Funded by
the European Union

HORIZON EUROPE PROGRAMME – TOPIC HORIZON-CL5-2021-D3-03-05
Wind energy in the natural and social environment
Research and Innovation action (RIA)

WIMBY

Wind in My Backyard: Using holistic modelling tools to advance social awareness and engagement on large wind power installations in the EU

Grant Agreement No. 101083460
Starting date: 1st January 2023 – Duration: 36 months

Deliverable D2.1
Wind Power Assessment Tool

DOCUMENT INFORMATION

Deliverable number	D2.1
Deliverable title	Wind Power Assessment Tool
Work Package	WP2
Deliverable type	R
Dissemination level	PU
Due date	31.12.2023 (Month 12)
Pages	24
Document version	4.0
Lead author(s)	Andrea N. Hahmann and Neil N. Davis, DTU
Contributors	Nicolas G. Alonso De Linaje, Maria V. Sarcos, Julian Quick, DTU

The WIMBY project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

DOCUMENT CHANGE HISTORY

Version	Date	Author	Description
DRAFT			
0.1	01.12.2023	Andrea N. Hahmann, DTU	Creation
0.2	06.12.2023	Andrea N. Hahmann, DTU	Consolidation of input from contributors
FIRST PEER REVIEW			
1.0	12.12.2023	James Price, UCL and Marianne Zeyringer, UiO	Proofreading and peer review
1.1	15.12.2023	Andrea N. Hahmann, DTU	Consolidation of input from reviewers
COORDINATOR APPROVAL			
3.0	20.12.2023	S. Arapoglou (VUB)	Coordinator review
3.1	20.12.2023	Andrea N. Hahmann, DTU	Consolidation of input from coordinator
3.2	20.12.2023	S. Arapoglou (VUB)	Coordinator approval
FINAL VERSION			
4.0	22.12.2023	S. Arapoglou (VUB)	Format review, version ready for submission

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains

SHORT ABSTRACT FOR DISSEMINATION PURPOSES

Abstract

The WIMBY Wind Power Assessment Tool is an API that evaluates the potential of wind farms in Europe.
The API considers the wind resources, wind farm boundaries, and user-supplied turbine specifications for any European region. The tool is currently set up to optimize for the wind farm's annual energy production, using the TOPFARM integrated wind farm model.

The API combines standalone models previously developed by DTU Wind, which are now coupled. The API uses openly available datasets and user-provided configurations. This report provides a detailed description of the WIMBY Wind Power Assessment Tool and its development process.

TABLE OF CONTENTS

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

LIST OF FIGURES

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

LIST OF TABLES

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

ABBREVIATIONS

Acronym	Description
AEP	Annual Energy Production
AGL	Above Ground Level
API	Application Programming Interface
EZ	Exclusion Zones
GIS	Geographic Information System
GPKG	GeoPackage
NEWA	New European Wind Atlas
NetCDF	network Common Data Form
REST	REpresentational State Transfer
UUID	Universally Unique Identifier

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

EXECUTIVE SUMMARY

This report presents the WIMBY Wind Power API, which combines wind resources, wind farm boundaries, and turbine specs to evaluate the energy potential of wind farms for any region in Europe. The tool couples models developed by DTU, previously stand-alone, to produce the optimized wind farm layout based on openly available datasets and user-provided configuration.

The WIMBY Wind Power API is built to be deployed on a web server. It provides an API interface to calculate the locations of turbines within a user-specified wind farm boundary. The user will provide a location of interest (e.g., a polygon) and necessary wind farm design criteria (number and type of turbines). The optimization tool will run and provide the wind turbines' locations to maximize the wind farm's energy production. The result is the geographic location of each turbine and the AEP for the entire wind farm.

The prototype WIMBY Wind Power API has been delivered on time. However, gaps exist in the development and application of the API. Future work will concentrate on the two models yet to be integrated: the noise and economic models. For the economic model, we consider several candidates of various complexity. The choice of a final model will largely depend on data availability at a European scale.

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

1 INTRODUCTION

The WIMBY Wind Power API combines wind resources, wind farm boundaries, and turbine specs to evaluate the energy potential of wind farms for any region in Europe. The tool couples models developed by DTU, previously stand-alone, to produce the optimized wind farm layout based on openly available datasets and user-provided configuration. Figure 1.1 shows a schematic diagram of the initial concept of the tool. Not all models and their linkages exist at the time of the writing of this report.

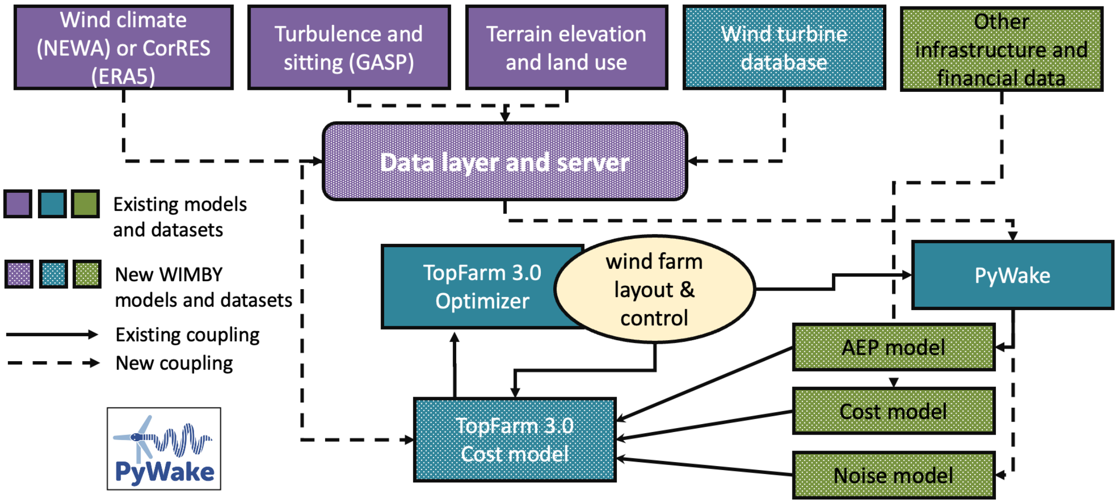


Figure 1.1: Initial proposed concept of the WIMBY Wind Power API. Solid lines and boxes show existing connections, models, and databases; dashed lines and boxes show those to be developed throughout the project.

For the AEP, the wind resource distribution (in terms of the distribution of wind speed and direction) is given to a wind farm layout model, which, combined with the wind turbine characteristics, estimates the wake losses and, thus, the maximum potential yield, before additional losses, of the wind farm.

The input wind resource considers the microscale (sub-kilometer) details of topography, land use, and their effect on the wind. TOPFARM 3.0

[1], a Python package for wind farm optimization, is used to optimize the wind farm layout according to the target minimization parameters. In the current version, TOPFARM is set up to maximize the annual energy production. The optimized layout, in turn, is then used in other WIMBY applications. For example, in determining noise levels, flicker, and the economic potential of the wind farm. These applications are currently under development by DTU and other WIMBY partners. The WIMBY Wind Power API and other calculations require information about the wind resources (provided by T1.1, see Deliverable D1.1 [2]) and the wind turbine characteristics, which are compiled for the project.

This report is divided as follows: Section 2 describes the data used. Section 3 describes the tool architecture and implementation. Section 4 summarizes the report's contents and identifies gaps and future developments.

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

2 DATA

The WIMBY Wind Power API uses data from various sources. Data on wind resources (Section 2.1) are used to compute the AEP of a wind farm, given the Wind turbine characteristics (Section 2.2). The area where wind turbines can be built is determined based on exclusion zones (Section 2.3). These three types of data are detailed below.

2.1 Wind resources data

Wind resource data for the tool comes from the New European Wind Atlas (NEWA) database, described in WIMBY deliverable D1.1 [2]. The particular data used is the NEWA microscale database [3]. The data are provided on a grid spacing of 50 m $\times$ 50 m and 3 vertical levels at 50,100 and 200 m AGL. The variables wind direction frequency f and the two-parameter Weibull wind speed distribution is provided for each 12 wind direction sectors of 30°. Additional wind-related parameters can be derived using the frequency and Weibull parameters and performing a weighted sum over all 12 sectors. For example, to obtain the mean wind speed, we use

$$\overline{u}_{\mathrm{MI}} = \sum_{n=1}^{12} f_n A_n \Gamma(1 + 1/\beta_n) \tag{2.1}$$

where the subscript MI stands for microscale, Γ is the Gamma function and n is the sector. A and β are the scale and shape Weibull parameters. Similarly, the Annual Energy Production ($\overline{\mathrm{AEP}}_{\mathrm{MI}}$) and Capacity Factor ($\overline{\mathrm{CF}}_{\mathrm{MI}}$) can be derived once a wind turbine power curve is chosen. Because the NEWA microscale wind climatology represents a 30-year average (1989–2018 [1]), these averaged quantities also represent that period.

2.2 Wind turbine data

Several wind turbine parameters are required to carry out the AEP calculation and optimize the wind turbine locations. These include the height of the rotor, rotor diameter and the power and thrust curves. The power curve of a wind turbine is a table that indicates how large the electrical power output will be for the turbine at different wind speeds. The thrust is the axial force applied by the wind on the rotor of a wind turbine; it can also be tabulated as a function of the wind speed. Other applications also require noise production but are outside the scope of this report.

---

[1] This data will soon be expanded to cover 1989–2022

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

A generic wind turbine is used in the prototype WIMBY Wind Power API presented here. Later in the project, the wind turbine characteristics will be provided by the user or determined from the type of wind turbine chosen by the user. Such details have yet to be agreed upon.

2.3 Exclusion zones

Exclusion Zones (EZ) data comes from several sources since it covers variables associated with various fields. So far, the tool takes Roads, Protected Areas, and Marine Protected Areas as a baseline, although several more variables are to be implemented. For the current version of the tool, the EZ database only has files of the vector class in GPKG format, which are discrete exclusion features such as the above-mentioned (Protected Areas, Roads, and Marine Protected Areas). However, further development will include EZ layers of the raster class for continuous exclusion futures (e.g., flooding probability, elevation, among others).

Several meetings have been organized among the WIMBY partners to decide on the various EZs necessary for the WIMBY Wind Power API. As these layers will also be used in the WIMBY Web GIS tool, discussions are still ongoing between WP1, WP2 and WP5.

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

3 METHODS

3.1 Introduction

The tool is built to be deployed on a webserver and provides an API interface to calculate the locations of turbines within a wind farm boundary. The user will provide a location of interest (e.g., a polygon) and necessary wind farm design criteria, including the turbine type, hub height, and a number of turbines. The optimization tool will run and provide the wind turbines' locations to maximize the wind farm's energy production. The result is the geographic location of each turbine and the estimated AEP for the entire wind farm.

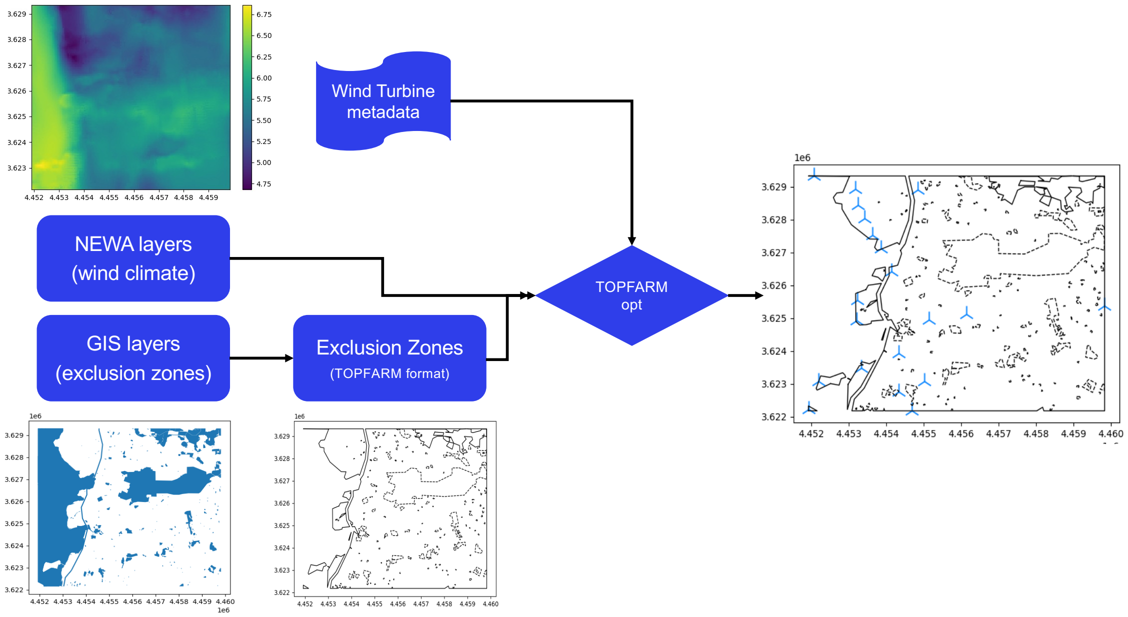


Figure 3.1: Flow diagram of the main components of the WIMBY wind power tool.

Figure 3.1 is the flow diagram of the software tool, where the wind resources (upper left), exclusion zones (bottom left) and wind turbine data are combined to create the optimized wind farm layout (right). The effect of the terrain and land use on the wind is implicitly included in the input wind resource data. This chapter describes the methods used in these three phases.

3.2 Application

The application is developed as a REST API. This means that it uses http requests to communicate. The API main purpose of the API is to provide a calculation engine for the WIMBY web tool developed as part of WP5. At present, the exposed processes are only focused on the calculation itself. As the web tool is developed, additional end points will be added as necessary to support additional functionality.

3.2.1 Docker container

The application is built within a Docker container, isolating all the application code and dependencies in a single package to ensure quick and reliable behavior between different computer environments. The container can be used during development on a user's local machine and then deployed to a web server, where users can perform their desired analysis through the available endpoints.

3.2.2 Endpoints

The application defines three endpoints, which were designed according to tool design and are listed as the following:

upload_polygon: Creates a polygon entry on the server based on a user-provided GeoJSON[1] polygon. The UUID of the created polygon is returned to be used in future API calls.

preproc_inputs: Takes the user's polygon UUID and performs the spatial calculations over the Wind resources and exclusion zone data (described in Sections 2.1, 2.3 for data details and Section 3.3 for the method.) This endpoint returns two intermediate files: one vector file containing the EZ layers and one raster file containing the wind resource data, each clipped to the requested area.

site_turbines: Takes the ids of the two datasets created from the **preproc_inputs** endpoint and the wind turbine characteristics (e.g., number of turbines, turbine type, and turbine hub height) as input. These are used to run the TOPFARM optimization routine, to place the wind turbines in the requested area. The output is a JSON file with every wind turbine location and index. Currently the code assumes that all turbines will fit in the area, but in the future, the endpoint will return an error code with a message about there being too many turbines for the given area.

---

[1] format for encoding a variety of geographic data structures

3.3 Pre-processing of the Exclusion Zone data

To provide a result in a reasonable time, the exclusion zone (EZ) datasets were pre-processed to create an optimized layer for clipping and sharing with TOPFARM.

3.3.1 Standardizing and merging the EZ layers

The EZ layers are provided in different structures, map projections, and file formats (e.g., Figure 3.2a). Therefore, an important step in processing the data is converting them to the same format and projection. Continuous EZs (e.g., flooding probability) are represented using the raster data structure, while discrete fields (e.g., protected areas) are reprezented as vector features. The Cloud Optimized GeoTiff[2] [4] (COG) and NetCDF file formats were chosen for the raster layers, and the GeoPackage (GPKG) [5] and GeoJSON [6] formats are used to store the vector layers. These formats were chosen for their performance, and ease of use with the common Python GIS libraries (Rasterio [7], Xarray [8], and Geopandas [9]).

One important step of the data optimization was combining all of the various exclusion zones into a single layer. However, in order to do this, first the raster layers had to be discretized into vector layers as required by TOPFARM. This was done by classifying the data into discrete categories (see Figure 3.2b). Once all layers were represented as vectors, they could be reprojected and combined into a single layer, which is passed to the optimization tool.

3.3.2 Preparing the NEWA microscale data

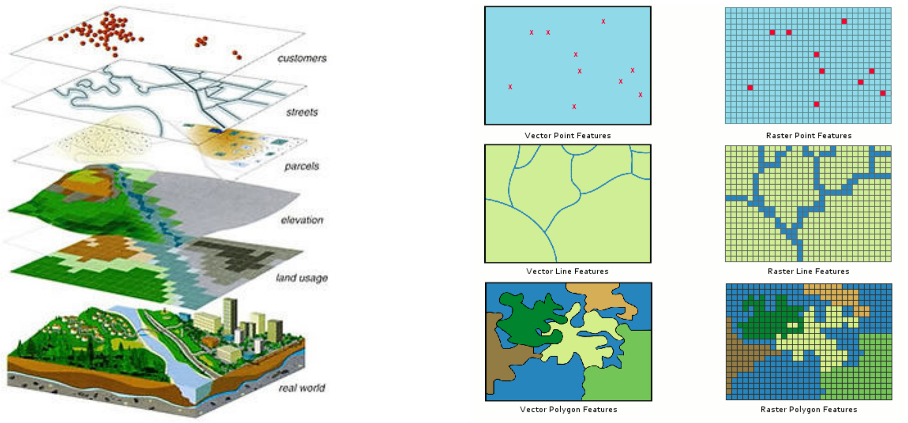
The wind resource data used for this tool is the NEWA microscale climate detailed in Section 2.1. As the data is at 50 m resolution for all of Europe, the data consists of several terabytes of data. Therefore, to allow for this data to be accessed from a single file, it was converted to a Zarr archive. This allows for us to use the Rioxarray [12], Xarray [8], and Dask [13] packages to work with the data, avoiding the saturation of RAM memory and ensuring maximum performance through parallelization.

3.3.3 Clipping the GIS data

When the **preproc_inputs** is called, the application reads in the data prepared in 3.3.1 and 2.1, clips it to the region specified by the user (see Figure 3.3), and converts it to a format that works well with TOPFARM. The NEWA data is clipped using the selection method of XArray's dataset, while the EZ layer is clipped using standard GeoPanda's tools. The EZ layer is passed to TOPFARM as a GeoPanda's GeoDataFrame, while the

---

[2] GeoTIFF is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file.



(a) Example of EZ layers being combined over an area. The different layers represent different variables such as roads or elevation. Source [10].

(b) Example of raster vectorization. The left column shows the vectorized raster, and the right column shows the raster source data. Source [11].

Figure 3.2: Combining and formatting EZ layers. Panel (a) shows the combination of different EZ layers, and panel (b) shows an example of raster reformatting into a vector class.

NEWA data is converted to a PyWake XRSite object (further details in 3.4.1).

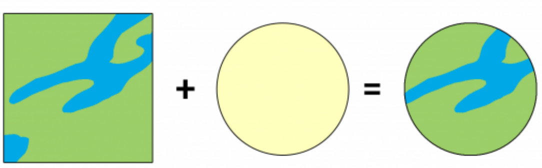


Figure 3.3: Clipping the wind resource layer. Source [14].

3.4 TOPFARM optimization

The information of the EZ layers and NEWA microscale data is used to perform wind turbine layout optimization with the open source tool TOPFARM. The goal is to maximize AEP for a given number of wind turbines and turbine positions while considering some constraints such as minimum inter-turbine spacing and boundary constraints. For this study, the terrain (or wind farm area) is characterized by non-uniform wind resource from the microscale data, and several exclusion zones are defined from the EZ layers corresponding to land and marine protected areas as well as roads. The changes in wind resources due to elevation and land use change are implicitly included in the wind resource data. However, changes in elevation are not included in the wake calculation and layout optimization.

3.4.1 Read and format EZ layers and NEWA microscale data

The NEWA microscale data are processed to create a PyWake XRSite object and the EZ layers are used to define the exclusion/inclusion zones as boundary constraints. This allows for the representation of non-convex exclusion zones and non-uniform wind resource. The roads are buffered by 50 m, which was necessary to give the associated lines an area, and the other exclusion zones (e.g., protected wildlife areas) are buffered by 10 m. These buffering parameters are used for demonstration purposes, and the production implementation will likely depend on the turbine rotor diameter.

3.4.2 The optimization problem

The software tool relies on an optimization algorithm to return the desired wind turbine locations for a given site. As previously mentioned, the goal is to maximize the AEP of the wind farm while taking into account wake losses due to wind turbine interaction. TOPFARM uses the information from the EZ layers and buffering parameters to create its own exclusion and inclusion zones that are then fed to the workflow as a set of boundary constraints. In addition, a minimum distance between turbines is specified and used as spacing constraints. In mathematical form, the optimization problem is defined as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{maximize}} \quad & \text{AEP}(\boldsymbol{x},\boldsymbol{y}) \\ \text{subject to} \quad & \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq S_{\text{min}}\, \forall i,j : i \neq j \\ & C_i \geq 0 \\ & x_l \leq x_i \leq x_u \\ & y_l \leq y_i \leq y_u \end{aligned} \tag{3.1}$$

where x and y are the turbine coordinate positions (with i and j defining a turbine-specific index), C_i is the exclusion zone distance constraint associated with turbine i (defined in [15]), and AEP is the annual energy production. The positions of the turbines are contained to the area selected for the wind farm. Thus, the turbine positions $\boldsymbol{x}$ and $\boldsymbol{y}$ have lower and upper bounds, which are represented by the lower horizontal and vertical bounds x_l and y_l, respectively. Similarly, the upper horizontal and vertical bounds are defined as x_u and y_u, respectively. The inter-turbine

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

spacing is represented by S_{min} which corresponds in this case to 2 turbine diameters in meters.

The AEP is calculated from

$$\text{AEP}(x,y) = \; 8760 \sum_{j=1}^{N_\theta} \sum_{u=1}^{N_u} \sum_{t=1}^{N_{wt}} P(u_{ijt}) \rho \left(U_{\infty i}, \theta_{\infty j}\right), \tag{3.2}$$

where P is power from the power curve, ρ is the site probability mass function, u_{ijt} is the local wind speed including the effects of the ambient wind speed and the turnings due to the complex landscape and 8760 is the number of hours in a year. The local wind speed is represented by the free stream wind speed (i), turbine location (t) and free stream direction (j). To model the wake effects, we used the NOJ Jensen wake deficit model [16] with a squared sum superposition model, as implemented in the PyWake tool.

3.4.3 Optimization of wind turbine positions

The choice of the optimization algorithm will determine the speed in which users can obtain a final solution for the wind farm layout. For this software tool, the main goal is to reduce as much as possible the optimization time while still yielding accurate results. In this case, we have chosen the site shown in Figure 3.1 to measure the performance of an optimization algorithm when a large number of exclusion zones are present. The number of inclusion/exclusion zones will impact the simulation time as the optimizer will need to consider different amount of boundary constraints.

As a preliminary study, the heuristics algorithm *smart-start* was used to perform an optimization of 50 wind turbines using the sample site shown in Figure 3.1. The algorithm works by placing wind turbines sequentially in the areas with the best wind resource to make a better guess for the initial wind farm layout. It considers the wake effects of the already added wind turbines as well as the boundary and spacing constraints, making sure these are not violated. The placement of the turbines is done with a degree of randomness, represented by the parameter r_{pct}, which allows to place the next turbine randomly at one of the n best positions. A configuration of $r_{pct} = 0$ places the turbines at the location with highest AEP, whereas $r_{pct} = 100$ makes the placement completely random, ignoring the AEP value and considering only the spacing and boundary constraints. In addition, the grid resolution defines the number of potential locations (number of points) within the domain. A high resolution grid

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

will yield better AEP but be computationally costly. On the other hand, a lower resolution grid will converge faster but result in a lower AEP value.

To find the best configuration of *smart-start* that could result in a fast optimization, a large set of parameters were examined to run the simulations, as seen in Table 3.1. For the wind speeds, a range between 3 and 25 $\mathrm{m\,s^{-1}}$ is selected, and for the wind direction a range of 0–360° is chosen. The wind speed and wind direction are discretized to either increase or decrease the number of bins when performing the AEP calculation. A low discretization yields less wind speeds or wind directions to study, which can affect the simulation time.

Table 3.1: Grid Sweep Parameters chosen for the Smart-Start algorithm

Parameter Name	Grid Sweep Parameters
Randomness percentage [%]	10, 30, 50, 70, 90, 100
Resolution	50, 100, 200, 400
Wind speed discretization [$\mathrm{m\,s^{-1}}$]	2, 3, 5, 13
Wind direction discretization [degree]	6, 12, 24, 90, 180, 360
Number of seeds	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

The results can be seen in Figure 3.4, where fairly fast optimizations can be obtained with high values of AEP.

An optimization of 50 turbines using the *smart-start* algorithm can take from less than 10 seconds to 1 hour depending on the set up. A simulation of 4 seconds can yield an AEP of 256.81 GWh, whereas the maximum AEP was found to be 281.40 GWh for a total workflow time of 5 minutes.

However, from Figure 3.4 it can be seen that there is a possibility for increasing AEP without sacrificing on computational time. If we apply a filter to the energy production as in $270 \leq \mathrm{AEP} \leq 280$ GWh, it is possible to obtain 276 GWh in 8 seconds, which represents a 7.4% increase in energy production for only 4 more seconds of simulation time. The parameters linked to this set up correspond to:

- $r_{pct} = 10\%$
- grid points = 50
- wind speed = 5 $\mathrm{m\,s^{-1}}$
- wind direction = 6°

For the wind speed and wind direction, the parameter refers to the discretization, for the ranges of 3 $\mathrm{m\,s^{-1}}$ to 5 $\mathrm{m\,s^{-1}}$ for the wind speed and 0°

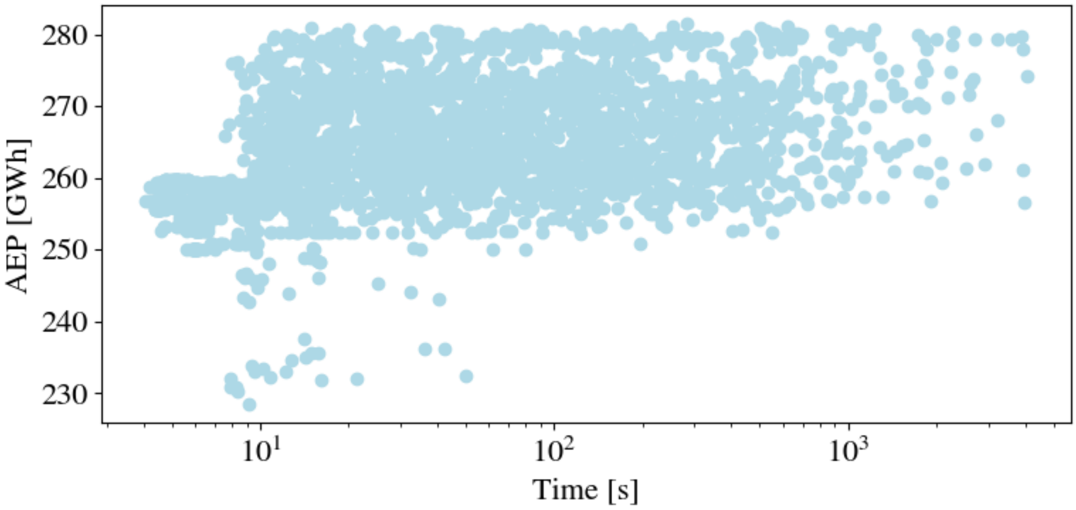


Figure 3.4: Simulation results from the grid sweep optimization performed with the *smart-start* algorithm for the case of 50 wind turbines. The site selected for study corresponds to our DTU Risø campus, which is represented with non-uniform wind resource and exclusion zones.

to 360° for the wind direction. The domain is discretized with a grid of 50 × 50 points.

Currently, the tool works by considering the exclusion zones of a selected area and using them as boundary constraints within the optimization problem. The NEWA microscale data gives the wind resource, which PyWake uses to create an XRSite object. The site is represented by a dataset containing the x and y coordinates of the domain, the wind direction sectors and heights. For the optimization, the *smart-start* algorithm was chosen to perform a preliminary study to maximize AEP for the site shown in Figure 3.1. A series of parameters were chosen to explore the performance of the algorithm and to suggest a specific setup that can yield very fast solutions to avoid large user wait times. From the results in Figure 3.4, it can be seen that the algorithm can perform quite fast optimisations in less than 5 seconds. However, increasing the user wait time results in higher AEP values.

3.5 Current gaps

Many tests will naturally be required once the Wind Power API is integrated into the web interface. For example, tests will be required to access if the calculation is possible because it is possible that the clipping returns an area of zero. It is also possible that the area is too large and the web interface will exit because a limit time is reached. These considerations are outside the scope of this report.

Some application models are yet to be incorporated into the calculations. For example, the economic model that will calculate the cost of a wind farm and the Levelized Cost of Energy (LCOE) is not yet incorporated. Also, the noise model, a shared task between DTU and ETH, is still under development. Both these tasks depend on the wind turbine layout and wind climate already calculated in the Wind Power API.

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

4 GAPS AND FUTURE WORK

As mentioned in the previous section, gaps exist in the development and application of the API. Future work will concentrate on the two models yet to be integrated: the noise and economic models. For the economic model, we consider several candidates of various complexity. The choice of a final model will largely depend on data availability at a European scale.

The greatest challenge within the next six months is integrating the WIMBY Wind Power API within the web server. A variety of tests and checks will need to be built in. Also, the ability to loop through various polygons and optimizations interactively will need to be included.

This project has received funding from the European Union's research and innovation programme Horizon Europe under the grant agreement No. 101083460. This document reflects only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Riccardo Riva and Jaime Yikon Liew and Mikkel Friis-Møller and Nikolay Krasimirov Dimitrov and Emre Barlas and Pierre-Elouan Réthoré and Mads Mølgaard Pedersen. *Welcome to TOPFARM*. URL: https://topfarm.pages.windenergy.dtu.dk/TopFarm2/index.html (visited on 07/03/2022).

[2] Andrea N Hahmann, Neil N. Davis, and Nicolas D. Gonzalez Alonso De Linaje. *WIMBY Deliverable D1.1 Report: Description of the wind resources API*. Tech. rep. DTU Wind, 2023.

[3] Consortimum NEWA. *Web interface to the New European Wind Altas*. 2018. URL: https://map.neweuropeanwindatlas.eu/ (visited on 12/05/2023).

[4] *Geotiff*. URL: https://pypi.org/project/geotiff/ (visited on 12/15/2023).

[5] *OGC GeoPackage*. URL: https://www.geopackage.org/ (visited on 12/15/2023).

[6] *GeoJSON*. URL: https://geojson.org/ (visited on 12/15/2023).

[7] *Rasterio: access to geospatial raster data*. URL: https://rasterio.readthedocs.io/en/stable/index.html (visited on 12/12/2023).

[8] *Xarray documentation*. URL: https://docs.xarray.dev/en/stable/ (visited on 12/12/2023).

[9] *GeoPandas*. URL: https://geopandas.org/en/stable/ (visited on 12/12/2023).

[10] Stavros Kolios et al. *GIS and Environmental Monitoring*. Vol. 20. Jan. 2017. ISBN: 978-3-319-53084-0. DOI: 10.1007/978-3-319-53086-4.

[11] ESRI. *Rasterizing features for 3D*. URL: https://desktop.arcgis.com/en/arcmap/latest/extensions/arcglobe/rasterizing-features-for-3d.htm (visited on 12/03/2023).

[12] Alan D. Snow et al. *corteva/rioxarray: 0.15.0 Release (0.15.0)*. 2023. DOI: Zenodo.https://doi.org/10.5281/zenodo.8247542. URL: https://corteva.github.io/rioxarray/stable/index.html (visited on 12/13/2023).

[13] *Dask — Scale the Python tools you love*. URL: https://www.dask.org/ (visited on 12/13/2023).

[14] GIS Geography. *Clip tool in GIS*. URL: https://gisgeography.com/clip-tool-gis/ (visited on 12/03/2023).

[15] Javier Criado Risco et al. “Gradient-based Wind Farm Layout Optimization With Inclusion And Exclusion Zones”. In: *Wind Energy Science Discussions* (2023), pp. 1–24. URL: https://wes.copernicus.org/preprints/wes-2023-5/.

[16] N. O. Jensen. *A note on wind generator interaction*. Tech. rep. 2411. 1983. URL: https://api.semanticscholar.org/CorpusID:55059791.