

VelantisWind: an evolutionary multi-population framework for wind farm layout optimization

Physical formulation of wind resource and wakes, public search architecture, and GIS-native engineering workflow

VelantisWind Technical Team • Public Technical White Paper • September 2026

Purpose and disclosure boundary. This public paper explains the physical problem, the engineering interfaces and the high-level behavior of the VelantisWind optimizer at a level intended to be technically credible and auditable. It deliberately does not disclose the operator catalogue, geometric descriptor definitions, adaptive selection logic, search-memory representation, migration rules, calibration data, schedules, thresholds, budgets or repair heuristics that constitute the proprietary implementation. The equations describe the engineering problem and established physical models; they are not a specification of the private search engine.

Abstract

Wind farm micrositing is a constrained and wake-coupled optimization problem: each turbine must satisfy spatial restrictions while its energy contribution depends on the positions of the other turbines. VelantisWind addresses this problem with a population-based evolutionary framework embedded directly in a GIS/QGIS workflow. The search is not reduced to a single mutation loop. Instead, complementary search populations are assigned different roles: intensifying competitive layouts, preserving alternative feasible layout families, and performing broader structural exploration. A coordinating layer shares selected candidate material and adapts the balance of the search as the run evolves. Candidate layouts are scored through an external wake-based physical evaluator, while the final result remains linked to the same georeferenced boundaries, exclusions, wind-resource data and turbine definitions used to define the project. This document describes those principles and the public system boundary without exposing the internal policies that determine exactly how candidate transformations are generated, selected, learned from or exchanged.

Keywords: wind energy • micrositing • layout optimization • wake effects • evolutionary optimization • multi-population search • GIS / QGIS

1 Introduction

The energy delivered by a wind farm depends strongly on the relative position of its turbines. A turbine operating in the wake of another receives a lower effective wind speed and a modified turbulence field, so two layouts built with the same turbine model and inside the same project boundary can produce materially different annual energy production (AEP). Because even modest percentage differences accumulate over the operating life of a project, micrositing is one of the design stages in which geometry has a direct and persistent economic effect.

The optimization problem is difficult because three characteristics coincide. The admissible terrain is fragmented by exclusions, setbacks and infrastructure; the objective is non-separable because moving one turbine can change the inflow at several others; and the landscape is multimodal, with distinct geometric families capable of becoming locally competitive. A practical optimizer must therefore search broadly enough to avoid premature convergence while still concentrating expensive wake evaluations around promising material.

VelantisWind is built around that engineering tension. Its public contribution is not a claim that a particular textbook metaheuristic is universally superior. The design instead separates search roles, physical evaluation and GIS constraints so each can evolve independently while the final result remains traceable.

This distinction is important in engineering optimization. Two implementations may both be described as evolutionary while behaving very differently once constraints, wake-evaluation cost, population collapse and late-stage convergence are considered. The public formulation therefore focuses on the observable behavior the architecture is intended to produce, while leaving the mechanisms that create that behavior inside the proprietary engine.

2 Physical and optimization formulation

2.1 Wind resource and annual energy production

The wind resource may be represented by directional sectors and a wind-speed distribution inside each sector. A common representation is the two-parameter Weibull distribution, written for sector θ as:

$$f(v|\theta) = (k/A)(v/A)^{k-1} \exp[-(v/A)^k] \quad (1)$$

where A and k are the scale and shape parameters for the sector under consideration. The optimizer itself does not depend on the Weibull representation: the relevant public interface is a probability-weighted description of the wind conditions used by the physical evaluator.

For a layout represented by turbine coordinates X , annual energy is the farm power integrated over the modeled wind conditions and weighted by their occurrence:

$$AEP(X) = T \sum_{\theta} \sum_{\nu} f(\theta, \nu) P_{farm}(X, \theta, \nu) \quad (2)$$

Here T is the number of hours in the year and P_{farm} is the wake-affected farm power. The important feature for optimization is coupling: P_{farm} cannot in general be decomposed into independent turbine terms because the inflow at one machine depends on the positions and operating states of others.

2.2 Engineering metrics

Two useful derived quantities are capacity factor and aggregate wake loss. Capacity factor relates AEP to the theoretical rated maximum:

$$CF = \frac{AEP(X)}{N \cdot P_{rated} \cdot T} \quad (3)$$

Aggregate wake loss compares the wake-affected farm with a no-wake reference calculated under the same resource and turbine assumptions:

$$L_{wake} = 1 - \frac{AEP(X)}{AEP_{free}(X)} \quad (4)$$

These quantities are useful because they separate two questions that should not be confused: how strong the site and turbine combination is in absolute terms, and how much production is being lost through turbine-to-turbine interaction.

2.3 Wake coupling

Candidate scoring can use established analytical wake models. In the classical linear-expansion family, a representative centerline velocity-deficit expression is:

$$\delta u = (1 - \sqrt{(1 - C_T)}) \cdot (D / (D + 2k_w x))^2 \quad (5)$$

where C_T is thrust coefficient, D is rotor diameter, k_w is a wake-expansion parameter and x is downstream distance. Gaussian and other modern formulations replace the top-hat profile with smoother self-similar deficits and may couple wake growth to turbulence. VelantisWind treats the selected wake formulation as a physical evaluation layer rather than as part of the proprietary search logic; this keeps optimization and aerodynamic-model choice modular.

2.4 Constrained micrositing problem

For a fixed number N of turbines, the public engineering objective can be expressed as:

$$X_{opt} = \arg \max_X AEP(X), \text{ with } X \in F(\Omega, S, N) \quad (6)$$

$F(\Omega, S, N)$ denotes the set of layouts with N turbines that satisfy the admissible project domain Ω , the applicable turbine-separation rule S and any other active hard project constraints. Equation (6) defines what is optimized; it intentionally does not define how VelantisWind explores that feasible set.

3 Design philosophy and public search architecture

A single search population must continuously trade exploration against exploitation. If it becomes too conservative, it can collapse around an early local optimum. If it remains too exploratory, it can consume a large wake-evaluation budget without adequately refining strong layouts. VelantisWind addresses this by dividing the search into complementary roles rather than forcing one mechanism to perform every function.

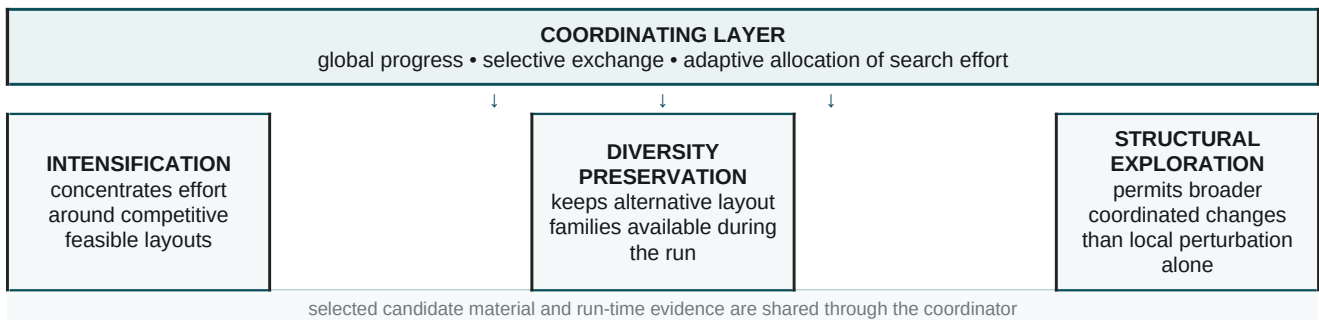


Figure 1: Public conceptual architecture. The diagram states functional search roles, not the underlying algorithms, operator families, exchange topology or adaptive decision rules.

3.1 Specialized search roles

The intensification role operates close to competitive material and is responsible for converting promising discoveries into stronger layouts. The diversity-preservation role prevents the run from becoming represented by a single geometric family too early. The structural-exploration role provides a route to different basins by allowing

coordinated changes at a broader spatial scale. The public point is the division of labor: the exact representation of each population and the mechanisms used inside it are proprietary.

This distinction matters because a layout problem may require different behavior at different stages of a run. Early improvements may be obtained from relatively broad exploration, whereas late improvements can require highly selective use of the remaining evaluation budget. VelantisWind can alter the relative emphasis placed on its search roles as evidence accumulates, rather than using a fixed search mix from start to finish.

The roles should be understood as search behaviors rather than as a public specification of particular algorithms. Intensification means that some computational effort remains close to material that has already proved competitive. Diversity preservation means that the run avoids representing the entire feasible design space by a single geometric family too early. Structural exploration means that the engine retains the ability to make coordinated changes at a larger spatial scale than an isolated turbine perturbation. The internal representation of those behaviors, and the rules that decide when each one receives effort, are not disclosed.

3.2 Online adaptation without disclosing the policy

The engine uses run-time diagnostics to adapt the broad allocation of search effort as the optimization progresses. This should not be interpreted as a published adaptation algorithm: the internal state variables, update rules, decision policy and parameterization all remain inside the proprietary boundary. Publicly, the relevant claim is narrower: the search mix is not assumed to be equally appropriate throughout every site and every phase of a run.

That statefulness is one of the intended differences from a fixed recipe. The search is allowed to react to what has happened during the current run, but this paper does not identify the diagnostic variables, their numerical definitions, the memory retained from earlier decisions, or the mapping from diagnostics to search actions. Those details are precisely where a high-level concept becomes a calibrated optimization system.

3.3 Feasibility and expensive evaluation

Wind-farm optimization is unusual in that an apparently simple geometric move can trigger an expensive full-farm energy evaluation. VelantisWind therefore separates cheap spatial validity checks from physical scoring. Candidate layouts are required to respect the active project domain and turbine-separation rules before they are admitted to the expensive wake-evaluation stage. Internal repair, screening, caching and pre-ranking procedures are deliberately not described because those details strongly affect evaluation efficiency in practice.

3.4 Public lifecycle of a candidate solution

At interface level, the life of a candidate layout can be described without revealing the private search policy. A population proposes a new candidate using its internal mechanism; the candidate is brought back into compliance with the active spatial constraints if necessary; redundant or unsuitable proposals can be screened before physical scoring; the surviving candidate is evaluated by the wake/AEP engine; and the result is returned to the search state so subsequent decisions can make use of the new evidence. The process repeats until the configured computational budget or stopping condition is reached.

What is deliberately absent from that lifecycle is the information required to implement it competitively: which population proposes the candidate, how the target layout or turbines are chosen, what geometric transformation is applied, how run-time evidence changes subsequent search decisions, which candidates are exchanged between populations, and how much evaluation budget each mechanism receives. Those choices are the calibrated algorithm rather than the public workflow.

3.5 Search behavior over the course of a run

A useful way to understand the public architecture is through the behavior expected over time. At the beginning of a run, the engine must establish enough geometric breadth to avoid committing to the first promising basin. During the middle of the run, information from competitive and alternative layouts should increasingly influence where evaluations are spent. Near convergence, improvement typically becomes sparse and expensive, so the system must distinguish between productive refinement and simply repeating variations of already explored material. VelantisWind is designed around this changing search regime rather than assuming that one fixed level of exploration is appropriate from start to finish.

This description deliberately stops at the behavioral level. It does not state how convergence is detected, how candidate families are compared, how historical evidence is weighted, or whether a particular internal mechanism is active at any given moment. Those decisions are implementation details and are intentionally absent from the public document.

4 Public algorithmic behavior and proprietary boundary

The preceding section is intentionally more informative than describing the optimizer only as a generic metaheuristic or genetic algorithm, but less informative than an implementation blueprint. An independent engineer can understand why the method is designed to resist premature convergence, why it can perform both local and broader search, and why the balance of search effort can change during a run. They cannot infer from this document how candidates are encoded, which operators exist, which layouts are selected as parents or targets, how diversity is quantified, how cross-population exchange is scored, or how the internal learning state is updated.

4.1 Publicly described behavior

- The optimizer is population-based and evolutionary rather than a deterministic single-start local solver.
- Several concurrent search roles are maintained so exploitation, diversity preservation and broader exploration do not have to be performed by one mechanism.
- The amount of effort assigned to those roles can change as the run progresses and evidence accumulates.
- Candidate solutions are kept consistent with the GIS-defined feasible problem before expensive physical scoring.
- The final objective remains the same engineering quantity throughout the run: wake-coupled AEP under the stated physical model and constraints.

4.2 Proprietary implementation boundary

- Exact operator catalogue, perturbation geometries and multi-turbine transformations.
- The internal layout representation, diversity state and spatial search state.
- The rules that select which candidate, region or transformation to act on at a given moment.
- Adaptive decision rules, internal search memory and parameter-update policy.
- Cross-population topology, exchange cadence, quotas, scoring functions and duplicate-handling policy.
- Activation thresholds, budgets, stopping rules and search schedules.
- Calibration datasets, empirically tuned constants, search-memory initialization and low-cost screening heuristics.

These omissions are not cosmetic. In a population-based optimizer, competitive behavior is often determined less by the names of the component techniques than by how candidate generation, selection, memory, constraint handling and evaluation budget are coupled. That coupling is the portion retained inside the implementation.

4.3 Why the architecture is more than a conventional genetic loop

A conventional population-based optimizer can be built from repeated selection, crossover and mutation. VelantisWind uses a broader architecture: distinct populations are allowed to preserve different search behaviors, broader structural changes coexist with fine exploitation, and accumulated run-time evidence can alter how search effort is allocated. The purpose of stating this publicly is to make the methodological claim falsifiable in validation - the optimizer should resist early collapse and continue finding useful material late in a run - without exposing the mechanisms needed to recreate that behavior.

4.4 Observable engineering properties

The architecture can be evaluated without exposing its private machinery. A healthy run should maintain more than one competitive geometric family long enough for useful alternatives to survive; improvements should continue after the easy early gains have been captured; infeasible proposals should not dominate the physical-evaluation budget; and repeated runs under the same benchmark should produce a defensible distribution of outcomes rather than relying on a single exceptional trajectory. These are observable properties that can be tested from run histories and final layouts.

Conversely, a public paper does not need to publish the exact mechanism responsible for each property. The same validation question can be asked from outside the engine: did the run collapse prematurely, did objective improvement stall unnecessarily, were constraints respected, and can the final AEP be independently recomputed? This separation allows technical scrutiny without turning a validation document into a recipe.

5 Traceability and computational discipline

5.1 Engineering traceability

A layout optimizer intended for engineering use should provide more than a final coordinate list. VelantisWind records enough information to connect the optimized layout to the data and physical assumptions under which it was produced: project coordinate reference system, active constraints, turbine definition, wind-resource representation, wake-model configuration, objective history and final physical metrics. This allows the result to be reviewed even when the private search state is not exposed.

Traceability also supports development and validation. Changes in final AEP can be separated from changes in physical assumptions; stochastic runs can be compared using a common benchmark protocol; and regressions can be detected by observing objective evolution, feasibility and computational effort. Internal operator-by-operator telemetry may exist for development purposes, but it is not required to be part of a public or client-facing report.

5.2 Protecting the physical-evaluation budget

Wake-based AEP evaluation is the dominant computational cost in many layout studies. The practical objective is therefore not simply to create many candidate layouts, but to allocate physical evaluations to candidates that are both feasible and informative. VelantisWind is designed around this principle. Cheap geometric work can be performed before the wake engine is invoked, and independent evaluations can be dispatched in parallel where the underlying physical implementation permits it. Concrete caches, screening metrics, batch sizes and budget-allocation rules remain proprietary.

6 Integration with GIS / QGIS workflows

VelantisWind is conceived as a GIS-native engineering workflow rather than a standalone optimizer that receives an opaque coordinate file. Project boundaries, exclusion zones, turbine locations, resource layers and optimized results share a common spatial reference system and can be inspected together. This is particularly important during early design, when setbacks, environmental constraints, candidate areas or turbine assumptions can change repeatedly.

The GIS integration creates a direct chain from project assumptions to the optimization problem. A modified exclusion polygon changes the feasible set; an updated resource layer changes the physical scoring; and an altered turbine definition changes both power production and wake interaction. Because the optimized coordinates are returned to the same spatial environment, the engineer can compare the resulting layouts without rebuilding a separate solver model.

6.1 Public engineering workflow

Stage	Engineering object	Auditable public information
1. Site	Boundary, exclusions, spacing and project CRS	Active GIS layers, CRS and constraint version
2. Resource	Directional / spatial wind description	Resource source, resolution and preprocessing assumptions
3. Turbine	Power and thrust behavior plus geometry	Turbine definition and density assumptions used by the physical model
4. Physics	Wake, superposition and turbulence configuration	Physical model configuration used to score layouts
5. Search	Proprietary multi-population optimization	Objective history, evaluation count and run configuration exposed at interface level
6. Delivery	Georeferenced optimized layout	Coordinates, final AEP / wake loss and the assumptions required to reproduce final scoring

Table 1: Public engineering workflow. The search stage is auditable at input/output and computational-reporting level while its internal decision logic remains proprietary.

6.2 Reproducibility at the engineering interface

For external review, reproducibility means that the engineering problem and the final result can be reconstructed: the same project geometry, turbine definition, wind-resource representation and wake-model assumptions should reproduce the reported physical score of a delivered layout. It does not require disclosure of the sequence of private search decisions that produced that layout. This distinction is useful for pilots and third-party validation because it allows the result to be challenged independently while protecting the optimizer itself.

6.3 Modular physical evaluation

The search engine and the physical evaluator are deliberately separated. In practice, this means that the optimizer can be benchmarked under a stated wake formulation without making that wake formulation part of the search algorithm itself. The public white paper therefore documents the physical configuration used for a reported result, while leaving the private search policy unchanged. This separation is useful for engineering comparison because a reviewer can recompute the final layout under the same physical model without needing access to the optimizer internals.

It also prevents a common source of ambiguity in optimization claims. A gain can come from a better search, from a different wake formulation, from a change in turbine data or from altered constraints. By keeping those interfaces explicit, VelantisWind can attribute changes to the relevant layer instead of mixing algorithmic and physical effects in one opaque score.

7 Validation philosophy

A single attractive layout is not sufficient evidence for an engineering optimizer. Validation must address objective quality, stochastic repeatability, constraint handling, physical coherence and computational effort. VelantisWind therefore treats validation as a program rather than a one-off demonstration. Public evidence can grow through additional benchmarks and multi-seed statistics without expanding disclosure of the proprietary search implementation.

7.1 IEA37 Case Study 4 reference run

The original expanded technical paper reported a reference run on the public IEA37 Case Study 4 benchmark with 81 turbines of 10 MW and the official wind rose. The comparison used a Bastankhah Gaussian wake model with $k_y = 0.0324555$ and ambient turbulence intensity $TI = 0.075$. These physical settings are stated so that the final layout score can be interpreted consistently with the published comparison. The optimization configuration itself is not disclosed.

Metric	VelantisWind (single run)	Thomas et al. 2023 (reported optimized range)	Thomas et al. 2023 (unoptimized)
AEP	2919 GWh	—	—

Metric	VelantisWind (single run)	Thomas et al. 2023 (reported optimized range)	Thomas et al. 2023 (unoptimized)
Wake loss	15.4%	15.48%–15.70%	17.28%

Table 2: Reference result retained from the expanded technical edition. The VelantisWind figure is one run and is not presented as a multi-seed statistical characterization.

For this specific physical configuration, the reported 15.4% wake-loss value is comparable with the 15.48%–15.70% optimized range cited in the expanded edition and lower than the 17.28% value of the corresponding unoptimized reference layout. The appropriate interpretation is evidence of competitive behavior on this case, not a universal performance claim. Site geometry, turbine count, constraints, physical model, computational budget and stochastic seed all influence optimization difficulty.

7.2 Reading the reference result correctly

A single reference run is evidence that the implementation can reach a competitive region of the benchmark under the stated physics and configuration; it is not proof of a universal advantage. Optimization is stochastic, and small differences near the end of a difficult run can depend on initialization, evaluation budget and the details of the feasible domain. The relevant engineering question is therefore not whether one number can be made slightly better in isolation, but whether strong results are obtained consistently, remain feasible, and survive independent physical re-evaluation.

For that reason, the validation program is intentionally moving toward distributions rather than anecdotes: multiple seeds, different site geometries and explicit computational budgets. This strengthens confidence in the optimizer while revealing far less about the proprietary search logic than publishing internal formulas or operator policies would.

7.3 Validation criteria for engineering use

- Repeatability. Report central tendency and spread over several stochastic seeds rather than only the best observed run.
- Constraint behavior. Verify that tighter or altered feasible regions produce understandable changes without silent infeasibility.
- Physical consistency. Re-evaluate final layouts under the stated wake assumptions and check that gains are not artifacts of bookkeeping or coordinate handling.
- Computational context. Report evaluation counts, runtime and hardware class so objective improvements can be compared against the effort required to obtain them.
- Cross-case generality. Use geometrically different public or partner cases so the method is not tuned only to one benchmark topology.

8 Limitations

Optimization cannot compensate for poor physical inputs. The credibility of a layout is bounded by the quality of the wind resource, turbine power and thrust information, active project constraints and wake-model assumptions. An optimizer can search the model it is given; it cannot recover information absent from that model. Likewise, maximizing AEP does not replace constructability, electrical, environmental, geotechnical or permitting analysis unless those requirements are explicitly represented in the optimization problem.

The public benchmark should also be read with the stated statistical limitation. A single run demonstrates that a competitive result can be reached under one configuration, but robustness is established through repeated runs and additional cases. The preferred direction for public validation is therefore more evidence, not more disclosure of the search recipe.

9 Disclosure scope

The boundary between a credible public method description and a reproducible implementation is deliberate. Established wind-energy equations, the constrained optimization objective, GIS interfaces, broad population roles and the validation protocol can be shared because they allow a reviewer to understand what problem is being solved and how final results should be checked. The elements that convert those principles into a competitive search engine remain proprietary.

PUBLIC HIGH-LEVEL FORMULATION	PROPRIETARY SEARCH IMPLEMENTATION
• Wind-resource and wake-coupled AEP formulation	• Exact populations, encodings and operator catalogue
• Constrained micrositng objective	• Internal state representation and diversity measures
• Population-based evolutionary design philosophy	• Adaptive decision rules, internal search memory and selection policy
• Functional roles: intensification, diversity preservation and structural exploration	• Cross-population exchange topology, scoring, quotas and cadence
• GIS / QGIS engineering interfaces and traceability	• Constraint handling, screening, caching and low-cost evaluation heuristics
• Physical benchmark configuration and published validation evidence	• Calibration data, constants, thresholds, budgets, schedules and stopping rules

Table 3: Disclosure boundary used for this public edition.

Knowing that an optimizer uses multiple populations, preserves diversity and adapts search effort is not sufficient to recreate its calibrated behavior. Reproduction would still require the hidden representation of search state, the operator set, the target-selection logic, the interaction between populations, constraint-repair procedures, adaptive decision rules, computational-budget policy and extensive calibration. Those elements are intentionally absent from this paper.

10 Conclusion

VelantisWind formulates wind-farm micrositing as constrained maximization of wake-coupled annual energy production inside a GIS-native workflow. The optimizer uses an evolutionary multi-population architecture in which different search roles coexist, useful candidate material can be coordinated across the run, and search emphasis can adapt as evidence accumulates. This is enough to distinguish the method from a single generic local or genetic search without turning the public document into an implementation specification.

The public technical case should therefore rest on three forms of evidence: a physically explicit objective, an auditable engineering workflow and reproducible external validation of final layouts. Competitive implementation details remain private. Future public editions should strengthen multi-seed statistics, additional benchmarks and computational reporting rather than disclose progressively deeper search mechanics.

11 Public validation roadmap

The preferred way to increase confidence in VelantisWind is to expand public evidence rather than progressively expose the search implementation. The next validation steps should therefore focus on breadth, repeatability and independent recomputation of final layouts under clearly stated physics.

- Multi-seed characterization: publish median, spread and worst/best ranges under a fixed benchmark configuration.
- Additional public geometries: repeat the protocol on cases with different turbine counts, boundary shapes and resource directionality.
- Constraint stress tests: vary the feasible domain systematically and verify that solution quality and feasibility degrade gracefully rather than discontinuously.
- Independent physical re-scoring: provide final coordinates and model assumptions so external reviewers can reproduce the reported AEP or wake loss without access to the proprietary optimizer.
- Computational reporting: pair energy results with evaluation counts, runtime and hardware context so algorithmic efficiency can be assessed alongside final objective value.

References

- [1] N. O. Jensen. A note on wind generator interaction. Risø-M-2411, Risø National Laboratory, 1983.
- [2] I. Katic, J. Højstrup, N. O. Jensen. A simple model for cluster efficiency. European Wind Energy Association Conference, 1986.
- [3] M. Bastankhah, F. Porté-Agel. A new analytical model for wind-turbine wakes. Renewable Energy, 2014.
- [4] G. Mosetti, C. Poloni, B. Diviacco. Optimization of wind turbine positioning in large wind farms by means of a genetic algorithm. Journal of Wind Engineering and Industrial Aerodynamics, 1994.
- [5] S. A. Grady, M. Y. Hussaini, M. M. Abdullah. Placement of wind turbines using genetic algorithms. Renewable Energy, 2005.
- [6] N. F. Baker, A. P. J. Stanley, J. J. Thomas, A. Ning, K. Dykes. Best practices for wake model and optimization algorithm selection in wind farm layout optimization. AIAA Scitech Forum, 2019.
- [7] J. J. Thomas et al. A comparison of eight optimization methods applied to a wind farm layout optimization problem. Wind Energy Science, 8, 865-891, 2023.
- [8] M. M. Pedersen et al. PyWake: an open-source wind farm simulation tool. DTU Wind Energy.
- [9] A. Niayifar, F. Porté-Agel. Analytical modeling of wind farms: a new approach for power prediction. Energies, 9(9):741, 2016.

© VelantisWind. Public high-level formulation. The implementation, parameters, operator definitions, search memory and adaptive policies are proprietary and are not disclosed here.