistic even for an properly implemented efficient model on a state-of-the-art computer

Based on some discussion with my supervisors I decided to mostly focus on improving the model and implementation, and if I got to a point where I could not find new ways to do that we could look into running this on better computers available from the university. Looking for such improvements would be the right first step even if the main problem was my computer, and it is a valuable skill to have experience with.

One of the most effective ways to reduce the computation times I used was through solving the problem in multiple stages. Instead of giving the solver the full model at once, I would leave out certain constraints at first (not adding them at all, or adding them as cuts), solve the resulting problem, and then add back the full constraint set and solve that problem. The total time of these two solves would often be significantly less than that of the single big solve I used before, likely because the first solve gave the solver a lot of information about the included constraints; information that helped in the second solve. The big question after finding this was selecting which constraints to include in the first solve, and which to leave out. Rather than determining this analytically I experimented with different combinations on small test cases. To facilitate this I build a Mode class in my program that could be used to cycle through modes, solving the same problem for every combination of constraint types excluded from the first solve, and recording the time each solve took. Instead of considering constraints individually I grouped them based on the equation they resulted from; the model in Section 3.1.4 can have any number of constraints based on the input, but it will always have 9 constraint types, each represented by one of the equations (22) through (30). Based on the results from small cases I identified which combinations worked best, and those are the settings I would use to solve larger test cases. Using this method, I managed to cut down average computation times for medium test cases roughly in half, and made larger cases for which I could previously not find a solution in 8 hours solvable to optimality in

under 4 hours.

While that sounds promising, I was still not able to find solutions for the model on test cases anywhere near the size of the full life cycle. However, considering the size of the model and the problem, and my consumer computer I was using, hoping to solve this was rather ambitious. This is why I chose to scale the problem down and move on to stochastic models (as discussed in the previous section) rather than get stuck trying to solve a problem that would likely never be solvable using my computer.

4.3   A different approach

It was while implementing the stochastic model from Section 3.2 that I decided to try a different approach; the multilevel model shown in Section 3.3. The main reason for this was that I had trouble getting the stochastic elements to work as desired, and I realized a large set of scenarios would be needed to represent 20+ years worth of stochastic events. This did not fully become clear to me until implementing the model, and at the same time I realized how the model would grow significantly with the amount of scenarios. The reason I was focusing on the maintenance phase for this model was to reduce the scale compared to the life-cycle model, but through this growth based on the number of scenarios this problem would not be scaled down. That's when I realized a completely new approach might be needed, and I moved on to the multilevel model.

This is when the final large programming change happened. Previously every model had followed the same structure, but this model broke that pattern, which required me to do a lot of extra programming for the model to be properly implemented. At the time of writing I have completed that for the two separate levels within this model, but I have not yet made it so the levels are called in succession; this is still something I wish to do in the near future.

5 Data acquisition

In the previous section I talked about implementations being tested; the test cases used for this were all artificially created test cases inspired by real data but heavily simplified. For the purposes I used these test cases for (testing model sanity and computation time) artificial cases are sufficient, but in order to properly compare the models to models from the literature more realistic test cases are needed, and in order to construct these data will be needed. I have identified two main types of data needed, which I call weather data and operational data.

Weather data is data about the weather circumstances at the site, which can be used to estimate the frequency weather delays and the amount of energy create over a given period. Data used should ideally come from a weather station in very similar weather conditions as the site of the windfarm will be in, and recorded over a long period of time. Many years of data are needed to accurately assess seasonal characteristics of the weather, and this data can contain many weather characteristics such as wave height, wind speed, wind direction, etc.

At the end of my first year review one of my immediate goals was to attempt to get access to data for my future testcases, and since then I have gained access to the data from the FINO1 and FINO3 stations. The FINO1 station (*FINO 1 Meteorological Dataset* 2020) is the primary station I aim to use as it is more commonly used in the literature (Barlow et al. 2018, Dinwoodie et al. 2015). This station is positioned in the North Sea, roughly 45 kilometres off the cost of Germany. This station has been active since 2003 which means many years of data are available, from a location representative of the location of the type of windfarm focused on in my research.

Operational data is the data about the windfarm, the vessels, and the tasks that need to be performed. If the model is used in an industrial context this data would obviously be specified by the user, as it is specific to the

project the model would be used for. However, in order to answer the research questions some realistic data is needed to populate the simulations. If I can get data from case studies used in other research projects, that might allow for some comparison between my models and the models from those projects.

Currently I have access to some operational data regarding the installation tasks, which my supervisors used for their research in (Barlow et al. 2018) and (Barlow et al. 2017). In (Leontaris et al. 2016) a case study of a OWF with 55 WTGs is used. This case study was provided by Van Oord, a Dutch company developing offshore windfarms. I will attempt to contact the authors and Van Oord to see if I can get access to this data. However I have not done this yet, as I wanted to have a more sophisticated model before contacting them, as the model dictated which specific data is required, and I wanted to avoid having to contact them multiple times if my required data changed.

These case studies all have to do with the installation phase, but in order to answer my research questions I need data from the other phases of the life cycle too, and ideally have data from a windfarm over the whole life-cycle. This will likely be difficult, as it requires the life-cycle to have finished, which (given the 20-30 years length of their life-cycles) means the windfarm could be too outdated for the data to be accurate for current windfarms. Therefore I am aiming to get accurate recent data about the maintenance, and estimates regarding decommission. To accomplish this I will contact OWF operators for cooperation. The OWEZ windfarm has previously provided data for other studies, such as (Dinwoodie et al. 2012), so I am hopeful this or other windfarms will help me acquire the data I need.

6   Additional activities

In addition to the work directly related to my PhD project, I have participated in some courses.

I was supposed to take part in the NATCOR course Convex Optimization, to be held at the University of Edinburgh in June 2020. However, due to the Covid-19 pandemic this course was canceled. I was also enrolled in the NATCOR course Forecasting and Predictive Analytics, which was to be held in September 2020 at Lancaster University. This course, due the same pandemic, has been postponed to February 2021.

While those courses not taking place as planned was disappointing, another course I did not originally plan to go to had to move entirely online, making it possible for me to follow it. The course in question was CO@Work (Combinatorial Optimization at work), hosted by the University of Berlin. This two-week course offered lectures (via Youtube) on varying topics, and exercise and Q&A sessions (via Zoom). Since the Zoom sessions took place at inconvenient times (due to timezones), I primarily partook in the lectures. The course was aimed at a wide variety of students, ranging from undergraduates new to optimization, to PhD students like myself. This meant that some lectures went over material I am already familiar with, like the workings of the simplex algorithm. Other material focused on techniques to help with solving linear and mixed-integer programs, such as column generation and branch-and-bound techniques. While I had previously been taught how these methods work, this was years ago, and the refresher was quite helpful. The course also had more time to go into details on these techniques, so I certainly learned new aspects of these techniques. Finally there were some corporate talks, at which various companies within the optimization industry talked about what they do and what a career with them could look like. While some of these talks seemed very specific to the company hosting it and less interesting, some also simply talked about work and careers as optimizers in general, which I found very helpful and interesting.

Generally I am glad I got to follow this course, and the format of having a few hours of lectures to watch at my own pace helped lower the workload (compared to traveling to Berlin for two weeks). This allowed me to still work on my own project during this course, and since the lectures were recorded videos rather than live, I could rewatch the parts that were most interesting or most complex. That said, I did miss the social aspect of this course, as normally with courses such as this you get to spend a week or two with students from all over the world who all study subjects similar to my own. This dimension was entirely missing, which is of course a strong drawback.

Apart from following courses, I have also helped teach a course. The course, Information Access & Mining (CS412), was a 4th year Computer Science course focusing on data analysis through machine learning. This is fairly far removed from the topic of my own project, but I was still fairly able to teach the course because of my Computer Science background. The main thing that was new for me was the Python language used in the course, with which I was previously unfamiliar. However, this simply meant that in addition to the teaching experience I got from teaching this course, I also made myself acquainted with Python, which turned out to be a relatively easy language to learn. I lead the labs, which meant I had to answer students questions regarding their exercises. Since I prepared the labs well, answering these questions was fairly straightforward. Additionally I had to mark the exercises, which took the majority of my time spend. But since I was provided with an decently detailed answer key, this was not very difficult either. After the labs stopped (due to the Covid-19 pandemic) my work solely consisted of the marking, lowering the workload.

This was my first real teaching experience, and I think it went very well. I enjoyed helping the students, and I enjoyed expanding my own knowledge of both the programming language and the subject matter. If I get another chance to help out with a course that interests me during my PhD, I will likely take it.

7 Conclusion

As is hopefully evident from the previous sections, a lot of progress has been made since the first year review. Where before I mostly had researched the installation scheduling part of the literature, I now have a broader understanding of the literature of each of the phases. In the same way, I previously had some ideas about a potential modeling approach but no concrete models yet, I now have a set of models of varying complexity, all implemented.

I'm happy with this progress, especially since there were also some delays following the Covid-19 pandemic. A lot of the hassle with trying out free MIP solver libraries could have been avoided if I had been working from university, and when the lockdown first started in March it took me a couple of weeks to adjust and develop a good working routine from home. That said, once I settled into a routine I managed to work rather productively, which is why I made quite a lot of progress over these past months.

7.1 The next step

As the basis of my knowledge (of the literature, modeling, and implementation) is now fairly solid, and I have a basis of experience with the main methods used, the next steps will focus on producing new knowledge and attempting to answer the research questions posed in Section 1.2. In the following months I will specifically focus on the first subquestion:

Sub-Question 1. *Can considering how phases in the life-cycle of a wind-farm overlap and share resources improve logistical decision making on these projects?*

This overlap of resources is the phenomenon already included in the combined model from Section 3.1.4, and my aim is to create a model that focuses in on the overlap between two phases, in order to help answer this question. I plan to develop a model that performs "installation-aware" maintenance, i.e. that is aware of an installation project happening at the same time,

with which it shares resources, and which gradually increases the amount of online turbines which may require maintenance. In this the installation schedule will be treated as a fixed input, and the goal is to see if allowing the maintenance schedule to adapt to this schedule results in new optimizations. In order to make this comparison we need an installation schedule, and a maintenance schedule/strategy that is "installation-unaware". My plan is to try and find historical schedules, either in the literature or by contacting windfarm operators, to use as input and comparison material. In case I am unable to find satisfactory historical schedules this way, my backup plan is to generate them myself, adapting the models I have already created to suit this specific use.

In this approach I choose to fix the installation schedule rather than the maintenance schedule, as the installation is likely to incur significantly more costs over these first years of the lifespan. Installation will use expensive specialized vessels for every turbine, while maintenance of that scale will not be necessary on most turbines in the first few years. This makes it less adaptable, as any improvement gained from maintenance is likely outweighed by extra installation costs incurred. Therefore having a maintenance schedule adapt to an installation schedule seems more likely to give interesting results.

What I described above is the minimum goal in order to answer the question, but there are two other thing I hope to try if time permits. The first is a fully integrated installation-maintenance model; in this the installation would no longer be treated as a fixed input but it would also be part of the problem to solve. This allows the installation schedule to be affected by the maintenance schedule, and vise versa. This model would be significantly more complex than the above model, both in difficulty of development as well as computationally, which is why I would only develop this model after the previously described model has been completed.

The other potential direction to investigate if possible, is the overlap between decommission and maintenance. Since installation and decommission are very similar phases, these models could likely be fairly similar, which

would make it easier to develop. The key difference will be the shift in focus; instead of getting the turbines online as fast as possible, the main way to increase the value will be opportunistic decommission by decommissioning turbines after a failure instead of repairing that failure.

Of these two extra research directions, I plan to pursue the integrated installation-maintenance model before looking at decommission as well, as the best way to approach the decommission question seems to be through a similar integrated approach.

If these research direction proves fruitful, our goal is to put these findings the novel model(s) in a paper to be published, as well as a chapter of my thesis.

The current goal is to get that first research question sufficiently answered in the next few months, ideally before March 2021. The exact timings will be unpredictable, as a lot will depend on whether models needs to be developed and implemented for the installation schedules and "installation-unaware" schedules, or whether these can be taken from historical data. This will inform which of the extra directions are feasible to be investigated within the timeframe, as well as the actual completion times of each of the phases on the path to answering the question. That said, I currently aim to have the initial models completed by end of November. This would include the "installation-aware" maintenance model as well as any of other models that may need to be created (or otherwise have access to historical schedules by this point). Then over December these models will be implemented, allowing for refinement of both the models and implementation over January. This implementation also includes construction of test cases and acquisition of any required data not yet accessible. Then finally over February and March analysis can be done, and I can write up my findings in a thesis chapter, and paper, if the results are significant enough to allow for a paper to be published.

7.2   Later steps

After that there are the other two questions left:

Sub-Question 2. *Can simulating the entire life-cycle of a windfarm provide useful data to base logistical decisions on in the later phases of these projects?*

Sub-Question 3. *Can considering the long-term effects of logistical decisions early on in the life-cycle of a windfarm improve these decisions?*

Since these observe the same phenomenon, they can likely be researched through similar models. From March on, my focus will be on these two joint questions. Exactly which approach to take, and what the exact timeline will look like, will have to be decided later on, but the topic appears to lend itself for a simulation approach. A key thing to focus on is how the failure rate of turbines changes over the course of their lifespan. Some research has already been found on this changing failure rate (Carroll et al. 2015, Papatzimos et al. 2019), which will be useful in my future research on this. My hope is that this aspect of the research can produce another paper, and at least it will form (an) extra chapter(s) for my thesis.

Once this part of the research is also complete, which I hope to be around summer 2021, I will have satisfying answers to each of my sub-questions, at which point I can finish writing my thesis and answer the main research question

Question. *Can considering the entirety of the life-cycle of an Offshore Wind Farm, and how each of the phases interact, improve logistical decision making on these projects?*

If all goes according to this plan, my this project and my thesis should be finished by fall 2021, roughly 12 months from this review.

References

Barlow, E., Öztürk, D. T., Revie, M., Akartunalı, K., Day, A. H. & 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.

Barlow, E., Tezcaner Öztürk, D., Day, A., Boulougouris, E., Revie, M. & Akartunalı, K. (2014), 'A support tool for assessing the risks of heavy lift vessel logistics in the installation of offshore wind farms', *Marine Heavy Transport & Lift IV* .

Barlow, E., Tezcaner Öztürk, D., Revie, M., Akartunalı, K., Day, A. & Boulougouris, E. (2017), 'On using simulation to model the installation process logistics for an offshore wind farm'.

Besnard, F., Patriksson, M., Strömberg, A.-B., Wojciechowski, A., Fischer, K. & Bertling, L. (2011), A stochastic model for opportunistic maintenance planning of offshore wind farms, *in* '2011 IEEE Trondheim PowerTech', IEEE, pp. 1–8.

Carroll, J., McDonald, A., Barrera Martin, O., McMillan, D. & Bakhshi, R. (2015), 'Offshore wind turbine sub-assembly failure rates through time'.

Computational Infrastructure for Operations Research (COIN-OR) (2004). Accessed October 15, 2020.
  URL: *https://www.coin-or.org/*

de Matos, V. L. & Finardi, E. C. (2012), 'A computational study of a stochastic optimization model for long term hydrothermal scheduling', *International Journal of Electrical Power & Energy Systems* **43**(1), 1443–1452.

Dinwoodie, I. A. & McMillan, D. (2014), 'Operational strategies for offshore wind turbines to mitigate failure rate uncertainty on operational costs and revenue', *IET Renewable Power Generation* **8**(4), 359–366.

Dinwoodie, I., Endrerud, O.-E. V., Hofmann, M., Martin, R. & Sperstad, I. B. (2015), 'Reference cases for verification of operation and maintenance simulation models for offshore wind farms', *Wind Engineering* **39**(1), 1–14.

Dinwoodie, I., Quail, F. & McMillan, D. (2012), Analysis of offshore wind turbine operation and maintenance using a novel time domain meteo-ocean modeling approach, *in* 'ASME Turbo Expo 2012: Turbine Technical Conference and Exposition', American Society of Mechanical Engineers, pp. 847–857.

FICO Xpress Optimization (1983). Accessed October 15, 2020.
URL: *https://www.fico.com/en/products/fico-xpress-optimization*

FINO 1 Meteorological Dataset (2020). Accessed October 19, 2020.
URL: *http://www.fino1.de/en/*

Gabrel, V., Murat, C. & Thiele, A. (2014), 'Recent advances in robust optimization: An overview', *European Journal of Operational Research* **235**(3), 471–483.

Irawan, C. A., Wall, G. & Jones, D. (2019), 'An optimisation model for scheduling the decommissioning of an offshore wind farm', *OR Spectrum* **41**(2), 513–548.

Kaiser, M. J. & Snyder, B. (2010), 'Offshore wind energy installation and decommissioning cost estimation in the us outer continental shelf', *US Dept. of the Interior, Bureau of Ocean Energy Management, Regulation and Enforcement, Herndon, VA TA&R* **648**.

Lacal-Arántegui, R., Yusta, J. M. & Domínguez-Navarro, J. A. (2018), 'Offshore wind installation: Analysing the evidence behind improvements in

installation time', *Renewable and Sustainable Energy Reviews* **92**, 133–145.

Leontaris, G., Morales-Nápoles, O. & Wolfert, A. R. (2016), 'Probabilistic scheduling of offshore operations using copula based environmental time series–an application for cable installation management for offshore wind farms', *Ocean Engineering* **125**, 328–341.

OR-Tools (2015). Accessed October 15, 2020.
URL: *https://developers.google.com/optimization*

Papatzimos, A. K., Thies, P. R., Lonchampt, J., Joly, A. & Dawood, T. (2019), Data-informed lifetime reliability prediction for offshore wind farms, *in* '2019 IEEE International Conference on Prognostics and Health Management (ICPHM)', IEEE, pp. 1–8.

Renewable-UK (2017), 'Offshore wind project timelines', *https://www.renewableuk.com/news/294516/Offshore-Wind-Project-Timelines.htm* . Accessed: 2019-03-24.

Shafiee, M. (2015), 'Maintenance logistics organization for offshore wind energy: Current progress and future perspectives', *Renewable Energy* **77**, 182–193.

Stålhane, M., Halvorsen-Weare, E. E., Nonås, L. M. & Pantuso, G. (2019), 'Optimizing vessel fleet size and mix to support maintenance operations at offshore wind farms', *European Journal of Operational Research* **276**(2), 495–509.