VelantisWind: an evolutionary island ecosystem for wind farm layout optimization

Physical formulation of the wind resource and wakes, and a multi-population algorithmic architecture with contextual learning

VelantisWind Technical Team • Technical White Paper (expanded edition)

Revision: expanded edition

Purpose and scope of this document. This white paper presents the physical, mathematical and architectural formulation of the VelantisWind optimization engine at a level intended to be technically convincing and auditable. It does not disclose the internal parameterization, the operator catalog, the update schedules, the calibration data or the budget policies, which constitute the proprietary core of the engine (see Section 12). The equations are included so that the principles are verifiable, not so that the engine can be reproduced.

Abstract

The micrositing of a wind farm - deciding where each wind turbine is placed within an allowed site - has a first-order effect on the energy a project produces over its operating life. The difficulty is that turbines are coupled: through its wake, each turbine modifies the wind field received by the others. The objective is therefore non-separable, the search space is large and discrete over the admissible terrain, and the landscape is strongly multimodal. VelantisWind does not address this with a single metaheuristic, but with an ecosystem of specialized populations that cooperate: a genetic exploitation island that intensifies the best layouts; a Quality-Diversity island that maintains a repertoire of diverse and competitive morphologies; and a structured recombination island that applies parameterized geometric transformations whose selection is governed by a contextual bandit defined over geometric descriptors, with multi-level statistical pooling and non-stationary weighting of learning. An orchestrator migrates solutions between islands, seeds them with the global best, diagnoses convergence and injects a bounded-budget surgical local-refinement phase only when a true plateau is detected. The whole process is designed to be traceable rather than opaque, and to integrate natively into GIS/QGIS workflows. This document formalizes the physics and high-level search architecture and frames the validation philosophy; it deliberately omits implementation details.

Keywords: wind energy • layout optimization • wake effect • evolutionary algorithms • Quality-Diversity • contextual bandit • GIS / QGIS

1 Introduction

The energy that a wind farm delivers over its operating life is highly sensitive to the relative position of its turbines. When a turbine operates in the wake of another, it receives slower and more turbulent air; this reduces production and increases fatigue loading on the structure. Two layouts built with the same turbines at the same site, differing only in geometry, can therefore exhibit materially different annual energy production (AEP) and different maintenance trajectories. Micrositing is the discipline of choosing that geometry well and, because a few percentage points of AEP accumulate over decades, it is one of the highest-leverage decisions in the early design of a project.

The problem is genuinely difficult because three properties coincide. First, the configuration space is huge and discrete: the admissible terrain - after removing exclusion zones, setbacks and infrastructure corridors - is a large set of candidate cells, and a layout is a choice of N of them. Second, the objective is non-separable: moving a single turbine changes the wind seen by many others, so AEP cannot be decomposed into independent turbine contributions. Third, the landscape is multimodal: many distinct geometric families are locally optimal, separated by valleys that a purely local search cannot cross. These characteristics have been recognized since the earliest applications of genetic algorithms to turbine placement [4, 5, 1].

VelantisWind approaches the problem from a GIS-native logic, unifying wind-resource reading, spatial constraints and energy calculation within a traceable workflow. Its contribution is not the adoption of one particular metaheuristic, but the search architecture itself: an ecosystem of specialized populations that exchange solutions, coordinated by a learning layer that decides which family of geometric transformation to apply according to the local context of the layout. Section 2 formalizes the physical problem; Section 3 explains why an ecosystem is the appropriate response; Section 4 describes the populations and their coordination; Section 5 walks through the anatomy of a round; and Sections 6-12 address interpretability, integration, validation and, finally, what this document reveals and what it does not.

2 Problem formulation

2.1 Wind resource and annual energy production

The wind resource of a site is described by the joint distribution of wind direction and wind speed. The wind rose is divided into directional sectors and, within each sector θ, wind speed is commonly modeled with a two-parameter Weibull distribution,

$$f(v \mid \theta) = \frac{k}{A}\left(\frac{v}{A}\right)^{k-1} \exp\left[-\left(\frac{v}{A}\right)^{k}\right] \tag{1}$$

where the scale parameter A sets the characteristic speed of the sector and the shape parameter k describes its spread. The set $\{(f_\theta, A_\theta, k_\theta)\}$ over all sectors is what encodes a wind rose: how frequently the wind blows from each direction and how strongly it tends to blow.

The annual energy production of a layout, identified by its coordinate set X, is the farm power integrated over all wind conditions, weighted by their occurrence frequency and scaled by the number of hours in the year T,

$$\text{AEP}(X) = T \sum_{\theta} \sum_{v} f(\theta, v)\, P_{\text{farm}}(X, \theta, v) \tag{2}$$

The essential subtlety lies in P_{farm}: it is not the sum of isolated turbine powers, because the wind that actually reaches each turbine has already been modified by the wakes of the others. Two derived magnitudes make the result interpretable for engineers and investors: the capacity factor,

$$\text{CF} = \frac{\text{AEP}(X)}{N\, P_{\text{rated}}\, T} \tag{3}$$

which expresses production as a fraction of the theoretical maximum, and the aggregate wake loss,

$$L_{\text{wake}} = 1 - \frac{\text{AEP}(X)}{\text{AEP}_{\text{free}}(X)} \tag{4}$$

the share of energy lost through turbine-to-turbine interference relative to a no-wake reference. Wind-farm micrositing is, in essence, the systematic minimization of L_{wake} under the site construction constraints.

2.2 Wake coupling

The velocity deficit induced downstream by a turbine is captured with analytical wake models. In the classical linear-expansion family [1, 2], the relative deficit at a downstream distance x is

$$\delta u = \left(1 - \sqrt{1 - C_T}\right)\left(\frac{D}{D + 2k_w x}\right)^2 \tag{5}$$

as a function of the thrust coefficient C_T, the rotor diameter D and the wake expansion rate k_w. More recent self-similar Gaussian models [3] replace the rectangular profile in Eq. (5) with a smoother radial shape and link the expansion rate to ambient turbulence intensity I_0 through an empirical relation of the form k_w ≈ α I0 + β [11], improving agreement with near-wake and far-wake measurements. What matters for optimization is qualitative and independent of the model: the deficit decays with downstream distance and expands laterally, so spacing along the dominant wind direction matters much more than transverse spacing.

When a turbine lies in the path of several wakes, the individual deficits must be combined. A common choice is quadratic, or energy-balance, superposition,

$$\left(\frac{\Delta u}{u_\infty}\right)^2 = \sum_{j \in W(i)} \delta u_j^2 \tag{6}$$

where W(i) is the set of turbines whose wakes project onto turbine i; linear and momentum-conserving variants also exist and are chosen to match the engine in use. This coupling - each turbine as a potential shadow over any other - is precisely what makes AEP non-separable. VelantisWind delegates the physical evaluation to an external and consolidated wake engine such as PyWake [10], and treats AEP as an expensive black-box function: each evaluation is costly, so the whole optimizer is designed to spend evaluations sparingly and where they are most likely to pay off.

2.3 The constrained optimization problem

The objective is to place a fixed number of turbines so as to maximize AEP, subject to every turbine lying in an admissible cell of the domain and to a minimum separation constraint between every pair:

$$X^\star = \arg\max_X \mathrm{AEP}(X) \quad \text{s.t.} \quad x_i \in \Omega,\ E(x_i, x_j) \geq 1\ \forall i \neq j,\ |X| = N \tag{7}$$

The admissible domain Ω is derived from the exclusion layers; the separation constraint is expressed by a safety region E(·, ·) around each turbine (Section 4.1). Three observations follow. The feasible set is a combinatorial object over a discretized terrain, so exact enumeration is infeasible for realistic N. The objective has no usable gradient and is multimodal, so descent methods stall in the first basin they encounter. And every candidate carries the cost of a wake evaluation, so the search must be sample-efficient. These three pressures - combinatorial feasibility, multimodality and evaluation cost - are the conditions under which the entire architecture is designed. All solutions handled by the islands are projected onto the feasible set through a common geometric repair operator, so feasibility is maintained by construction rather than recovered by rejection.

3 Design philosophy: why an ecosystem

A single metaheuristic must reconcile with one mechanism two opposing requirements: broad exploration of a landscape filled with local optima and, at the same time, fine exploitation of the best basins it has found. Tuning that balance is notoriously fragile. If the algorithm is biased toward exploration, it wanders without converging; if it is biased toward exploitation, it collapses prematurely into the first decent basin and never escapes. Worse still, the ideal balance is not fixed: it shifts as the search progresses and differs from one site to another.

An ecosystem dissolves this compromise by distributing it. Instead of asking one population to be simultaneously exploratory and exploitative, VelantisWind runs several populations in parallel, each specialized in one facet, and allows each to take its role as far as it remains useful:

- an exploitation population that intensifies and polishes the strongest layouts;
- a Quality-Diversity population that deliberately preserves a range of distinct, high-quality morphologies;
- a structured recombination population that performs large, coherent geometric reorganizations and learns which ones suit each context.

Specialization alone would produce three narrow algorithms. What turns it into an advantage is cooperation: the orchestrator migrates solutions between populations, so a promising basin discovered by diverse exploration can be handed to the genetic algorithm for exploitation, while a pattern that resists local refinement can be unlocked by a structural transformation taken from the recombination island. Three properties emerge from this organization that a single metaheuristic rarely obtains at the same time: robustness against premature convergence, knowledge transfer across morphologies and sites, and efficiency in the final stage, where refinement is concentrated only where it is justified.

4 The island ecosystem

The search is not performed by a single algorithm, but by several populations that evolve in parallel, specialize and exchange solutions under a coordinating orchestrator (Figure 1). We first describe the shared geometric substrate, then each population, and finally their coordination.

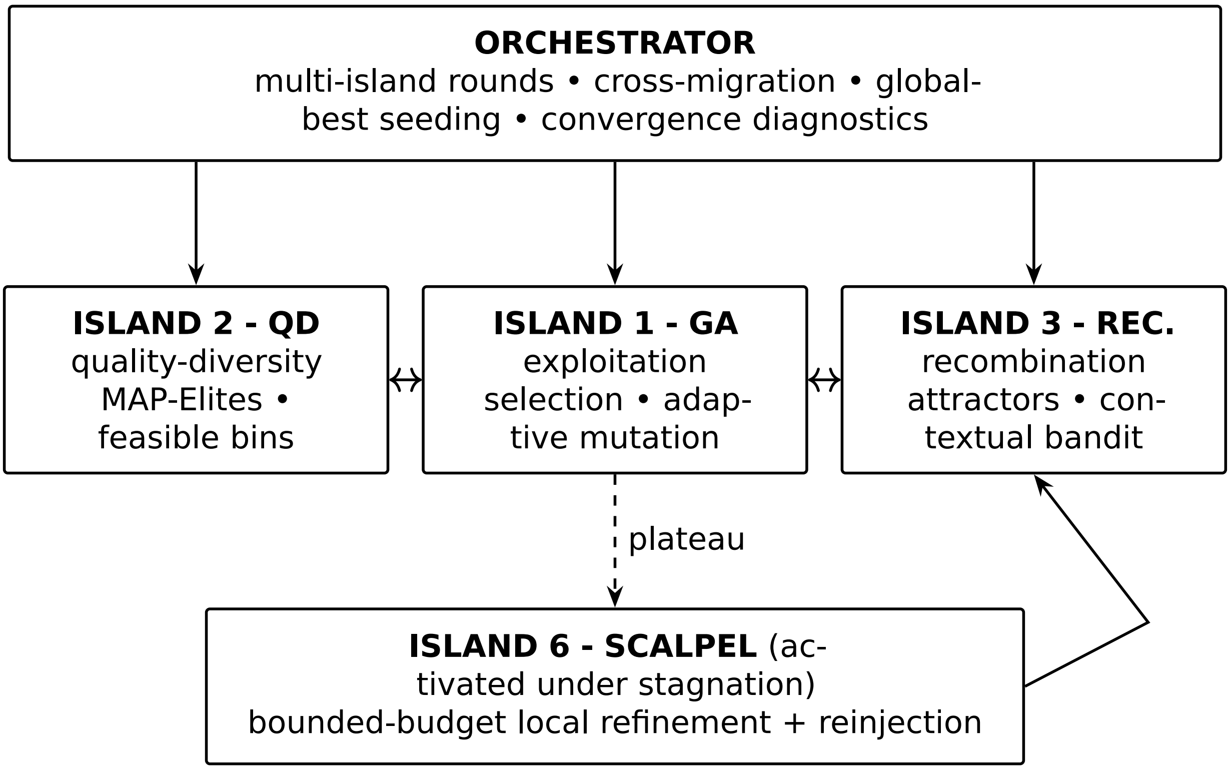

Figure 1: Conceptual architecture of the optimizer: specialized islands coordinated by an orchestrator, with a surgical local-refinement phase activated only under stagnation. Arrows between islands denote solution migration; the dashed path to Island 6 is taken only when a true plateau is diagnosed.

4.1 Geometric substrate and orientation to the wind resource

The constraints raster - exclusions, setbacks, infrastructure - is compiled once into a Boolean mask on a regular grid. From that point on, checking whether a coordinate is admissible is a fast lookup rather than a geospatial operation, allowing the islands to validate millions of candidate positions cheaply. Turbine separation is not modeled as an isotropic minimum distance. Instead, the safety region is anisotropic and elongated along the energetically dominant wind direction, because that is where wakes persist and where clustering is most damaging. The dominant sector and its orientation are obtained by weighting the frequency of each sector by the cube of its scale parameter - an approximation to its power content,

$$s^{\star} = \arg\max_{s} \left(f_s A_s^3\right), \qquad \alpha = (s^{\star} - 1) \cdot 22.5^{\circ} \tag{8}$$

Encoding physics directly into the feasibility constraint makes the optimizer spend its effort on the residual problem rather than rediscovering, layout by layout, that turbines should not be stacked downstream of one another. The same estimate of the dominant direction feeds both this feasibility geometry and the local morphological descriptors, so the resource reading is consistent throughout the engine and portable across sites. The exact geometry of the safety region is part of the implementation.

4.2 Island 1 - exploitation genetic algorithm

The first island is a genetic algorithm whose role is exploitation. Within the ecosystem it is the intensifier: it consumes material arriving through migration, refines it and returns its best energy elites to the loop. It operates on a population ranked by AEP and, despite its exploitative character, includes several feedback loops that prevent the population from collapsing before inter-island migrations have been able to act.

One principle runs through all its operators: every layout must be feasible by construction. Instead of generating infeasible solutions and discarding them - which would waste expensive wake evaluations - every operation passes through incremental geometric repair that preserves valid positions, relocates conflicts and fills gaps with admissible candidates.

Selection. Parents are chosen by fitness-proportional selection (roulette wheel),

$$p_i = \frac{\mathrm{AEP}(X_i)}{\sum_j \mathrm{AEP}(X_j)} \tag{9}$$

and the effective selection pressure is monitored through a participation ratio, the effective number of individuals that compete,

$$n_{\mathrm{eff}} = \left(\sum_i p_i^2 \right)^{-1} \tag{10}$$

When n_{eff} falls far below the population size, a few elites are monopolizing reproduction and diversity is being lost - a signal the island uses to adjust its own pressure.

Recombination. Crossover is spatial, not binary. Given a pair of parents ranked by fitness, several offspring are generated with adaptive proportions of spatial inheritance: each child preserves part of the structure of the higher-energy parent, incorporates information from the second parent and completes the remaining positions with admissible candidates, checking feasibility at every step. This recombines spatial subpatterns while keeping layouts valid at all times, rather than producing broken layouts and blindly repairing them afterwards.

Feasible candidate reserve. So that crossover and repair never complete a layout with purely random points, the island maintains throughout the run a reserve of candidate coordinates that combines geometric criteria with simplified resource signals. Proposed coordinates are never accepted directly: they always pass through the common geometric validation, so exclusions, boundaries, duplicates and minimum separation are respected. The reserve does not replace the AEP calculation; it reduces unproductive repairs and concentrates costly evaluations on feasible and reasonable layouts.

Adaptive mutation. Mutation relocates individual turbines to nearby valid gaps, and its intensity is not constant. The probability is updated at each generation according to the state of the search,

$$p_{t+1} = \mathrm{clip}(p_0 + s g_t + a D_t - b R_t,\ p_0,\ p_{\mathrm{max}}) \tag{11}$$

where g_t counts generations without improvement, D_t is a diversity pressure measured along several axes (mean nearest-neighbor distance, fraction of unique solutions, early clustering signals) and R_t is a repair pressure. Mutation intensifies when the search stagnates or loses variety, and moderates when the terrain is saturated with constraints and repairs become costly. The coefficients in Eq. (11) are part of the internal calibration.

Escape mode. If the best result does not improve for several consecutive generations, a controlled and reversible intensification is briefly activated: exploration capacity increases, local refinement is strengthened and the diversity quota in survivor selection is enlarged. An acceptance criterion ensures that an escape episode can never destroy already secured material.

Cost-aware directed local search. An optional bounded-cost local search is applied to the best layouts, prioritizing turbines with lower relative contribution and moves with higher geometric probability of improvement. For each turbine, angular directions that have historically worked are estimated with a forgetting UCB1 bandit over sectors,

$$q(s \mid i) = \frac{\rho_s}{N_s} + c\sqrt{\frac{\ln(t+1)}{N_s}} \tag{12}$$

whose visit counts and rewards are updated online. Before spending a full wake evaluation, candidate moves are pre-ranked with cheap geometric approximations and a cache avoids reevaluating configurations already seen. Only moves that increase AEP are accepted, and the improvement obtained feeds back into the search memory.

Survival with diversity and a safety net. The new population is formed by mixing offspring with parents, reinjecting the previous generation as a safety net so that a bad generation cannot erase discovered material, and deduplicating by signature to avoid carrying repeated genomes. The elite is strictly preserved, a quota is reserved for morphologically distant layouts and the rest is completed by AEP. The morphological distance between two layouts is measured as the mean distance from each turbine to its nearest neighbor in the other layout,

$$d(A,B) = \frac{1}{|A|} \sum_{i \in A} \min_{j \in B} \|a_i - b_j\| \tag{13}$$

The combination of strict elitism, an explicit diversity quota and adaptive mutation allows Island 1 to exploit aggressively without prematurely collapsing the shape of its population. Each generation records elitism, improvement, diversity and repair pressures, reserve state and refinement statistics, so the island behavior is reproducible under a fixed configuration and suitable for multi-seed statistical characterization.

4.3 Island 2 - Quality-Diversity and solution repertoire

Unlike the other islands, Island 2 does not attempt to produce one single best layout. Its objective is to illuminate the space of possible forms: for each morphological type of wind farm it preserves the best representative, assembling a repertoire of solutions that are simultaneously diverse and high-quality. This repertoire plays a double role. It is a safeguard, because it preserves the variety that exploitative islands naturally consume; and it is a quarry, because it supplies morphologically distinct building blocks for the genetic algorithm and the recombination island to combine and refine.

The island follows a Quality-Diversity paradigm in the style of MAP-Elites [7]: it maintains an archive with the best example in each region of a morphological descriptor space. A descriptor function Φ maps each layout to a cell, and the archive keeps only the dominant occupant of each cell,

$$\mathcal{A}[b] \leftarrow X \quad \text{if} \quad \text{AEP}(X) > \text{AEP}(\mathcal{A}[b]), \qquad b = \Phi(X) \tag{14}$$

What defines a cell. The descriptors that organize the map are not coordinates, but shape features: compactness, spacing regularity, distribution across available parcels and relationship with the global geometry of the domain. In simplified terms they combine nearest-neighbor distance measures with a relative variability, schematically

$$d_{\text{exp}} \approx g\sqrt{\frac{M_c}{n_c}}, \qquad \text{CV} = \frac{\sigma(d_{\text{nn}})}{\mu(d_{\text{nn}})} \tag{15}$$

with grid step g, valid cells M_c in the main component and its number of turbines n_c. Two layouts fall into the same cell when they are morphologically equivalent even if their coordinates differ: the map records solution types, not solutions.

Feasible cells. In a constrained site, many cells of an abstract descriptor grid correspond to forms that simply cannot be built. Treating those empty cells as a diversity failure would mislead the search. Island 2 therefore estimates, by probing the domain with valid layouts, which cells are actually reachable, and measures coverage against that reachable set rather than against the full grid; it can compact the map when too few cells are reachable. The discretization can also be set by quantiles of an observed descriptor rather than by rigid thresholds, so the archive resolution adapts to the actual distribution of layouts encountered by the search instead of imposing it a priori.

Hierarchical description. The repertoire does more than store diverse layouts: it turns each one into a morphological context for decision-making. It first captures the macrostructure - global pattern, compactness, spacing regularity, distribution across parcels, relationship with the domain - and then the microstructure within that family - local ordering of turbines, gaps, clusters, dense and open areas. These macro-micro labels turn map cells into decision contexts consumed by Island 3: the system learns not only whether a transformation is good on average, but in which type of structure it is appropriate.

These descriptors are not reduced to one or two magnitudes. The engine maintains a broad morphological vocabulary - anisotropy and principal axis by component analysis, concentration at the center versus the perimeter, balance of occupation between subzones, spacing regularity (grid-like versus clustered), local alignment with the dominant wind, relative load of each parcel and its proportion relative to available area, among others - and, more importantly, it records several families of descriptors in parallel through passive observers. This allows it to determine empirically which family best discriminates operator behavior, rather than fixing a representation a priori. These descriptors are normalized by the site geometry itself - from a site profile that precomputes the distance of each cell to the edge, the dominant axis and the expected spacing according to available area and number of turbines - so the same form receives the same reading regardless of the size or contour of the site, again supporting transfer between wind farms. The distribution of turbines across parcels, for example, is quantified as the discrepancy (total variation distance) between the observed distribution and the area-proportional distribution, a clean distributional criterion. The specific families and thresholds are internal.

Recombination credit. Finally, Island 2 does not merely preserve diversity: it learns which diversity turns into energy. For each combination of morphological types and operator family it records a credit score that combines how often the child beats the better of its parents and how often it beats their average - the latter keeps exploration alive - together with the average improvement achieved. The score is estimated with Bayesian smoothing, so a poorly observed pair is not judged on sparse data, and updated incrementally through the run; when a specific pair of morphologies lacks evidence, the credit falls back to the operator-only statistic. That credit feeds back into pair selection: pairing is jointly weighted by morphological distance (diversity), accumulated credit (history of converting that diversity into AEP) and the energy of the representatives, so recombination favors crosses that have proven productive without giving up variety. This memory does not impose any universal rule; it is learned during the run and depends on the terrain, resource and constraints.

4.4 Island 3 - structured recombination with contextual learning

The third island recombines solutions in a directed way. Its role is that of a large-scale combiner: instead of small perturbations, it applies parameterized geometric transformations at the scale of the whole farm, individual parcels and local substructures. These transformations reorganize a layout coherently and allow the search to jump between basins that local mutation could never bridge.

It uses the history of each solution and the best global solutions as attractors, in the spirit of particle swarm optimization [6], but acts through discrete transformations followed by repair,

$$X' = \mathcal{R}(X \oplus \omega_c(p_i - X) \oplus \omega_s(g - X)) \tag{16}$$

where the geometric recombination ⊕ pushes the layout toward its cognitive attractor p_i and its social attractor g, and the repair R returns it to the feasible set. The distinguishing element is how the transformation family is selected. Each situation is described through a geometric context extracted from the hierarchical repertoire of Island 2 - global macrostructure, parcel role, local microstructure - and, for each context, a distribution over operators is learned through a contextual bandit [8, 13].

For an operator o in a context c, its expected gain is estimated as the product of its success rate and its mean improvement when it succeeds. Because local evidence is often scarce, that estimate is shrunk toward the global estimate with a weight that grows with the number of observations,

$$EG_c(o \mid c) = \lambda\, EG_{\text{local}} + (1-\lambda)\, EG_{\text{global}}, \qquad \lambda = \frac{n}{n+n_0} \tag{17}$$

A confidence-bound term is then added to this gain, rewarding less-tested operators so that none is discarded by chance,

$$s(o \mid c) = EG_c(o \mid c) + c_u \sqrt{\frac{\ln(N_c+1)}{n_o+1}} \tag{18}$$

and the final choice is sampled from a temperature-controlled softmax distribution, mixed with a uniform exploration floor,

$$P(o \mid c) = (1-\varepsilon)\frac{\exp(s(o \mid c)/\tau)}{\sum_{o'} \exp(s(o' \mid c)/\tau)} + \varepsilon \frac{1}{|O|} \tag{19}$$

The operator catalog O, the descriptor that defines c and the constants λ, n0, c_u, τ, ε are part of the internal implementation.

Contextual learning: multi-level pooling and phase weighting. Operator selection is not governed by a single table of averages. The engine faces a familiar problem from multi-level statistical modeling: the most specific context - the local microstructure of a subregion - is precisely where evidence is scarcest, while contexts with abundant evidence - the global average - are too coarse to be useful. VelantisWind resolves this with a partial-pooling scheme over a three-level geographical hierarchy - local microstructure, parcel container and global - in which each level estimate is blended with the estimate from the level above using a weight that grows with the evidence accumulated at that level. A cold local context is therefore not abandoned to the global average: it first borrows strength from the statistics of its parcel, an intermediate level with geographical meaning that captures regularities blurred by the global level. The resulting signal combines the specific and the robust in the proportion justified by the evidence itself.

Second, the engine treats operator utility as non-stationary. At the beginning of a run, gains are easy and abundant; at the end, they are rare but decisive. Counting both equally would allow early and easy evidence to dominate the policy and mislead it in the decisive phase. The engine therefore weights the value of evidence according to search maturity while preserving the evidence count for confidence: a context is judged reliable when it has been sampled enough, but valued according to the evidence that truly matters. Each decision also records which level of the hierarchy - local, parcel or global - it came from, making the policy fully auditable. Phase thresholds, weights, pooling constants and the parcel taxonomy are internal.

Two design decisions make the learned behavior travel from one site to another. The first is scale invariance: the expected gain of each operator is normalized by the best in its context, so scores are comparable across contexts and across wind farms with different resources and sizes, and what is learned is a distribution of operators by morphology, not a fixed winning operator - in one site an edge repacking may emerge, while in another a transfer between parcels may emerge, depending on the evidence. The second is that the intermediate level of the hierarchy is not indexed by concrete parcel identifiers, which do not translate between sites, but by portable roles - main parcel, secondary parcel, medium parcel, fragment - so statistics accumulated for a role can be reused wherever that role appears again.

Where to attack: competitive selection with cooling. Deciding which operator to apply is not enough; it also matters which layout to act on. To avoid wasting effort by repeatedly hitting the current champion, the island chooses its targets through a stochastic mixture of the most competitive layouts, a quota of exploratory layouts from the diverse repertoire and a purely random fraction, with cooling that penalizes acting again on a layout that has just been attacked. This maintains pressure on the promising material without collapsing into a single solution or neglecting the diversity preserved by Island 2.

Structural operators as a morphological scalpel. Alongside fine perturbations, the island has structural transformations - transferring a turbine to an underused parcel according to its load versus area, compacting toward a valid edge or releasing toward the interior, sliding tangentially along an edge - that reorganize occupation at parcel scale. The division of labor is deliberate: Island 2 discovers which structures exist and are reachable, and Island 3 tests, surgically and with low budget, whether they are worth exploiting. The exact inventory of these transformations is proprietary.

Directed attack on stressed zones. Structured recombination is complemented by low-budget directed refinement. Before spending costly wake evaluations, the island flags locally stressed zones using cheap geometric signals - turbine proximity, local density, alignments compatible with the dominant resource, relationship with boundaries and parcels - and uses them to decide which turbines should be moved and which perturbation to test. The goal is not exhaustive search, but placing AEP evaluations where improvement is most likely. Candidate moves are accepted only if they increase production, and their results feed the contextual memory. The island also monitors the recent contribution of each transformation family: when a family stops helping and starts degrading AEP, its budget is temporarily reduced and the search falls back toward more conservative alternatives.

Why it matters. Because learning is associated with general geometric descriptors and not with one particular map, what is acquired is not ”operator o is universally good”, but ”this mixture of transformation families suits this morphology and this level of evidence”. Knowledge therefore transfers across sites. The definitions of the descriptors, the operator catalog, the update schedules and the budget policy remain proprietary.

4.5 Orchestration and migration

An orchestrator runs the cycle in rounds. Within each round the islands advance and solutions are migrated among them - energy elites, diverse representatives and contextual memory - with duplicate suppression, so migration adds information rather than redundancy. Migration is neither symmetric nor reduced to copying the best solutions by energy: toward the exploitation island, each candidate is scored by combining its morphological novelty relative to the best target layout, its energy and the learned credit of the combination that produced it, and the migrant batch is built to be diverse - each new migrant is reweighted by how much novelty it adds relative to those already selected, avoiding several nearly identical copies. In this way, the material transferred is simultaneously good, different and historically productive. Before migration, the orchestrator seeds each island with the best global layout known, so no island loses sight of the current optimum. It continuously monitors progress, diversity and the spatial distribution of solutions, and uses those diagnostics to decide when to intensify, when to diversify and when to activate refinement under plateau conditions. The migration topology, quotas and cadence are part of the implementation.

4.6 Island 6 - surgical local search under stagnation

When the global best result does not improve above a threshold for several consecutive rounds - a genuine plateau, not a momentary pause - the orchestrator activates a dedicated refinement phase,

$$\text{activate Island 6} \quad \Longleftrightarrow \quad \Delta_r^\star \le \varepsilon_6 \text{ for } R \text{ rounds} \tag{20}$$

Island 6 takes the best layouts from each island and subjects them to bounded-budget local micro-restructuring, accepting only changes that improve the objective, and reinjects the results into the islands for subsequent exploitation. Isolating it as an on-demand phase, rather than keeping it always active, avoids spending the most expensive resource - wake evaluations - on polishing that only pays off when the global search has truly stagnated. By design, it never interferes while the search is progressing healthily; it acts only in the final phase. The activation threshold, refinement budget and stopping rule are part of the implementation.

The budget of this phase is also adaptive. Instead of using a fixed number of refinement passes, the phase keeps acting while its marginal improvement exceeds a significance threshold and stops when it converges, avoiding both cutting off a still-profitable advance and insisting without gain. And if the plateau is diagnosed so late that there is barely any runway left to exploit what the scalpel unlocks, the orchestrator extends the execution by a few additional rounds, so a late discovery is not wasted. In the same spirit, the confidence the engine places in its learned contextual policy ramps up as the search matures and accumulates evidence, consistent with the phase weighting described above.

5 Anatomy of a round

It is useful to see how the pieces fit together in time, because the order of operations is part of the design. At a high level, an orchestrator round proceeds as follows.

First, the orchestrator seeds each island with the best global layout known, so no population remains anchored to a local optimum while another has already surpassed it. Next, each island advances with its own engine for a tranche of work: Island 1 executes its genetic generations, Island 2 updates its morphology archive, and Island 3 proposes and evaluates structural transformations guided by its contextual policy. These advances are largely independent, which allows energy evaluations - the bottleneck - to be distributed in parallel when the underlying engine permits it.

Once those advances are complete, the orchestrator migrates solutions between islands. Migration is not symmetric: energy elites and diverse representatives travel toward the exploitation island, selected so that they add morphological novelty as well as quality, in order not to homogenize the population or contaminate it with poor material; contextual memory also travels toward the recombination island. Every migration passes through duplicate suppression, so what is transferred is new information.

Finally, the orchestrator diagnoses. It measures progress of the global best, aggregate diversity and the spatial distribution of solutions, and uses this to decide the character of the next round:

intensify, diversify or, if a real plateau sustained over time is detected, activate Island 6 surgical phase.

Following the life cycle of a candidate move inside the recombination island is particularly illustrative, because it summarizes the cost discipline that governs the entire system. A move is born from a geometric context; the contextual policy, with its multi-level pooling and phase weighting, chooses which transformation family to apply; the transformation is materialized and repaired to return it to the feasible set; the resulting candidate is pre-ranked with cheap geometric criteria and checked against a cache, so configurations already seen or manifestly poor never consume a physical evaluation; only then, if it passes that filter, is its AEP evaluated with the wake model; and the result - whether it improves or not - feeds back into the contextual memory, which learns for the next time. In this way, the expensive evaluation is reserved for candidates most likely to inform the search.

6 Interpretability and traceability

A recurring weakness of metaheuristic optimization in engineering practice is that it behaves like a black box: a layout comes out, but the reasoning that produced it is invisible, making it difficult to trust the result and even harder to audit it. VelantisWind is built to avoid this. Each generation and each round record the quantities that explain search behavior - elitism and improvement, diversity and repair pressures, candidate-reserve state, migrations performed, convergence diagnostics and operator decisions taken in each geometric context, including the evidence level each decision came from.

This traceability serves three audiences. For the engineer, it turns a result into an explanation: why a layout was favored, where energy was being lost, which transformations unlocked progress. For the reviewer or client, it transforms the optimizer from an opaque oracle into an auditable process whose assumptions can be inspected. And for the development of the method itself, it provides the evidence base used to characterize operator utility and search behavior across seeds and sites. It is important to note that traceability and confidentiality are not in tension: the engine can expose what it decided and why, in terms of energy and geometry, without exposing the internal constants that make those decisions possible.

7 Evidence-based development

The same traceability that makes the result auditable also governs how the engine is developed. Every mechanism described in this document is incorporated through an explicit experimental discipline, and it is worth explaining because it is itself a quality assurance mechanism. Before acting on a hypothesis, the system instruments it: passive observers record candidate behavior without altering the search, so evidence accumulates on whether a signal discriminates anything useful before it is allowed to influence a single decision. When an idea becomes active, its success is not judged only by final energy - a noisy and late metric - but by its direct contribution: whether the guided mode measurably outperforms the base behavior in the same contexts. And every change is reversible: each mechanism can be disabled to recover exactly the previous behavior, making it possible to isolate its effect and revert, at no cost, a line that does not contribute. The result is a traceable and cumulative evolution of the method, in which design decisions rest on evidence rather than intuition, and lines that do not perform are closed with data instead of being carried forward. This methodology is independent of the specific parameters it produces, which remain internal.

8 Computational considerations

The dominant cost of the entire process is the wake-based AEP evaluation; everything else - geometric validation, descriptor calculation, bandit update - is cheap in comparison. The architecture is therefore organized around one discipline: do not spend a wake evaluation unless it is likely to provide information. Three mechanisms, already described in context, serve this purpose. Feasibility by construction eliminates evaluations wasted on invalid layouts. Cheap geometric pre-ranking and caching, both in Island 1 local search and Island 3 directed attack, filter and deduplicate candidate moves before invoking the physical model. In candidate generation, this is complemented by a no-wake energetic pre-screen with a spatial cache that prioritizes areas of stronger resource; characteristically, that shortcut is used only when the resource varies enough spatially to provide information, falling back to uniform sampling when it does not. And the on-demand character of Island 6 confines intensive refinement to moments when it is justified. Where the underlying engine permits it, independent evaluations are dispatched in parallel. Concrete budgets, batch sizes and scheduling policies are part of the implementation; the design principle - evaluation as a scarce resource that must be protected - is the part worth stating publicly.

9 Integration with GIS / QGIS workflows

VelantisWind is conceived as a GIS-native tool, not as a standalone solver. Constraint layers, wind-resource surfaces, turbine positions and optimized results are not handled as disconnected files, but as spatial information within a single working environment. In practice this yields three advantages. It reduces friction in early design, where assumptions about exclusions, setbacks and resource change frequently and must propagate immediately to the optimization. It makes those assumptions auditable, because every constraint that shaped a result is a visible and georeferenced layer rather than a hidden input. And it allows optimization to connect directly with the deliverables that downstream stakeholders actually use - maps, georeferenced layouts and reports in the project coordinate reference system - so the output of the search is immediately usable in the surrounding engineering process.

10 Validation

The method has been verified on the public IEA37 Case Study 4 benchmark [9], with 81 turbines of 10 MW and its official wind rose, over a domain clipped by a constraint mask. In a complete and stable run, the AEP of the best layout grows monotonically throughout the optimization, without evaluation errors, and the final layout is delivered georeferenced in the project coordinate reference system. The full trace - migrations, convergence and operator decisions by context - is recorded, so the optimizer behavior can be audited rather than accepted on faith.

Validation is deliberately framed as more than a single final-AEP comparison. A method intended for engineering use must be stable (similar quality across random seeds, not occasional lucky runs), well behaved under constraints (a gradual response as exclusion zones narrow the feasible set) and physically coherent (layouts that respect the logic of the dominant wind rather than exploiting model artifacts). Statistical characterization over multiple seeds and additional cases is part of the ongoing validation program, and the traceability described in Section 6 is what makes that program tractable.

10.1 Quantitative result from a reference run

As a complement to the qualitative validation described above, the quantitative result of a reference run on the same IEA37 Case Study 4 benchmark (81 turbines of 10 MW, official wind rose) is reported below. The run was evaluated under the Bastankhah Gaussian wake model with k_y = 0.0324555 and TI = 0.075 - the same parameters and wind rose used as reference in Thomas et al. (2023, Wind Energy Science) - to allow direct comparison with the published state of the art.

Metric	VelantisWind (single run)	Thomas et al. 2023 (optimum)	Thomas et al. 2023 (unoptimized)
AEP	2919 GWh	–	–
Wake loss	15.4%	15.48%–15.70%	17.28%

The wake loss obtained, 15.4%, is in line with - or slightly below - the 15.48%-15.70% range reported by state-of-the-art optimization methods in Thomas et al. (2023), and notably below the 17.28% of the unoptimized layout in the same reference.

For the moment, this is the result of a single run rather than a multi-seed characterization; the figure is reported with that explicit caveat and will be expanded with additional statistics as part of the ongoing validation program described in Section 10.

11 Discussion

The design rests on four principles - specialization, cooperation, contextual learning and surgical refinement - and their combination is what produces a search that is robust to local optima, transfers knowledge between sites and keeps its decisions interpretable. The honest limitations follow directly from the problem. Search cost is dominated by wake evaluation, so achievable performance is bounded by the physical engine; and the quality of any result is bounded by the quality of its inputs - the wind resource and the constraint layers. Neither limitation is specific to VelantisWind, but both should be stated clearly: an optimizer cannot recover energy that the resource data do not contain, nor can it respect a constraint it was never given.

12 Disclosure scope

This document is intended to make the method convincing and auditable, not to make the engine reproducible. The distinction is deliberate. The physics (Section 2), the justification for an ecosystem (Section 3), the role of each population and the mathematical form of the public concepts on which they rest - roulette-wheel selection, MAP-Elites dominance, UCB exploration, operator selection by softmax, recombination by attractors from swarm methods, and the multi-level partial pooling characteristic of hierarchical statistical modeling - are presented openly because they are established science and because disclosing them is what justifies trust in the approach.

What is not disclosed, and constitutes the proprietary core of the engine, includes: the internal operator catalog and the exact geometric and structural transformations; the descriptor functions that define morphological contexts and cells, as well as which descriptor family is selected; the parcel taxonomy and definition of the levels of the learning hierarchy; the rules and constants for recombination-credit smoothing and the edges of quantile discretization; all constants, coefficients, temperatures and thresholds that appear in update rules; the phase weights, confidence ramps and schedules governing adaptation, escape, target cooling and budget modulation; the migration topology, novelty/energy/credit weights, quotas and cadence; the calibration data and the empirical weighting of operators derived from them; and the heuristics that pre-rank movements before evaluating them. These elements are the result of extensive experimentation and are what differentiate the engine in practice. Their omission here is not a gap in the formulation, but a deliberate boundary between what can be shared to build confidence and what must remain protected to preserve the contribution.

13 Conclusion

VelantisWind formulates micrositing as the maximization of wake-coupled AEP under exclusion constraints, and solves it through an ecosystem of specialized cooperating populations equipped with contextual learning - with multi-level statistical pooling and non-stationary weighting - and an on-demand surgical local-refinement phase. The architecture distributes the exploration-exploitation trade-off across specialized islands, transfers knowledge among them through migration and concentrates its most expensive resource - physical wake evaluation - where it is most likely to pay off, all within a traceable and GIS-native workflow. This document has presented the physics, mathematics and architecture of the method at a high level; the parameters, operators and implementation decisions that turn that architecture into a calibrated and competitive engine remain proprietary.

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. J. Wind Eng. Ind. Aerodyn., 1994.

[5] S. A. Grady, M. Y. Hussaini, M. M. Abdullah. Placement of wind turbines using genetic algorithms. Renewable Energy, 2005.

[6] J. Kennedy, R. Eberhart. Particle swarm optimization. Proc. IEEE Int. Conf. on Neural Networks, 1995.

[7] J.-B. Mouret, J. Clune. Illuminating search spaces by mapping elites. arXiv:1504.04909, 2015.

[8] P. Auer, N. Cesa-Bianchi, P. Fischer. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 2002.

[9] 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.

[10] M. M. Pedersen et al. PyWake: an open-source wind farm simulation tool. DTU Wind Energy.

[11] A. Niayifar, F. Porté-Agel. Analytical modeling of wind farms: a new approach for power prediction. Energies, 9(9):741, 2016.

[12] A. Cully, Y. Demiris. Quality and diversity optimization: a unifying modular framework. IEEE Transactions on Evolutionary Computation, 22(2):245-259, 2018.

[13] T. Lattimore, C. Szepesvári. Bandit Algorithms. Cambridge University Press, 2020.