Master Thesis

An analysis on wind farm design

The comparison between manual and computed design

Jeroen Ohm, BSc

Supervised by:
Dr. Ir. Michiel Zaayer

Delft University of Technology
Faculty of Aerospace Engineering
Section Wind Energy
Kluyverweg 1
2629 HS Delft, The Netherlands

April, 2020

Abstract

Wind farms are increasing in size to keep up with the increased demand for renewable energy. Larger wind farms, leads to more complex wind farm design problems for a designer. This study investigates the wind farm design process and compares computed and manual design approaches in order to find the best solution to the wind farm layout optimization problem.

In this study, the design process of a wind farm is described in detail based on the experiences of two designers in order to get a clear view on the current state of wind farm designing. Different experiments were performed to compare a manual and a computed design approach for an offshore wind farm. It turned out that the computed designs led to a 1 % increase in energy capture compared to the manual designs. Further research is required to conclude that computed design approaches always lead to higher energy yields, however, it was found likely that the computed approach is the best method to solve the wind farm layout optimization problem. The main problem with the current available optimization software is that it cannot deal with all constraints. Therefore, optimizing a wind farm design still requires some manual changes to fulfill all requirements for the project.

Acknowledgements

First of all, I would like to thank Michiel very much for the daily supervision during the last months of my thesis. Although the corona virus made the working situation different, you were always quick with a response to help me out. You kept me motivated and was always available for questions or some work reviews. Secondly, I would also really like to thank Erik. Your daily supervision in the first months of my thesis helped me to find this nice topic. You helped me to quickly contact as many designers as possible to get the information I needed.

I would also like to thank Vera. You were always helping me stay motivated and ready to help with my latex struggles at any moment. I also really appreciated the time you took to proofread my work. The same appreciation goes to Marianne and Lucie, thanks for the great feedback after proofreading.

Last of all I would like to thank my parents for their unconditional support during my entire study time. Thank you for always believing in me and giving me the chance to study in Delft.

Contents

1 Introduction

Sustainability is a trending topic. Almost everyday, newspapers and websites publish articles about our future energy resources and green alternatives to everyday products. Students strike for better climate policies [1] and an increased number of people is considering to drive electric or eat more meat replacements to reduce meat consumption [2]. As stated in the Paris Climate Agreement [3], the goal is to limit the global warming with a maximum of 2 degrees Celsius, preferably even below 1.5 degrees Celsius. Although the government of the United States has withdrawn themselves from the agreement, still 195 countries have agreed to lower their emissions drastically in order to reach the set goals before 2050.

One of the biggest CO2-emitting sectors is the energy producing industry. With most of the energy in the Netherlands coming from coal, oil and gas resources, see figure 1, alternatives are needed to lower the overall emissions. Where hydropower options are very limited, due to the extremely flat characteristic of the landscape, there are plenty of options for solar and wind power. Especially wind power has great potential, since the average wind speed in the Netherlands is higher than most European countries [4].

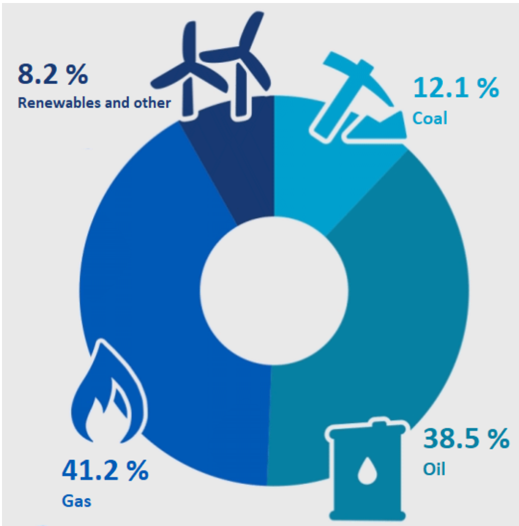


Figure 1: Energy production per energy carrier in percentages in the Netherlands in 2017. *Adapted from [5].*

Many wind turbines have been placed to harvest the energy of the wind. To maximise profits the wind farm layouts are optimized, but wind farm design is becoming more and more complex. Originally wind farms were rather small with several turbines and mostly on land. But along with the increasing need for sustainable energy production, wind farms are growing in size. Larger wind farms are deployed on sea to reduce visual and noise pollution and harvest the stronger sea winds. Yet, onshore wind farms still have a lower cost per unit of energy, but offshore wind farms will probably be cheaper within a decade [6]. This increased number of wind turbines per wind farm leads to tougher layout optimization problems, since turbine interactions are essential for calculating the energy yield of a design.

For this purpose, many different optimization techniques are investigated in literature to find the optimal approach for designing an offshore wind farm, such as genetic algorithms [7][8] and gradient based optimization [9]. Also many software packages have been developed to support designers in their task to find the best design, for example Windpro [10], WindFarmer [11] and Openwind [12]. These applications definitely support a wind farm designer, but still finding a suitable optimal design for a site can be challenging. Often some requirements cannot be included in the optimization tool or calculation times are long. The final result might therefore not always satisfy all required constraints. It might, for example, be impossible to include some constraints in the definition of the site at the start of the optimization run. This could lead to infeasible optimal results, which would be meaningless for a wind farm designer.

For that reason, in practice, the final design for a wind farm project is not the solution provided by software. Designers use programs to support their design procedure, but the final design is often requires manual adjustments. Therefore manual design should not be underestimated and it can be very interesting to see how a numerical optimization application performs against a manual design approach.

The goal of this thesis is to show the strengths and weaknesses of different wind farm design approaches. It will describe and analyze the current situation a wind farm designer is facing and thoroughly inspect the performance of wind farm design applications. First different wind farm optimization algorithms are analyzed and their advantages and disadvantages are listed. Then a wide overview of existing software packages is given to show the different tools a designer has at hand. After that, the entire design process is described to see where different tools or approaches could improve the designers' results. Lastly, a series of experiments demonstrates the capabilities of the Openwind software package. This numerical approach is compared with a manual design approach to figure out if the software can meet up to the human design intelligence.

This thesis covers the following topics. Chapter 2 summarises the different available optimization algorithms for wind turbine design. Advantages and disadvantages for the different techniques are given. Chapter 3 describes the different wind farm optimization tools in the academic and commercial world. In chapter 4 the wind farm design process is described in detail based on the experience of two designers. Chapter 5 shows the results of multiple experiments where the software package Openwind is compared to a manual design approach. Finally, chapter 6 and chapter 7 contain the discussion of the results and the conclusions based on the experiments respectively.

2 Optimization techniques

This section describes the different optimization techniques involved in the wind farm layout optimization problem. For each of the individual techniques, the advantages and disadvantages are listed to get a good view of the promising optimization strategies to tackle the wind farm layout optimization problem. The problem consists of two different parts to be optimized, the turbine positioning and the cable topology. Both parts can be optimized using different algorithms. In most literature, these two optimization processes are uncoupled, but coupling the algorithms could lead to better results. The first sections describe four different wind farm layout optimization algorithms, where the last section covers the cable topology algorithms.

2.1 Turbine positioning

This section describes the strengths and weaknesses of four different optimization algorithms: Genetic, Particle Swarm, Gradient and Pseudo gradient optimization. These algorithms have been used in many literature studies to optimize wind farms.

2.1.1 Genetic algorithms

Genetic algorithms (GAs) are inspired by Charles Darwin's theory of natural evolution. Genes of two parents are passed on to offspring, using various combinations of parental genes to create different children. Charles Darwin's theory states that the fittest parents produce the fittest offspring, leading to higher surviving chances. The evolutionary principle can be applied in many other problems, also in the wind farm layout optimization problem. Here the fitness of a layout is determined by the value of the objective function, which is often the levelized cost of energy or the energy yield.

The genetic algorithm contains a few different steps. The first step requires the initial population. In nature, to start breeding, an initial group of animals is required. In a GA this is similar, an initial population of solutions is required. Each of these solutions is tested with the objective function, which determines the fitness of every individual. The fitness determines the probability of this individual to be used for reproduction. The better the solution to the problem, the higher the chances to be selected for reproduction.

The next step involves selecting a pair of solutions to let their genes pass on to the next generation. The two selected individuals will go to the most significant step in the GA, the crossover. Crossover is an exchange of a part of each solution. A random point in the solution is selected. The part of the solution on the right side of this point in solution A is swapped with the right side of solution B, where solution A and B are the selected 'parents' for reproduction. After this crossover, two new individuals are found, which are rated with the objective function again. This new offspring is added to the population, where the worst solutions are discarded to keep the population small. A visual representation of the cross-over process can be found in figure 2.

One more process from Darwin's evolution theory is adopted. This is the process which is called mutation. Mutation in nature occurs at random. Without any direct cause, a gene suddenly changes, as shown in figure 2. This process is very important in the GA to give the algorithm the ability to search around the optimal solution, preventing it to get stuck in a local optimum very easily. So after the offspring is produced, there is a low chance for mutation. This would affect a small part of a solution.

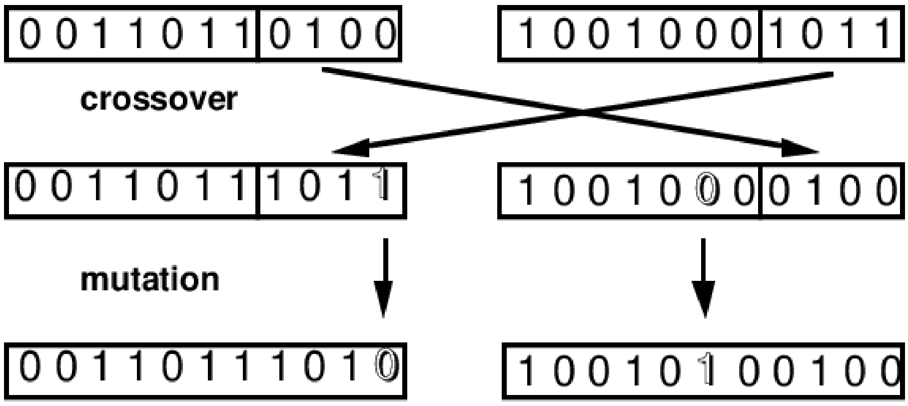


Figure 2: An example of crossover and mutation happening in the genetic algorithm. *Reprinted from [13].*

The solution is represented by a string of bits by the GA, where mutation then flips a small amount of these bits.

The first proposal for using a GA to solve the wind farm layout optimization problem has been published in Mosetti et al. [7]. To use the principles of genetics, an available terrain is split in 100 cells. These cells form a ten by ten grid. For ease, the total number of wind turbines and the turbine type is kept constant. A solution can be represented as a series of zeroes and ones, indicating if there is a turbine at a certain location or not. Each wind farm configuration is divided in ten cells of ten binary numbers. The cells are subjected to the crossover and mutation processes, leading to new solutions. Again, the higher the fitness, the larger the chance that it will be used for reproduction. Mosetti et al. [7] use three scenarios to test the GA. One with a single wind direction with constant intensity, one with multiple wind direction with one single intensity and one with multiple wind direction with varying intensities. The latter is most representative for a real life wind farm. These three scenarios are often used in later literature as benchmark to newly developed algorithms.

Grady et al. [8] continues on the work of Mosetti et al. with their own GA. With this algorithm a different optimal design is found, which is an improvement compared to Mosetti et al., mostly due to a higher number of individuals in the initial population. An even more complete study is found in Gonzalez et al [14]. This study shows a GA which can handle more complicated scenarios. For example some location in the site being unavailable for turbine placement, a more complex wake model and inclusion of life cycle costs in the objective function. This shows a GA could possibly be implemented in a very complete framework for optimizing a wind farm.

The genetic algorithm is a heuristic, which means the solution in not guaranteed to be optimal. However, the computation times are relatively short for a good quality result. Also the mutation process in a genetic algorithm allows it to discover many different areas in the solution region, making the search space quite wide. It is on the other side not so easy to define correct parameters, such as population size or mutation chances. When applying this algorithm, literature values will have to be used to start with and later on be adjusted to the specific problem. This gives the genetic algorithm the following advantages and disadvantages:

+ Good quality solution in short computation time.
+ Wide search space due to mutation.

– Cannot guarantee the optimal solution.
– Hard to define best algorithm parameters, such as population size and mutation chances.

2.1.2 Particle swarm optimization

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique inspired by the social behaviour of bird flocking or fish schooling. It has been developed by Dr. Eberhart and Dr. Kennedy in 1995 [15]. PSO is an evolutionary algorithm like a GA, it requires an initial candidate solution as input, from where it evolves to a final solution. PSO lets all solutions, also called particles, fly through the problem space to search for the optimum [16].

As said, PSO is based on a flock of birds where every solution is one bird in the flock. Imagine these birds are searching for food and there is only one piece of food in the area. The birds do not know where this food is, but they know exactly which bird is closest to the food. (This bird represents the current best solution to the objective function.) The ideal strategy would be if all birds moved towards the closest bird until they find the piece of food. Similarly, this is possible with wind farms. By using the objective function, one particle can be determined to be the current optimal solution. By moving all other particles in this direction in multiple iterations, the best solution might improve. Alike the birds, each particle in PSO has a velocity and a position. This velocity and position is updated every iteration based on their relative position to the current best solution, their own previous best solution and a random component. This velocity helps to come faster to a final solution. The PSO is stopped when a maximum error criterion is met or when a maximum number of iterations is done.

Just like GA, PSO is a heuristic, but it has a few advantages over GAs. First of all, PSO is more simple to implement, because there are fewer parameters used. Secondly in PSO every particle remembers its own previous best position as well as the neighbourhoods best value. This gives the algorithm more effective memory capacity than a GA has. Lastly, PSO is more efficient in keeping a diverse set of solutions. This is due to the fact that PSO tries to improve all solutions, where GA discards all the worst solutions. This lowers the chance of getting stuck in a local optimum [17].

In Rahmani's paper [17] the PSO is compared to the GA from Mosetti et al. [7]. The exact same wind farm is optimized using the same type of turbines. It shows that the PSO algorithm is suitable for optimizing wind farms and similar results to the GA are obtained, where the authors claim there is room for improvement and further study should show how the PSO performs on different more realistic scenarios.

According to Abdmouleh [18], PSO also has some disadvantages. It can be difficult to define the initial parameters used in the optimization. It also mentions the PSO algorithm has the downside of sometimes prematurely converging to a local optimum, trapping the algorithm. In this way an early solution is found, which is not the optimal result. The paper agrees on the earlier mentioned advantages. In summary, these are the main characteristics:

+ Can be simple to implement, because of few parameters used.
+ High effective memory and wide search area, creating better chances for finding the global optimum.
– Can be hard to define the correct settings for the algorithm.
– Can converge prematurely, getting trapped into local optimum.

2.1.3 Gradient optimization

Gradient optimization is based on the gradient of the objective function. Gradient search methods originate all the way back to 1847, when Cauchy proposed the idea first [19]. The gradient is the derivative of a multi-variable function. It is a vector pointing in the direction of the greatest increase of a function. The size of the vector shows the magnitude of the rate of change in a certain direction. This gradient forms the basis for the gradient based optimization. Gradient based is often referred to as gradient descent optimization, since most algorithms try to minimize a cost function. The mathematical notation of the gradient is shown in equation (i).

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{i}$$

The gradient is applied in an algorithm in the following way. A starting point is selected, where the gradient is determined. This gradient will point in the direction of largest increase. Since the cost function is minimized, the next step is taken in the opposite direction of this gradient. The size of this step is depending on the magnitude of the gradient and the set learning rate and is determining the speed and accuracy of the algorithm. The larger the learning rate, the bigger the steps during the optimization are. This leads to missing out on some detail. On the other hand, too small learning rates will keep the algorithm searching for a long time. The difference in learning rates is shown in figure 3 [20]. Finding the correct algorithm settings can be difficult as the ideal settings can be different for each problem.

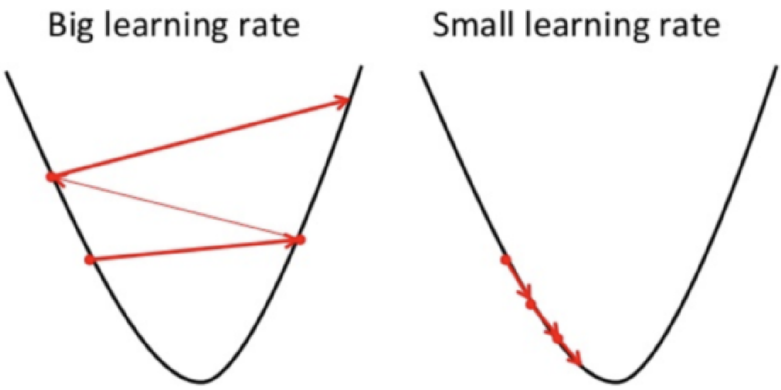


Figure 3: Two examples of the learning rate in a gradient-based algorithm. *Reprinted from [20].*

When the cost function is not decreasing anymore with extra steps, the optimization is finished and a (local) optimum is found. Multiple starting points should be used to find the global optimum in a complex problem.

The gradient algorithm is often explained with the following example: Imagine you are on a very foggy mountain, where you cannot see the surface or the surroundings. Your goal is to reach the bottom of the mountain (minimum). All you can do is measure the steepness of the mountain at your location using your, rather slow, instrument. This measurement will indicate in what direction you are descending downwards, and this is where you will move until your next measurement. Each measurement represents the gradient function and the distance between measurements indicates the step size. This analogy also shows that too small step sizes will never get you at the bottom, since you are only measuring and not moving, while too large step sizes will lead you to miss the bottom and climb another nearby mountain.

In a paper by Stanley and Ning [21] a complicated wind farm layout optimization is solved using gradient based algorithms. The article states that although gradient-free algorithms may be superior in finding global optima, their computational expenses rise dramatically for more complex problems compared to gradient based algorithms. As gradient based algorithms scale well with problem complexity, it might be very useful for larger offshore optimization problems compared to other algorithms. As shown by Stanley and Ning [21], Gradient-based algorithms scale much better with increasing number of design variables, allowing the authors to successfully optimize a wind farm with 32-60 wind turbines and up to eighteen additional variables and two different turbine types. This all shows the following advantages and disadvantages of the gradient based optimization:

+ Can be simple to implement.
+ Computation time scales well with complexity, allowing gradient-based algorithms to solve complex problems.
– Other algorithms might be better in finding global optima.
– Can be difficult to select correct algorithm settings.

2.1.4 Pseudo gradient optimization

Another different optimization approach is the pseudo-gradient optimization. As the name suggests, it is similar to gradient optimization, but it is not identical. Just like the gradient optimization methods, the pseudo gradient optimization (PGO) is based on vectors. For each wind direction, the power deficit of a downstream turbine due to an upstream turbine's wake determines the size of a vector. The direction of this vector is determined by the line between the two turbines. When taking the expectation value of this vector over all wind directions a pseudo-gradient is found, which can then be used for a gradient-based algorithm.

In Quaeghebeur's algorithm [22], two different versions of these pseudo-gradients are used. One in which the upstream turbine is pushed back from the downstream turbine, and one in which it is the other way around, so the downstream turbine is pushed more downstream. The PGO algorithm uses an adaptive learning rate, to speed up the calculations while keeping good quality solutions. It repositions turbines each iteration according to the most optimal pseudo-gradient. When turbines violate any constraint they are corrected at the end of the iteration by moving them to the nearest feasible position.

The pseudo gradient optimization applied in the current fashion is leading to medium quality solution with very short computation times. Due to the short computation times it is very suitable for interactive optimization. The first exploration could possibly be done with the PGO where a final layout can be generated from a prepared initial layout using a different algorithm. In short, the advantages and disadvantages are listed:

+ Very short computation times, faster than normal gradient-based algorithms.
+ Medium quality solutions, suited to use as initial solution in more complex algorithms.
− Medium quality solutions, might not be good enough.

2.2 Cable topology

So far, all discussed optimization techniques have been about the positioning of the turbines: Where does one place the turbines to maximise the energy yield of a set-up? However, next to the optimal placement of the turbines, the optimal cable connection can influence the profitability of a wind farm project. Obviously, the shorter the cable length, the lower the material costs and energy losses are, but also the thickness of cables has an impact on the total costs.

Most cable layout optimization algorithms have a heuristic approach. A heuristic sacrifices optimality, accuracy and precision for speed. This leads to the algorithm not always finding the global optimum, but working must faster. In this way, very complex problems which are unsolvable due to the large amount of combinations can be tackled. Where the computation time would be really high to test all possible cable connections, a heuristic approach can be used to reduce the computation time to a useful level. The resulting solution is still a bit off of the optimum solution, but the solution is of good enough quality to be used. Examples of heuristics shown in previous sections are the genetic algorithm or the particle swarm optimization.

In his work, Katsouris, described two different algorithms: The Esau-Williams (EW) [23] and the Planar Open Savings (POS) algorithm [24]. Both of these are heuristic approaches. The Esau-Williams heuristic works for the capacitated spanning tree problem, which has the aim to connect locations in the shortest possible way. The Esau-Williams heuristic finds suboptimal solutions, which lie very close to the exact solution. Katsouris writes that on average EW still performs better than any alternative algorithm. In Katsouris' work the EW algorithm is used for creating branched designs. The POS algorithm on the other hand creates radial designs. It starts by connecting all turbines to the substation and then greedily searches which changes lead to lower cost. This in the end leads to a close to optimal solution. The POS algorithm also avoids any cable crossings in the layout, which usually is a construction constraint. Katsouris concludes with a hybrid solution using the advantages of both different algorithms.

In conclusion, multiple cable layout algorithms show potential and possibly the combination of multiple strategies leads to the best design. The most important characteristic is shared between all algorithms: they have a heuristic behaviour. This means that the final results will almost always be suboptimal, but the computation times are low. Luckily the heuristic algorithms have shown to get very close to the optimal solutions [25].

3 Existing software

This chapter covers the different existing wind farm optimization tools. The chapter consists of two parts: The commercially available software and the academic software. The first part of the chapter shows three different commercially available models, which can all be used to design and manage wind farms. These tools are used by companies to design and build wind farms worldwide. One of these tools is used to perform experiments, this will be described in chapter 5. Secondly, three different academic tools are explored. These tools show what different aspects of the wind farm optimization problem are currently explored in the academic world. A broad view on the available software leads to a better understanding of the current situation a wind farm design is facing.

3.1 Commercially available software

This section covers three different commercially available wind farm design applications: WindFarmer, windPRO en Openwind. These tools are representative examples of tools used by designers to design wind farm layouts. All three tools are analyzed and the advantages and disadvantages of each tool are discussed.

3.1.1 WindFarmer by DNV GL

DNV GL is a global company situated in over 100 countries. They are a provider of risk management and quality assurance services to the maritime, oil and gas, and power and renewables industries. DNV GL sells their windfarm design tool as WindFarmer:Analyst[11], which will be, from now on, referred to as WindFarmer. WindFarmer is build to support the design procedure. The different options are offered in a workflow, leading you through all the steps of a wind energy production assessment. In this way, wind farm designs can easily be done repeatedly without missing out on any steps.

WindFarmer is clearly built for commercial users, leaving out some useful options for experimenting. For example, generating a simple wind climate can be quite tough as the application requires WAsP models [26] or longer time series. These time series can, however, be analyzed and corrected with WindFarmer, making it ideal for importing on-site measurements for a wind farm site analysis. WindFarmer has included validated wake models and allows for loading different wake models. WindFarmer is not very suited for creating optimized design using just the application. WindFarmer has included some optimization procedures, but those are mainly focused on tweaking an existing design. Generating layouts from scratch has limited options and WindFarmer is clearly not built for that purpose. The application is meant for adjusting and analyzing a wind farm design, with the possibility to generate attractive and detailed reports of a set-up. The advantages and disadvantages of WindFarmer are listed below:

- \+ Clear interface.
- \+ Includes validated wake models with options to add other wake models.
- − Limited options for optimization.
- − Hard to set-up simple wind climate.

3.1.2 windPRO by EMD International

windPRO is a software package created and sold by EMD International[10]. EMD international is a global software and knowledge center situated in over 100 countries, providing software for preand post-construction of wind farm projects. An extensive analysis of the software can be found in appendix A. To briefly summarise the finding in this analysis, the most striking characteristic of the application is that it is very visually oriented. The graphical interface is very clear in its use and all calculations are done by loading different modules. Also these modules can be used to generate visual renders of the environment, containing the newly placed wind turbines. This is something not found in WindFarmer or Openwind.

The possibilities for optimization are quite large. Wind farms can be optimized using regular patterns or random patterns. Also optimizations based on noise productions are possible. Noise on critical locations will be minimised while maximising the energy output of the wind farm. The main disadvantages come from some unclarities about the software. There is no easy way found to insert a wind climate into the application. The application was able to load WASP files, but entering a simple wind rose by weibull parameters was unsuccessful. Next to that the use of other wake models or cost parameters was not possible. The advantages and disadvantages of windPRO are listed below:

+ Contains multiple optimization options.
+ Includes many possibilities for visualisation.
– Hard to create a basic wind climate.
– Cannot enter other wake models or cost functions.

3.1.3 Openwind by UL

Openwind is a software package created and sold by UL[12]. UL is an independent advisory, testing, inspection and certification body for a broad range of industries. Openwind looks very basic when opened. Unlike windPRO it is not graphically oriented and it is really focused on the computation instead of the looks. It is for example not possible to create a visual render of a design. The software does not support the design procedure with a workflow like WindFarmer and therefore it is not very accessible for beginning users. UL does offer good tutorial videos and a clear manual to overcome this initial problem.

One of the main advantages of Openwind is the option to optimize turbine positions and cable topologies at the same time. There is a wide range of optimization options available. Different wind wake models can be selected and the cables and access roads can be included or excluded from the optimization. On the other hand, Openwind only has one specific optimization algorithm available, which is also a bit of a black box. Not much about the optimization algorithm is known and no other algorithm can be used. This makes it less flexible for testing different approaches or selecting another algorithm if the used algorithm gets stuck in a local optimum. A positive quality from Openwind is the easily created wind climates. It is very easy to generate a wind climate based on weibull parameters in a wind rose. This is useful for quickly assessing different situations on a site.

A downside to Openwind that it is quite hard to load any land maps or background imagery. Where other tested applications provided simple options to load a land map in the background, Openwind makes it hard to load simple land maps due to the usage of different EPSG projections. In the end, it can be worked with, but it is not easy to quickly check something on a different site than the loaded sites. A big advantage in Openwind is the endless possibilities to control the used parameters. It is easy to adjust any parameter in the application, where the default values are correctly set for the less experienced user. There are multiple wake models available, and other wake models can be imported. Standard turbines are included, but new turbines can be imported or created as well. Also all costs of components can be individually edited and the optimizations can be stopped at any time. The advantages and disadvantages of Openwind are listed below:

+ Can optimize turbine positions and cables in parallel
+ Wind climates can be set-up using weibull parameters
+ All parameters can easily be adjusted
– Background imagery is hard to correctly input into the application
– The used optimization algorithm can not be changed

3.2 Academic wind farm design tools

In this section three academic tools are analyzed. The investigated applications are: the Pseudo Gradients Optimizer [22], WINDOW openMDAO [27] and TOPFARM [28]. This paragraph gives an outlook on these tools and shows the development in the academic world, which might later be used in commercial applications.

3.2.1 Pseudo Gradients Optimizer

The first tool has been developed by E. Quaeghebeur. This tool works with the concept of pseudo gradients to find the optimal wind farm layout, as described in section 2.1.4. He investigated this concept, because it could lead to very short calculation times with medium quality layouts.

The application, written in Python, is easy to use and operate. The module uses .yaml files, which are basic text files, to load all settings. The problem defining file contains all required settings for a wind farm optimization run. The site boundaries need to be given, which can be circular or polygonal defined by linear constraints. The module also handles different separate zones and exclusions can be included, for example when an archaeological shipwreck is located somewhere in the perimeter. Next an initial layout can be served to the script or a random starting layout can be generated based on a set number of turbines with a given pattern.

Also important for wind farm optimization problems are the wind conditions. In the pseudo gradients optimizer, the number of sectors in the wind rose can be changed easily. When this number is decreased, it speeds up the calculations because fewer wind directions are analyzed. On the other hand, a more detailed wind rose can lead to more realistic results. Another important parameter is the turbine type. The size, height and model of a turbine have a huge impact on the optimal layout and the corresponding energy yield. In this tool only a single type of turbine can be used during an optimization run.

The last required settings are the objective function and the wake model. Different wake models can be used: the Jensen model [29] and a few variations on that are included. Since the application is open-source, easily other models can be added as well.

The pseudo gradients optimizer is limited in the available options, since its focus is to investigate the use of pseudo gradients in the optimization process. If the use of this optimization strategy can lead to good results, it can definitely be interesting to use in other applications because of the very short computation times. This application is ideal for the verification of new optimization algorithms, like the pseudo gradients.

3.2.2 WINDOW in openMDAO

'WINDOW in openMDAO' is a tool created by Sebastian Sanchez Perez-Moreno. The tool is build on openMDAO, which means Open-source Multidisciplinary Design, Analysis and Optimization. openMDAO is an open-source high-performance computing platform for efficient optimization [27]. WINDOW is described as a MDAO workflow meant to support the design of offshore wind farms.

WINDOW is unique in the sense that it can use different models in the same disciplinary module. So within one optimization run, multiple different optimization algorithms can be combined. WINDOW also combines turbine positioning and cable topology optimization into one optimization procedure. Some models are included in WINDOW and other models can be added as well. This tool is ideal for testing the impact of integrated optimization compared to a sequential approach.

3.2.3 TOPFARM

TOPFARM is a Python package developed by DTU Wind Energy to help with wind-farm optimizations. It is easily installed using the Python installation manager PIP [30]. TOPFARM is, like WINDOW, based on the openMDAO framework [28].

Recently, a second version of TOPFARM was released, which includes most of the common wake models and cost functions. Therefore different optimization strategies can easily be tested and compared. TOPFARM is not just suited for optimizing layouts, it can also select the ideal turbine for a set-up or optimize wind farms with multiple turbine types.

TOPFARM is ready for use for tackling different optimization problems, just like a commercial package. There are many options included in TOPFARM. It includes for example different optimization algorithms, different wake models and the possibility to include cable and access roads in the optimization procedure. The documentation is quite elaborate with many examples, allowing users to quickly get started. Since the application is written in Python, changes can be easily made and extra features can be added by the user. TOPFARM is a great environment for testing in an inclusive optimization tool.

4 The designers' process

4.1 Introduction

In this chapter the designers' process is described. This chapter will help to understand what decisions a designer has to make during the design of a wind farm. The design process is split in three steps: the preliminary investigation, the turbine selection and the design phase. Details on each step can be found in section 4.2, section 4.3 and section 4.4 respectively. The information in this chapter comes from a literature review as well as the input from two designers from an interview. This chapter is the interpretation of the interview and the gathered information from literature. The interview questions can be found in appendix B.

4.2 Preliminary investigation

The preliminary investigation is the phase of the design process where all information, which can be useful in the design phase of the wind farm project, is gathered. The preliminary investigation includes many different aspects. The first step is to find a suitable location for a wind farm. Usually this is not the task for a designer, but they get a request for a wind farm design on a designated location. Although this step is often out of the hands of the designer, picking the wind farm site is of course an influential step in the entire design process.

Selecting a suitable site is quite different onshore and offshore. Onshore there are a lot of different scenarios. First of all large wind farms are built in remote areas. This way there will be almost zero nuisance, since there are few people living close to the site. These projects are often government controlled to meet up to their renewable energy targets. For instance in the Netherlands, this holds for wind farms larger than 100 MW [31](in Dutch). Also very small wind farms can be constructed, in farm lands for example. These are usually privately owned by the farmer or a small company. In both cases it is very important that for onshore construction permits from the local governments are required, which can be hard to acquire due to the visual pollution of the wind turbines.

In contrast to onshore wind farm design, site selection is much more limited offshore. In the Netherlands offshore wind farm construction is regulated through request for tenders (RFTs). A RFT is a procedure where different interested developers can make a blind offer for a project. The best offer will win the RFT and get the right to build the project for the offered conditions. For offshore wind farm construction, different allocated areas are tendered in stages. The Netherlands Enterprise Agency organises the RFTs for those areas in the Netherlands [32]. The required information to participate to a RFT is offered online, so a company can do its own analysis and participate in the RFT. The government also offers subsidies for some of the offshore locations. Recently the first two unsubsidized RFTs have been completed successfully [33] (in Dutch).

After a location is chosen, a lot of data is acquired to analyze the location. This data helps to determine the profitability of the wind farm site. Many different aspects are included. First of all, soil investigation is required to calculate the construction costs for the supporting structures and access roads. Next to that, it is very important to have detailed wind information for this specific site. Ideally this wind information is available for a longer period of time, so accurate estimations of the yield can be made. For offshore RFTs, the bathymetry and wind information are offered to all participating instances. Of course, locations with higher average wind speeds are favoured over other locations, but the economic profitability is in the end the determining factor for the construction of a wind farm.

Next to the measurements, it is also important to know in this phase of the project what the limitations to the project are. A designer requires knowledge about site boundaries, noise restrictions, electrical connections and the required amount of wind turbines or rated power. Using all this information a designer quickly makes some initial calculations. These calculations are not very detailed yet, but are used to estimate the yield and costs of the project. In this stage of the project wind farm optimizations are supposed to run within an hour, so in short time many options can be explored [34]. It is not necessarily important to have a perfect design yet, as long as the uncertainties are known. It is important to have this estimation quickly, so other people in the company can start working with the information of the design, as a multi megawatt wind farm project team consists of many more people than just a designer.

4.3 Turbine selection

There are many different turbine manufacturers constructing different types of wind turbines. To be able to make a proper choice for a project, a selection has to be made. This leaves a designer with the task to select the proper turbine for a design. The wind turbine market is quite innovative, therefore sometimes designers have to work with models which will be released soon. Usually it is best to select the newest models, as the technology keeps developing. On the other hand, sometimes the older turbines can be bought with a discount, since the manufacturer needs to get them out of his supply. Next to size and power production, a second selection criteria can be the maximum possible size and noise production due to the restrictions of the project. It is also important to fit the size of a turbine to the wind conditions on site.

Typically only one turbine type is used in the optimization procedures. This turbine needs to be selected beforehand, but parallel calculations can allow for comparing different turbines. Other more complicated optimization algorithms can combine different turbine types in one optimization procedure, which might be an option to create more efficient wind farms. The choice for a wind turbine type also influences the total number of required turbines. One will need more small turbines to match the production of a park with larger turbines. Also the electrical connection of an (offshore) site can limit the total possible power production. All these factors combined will make the designer come to a selection of possible turbine types to use in the design.

4.4 Design phase

After preliminary investigations are done and only a few turbine options are left, the design phase starts. This is the phase where the wind farm has to be designed for construction. At the end of this phase a design should be ready to be constructed. The designer has two main design goals. First, the localisation of the wind turbines and the corresponding required supporting structures and second the electrical connection of the wind turbines to the electrical grid.

For designing the wind turbine layout, different strategies are possible. First of all there is regular pattern design. In this type of designing, turbines are lined up in several rows with equal spacing. This regular pattern is then modified to optimize the energy yield of the set-up. Software packages can help to quickly test different cases of the regular set-up. Secondly, it is possible to let the regular pattern go and place turbines in an irregular lay-out. Theoretically these layouts have higher possible energy yields, but they can be impractical in the construction phase. The random layouts are generated with help of optimization software, which uses an algorithm to find the best possible set-up. In practice, these applications produce a design, which is modified by the designer to fit to all conditions. This is due to the fact that the optimization software often does not include all factors in the optimization process. For example visual impact can be something the designer has to judge and if this impact is too big the design has to be modified accordingly. In this final stage of the wind farm layout design, the optimization is allowed to take up to a week of run time [34]. It is, however, important to realize that there can be a lot of time pressure on a designer.

It is also possible that wind farm layouts are not optimized by software packages at all. Sometimes there are so many conditions which have to be met, that the optimization software almost has no options. The software is then only used for evaluating the yield, visual impact and noise impact of a design. This is more likely for onshore design, where visual and noise restrictions often have a bigger impact. For example Windpark Fryslân was designed using this approach [35]. Software was used for two reasons: To check if all legal terms were met for shadow flicker and noise and to compare different hub heights and turbine types [36].

Besides the localisation of the wind turbines, the turbines have to be connected to the electrical grid. The need for the cable connections can also influence the optimal wind farm design. The larger the total cable length is, the higher the costs, so designers try to minimise the cable lengths. Also prices increase on cable thickness as more material is required. For a regular pattern design, the cable connections are usually simple to make, as the turbines are lined up already. For more complex patterns, the ideal cable connection can be harder to determine, and optimization software can be used to find the best topology. It is important for a designer to keep track of the installability of a cable design. Cable layering ships have a certain rotational circle and the trencher (hole digging machine for laying electrical cables) cannot pass all slopes on the seabed.

Some software tools can also optimize the turbine positions and the cable topology simultaneously. By using the total costs of electricity as objective function, the best combination of wind turbine positioning and cable connection is found. Optimizing both measures at the same time can lead to better results, but it will also require more computational resources. Not all software is yet capable of combining these aspects in one optimization procedure, as can be read in chapter 3.

It is important to remember that there are always uncertainties in the final design. There are many variables with large uncertainties, such as year to year energy yield, price of electricity, costs of steel, copper and oil. This means that there are always possibilities for improvement when these variables change, therefore it is good to test the sensitivity of a design to the different parameters. When the design stage ends, a final design choice is made and it is ready to be constructed.

5 Comparison between manual and computational wind farm design

In this chapter the industrial software package Openwind [12] is compared with a manual design approach. This software package is chosen because it was the most complete available wind farm design tool. The comparison between Openwind and manual design allows one to find the strengths and weaknesses in both approaches. In section 5.1 the approach for the experiments is explained into detail. Section 5.2 shows two small experiments to first test the behaviour of Openwind, after which the experiments can be set up. Then section 5.3 and section 5.4 show a comparison of manual design and automated design with Openwind with and without cable topology design. Finally, section 5.5 and section 5.6 show experiments to test the effectiveness of automated design with a different turbine density or a scattered wind farm site. All experiments are structured in the same four paragraphs: Aim, Set-up, Results and Conclusion.

5.1 Methodology

In this chapter wind farms were designed with two different approaches: A manual design approach and a computed design strategy. These approaches would both design the same wind farm, so the comparison on performance could be made. All scenarios are introduced at the particular experiment, but the general outline was mostly the same. The location of the designed wind farm was in the North Sea and wind conditions representative for that location were used. This particular area has been chosen since there are a lot of wind farms built in that region, which thus gives realistic conditions for a test case.

For all scenarios a 3.0 MW wind turbine model was used, which is standard included in the Openwind library. The park in the experiments consisted of 50 of these 3.0 MW wind turbines. The model for the wind turbines was the Alstom ECO 100 3.0 Class 1A. This turbine had a rotor diameter of 100 m and a hub height of 90 m. More details can be found in appendix C. The area of the wind farm was based on the turbine density in an existing wind farm, wind park 'Luchterduinen'. This is an offshore wind farm in the North Sea, with 3.0 MW turbines. Using the wind turbine density of 0.58 3 MW-turbines/km^2 of 'Luchterduinen' led to a square area of 5.4 km x 5.4 km for the test site. The total area of the test site was 29.16 km^2. The left bottom corner of the area was located at 52°25' N, 4°10' E, which translates to [579345.17,5808022.92] in the ESPG 32361 WGS 84 coordinate system loaded in Openwind. This site was used for all experiments with a square site.

5.1.1 Manual design

The manual design approach was, as it suggests, done mostly by hand. However, the Openwind software was used as well. For this approach, decisions were made based on the wind conditions and shape of the wind farm site. The rationale for each decision has been explained at the particular designs of each experiment. When a design was made, it was imported into Openwind and tested on its performance. Openwind can point out individual performance of each turbine and in this way the worst performing wind turbines could be identified. These turbines could then be moved to improve the design. This energy capture calculation was also used to identify the best layout if there were multiple manual designs for the same experiment. It could be discussed if this approach is manual, since the software was used to some extent. The computed approach did use optimization algorithms instead of just energy capture calculations and therefore this approach was considered manual.

5.1.2 Computed design

As described in the manual design section, the computed approach used Openwind to optimize the turbine placement or cable layout. This optimization was tested by performing test experiments to identify some essential behaviour of the algorithm. After these tests, most settings were kept the same at all experiments, unless mentioned otherwise. When the experiments were completed, again the energy capture function was used to determine the energy yield of the calculated design.

After an optimization was completed or a manual design is constructed, the energy yield was calculated using the relatively fast Jensen wake model. In this thesis a more complicated and therefore probably slower wind wake model would not make a difference as the Jensen wake model shows to be sufficient in many other cases in literature to get a good view on wake effects in a design. Examples of such studies are Mosetti et al. [7] and Grady et al. [8].

5.1.3 Scenarios

In order to analyze the difference between the manual and computed design strategy, different scenarios were explored. Before these scenarios could be executed for the computed design approach, the algorithm had to be tested. Therefore two tests were done to define the correct parameters to start the optimization runs in the various experiments. These tests are described in section 5.2.

Four different experiments were used to compare the performance of Openwind and the manual design approach. These scenarios were required to find the strengths and weaknesses of both approaches, since the performance can be effected by various factors. For each experiment only small adjustments were made, so the cause of the change was always very clear and cannot be caused by multiple variables. The details of each experiment are introduced at the particular section. The different experiments were as follows:

- Turbine positioning in different wind conditions
- Turbine positioning and cable topology in different wind conditions
- Turbine positioning and cable topology with different turbine densities
- Turbine positioning and cable topology in more complex wind farm sites

5.1.4 Runtime

Before any experiments could be done, the runtime for the experiments had to be determined. Since it is never known if the optimum was reached in a wind farm optimization problem, the simulation could not simply run until completed. To determine a fitting runtime, the convergence of the array losses, which is similar to the energy capture, was used as can be seen in figure 4.

Figure 4 showed that the optimizer already finds a reasonable optimum after a short runtime. This can be seen in the flattening of the array losses curve. The array losses in a wind farm follow the same trend as the energy yield of a wind farm and can therefore be used as an indication of the progress of the optimizer. When extending the total number of iterations to 10,000, the optimum had only small changes in the last 5,000 iterations. This test ran for about 5 hours on a regular desktop computer. This test case proved that 10,000 iterations were enough to get a good indication of what the optimal layout would look like, as the energy capture almost showed no increase any further. For all further experiments, at least 10,000 iterations were performed. This ensured that the improvement

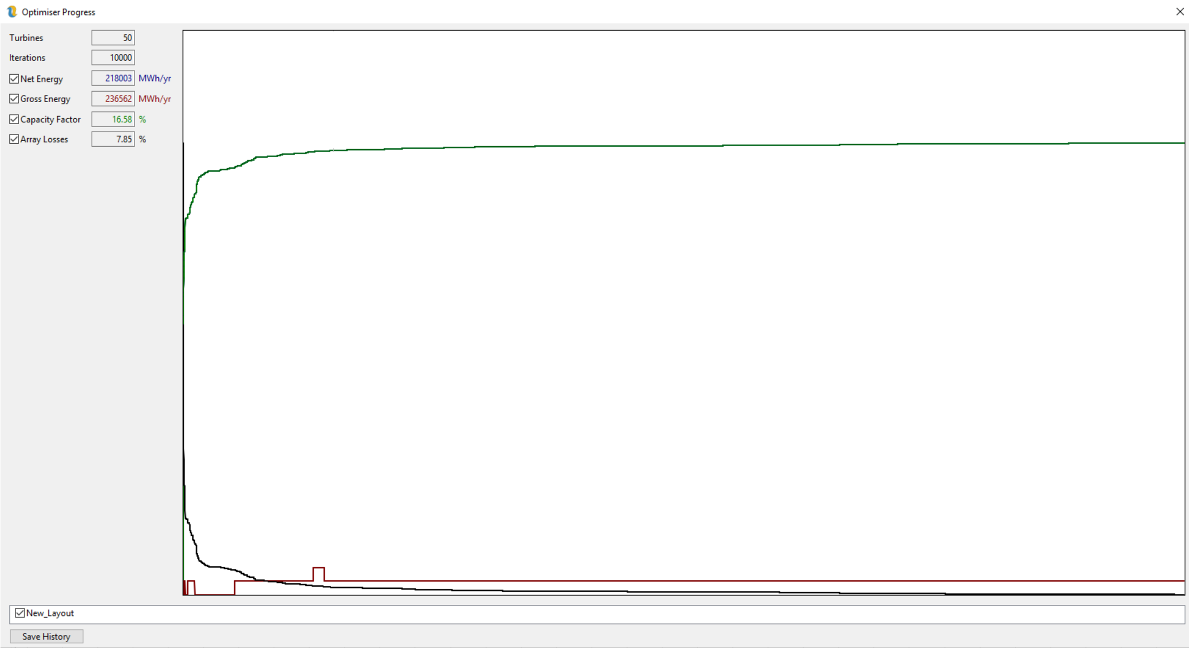


Figure 4: The progress of the optimizer in Openwind. The energy yield is not shown, but the array losses follow the same trend.

of the energy capture was more or less stabilized and no large changes were missed by stopping the optimization too early.

5.2 Assessment of Openwind behaviour to set-up the experiments

5.2.1 Aim

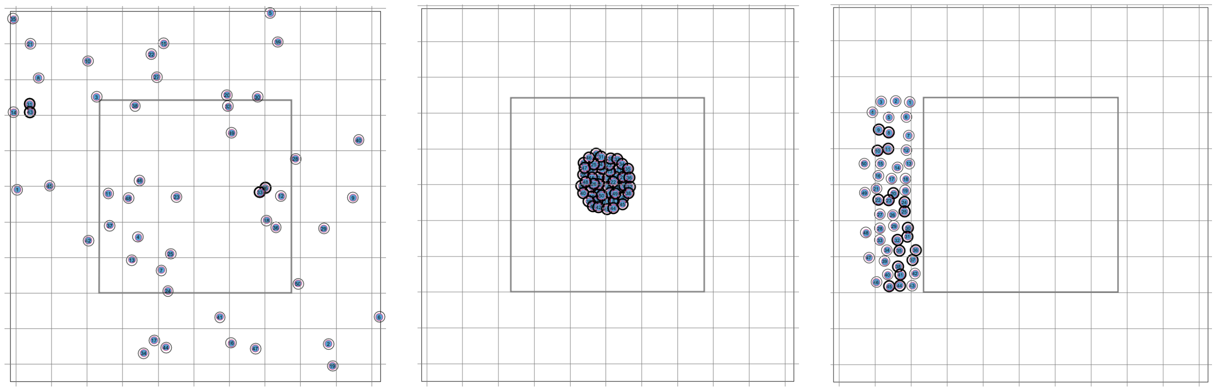
To get better knowledge about the optimization algorithm used in the Openwind software package, two small tests are executed to determine some key characteristics of the algorithm. The results of these tests will determine how the successive experiments are set-up. The two tests consider the following two topics:

1. Initial layout
2. Repeatability

The first test is to measure the impact of the starting position of the turbines on the final layout. Ideally all starting layouts lead to the same final result, so the initial positions can be randomised at the start of an experiment by the software. The second test should show if a repeated experiment with the same starting conditions leads to identical results or at least a similar result. If this is not the case, all experiments will have to be conducted multiple times to get trustful results. This would be time consuming, allowing a lower number of different ideas to be explored.

5.2.2 Set-up

The used wind farm site contained 50 wind turbines and ranged 5.4 km x 5.4 km as described in the previous section. Starting with the first test, three initial set-ups were compared as shown in figure 5. The random initial layout, generated by Openwind was compared with a hand-made layout with all



(a) *(b)* *(c)*

Figure 5: Three different initial layouts to evaluate the algorithm. **a:** 'Random': Randomised initial turbine positions. **b:** 'Center': All turbines are initially placed in the center of the site. **c:** 'Left': All turbines are initially placed at the left side out of the site.

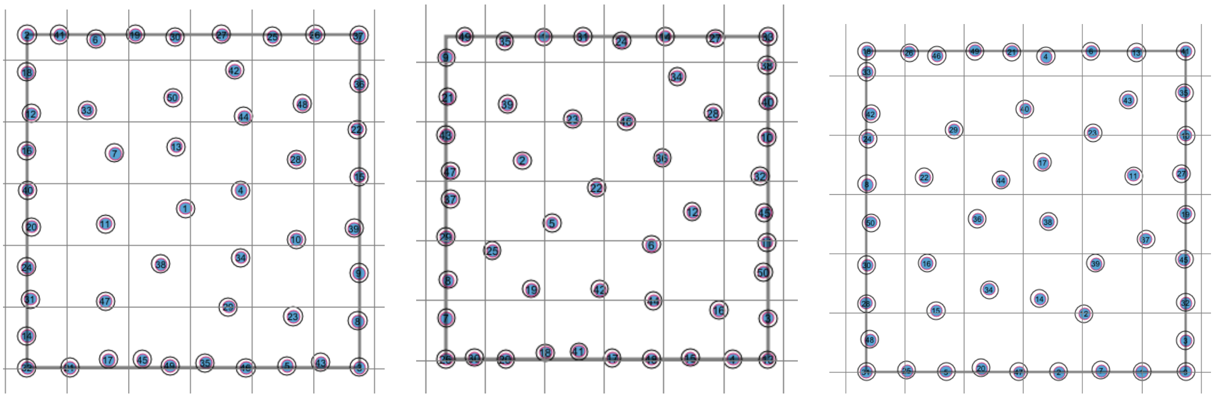
turbines in the center or a handmade layout with all turbines placed left of the available site. The layouts were named 'Random', 'Center' and 'Left' respectively. The inner square in the pictures depicts the available area, where the outer square marks the boundaries of the wind map. The wind map needed to be larger than the site to avoid any boundary issues, as stated in the Openwind application. In all situations the wind rose was very simple, the wind was spread evenly over all directions with an equal wind speed distribution. This means the wind rose consisted of 12 equal parts, each with a range of 30°. The weibull parameters were left unchanged in Openwind, and are $A = 6\,\mathrm{m\,s^{-1}}$ and $k = 2$ by default.

For the first test, the three initial layouts from figure 5 were optimized by Openwind on their energy capture. Each layout has run for 10,000 iterations. This takes about 5 hours on a standard desktop. After 10,000 iterations, the energy capture of the different final layouts as well as the turbine positions were analyzed. The second test consisted of three runs of the same 'Random' lay-out as test one. These three runs were also executed for 10,000 iterations. Comparison of the energy capture and turbine positioning of the final layouts will make clear if different runs lead to identical or similar results. Only the 'Random' starting layout is optimized three times, as this is the preferred setting to use if the results turn out to be similar each run.

5.2.3 Results

The energy captures are listed in table 1, the different final layouts of the first test are shown in figure 6. By just looking at the energy capture of the different runs, the results are quite similar. The difference between the highest energy capture ('Random') and the lowest energy capture ('Center') is less than 0.1 %. Since this difference is reached after a relatively short run of just a few hours and 10,000 iterations, it can be concluded that the initial layout or the repetition of the test does not lead to significant different results in terms of energy produced.

Although the layouts look very similar at first, they are totally different. No turbine is placed at exactly the same location as another, but the overall properties of each layout are comparable. There are a few explicit differences to point out. The lay-out with the smallest energy capture, 'Center' (table 1), has an unoccupied corner on the left top side (figure 6b).



(a) *(b)* *(c)*

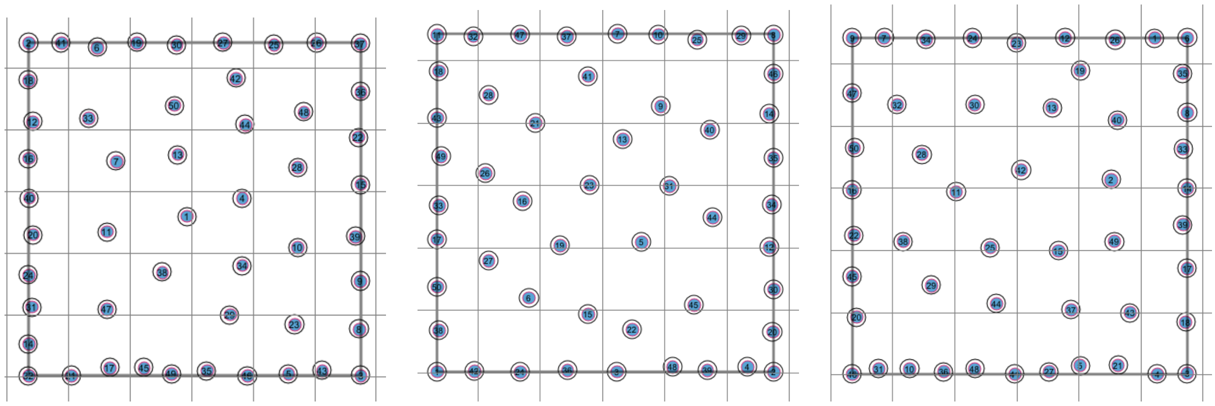
Figure 6: Final design outcome from the different initial layouts after 10,000 iterations. **a:** Started from 'Random' layout. **b:** Started from 'Center' layout. **c:** Started from 'Left' layout.

Table 1: Annual energy capture (GWh) of different designs. The columns show different initial layouts and the rows show different repeatability tests. All tests were stopped after 10,000 iterations. Only the 'Random' experiment has been conducted multiple times.

Run	'Random'	'Center'	'Left'
1	223.900	223.799	223.876
2	223.879	-	-
3	223.801	-	-

The unoccupied corner seems a disadvantageous characteristic, since all other layouts have higher yields with the corner filled. This event can be subscribed to the randomness in the optimizer and might disappear if the runs would be extended to more iterations. Furthermore, it is clear that the optimizer prefers to place a lot of turbines on the edge of the wind farm. The difference in the number of turbines on the edge of the site between the 'Center' layout (figure 6b) and the other two layouts (figure 6a and figure 6c), is exactly one, 33 to 34 turbines respectively. This explains the open corner on the left top side, as there is one extra turbine placed on the edge of the layout of the 'Center' test. It is assumed that the issue will be resolved most of the times in longer runs, but it is something that can happen with the other experiments as well. It could also be a trap in the algorithm. To verify this, a wind turbine is manually moved to the corner and one other to the center. After this the optimization process is resumed and a better optimum is found within 500 iterations. It is therefore good to keep in mind that possible strange outcomes of experiments could be subscribed to a too short optimization run, or simply the algorithm getting stuck in a local optimum. In conclusion, the different initial layouts have no clear impact on the outcome and it is safe to use the random option in Openwind to perform any experiment.

In the second test, three separate 'Random' test runs are compared, shown in figure 7. While looking at the numbers in table 1 the differences were very small, visually the layouts differ quite a lot. First of all, the number of turbines on the edge range from 32 to 33 and furthermore the turbines are also not equally spread over the edges in every set-up. The number of turbines per edge is not necessarily equal, figure 7b shows 9 turbines on each edge, but both the other layouts (figure 7a and figure 7c)



(a) *(b)* *(c)*

Figure 7: Final design outcome from the 'Random' initial layout after 10,000 iterations. **a:** Run 1. **b:** Run 2. **c:** Run 3.

have the number of turbines on the edge varying between 8 and 11. Also the center areas are not identical, there is just a random spread of turbines with a more or less regular spacing. It is difficult to draw conclusions from this; apparently the exact situation in the middle has no big impact as long as the turbine spacing is large. The similarity in the energy yield between the solutions gives reason to think that the found solutions are local optima close to the global optimum. Since the results are all so close in terms of energy yield and show similar patterns, they are not expected to cause any problems in further experiments.

5.2.4 Conclusion

Two tests have been conducted to assess the algorithm used for optimization in Openwind. All experiments were run on regular desktop computers, for a maximum of 8 hours. This time was found to be long enough for the software to get close to its optimum and show its behaviour. The first test led to the conclusion that the results are not heavily depending on the initial layout after at most 10,000 iterations. The 'Random' initial layout option (figure 5a) within Openwind can therefore be used in all further experiments. The second test shows that different runs in the same conditions lead to different results due to a random factor in the algorithm. All experiments are only ran once. Different optimization runs within an experiment are always executed in the same runtime. This can lead to small differences due to a different number of iterations, but these differences have a small impact as the optimizer is close to its optimal value after about five hours of runtime.

5.3 Experiment 1: Wind farm design in different wind conditions

In this section an experiment will be discussed where three different wind roses are used to design a wind farm. Computational designs are compared to manual designs in performance and appearance.

5.3.1 Aim

When designing a wind farm for a particular location, the wind conditions are often known in advance. These wind conditions heavily influence the ideal design of a wind farm. This experiment is used to determine if the Openwind package will perform different when using various wind conditions. It could be that the complexity of the wind rose on the site leads to different quality results. It is also interesting to see what patterns come up in the designs of the software and if this is similar to what a manual approach produces.

5.3.2 Set-up

For this experiment the same wind farm was used as before. A 5.4 km x 5.4 km square area in the North Sea with 50 3 MW turbines to be placed. This time around, however, three different wind roses were used:

1. Uniform wind distribution
2. Unidirectional wind distribution
3. Realistic wind distribution

To get representative wind conditions for an offshore wind farm, the long time measurements from a met mast in the North Sea at Borssele, 'Borssele point 1', were used. Accidentally the measurements at 20 m height instead of hub height were used from the met mast. This leads to a slightly different wind direction spread and a bit lower wind speeds. This was unfortunately discovered too late to rerun the experiments. However, the wind measurements were still a good representation for a location offshore. The met mast data were not only used to create the realistic wind distribution, but also the uniform and unidirectional wind distribution were created using this information.

For the uniform wind scenario, wind comes from all directions with the same probability and speed. The weibull parameters were chosen to be equal to the average of the Borssele point 1 met mast, and are $A = 9\,\mathrm{m\,s^{-1}}$ and $k = 2.1$. The second scenario only used wind from the main wind direction bin (225-255 degrees), the weibull parameters for this bin were kept equal to those from the met mast, but its probability of occurrence was 100 %. Wind from other directions outside the range simply did not occur. The weibull parameters are $A = 10.9\,\mathrm{m\,s^{-1}}$ and $k = 2.4$. The final scenario might be the most interesting, since this one is the closest to reality. Here the wind rose consisted of twelve parts with different occurrence and distribution. The average weibull parameters were equal to the uniform wind. The exact weibull parameters per direction can be found in appendix D. The respective wind roses can be found in the results section in figure 8b, figure 9b and figure 10b.

5.3.3 Results

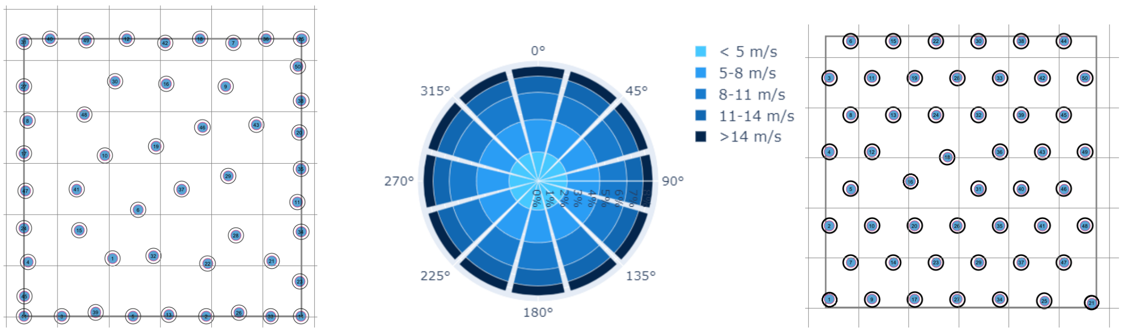
The results of over 20,000 iterations in the different wind climates are shown in figure 8, figure 9 and figure 10. This is double the 10,000 minimum iterations, because this experiment required less run time per iteration than expected. It was therefore easy to run it for more iterations than the minimum. These figures all contain the computed design by Openwind on the left, the wind rose in the middle and the manual design on the right.

5.3.3.1 Uniform wind distribution

The uniform wind scenario in figure 8 looks just like the earlier tested scenarios with different initial layouts, but has done double the amount of iterations. This has led to a negligible small increase in energy capture, and looks visually similar to the previous results in figure 6 and figure 7. Most turbines are located at the edge of the wind farm, minimising wind wake interaction. After filling the sides, the remaining turbines are more or less spread evenly around in the center, avoiding the crowded edge of the site. This spread is again to minimise the wake losses.

When manually designing this wind farm, instead of placing a lot of turbines at the edge, a symmetrical layout was chosen. The reason for this symmetry is that the symmetrical wind rose does not lead to any preferred wind direction, so all directions have the same impact on the energy capture. The symmetrical layout is based on rows of turbines in a hexagonal grid structure. This structure allows for compact placement, while still placing the turbines as far away from each other as possible. The found, best fitting, hexagonal design contains eight lines of alternating six or seven turbines. Because this design leads to a total of 52 turbines, the two turbines with the biggest losses are removed. These turbines are selected with the help of the energy capture analysis in Openwind. In order to maximise the yield, the opened spot in the regular pattern is filled by moving three neighbouring turbines slightly away from their grid position towards the opened spot. This design should then be quite a good design for 50 turbines at the given site.

The yearly yield of the manual design is 518.567 GWh and the computed design by Openwind yields 525.071 GWh. This means the algorithm created a design which perform a little over 1.2 % better in terms of energy yield. This difference can attributed to the fact that the algorithm exploits the boundaries of the wind farm site more than the manual design approach. The benefits of this would become smaller on larger wind farm sites. The manual design could be easily scaled up and deliver the same performance. The difference between the two approaches can therefore be assumed to be smaller on larger projects.



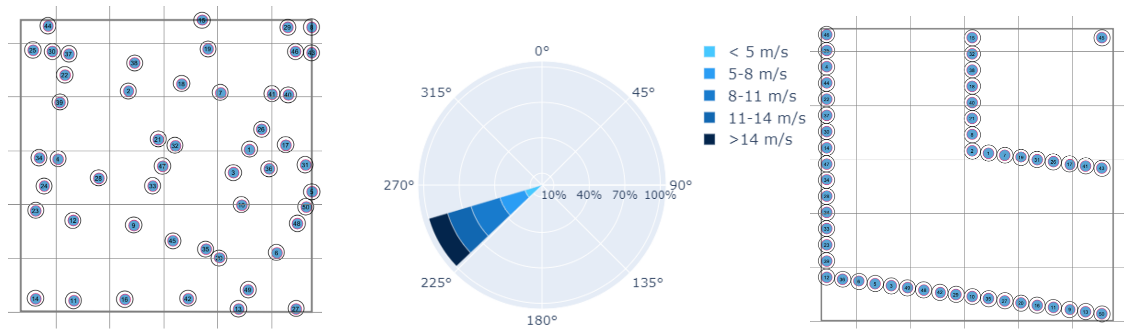
(a) *(b)* *(c)*

Figure 8: The results for the uniform wind scenario. **a:** Computationally optimized design. **b:** Wind rose. **c:** Manually optimized design.

5.3.3.2 Unidirectional wind distribution

Looking at the unidirectional wind results in figure 9, the final layout from the algorithm is hard to explain. It is expected that the turbines will be lined up in lines perpendicular to the wind direction to avoid energy losses due to wind wakes, but the turbines seem to cluster up and form even lines along the wind direction. The yield of this design is 721.374 GWh. Note that this is quite high due to the fact that the average wind speed of this experiment is much higher than previous experiments. It seems like the algorithm has not correctly interpreted the absence of all wind directions but one.

For the manual design in figure 9c, turbines were placed to minimise the individual turbine losses at first. This led to the left and the bottom side of the square site being stacked with turbines. Since these edges should have undisturbed wind from the southwest. Next the remaining turbines were placed as far to the right top corner as possible following the same pattern as on the left and bottom side. When the energy capture analysis showed the turbines on the bottom edge were still making wake losses, the bottom line was placed at an angle to avoid any wake forming with the vertical line of turbines on the left side of the wind farm. For the remaining turbines, two different set-ups were compared: One with a single turbine in the far corner and the remaining turbines following the shape of the edge and one with all remaining turbines in that same shape as the edge. This comparison resulted in the shown design, where the wake losses are very small and the yield comes to a total of 730.771 GWh per year. This means the energy yield of the manual design is almost 1.3 % better than the energy yield from the design from Openwind, which clearly has some problems in the optimization process for a single direction.



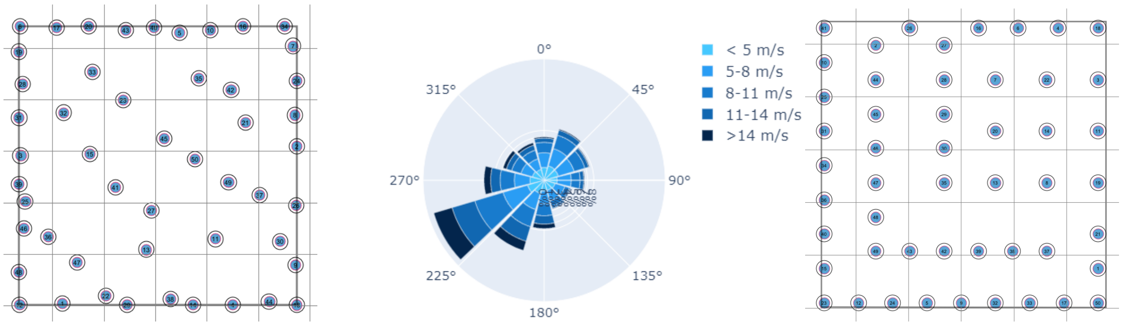
(a) *(b)* *(c)*

Figure 9: The results for the unidirectional wind scenario. **a:** Computationally optimized design. **b:** Wind rose. **c:** Manually optimized design.

5.3.3.3 Realistic wind distribution

The third scenario of this experiment aims to get the wind conditions close to reality. The results are displayed in figure 10. The results from Openwind initially look a lot like the uniform wind scenario. Again 32 turbines are on the border of the wind farm site, where the remaining 18 are spread around in the center. This time, however, the spread in the center is not just a random spread. The turbines are more or less lined up in four diagonal rows, perpendicular on the prevailing wind direction. The four lines of turbines avoid the wake losses when the strongest and most common wind direction blows through the farm, therefore optimizing their yield. This same lining up is slightly visible on the bottom edge, where two pairs of turbines are tilted slightly towards the southwestern wind direction. The final layout by the algorithm in figure 10a is a mix of turbines at the edge and turbines perpendicular to the prevailing wind direction. The total yield of the realistic wind farm is 526.568 GWh.

The manual design in figure 10c is focused on the most important wind direction first, the southwestern wind. Therefore the left and bottom edge are filled with turbines. Then a parallel inner row is placed, where southwestern wind could pass exactly between the outer edge turbines to reach this inner line. This ideal scenario will maximise the energy capture, but in practice there will be wake interactions also at this wind direction. Next to the two outer lines in the bottom left, some turbines are placed on the top and right edge. Turbines on the edge have little wake interactions and take therefore advantage from wind coming from the northeast side of the wind park. Also the distance to the other turbines is larger, lowering wake interaction. Then five of the remaining turbines are placed in one extra vertical line in the center, still trying to make most use of the southwestern wind. The last six turbines are spread in the last open space in the wind farm. The manual design is made with help of the energy capture evaluations in Openwind, which helped to identify the least efficient turbines in the configuration. The total yield of the manual design is 522.937 GWh. This makes the energy capture of the computed design almost 0.7 % higher than the energy capture of the manual design. This difference is a little smaller than with the uniform wind conditions. This could be caused by the better exploitation of the boundaries by the manual design approach in the unidirectional wind conditions.



(a) *(b)* *(c)*

Figure 10: The results for the realistic offshore wind scenario. **a:** Computationally optimized design. **b:** Wind rose. **c:** Manually optimized design.

5.3.4 Conclusion

Summing up the results, the Openwind designs are slightly better than the manual designs in two out of three cases. In the unidirectional case, the manual approach showed a better result, but the software seemed to have some interpretation problem with the extraordinary wind conditions. Therefore it can be concluded that the computational approach leads to slightly better results independent of the wind conditions. In this straightforward site with different wind conditions, using Openwind to create a design will increase the energy capture with almost 1 %.

5.4 Experiment 2: Wind farm design in different wind conditions with cables

This experiment will continue on the set-ups from experiment 1, where different wind conditions were analyzed. Here, however, the final layout will include electricity cables to connect the wind turbines with the grid.

5.4.1 Aim

The goal of this experiment is to analyze the impact of cable inclusion in the optimization procedure. In Openwind, it is possible to optimize a wind farm on Cost of Energy (CoE) instead of just on energy capture. The optimization for CoE includes costs for cables, substations and even roads (onshore). The algorithm aims to minimise the construction costs of the wind farm, while aiming for a high energy production. It is interesting to see the performance of the heuristics behind the cable optimization compared to the turbine position optimization. The question is if the results will be very different to the cableless designs. This will show if the quality of the solutions in both situations is similar, or that cable inclusion leads to totally different layouts.

This experiment consists of two comparisons. In the first comparison the layouts created by the cost of energy optimizer are compared with the layout from the former experiment. In the second comparison, the manual layouts will be compared to the CoE optimized solutions from Openwind.

5.4.2 Set-up

The first comparison is between the different optimizer options in Openwind. In Openwind it was possible to optimize the turbine positions and cable layout simultaneously using the Cost of Energy. This optimization has run for over 10,000 iterations to find the optimal balance between yield and cable length. Again this optimization was done for the three different wind scenarios as described in section 5.3. The resulting layouts were compared with the results from experiment 1. Secondly, the turbines in the manual designs will be connected by cables using Openwind. The turbines will remain in the original position and just the cables and a substation will be added. The final cabled layouts will be compared to the optimized cabled layouts created by Openwind for each of the three wind scenarios. This cable connection was not done manually since it was too complex to estimate the impact of the cable on turbine positions. Finding the optimal connection between the turbines by Openwind is therefore the best strategy for a (semi-)manual design.

When running the optimizer for CoE in Openwind, there were a lot of cost components that need to be determined. For this experiment, all basic settings in Openwind were used, since it is a lot of work to come up with exact costs for turbines, different cable types, substations, etc. The default costs in Openwind were assumed to be balanced and suitable for this experiment. The substation was located in the middle of the wind farm and the grid connection cable was drawn until the right edge of the farm towards land.

Openwind had access to four different cable types, which are shown in table 2. In this case, the maximum number of turbines in one line is ten, as the maximum line capacity is 30 MW. Cable crossing is allowed by default, but it is possible to change this setting. This used a different algorithm, which was not tested in this experiment.

Table 2: Available cables and characteristics in Openwind. These cables are available for the optimizer in all scenarios.

Name	Capacity [MW]	Cost per meter	Cable resistance [Ω/km]
1/0	10	50	0.3224
4/0	14	65	0.1608
500 kcmil	22	95	0.0689
1000 kcmil	30	122	0.0361

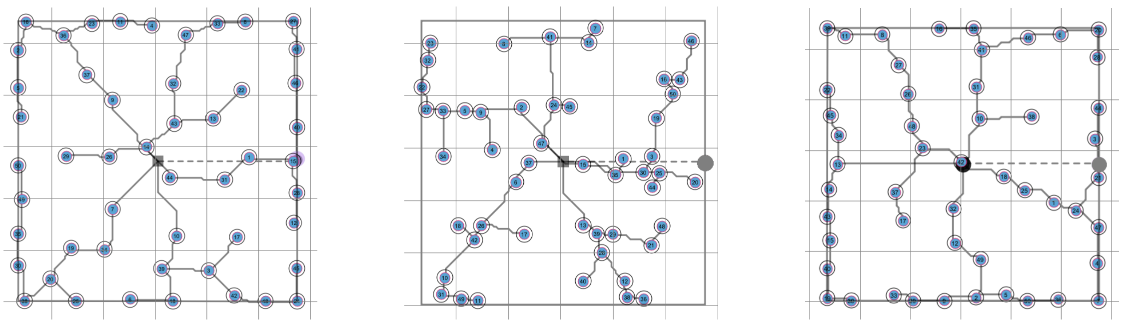
5.4.3 Results

The results for the comparison between the energy capture optimization and the cost of energy optimization are shown in table 3. Here the expectation is that the cabled designs will have a lower energy capture, since the algorithm is optimizing for costs instead of just the energy production. The differences, however, are very small. In the uniform wind and realistic wind scenario, see section 5.3, the losses in energy yield are just 0.2 % and 0.1 % respectively. This is a really small difference, which substantiates the choice to keep turbines at their positions when drawing cables in the manual design strategy. The unidirectional wind scenario performs better with cables in terms of energy capture, which is quite strange. It confirms the supposition that the Openwind software cannot handle wind from just one direction well. This result is therefore not used to draw any conclusions.

Table 3: Annual energy capture (GWh) of different designs with- and without cables. The columns show the different wind conditions, where the rows distinguish the set-ups with and without cables.

	'Uniform'	'Unidirectional'	'Realistic'
Without cables	525.071	721.374	526.568
With cables	524.131	721.859	525.878
Difference	-0.2 %	0.1 %	-0.1 %

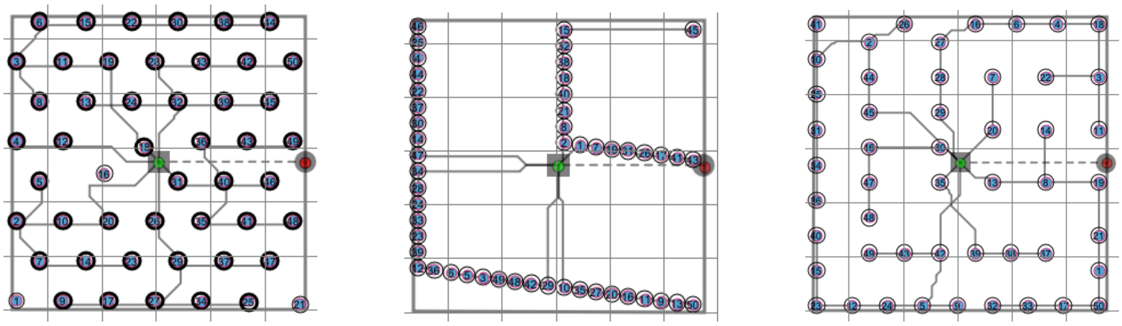
When visually comparing figure 11 with the final designs in figure 8a and figure 10a, the similarities are clear. When ignoring the failed unidirectional experiment, both the uniform wind and realistic offshore wind scenario show only some slight changes. Most turbines are still located at the edges, but some turbines are moved a little bit to the inside to guide the cables to the edge of the farm. This small movement leads to lower cable lengths, with only small production losses. This phenomenon is for example visible in the left top and left bottom of figure 11a. The turbines in the center area of both scenarios are also quite similar to the set-ups without cables. As can be seen in figure 11c, the centered turbines still form lines perpendicular to the prevailing wind direction to minimise wake interactions. It can be concluded from this first comparison that the impact of the cables is minimal and that there are great similarities between the design based on energy capture and the design based on cost of energy.



(a) *(b)* *(c)*

Figure 11: Three final computed designs including cables after more than 10,000 iterations for three different wind conditions. **a:** Final design for uniform wind. **b:** Final design for unidirectional wind. **c:** Final design for realistic offshore wind.

The second comparison is between the manual design with added cables in figure 12 and the cost of energy optimized layout in figure 11. In table 4, the costs of energy are listed. The differences between the designs are very similar to the comparison without cables. The Openwind algorithm outperforms the manual approach with 1 – 2 % higher energy yield in the uniform wind and in the realistic offshore wind scenario. The unidirectional scenario again shows inconsistent results compared to the expectation due to a flaw in the optimization. While the manual designs are not specifically optimized for cables due to its complexity, the designs still give an energy capture close to the results from the Openwind simulations. Nevertheless, it seems that integrated optimization has a slight benefit over the manual sequential process in terms of cable costs. The difference is small though, making it hard to conclude if this is caused by the less optimal manual layout or due to the sequential process.



(a) *(b)* *(c)*

Figure 12: Three final manual designs including cables. **a:** Final manual design for uniform wind. **b:**Final manual design for unidirectional wind. **c:** Final manual design for realistic offshore wind.

Table 4: Costs of Energy in [cost/MWh]. The three different wind scenarios are listed in the columns, the rows show the energy yield of the manual design, the energy yield of the design produced by Openwind and the difference between the two approaches. A positive difference means Openwind outperformed the manual approach.

	'Uniform'	'Unidirectional'	'Realistic'
Manual	26.243	18.438	25.924
Openwind	25.784	18.656	25.642
Difference	1.8 %	-1.2 %	1.1 %

5.4.4 Conclusion

In conclusion, adding the cables in the simulations shows similar results to the former experiment without cables. Openwind can create designs that have a $1 - 2$ % lower cost of energy than the manual designs. Optimizing with cables in Openwind leads to similar layouts as the optimization in section 5.3, so the cables have a minimal impact on the wind farm layout problem.

5.5 Experiment 3: Wind farm design with different wind turbine densities

This experiment compares designs with different number of turbines in the same site. For each number of turbines a layout with and without cables is optimized.

5.5.1 Aim

The aim for this experiment is to compare different turbine densities in the same wind farm site. The question is if a different number of turbines will lead to a totally different design or that the designs are scaled versions to fit more turbines. Next to that question, the impact of including cables with different turbine densities is analyzed. It is expected that including cables leads to pulling turbines from the edge of the wind farm closer to the center. This algorithm behaviour is predicted to be most visible in the design with the lowest number of turbines, since lower turbine densities will lead to smaller wake losses. Moving the turbines from the edge to the center will than result is smaller energy losses, while decreasing the cable costs a lot.

5.5.2 Set-up

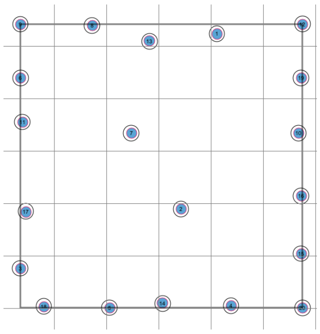
This experiments continued on the site in the North Sea as used in the previous experiments. The site was still a square of 5.4 km x 5.4 km. The used wind rose was identical to the realistic wind rose in the former experiments (see figure 10b), which represented a real case scenario of wind at the North Sea. With these conditions, the optimal layouts are calculated for different number of turbines. The used number of turbines were: 20, 30, 35, 40, 45 and 75. The resulting wind turbine layouts were compared to the original case with 50 turbines. The same cases will be optimized on cost of energy, where cable design is also included. The cases without cables and with cables will be compared on turbine placement and on energy yield. This study should show how the different optimization algorithms compare to each other and how this is affected by the turbine density.

Unlike the previous experiment, no manual designs had been made for different turbine densities. It would have been a lot of work to make a design for each situation and the designs would probably look a lot like each other. It was hard for a designer to find the ideal trade-off between optimizing for energy capture and optimizing the cable costs manually. It would also not add much to the experiment, as the main goal is to analyze the computational behaviour of different turbine densities.

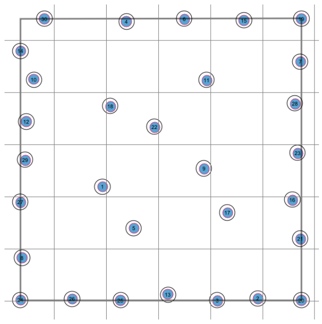
5.5.3 Results

The results of the simulations are shown in figure 13 and figure 14. When looking back at the 50 turbine design in figure 10a, the pattern perpendicular to the wind direction was recognised. Also the fact that the algorithm tends to place many turbine on the site edge to minimise wake interactions was discovered. Both these characteristics are visible in figure 13 as well. When starting with quite an empty site, only two turbines are placed in the center and the others are placed on the edge by the algorithm. This is just like the 50 turbine case, but since there is more distance between the turbines almost all turbines fit on the edge of the wind farm site. In the other wind farm designs with a growing number of wind turbines, the trend remains the same. The biggest share of turbines is fit at the edge and the remaining turbines form lines in the center part of the area. Although sometimes one or two turbines do not follow these lines perpendicular to the prevailing wind direction, the trend to form lines is visible from 30 turbines and more.

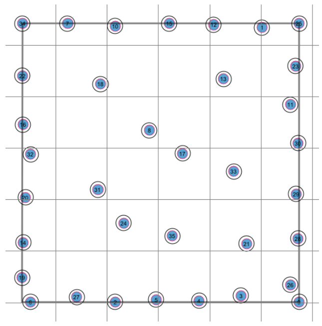
When reviewing the results of the optimization of cost of energy, the results are following the previously explained expectations. It was assumed that the turbines would be drawn closer to the substation to minimise cable cost while sacrificing some energy capture. Especially in the first few layouts in figure 14, the edges of the site are not covered anymore. The turbines have been moved to form similar lines as the center of figure 13. In figure 14d, figure 14e and figure 14f the edges are filled up again, because there are more turbines available to be placed at the edge and the algorithm forces the turbines away from each other to minimise the wake loss. With more turbines, the effect from the wake losses becomes stronger than the pulling to the center caused by the cable length minimisation. When scaling up to 75 turbines in the wind farm site, the pattern is very similar to the uncabled layout.



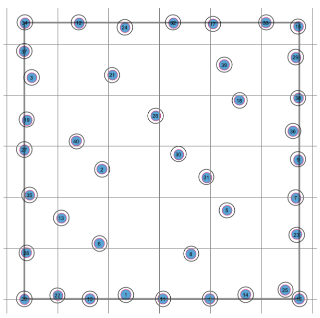
(a) 20 turbines



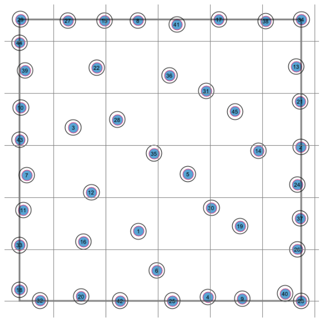
(b) 30 turbines



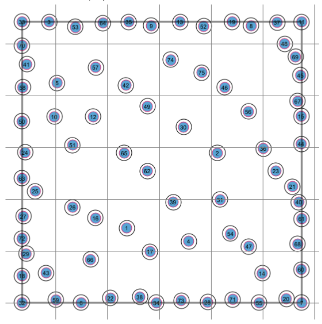
(c) 35 turbines



(d) 40 turbines

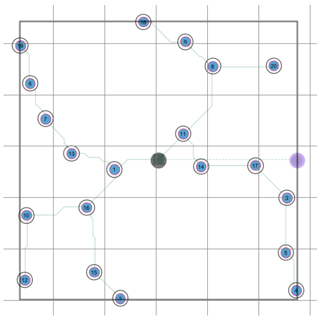


(e) 45 turbines

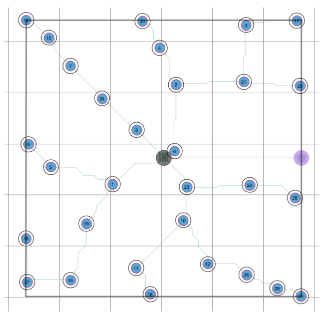


(f) 75 turbines

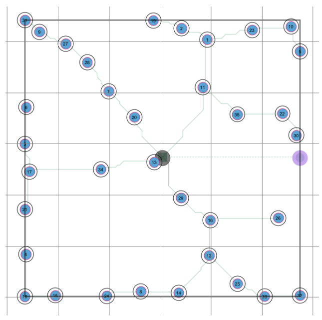
Figure 13: Final wind farm designs. Optimized for energy yield with different number of turbines and a realistic wind distribution. The total number of turbines in the design is listed below each image.



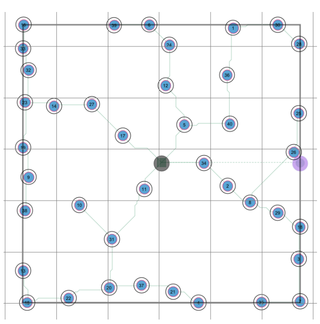
(a) 20 turbines



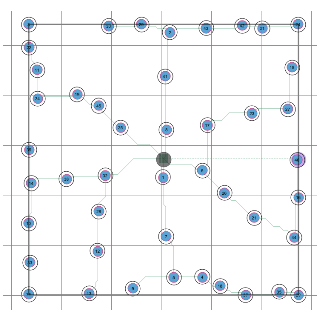
(b) 30 turbines



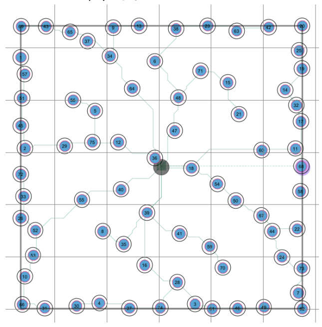
(c) 35 turbines



(d) 40 turbines



(e) 45 turbines



(f) 75 turbines

Figure 14: Final wind farm designs with cable connections. Optimized for cost of energy with different number of turbines and a realistic wind distribution. The total number of turbines in the design is listed below each image.

5.5.4 Conclusion

This experiment confirms the hypothesis that the difference between the two optimization strategies is largest at lower number of wind turbines. When the wind farms are scaled up to 75 turbines, the similarities are very clear. Still at some points on the edge of the site in figure 14f, the turbines are drawn towards the center to minimise the cable length, but this effect is much smaller with high turbine density than with low turbine density. No comparison with manual design has been made, as it would be very hard to find the trade-off between cable costs and wake effects.

5.6 Experiment 4: Wind farm design for more complex sites

This experiment challenges the Openwind software with more difficult wind farm sites. Various scenarios with increased site complexity were optimized and the results were analyzed.

5.6.1 Aim

The goal of this experiment is to explore the limits and possibilities of Openwind. How realistic and complex can scenarios be, while still remaining suitable for optimization with Openwind. The final designs for some cases are analyzed and the performance of the software package is evaluated. These scenarios would take more time for a manual design strategy, as the situations are more complex. It is interesting to see how the computational design strategy handles these cases. There will not be any manual designs in this experiment as it would be too challenging and too time consuming to create a proper design.

5.6.2 Set-up

In this experiment three different wind farm sites were optimized with and without cables. In realistic wind sites, the site is almost never a perfect square as seen in the previous experiments. Some parts of a site can be inaccessible, or a site can be split in multiple parts. Therefore it was important that a wind farm optimization application is able to process these more complex scenarios as well. To see how Openwind takes these tougher conditions, the following three scenarios were executed. A square area separated in the middle into two parts (figure 15a and figure 15d), a square area split into four equal squares (figure 15b and figure 15e) and a very complex area (figure 15c and figure 15f). The latter lost the regular shape.

In the first two situations, the algorithm was still allowed to let the cables cross the unavailable area. The grey areas were only restricted for turbine placement. The substation was still placed in the center of the entire area, just like the previous experiments. It is especially interesting to see if the turbines will be distributed equally over all parts. Theoretically equally spread turbines should lead to a lower wake loss, although the wind characteristics can change this balance a bit. The optimization runs were started with 50 turbines spread randomly in and around the areas, in order to not give the optimizer any bias at the start.

The last situation shows a complex site in a stranger shape. This site was a bit spiky at some edges to see if the optimizer utilises the far-most parts of the area well. The area was also crossed by a pipeline, which should be avoided with some distance at both sides. Also multiple grey circles were placed in the area, these represent shipwrecks and reefs. It was not allowed to place any turbines at these points as well. Here again it is interesting to see if the optimizer can handle these different restrictions and if the turbines get distributed in a proper way.

5.6.3 Results

Looking at the spread across the different parts of the wind farm site in figure 15, it is not completely as expected. The four area site in figure 15b and figure 15e distributes the 50 turbines evenly across the different squares placing 12 or 13 in each of them. The two area site, however, does not split the turbines in groups of 25, but has a 24 to 26 division. This could be due to better wind conditions in one side of the wind farm, but then figure 15a and figure 15d should have the same distribution. The difference between the two similar scenarios is that figure 15a has 26 on the left and figure 15d has 26 on the right. It is hard to determine the cause of this with only one experiment. It could mean that the algorithm has problems moving a turbine from one region to another and that it simply started with optimizing with this spread over the two areas. Another possible reason is that the wake losses are relative small and the optimal solutions are close to each other, making the optimizer getting stuck in a local optimum. Analyzing intermediate results of the optimization could help determine the cause, but these are unfortunately not available.

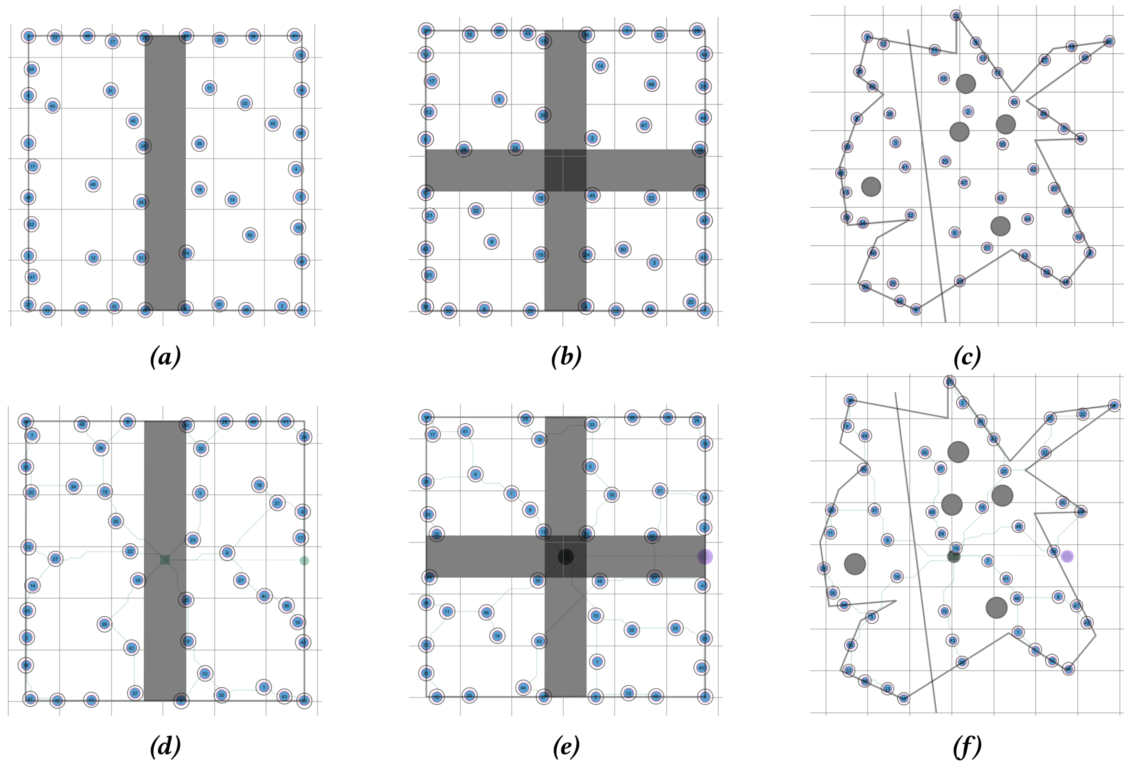


Figure 15: Final wind farm designs with and without cable connections. Different site boundaries and restrictions. **a:** 2 separate areas. **b:** 4 separate areas. **c:** A more complex shape with a pipeline crossing the site (line). Also some shipwrecks and reefs block the site at several points (circles). **d:** 2 separate areas with cable connection. **e:** 4 separate areas with cable connection. **f:** A more complex shape including cable connection with a pipeline crossing the site (line). Also some shipwrecks and reefs block the site at several points (circles).

Figure 15c and figure 15f show the complex wind farm site. Here the optimizer easily found a fitting solution and the more complex restrictions did not seem to effect the run time per iteration too much. The optimizer also has no trouble finding all the spikes on the edges of the wind farm area and seems to use all available area quite effectively. Interesting to see is that this scenario still leads to a design with a lot of turbines at the edge. That is something characteristic for the algorithms in Openwind. The comparison between the cabled and uncabled layout shows the same as in previous experiments. Including cables in the design tends to line up turbines to form shorter paths for cabling. The rest of the layout is similar in both designs.

5.6.4 Conclusion

The optimizer in Openwind is challenged with different scenarios consisting of multiple separate areas or containing various obstacles. In the case of two split areas, the optimizer seemed to have trouble with equally dividing the turbines over both parts. This could mean that moving from area to area is limited in the optimization process, possibly leading to a worse final solution. The complex case was solved by the optimizer in a good fashion. It could be interesting to investigate this more complex scenario in more detail by using different starting situations and multiple runs to see if it produces similar results all the time. The complex design is not created with a manual design approach, since that would be too complicated and time consuming. Therefore no comparison with the manual approach can be made for this experiment.

6 Discussion and reflection

This thesis aims to show the strengths and weaknesses of different wind farm design approaches. To find the strengths and weaknesses it was required to get a clear view on the current state of wind farm design software. To get a good picture on how wind farms are designed currently, some designers have been interviewed. These interviews form the basis to investigate the performance of the Openwind software package. This chapter first discusses all results and findings and then reflects back on the followed procedures.

6.1 Discussion

6.1.1 The wind farm design process

Wind farms are currently designed with help of software, but some things have to be designed by the hand of the designers as well. As followed from the interviews, optimization software is far from perfect and many constraints can not be inserted into the optimization. To meet all requirements, designers have to manually adjust optimized designs, or even make the entire design by hand. However, the software also supports a designer by making clear visualizations and by measuring performance of a created design.

The sketched design process in chapter 4 is based on two interviews with designers. This should be a good image of the current situation, but it could be more substantiated if more designers were interviewed. The problem was that not many different wind farm designers were found to answer all questions. If more designers were found to be interviewed, this could give extra details for the design process. They could also be asked to judge the manual designs in the experiments, which has not been done in this thesis.

Where currently designing a wind farm is not just a push of a button, optimization software keeps developing. Many features are included in current programs and therefore some experiments have been done in Openwind, to examine the possibilities and see the performance compared to manual designing.

6.1.2 Experimental set-up

All experiments were conducted in Openwind. It was decided to let all experiments run for a minimum of 10,000 iterations, after which no large improvement was seen anymore. Longer experiments would give slightly better layouts, but since the difference is so small this would not lead to any different conclusions. The relative short computation times of the experiments allowed for many different set-ups to be tested.

The manual design were actually semi-manual, as Openwind was used to determine the worst performing turbines in different layouts. The steps taken in the manual design process were written down clearly, so the rationale for certain placement is clear. It is quite possible that these designs are different than real constructed wind farms as they were not designed by an experienced designer. It could be valuable to let an experienced designer judge the designs to see if some things were overlooked. The latter experiments were only conducted in Openwind as the lack of experience made the scenarios too difficult to create a high quality design. The comparison with the software would then be unfair, as the designs could be easily improved.

6.1.3 Manual versus computed design

In Experiment 1 in section 5.3, the computed designs get a larger energy yield than the manual designs. The energy yield was up to 1 % higher when using Openwind to optimize the layouts. This is as expected, since optimization software should be able to test many different layouts in the same time a designer can only make a few attempts.

The site in the first experiment is very simple as it has no more restrictions than the borders of the site and the number of 3 MW turbines. This simplicity allowed for good manual designs, even without any design experience. Next to the difference in energy output, visually the manual design and the computed design also differ a lot. This visual difference shows that it can be quite easy to find a reasonable design, but that it is difficult to improve the manual design as there are many different reasonable layouts to be found.

The optimization algorithm in Openwind uses the boundaries of the wind farm site very well to maximise the energy output. All positions on the border of the wind farm have fewer neighbouring turbines, and therefore reduced wake losses. Maximising the number of turbines on the edge, also leads to larger average turbine spacing, contributing to the reduction of wake losses. The tendency to put turbines on the edge was visible in all Openwind layouts.

The manual designs on the other end, have less optimally used the boundaries of the wind farm site. The goal of the designs was of course to optimize the energy output, but the manual design follows a regular pattern. This leads to some losses compared to the computed design on the edge, but it allows for easier scaling to larger or smaller wind farms. The manual designs were created before the results of the computed designs were known. Possibly with the advantage of using the edges in mind, a designer could find better manual solutions. More experienced designers could maybe also find a layout with a higher energy yield, but this has not been confirmed.

Considering all these things, it is hard to state clearly that the computed approach is better than the manual approach for a simple wind farm site with realistic wind conditions. The uncertainty in the manual designs is large, while the differences in energy yield were just below 1 %. However, the software could probably also find slightly better solutions in repeated runs, or during some extended runtime. It can therefore be concluded that it is very likely that the computed design approach outperforms the manual design approach in a simple scenario.

6.1.4 Cable topology

When including more aspects of the wind farm into the optimization, the optimization will become slower. On the other hand, a designer will also require more time to create a design when there are more features required. One of these features is the cable topology. Even if the wind turbines are placed optimally, an inefficient cable layout can still cause a higher cost of energy. Therefore the second experiment, section 5.4, analyzed the impact of the cable layout on the design.

With the inclusion of cables, the designs were optimized in cost of energy instead of energy yield. This allowed for optimizing the turbine positions and the cable topology at the same time. The computed designs had an average of 1.4 % lower cost of energy than the manual designs. This difference is larger than at the previous experiment. It can be explained by the fact that the turbine positions in the manual design were not altered compared to the previous experiment. The Openwind algorithm moved the turbines around to find the correct balance between the wake losses and the cable length. It was too difficult to create a manual design taking into account the turbine positions and cable topology simultaneously.

Since creating cable layouts can be difficult and time consuming, Openwind is used to quickly create the cable topology for the manual designs. The advantage of using Openwind for the cable topology is that it can really show the advantage of integrated optimization versus sequential optimization. Integrated optimization leads to better results than sequential optimization. Where in the previous experiment the difference in energy yield between the manual and computed approach was just below 1 %, the difference in energy yield is now 1.4 %, meaning the integrated optimization increased the energy yield difference with almost 0.5 %.

6.1.5 Turbine Density

To evaluate the difference in layouts between the wind farms with and without cables, a series of experiments with different turbine densities is executed and described in section 5.5. In the same simple site as the experiments before, the algorithms behaved like expected. The fewer turbines in the wind farm, the larger the difference between the cableless and cabled designs. This can be subscribed to the smaller impact the turbine position has with a lower turbine density. With just a few turbine in a large space, wake losses are small and moving a turbine has a small impact. Therefore the cable costs dominate the cost of energy and turbines are moved around more compared to layouts with a higher turbine density. In the same way, wind farms with higher turbine density have less flexibility in moving the turbines. There are many interactions with other turbines, leaving fewer possibilities for the cable optimization without having a big impact on the energy yield.

It is also seen that it is efficient to place wind turbines along the cable path from the substation in the center to the most outer turbines. Where turbines on the edge of the wind farm site have the highest energy yield, the cable costs become too high if no turbines are placed in center of the farm along the cable path. Wind farms with a higher density required less changes in turbine positions, as there were already many turbines close to the substation that could be placed next to the cables.

The interaction between the turbine position optimization and the cable topology optimization is too complex for a manual designer. Therefore the manual design would be done sequentially. Since the cable costs have a larger impact on lower turbine densities, it is expected that manual design would be worse for lower wind farm densities. However, no manual designs have been made for all different wind turbine densities to support these expectations as this would be very time consuming.

6.1.6 Complex wind farm sites

Two situations were optimized in the experiments in section 5.6. In this experiment there were two different complex wind farm sites: A wind farm area with separate regions and a strangely shaped area with some unavailable spaces inside. The first was used to see how Openwind could distribute wind turbines between two areas. It can be difficult for some algorithms to let turbines move from one part of the site to the other. It was expected that in the set conditions the turbines would be evenly spread across both halves. The results showed that one turbine too many was placed at one part and one too few at the other.

The unequal placement across the different parts can be interpreted as a flaw in the optimization algorithm. It appears to struggle to move a turbine from one half to the other. However, it is premature to conclude this from just one single experiment. To further investigate this a series of optimizations could be done where the starting positions of the wind turbines are set. For example all 50 turbines could start at one side of the farm to see how they spread or the turbines start with an equal division over the parts. Further experiments like this should prove if the algorithm of Openwind indeed struggles to move turbines across a barrier. It should be noted that the intermediate results of such experiments can be important to evaluate the progress. It is not used in the project, but Openwind has the capability to save intermediate results, which can support this type of experiments.

6.1.7 Improvement of the wind farm design process

Currently wind farm designers are able to design good wind farms, using various optimization and analysis tools to support them. There are two things which designers could help improve their designs and the design process: Faster optimization algorithms and more complete tools.

First it goes without saying that faster optimization tools would improve the design process. The quicker a design can be made, the lower the costs will be for making the design. In this thesis the optimization runs were quite short compared to what designers currently find acceptable. Of course the experiments in Openwind were quite basic and they did not run until the (local) optimum, but the quality of the results in short time show that short calculation times might be possible.

Secondly, according to designers the optimization tools lack some options. This requires a designer to edit the outcome of the optimization software to meet all constraints. Ideally, the final outcome of the software can be directly used for construction. This goal is far away and might never be reached, but there are some possible improvements possible in the software to include more constraints. For example visual constraints or some limits to cable routes are not yet a possibility. It has to be mentioned that these options are to some limit available in other programs than Openwind. However, none of the programs is close to having a complete package to design a wind farm with a push of the button and this might also never be reached due to the large number of different possible constraints and demands for a design.

6.2 Reflection on the procedure

6.2.1 Openwind

All experiments in this thesis have been conducted in Openwind. It must therefore not be forgotten that all results are also related to the performance of Openwind. While Openwind is a good software package for wind farm designing, the choice was not merely based on performance. Three software packages were reviewed: WindFarmer, Openwind and windPRO. As the access to windPRO was not available for the entire span of the project, this software package could not be used. However, both interviewed designers mentioned this program as their mainly used optimization program. It should thus be kept in mind that the performance of windPRO is possibly better than Openwind, although Openwind is a good representation of the capability of wind farm optimization software.

6.2.2 Cable topology

It is likely that the difference of 0.5 %, as mentioned in section 6.1.4, is caused by the integrated optimization compared to the sequential optimization. However, it is also possible that Openwind uses different algorithms for creating a cable layout for a design with fixed turbines than it uses for optimization of turbine positions and cable topology simultaneously. There is not much known about the specifics of the algorithms in Openwind. To verify if the algorithms are identical or not, another experiment is required. This experiment should make a sequential design with just Openwind. This sequential design should be compared with the manual design, to see the difference. If the difference stays around 1 %, the algorithms are probably the same and the effect can be fully subscribed to the integrated optimization.

6.2.3 Design experience

The results of the manual design are influenced by the experience of the designer. Where a reasonable manual design can be made by a lot of people, an experienced designer will be able to get a good quality design. They will be better at understanding the consequences of a change in turbine position to the cable layout for example.

All designs in this study have been made by an inexperienced designer. As it is hard to find a step-by-step guide for designing a wind farm, choices have been made based on literature review and some layouts of real constructed wind farms. The design procedure has been described at each manual design as well, so one can follow the reasoning behind it. While the manual designs can be considered of good quality, it has to be kept in mind that an experienced designer might get better results than used in this report.

For fair comparison, two ideas are proposed. First of all, an experienced designer could be asked to review the current manual designs. The impact of possible modifications can then be analyzed, leading to a better comparison of manual design versus computed design. A second thought is to recreate an existing wind farm in Openwind and compare the results. This requires the knowledge on the design procedure of the existing wind farm, as well as all set constraints at the start of that project. If this information can be gathered, this would be a great experiment to see the performance of just software against a (manual) designed and constructed wind farm.

7 Conclusion

In this study the wind farm design process has been described based on literature and the information from two experts in the field. The designing of a large wind farm is a complicated process, where software is used to support the process. However, the current wind farm design software cannot design a wind farm in one click. The wind farm designer always has to make adjustments to fulfill all requirements and assure no constraints are breached.

The resulting energy yield of the computed designs is about 1 % higher than the manual designs. This difference is too small to conclude that software always performs better than a manual approach. The improvement was achieved with short optimization runs of several hours. It was shown by the performance of one wind farm design tool only, so there might be room for more improvement using other tools or algorithms. Also the manual designs were created by an inexperienced designer. This combined gives a large uncertainty, but it is likely that software is required to create better designs for a wind farm.

One main characteristic of the optimization algorithm that led to the higher quality of the computed designs, is the use of the boundary of the wind farm site. Where manual design are prone to use structured turbine patterns, the optimization software places many turbines on the edge of the wind farm site to avoid wind wake interactions. Also the integrated design of the cable topology and the turbine positioning gives the algorithm an advantage over the sequential optimization of the manual approach. This effect was visible with a turbine density equal to an existing wind farm. The effect gets stronger at lower densities.

When increasing the complexity of the wind farm site, it gets almost impossible for a manual designer to find a good solution. In the experiments, the software was able to still find solutions easily. Layouts were designed for a wind farm site consisting of separate areas and a wind farm site with some restricted areas within. Nevertheless, the final layout of the separate site showed that the turbines were not divided evenly over the different parts. This means that the optimization not necessarily struggles to find a feasible solution, but that the solution probably still can be improved.

It is unlikely that there will be an optimization tool that will fully replace the designers' job. The designing of a wind farm will probably remain an interaction between a designer and design software. Software is already playing a big role in the design process, but when the software gets improved and more options are included, it can become even more important. Due to the complexity of wind farm projects, the amount of involved parties and the number of possible constraints the wind farm designer will always have a crucial role in the design process.

References

[1] Dutchnews.nl, 'Thousands march for action on climate in The Hague', 27/09/2019, Available: https://www.dutchnews.nl/news/2019/09/thousands-march-for-action-on-climate-march-in-the-hague/. (visited on 20/02/2020).

[2] NOS.nl, 'Vleesvervangers bezig met snelle opmars, verkoop vlees daalt', 14/08/2019, Available: https://nos.nl/artikel/2297492-vleesvervangers-bezig-met-snelle-opmars-verkoop-vlees-daalt.html. (visited on 20/02/2020).

[3] United Nations Climate Change, 'The paris agreement', 2015.

[4] K. Zipp, 'Wind energy performance map for Europe', 02/02/2012, Available: https://www.windpowerengineering.com/wind-energy-performance-map-for-europe/. (visited on 20/02/2020).

[5] Centraal Bureau voor de Statistiek, 'Energy consumption by energy carrier', 2018, Available: https://longreads.cbs.nl/trends18-eng/economy/figures/energy/. (visited on 20/02/2020).

[6] Richard, C., 'UK offshore wind could beat onshore on cost', 28/01/2019, Available: https://www.windpowermonthly.com/article/1524017/uk-offshore-wind-could-beat-onshore-cost. (visited on 30/10/2019).

[7] Mosetti, G., Poloni, C. and Diviacco, B., 'Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm', *J. Wind. Eng. Ind. Aerod.*, vol. 51, no. 1, pp. 105–116, 1994.

[8] Grady, S.A., Hussaini, M.Y. and Abdullah, M.M., 'Placement of wind turbines using genetic algorithms', *Renew. Energy*, vol. 30, no. 2, pp. 259–270, 2005.

[9] Stanley, A.P., Thomas, J., Ning, A., Annoni, J., Dykes, K. and Fleming, P.A., 'Gradient-Based Optimization of Wind Farms with Different Turbine Heights', SciTech, National Renewable Energy Laboratory, Colorado, 2017.

[10] EMD international A/S, 'Windpro', Available: https://www.emd.dk/windpro/. (visited on 04/03/2020).

[11] DNV GL, 'Windfarmer analyst', Available: https://www.dnvgl.com/services/wind-resource-assessment-software-windfarmer-analyst-3766. (visited on 04/03/2020).

[12] UL, 'Openwind', Available: https://aws-dewi.ul.com/software/openwind/. (visited on 04/03/2020).

[13] Commelas F. and Ozon J., 'Graph Coloring Algorithms for Assignment Problems in Radio Networks', *Applications of Neural Networks to Telecommunications 2*, pp. 49–56, 1995.

[14] González, J.S. Gonzalez Rodriguez, A.G., Mora, J.C., Santos, J.R., and Payan, M.B., 'Optimization of wind farm turbines layout using an evolutive algorithm', *Renew. Energy*, vol. 35, no. 8, pp. 1671–1681, 2010.

[15] Kennedy, J. and Eberhart, R., 'Particle Swarm Optimization', Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, Australia, 1995.

[16] Hu, X., 'Particle Swarm Optimization: Tutorial', Available: http://www.swarmintelligence.org/tutorials.php. (visited on 31/10/2019).

[17] Rahmani, R., Khairuddin, A., Cherati, S.M. and Pesaran, H.A.M., 'A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (PSO)', 2010 Conference proceedings IPEC, Singapore, Singapore, 2010.

[18] Abdmouleh, Z., Gastli, A., Ben-Brahim, L., Haouari, M. and Ahmet Al-Emadi, N., 'Review of optimization techniques applied for the integration of distributed generation from renewable energy sources', *Renew. Energy*, vol. 113, pp. 266–280, 05/2017.

[19] Cauchy, A.M., 'Méthode générale pour la résolution des systemes d'équations simultanées', *Comptes Rendus Hebd. Séances Acad. Sci.*, vol. 25, pp. 536–538, 1847.

[20] Garcia, S.I., 'An introduction to Gradient Descent Algorithm', 03/06/2018, Available: https://medium.com/@montjoile/an-introduction-to-gradient-descent-algorithm-34cf3cee752b. (visited on 31/10/2019).

[21] Stanley, A.P.J. and Ning, A., 'Coupled wind turbine design and layout optimization with nonhomogeneous wind turbines', *Wind Energ. Sci*, vol. 4, pp. 99–114, 2019.

[22] Quaeghebeur, E., 'Robust wind farm layout optimization using pseudo-gradients', Abstract from The 11th International Symposium on Imprecise Probability, Ghent, Belgium, 2019.

[23] Esau, L.R. and Williams, K.C., 'On teleprocessing system design, Part II: A method for approximating the optimal network', *IBM J. Res. Dev.*, vol. 5, no. 3, pp. 142–147, 1966.

[24] Bauer, J. and Lysgaard, J., 'The offshore wind farm array cable layout problem: a planar open vehicle routing problem', *Journal Oper. Res. Soc.*, vol. 66, pp. 360–368, 2015.

[25] Katsouris, G., 'Infield cable topology optimization of offshore wind farms', Master of Science Thesis, Delft University of Technology, Delft, 2015.

[26] Technical University of Denmark, 'Wasp', Available: https://www.wasp.dk/WAsP. (visited on 21/04/2020).

[27] Gray, J.S., Hwang, J.T., Martins, J.R.R.A., Moore, K.T. and Naylor, B.A., 'OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization', *Struct. and Multidisciplinary Optim.*, vol. 59, no. 4, pp. 1075–1104, 2019.

[28] DTU Wind Energy, 'TOPFARM 2.1.0 documentation', Available: https://topfarm.pages.windenergy.dtu.dk/TopFarm2/. (visited on 31/10/2019).

[29] Jensen, N O, 'A note on wind generator interaction', *Roskilde: Riso National Laboratory*, vol. 2411, 1983.

[30] The pip developers, 'Package installer python (pip)', Available: https://pypi.org/project/pip/. (visited on 31/10/2019).

[31] Ministerie van Infrastructuur en Milieu, 'Structuurvisie windenergie op land', 2013, Available: https://www.rijksoverheid.nl/documenten/rapporten/2014/03/31/bijlage-1-structuurvisie-windenergie-op-land. (visited on 21/04/2020).

[32] Ministry of Economic Affairs and Climate Policy, 'Offshorewind website rvo', Available: https://offshorewind.rvo.nl/. (visited on 31/10/2019).

[33] Rijksoverheid, 'Vattenfall bouwt tweede windpark op zee zonder subsidie', 10/07/2019, Available: https://www.rijksoverheid.nl/actueel/nieuws/2019/07/10/vattenfall-bouwt-tweede-windpark-op-zee-zonder-subsidie. (visited on 06/04/2020).

[34] R. Bos, 'Mail contact: Re: Interactivity in wind farm design', 17/09/2019.

[35] Koepel Windenergie Noordoostpolder, 'Windpark Noordoostpolder', Available: https://www.windparknoordoostpolder.nl/en/. (visited on 31/10/2019).

[36] B. Ummels, 'Mail contact: Re: Windpark ontwerp', 02/10/2019.

[37] Google, ‘Google Earth’, Available: https://www.google.com/earth/. (visited on 07/11/2019).

[38] EMD international A/S, ‘windPRO online manual’, Available: http://help.emd.dk/knowledgebase/. (visited on 14/11/2019).

Appendices

A windPRO by EMD International

This appendix contains an in-depth analysis of the optimization program windPRO. In the end OpenWind was used, because windPRO was unavailable. Therefore this analysis is moved to the appendix.

windPRO is a software package created and sold by EMD International. EMD international is a independent software and consulting company based in Denmark. As written on their website, windPRO is the industry leading software suite for design and planning wind farms projects. It covers everything from wind data analysis, calculation of energy yields, quantification of uncertainties, assessment of site suitability, to calculation and visualization of environmental impact. windPRO can also be used for detailed post-construction analysis of production data. After hearing from different people about windPRO, the program was analyzed, as it should give a clear picture about the current possibilities. I am thankful that EMD could provide me with a full access trial, in which all possibilities could be explored. I am running windPRO version 3.3.

When windPRO opens, there is an overview on a land map of all existing projects. This is an easy overview of all your available ward farm designs. To assess all possibilities in windPRO, two different projects will be tested. The first project will be an included sample project onshore in Germany, which will show a variety of options in windPRO. The second project will be to design and optimize an offshore wind farm, with a fixed number of wind turbines generators (WTG) on a selected site.

Immenhausen, Germany

The first project is located in central Germany near Kassel, in Immenhausen particularly. The set-up is shown in figure 16. The center of the wind farm is located at N 51.415° E 9.500°. The project contains 5 existing WTG (blue) and is extended with 7 new WTG (red). The site is indicated by the yellow surface, these constraints had to be followed by the designer, there are no restrictions within this area. The existing wind farm is built on top of a hill as can be seen in the elevation map, which is available as well, see figure 17. The construction of wind farms on a hill is preferred, since the wind resource will interrupted by less objects, leading to higher production values. The dashed marked objects around the site in the elevation map indicate obstacles around the wind farm. Another visualization method offered in windPRO is the exportation to google Earth [37]. As can be seen in figure 18, this gives a nice overview of the visual impact of the wind farm and helps the designer visualize the impact of obstacles around the site. Neighbouring buildings can be marked on the map to apply noise or shadow restrictions for those locations.

After the site has been defined and the new turbines are in position, a wind resource in required. This wind resource can be calculated for an area using the build-in RESOURCE module. In this case, the module generated a wind resource map with information on 4 different heights (50m, 100m, 130m, 170m) for the entire area of the site, with a spacing of 25m in both x- and y-direction. The covered area is about 3.5km x 3.5km, which took 154 minutes to be calculated. The wind conditions used for this project do not reflect reality, but the design procedure follows the same steps as for a real wind resource.

Now the new WTG are put in place and the wind resource is available, wind farm production calculations are possible. The module PARK is used to calculate the AEP of the wind farm and compare the placement of each different turbine. In the main result output file form the PARK module the

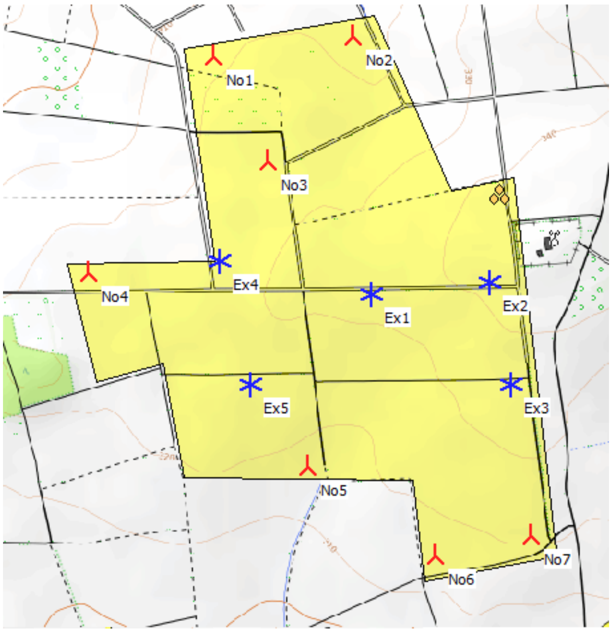


Figure 16: Wind farm design at Immenhausen, Germany. The blue symbols represent the existing turbines, the red markers are the new placed turbines.

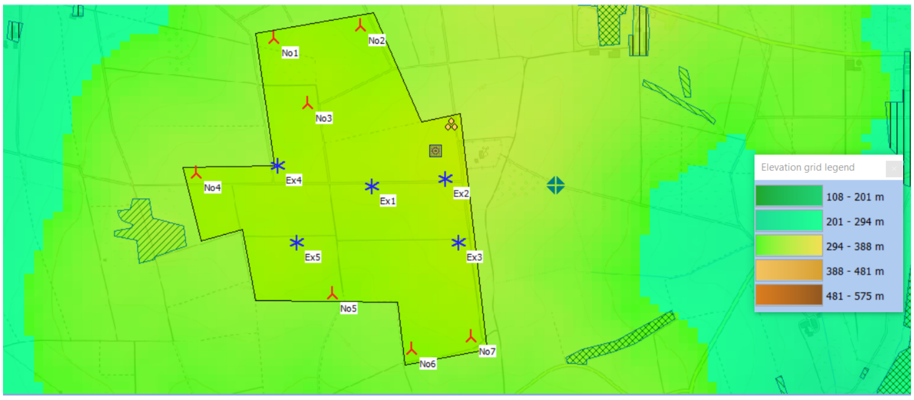


Figure 17: Height map of the wind farm at Immenhausen, Germany. The wind farm project is located on top of a hill.

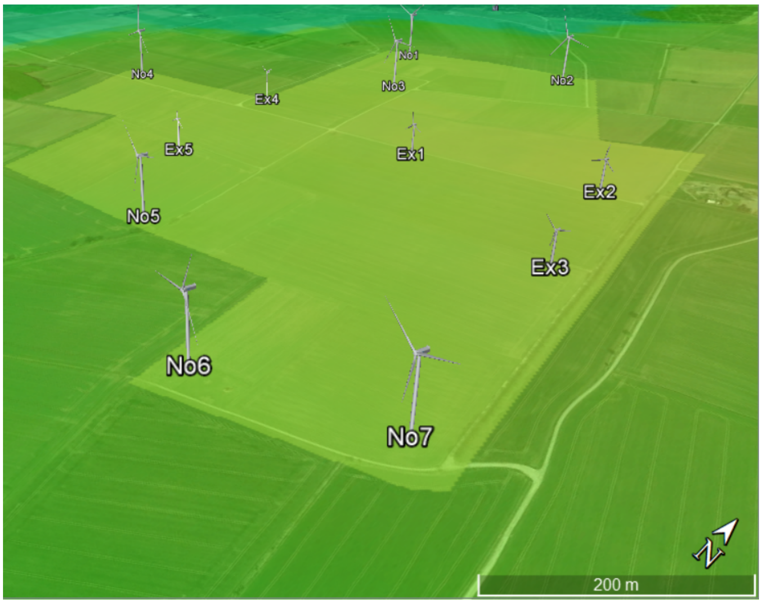


Figure 18: Google earth view of the wind farm at Immenhausen, Germany. The five existing turbines surrounded by the 7 newly added models.

Table 5: Wind farm production results using PARK for the full wind farm project.

WTG combination	Result PARK [MWh/y]	Free WTGs [MWh/y]	Wake loss [%]
Wind farm	42,208.30	47,834.10	11.8
New WTGs only	39,471.90	44,518.90	11.3
Existing park WTGs only	2,736.40	3,315.20	17.5
Existing without new WTG	3,195.20	3,315.20	3.6

energy production of all 7 new WTG is shown, as well as the production of the 5 existing WTG, see table 6. Next to the individual results, the total design is compared with the old set-up as shown in table 5. Three different turbine types are used: TACKE TW 600e-600/200, VESTAS V42-600 and SENVION 3.2M114VG-3.200. In the table only the manufacturer is mentioned, refering to these turbine types.

Next to the main result, the PARK module outputs many other files:

- Reference WTGs
- Production analysis
- Power curve analysis
- Wind data analysis
- Park power curve
- WTG distances
- Map

The reference WTGs can be used to review the impact on the original wind farm, the production analysis shows wind direction dependant wake losses. For 12 different wind directions, the production and the wake losses are shown. The power curve analysis shows the power curve of the selected new WTG type. The wind data analysis lists the Weibull parameters of the wind resource, in this case in 12 different wind directions. The park power curve is similar to the normal power curve but averaged over the entire wind farm. This includes wake effects, but does use only one wind resource input.

Table 6: Production results per wind turbine using PARK.

WTG	Longitude [°E]	Latitude [°N]	z [m]	Manufacturer	Hub [m]	Rated Power [kW]
1	9.500932	51.415222	332.5	TACKE	50	600
2	9.505184	51.415479	342.1	TACKE	50	600
3	9.505911	51.413187	334.8	TACKE	50	600
4	9.495469	51.415950	328.4	VESTAS	53	600
5	9.496596	51.413187	326.2	VESTAS	53	600
6	9.495254	51.420555	317.5	SENVION	93	3,200
7	9.500276	51.420985	317.8	SENVION	93	3,200
8	9.497215	51.418186	324	SENVION	93	3,200
9	9.490783	51.415685	322.9	SENVION	93	3,200
10	9.498627	51.411356	315.8	SENVION	93	3,200
11	9.503233	51.409351	313.7	SENVION	93	3,200
12	9.506630	51.409810	320.6	SENVION	93	3,200

Therefore the impact from the terrain on the wind is not taken into account and this is not valid on complex terrains. Finally the WTG distances shows the spacing between neighbouring WTG and the map shows the design which is used for the calculations, this is the design from figure 16. In this sample case the PARK module is used for a second time with a different turbine type, but I will not go into detail on this again since there is no other difference with the above situation.

After the usage of the PARK module, the AEP production is known. However, it is important that the wind farm does meet all constraints. Therefore 4 other modules are ran to test for different constraints.

1. SHADOW
2. DECIBEL
3. VISUAL
4. ZVI

As the name suggests, the shadow model involves the shadow induced by the rotor and the tower of a WTG. Especially flicker caused by passing rotors can be very annoying for people nearby the turbines. The shadow module does calculate the shadow impact on the surrounding for sunny days with different sun heights. The modules operates under the assumption that all WTG are operating all the time. In practice, the WTG are stopped at too low or too high wind speeds or for maintenance work. After the mapping of these shadows, the module is called again with the curtailment option. Some WTG can then be stopped when flicker occurs at nearby locations. This lowers the impact on the surroundings, but obviously also lowers the AEP somewhat. Curtailment should be minimised, but can help to lower hindrance from a project to the environment.

The decibel model calculates the sound produced by the wind farm. On designated location the different noise demands are set. The module will evaluate the acoustic noise and compare this to the set demands. The result will show if all demands are met or not. If any demands are breached, the distance to al WTG is shown, so the designer can quickly see which turbine is causing the breach.

The third and fourth module are both about the visibility of the wind farm. The visual module is used to generate a view of the landscape as if the park has been constructed already. An example is shown in figure 19. The ZVI (zone of visual influence) maps the visibility of the wind farm up to 15km

Figure 19: Visual render of the wind farm design in Immenhausen, Germany.

of the site. The result will show in an area around the wind farm how many WTG are visible.

Optimized offshore wind farm

The second project in windPRO is an offshore wind farm. For the design of this wind farm the (interactive) optimization options are explored. Since there is not real interest in the wind resource and the behaviour of different modules in different scenarios, a simple available wind resource is used.

The first step in the design process is defining the boundaries of the wind farm site. The site for this farm is shown in figure 21. This figure contains the site layout with the found optimal turbine layout. The site consists of two separate areas, indicated in green. The southern area has a small area in the middle, which is not suited for placing wind turbines. This area is marked with dashed red lines. The upper area is about 0.50 km^2, the lower area is 0,51 km^2 from which 0.027 km^2 is excluded for construction. For this project a Vestas V100 2.6MW turbine is used, this turbine has a 100m high hub and a rated power of 2.6MW.

There are 3 different optimization methods available. I will discuss the first two methods in detail, the third one is outside my scope, but more info an be found in the windPRO user manual chapter 8: optimization [38]. The methods are:

1. Random pattern
2. Regular pattern
3. Noise optimization

Random pattern optimization

The random pattern method is designed mainly to design a wind farm in an area with a lot of different wind conditions close to each other. Normally a high resolution wind map (10m - 25m grid typically) is used to run the random pattern method. However in this example, this is not the case, but the functionality of the method will be analyzed nonetheless. The choice whether the optimizer creates new turbines or moves already defined turbines can be made by the user. The moving of defined turbines allows for mixing up different types of turbines.

To place turbines with the optimzer, three options are offered. The first one is the autofill option, with this option windPRO will simply pack as many turbines as possible in the area while not violating the minimum distance constraints. This will lead to a lay-out with a high number of closely packed turbines. The autofill runs within 10 seconds on this particular wind site. The set-up will have a high yield, but a low efficiency due to wake effects. In this situation, the filled layout contained 38 turbines, and the efficiency is 68.8 %. The investment costs will also be quite high due to the high number of turbines. Note that the results are derived from a single layout, so they are not very significant. It merely shows the different approach each method has.

The next option for the random pattern is called: fast energy layout. This is a quick optimization method. Unlike the autofill option, this algorithm is really optimizing the positions instead of just filling up the lay out. In this method windPRO places turbine on the grid position with the highest energy resource. The next turbine is placed on the next available location with highest energy resource. It does not include any wake effects and is therefore very quick. The fast energy layout option is perfect for making initial sketches of turbine positions. For the test case, this algorithm took just a few minutes to come up with a solution. This solution had 32 turbines and the efficiency was quite higher than the autofill option, 73.9 %.

The last option is the computationally heaviest option, the full energy optimization. The full energy optimization runs like a lot of previously described algorithms and uses the wakes of previously placed turbines to determine the ideal location for the next turbine. The algorithm uses the Jensen wake model [29] , because of its speed. The full calculation for the shown site took to long to complete, but in the manual it improved a fest layout from 97 % efficiency to 98 % efficiency.

Interactivity in the random pattern module

The random pattern method entails some interactive options. First of all, the design is not created with just pressing one button, its an iterative process. The first calculation with the optimize module in the benchmark, then changes are registered and can be tested again to see possible improvements. Changes can be done manually on the map, or by changing the strategy of the different optimization options. For example, the full energy optimization can be allowed to create extra turbine or just to move existing turbines around. It is also possible to lock turbine within certain areas, or let turbine move from one part of the site to another.

Another interesting possibility is the option to lock turbines in place. This allows a user to optimize partial areas by locking a lot, or let just a few wind turbines hold position for practical reasons. It is not possible to stop the optimization procedure and continue afterwards, but you can run multiple optimizations in a row with changes between. After testing multiple different layouts, it is very easy to selected the preferred layout again, since every step in the optimization process is saved. An example of an optimization procedure is shown in figure 20. Here first the autofill option is used as benchmark, then the fast energy layout calculation leads to 32 used turbines instead of 38. These 32 turbines are then manually adjusted a bit, increasing the efficiency by 0.3 %. Finally the full optimization was run to get even further improvement. Unfortunately this optimization did not succeed sue to turbines being placed to close to each other. This led to a lower efficiency, and it violated the set restrictions. Due to the calculation time of about 6 hours, no new run has been submitted. It is probably caused by the relatively high amount of turbine in a small area, leading to conflicts during the optimization procedure.

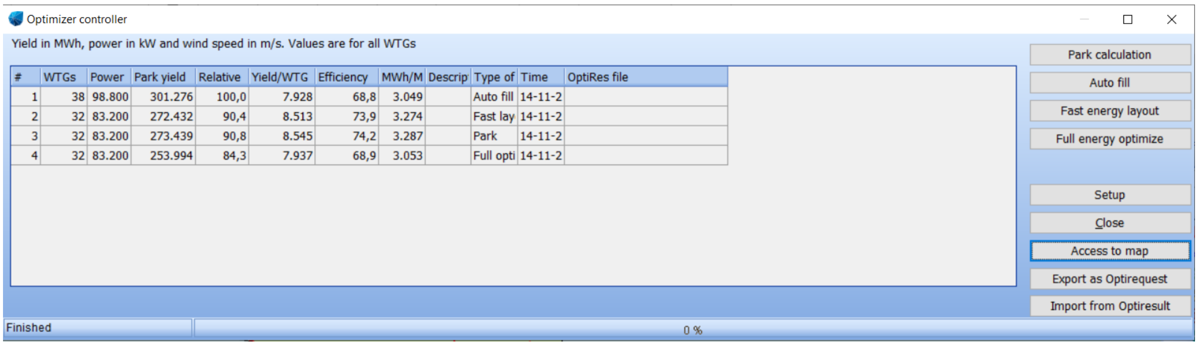


Figure 20: Example of the optimization procedure in windPRO.

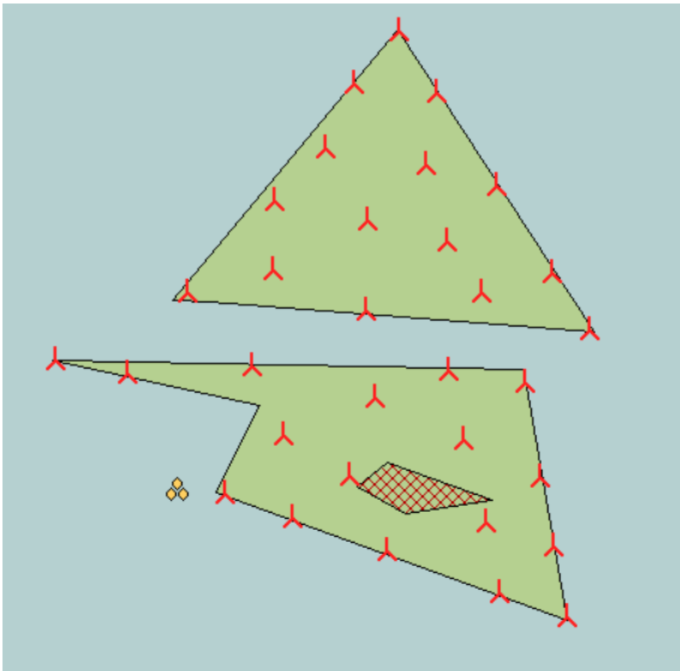


Figure 21: The resulting layout from the described optimization procedure. The red indicators show the turbine positions in the green site. The red dashed pattern is unavailable area within the site.

Regular pattern optimization

The regular pattern method is ideal for location with low variations in wind speed and terrain. This can be flatlands, but also offshore or coastal projects. To use the regular pattern optimization a park design object needs to be placed. This object is simply a group of turbine with a certain pattern. The available patterns are parallel rows or arcs. The patterns can be rotated around a selected fixed turbine in the set-up. The optimization can be started when this park design object is realized and the turbines are placed in and around the site. This optimizer can be used in a manual or in a automatic way.

The manual way does not involve any optimization. It is just moving and turning the park design object around until a decent solution is found. This solution can then be tested with some calculations and corrected. The automatic way does involve some optimization strategies. You can select a minimum required efficiency before a solution will be accepted. The regular pattern optimization work as follows: The user gives in which variables may be changed and in what range. windPRO then tries all possible combinations and output the best solution that meets at least the set minimum efficiency. The available variables are:

- Starting point X and Y of the fixed turbine
- Number of rows
- Number of turbines per row
- Row distance
- In row distance
- Base and side angle
- Row off set

It must be noted that varying all parameters at once with a wide range will lead to long calculation times. It is best for a user to first look globally for the best solution before using smaller step sizes and windows in the optimization window. In the example, a parallel layout of 8x8 turbine is placed on top of the site. The layout is manually scaled so it fits to the site. All turbines outside the area are discarded, but they merely show the pattern the used pattern. In figure 22 the changes made by the algorithm are shown. The green turbines indicate the locations after optimization. Just as the random pattern optimization, 32 turbines are fit in the site. The efficiency is slightly lower than with the random pattern optimization, other patterns might give better results for this irregularly shaped site. The optimization window is shown in figure 23. For each step some parameters have been varied to find an estimation of the optimum, then detailed search around that point is done to find the final layout.

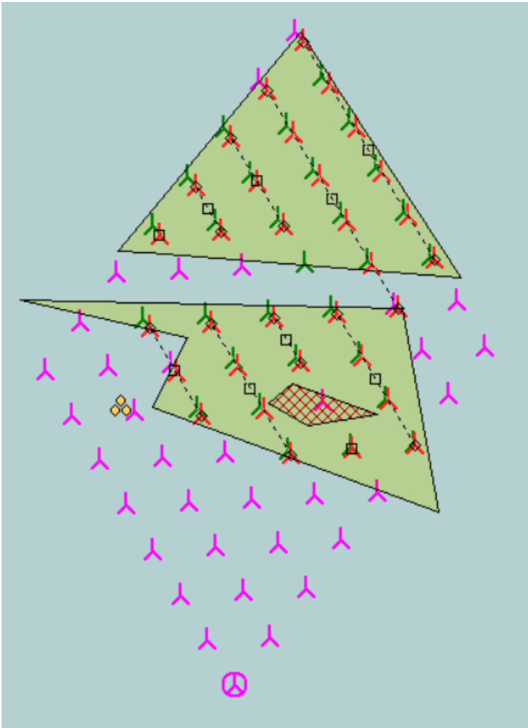


Figure 22: The resulting layout from the described optimization procedure using a regular pattern. The green turbines show the ideal positions, the red turbines the starting positions and the purple turbines are discarded.

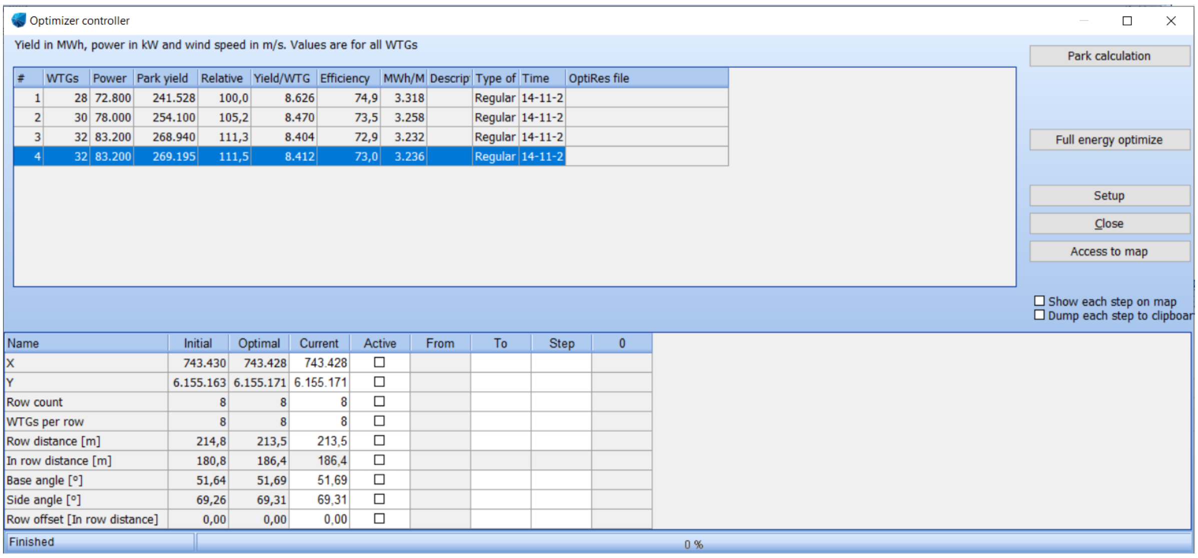


Figure 23: Example of the optimization procedure for a regular layout in windPRO.

B Interview questions

This appendix contains the questions that were asked to two designers during an interview. These two designers both had quite some experience in the wind farm design field. The outcome of this interview is reported in chapter 4.

- What is your experience with wind farm designing?
- Which project(s) have you been working on?
- Which software do you use for designing?
- What is nice about this software?
- What is missing in this software?
- Which interactive options are in the software?
- Where are you missing those interactive options?
- How much time does an optimization run take?
- What would this runtime ideally be?
- Does the final layout, created by the software, require any modifications?
- Why are these modifications necessary?
- Is the turbine positioning the only aspect for optimization or are other aspects included?

C Wind turbine characteristics in Openwind

This appendix contains the wind turbine characteristics in detail. This wind turbine type is used in all experiments.

Table 7: Wind turbine characteristics for the Alstom ECO 100 3.0 Class 1A

Property	Value
IEC Class Ia	Edition 3 NTM TI
Rated capacity	3000 kW
Peak output	3000 kW
Rotor diameter	100 m
Hub height	90 m
Cut-in wind speed	3 m/s
Cut-out wind speed	25 m/s
Number of blades	3
Rotor is tilted back	5 degrees
Power uncertainty	2.4%
Pitch or stall regulated	Pitch
Fixed or variable speed	Variable

D Detailed wind distribution for each scenario

This appendix contains three tables with the detailed parameters for the used wind distributions in the experiments in chapter 5. Each wind rose was built up from 12 bins of 30 degrees. P is the probability of occurrence, A and k are the weibull parameters and mean wind speed is shown in [m/s]. Table 8 shows the realistic scenario, table 9 shows the uniform scenario and table 10 shows the unidirectional scenario.

Table 8: Weibull distribution for the realistic wind distribution.

Sector	Degrees	P [%]	A [m/s]	k	Mean [m/s]
1	345-15	6.993	8.000	2.195	7.085
2	15-45	8.591	8.101	2.390	7.180
3	45-75	7.592	7.700	2.290	6.821
4	75-105	6.494	7.700	2.190	6.819
5	105-135	4.496	6.700	2.190	5.934
6	135-165	4.795	7.300	2.190	6.465
7	165-195	7.792	9.500	2.195	8.413
8	195-225	11.788	10.300	2.295	9.125
9	225-255	18.581	10.900	2.395	9.663
10	255-285	0.969	9.600	2.095	8.503
11	285-315	6.893	8.800	1.995	7.799
12	315-345	6.294	8.600	1.995	7.622
All	-	100	9.053	2.231	8.018

Table 9: Weibull distribution for the uniform wind distribution.

Sector	Degrees	P [%]	A [m/s]	k	Mean [m/s]
1	345-15	8.333	9.000	2.095	7.971
2	15-45	8.333	9.000	2.095	7.971
3	45-75	8.333	9.000	2.095	7.971
4	75-105	8.333	9.000	2.095	7.971
5	105-135	8.333	9.000	2.095	7.971
6	135-165	8.333	9.000	2.095	7.971
7	165-195	8.333	9.000	2.095	7.971
8	195-225	8.333	9.000	2.095	7.971
9	225-255	8.333	9.000	2.095	7.971
10	255-285	8.333	9.000	2.095	7.971
11	285-315	8.333	9.000	2.095	7.971
12	315-345	8.333	9.000	2.095	7.971
All	-	100	9.000	2.095	7.971

Table 10: Weibull distribution for the unidirectional wind distribution.

Sector	Degrees	P [%]	A [m/s]	k	Mean [m/s]
1	345-15	0	-	-	-
2	15-45	0	-	-	-
3	45-75	0	-	-	-
4	75-105	0	-	-	-
5	105-135	0	-	-	-
6	135-165	0	-	-	-
7	165-195	0	-	-	-
8	195-225	0	-	-	-
9	225-255	100	10.9	2.395	9.663
10	255-285	0	-	-	-
11	285-315	0	-	-	-
12	315-345	0	-	-	-
All	-	100	10.9	2.395	9.663