Evolutionary Wind Farm Layout Optimization

DD2365 Advanced Computation in Fluid Mechanics - Project Report

Jonathan Haag

June 03, 2023

Abstract

As the climate crisis intensifies, so does the search for emission-friendly energy resources. With wind power taking on a central role in that quest, optimizing the efficiency of wind farms is of interest from an environmental as well as an economic point of view. This project investigates how an evolutionary algorithm can be applied to find favorable layouts for a simplified 2D porosity-based model of a wind farm. The results suggest that the identified layouts lead to an increase of produced energy of up to 20% and reveal some general trends regarding favorable placement of turbines.

1 Introduction

"Affordable and clean energy" - that is how the United Nations term the seventh of their sustainable development goals. And while the political discourse on global cooperation for the production and distribution of renewable energy continues, the planning, construction, and operation of individual plants remains a challenge in itself. Regardless of these struggles, wind power has cemented its position as one of the key technologies for the energy transition [1, 2, 3].

To supply substantial amounts of energy, wind farms consisting of a few single up to a few hundred turbines are necessary, both on- and offshore. Predicting their performance is difficult due to "the complex multiscale two-way interactions between wind farms and the turbulent atmospheric boundary layer" [4]. One important consequence of these interactions is the so-called wake effect, the aggregated influence of multiple turbines on the overall power production, mainly due to changes of the wind speed, directly affecting the following turbines. Not surprisingly, describing the phenomenon itself and taking it into account when designing new wind farms has been a popular topic in the research community with straightforward practical implications. This has resulted in a variety of wake models, the most prominent being the rather simple Jensen model [5, 6].

These models can then be used to optimize the layout of a wind farm, that is to change the positions of a given number of turbines to maximize the overall energy production while satisfying a set of constraints such as the spacing between the turbines or specific features of the terrain. The review paper [7] indicates that this procedure known as wind farm layout optimization (WFLO) is one of the core challenges for large-scale production of (offshore) wind energy. The authors of [8] point out that it is a NP-hard problem and a variety of optimization algorithms have been used to obtain approximate solutions. These include, among others, random search [9], greedy algorithm [10], particle swarm optimization [11], simulated annealing [12], reinforcement learning [13, 14], and genetic algorithms [15, 16, 17, 18].

Clearly, WFLO poses a problem that is not only of environmental, but also financial interest. In [19], the authors perform a techno-economic assessment, highlighting not only the influence of the layout on the investment risk, but also on resulting tariff prices. However, while full 3D-CFD studies exist, the high computational effort associated with them, combined with the use of gradient-free optimization methods that rely on many evaluations, is limiting, especially for smaller projects. To avoid or at least mitigate this problem, alternative approaches are needed. One is to consider the 2D Navier-Stokes equations coupled with an actuator disk model which describes the change in downstream velocity [20], another to disregard the CFD altogether and instead only consider a modification of Jensen's wake model for the optimization [21].

This paper takes a slightly different route by considering a 2D porosity-based model of a wind farm using the Navier-Stokes Brinkman equations. The main objective is to employ this approach together with an evolutionary algorithm to determine favorable layouts for wind farms with varying number of turbines. Here, the central question concerns the achievable increase in produced energy, while considerations regarding the computational burden are discussed along the way.

The remainder of this report is structured as follows: Section 2 introduces the proposed wind farm model as well as the optimization scheme. Results are presented in Section 3 and discussed in Section 4, before the central findings are summarized in Section 5.

2 Method

This section describes the procedure used to obtain approximate solutions to the WFLO problem, as illustrated in Figure 1. In short, it can be summarized as follows: An evolutionary algorithm is used to generate candidate layouts which are then evaluated by estimating their production with a CFD simulation. The results are fed back to the evolutionary algorithm to create a new set of candidates. This process is repeated until some convergence criterion is met. For this project, the number of iterations since the last global solution update is used for that purpose. The following subsections discuss the two main components of this approach, the wind farm simulation and the evolutionary algorithm, in greater detail. Finally, some implementation details are mentioned.

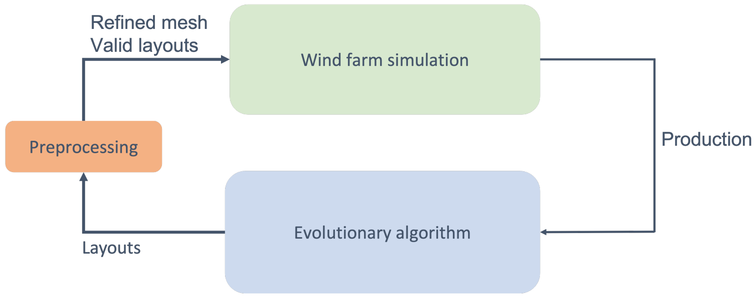

Figure 1: Schematic illustration of the coupling between the CFD simulation and the optimization algorithm.

2.1 Wind farm model

For this project, we assume that the wind flow is laminar and incompressible and can thus be described by the Navier-Stokes equations. To reduce the computational demand, a top-down two-dimensional view is used. However, this yields the question of how to capture the wake effect as any impermeable object placed inside the domain would only redirect the flow. Instead, porous media theory is utilized by solving the Navier-Stokes Brinkman equations

$$\begin{aligned} \frac{\partial}{\partial t}u + u \cdot \nabla u - \frac{1}{\rho}\nabla \cdot \sigma(u,p) + \frac{\nu}{K(x,y)}u = 0 \\ \nabla \cdot u = 0, \end{aligned} \tag{1}$$

with the velocity u, pressure p, density ρ, and viscosity ν. The Brinkman model introduces an additional parameter $K(x,y)$ which controls the permeability of the domain. Notice that for $K \to \infty$ we are left with the regular Navier-Stokes equations. Consequently, we define

$$K(x,y) = \begin{cases} 1/K_{inv}, & \text{if } (x,y) \in turbines \\ \infty, & \text{else} \end{cases} \tag{2}$$

that is, we view the turbines as parts of the domain with reduced permeability. The parameter K_{inv} can be seen as turbine-specific axial induction factor as used in [20] and [21]. Numerical experiments suggest the choice of $K_{inv} = 8 \cdot 10^4$ as this approximately leads to a reduction in wind speed of one third and a wake region of roughly ten times the rotor diameter. These values were chosen based on common results in the literature. Also note that while the velocity is reduced in the wake region, it is increased above and below the turbine, the effects of which will be discussed later on.

The wind farm is described as a rectangular subdomain within a larger rectangular domain of length L and height H. For simplicity, assume an inflow boundary on the left side of the domain with Dirichlet boundary condition on the velocity, $u_x = 1$. On the upper and lower boundary we prescribe slip boundary conditions, i.e., the tangential component of the velocity u_x is not restricted, while the normal component u_y is set to zero. On the outflow side, another Dirichlet boundary condition is applied, this time to ensure zero pressure.

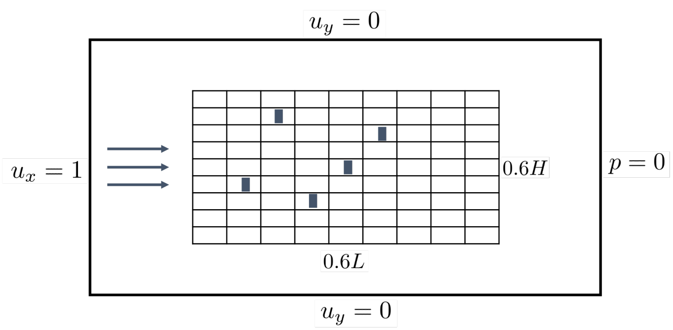

Figure 2: Illustration of the rectangular domain with the wind farm subdomain as a 9x9 grid. As an example, 5 turbines (small rectangles) are placed on the grid.

Figure 2 shows the domain and highlights two more important aspects: Inspired by the actuator disk theory, the turbines are being modeled as small rectangles defined by a center point, width, and height. Further, the subdomain is discretized into a 9x9 grid to reduce the search space for the algorithm. Nonetheless, obtaining a solution by brute forcing is not an option, since for 10 turbines nearly $2 \cdot 10^{12}$ possible layouts exist. Finally, a minimum distance between two turbines in the y-direction is specified so that layouts consisting of two adjacent grid cells with turbines are discarded. This further reduces the search space and is a common measure to obtain more interesting layout suggestions during the optimization, as it prohibits simple straight-line arrangements even when the number of turbines is relatively small.

To approximate the production of a wind turbine, a rather crude but simple estimate is used: The velocity of the incoming wind in normal direction (here, this is simply u_x) is integrated along the upstream boundary of the turbine. For a given layout, we sum the contributions of the individual turbines. Since we are dealing with normalized velocities, the production of a wind farm consisting of n_t turbines of height h_t without any wake losses is given by $n_t \cdot h_t$.

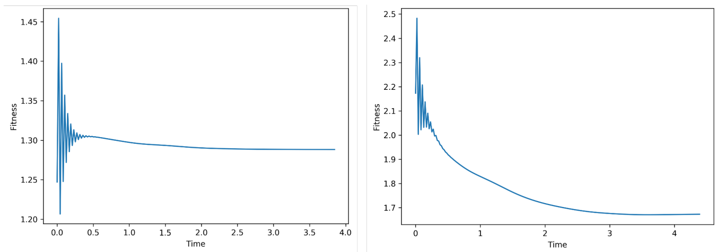

Figure 3: Fitness (production) over time for a single simulation with 8 (left) and 18 (right) turbines.

As we will see in the next subsection, the optimization scheme does rely on the evaluation of a large number of layouts. Therefore, it is important to consider the computational effort of any individual fluid mechanical simulation. In this context, two aspects should be highlighted: First, by monitoring the production we can establish a convergence criterion based on its rate of change to stop the simulation, rather than prescribing a fixed end time T. In fact, the flow usually reaches a steady state after 3-5 seconds, as illustrated in Figure 3. Second, local mesh refinements around the turbine centers can be used to contain the number of mesh cells. This is particularly useful since the turbines are small enough that they become invisible if the resolution is too low, compare Figure 4. However, doubling the resolution, as shown in the first two subfigures, results in more than a fourfold increase in computation time. Checking the above explained spacing constraint and refining the mesh can be viewed as preprocessing steps, see Figure 1.

2.2 Evolutionary algorithm

An evolutionary algorithm is a population-based optimization metaheuristic. As the name suggests, it is heavily inspired by biological evolution and makes use of the concepts of selection, reproduction, recombination, and mutation. In each iteration, also referred to as generation, of the algorithm, a set of candidate solutions is evaluated.

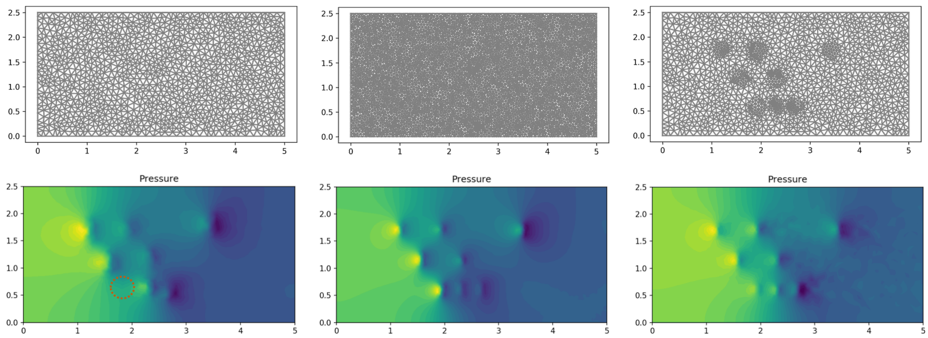

Figure 4: Mesh and resulting pressure field for random layout with 8 turbines and resolution = 32 (left), 64 (middle), and 32 with local refinements (right).

In evolutionary terms, this corresponds to selecting those individuals (wind farm layouts) of a population (set of layouts) that have the highest fitness (energy production). Afterwards, new candidate layouts are created by recombining the best performing individuals, i.e., one turbine at a time is picked randomly from either one of two "parent" layouts. "Child" layouts are then mutated by randomly shifting single turbines to an adjacent grid cell. Finally, the population for the next generation is formed by joining the parent and child layouts and adding additional mutated survivors from the previous generation to maintain the population size. The central steps regarding the crossover and mutation are illustrated in Figure 5. Algorithm 1 contains a pseudo-algorithmic formulation of the main optimization loop, which served as the basis for the implementation.

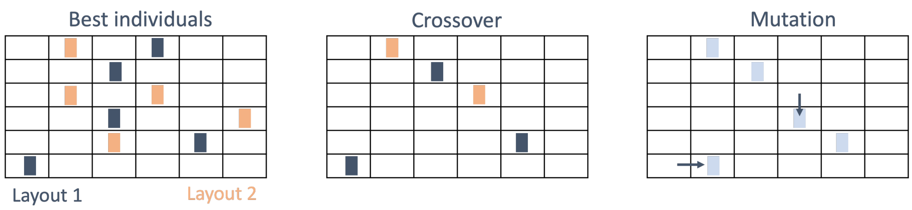

Figure 5: Schematic illustration of the recombination and mutation steps of the evolutionary algorithm. First, two well performing individuals from the previous iteration are chosen (left), then the layouts are recombined to form a new individual (middle). Finally, that individual is mutated (right) and added to the new population.

Algorithm 1 Evolutionary Optimization Metaheuristic

Set parameters, create initial population
while termination condition not met **do**
  Evaluate fitness of individuals
  Select best individuals
  Crossover to create offspring
  Mutate offspring and survivors
end while

It should be noted that evolutionary algorithms can be seen as randomized, greedy search approaches and their ability to find good approximate solutions largely depends on suitable parameter choices. However, even with a good configuration, the sheer number of required evaluations of the objective function, which in this case refer to full simulations of the fluid flow, limit their efficiency. On the other hand, not only one but a set of reasonably good solutions is returned. This is advantageous for the WFLO problem, as practical aspects regarding the construction and logistics of the project may complicate the realization of the best identified layout [22].

2.3 Implementation

The Finite Element library *FEniCS* was used to implement the simplified wind farm model by extending the code for the Navier-Stokes Brinkman example provided in the course repository. The evolutionary algorithm was implemented from scratch in Python using *NumPy*. All plotting is done using *Matplotlib* and *FEniCS* built-in functions. For more details, including the choice of all relevant parameters for the performed experiments, please refer to GitHub, where the full source code and additional information is provided.

3 Results

The implementation was used to find optimized layouts for wind farms consisting of 10, 14, and 18 turbines, respectively. For the sake of brevity only some the obtained results are shown in this section. Additional images and videos with short explanations are also provided on GitHub.

3.1 Optimized production

To begin with, consider the convergence behavior of the algorithm as shown in Figure 6a for the case of 14 turbines. Here, each individual evaluation of fitness function, i.e., each wind farm simulation is shown. Two things stand out: Firstly, the curve seems to flatten out, signalizing convergence. Secondly, there are visibly more gaps related to invalid layouts (zero fitness) early on in the optimization process, which indicates that the general setup works in the sense that the evolutionary algorithm "learns" to satisfy the spacing constraint. Also note that in this case 1725 individuals were evaluated which corresponds to a total runtime of roughly 18 hours on a 2018 MacBook Pro with 2,3 GHz Quad-Core Intel i5. The corresponding plots for 10 and 18 turbines look similarly and are thus not included here. Figure 6b shows the obtained optimized production in comparison to a random starting layout as well as the estimate of a wind farm that does not suffer from the wake effect (compare Section 2.1).

3.2 Optimized layouts

Figure 7 shows the best identified layouts for the three considered scenarios. Also, the third best layout for 14 turbines is shown as an example of a similarly good performing alternative. Despite the clearly visibly differences between the two, the relative discrepancy in production is only 0.5%.

(a) Fitness history, $n_t = 14$ (b) Production(n_t)

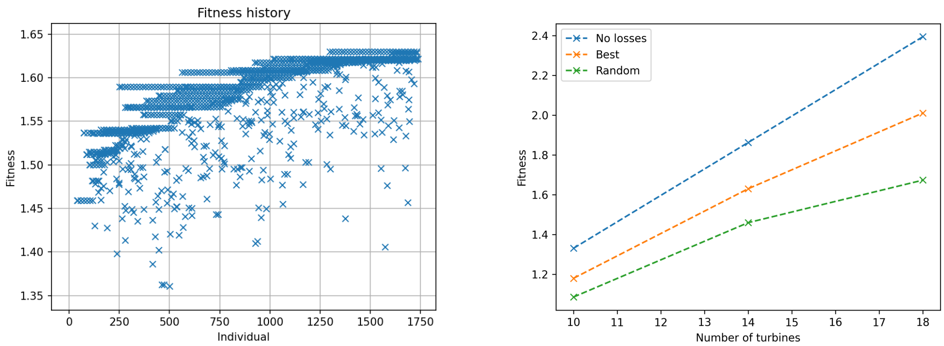

Figure 6: Fitness function evaluations for the 14 turbine wind farm (a) and scaling with the number of turbines (b). The latter shows the production before and after optimization as well as the corresponding loss-free estimate.

(a) $n_t = 10$ (b) $n_t = 14$, best layout

(c) $n_t = 14$, alternative layout (d) $n_t = 18$

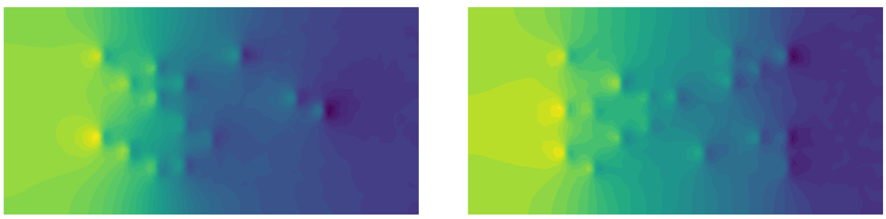

Figure 7: Best found layouts with the evolutionary algorithm for (a) 10, (b) 14, and (d) 18 turbines. Additionally, (c) shows the second best layout for 14 turbines.

As an example, the associated flow field for the 18 turbine wind farm is shown in Figure 8, displaying the wake effect as a feature of the simulation and avoidance thereof by the turbine placement.

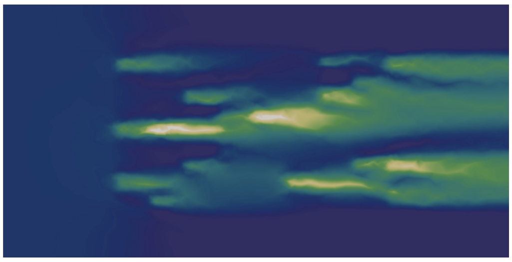

Figure 8: Resulting fluid flow of the best found 18 turbine wind farm.

4 Discussion

The results in the previous section generally indicate that the optimization was successful. One way to check this is to consider the production of the optimized layouts compared to random ones for different number of turbines, see Figure 6b. The increase is between 9% for 10 turbines and 20% for 18 turbines, which is significant. Intuitively, this makes sense, since larger wind farms are more susceptible to wake losses, and thus have a greater potential for improvement through layout optimization. This is backed up by the fact that the production per turbine decreases for larger n_t, which is obvious when comparing the optimized with the loss-free production and a result of the accumulated effects of the wake development.

Looking at the obtained layouts, we can conclude that diagonal layouts seem to perform best. One possible explanation for this is that the wind velocities increase above and below turbines, which allows for leverage of the wake effect, see for example Figure 8. This can also be observed when considering the contributions of the individual turbines. For example, the two turbines with the highest production of the best 14 turbine layout are on the downstream side of the wind farm. Note that overall the distribution is quite balanced and the algorithm seems to find individual positions that avoid the previous wakes quite well. To achieve this, the shape can vary quite significantly: for 14 turbines both, one and two main diagonals, can be observed, while for 18 turbines an hourglass layout is obtained.

Overall, the wake effect seems to be represented quite well with the porosity approach (in terms of size and development of the wake, compare Figure 8), although additional tweaking of the permeability function might lead to even better results. Related literature also often identifies diagonal layouts to be favorable, which matches the findings in this project nicely. While these factors speak for the used 2D Brinkman approach and some measures to limit the computational cost were introduced, the optimization still requires substantial amounts of time on a regular machine. Further, the used approach may lead to premature convergence either by running into local minima, for example, due to unfavorable parameter choices or random initialization or because of a poorly defined termination criterion.

5 Conclusion

This project introduced a 2D top-down, porosity-based modeling approach for a wind farm. By coupling a finite element simulation of said model with an evolutionary algorithm, the wind farm layout optimization problem was solved approximately for different sized farms. Given the assumptions, the approach worked reasonably well and the improved layouts were estimated to deliver up to 20% more energy compared to random ones. Comparing the flow fields and identified layouts with the literature, the approach proved its ability to perform the considered task.

Still, the required computational effort is a barrier, which is why parallelization of the evaluation of different layouts within one generation of the evolutionary algorithm, would be a natural extension of this work. Also, a more detailed comparison with 3D models in terms of the accuracy of the wake model and optimization process seems interesting. In this context, including additional aspects of the 3D world such as terrain into the 2D model could be investigated. This also includes a more complex wind model, allowing for varying velocities and angles, which also requires a more involved estimate of the production of the individual turbines.

References

[1] David W Keith et al. "The influence of large-scale wind power on global climate". In: *Proceedings of the National Academy of Sciences* 101.46 (2004), pp. 16115–16120.

[2] Simon R Sinsel, Rhea L Riemke, and Volker H Hoffmann. "Challenges and solution technologies for the integration of variable renewable energy sourcesâa review". In: *renewable energy* 145 (2020), pp. 2271–2285.

[3] Atika Qazi et al. "Towards sustainable energy: a systematic review of renewable energy sources, technologies, and public opinions". In: *IEEE access* 7 (2019), pp. 63837–63851.

[4] Fernando Porté-Agel, Majid Bastankhah, and Sina Shamsoddin. "Wind-turbine and wind-farm flows: A review". In: *Boundary-layer meteorology* 174 (2020), pp. 1–59.

[5] Niels Otto Jensen. *A note on wind generator interaction.* Vol. 2411. Citeseer, 1983.

[6] I Katic, Jørgen Højstrup, and Niels Otto Jensen. "A simple model for cluster efficiency". In: *European wind energy association conference and exhibition.* Vol. 1. A. Raguzzi Rome, Italy. 1986, pp. 407–410.

[7] Hugo Díaz and C Guedes Soares. "Review of the current status, technology and future trends of offshore wind farms". In: *Ocean Engineering* 209 (2020), p. 107381.

[8] Rabia Shakoor et al. "Wake effect modeling: A review of wind farm layout optimization using Jensen$\times^3 smodel$". In: *Renewable and Sustainable Energy Reviews* 58 (2016), pp. 1048–1059.

[9] Ju Feng and Wen Zhong Shen. "Solving the wind farm layout optimization problem using random search algorithm". In: *Renewable Energy* 78 (2015), pp. 182–192.

[10] K Chen et al. "Wind turbine positioning optimization of wind farm using greedy algorithm". In: *Journal of Renewable and Sustainable Energy* 5.2 (2013), p. 023128.

[11] Ajit C Pillai et al. “Offshore wind farm layout optimization using particle swarm optimization”. In: *Journal of Ocean Engineering and Marine Energy* 4 (2018), pp. 73–88.

[12] Martin Bilbao and Enrique Alba. “Simulated annealing for optimization of wind farm annual profit”. In: *2009 2nd International symposium on logistics and industrial informatics*. IEEE. 2009, pp. 1–5.

[13] Fangyun Bai et al. “Wind farm layout optimization using adaptive evolutionary algorithm with Monte Carlo Tree Search reinforcement learning”. In: *Energy Conversion and Management* 252 (2022), p. 115047.

[14] TC Vyshnav, MC Lavanya, and KC Sindhu Thampatty. “Reinforcement learning based wind farm layout optimization”. In: *2022 International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE. 2022, pp. 1393–1398.

[15] Ying Chen et al. “Wind farm layout optimization using genetic algorithm with different hub height wind turbines”. In: *Energy Conversion and Management* 70 (2013), pp. 56–65.

[16] Dennis Wilson et al. “Evolutionary computation for wind farm layout optimization”. In: *Renewable energy* 126 (2018), pp. 681–691.

[17] Yan Wu et al. “A design methodology for wind farm layout considering cable routing and economic benefit based on genetic algorithm and GeoSteiner”. In: *Renewable Energy* 146 (2020), pp. 687–698.

[18] Luís Eduardo Boni Cruz and Bruno Souza Carmo. “Wind farm layout optimization based on CFD simulations”. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42 (2020), pp. 1–18.

[19] Seiied Mohsen Masoudi and Mehdi Baneshi. “Layout optimization of a wind farm considering grids of various resolutions, wake effect, and realistic wind speed and wind direction data: A techno-economic assessment”. In: *Energy* 244 (2022), p. 123188.

[20] Sjoerd Boersma et al. “Quasi linear parameter varying modeling for wind farm control using the 2D Navier-Stokes equations”. In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 4409–4414.

[21] Xiaoxia Gao, Hongxing Yang, and Lin Lu. “Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model”. In: *Applied Energy* 174 (2016), pp. 192–200.

[22] Michele Samorani. “The wind farm layout optimization problem”. In: *Handbook of wind power systems* (2013), pp. 21–38.