Wind Farm Layout Optimization problem

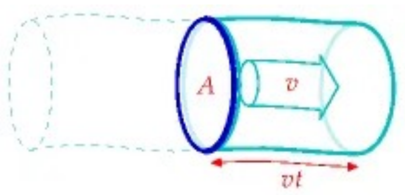
I. Understanding Wind Farm Problems

The physics of wind turbines

A wind turbine converts the **kinetic energy** of the incoming air into a rotational energy acting on the rotor blades, which in turn will set the connected generator going.

$$E = \frac{1}{2}mv^2 = \frac{1}{2}(Avt\rho)v^2 = \frac{1}{2}At\rho v^3$$

We can imagine the air moving toward the rotor in cylindrical form. Its mass ***m*** can be expressed as a function of the air density **ρ***,* the rotor area ***A*** and the length of the "air cylinder", which is equal to speed **v** times ***t.***



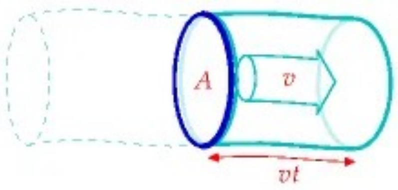
$$P = \frac{E}{At} = \frac{1}{2}\rho v^3$$

The physics of wind turbines

The wind power is the kinetic energy per unit time:

$$P = \frac{E_k}{t} = \frac{\frac{1}{2}mv^2}{t} = \frac{\frac{1}{2}\rho Avtv^2}{t} = \frac{1}{2}\rho Av^3$$

The wind power increases with the cube of the wind speed. Therefore, the selection of a "windy" location is very important for a wind turbine.

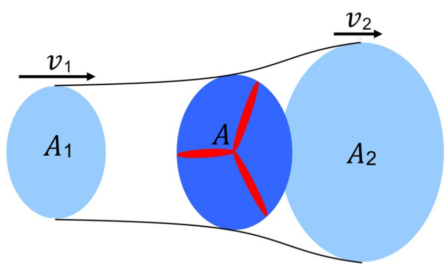


The physics of wind turbines

The effective usable wind power is less than indicated by the above equation. The wind speed behind the wind turbine can not be zero, since no air could follow. Therefore, only a part of the kinetic energy can be extracted.

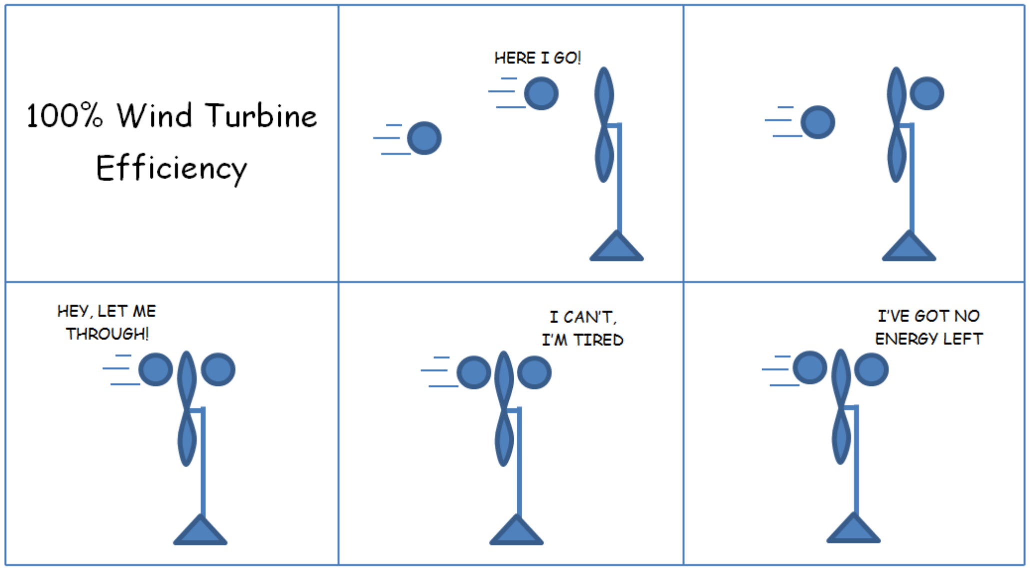
The wind speed before the wind turbine is larger than after. Because the mass flow must be continuous, A v = constant, the area A2 after the wind turbine is bigger than the area A1 before. The effective power is the difference between the two wind powers:

$$P_{\text{eff}} = P_{\mathbf{1}} - P_{\mathbf{2}} = \frac{\Delta V \rho}{2\ \Delta t}\left(v_{\mathbf{1}}^{\mathbf{2}} - v_{\mathbf{2}}^{\mathbf{2}}\right) = \frac{\rho\ A}{4}(v_{\mathbf{1}} + v_{\mathbf{2}})\ \left(v_{\mathbf{1}}^{\mathbf{2}} - v_{\mathbf{2}}^{\mathbf{2}}\right)$$



The physics of wind turbines

The maximum percentage of the potential energy that can be extracted is 59% (Betz' law).

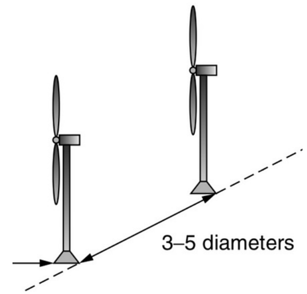


Visualization of wake effect

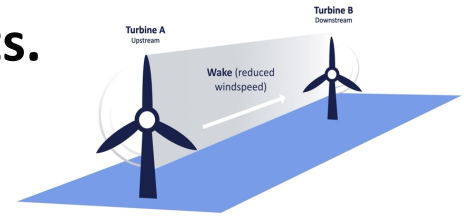
https://www.youtube.com/watch?v=qEtcCjln-0Q

In wind farms, turbines are **spaced close together** for economic benefits related to land use and infrastructure such as access roads and transmission lines.

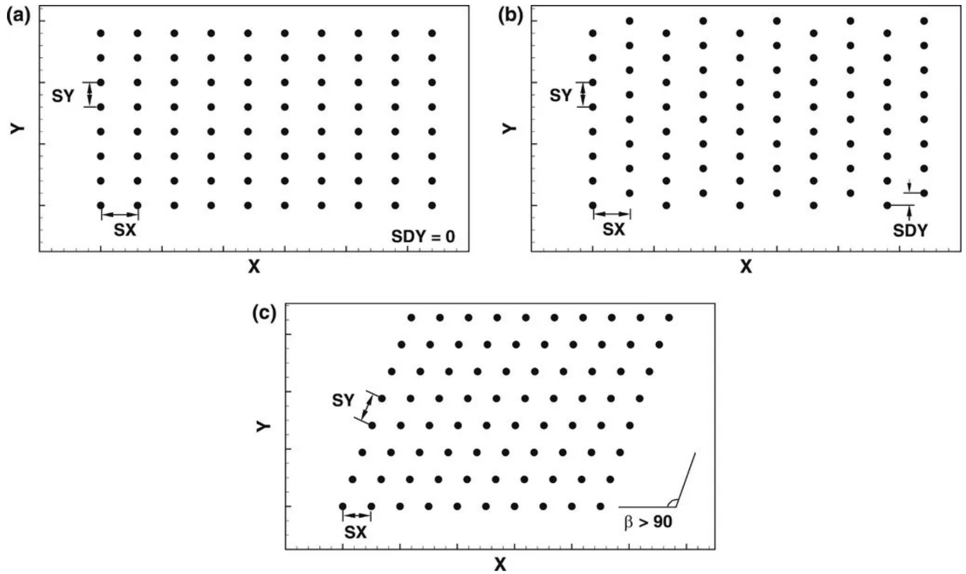
However, this proximity causes turbines to be affected by **the turbulent wakes** of others upwind, which turbine-control systems do not account for.



Ideally, turbines should be spaced as far apart as possible to maximize energy production, but this would increase costs.



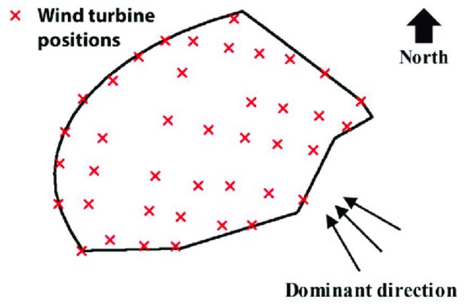
The heuristic approach

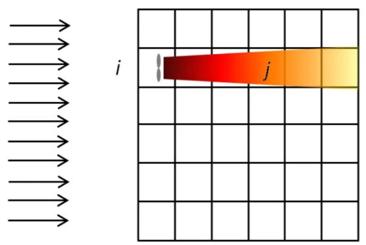


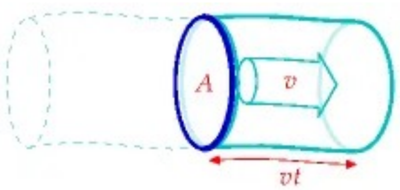
More recent studies suggest that this kind of heuristics leads to **sub-optimal efficiency**, especially in large wind farms with hundreds or thousands of turbines

Wake Effect

- Wake is a long trail of wind which is quite turbulent and slowed down when compared to the wind arriving in front of the turbine.
- Wakes may cost as much as 10-20% of energy losses to the wind farms.







Jensen wake model

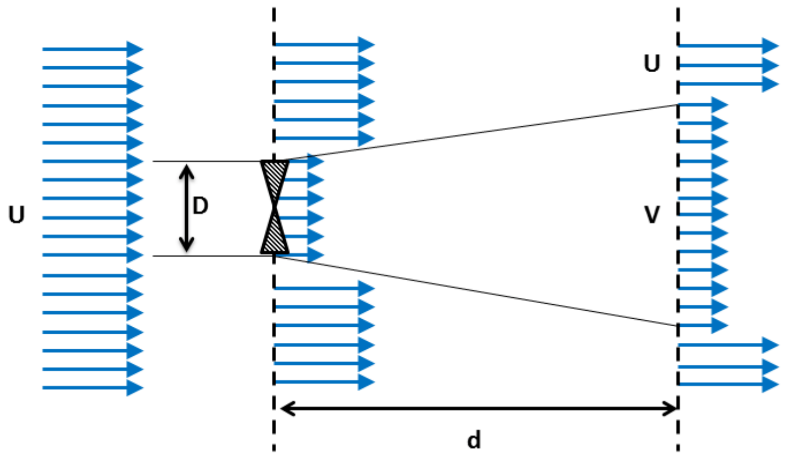


Figure 1: Jensen’s wake effect model.

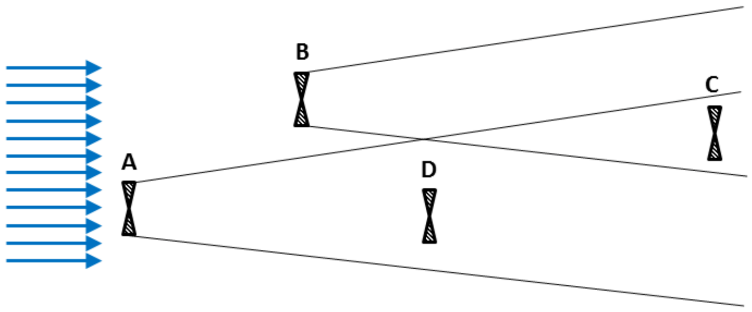
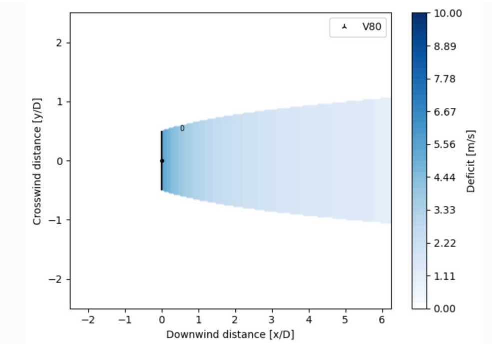
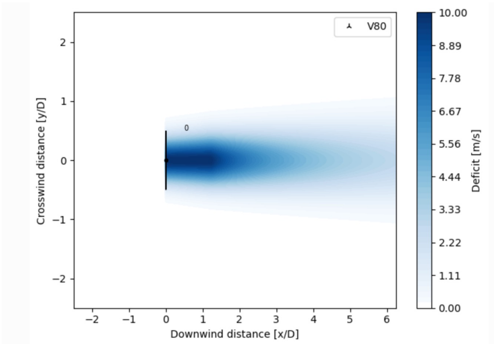
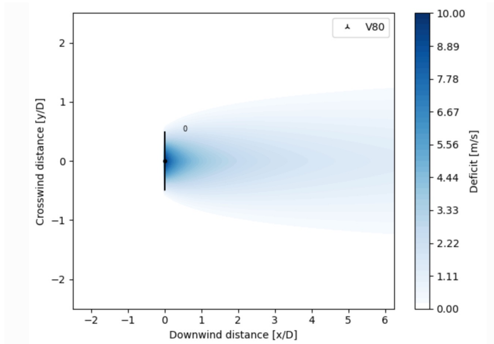


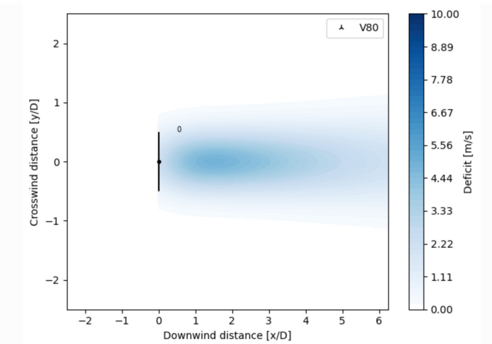
Figure 2: Turbine affected by other turbines’ wake effect.

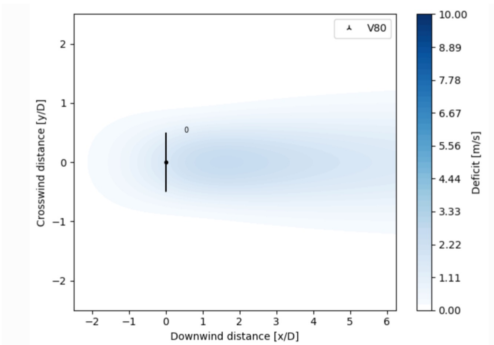
Different wake models

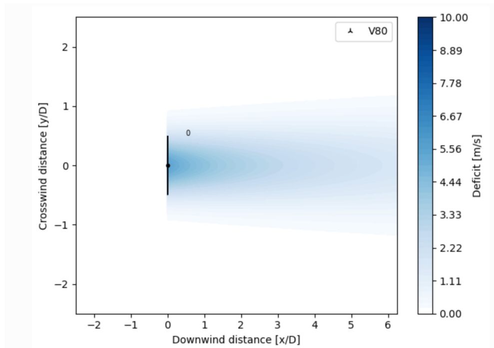




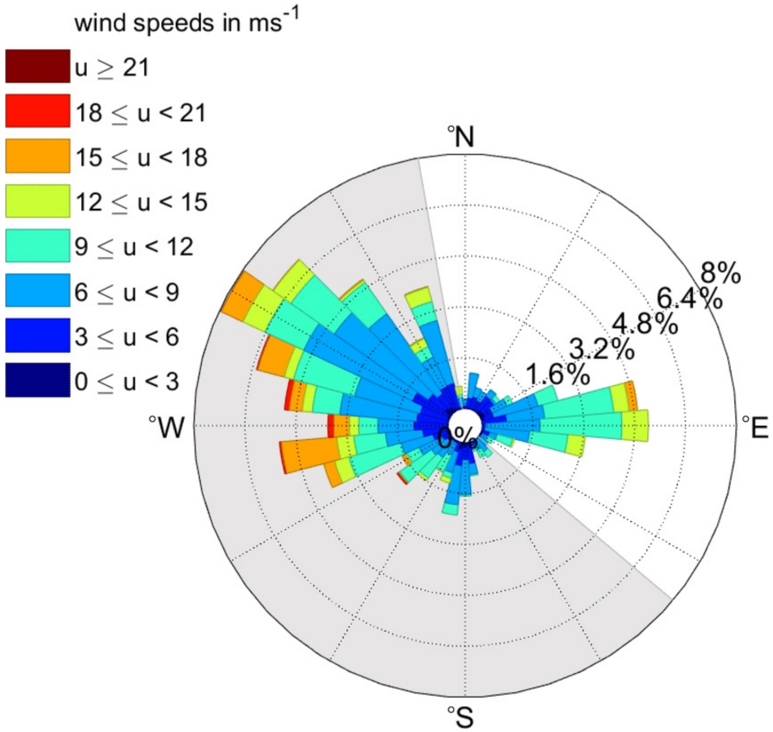




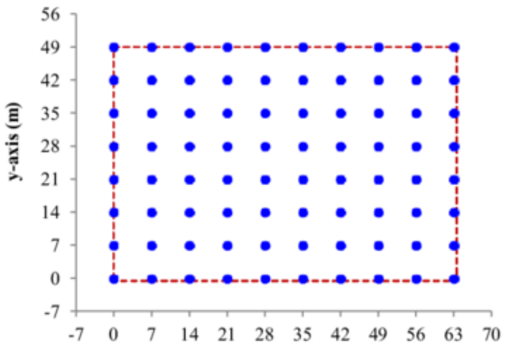


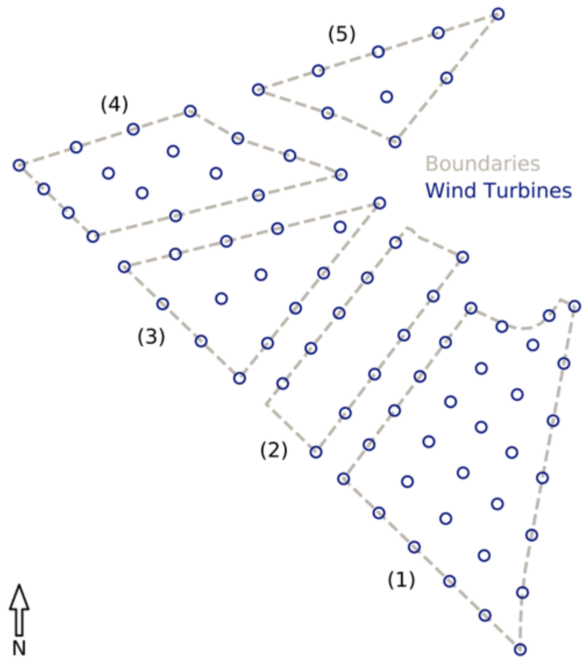


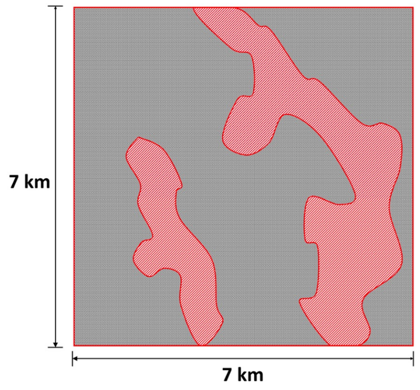
Wind speed and direction



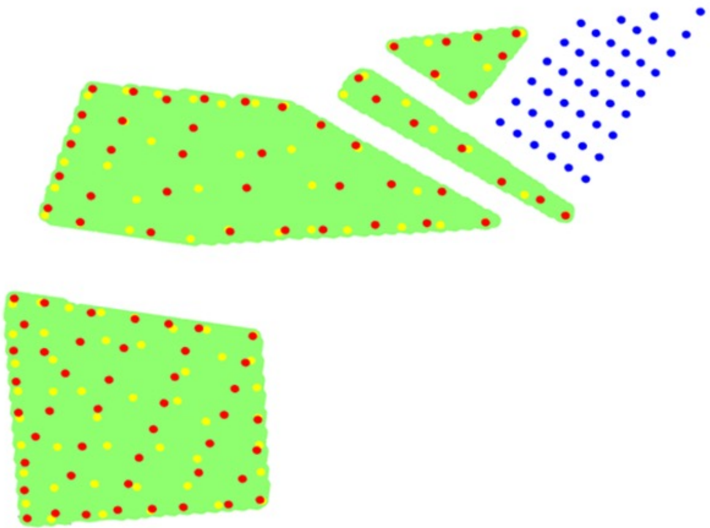
Turbine layout boundaries







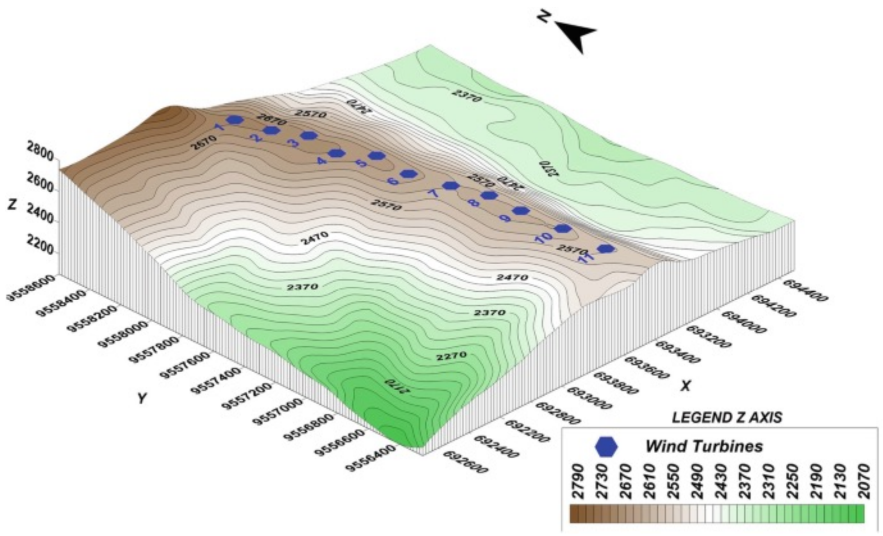
Turbine layout boundaries cont.

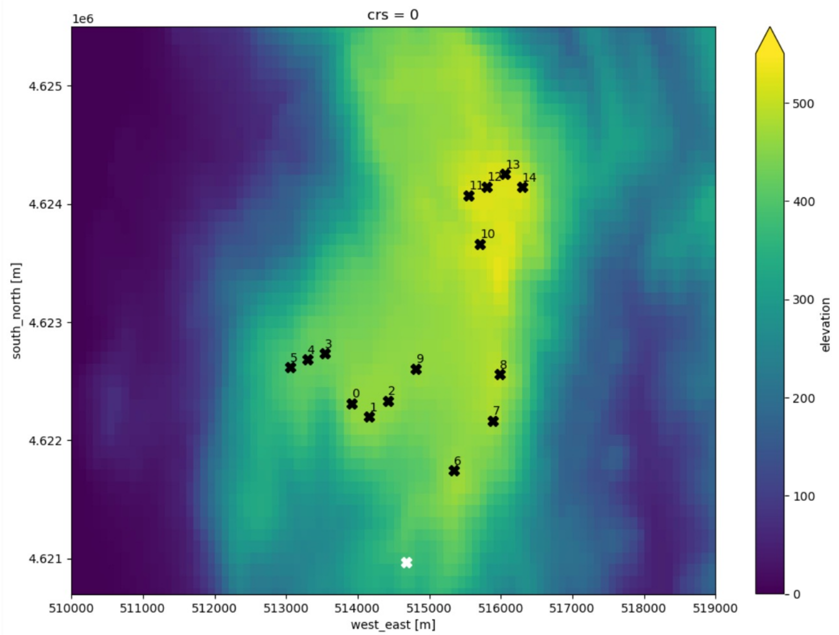


Optimized layout using the optimization tool (red) and the one obtained by traditional methods (yellow). The blue dots are an already existing farm.

The optimized layout allowed for an extra **10.2 million euro gain over the lifetime of the farm by just reducing wind “shadows” among the turbines**.

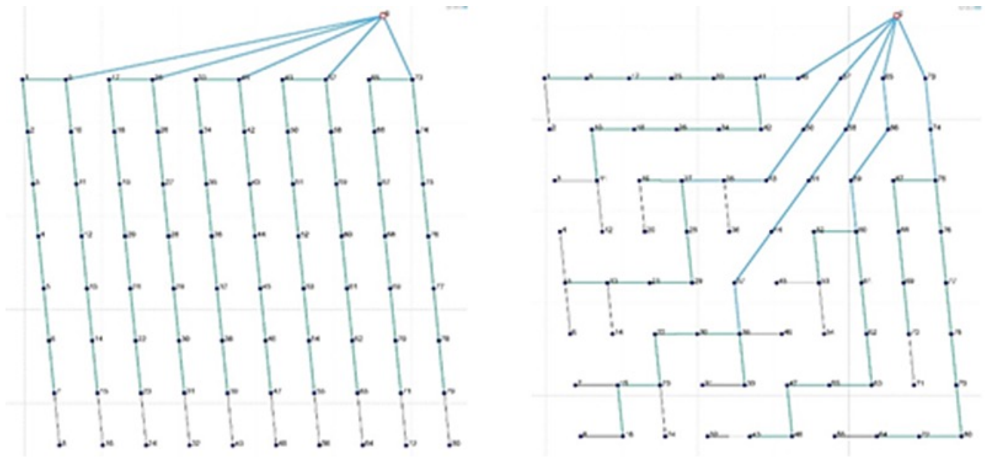
Terrain elevations





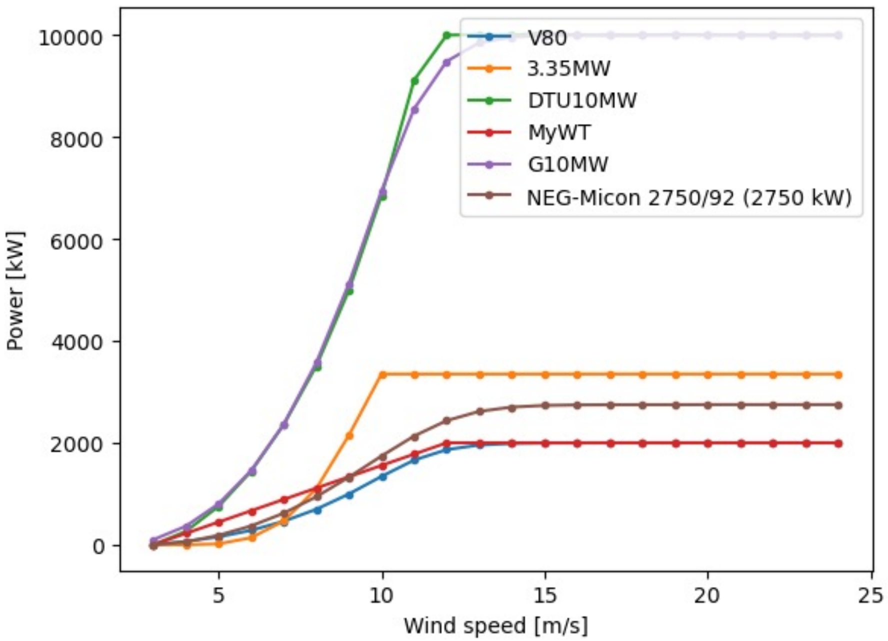
https://docs.wasp.dk/pywasp/

Cable routing

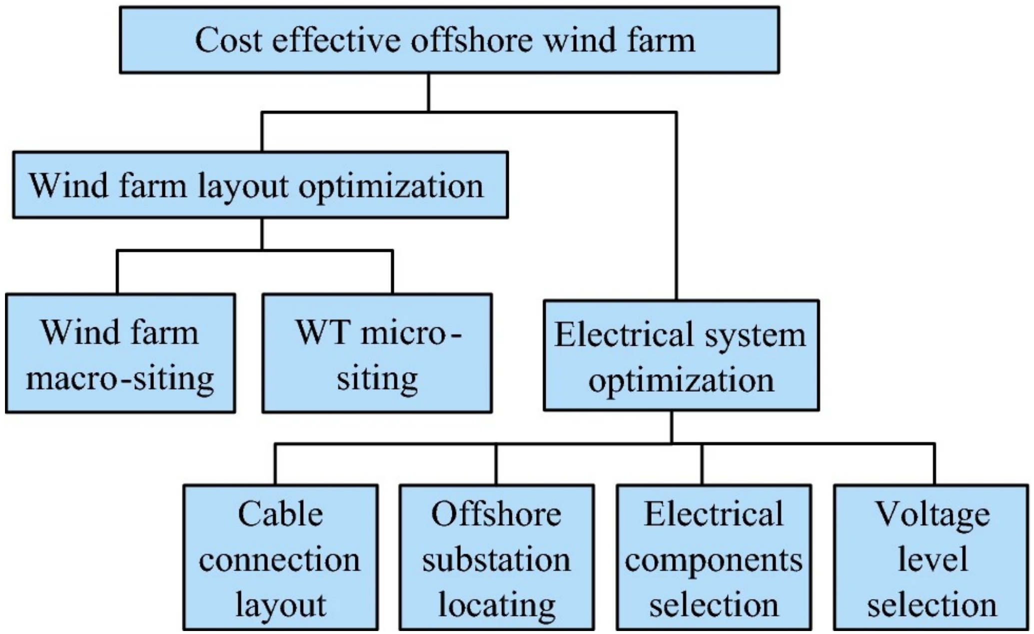


The original, manual layout for the farm, as it looks now (left plot) compared with the layout as it would have looked with the new optimization tool (right plot).
The optimized layout is 1.5 million euros cheaper than the manual solution.

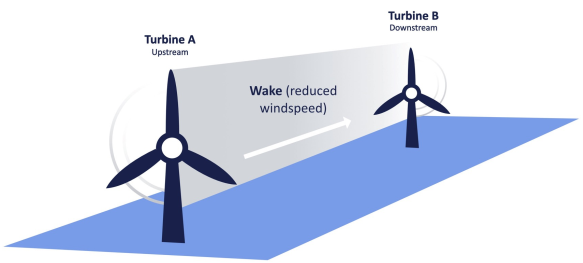
Wind turbine models



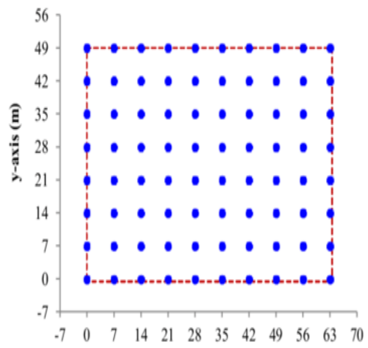
A lot to optimize…



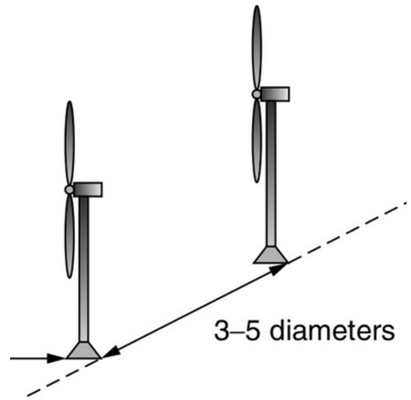
II. Problem Statement



wake penalty



boundary constraint (convex)



spacing constraint ($||x||_2$)

Assumptions

1. All the wind turbines and their power curve functions are identical.
2. The layout of a wind farm is based on a 2D coordinate system, and the search space is $S = [0, L]\times[0, L]$, where L is the upper bound of x and y.
3. There should be set a minimum distance between any two wind turbines to ensure safety and prevent wake effects.
4. A wind turbine turns its nacelle to keep the rotor plane perpendicular to wind direction θ.

Constraints

$$x_i, y_i \in [0, L], \quad \forall i = 1, 2, \ldots, N$$

$$\text{Distance}(i, j) \geq D, \quad \forall i, j = 1, 2, \ldots, N, \; i \neq j$$

where:

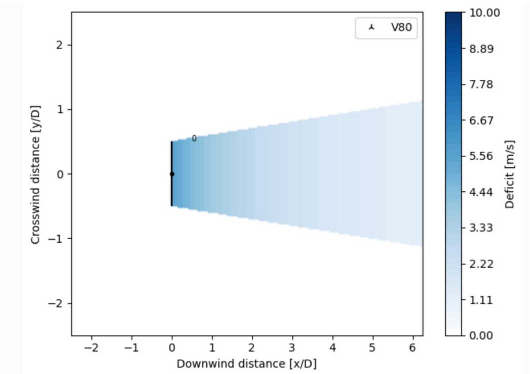
- x_i, y_i are decision variables representing the coordinates of turbine i.
- N is the number of turbines.
- $\text{Distance}(i, j)$ is the Euclidean distance between turbines i and j, and D is the minimum required distance.

Wake Effect Model

wind speed of passing through each wind turbine by considering the wind wake effect, Jensen's model is utilized in this study. Considering that, the momentum is preserved in the wake between two wind turbines in Fig. 1, the momentum equilibrium equation can be expressed as

$$\pi r_{\mathrm{r}}^2 v + \pi\left(r_1^2 - r_{\mathrm{r}}^2\right) u_0 = \pi r_1^2 u_{ij}, \tag{1}$$

where r_{r} is the rotor radius, v is the wind speed behind the front rotor, r_1 is the wake radius, u_0 is the oncoming wind speed, and u is the downstream wind speed at a distance x.



Then, assuming that the wind speed right behind the rotor is 1/3 of u_0 [21], the downstream wind speed passing the turbine i under influence of the upstream wind of the turbine j is written as

$$u_{ij} = u_0\left(1 - \frac{2}{3}\left(\frac{r_{\mathrm{r}}}{r_1}\right)^2\right). \tag{2}$$

To apply the Jensen model, additional assumption about the turbine rotor radius r_{r} and distance x is introduced as

$$r_1 = r_{\mathrm{r}} + \alpha x, \tag{3}$$

where entrainment constraint α is defined by

$$\alpha = \frac{0.5}{\ln\left(z/z_0\right)}. \approx 0.1 \tag{4}$$

In Eq. (4), z is the hub height of the wind turbine and z_0 is the surface roughness of ground. In general, the wind speed through the ith wind turbine for n upstream turbines can be written as

$$u_i = u_0\left(1 - \sqrt{\sum_{j=1}^{n}\left(1 - \frac{u_{ij}}{u_0}\right)^2}\right). \tag{5}$$

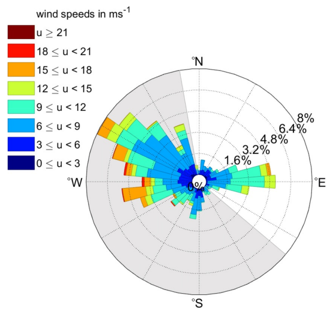
Annual energy production calculation

Power generation amount of a wind turbine is determined by wind speed and wind turbine type. The wind turbine installed in the Daegwallyeong farm and its power curve will be explained in detail in Sect. 3.2. Once the power generation of a wind turbine for given wind speed is determined, the total annual energy of the wind farm can be calculated as

$$\mathrm{AEP}_{\mathrm{total}}(\mathbf{X}) = \sum_{i=1}^{N(\mathbf{X})} \mathrm{AEP}_i(\mathbf{X}) = \sum_{i=1}^{N(\mathbf{X})} \int_0^{360^\circ} \int_0^{u_{\max}} P_i\left(u_i\left(u_0, \theta\right)\right) \times p\left(u_0, \theta\right) \times t \mathrm{d}u_0 \mathrm{d}\theta, \quad (6)$$

where $\mathbf{X}$ is a binary design variable vector in grid area, $\mathrm{AEP}_{\mathrm{total}}(\mathbf{X})$ is the total annual energy of the farm in MWh,

$N(\mathbf{X})$ is the number of the installed wind turbines, $\mathrm{AEP}_i(\mathbf{X})$ is the annual energy for the ith turbine in MWh, u_i is the wind speed for the ith turbine considering the wake effect described in Sect. 2.1, θ is the wind direction, and t is the total hours in 1 year, that is, 24×365. Eq. (6) shows that the multiplication of the power of the ith turbine denoted as $P_i\left(u_i\left(u_0, \theta\right)\right)$ and the probability density function of u_0 and θ denoted as $p\left(u_0, \theta\right)$ is integrated to calculate $\mathrm{AEP}_i(\mathbf{X})$.



Objective Function

Maximize the total energy production of the wind farm, accounting for wake effect.

$$P_{\mathrm{WF}} = \sum_{i=1}^{N} P_{\mathrm{WT}_i}$$

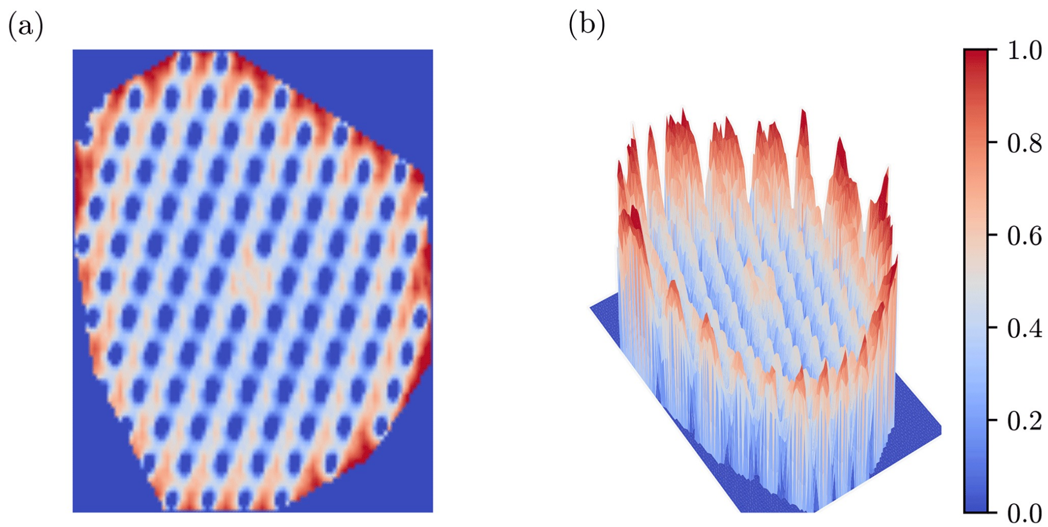
Objective Function

Maximize the total energy production of the wind farm, accounting for wake effect, boundary and spacing constraints.

$$P_{\mathrm{WF}} = \sum_{i=1}^{N} P_{\mathrm{WT}_i}$$

$$\max_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n} \sum_{i=1}^{n} P_{\mathrm{WT}_i} - \sum_{i=1}^{n} \text{boundary\_penalty}(i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{spacing\_penalty}(i, j)$$

The complexity and multimodality of wind farm layout design space



Shown is the normalized annual energy production of a 100-turbine wind farm as a function of the location of one turbine; 99 turbines remain fixed, while one is moved throughout the wind farm. (a) A 2-D view of the design space. (b) A 3-D surface, which highlights the extreme variation of the peaks and valleys. This figure shows only the multimodality from two dimensions, where the true design space has 200 design variables.

Wake loss function

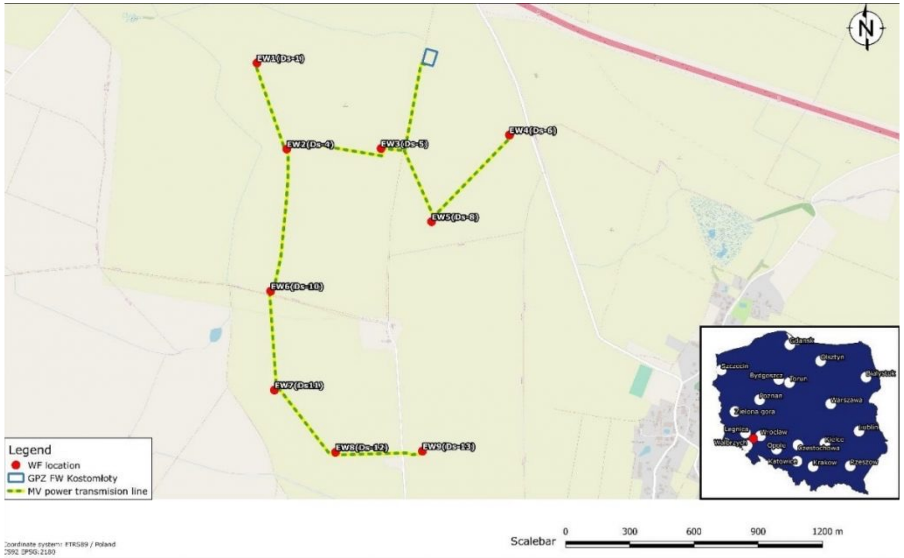
It is a common metric to compare the layouts that indicates how much potential energy conversion was missed due to wake effects.

$$L_w = 1 - \frac{\text{AEP}}{\text{AEP}^*},$$

where AEP* represents the ideal AEP that would exist if all of the wind turbines were exposed to the freestream wind.

Real wind farm example

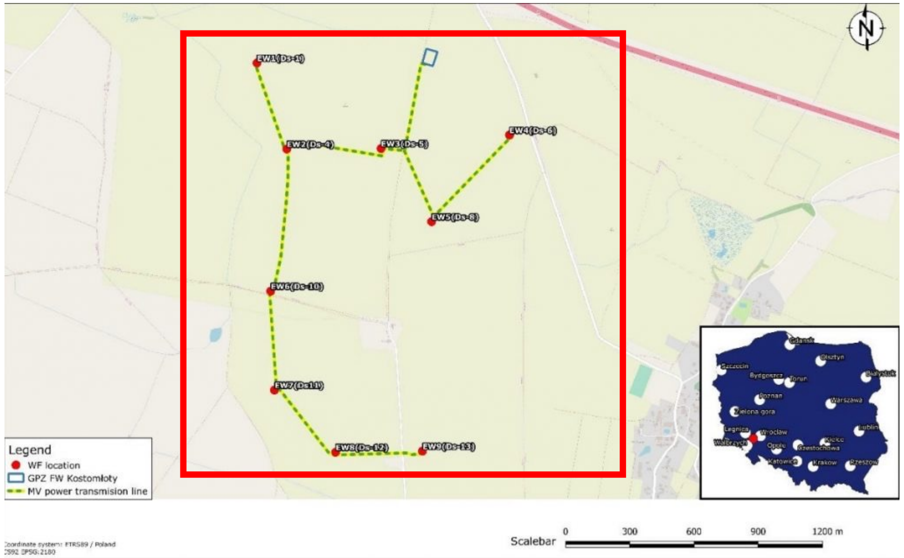
FW Kostomłoty



www.polenergia.pl/nasze-aktywa/wytwarzanie/ladowe-farmy-wiatrowe/fw-kostomloty/

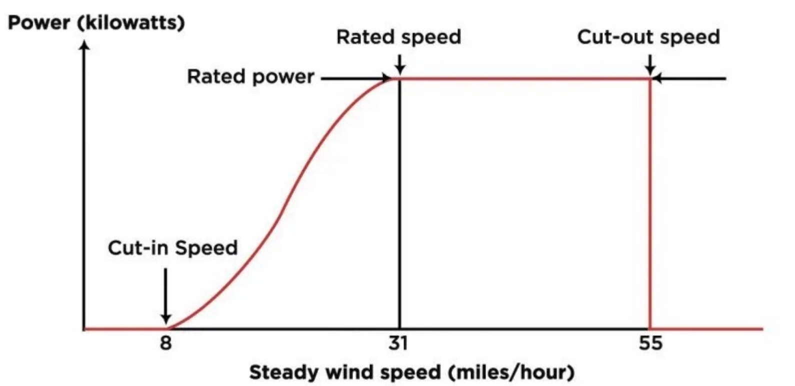
Real wind farm example

FW Kostomłoty



n_turbines = 9
area_size = 2100
min_spacing = 300

Power Curve Model



Typical wind turbine power with steady wind speed

$$E_i = \begin{cases} 0 & \text{if } v < v_{\text{ci}} \text{ or } v \geq v_{\text{co}} \\ P_r \left(\frac{v - v_{\text{ci}}}{v_r - v_{\text{ci}}} \right)^3 & \text{if } v_{\text{ci}} \leq v < v_r \\ P_r & \text{if } v_r \leq v < v_{\text{co}} \end{cases}$$

where:

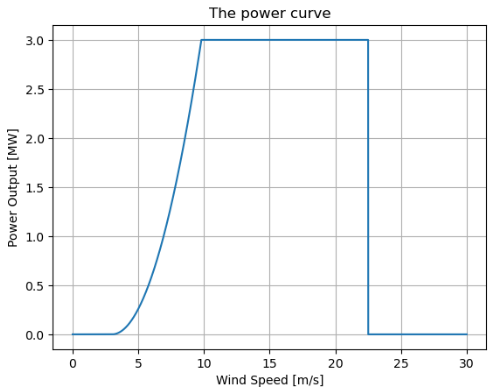
- E_i is the power output of the turbine.
- P_r is the rated turbine power output (in MW).
- v_{ci} is the cut-in wind speed (in m/s).
- v_r is the rated wind speed (in m/s).
- v_{co} is the cut-out wind speed (in m/s).

The **rated speed** is the wind speed at which the turbine reaches its maximum capacity power output.
The goal is to prevent the generator from producing more power than what it are designed to handle.

Power Curve Model

```python
def power_output(wind_speed):
    """Calculate the energy production for a single turbine based on wind speed."""
    P_r = 3.00  # Rated turbine power output [MW]
    v_ci = 3.0  # Cut-in wind speed [m/s]
    v_r = 9.8  # Rated wind speed [m/s]
    v_co = 22.5  # Cut-out wind speed [m/s]

    if wind_speed < v_ci or wind_speed >= v_co:
        return 0
    elif wind_speed < v_r:
        return P_r * ((wind_speed - v_ci) / (v_r - v_ci)) ** 3
    else:
        return P_r
```



Technologia

Farma wiatrowa składać się będzie z turbin firmy Vestas o mocy 3,45 MW każda

Producent turbin: Vestas Wind Systems A/S. (Dania)

Model turbiny: Vestas V136-3.45

Moc nominalna: 3,45 MW

Moc dla trybu zoptymalizowanego: 3,0 MW

Prędkość włączania (startowa): 3.0 m/s

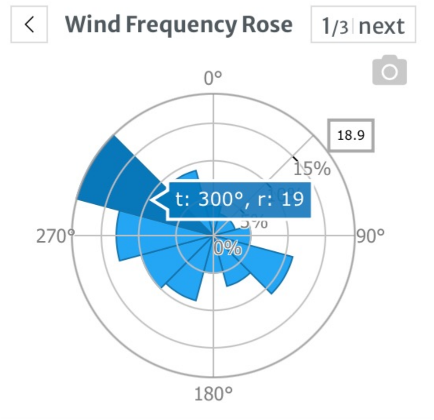
Prędkość wyłączania: 22.5 m/s

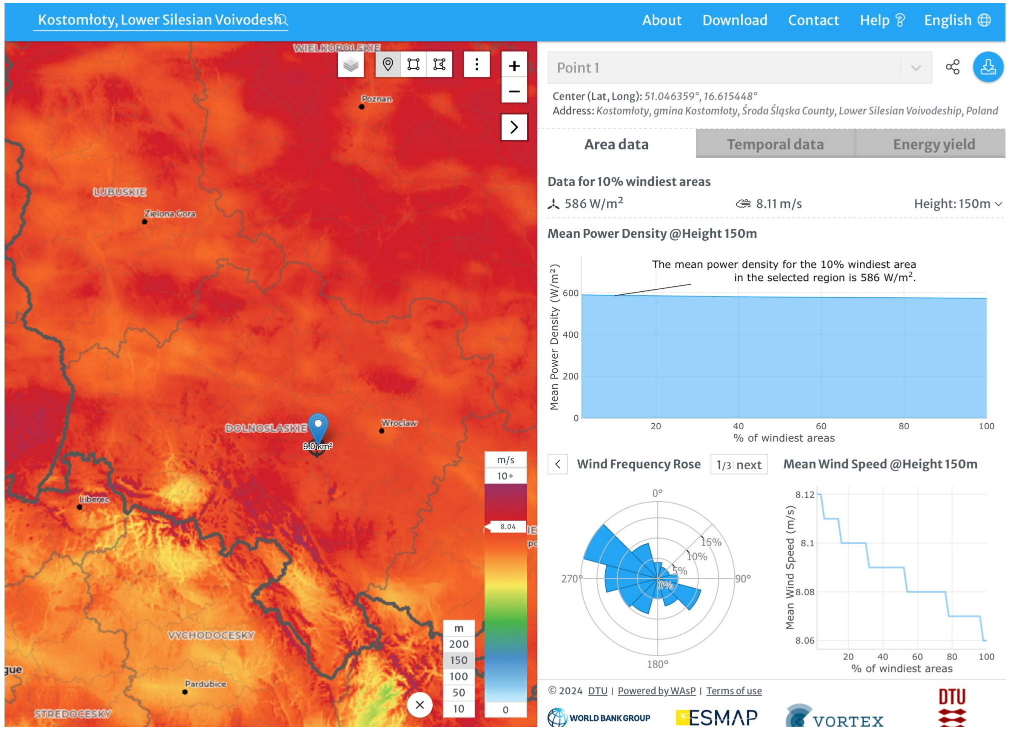
Wysokość piasty: 122 m

Średnica łopat wirnika: 136.0 m

Wind parameters

wind_speed = 9.8
wind_direction = 300.0

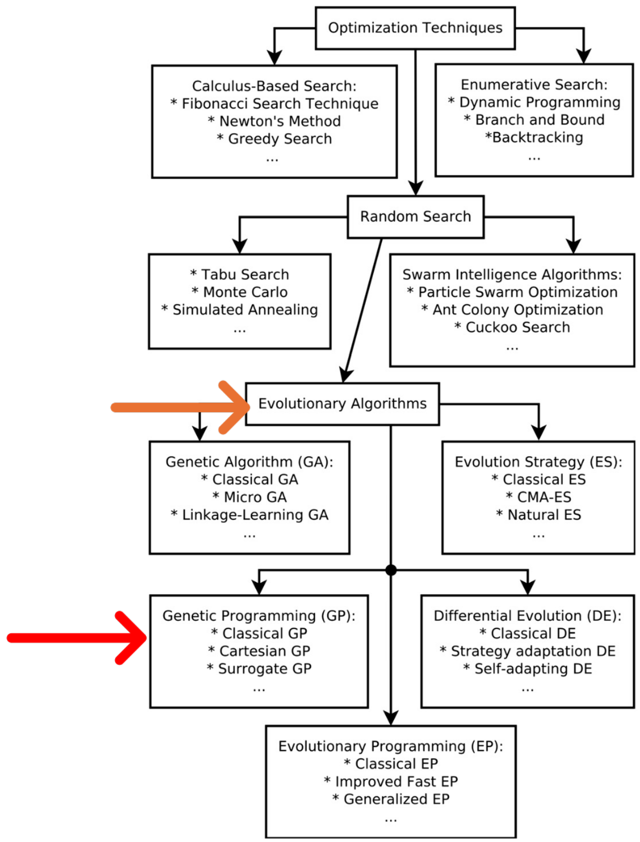




https://globalwindatlas.info/

III. Optimization Strategies

Taxonomy of nature-inspired methods



“In last years it has been shown that non-deterministic methods (especially Genetic Algorithms, and to a lesser extent also others as Particle Swarm Optimization or Ant Colony Optimization) are able to **reach a higher level of evolved optima compared to gradient-based techniques.** ”

GAs vs other methods

GA are different from more normal optimization and search procedures in four ways:

1. GAs work with a **coding** of the parameter set, not the parameters themselves.
2. GAs search from a **population** of points, not a single point.
3. GAs use **objective function values information**, not derivatives or another auxiliary knowledge.
4. GAs use **probabilistic** transition rules, not deterministic rules.

GAs pros

- **No gradients** are required
  (wake model can be considered as a black box)
- It can handle unconnected and **non-convex boundary** constraints
- GAs are adaptive to changes in the objective function during computation, allowing them to handle **dynamic optimization problems**.

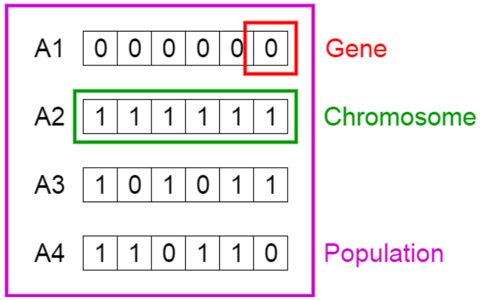
Genetic Algorithms (GA)

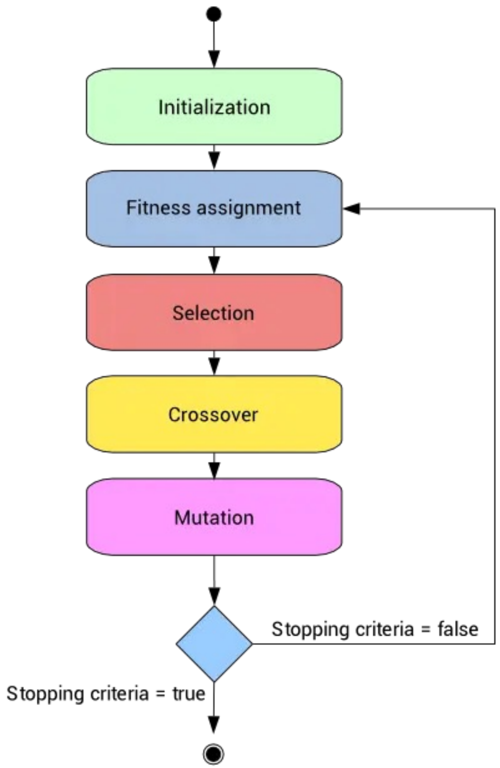
A genetic algorithm is a search heuristic that is inspired by Charles Darwin's **theory of natural evolution**.

This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

Five phases are considered in a genetic algorithm.

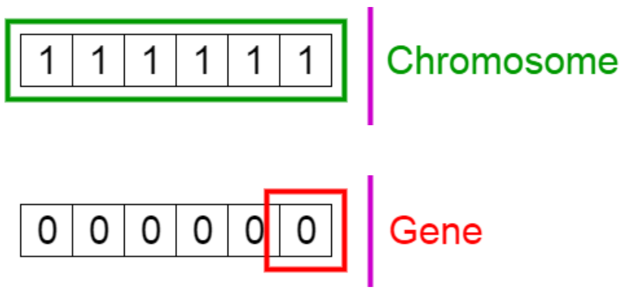
1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

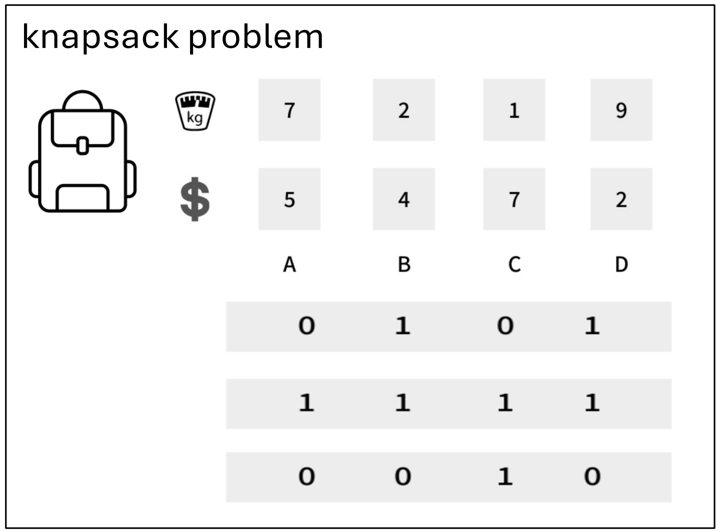




Binary representation

Chromosomes are string of 1s and 0s and each position in the chromosome represents a particular characteristics of the solution.





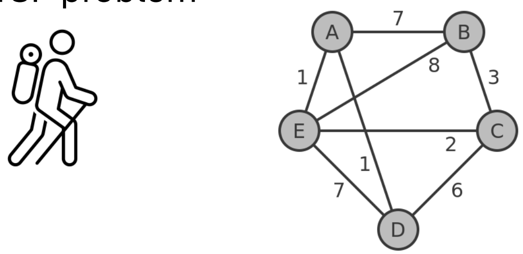
Permutation representation

Useful in ordering such as the Travelling Salesman Problem (TSP). In TSP, every chromosome is a string of numbers, each of which represents a city to be visited.

Chromosome

1	5	8	4	7	6	5	3	1

TSP problem



D	J	F	E	C	B	I	H	G
E	G	H	I	B	F	C	D	J

Value representation

This technique is also for problems with real numbers, where binary encoding isn't enough.
It often needs specialized crossover and mutation methods.

Chromosome

```
[821, 1262, 1456, 1162, 50]

[[1589, 821], [1233, 1262], [750, 1456]]
```

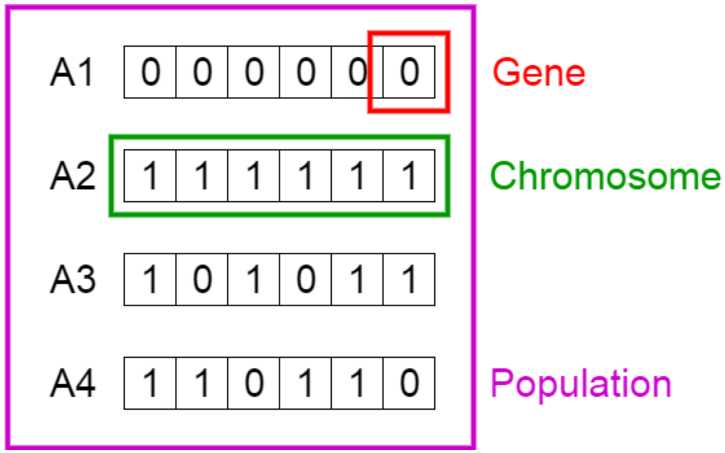
continuous search space problems

$$f(x,y) = x^3 - x^2y + xy^2 + y^3$$

```
[[1589  821]
 [1233 1262]
 [ 750 1456]
 [ 825 1162]
 [ 218   50]
 [ 840  822]
 [1241   61]
 [ 330 1110]
 [ 546  543]]
```

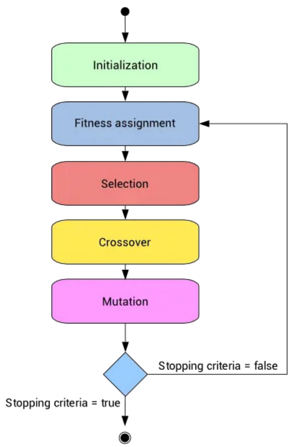
Initial Population

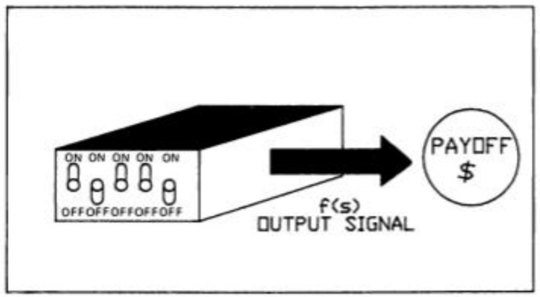
The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve.



Fitness Function

The fitness function determines **how fit an individual is**. It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.
Without fitness-based selection mechanisms for mate selection and offspring acceptance, EA search would be blind and hardly distinguishable from the Monte Carlo method.





Penalty Methods

The penalty methods are the most common approaches for constraint-handling in GA.

$$\psi(\vec{x}) = f(\vec{x}) \pm \left[\sum_{i=1}^{n} a_i \times G_i + \sum_{j=1}^{m} b_j \times H_j \right] \qquad (1)$$

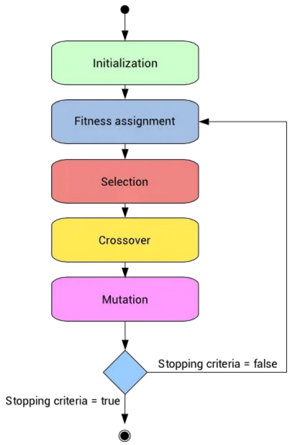
where, $\psi(\vec{x})$ is the new fitness function to be optimized, G_i and H_j depend on the inequality constraints $g_i(\vec{x})$ and equality constraints $h_j(\vec{x})$ respectively and a_i, b_j are called penalty factors.

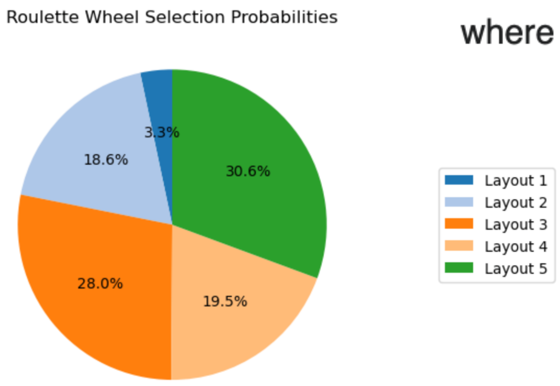
Selection

The idea of **selection** phase is to select the fittest individuals and let them pass their genes to the next generation.

Probability of choosing individual i is equal to $p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}$,

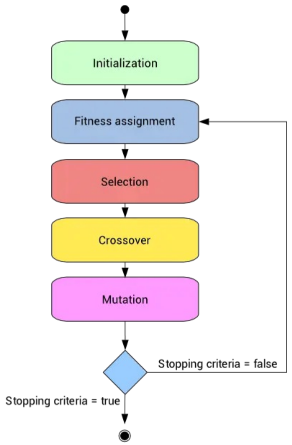
where f_i is the fitness of i and N is the size of current generation

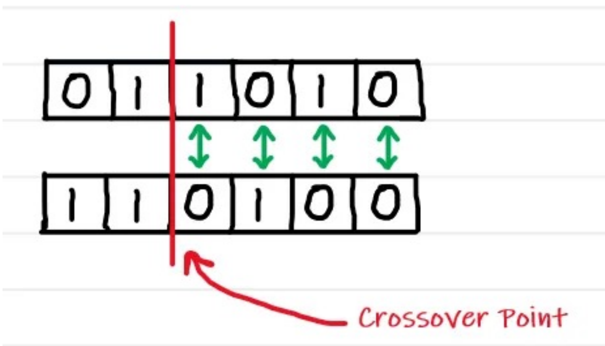


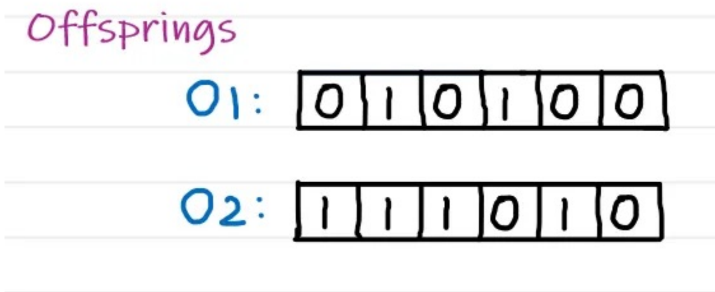


Crossover

For each pair of parents to be mated, a crossover point is chosen at random from within the genes. Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached. The new offspring are added to the population.







Crossover cont.

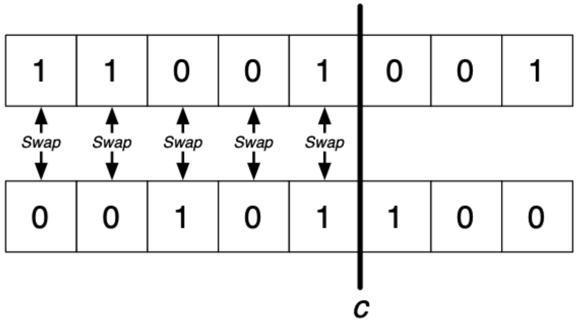


Figure 9 One-Point Crossover.

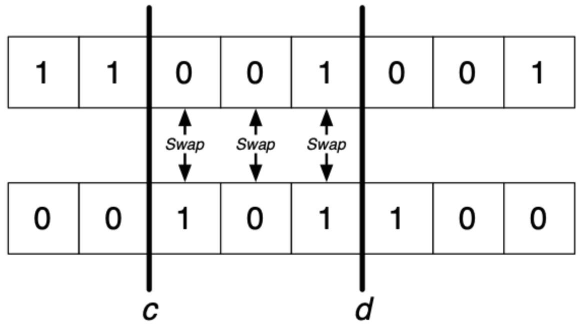


Figure 10 Two-Point Crossover.

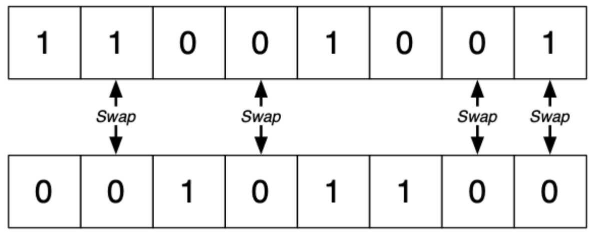
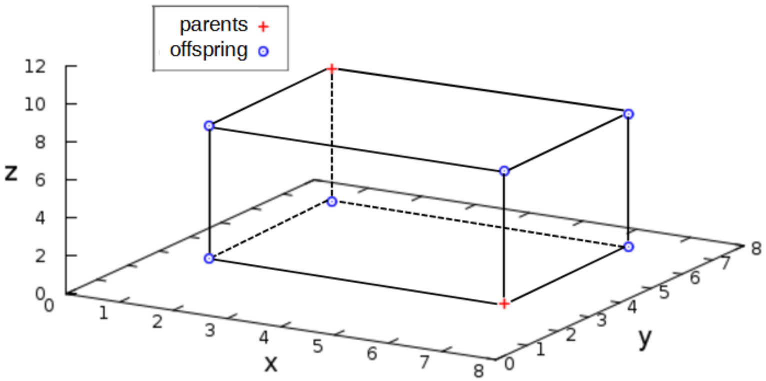
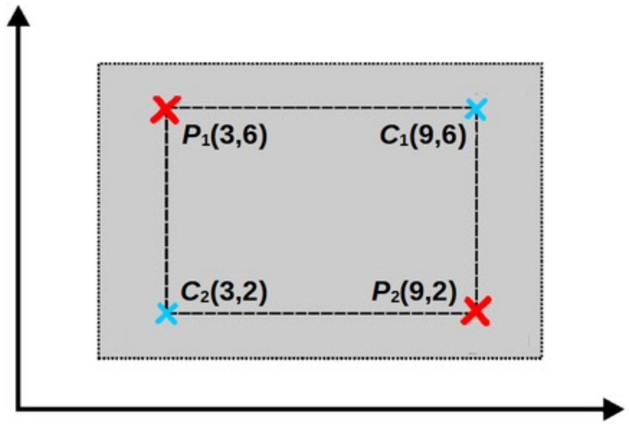


Figure 11 Uniform Crossover.

Crossover cont.

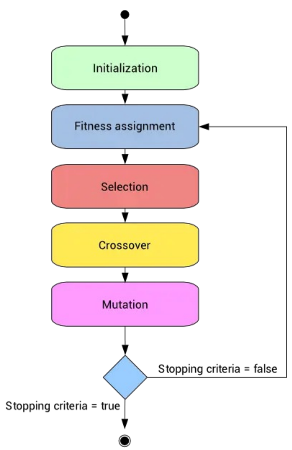


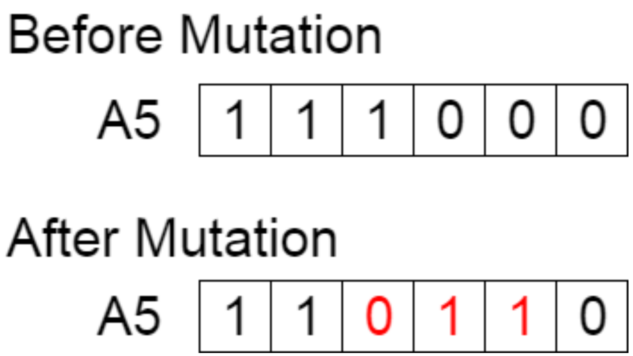


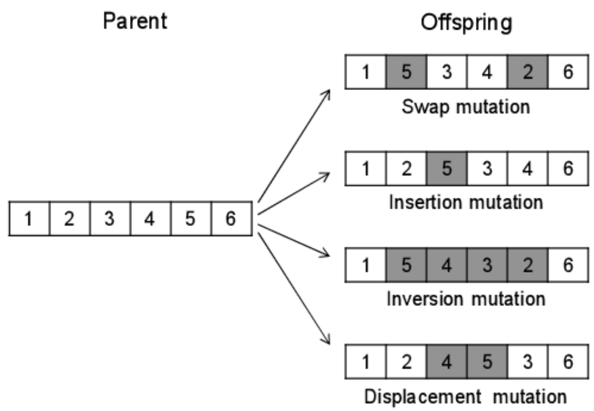
$\alpha_i = \alpha_{i,P_1} \cdot \beta_i + \alpha_{i,P_2} \cdot (1-\beta_i)$ with $\beta_i \in [-d, 1+d]$ randomly equally distributed per gene i

Mutation

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped.



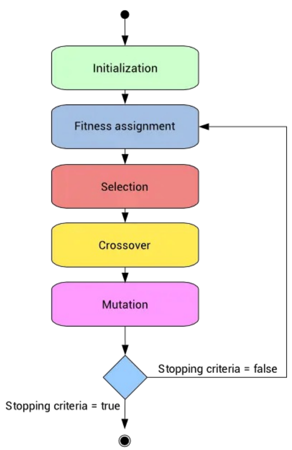




Termination

- Stopping Evolution after n Generations
- Stopping Evolution with No Improvements
- Temperature/momentum:

temperature = (1 — cooling_rate) * (temperature + (best — last_max_fitness))



Schema

A **schema** is a template that identifies a subset of strings with similarities at certain string positions.

1*101*10

Order of a schema refers to the number of fixed positions in the schema.

Schema Theorem

Holland's Schema Theorem states that short, low-order schemas with above-average fitness will appear with exponentially increasing frequency in subsequent generations of a genetic algorithm.

The reason that the Schema Theorem cannot explain the power of GAs is that it holds for all problem instances and cannot distinguish between problems in which GAs perform poorly, and problems for which genetic algorithms perform well.

Principles for designing GA

- **Understanding the problem** allows us to create more efficient search procedures compared to blind random sampling
- There is no single best way to solve a problem. Instead, there may be many effective methods and even more highly efficient solutions. The goal is to find or design the right tool that fits the specific problem.
- Evolutionary algorithms are highly **adaptable**. By adjusting various aspects such as representation, variation operators, population size, selection mechanisms, initialization, and evaluation functions, we can tailor the search procedures to address a wide range of tasks, much like a Swiss Army knife.

Challenges in GAs

- premature convergence
- optimal trade-off between exploration and exploitation
- computational time (requires fast fitness function)
- proper definition of an objective function
- repeatability of the EA methods

Genetic algorithms in real-life problems

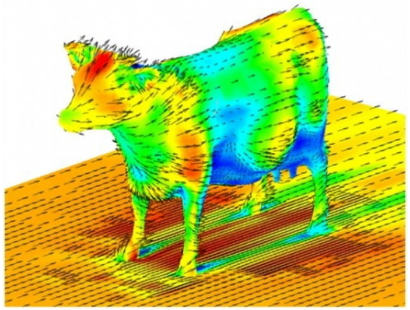
The GAs are a universal optimization tool. Using GAs, we can solve **constrained optimization problems, multimodal optimization problems, continuous optimization problems, combinatorial optimization problems, and multi-objective optimization problems**. Thus, there is a wide range of real-world applications of GAs.

IV. Implementation

PyWake

open-sourced and Python-based wind farm simulation tool

Bibliography



- Evolutionary algorithms and their applications to engineering problems - Adam Slowik, Halina Kwasnicka (2020)
- Algorytmy genetyczne i ich zastosowania – David E. Goldberg
- Wykłady z algorytmów ewolucyjnych – Jarosław Arabas
- Jak to rozwiązać, czyli nowoczesna heurystyka - Zbigniew Michalewicz, David B. Fogel
- Realistic wind farm design layout optimization with different wind turbines types - Naima, Charhouni & Sallaou, Mohammed & Mansouri, Khalifa (2019)
- www.polenergia.pl/nasze-aktywa/wytwarzanie/ladowe-farmy-wiatrowe/fw-kostomloty/
- www.energy-fundamentals.org/15.htm

- Offshore wind farm layout optimization: What is the hype? - Poursaeidi, Mohammad & Samorani, Michele & Kundakcioglu, O. (2013)
- Analysis of the Annual Energy Production of the Villonaco Wind Farm - Maldonado-Correa, J. et al. (2023)
- group.vattenfall.com/press-and-media/newsroom/2019/mathematics-generate-million-euro-savings-on-future-offshore-wind-farms
- www.synapse-energy.com/offshore-wind-wake-effects-are-real-we-should-plan-them
- news.mit.edu/2022/wind-farm-optimization-energy-flow-0811
- https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection/
- https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
- https://medium.com/du-phan/wind-energy-for-dummy-bc9e181c8bc8