Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Voronoi-Elitism Genetic Algorithm: A Generic Derivative-Free Routine With Theory and Implementation for Statistical Optimization

Anthony Zou
River Bluff High School
Yizhou Cai
yizhouc@email.sc.edu
University of South Carolina
Ting Fung Ma
University of South Carolina

Research Article

Keywords: Breakpoint estimation, Evolutionary computation, High dimensional optimization, Segmented regression, Stochastic optimization, Voronoi partition

Posted Date: June 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9861196/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Voronoi-Elitism Genetic Algorithm: A Generic Derivative-Free Routine With Theory and Implementation for Statistical Optimization

Anthony Haitao Zou[1], Yizhou Jake Cai[2*], Ting Fung Ma[2]

[1]River Bluff High School, 320 Corley Mill Road, Lexington, 29072, South Carolina, USA.

[2]Department of Statistics, University of South Carolina, 1523 Greene Street, Columbia, 29208, South Carolina, USA.

*Corresponding author(s). E-mail(s): yizhouc@email.sc.edu;
Contributing authors: loggamma@gmail.com; tingfung@mailbox.sc.edu;

Abstract

In this paper, we propose a generic optimization approach for challenging objective functions that finds applications in various statistical problems. We focus on objective functions with two parameter blocks of one amenable to analytic optimization, and another that is irregular or computationally expensive. To address this setting, we propose the Voronoi-Elitism Genetic Algorithm (VEGA), a derivative-free optimization method that embeds geometric information into genetic search. The proposed algorithm retains elite candidates and constructs Voronoi-based neighborhoods around them, whose crossover and self-adaptive mutation balance exploitation of promising solutions with exploration of undercovered regions. We study the high dimensional behavior of genetic search by analyzing distance concentration, and the effects of population size and shrinking mutation, which shows that the algorithm improves spatial coverage and yields sharper distance bounds under limited computational budgets. Simulation studies are conducted to compare VEGA with two genetic-type algorithms competitors in finite samples. A real data application on Stack Exchange activity data further illustrates its ability to identify stable structural changes, implying the algorithm is computationally flexible for high-dimensional, derivative-free optimization and applicable for various statistical problems.

Keywords: Breakpoint estimation; Evolutionary computation; High dimensional optimization; Segmented regression; Stochastic optimization; Voronoi partition

1 Introduction

Parameter estimation is often formulated as an optimization problem. This perspective underlies a large class of statistical inference and learning, where model fitting is typically driven by the optimization of a loss function. When the objective function is convex and differentiable, estimators can often be obtained by characterizing the optimizer directly or by applying classical iterative procedures such as Newton-Raphson and gradient descent (Ypma 1995; Cauchy 1847). In less regular problems, however, optimization becomes substantially more challenging.

When the objective function is not differentiable, methods that rely directly on derivatives or gradients are no longer applicable in their classical form. If convexity is preserved, one may instead resort to sub-gradient-based methods. Adaptive first-order methods, including AdaGrad, ADADELTA, and Adaptive Moment Estimation (ADAM), are widely used in modern data science applications and can be implemented with sub-gradients (Duchi et al. 2011; Zeiler 2012; Kingma and Ba 2015). Empirically, these methods often perform well for common objective and loss functions, illustrating their practical effectiveness in large-scale optimization. From a theoretical perspective, however, objective functions may be nonconvex, non-smooth, or otherwise irregular. In such settings, convergence analysis often requires additional structural assumptions, such as the Kurdyka-Lojasiewicz property (Attouch et al. 2013), while some procedures may still be used heuristically even when few regularity assumptions are available.

For particularly difficult objective functions, grid search is often used as a last-resort optimization strategy. Suppose, after suitable rescaling, the target function $f(\boldsymbol{X})$ is defined on a compact domain $\boldsymbol{S} = [0,1]^p$. Let $a_0 < a_1 < \cdots < a_k$, where $a_0 = 0$ and $a_k = 1$, be a partition of $[0,1]$ such that $\min_i |a_i - a_{i-1}| \to 0$. Applying this partition along each coordinate yields a p-dimensional grid over $\boldsymbol{S}$. The point attaining the largest function value on this grid provides a discrete approximation to $\operatorname{argmax}_{\boldsymbol{X}\in\boldsymbol{S}} f(\boldsymbol{X})$. Although grid search is conceptually straightforward and can be accurate when the grid is sufficiently fine, it is computationally expensive (Bergstra et al. 2011). Its complexity is of order $\mathcal{O}(k^p)$, which quickly becomes unaffordable in moderate or high dimensional problems and reflects the curse of dimensionality (Bellman 1957). This motivates the development of alternative optimization strategies in problems with exploitable structure.

A further limitation of grid search is that it is naturally designed for continuous parameter spaces, whereas many applications involve discrete, categorical, or mixed-type variables. Such settings arise frequently in hyperparameter optimization, where the search space may contain continuous, discrete, categorical, and conditional components (Bergstra et al. 2011). In practice, procedures such as the Stepwise Categorical Progress algorithm (Cai and Ma 2025) may be used in such settings. Discrete optimization is itself a broad and active area of research, but it lies outside the scope of this paper.

From the perspectives of statistics and data science, we are often interested in maximizing an objective function over a parameter space and treating the maximizer as the estimator. The class of problems we study occupies an intermediate regime that the objective is neither fully regular nor highly irregular objective functions. As examples, (Cai and Ma 2025) raised a specific scenario as following.

Suppose the target function is $f(\boldsymbol{\theta}, \boldsymbol{Z})$, where $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{k_1}$ and $\boldsymbol{Z} \in \mathcal{Z} \subset \mathbb{R}^{k_2}$ are parameter vectors for some $k_1, k_2 \in \mathbb{N}^+$. We separate the parameters into $\boldsymbol{\theta}$ and $\boldsymbol{Z}$ because they have different mathematical properties. Conditional on $\boldsymbol{Z}$, $f(\boldsymbol{\theta}, \boldsymbol{Z})$ is mathematically well-behaved and easy to be optimized. In contrast, given $\boldsymbol{\theta}$, optimization over $\boldsymbol{Z}$ is complicated and computational challenging. This pattern captures the intermediate regime above. Our goal is to maximize this function and to get the joint maximizers, i.e.,

$$(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{Z}}) = \underset{(\boldsymbol{\theta}, \boldsymbol{Z}) \in \Theta \times \mathcal{Z}}{\operatorname{argmax}} f(\boldsymbol{\theta}, \boldsymbol{Z}). \tag{1}$$

Notice that $f(\boldsymbol{\theta}, \boldsymbol{Z})$ may be analytically complicated because of its dependence on $\boldsymbol{Z}$. In particular, it could ask for joint differentiability. Hence, many standard optimization methods, including Newton-Raphson and Block Ascending Gradient, would fail. In line with statistical applications, we therefore assume the target function is smooth on $\boldsymbol{\theta}$, while remaining $\boldsymbol{Z}$ as parameters that are numerically challenging to optimize.

Formally, we could descrine the intermediate structure as follows. Given any $\boldsymbol{Z}_0 \in \mathcal{Z}$, it is straightforward to derive $\widehat{\boldsymbol{\theta}}$, where

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} f(\boldsymbol{\theta}, \boldsymbol{Z}_0), \tag{2}$$

but on the reverse, it is still demanding to derive $\widehat{\boldsymbol{Z}}$ given $\boldsymbol{\theta}_0 \in \Theta$, where

$$\widehat{\boldsymbol{Z}} = \underset{\boldsymbol{Z} \in \mathcal{Z}}{\operatorname{argmax}} f(\boldsymbol{\theta}_0, \boldsymbol{Z}). \tag{3}$$

An iterative algorithm can therefore be constructed by alternating between the optimization steps in (2) and (3), yielding a joint local estimator for (1). Furthermore, under joint concavity of the objective function and suitable regularity conditions on the parameter space, a local maximizer is also a global maximizer. Although the optimization in (3) may remain computationally expensive, the alternating strategy is closely related to the principle of block coordinate ascent. Equivalently, it can be viewed as block coordinate descent applied to $-f$ (Tseng 2001).

2 Background and Approaches

2.1 Optimization in Statistical Methods

Optimization has been central to statistical methodology, not only as a computational step but also as the fundamental definition of various estimators. Many estimators such as Maximum likelihood estimation (MLE), penalization estimator (Tibshirani 1996), variational inference (Blei et al. 2017), and M-estimation (Huber 1964) can all be written as optimization problems. In these settings, the statistical properties of an estimator are tied to the geometry of the objective function, whose identification is expressed through the population optimizer, while asymptotic behaviors depend on curvature and local behavior near the optimum (Hansen 1982).

The computational difficulty, however, varies substantially across statistical problems. Smooth and convex objectives can often be handled by classical gradient, Newton-type or coordinate-wise methods, whereas non-smooth, discrete and non-convex objectives require different treatment. Important examples include regression quantiles (Koenker and Bassett 1978), maximum score estimation (Manski 1975), and structural-change or segmented-regression models with unknown breakpoints (Bai and Perron 1998; Tseng 2001).

The setting considered in this paper lies in this intermediate regime. Conditional on the irregular block $\boldsymbol{Z}$, the regular block θ can often be optimized by analytic or standard statistical procedures. In contrast, the optimization over $\boldsymbol{Z}$ is derivative-free and computationally expensive. This separation motivates the GA framework developed as following. The genetic algorithm searches the irregular $\mathcal{Z}$, while traditional estimation is retained for θ. Specifically, the reliability of the framework depends heavily on the efficient of the search on $\mathcal{Z}$ and we propose a mechanism to improve the high dimensional performance under a limited computational budget. See Kolda et al. (2003); Rios and Sahinidis (2013) for examples.

2.2 Genetic Algorithm

The main computational difficulty in our framework is driven by the $\boldsymbol{Z}$ block, for which standard derivative-based approaches are generally unsuitable. Accordingly, we focus on the derivative-free methods employment.

Genetic algorithm (GA) is a class of derivative-free numerical methods designed for optimizing irregular or pathological objective functions. GA are efficient random search procedures that have been studied and modified over several decades. Figure 1 illustrate the structure of a standard GA. In our setting, the GA is used to optimize over $\boldsymbol{Z}$, whereas $\boldsymbol{\theta}$ is obtained analytically under the assumed structure of the objective function.

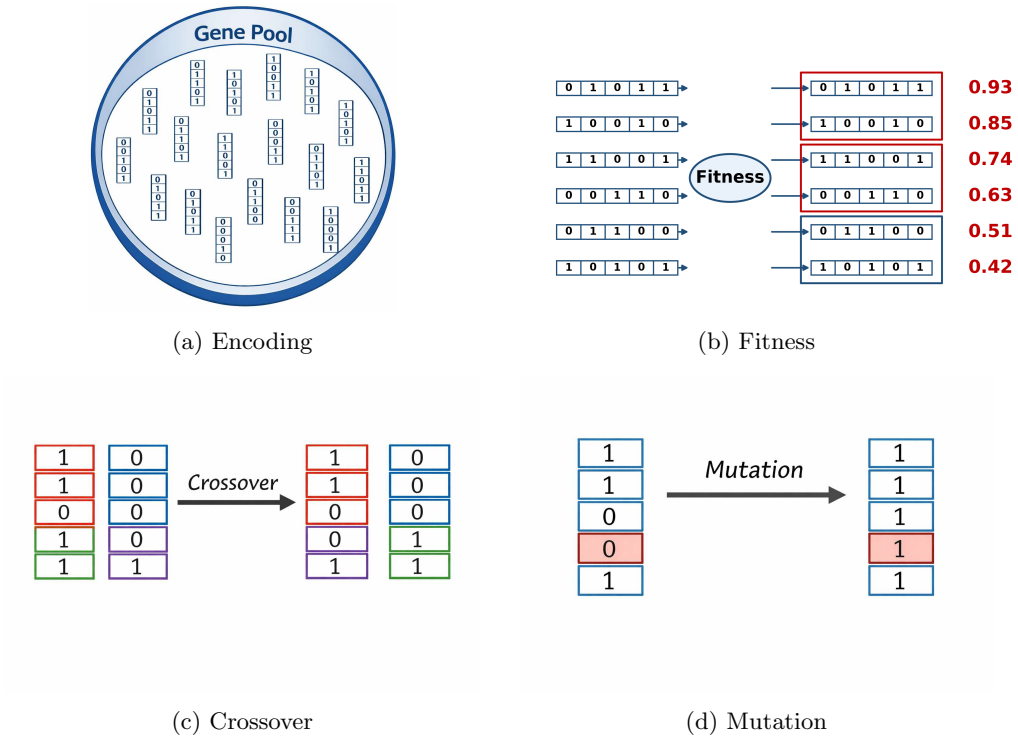
GA mimics the evolution mechanism in nature, particularly the principle of survival of the fittest. The algorithm maintains a population of candidate solutions for $\widehat{\boldsymbol{Z}}$, which could be viewed as a gene pool of a species. Through repeated selection and reproduction, the population evolves toward regions of the parameter space associated with higher objective values. At the end of the procedure, the best performing individual is taken as the solution. The main components of the algorithm are encoding, fitness evaluation, crossover, and mutation.

- **Encoding** maps each candidate solution to a chromosome representation, typically a vector or another equivalent structured object. Depending on the application, the chromosome may consist of binary, integer-valued, or real-valued components, which represent the underlying parameter values after decoding. Encoding could be viewed as a mapping $f : \mathcal{Z} \to \mathcal{A}$, where $\mathcal{A}$ denotes the chromosome space.
- **Fitness Evaluation** quantifies the quality of each candidate chromosome, after which the population is ranked according to fitness. This ranking determines the extent to which a candidate contributes genetic material to the next generation. Such contribution could be defined through different mechanisms, including

survival probabilities or expected offspring counts. In unconstrained problems, the fitness function is typically the objective function itself, while in constrained problems, a penalized version is often used instead. Common examples include cross-entropy loss and negative log-likelihood function.

- **Crossover** produces new chromosomes from parents. Chromosomes are paired according to a specified selection rule, typically preferring individuals with higher fitness. Each pair serves as parents, and crossover combines chromosomes from the parental chromosomes to produce offspring for the next generation. These offspring form new candidate solutions for $\widehat{\boldsymbol{Z}}$.
- **Mutation** is the final step in each generation and used to introduce additional diversity into the population. The mutation mechanism depends on the representation of the chromosome. For example, real-valued components may be perturbed continuously, whereas binary components may be flipped with a specified probability.

For a comprehensive treatment of GA, see Eiben and Smith (2015).



(a) Encoding (b) Fitness

(c) Crossover (d) Mutation

Fig. 1: Key components of the genetic algorithm: chromosome encoding, fitness evaluation, crossover, and mutation. The schematic summarizes how candidate solutions are represented, scored, recombined, and perturbed across generations.

In practice, GA is often used as a global search technique, since the initial population is sampled from the entire parameter space. Its empirical performance generally improves as the population size increases and more generations are reproduced, although it comes with greater computational cost and complexity. Given a limited computational budget, small modifications to the selection, crossover, or mutation steps materially affect performance. With a sufficiently diverse initial population, GA often identifies regions close to a global maximizer within relatively few generations. However, the quality of later stage refinement depends strongly on the mutation mechanism .

To improve eventual convergence, a shrinkage is proposed into the mutation. Under this strategy, the region over which mutation is allowed shrinks as the number of generations increases. In practice, the shrinkage rate is often taken to be $\mathcal{O}(m^{-1/2})$, where m denotes the generation index. This scaling is broadly consistent with convergence rate from classical optimization, as well with rates that arise in some statistical problems under independence or weak dependence.

Although mutation shrinkage can significantly improve GA performance in many settings, its limitations are more pronounced in high dimensional scenarios. Due to the curse of dimensionality, randomly initialized candidates tend to concentrate near the boundary of the parameter space, and a very large population are required to sufficiently explore its interior. Therefore, when the dimension is high and the population size is small or moderate, mutation shrinkage at rate $\mathcal{O}(m^{-1/2})$ may become too aggressive to leave substantial regions of the space unexplored.

2.3 Challenges on Complexity

We specialize the case in which $\partial f(\boldsymbol{\theta}, \boldsymbol{Z})/\partial \boldsymbol{\theta}$ exists but $\partial f(\boldsymbol{\theta}, \boldsymbol{Z})/\partial \boldsymbol{Z}$ does not. In our specific framework, GA treats $\boldsymbol{Z}$ as the chromosome and evaluates each candidate using the target function itself. Given a candidate value of $\widehat{\boldsymbol{Z}} = \boldsymbol{Z}$, we solve (2) to obtain the corresponding optimizer with respect to $\boldsymbol{\theta}$. Iterating this process over successive generations produces a numerical approximation $(\widehat{\boldsymbol{\theta}}_{GA}, \widehat{\boldsymbol{Z}}_{GA})$ to the joint maximizer.

Suppose the population size in each generation is fixed as n, the GA is run for m generations, and the computational complexity of evaluating $f(\boldsymbol{\theta}, \boldsymbol{Z})$ is $\mathcal{O}(G)$. Ignoring the costs of crossover and mutation, the total complexity of the algorithm is $\mathcal{O}(nmG)$. Since G is pre-determined by the structure of the target function, the dominant computational burden is driven primarily by the population size and the number of generations.

Empirically, the population size m often has a greater effect on optimization accuracy than the number of generations. A natural way to improve precision is therefore to increase the population size. To illustrate the high-dimensional effect, consider the extreme case of a single generation search. Supposing $\boldsymbol{Z}$ is a $p \times 1$ parameter vector, the euclidean distance between the best sampled candidate $\widehat{\boldsymbol{Z}}_{GA}$ and the true optimizer $\widehat{\boldsymbol{Z}}$ is approximately Weibull distributed. We may write $||\widehat{\boldsymbol{Z}}_{GA} - \widehat{\boldsymbol{Z}}|| = \mathcal{O}_p(n^{-1/p})$, which implies the curse of high dimensionality. For fixed n, the approximation error deteriorates to 1 rapidly as p increases. Maintaining the same precision, therefore, requires

population size to grow exponentially with dimension, which is computationally infeasible in high-dimensional problems.

Considering a small n, parts of the parameter space are unlikely to be explored at all, making it easy for the algorithm to miss the global maximizer. This problem is further exacerbated by the mutation step, which is restricted to a randomly centered cube whose side length shrinks at rate $\mathcal{O}(m^{-1/2})$ in generation m, since shrinkage progressively intensifies the omission.

2.4 Improvement by Voronoi Partition

Recent application-oriented studies have used Voronoi tessellation primarily as a compact representation of spatial or structural design, with GA serving as an outer-loop search routine. A recent review of Voronoi structures in materials science (Almeida and Rios 2025) shows applications in lightweight automotive, aerospace components, biomedical implants, and mechanically efficient cellular structures, which also describes GA-based optimization of bi-dimensional and triple-dimensional Voronoi structures as an emerging topic. In this work line, the Voronoi construction usually maps a relatively small set of design variables, such as seed locations and seed regularity, to a high-dimensional physical geometry. The fitness of each geometry is then evaluated by finite-element analysis, computational fluid dynamics, or surrogate prediction. For example, GA is used to improve the mechanical response of (quasi-)three-dimensional Voronoi porous structures (Tung et al. 2023; Lin et al. 2024), while Asakawa et al. (2024) optimize stiffener layouts in composite panels by placing stiffeners along Voronoi edges. Fontalvo García et al. (2025) also apply it to determine the topology of Voronoi flat roofs with a varying number of seeds and structural member sizes, and Suzuki et al. (2025) employ Voronoi cellular heat sinks through a neural-network surrogate model. Similar ideas have also appeared in deployment and coverage design, where Voronoi cells are used to prevent clustering and enforce segment-wise coverage in a GA search over pseudo-elite locations (Xie et al. 2026). These studies show the practical value of the combination, but in most cases Voronoi tessellation acts mainly as a domain-specific geometry generator or feasibility constraint, while the GA operators themselves remain largely conventional.

From a methodological perspective, the literature is more limited and tends to treat Voronoi partitioning as a modular auxiliary device for exploration, sampling, or representation. The Voronoi-established GA constructs models so that the offspring distribution can adapt to the objective landscape (Shimosaka et al. 2004), and recent multi-objective learning work decomposes a high-dimensional design or preference space into Voronoi grids and applies GA to the grid-partitioning problem (Chen et al. 2025). In surrogate-assisted and adaptive sampling settings, PF-Voronoi sampling classifies regions containing predicted Pareto front points into Voronoi cells and selects new samples according to different criteria before applying Kriging (Wu et al. 2024). In spatial data analysis, Voronoi-type methods use Voronoi cells and convex hull volumes as density units and then applies GA to identify cell blocks with low density (Ghafour et al. 2025). Thus, existing methods typically assign a separate role to each component that Voronoi partitions define local neighborhoods and regions,

or geometric encodings, whereas GA supplies a generic global search engine, whose combination is more of multi-stage applications.

The present paper is to integrate the two components more tightly. Instead of using a Voronoi partition only for initialization, representation, or spatial segmentation, we introduce the Voronoi cell as parts of the evolutionary mechanism itself. The graph partition controls how elite regions are recombined in crossover and how mutation neighborhoods are selected and shrunk. Consequently, the graph partition directly influences the offspring distribution, mutation refinements, the optimization trajectory, and ultimately reflects on the GA solution.

Motivated by these observations, we propose to represent the parameter space through a collection of non-overlapping convex cells and to use the induced cell adjacency graph to guide both exploration and genetic operations. Related ideas for generating new candidates appear in Richter and Schicker (2017) and Tian et al. (2009). Although Voronoi partitions have considerable potential in global optimization, the literature on combining them with generally purpose optimization routines remains limited. For example, Shimosaka et al. (2004) used Voronoi diagrams within a GA-type algorithm primarily at the crossover stage, whereas Tung et al. (2023) combined Voronoi diagrams with GA to optimize mechanical properties of tension bearing structures. In this paper, we develop a flexible and general optimization framework that integrates Voronoi partitioning with GA. The proposed procedure could be viewed as a more systematic and broadly applicable variant of Basin hopping. Throughout, we assume that the objective function is continuous on $\mathcal{Z}$, since extending the approach to discrete valued $\boldsymbol{Z}$ would raise additional challenges and would not interact naturally with Voronoi partitioning.

The preceding discussion motivates an evolution operator that both retains high-fitness solutions and spreads exploration across geometrically distinct neighborhoods. Next section formalizes this idea through the proposed Voronoi-Elitism Genetic Algorithm.

3 Optimization with Genetic Algorithm

We propose a novel GA that improves crossover efficiency and robustness by incorporating geometric information extracted from high-quality solutions through a Voronoi-based partitioning. The proposed method develops from a GA framework and differs from conventional variants in its evolution operator. Its central innovation is a Voronoi based augmentation of the parent selection pool, which introduces structured local diversity during crossover while preserving emphasis on elite regions of the search space. To isolate the contribution of this design, we compare the proposed algorithm with two benchmark control variants as naive selection, denoted Control 0 (Random Mating) and Control 1 (Elite Weighted), which progressively incorporate standard mechanisms such as elitism and fitness-guided parent selection.

3.1 Evolution Framework

Starting from the initialized population, the algorithm proceeds iteratively by applying an evolution operator that updates the population from one generation to the next.

In Algorithm 2, the update is represented by a variant-specific EVOLVE procedure, which encapsulates parent selection, crossover, mutation, and any additional mechanisms. The same high-level control flow is used for all algorithmic variants. only the implementation of EVOLVE differs across methods.

Termination is determined by the relative improvement in the best fitness value between successive generations. Specifically, the algorithm stops when the best observed fitness fails to improve by more than a prespecified tolerance. In island-based implementations, the same criterion is applied to the best fitness across islands.

In the following subsections, we describe the evolution operators for Control 0, Control 1, and the proposed **Voronoi-Elitism GA (VEGA)**, emphasizing the incremental design changes that distinguish the three variants.

Control 0 implements a minimal GA that serves as a baseline for evaluating more structured variants. It employs random parent selection, single point crossover, and additive Gaussian mutation at a given mutation rate, with neither elitism nor fitness-guided sampling. Thus, beyond the common fitness evaluation shared by all variants, Control 0 contains no additional structural mechanisms. See Algorithm 3 for pseudo code.

Control 1 extends the baseline algorithm in Control 0 by incorporating another two GA mechanisms: elitist retention and fitness-guided parent selection. In each generation, a fixed fraction $r \in (0, 1)$ of the highest fitness individuals is copied directly into the next generation as elites. The remaining offspring are then generated from parents sampled randomly according to fitness, thereby preferring high-quality solutions while preserving randomness in the search process. See Algorithm 4 for pseudo code.

For Control 0, offspring are produced through single point crossover. Mutation is applied independently to each gene at a fixed mutation rate. Unlike in Control 0, however, the mutation magnitude is scaled by the difference between the two parent solutions and is bounded below by a minimum mutation threshold. The construction preserves a non-trivial exploratory step even when the selected parents are close to one another. After mutation, each offspring is projected back to $[0, 1]$ by coordinate-wise clipping.

3.2 Voronoi-Elitism Genetic Algorithm

The proposed method extends Control 1 by modifying the parent-selection mechanism used during crossover. For Control 1, a given elite fraction $r \in (0, 1)$ is retained unchanged at each generation, and the remaining individuals are generated via crossover and mutation. The distinguishing feature of the proposed algorithm is the construction of an augmented parent pool, in which each elite solution is associated with locally generated candidates that reflect the geometric arrangement of elite points in the search space. This augmentation promotes structured local exploration while preserving emphasis on high-quality regions.

The Voronoi-based augmentation is implemented in a coordinate-wise, axis-aligned manner. For each dimension, the elite values are sorted, and adjacent midpoints are used to define interval boundaries. The Cartesian product of these intervals yields a hyper-rectangular region that approximates the local Voronoi cell associated with a

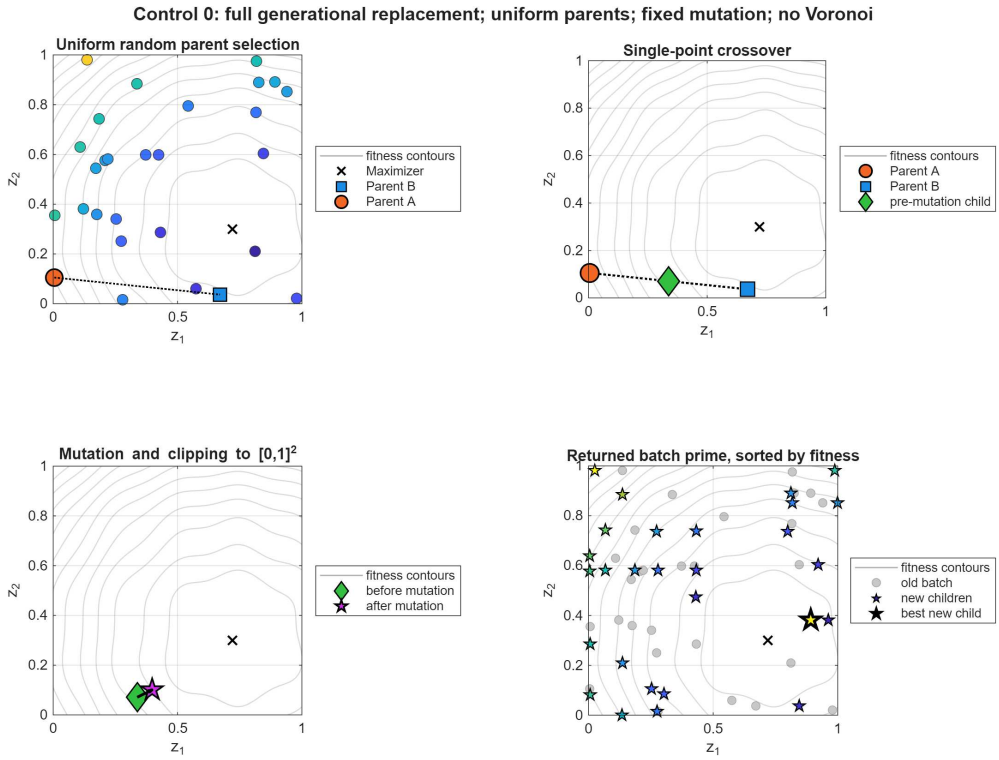


Fig. 2: Control 0 evolution operator: parents are sampled uniformly from the current batch, recombined by single-point crossover, mutated with fixed-probability Gaussian noise, and returned as a fully replaced offspring batch sorted by fitness, with no Voronoi-based selection.

given elite point. Notice that the exact Voronoi cells is also applicable, but it could be computationally expensive in high dimensional scenarios, including boundary identification and in augmentation generation. See Algorithm 5 and Figures C1 and C2 for exact cases. To be convenient under limited budget, we focus on the rectangular approximation. A small number of candidate vectors is then sampled uniformly from this region. The first sampled candidate with finite fitness is retained. if no feasible candidate is obtained, the elite point itself is carried forward as a fallback.

The local candidates generated from the Voronoi construction are used to expand the parent pool for crossover. Specifically, reproduction draws from both the retained elites and their associated local samples, with parent selection with fitness-based probabilities. Offspring can inherit information from either the original elite solutions or nearby structured perturbations, which improves exploratory coverage preserving exploitation of high-fitness regions.

Voronoi construction is particularly robust in high dimensional settings, where a finite population often provides poor coverage of the parameter space, as random mutation with shrinkage may leave large regions unexplored. By generating local candidates

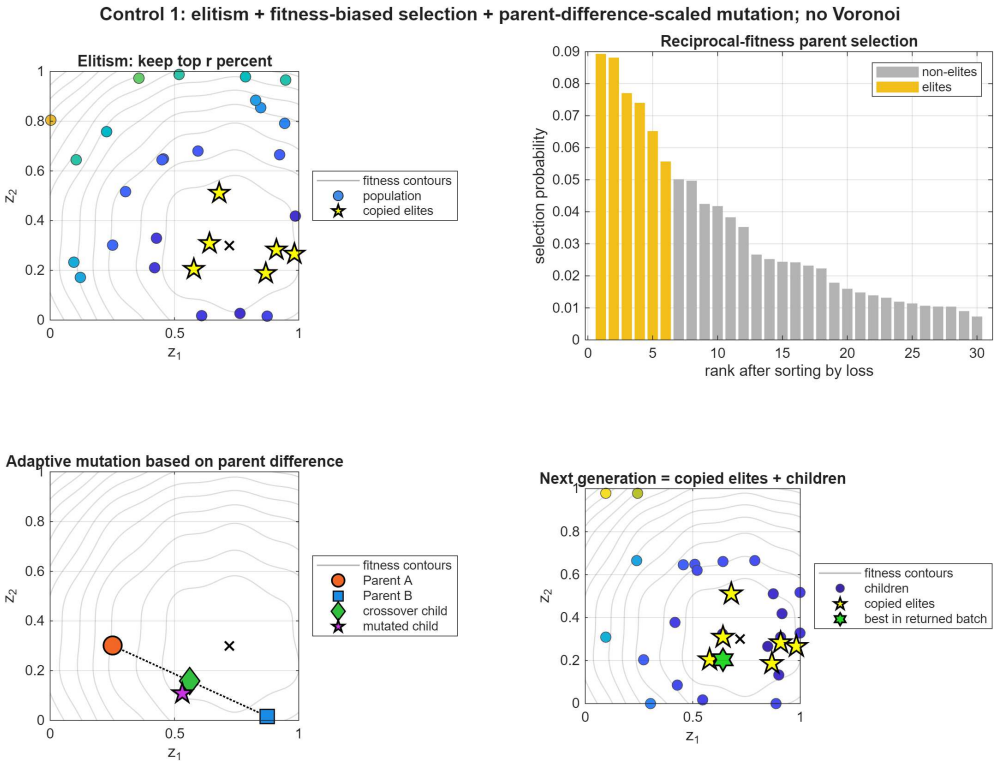


Fig. 3: Control 1 evolution operator: top r(%) elites are copied, while remaining children are generated from reciprocal-fitness-biased parents using single-point crossover and parent-difference-scaled mutation. The returned batch combines elites and children, sorted by fitness, without Voronoi selection.

from non-overlapping neighborhoods associated with distinct elite solutions, VEGA distributes exploratory effort systematically across promising parts and unexplored parts of the domain. In this sense, the method mitigates the tendency of the population to concentrate around a small subset of elites, but preserve potential geometric diversity in unexplored areas during crossover. See Algorithm 6 for pseudo code.

Next section discusses the theoretical performance of VEGA under limited budget. Section 5 evaluates VEGA through simulation studies and comparative experiments against Control 0 and Control 1.

4 Theoretical Results

This section studies the convergence behavior of genetic algorithms under high dimensional settings. We analyze how population size, crossover, mutation, and the Voronoi mechanism affects the attainable distance. These results provide justification for the efficiency of VEGA under limited computational budgets, without requiring strong structural assumptions on the target function, while the asymptotic convergence

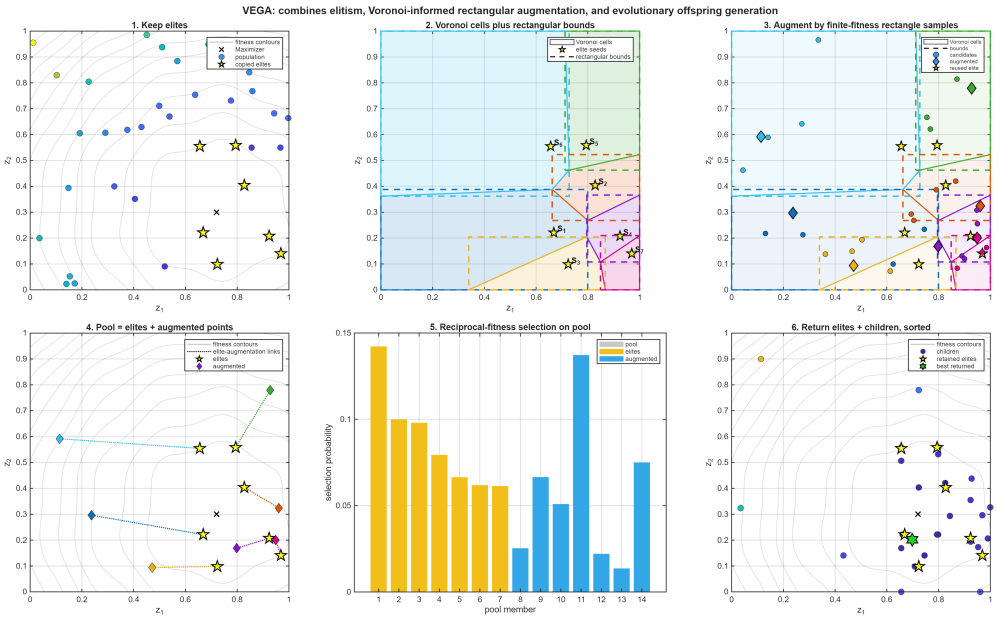


Fig. 4: VEGA evolution operator: top elites are retained, each elite's Voronoi-informed rectangular region is sampled to add finite-fitness augmented points, and elites plus augmented points form a fitness-biased parent pool. Offspring are generated by crossover and adaptive mutation, then returned with elites sorted by fitness.

of genetic algorithms is guaranteed by random perturbations theory (Freidlin and Wentzell 2012) using unlimited computational time (Cerf 1998).

We begin with the initialization stage of a standard GA. Since the initial population is typically sampled uniformly from the search domain, the first step is to understand the geometric behavior of a single uniformly sampled chromosome in high dimension.

Suppose the high dimensional search area is $\boldsymbol{S} = [0,1]^p$, and the chromosomes are sampled uniformly from $\boldsymbol{S}$. Consider the simplest case of single chromosomes $\boldsymbol{X} = (X_1, X_2, \cdots, X_p) \sim \text{Unif}(\boldsymbol{S})$. Notate the true maximizer as $\boldsymbol{a} = (a_1, a_2, \cdots, a_p) \in \boldsymbol{S}$ and the square distance between $\boldsymbol{X}$ and $\boldsymbol{a}$ could be denoted as $R^2 = \|X - a\|^2 = \sum_{i=1}^{p} (X_i - a_i)^2$.

Lemma 1. *Let* $\boldsymbol{X} = (X_1, X_2, \cdots, X_p) \sim \text{Unif}(\boldsymbol{S})$*, where* $\boldsymbol{S} = [0,1]^p$ *is a high dimensional space, and* $\boldsymbol{a} = (a_1, a_2, \cdots, a_p) \subset \boldsymbol{S}$ *but* $\boldsymbol{a} \notin \partial\boldsymbol{S}$*. Then* $\boldsymbol{X}$ *is asymptotically normal near a spherical shell centered at* $\boldsymbol{a}$ *with radius* $r_p > 0$*. Further for some* $\sigma_0 > 0$*,*

$$p^{-1}\|\boldsymbol{X} - \boldsymbol{a}\| \sim \mathcal{N}\left(r_p, \frac{\sigma_0^2}{p}\right),$$

where $r_p = p^{-1}\sqrt{\sum_{i=1}^{p}\left[\frac{1}{12} + \left(a_i - \frac{1}{2}\right)^2\right]}$.

Lemma 1 shows that in high dimensions, a uniformly sampled chromosome is not spread evenly in terms of distance from $\boldsymbol{a}$. Instead, its distance concentrates around a thin spherical shell. This concentration provides a baseline for analyzing the initial performance of GA that, before selection or genetic operations are applied, most individuals are expected to lie at a similar distance from the true maximizer.

We, afterward, move from a single chromosome to an initial population of size n. Since selection in GA favors individuals with higher fitness, an idealized first generation estimator can be represented by the individual closest to the maximizer. Suppose it initially consists of n individuals $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$, consisting the GA estimator at the first generation $\boldsymbol{X}_i$, for some $i = \text{argmin}_j \, ||\boldsymbol{X}_j - \boldsymbol{a}||^2$.

Theorem 1. *By notations and definitions in Lemma 1, let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n \overset{i.i.d}{\sim} \text{Unif}(\boldsymbol{S})$. *The distance between sample to* $\boldsymbol{a}$, $R_i = \|\boldsymbol{X}_i - \boldsymbol{a}\|$, *then the minimum distance satisfies*

$$R_{min} = r_p - \sigma_p \sqrt{2 \log n} + \mathcal{O}\left(\frac{\log \log n}{\sqrt{\log n}}\right) + \mathcal{O}_p(\log^{-\frac{1}{2}} n).$$

In Theorem 1, the negative term $-\sigma_p \sqrt{2 \log n}$ captures the benefit of increasing the population size. As n grows, the best individual among the initial samples is expected to be closer to $\boldsymbol{a}$, with the improvement occurring at the order of $\sqrt{\log n}$ under Gaussian approximation.

The previous result is derived for the unit hypercube. To extend the analysis beyond $\boldsymbol{S} = [0,1]^p$, we consider a more general high dimensional domain and describe the distance behavior in terms of moment bounds. Assume that the components of individual $\boldsymbol{X}$ satisfy $\mathbb{E}X_i^2 = \mathcal{O}(S^2)$, $\mathbb{E}X_i^4 = \mathcal{O}(B)$, with $\mathcal{O}(B) = \mathcal{O}(S^4)$. Hence, R^2 has expectation in (4) and variance in (5).

$$\mathbb{E}R^2 = \mathcal{O}(pS^2) + \sum_{i=1}^{p} (a_i - EX_i)^2 = \mathcal{O}\left(pS^2\right) \tag{4}$$

$$\text{Var}(R^2) = \mathcal{O}\left(pB + pS^4\right) \tag{5}$$

Theorem 2. *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n \overset{i.i.d}{\sim} \boldsymbol{X}$ *for some uniform population* $\boldsymbol{X} = (X_1, X_2, \cdots, X_p)$ *on a given non-degenerate high dimensional domain* $\boldsymbol{K}$, *with* $p^{-1}\|\mathbb{E}\boldsymbol{X}\|$, $p^{-1}\|\mathbb{E}\boldsymbol{X}^2\| = \mathcal{O}(S^2)$, *and* $p^{-1}\|\mathbb{E}\boldsymbol{X}^4\| = \mathcal{O}(B)$ *not relying on* p. *The distance between sample to some inner point* $\boldsymbol{a} \subset \boldsymbol{K}$ *but* $\boldsymbol{a} \notin \partial\boldsymbol{K}$ *is* $R_i = \|\boldsymbol{X}_i - \boldsymbol{a}\|$, *then the minimum distance satisfies*

$$\mathbb{E}R_{min} = \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| - \mathcal{O}(S^2 \log^{\frac{1}{2}} n) + \mathcal{O}_p(S \log^{-\frac{1}{2}} n).$$

Having established the effect of population size at initialization, we further examine how crossover changes the distance distribution. We consider the simple arithmetic crossover operator $(X_1^{(2)}, X_2^{(2)}, \cdots, X_p^{(2)}) = \boldsymbol{X}^{(2)} = (\boldsymbol{X}_1 + \boldsymbol{X}_2)/2$, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_1$ are two independently sampled parent chromosomes. Therefore, the distance is defined as $R^2 = ||\boldsymbol{X}^{(2)} - \boldsymbol{a}||^2$. Similarly, its expectation and variance are

$$\mathbb{E}R^2 = \frac{p}{24} + \sum_{i=1}^{p}\left(a_i - \frac{1}{2}\right)^2 = \sum_{i=1}^{p}\left(a_i - \frac{1}{2}\right)^2 + \mathcal{O}\left(pS^2\right), \tag{6}$$

and

$$\mathrm{Var}(R^2) = \sum_{i=1}^{p}\left\{E\left(X_i^{(2)} - a_i\right)^4 - \left[\mathbb{E}\left(X_i^{(2)} - a_i\right)^2\right]^2\right\} = \mathcal{O}\left(pB + pS^4\right), \tag{7}$$

Similar to Theorem 1, the minimum distance between GA estimator at the next generation and maximizer satisfies

$$\begin{aligned} R_{min} =& \mu_p + \sigma_p Z_{(1)} \\ =& \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| + \mathcal{O}\left(p^{\frac{1}{2}}S\right) - \mathcal{O}\left(p^{\frac{1}{2}}(B+S^4)^{\frac{1}{2}}\log^{\frac{1}{2}} n\right) \\ &+ \mathcal{O}_p\left((B+S^4)^{\frac{1}{4}}\log^{-\frac{1}{2}} n\right) \\ =& \sqrt{\sum_{i=1}^{p}\left(a_i - \frac{1}{2}\right)^2} - \mathcal{O}\left(S^2 p^{\frac{1}{2}}\log^{\frac{1}{2}} n\right) + \mathcal{O}_p\left(S\log^{-\frac{1}{2}} n\right). \end{aligned} \tag{8}$$

We then incorporate shrinking mutation. At generation m, component-wisely independent mutation is modeled as an additive random perturbation $\boldsymbol{Z} = (Z_1, Z_2, \cdots, Z_p) = \boldsymbol{X} + \boldsymbol{W} = (X_1 + W_1, X_2 + W_2, \cdots, X_p + W_p)$, where $\boldsymbol{W} = (W_1, W_2, \cdots, W_p)$ is independent of $\boldsymbol{X}$, satisfying $\mathbb{E}W_i = 0$, $\mathrm{Var}\,(W_i) = \mathcal{O}\left(m^{-1}\right)$, and $\mathbb{E}W_i^4 = \mathcal{O}\left(m^{-2}\right)$. This formulation captures the common practice of reducing the mutation magnitude as the number of generations increases.

Then distance is defined as $R^2 = \|\boldsymbol{Z} - \boldsymbol{a}\|^2 = \|\boldsymbol{X} + \boldsymbol{W} - \boldsymbol{a}\|^2 = \sum_{i=1}^{p}\left(X_i - a_i + W_i\right)^2$, with expectation and variance

$$\begin{aligned} \mathbb{E}R^2 &= \mathbb{E}\sum_{i=1}^{p}\left(X_i^2 + a_i^2 + W_i^2 - 2a_iX_i - 2a_iW_i + 2X_iW_i\right) \\ &= \sum_{i=1}^{p} Var\left(X_i\right) + \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\|^2 + \mathbb{E}W_i^2 \\ &= \sum_{i=1}^{p}\left(a_i - \frac{1}{2}\right)^2 + \mathcal{O}\left(pS^2 + \frac{p}{m}\right) \end{aligned} \tag{9}$$

$$\begin{aligned}\mathrm{Var}R^2 &= \sum_{i=1}^{p}\left\{\mathbb{E}\left(X_i+W_i-a_i\right)^4-\left[\mathbb{E}\left(X_i+W_i-a_i\right)^2\right]^2\right\}\\ &= \mathcal{O}\left(p\left(S^4+\frac{1}{m}\right)\right).\end{aligned} \tag{10}$$

Using arguments in previous proofs, the minimum of distance satisfies

$$\begin{aligned}R_{min} =&\mu_p+\sigma_p Z_{(1)}\\ =&\|\mathbb{E}\boldsymbol{X}-\boldsymbol{a}\|+\mathcal{O}\left((pS^2+pm^{-1})^{\frac{1}{2}}\right)-\mathcal{O}\left(p^{\frac{1}{2}}(S^4+m^{-1})^{\frac{1}{2}}\log^{\frac{1}{2}}n\right)\\ &+\mathcal{O}_p\left((S^4+m^{-1})^{\frac{1}{4}}\log^{-\frac{1}{2}}n\right)\\ =&\sqrt{\sum_{i=1}^{p}\left(a_i-\frac{1}{2}\right)^2}-\mathcal{O}\left((S^2+m^{-\frac{1}{2}})p^{\frac{1}{2}}\log^{\frac{1}{2}}n\right)\\ &+\mathcal{O}_p\left((S+m^{-\frac{1}{4}})\log^{-\frac{1}{2}}n\right).\end{aligned} \tag{11}$$

In shrinkage mutation implement, mutation errors are accumulative. It is easy to have

$$S^2=\mathcal{O}\left(\sum_{i=1}^{m}\frac{1}{i}\right)=\mathcal{O}(\log m). \tag{12}$$

Hence

$$R_{min}=\sqrt{\sum_{i=1}^{p}\left(a_i-\frac{1}{2}\right)^2}-\mathcal{O}(p^{\frac{1}{2}}\log^{\frac{1}{2}}m\log^{\frac{1}{2}}n)+\mathcal{O}_p(\log^{\frac{1}{2}}n\log^{\frac{1}{2}}m). \tag{13}$$

Equation (13) suggests that the accumulated effect of shrinking mutation can enlarge the search dispersion over generations. In high dimensions, this additional dispersion may improve the chance of producing individuals closer to the maximizer, provided that the mutation scale is controlled so that the approximation remains valid under limited budget.

To analyze the Voronoi mechanism used in VEGA, we introduce order notation and a geometric bound for uniform samples over a high-dimensional subset of the unit cube. We use the notation $A \lesssim B$ to show that $A \le cB$ for some universal constant $c > 0$. We also denote $A \asymp B$ if $A \lesssim B$ and $B \lesssim A$.

Theorem 3 (Order bounds)**.** *Let* $\boldsymbol{K} \subseteq [0,1]^p$ *with* $|\boldsymbol{K}| = v$*, where* $0 < v \le 1$*, and let* $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$*. Then*

$$\sqrt{p}\, v^{1/p} \lesssim \mathbb{E}\|\boldsymbol{X}\| \lesssim \sqrt{p},$$

and

$$\mathrm{Var}\|\boldsymbol{X}\| \lesssim \mathcal{O}(p) - \mathcal{O}\Big(pv^{1/p}\Big) - \mathcal{O}\Big(pv^{2/p}\Big).$$

Finally, we incorporate the Voronoi partition strategy of VEGA. Unlike standard GA, which relies only on genetic operations within the existing population, VEGA partitions the feasible domain and encourages exploration across Voronoi cells. This mechanism effectively increases spatial coverage, which can improve the order of the best attainable distance in high dimensions.

Theorem 4. *$\boldsymbol{K} \subset [0,1]^p$ is a non-degenerate high-dimensional convex set with volume $|\boldsymbol{K}| \in [0,1]$, and samples $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n \overset{i.i.d}{\sim} \mathrm{Unif}(K) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \subset \mathbb{R}^p$ is an independent random variable with $\|\mathbb{E}\boldsymbol{\varepsilon}\| = 0$ and $\|\mathrm{Var}(\boldsymbol{\varepsilon})\| = \mathcal{O}(pg(m))$. Let $\boldsymbol{C}_1(\boldsymbol{K}), \boldsymbol{C}_2(\boldsymbol{K}), \ldots, \boldsymbol{C}_{\lfloor rn \rfloor}(\boldsymbol{K})$ is Voronoi Partition of $\boldsymbol{K}$, which is applied to VEGA in Algorithm 5. Then the minimum distance of m-generation VEGA estimator $\boldsymbol{X}^{(m)}$ to true value $\boldsymbol{a} \notin \partial \boldsymbol{K}$, for some small ξ, $R_{min}^{(m)} = \|\boldsymbol{X}^{(m)} - \boldsymbol{a}\|$ satisfies*

$$R_{min}^{(m)} \lesssim \mathcal{O}\left(p^{\frac{1}{2}} n^{-\frac{(1+\xi)}{p}} - p^{\frac{1}{2}} g(m) \log^{\frac{1}{2}}(n + mrn)\right) + \mathcal{O}_p\left(g^{\frac{1}{2}}(m) \log^{-\frac{1}{2}}(n + mrn)\right).$$

Theorem 4 suggests that the Voronoi mechanism improves the spatial coverage of the search domain and can yield a sharper order of the best distance based only on crossover or shrinking mutation. This advantage is particularly relevant high dimensional settings under limited budget, where increasing the population size or the number of generations alone may be insufficient to overcome distance concentration.

5 Simulation Studies: Segmented Regression

VEGA is general and can be applied to a wide range of optimization problems through the specification of an appropriate fitness function and, when necessary, a problem-specific population representation. In this work, we instantiate the algorithm for segmented regression by defining a custom fitness function and a population initialization procedure that enforces the ordering constraints required for breakpoint representations.

5.1 Data Generation and Simulation Setting

Synthetic Data Generation.

Given K segments, we generate and design a predefined linear model piece-wisely, which is well-conditioned and non-degenerate. Breakpoint locations are sampled randomly over the domain subject to ordering and minimum separation constraints, ensuring that all segments have comparable width to each other.

Specific segment slopes and intercepts are then generated sequentially. The parameters for the first segment are drawn at random, and parameters for following segments are sampled with a minimum difference from the preceding segment. This constraint

enforces a clear structural change at each breakpoint, either through a change in slope or through a discontinuity in the fitted value at the segment boundary. As a result, adjacent segments are guaranteed to be meaningfully distinct, avoiding cases in which breakpoints correspond to only negligible changes in the underlying signal.

Given the resulting segmentation and segment parameters, the response values are generated according to a piecewise linear model with additive Gaussian noise. Specifically, for each observation i,

$$y_i = \theta_{s(i)}\, x_i + \alpha_{s(i)} + \varepsilon_i, \qquad \varepsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2),$$

where x_i denotes the covariates, y_i the corresponding response, and $s(i) \in \{1, \ldots, K\}$ the segment index determined by the true breakpoints. The parameters $\theta_{s(i)}$ and $\alpha_{s(i)}$ denote the slope and intercept of segment $s(i)$, respectively.

Population Initialization.

All GA candidates share the same population initialization procedure. Each individual is represented by a sorted breakpoint vector $\boldsymbol{Z} \in [0,1]^{K-1}$. Initial candidates are generated by uniformly sampling breakpoint values in $[0,1]$, and afterward evaluated and sorted according to the fitness of segmentation results. Candidates with infinite fitness are dropped, and sampling is repeated until the desired population size is reached. This rejection-driven initialization ensures that all individuals in the initial population correspond to valid segmentations while introducing no additional structural bias.

Fitness Evaluation.

Candidate solutions are encoded as breakpoint vectors $Z \in [0,1]^{K-1}$, which induces K contiguous segments by partitioning the input variable X. For given $\boldsymbol{Z}$, segment boundaries are first determined, and the data are divided accordingly into specific segment subsets $(\boldsymbol{X}_s, \boldsymbol{Y}_s)$ for $s = 1, \ldots, K$. Fitness is computed by fitting an independent Ordinary Least Squares (OLS) linear model within each segment and aggregating the squared residuals. If any segment is empty or fails to meet minimum support requirements, the candidate solution is deemed invalid and assigned with infinite fitness. For valid segmentations, the total loss is calculated as the sum of squared residuals across all segments, and the final fitness value is obtained by normalizing this aggregated loss by the sample size. This fitness definition is used consistently across all algorithmic variants. See Algorithm 1 for pseudo code.

5.2 Simulation Results

We evaluate the VEGA by comparing it with Control 0 and Control 1. Although the algorithm itself is general, all experiments are conducted in a segmented regression setting, which provides a clear and structured context for comparing optimization behavior.

Algorithm performance is evaluated under three criteria. First, we assess prediction accuracy by Mean Squared Error (MSE) given known number of segments. Second, we examine breakpoint recovery to evaluate how accurately each method identifies

Algorithm 1 Fitness Evaluation

procedure FITNESS(X, y, Z, K)
  Compute segment index boundaries induced by Z on sorted X
  loss $\leftarrow 0$
  for $s = 1$ to K **do**
    Extract segment (X_s, y_s)
    if segment is empty or too narrow **then**
      return ∞
    end if
    Fit (β_s, α_s) on (X_s, y_s) by OLS
    loss $\leftarrow$ loss $+ \sum_{(x,y)\in(X_s,y_s)} (\beta_s x + \alpha_s - y)^2$
  end for
  return loss$/|X|$
end procedure

the underlying segmentation structure under simplified conditions. Finally, we study model selection performance by Bayesian Information Criterion (BIC), assuming the number of segments is unknown, with particular emphasis on the ability of proposed algorithm to recover the correct model complexity.

5.2.1 MSE Performance

Prediction performance is evaluated using synthetic data generated from piecewise linear regression models with known segmentation structure. Experiments are conducted for segment counts $K \in \{4, 8\}$, segment lengths in $\{100, 300, 1000\}$, and noise levels $\sigma \in \{0.1, 0.2, 0.4\}$. For each configuration (K, n, σ), breakpoint locations and corresponding segment slopes and intercepts are randomly generated and held fixed, defining the underlying signal. Covariates are taken on an evenly spaced grid over $[0, 1]$, with total sample size $n = K \times$ segment length.

For each simulation replicate, Gaussian noise is independently generated and added to the fixed signal to obtain observed responses. Given the true segmentation, segment-wise regression parameters are refit via ordinary least squares (OLS) to account for the variability introduced by noise, providing a baseline corresponding to the best achievable fit under the true model.

Prediction performance is evaluated using mean squared error (MSE). For each candidate model, breakpoint locations are estimated by Control 0, Control 1, and VEGA under Simple and Island settings, and segment-wise parameters are subsequently refit via OLS given the estimated segmentation. For all GA-based methods, we use a population size of $50K$, a maximum of $2K$ iterations, and an island model with 5 islands, where crossover between islands occurs every 2 generations for $K = 4$ and every 4 generations for $K = 8$. See Table 1 for results.

5.2.2 Breakpoint Recovery

To evaluate the ability of each algorithm to recover structural information beyond prediction accuracy, we conduct breakpoint recovery experiments in a simplified setting

Table 1: Average Bias and MSE for $K = 4, 8$ across segment lengths (l) and noise levels (σ).

			Control 0		Control 1		VEGA	
K	l	σ	Bias$\times 10^3$	MSE$\times 10$	Bias$\times 10^3$	MSE$\times 10$	Bias$\times 10^3$	MSE$\times 10$
Simple GA								
4	100	0.1	30.054	0.399	4.799	0.146	0.184	0.100
		0.2	29.922	0.692	6.419	0.457	0.111	0.394
		0.4	27.515	1.844	7.513	1.644	-0.381	1.565
	300	0.1	40.669	0.506	6.786	0.167	0.864	0.108
		0.2	39.797	0.796	8.778	0.486	0.694	0.405
		0.4	39.038	1.979	10.794	1.696	0.663	1.595
	1000	0.1	30.503	0.405	5.459	0.155	0.377	0.104
		0.2	30.294	0.703	7.500	0.475	0.359	0.403
		0.4	29.884	1.896	9.573	1.693	0.274	1.600
8	100	0.1	56.867	0.666	11.805	0.216	1.702	0.115
		0.2	56.689	0.958	13.190	0.523	1.668	0.408
		0.4	55.164	2.117	14.744	1.712	0.903	1.574
	300	0.1	52.762	0.627	14.386	0.243	3.126	0.131
		0.2	52.668	0.924	16.484	0.562	3.185	0.429
		0.4	52.330	2.114	18.611	1.777	2.748	1.618
	1000	0.1	49.616	0.596	12.902	0.229	3.671	0.137
		0.2	49.040	0.889	14.407	0.543	3.695	0.436
		0.4	49.551	2.092	16.579	1.762	3.528	1.632
Island GA								
4	100	0.1	30.388	0.402	4.682	0.145	0.201	0.100
		0.2	29.782	0.689	6.555	0.456	0.130	0.392
		0.4	27.794	1.845	7.773	1.645	-0.376	1.563
	300	0.1	39.455	0.494	6.772	0.167	0.788	0.107
		0.2	39.638	0.793	8.320	0.480	0.996	0.407
		0.4	38.873	1.977	10.601	1.695	0.476	1.593
	1000	0.1	30.476	0.404	5.544	0.155	0.402	0.104
		0.2	30.457	0.704	7.166	0.471	0.349	0.403
		0.4	29.847	1.895	9.322	1.690	0.310	1.600
8	100	0.1	56.323	0.661	11.574	0.214	1.796	0.116
		0.2	56.299	0.955	13.523	0.527	1.780	0.410
		0.4	55.112	2.117	14.785	1.713	0.876	1.574
	300	0.1	52.447	0.624	14.728	0.246	3.158	0.131
		0.2	52.862	0.926	16.495	0.563	3.319	0.431
		0.4	52.220	2.112	18.551	1.775	2.827	1.618
	1000	0.1	49.495	0.595	12.958	0.229	3.554	0.135
		0.2	49.441	0.893	14.555	0.545	3.601	0.435
		0.4	49.229	2.089	16.770	1.765	3.513	1.632

with a single interior breakpoint. The number of segments is fixed at $K = 2$, with the true breakpoint located at 0.5. This setup isolates the task of breakpoint estimation and allows direct comparison between estimated and true breakpoint locations.

Segment slopes and intercepts are independently sampled from the uniform distribution on $(-1, 1)$ and held fixed across all simulation replicates. To ensure identifiability, the absolute difference between the slopes of the two segments is constrained to be at least 0.5, guaranteeing a meaningful structural change at the breakpoint. Covariates are generated on an evenly spaced grid over $[0, 1]$, with sample sizes $n \in \{1000, 2000, 4000\}$. Observed responses are obtained by evaluating the corresponding piecewise linear model and adding i.i.d. Gaussian noise with standard deviation $\sigma \in \{0.1, 0.2, 0.4\}$. For each configuration, 1,000 independent simulation replicates are generated.

For model fitting, all methods are initialized with a population of size 50, with candidate breakpoint locations sampled uniformly from $\{0, 1\}$ to mitigate the effects of random initialization in this low-dimensional setting. Each algorithm is run for a maximum of 10 iterations, and for island-based methods, crossover between islands occurs every 3 generations. Breakpoint estimates are obtained from Control 0, Control 1, and VEGA under both Simple and Island configurations. See Table 2 for results and Figure 5 for breakpoint recovery distribution, which aligns with the theoretical results about non-standard asymptotic distribution of breakpoint estimators in Ling (2016); Ma and Yau (2016).

6 Real Data Analysis

We apply VEGA to real-world data from Stack Exchange to study long-term activity patterns and structural changes over time. The dataset consists of post and vote counts aggregated over approximately 5.5 years, from February 8, 2018 to July 23, 2023. We focus on three primary measures of platform engagement:

- **Questions**: Number of questions posted per trading day,
- **Answers**: Number of answers posted per trading day, and
- **Accept Votes**: Number of accepted-answer votes per trading day, where an accepted-answer vote corresponds to a question author marking an answer as accepted.

These variables capture complementary aspects of content generation and resolution dynamics on the platform.

Daily activity exhibits substantial variability; however, all model fitting and breakpoint estimation are performed directly on the original data. We fit segmented regression models using each genetic-algorithm variant for candidate segment counts ranging from $K = 2$ to 10. For each value of K, the algorithm is run five times with different random initializations, and the solution with the lowest mean squared error (MSE) is retained.

Model selection is conducted using the Bayesian Information Criterion (BIC), and the optimal number of segments is chosen by minimizing the BIC value. Across all three measures of platform engagement, the selected model consistently yields $K = 4$

Table 2: SD and MSE of breakpoint estimates across segment lengths l and noise levels σ, computed relative to the true breakpoint $\tau = 0.5$.

		Control 0		Control 1		VEGA	
l	σ	SD×10	MSE×10^2	SD×10	MSE×10^2	SD×10	MSE×10^2
Simple GA							
1000	0.1	3.08	10.21	0.89	0.80	0.14	0.02
	0.2	3.03	9.76	0.75	0.56	0.33	0.11
	0.4	3.18	10.30	0.74	0.55	0.57	0.32
2000	0.1	3.08	9.90	0.98	0.96	0.15	0.02
	0.2	3.06	9.99	0.77	0.60	0.18	0.03
	0.4	3.06	9.65	0.69	0.47	0.32	0.10
4000	0.1	3.14	10.14	0.96	0.92	0.10	0.01
	0.2	3.06	9.85	0.75	0.57	0.15	0.02
	0.4	3.03	9.65	0.66	0.43	0.28	0.08
Island GA							
1000	0.1	3.11	9.92	0.92	0.85	0.16	0.03
	0.2	3.10	9.98	0.84	0.70	0.27	0.08
	0.4	3.14	10.23	0.78	0.61	0.55	0.30
2000	0.1	3.12	10.10	0.91	0.83	0.13	0.02
	0.2	3.09	9.88	0.79	0.63	0.19	0.04
	0.4	3.05	9.82	0.70	0.50	0.37	0.14
4000	0.1	3.08	10.06	0.96	0.92	0.10	0.01
	0.2	3.18	10.54	0.81	0.66	0.17	0.03
	0.4	3.12	10.15	0.60	0.36	0.27	0.07

segments, indicating the presence of four major regimes in the underlying activity patterns.

6.1 Residual Bootstrapping

To assess the stability and uncertainty of the estimated breakpoints on real data, we apply a residual-based bootstrap that preserves the segmented structure in the fitted model. After selecting the best model and its breakpoints using BIC on the training series, we compute segment-wise fitted values and residuals from the original data. Bootstrap datasets are then generated by independently shuffling the residuals within each fitted segment and adding them back to the concatenated fitted values, thereby retaining the estimated segment shapes while resampling only the unexplained variation. For each bootstrap replicate (5000 total), we re-run the genetic algorithm with the same number of segments K to obtain a new set of breakpoint estimates. The resulting collection of bootstrap breakpoint vectors is used to evaluate stability and uncertainty by examining the empirical distribution of breakpoint locations, constructing percentile-based confidence intervals, and measuring the fraction of replicates that place a breakpoint within a small window of the original estimate, reflecting how consistently the algorithm recovers similar segmentation structure.

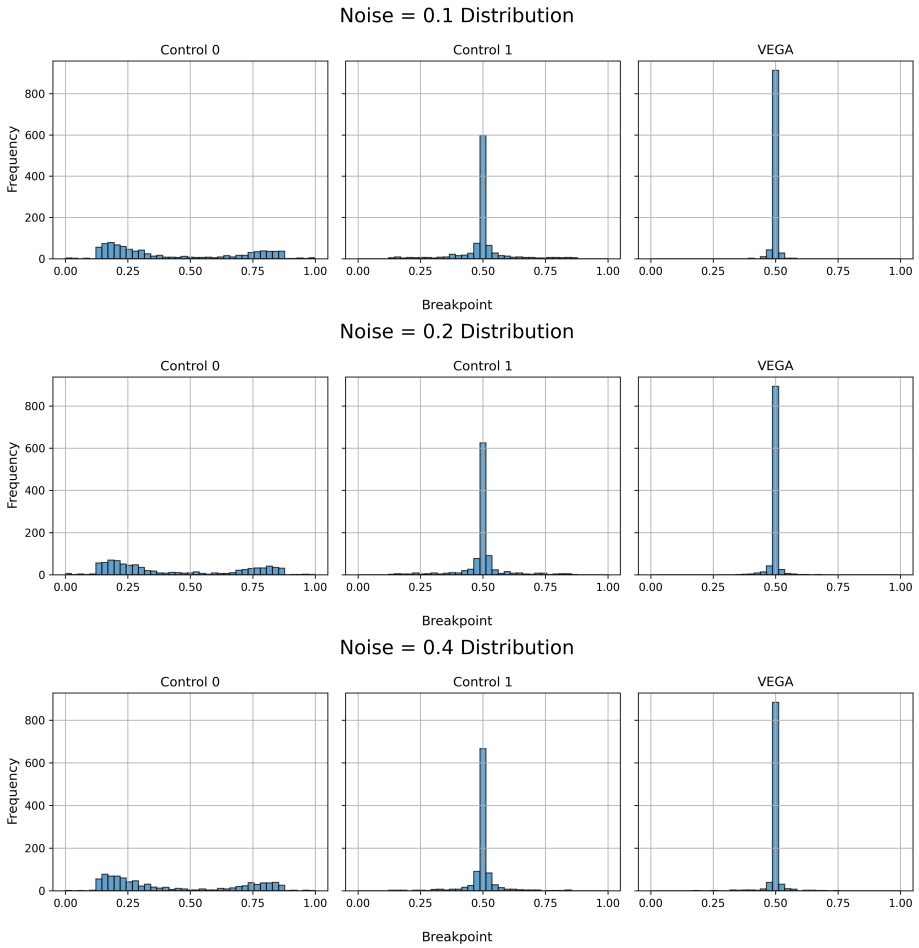


Fig. 5: Breakpoint distribution for the three models at $l = 4000$ and $\sigma = 0.1, 0.2, 0.4$

The segmentation results indicate several clear shifts in user activity over time. For questions, four regimes emerge: February 2018-December 2019, December 2019-May 2020, May 2020-October 2022, and October 2022-July 2023. The change at the end of 2019 is followed shortly by the global disruption associated with the COVID-19 pandemic, and the additional transition in May 2020 coincides with the period when remote work and online learning became widespread. These changes likely influenced both the volume and nature of mathematical engagement on the platform. A further structural break in October 2022 appears immediately prior to the public release of ChatGPT (November 2022), suggesting that the emergence of large language models may have altered user behavior in a sustained way.

A similar pattern is visible in the answers and accepted-answer vote series. Both show a pronounced shift in late March 2020, consistent with the early pandemic period, followed by a relatively stable regime through much of 2021 and 2022. All three

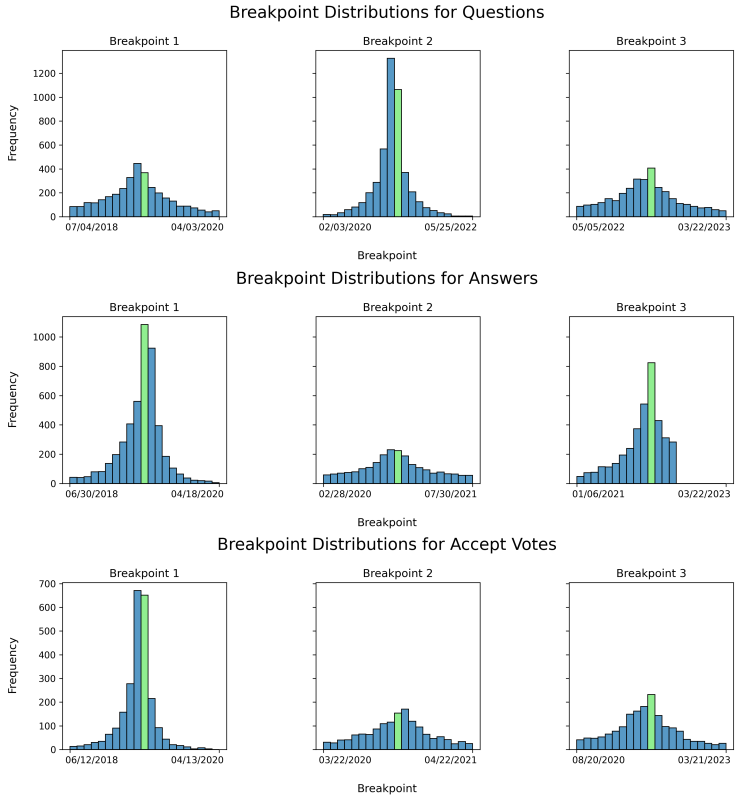


Fig. 6: Stack Exchange: Bootstrapping results for Questions (top), Answers (middle), and Accept Votes (bottom) breakpoint estimate highlighted in lightgreen.

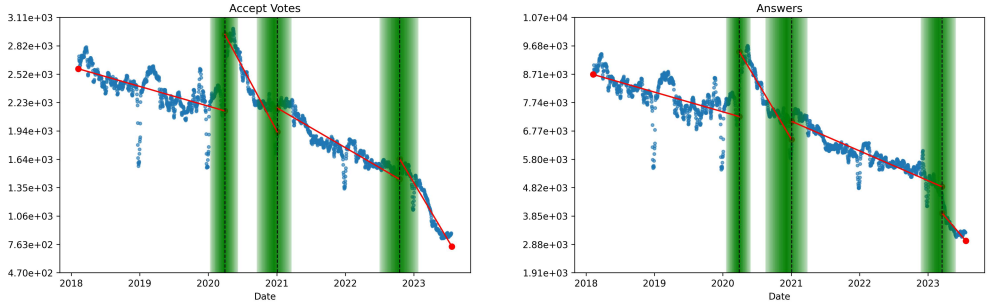


Fig. 7: Stack Exchange: Time series plots for Accept Votes (left) and Answers (right) with estimated mean function in red, detected breakpoints as dotted black line, and 95% confidence intervals in the green shaded region.

measures—questions, answers, and accepted votes—identify a structural transition in October 2022, pointing to a broad change in participation rather than an isolated fluctuation in a single metric. Taken together, these results suggest that major external events, particularly the pandemic and the introduction of generative AI tools, coincide with measurable changes in activity on the platform. See Table 3 for parameter estimates and confidence intervals.

Table 3: Segment-wise regression estimates for Questions, Answers, and Accepted Votes. All values are reported /100. For each segment, the table reports slope and intercept estimates with bootstrap standard errors (SE) and 95% bootstrap percentile confidence intervals.

Segment		1	2	3	4
Questions					
Slope	Est.	-1.56	25.67	-3.00	-10.97
	SE	0.28	14.66	13.93	14.47
	CI	(-2.20, -1.15)	(-16.62, 40.86)	(-20.21, 34.30)	(-25.89, 30.84)
Intercept	Est.	8.23	-10.92	11.08	26.39
	SE	0.04	10.44	10.37	13.56
	CI	(8.17, 8.34)	(-21.92, 19.03)	(-17.60, 23.03)	(-17.99, 35.14)
Answers					
Slope	Est.	-1.85	-10.21	-2.81	-8.29
	SE	0.13	4.44	3.90	3.88
	CI	(-2.10, -1.59)	(-14.48, 2.94)	(-12.40, 2.89)	(-13.38, 1.82)
Intercept	Est.	8.70	17.35	10.07	19.46
	SE	0.02	4.48	3.90	5.23
	CI	(8.66, 8.74)	(4.04, 21.60)	(4.21, 19.51)	(3.78, 24.30)
Accept Votes					
Slope	Est.	-0.56	-3.94	-1.04	-3.17
	SE	0.04	1.50	1.30	1.29
	CI	(-0.64, -0.47)	(-4.92, 0.96)	(-4.23, 0.85)	(-4.59, 0.48)
Intercept	Est.	2.58	6.01	3.24	7.05
	SE	0.01	1.55	1.32	1.81
	CI	(2.56, 2.59)	(1.04, 7.11)	(1.24, 6.40)	(1.08, 8.16)

The segment-wise regression estimates clarify how activity evolved within each detected regime. For Questions, the first segment (February 2018–December 2019) exhibits a modest declining trend, followed by a sharp increase during the early pandemic period (December 2019–May 2020), as reflected by the large positive slope in Segment 2. Activity then transitions to a sustained downward trend through October 2022, with an even steeper decline in the final segment after October 2022.

The Answers and Accepted-Vote series display similar patterns. Both show a pronounced negative slope beginning in early 2020, consistent with the structural break identified during the onset of the COVID-19 period. In the postOctober 2022 regime,

all three measures exhibit significantly negative slopes, indicating a sustained decline in engagement following the emergence of large language model tools. The narrow bootstrap confidence intervals across segments suggest that these trend estimates are statistically stable and not driven by sampling variability.

7 Discussion and Future Work

This paper proposed VEGA, a Voronoi-augmented genetic algorithm for high-dimensional derivative-free optimization. The main idea is to combine elitist genetic search with coordinate-wise Voronoi-informed local exploration. By augmenting each elite solution with candidates sampled from its associated local region, VEGA preserves exploitation of high-fitness points while maintaining geometric diversity in the search space. Theoretical results show that, under high-dimensional regularity conditions and a fitness ordering consistent with distance to the target maximizer, the Voronoi mechanism improves the expected convergence behavior relative to standard genetic-algorithm variants. Simulation studies in segmented regression and the Stack Exchange data analysis demonstrate that this improvement translates into more stable breakpoint recovery and useful structural-change detection in practice.

Several theoretical extensions remain important. The current analysis is primarily asymptotic in dimension and is designed to explain why Voronoi-based elite augmentation is useful when high-dimensional random search suffers from poor coverage. A natural next step is to develop fixed-dimensional and non-asymptotic guarantees. Such results would bound the probability that VEGA fails to enter an epsilon-neighborhood of the global maximizer after a finite number of generations, as a function of population size, mutation radius, elite ratio, and the geometry of the objective function. A possible direction is to combine covering-number arguments with finite-sample concentration inequalities for random partitions. This would complement the high-dimensional theory in this paper and provide more explicit guidance for tuning VEGA under a fixed computational budget.

A further limitation concerns pathological or ill-conditioned objective functions. The theoretical justification of VEGA assumes that the elite region continues to contain, or at least remains sufficiently close to, the true maximizer. In highly multimodal, noisy, flat, or misspecified objective functions, this assumption may fail that early elite points may concentrate around a local optimum, and the Voronoi-augmented search may then refine the wrong region. In such cases, elitism can accelerate premature convergence rather than improve global search. Future work should therefore study robust variants of VEGA that preserve a nonzero probability of global exploration. Possible safeguards include multi-start initialization, island-based evolution, adaptive elite-region expansion, occasional uniform resampling, non-vanishing mutation probabilities, and restart rules triggered by stagnation in fitness. A more general theory should replace the assumption that the elite region contains the true maximizer with probabilistic basin-of-attraction conditions, thereby characterizing when VEGA converges globally and when it should be interpreted only as a local derivative-free optimizer.

Appendix A Definitions, Lemmas and Proofs

This Appendix discussed how VEGA performs in high dimensional scenarios under limited budget. Section A.1 provides shorthands for later proofs. Section A.2 presents proofs of technical lemmas. The proofs of main Theorems are given in Section A.3.

A.1 Definitions

Definition 1. *Let $\boldsymbol{K} \subseteq [0,1]^p$ be a set with positive p-dimensional volume $|\boldsymbol{K}| = v$, $0 < v \leq 1$, and $\boldsymbol{X}$ be uniformly distributed over K. Define $R = \|\boldsymbol{X}\|_2$.*

Since $\boldsymbol{K} \subseteq [0,1]^p$, we always have $0 \leq R \leq \sqrt{p}$. We write

$$F_p(r) = \left|\{\boldsymbol{X} \in [0,1]^p : \|\boldsymbol{X}\|_2 \leq r\}\right|,$$

where $0 \leq r \leq \sqrt{p}$. Thus $F_p(r)$ is the volume of the intersection of the cube $[0,1]^p$ with the ball of radius r centered at the origin. Define $r_-(v) = F_p^{-1}(v)$, and $r_+(v) = F_p^{-1}(1-v)$, where F_p^{-1} denotes the generalized inverse,

$$F_p^{-1}(a) = \inf\{r : F_p(r) \geq a\}.$$

Definition 2. *Define three functions as*

$$m_p(v) = r_-(v) - \frac{1}{v}\int_0^{r_-(v)} F_p(t)\,dt,$$

$$M_p(v) = r_+(v) + \frac{1}{v}\int_{r_+(v)}^{\sqrt{p}} \big(1 - F_p(t)\big)\,dt,$$

and

$$Q_p(v) = r_+(v)^2 + \frac{2}{v}\int_{r_+(v)}^{\sqrt{p}} t\big(1 - F_p(t)\big)\,dt.$$

A.2 Lemmas and their Proofs

Proof of Lemma 1

Proof By the independence of $(X_i - a_i)^2$ across i, i.e. $(X_i - a_i)^2 \perp (X_j - a_j)^2$ for $\forall\ i \neq j$, and $\mathrm{Var}\left((X_i - a_i)^2\right) < \infty$ for $\forall\ i = 1, 2, \cdots, p$, $R^2 = \|\boldsymbol{X} - \boldsymbol{a}\|^2 = \sum_{i=1}^p (X_i - a_i)^2$ is asymptotically normal distributed as in high dimensional scenarios when $n, p \to \infty$. To derives its parameters, we further study its expectation and variance.

By rewriting R^2, we have

$$R^2 = \sum_{i=1}^p X_i^2 + \sum_{i=1}^p a_i^2 - 2\sum_{i=1}^p a_i X_i.$$

Hence, the expectation is

$$\begin{aligned}\mathbb{E}R^2 &= \sum_{i=1}^{p}\mathbb{E}X_i^2 + \sum_{i=1}^{p}a_i^2 - 2\sum_{i=1}^{p}a_i\mathbb{E}X_i \\ &= \sum_{i=1}^{p}VarX_i + \sum_{i=1}^{p}\left(\mathbb{E}X_i\right)^2 + a_i^2 - 2a_i\mathbb{E}X_i \\ &= \sum_{i=1}^{p}\left[\frac{1}{12} + \left(a_i - \frac{1}{2}\right)^2\right].\end{aligned} \tag{A1}$$

Similarly, the variance is given by

$$\begin{aligned}\mathrm{Var}(R^2) &= \sum_{i=1}^{p}Var\left(X_i - a_i\right)^2 \\ &= \sum_{i=1}^{p}\left\{\mathbb{E}\left(X_i - a_i\right)^4 - \left[\mathbb{E}\left(X_i - a_i\right)^2\right]^2\right\}.\end{aligned} \tag{A2}$$

Denote the expectation of R^2 as $\mu = r_p^2 = p^{-1}\sum_{i=1}^{p}\left[\frac{1}{12} + \left(a_i - \frac{1}{2}\right)^2\right]$, and the variance $\sigma^2 = p^{-1}\sum_{i=1}^{p}\left\{\mathbb{E}\left(X_i - a_i\right)^4 - \left[\mathbb{E}\left(X_i - a_i\right)^2\right]^2\right\}$, implying $\mathbb{E}R^2 = p\mu$ and $\mathrm{Var}(R^2) = p\sigma^2$. Thus, by (A1) and (A2), the asymptotic distribution of $R^2 \dot{\sim} \mathcal{N}\left(\mu p, \sigma^2 p\right)$, or equivalently $p^{-1}R^2 \dot{\sim} \mathcal{N}\left(\mu, p^{-1}\sigma^2\right)$.

Applying the Delta Method with $g(x) = \sqrt{x}$, we get the asymptotic distribution of R as $p^{-1/2}R \dot{\sim} \mathcal{N}\left(\sqrt{\mu}, p^{-1}\sigma_0^2\right)$, where $\sigma_0^2 = \left[g'(\mu)\right]^2\sigma^2 = \frac{\sigma^2}{4\mu}$. Notate $\mu_p = \sqrt{p\mu}$ and $\sigma_p^2 = \sigma_0^2$, thus $R \dot{\sim} \mathcal{N}\left(r_p, \sigma_p^2\right)$.

Naturally denote that $R_{min} = \min_j ||\boldsymbol{X}_j - \boldsymbol{a}|| = \min_j R_j$, where $R_j \dot{\sim} \mathcal{N}\left(\mu_p, \sigma_p^2\right)$ independently. It completes the proof.

□

Lemma 2 (Minimum of Gaussian Samples Asymptotics)**.** *Let* $Z_1, \ldots, Z_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$*, and define*

$$L_n = \min_{1\le i\le n} Z_i.$$

As $n \to \infty$*,*

$$\mathbb{E}[L_n] = -\sqrt{2\log n} + \frac{\log\log n + \log(4\pi) - 2\gamma}{2\sqrt{2\log n}} + \mathcal{O}\left(\frac{1}{\sqrt{\log n}}\right),$$

where γ *is the Euler–Mascheroni constant. Moreover,*

$$\mathrm{Var}(L_n) = \frac{\pi^2}{12\log n} + \mathcal{O}\left(\frac{1}{\log n}\right).$$

In particular,

$$\mathbb{E}[L_n] = -\sqrt{2\log n} + \mathcal{O}\left(\frac{\log\log n}{\sqrt{\log n}}\right) \quad \textit{and} \quad \mathrm{Var}(L_n) = \mathcal{O}\left(\frac{1}{\log n}\right).$$

Proof Let $M_n := \max_{1\le i\le n} Z_i$. By symmetry of normal distributions,

$$L_n \overset{d}{=} -M_n.$$

Let

$$b_n = \sqrt{2\log n} - \frac{\log\log n + \log(4\pi)}{2\sqrt{2\log n}}.$$

The classical extreme-value asymptotics for Gaussian maxima gives

$$b_n(M_n - b_n) \xrightarrow{d} G,$$

where G has the standard Gumbel distribution. Hence

$$\mathbb{E}[M_n] = b_n + \frac{\gamma}{b_n} + o\left(\frac{1}{b_n}\right), \qquad \mathrm{Var}(M_n) = \frac{\pi^2}{6b_n^2} + \mathcal{O}\left(\frac{1}{b_n^2}\right).$$

Since $b_n^2 \sim 2\log n$ and $L_n \overset{d}{=} -M_n$, we obtain

$$\mathbb{E}[L_n] = -b_n - \frac{\gamma}{b_n} + \mathcal{O}\left(\frac{1}{b_n}\right),$$

and

$$\mathrm{Var}(L_n) = \mathrm{Var}(M_n) = \frac{\pi^2}{12\log n} + \mathcal{O}\left(\frac{1}{\log n}\right).$$

Substituting the expansion of b_n gives the stated expectation estimate. □

Lemma 3 (Layer Cake Representation)**.** *Let $\boldsymbol{A} \subseteq [0,1]^p$ be measurable with $|\boldsymbol{A}| = v > 0$, and let $\boldsymbol{Y} \sim \mathrm{Unif}(\boldsymbol{A})$. If $S = \|Y\|$, then*

$$\mathbb{E}S = \frac{1}{v}\int_0^{\sqrt{p}} |A \cap \{x : \|X\| > t\}|\, dt,$$

and

$$\mathbb{E}S^2 = \frac{1}{v}\int_0^{\sqrt{p}} 2t\,|A \cap \{x : \|X\| > t\}|\, dt.$$

Proof For $\forall\ \boldsymbol{X} \in [0,1]^p$,

$$\|\boldsymbol{X}\| = \int_0^{\sqrt{p}} \mathbb{1}_{\{t<\|\boldsymbol{X}\|\}}\, dt.$$

Therefore

$$\int_{\boldsymbol{A}} \|\boldsymbol{X}\|\, dx = \int_{\boldsymbol{A}}\int_0^{\sqrt{p}} \mathbb{1}_{\{t<\|\boldsymbol{X}\|\}}\, dt\, dx.$$

By Fubini's theorem,

$$\int_{\boldsymbol{A}} \|\boldsymbol{X}\|\, dx = \int_0^{\sqrt{p}} |\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}|\, dt.$$

Dividing by $v = |\boldsymbol{A}|$ gives the formula for $\mathbb{E}S$.

Similarly,

$$\|\boldsymbol{X}\|^2 = \int_0^{\|\boldsymbol{X}\|} 2t\,dt = \int_0^{\sqrt{p}} 2t\mathbb{1}_{\{t<\|\boldsymbol{X}\|\}}\,dt.$$

Repeating the same argument yields the stated formula for $\mathbb{E}S^2$. □

Lemma 4 (Tail Volume Bound)**.** *Let $\boldsymbol{A} \subseteq [0,1]^p$ be measurable with $|\boldsymbol{A}| = v$. Then, for $\forall\ 0 \le t \le \sqrt{p}$,*

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| \ge \big(v - F_p(t)\big)_+,$$

and

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| \le \min\{v, 1 - F_p(t)\}.$$

Proof As we have

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| \le t\}| \le F_p(t),$$

so

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| = v - |\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| \le t\}| \ge v - F_p(t).$$

Notice that the left-hand side is nonnegative, implying

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| \ge \big(v - F_p(t)\big)_+.$$

For the upper bound, clearly

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| \le v,$$

and

$$\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\} \subseteq \{\boldsymbol{X} \in [0,1]^p : \|\boldsymbol{X}\| > t\},$$

whose volume is $1 - F_p(t)$. Hence

$$|\boldsymbol{A} \cap \{\boldsymbol{X} : \|\boldsymbol{X}\| > t\}| \le \min\{v, 1 - F_p(t)\}.$$

□

Lemma 5 (Expectation Lower Bound)**.** *Let $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$, where $|\boldsymbol{K}| = v$. Then*

$$\mathbb{E}\|\boldsymbol{X}\| \ge m_p(v).$$

Proof By Lemmas 3 and 4,

$$\mathbb{E}\|\boldsymbol{X}\| \ge \frac{1}{v}\int_0^{\sqrt{p}} \big(v - F_p(t)\big)_+\,dt.$$

By the definition of $r_-(v) = F_p^{-1}(v)$, the positive part is supported on $[0, r_-(v)]$. Hence

$$\mathbb{E}\|\boldsymbol{X}\| \ge \frac{1}{v}\int_0^{r_-(v)} \big(v - F_p(t)\big)\,dt.$$

Therefore

$$\mathbb{E}\|\boldsymbol{X}\| \ge r_-(v) - \frac{1}{v}\int_0^{r_-(v)} F_p(t)\,dt = m_p(v).$$

□

Lemma 6 (Expectation Upper Bound). *Let* $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$, *where* $|\boldsymbol{K}| = v$. *Then*

$$\mathbb{E}\|\boldsymbol{X}\| \le M_p(v).$$

Proof By Lemmas 3 and 4,

$$\mathbb{E}\|\boldsymbol{X}\| \le \frac{1}{v}\int_0^{\sqrt{p}} \min\{v, 1 - F_p(t)\}\, dt.$$

By the definition of $r_+(v) = F_p^{-1}(1-v)$, we have $1 - F_p(t) \ge v$, for $0 \le t \le r_+(v)$, and $1 - F_p(t) \le v$ for $t \ge r_+(v)$. Therefore

$$\mathbb{E}\|\boldsymbol{X}\| \le \frac{1}{v}\left[\int_0^{r_+(v)} v\, dt + \int_{r_+(v)}^{\sqrt{p}} \big(1 - F_p(t)\big)\, dt\right].$$

Thus

$$\mathbb{E}\|\boldsymbol{X}\| \le r_+(v) + \frac{1}{v}\int_{r_+(v)}^{\sqrt{p}} \big(1 - F_p(t)\big)\, dt = M_p(v).$$

□

Lemma 7 (Second Moment Upper Bound). *Let* $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$, *where* $|\boldsymbol{K}| = v$. *Then*

$$\mathbb{E}\|\boldsymbol{X}\|^2 \le Q_p(v).$$

Proof By Lemmas 3 and 4,

$$\mathbb{E}\|\boldsymbol{X}\|^2 \le \frac{1}{v}\int_0^{\sqrt{p}} 2t \min\{v, 1 - F_p(t)\}\, dt.$$

Splitting the integral at $r_+(v)$ gives

$$\mathbb{E}\|\boldsymbol{X}\|^2 \le \frac{1}{v}\left[\int_0^{r_+(v)} 2tv\, dt + \int_{r_+(v)}^{\sqrt{p}} 2t\big(1 - F_p(t)\big)\, dt\right].$$

The first integration equals $r_+(v)^2$, so

$$\mathbb{E}\|\boldsymbol{X}\|^2 \le r_+(v)^2 + \frac{2}{v}\int_{r_+(v)}^{\sqrt{p}} t\big(1 - F_p(t)\big)\, dt = Q_p(v).$$

□

Lemma 8 (Expectation Closed-form Lower Bound). *For* $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$,

$$\mathbb{E}\|\boldsymbol{X}\| \ge \frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}.$$

Proof By definition, $F_p(t) = |\{\boldsymbol{X} \in [0,1]^p : \|\boldsymbol{X}\| \le t\}| \le \frac{\kappa_p}{2^p} t^p, \forall t \ge 0$. Let $c_p = \kappa_p / 2^p$. By Lemma 5 and $F_p(t) \le c_p t^p$, we have

$$\mathbb{E}\|\boldsymbol{X}\| \ge \frac{1}{v}\int_0^{\sqrt{p}} \big(v - F_p(t)\big)_+\, dt$$

$$\geq \frac{1}{v}\int_0^{(v/c_p)^{1/p}} \left(v - c_p t^p\right) dt$$
$$= \frac{p}{p+1}\left(\frac{v}{c_p}\right)^{1/p}.$$

Since $c_p = \kappa_p/2^p$, we have

$$\mathbb{E}\|\boldsymbol{X}\| \geq \frac{2p}{p+1}\left(\frac{v}{\kappa_p}\right)^{1/p},$$

which completes the proof. □

Lemma 9 (Simplex Volume Estimate). *For* $\boldsymbol{X} = (X_1, X_2, \cdots, X_p) \in [0,1]^p$, *define*

$$S(\boldsymbol{X}) = \sum_{i=1}^{p}(1 - X_i).$$

Then

$$\mathbb{E}S(\boldsymbol{X}) \geq \frac{p}{p+1}(p!v)^{1/p}.$$

Proof Let $u_i = 1 - X_i$ and $\boldsymbol{u} = (u_1, u_2, \cdots, u_p)$. The mapping of $\boldsymbol{X} \mapsto \boldsymbol{u} = \boldsymbol{1} - \boldsymbol{X}$ preserves volume. Therefore, $\boldsymbol{u} = \boldsymbol{1} - \boldsymbol{X}$ is uniformly distributed on a measurable subset of $[0,1]^p$ with volume v.

Define $G_p(t) = |\{\boldsymbol{u} \in [0,1]^p : u_1 + \cdots + u_p \leq t\}|$. The set $\{\boldsymbol{u} \in [0,1]^p : u_1 + \cdots + u_p \leq t\}$ is contained in the simplex $\{\boldsymbol{u} \in \mathbb{R}_+^p : u_1 + \cdots + u_p \leq t\}$, whose volume is $t^p/p!$. Hence

$$G_p(t) \leq \frac{t^p}{p!}.$$

Applying the layer-cake representation in Lemma 5 to the random variable

$$S(\boldsymbol{u}) = u_1 + \cdots + u_p,$$

obviously we have

$$\mathbb{E}S(\boldsymbol{X}) = \mathbb{E}S(\boldsymbol{u}) \geq \frac{1}{v}\int_0^p \left(v - G_p(t)\right)_+ dt.$$

Using $G_p(t) \leq t^p/p!$ gives

$$\mathbb{E}S(\boldsymbol{X}) \geq \frac{1}{v}\int_0^{(p!v)^{1/p}} \left(v - \frac{t^p}{p!}\right) dt.$$

Computing the integral yields

$$\mathbb{E}S(\boldsymbol{X}) \geq \frac{p}{p+1}(p!v)^{1/p}.$$

□

Lemma 10 (Expectation Closed-form Upper Bound). *For* $\boldsymbol{X} = (X_1, X_2, \cdots, X_p) \sim \mathrm{Unif}(\boldsymbol{K})$,

$$\mathbb{E}\|\boldsymbol{X}\| \leq \sqrt{p} - \frac{p}{2(p+1)\sqrt{p}}(p!v)^{1/p}.$$

Proof For $\boldsymbol{X} \in [0,1]^p$,

$$\sqrt{p} - \|\boldsymbol{X}\| = \frac{p - \|X\|^2}{\sqrt{p} + \|X\|},$$

because

$$p - \|\boldsymbol{X}\|^2 = \sum_{i=1}^{p}(1 - X_i^2) = \sum_{i=1}^{p}(1 - X_i)(1 + X_i) \geq \sum_{i=1}^{p}(1 - X_i) = S(X),$$

Note that

$$\sqrt{p} + \|\boldsymbol{X}\| \leq 2\sqrt{p},$$

which implies that

$$\sqrt{p} - \|\boldsymbol{X}\| \geq \frac{S(\boldsymbol{X})}{2\sqrt{p}} \quad \text{and } \|\boldsymbol{X}\| \leq \sqrt{p} - \frac{S(x)}{2\sqrt{p}}.$$

Taking expectations on both sides gives

$$\mathbb{E}\|\boldsymbol{X}\| \leq \sqrt{p} - \frac{1}{2\sqrt{p}}\mathbb{E}S(\boldsymbol{X}).$$

Therefore

$$\mathbb{E}\|\boldsymbol{X}\| \leq \sqrt{p} - \frac{p}{2(p+1)\sqrt{p}}(p!v)^{1/p},$$

by Lemma 9 as $\mathbb{E}S(\boldsymbol{X}) \geq p(p!v)^{1/p}/(p+1)$. It completes the proof. □

Lemma 11 (Second Moment Closed-form Upper Bound)**.** *For* $\boldsymbol{X} = (X_1, X_2, \cdots, X_p) \sim \text{Unif}(\boldsymbol{K})$,

$$\mathbb{E}\|\boldsymbol{X}\|^2 \leq p - \frac{p}{p+1}(p!v)^{1/p}.$$

Proof For $\forall\ \boldsymbol{X} \in [0,1]^p$,

$$\|\boldsymbol{X}\|^2 = \sum_{i=1}^{p} X_i^2 = p - \sum_{i=1}^{p}(1 - X_i^2).$$

Since

$$1 - X_i^2 = (1 - X_i)(1 + X_i) \geq 1 - X_i,$$

we have

$$\|\boldsymbol{X}\|^2 \leq p - \sum_{i=1}^{p}(1 - X_i) = p - S(\boldsymbol{X}).$$

Taking expectation on both sides yields

$$\mathbb{E}\|\boldsymbol{X}\|^2 \leq p - \mathbb{E}S(\boldsymbol{X}).$$

Using Lemma 9,

$$\mathbb{E}S(\boldsymbol{X}) \geq \frac{p}{p+1}(p!v)^{1/p}.$$

Thus

$$\mathbb{E}\|\boldsymbol{X}\|^2 \leq p - \frac{p}{p+1}(p!v)^{1/p},$$

which completes the proof. □

Lemma 12. *As* $p \to \infty$, $\kappa_p^{1/p} \asymp p^{-1/2}$ *and* $(p!)^{1/p} \asymp p$. .

Proof By Stirling's formula,

$$\Gamma\left(\frac{p}{2}+1\right) \asymp \left(\frac{p}{2}\right)^{p/2} e^{-p/2}\sqrt{p}.$$

Hence

$$\kappa_p = \frac{\pi^{p/2}}{\Gamma(p/2+1)} \Rightarrow \kappa_p^{1/p} \asymp p^{-1/2}.$$

Similarly, we have

$$p! \asymp \sqrt{p}\left(\frac{p}{e}\right)^p \Rightarrow (p!)^{1/p} \asymp p.$$

□

Lemma 13. *Let $\boldsymbol{K} \subseteq [0,1]^p$ with $|\boldsymbol{K}| = v$, where $0 < v \leq 1$, and let $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$. Then*

$$m_p(v) \leq \mathbb{E}\|\boldsymbol{X}\| \leq M_p(v).$$

Moreover,

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq Q_p(v) - m_p(v)^2.$$

Consequently,

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq \min\left\{\frac{p}{4},\, Q_p(v) - m_p(v)^2\right\}.$$

Proof The lower bound and upper bound of $\mathbb{E}\|\boldsymbol{X}\|$ is given by Lemma 5 and Lemma 6 directly.

For the variance, we use the second order moment of

$$\mathrm{Var}(\|\boldsymbol{X}\|) = \mathbb{E}\|\boldsymbol{X}\|^2 - \left(\mathbb{E}\|\boldsymbol{X}\|\right)^2.$$

By Lemma 7,

$$\mathbb{E}\|\boldsymbol{X}\|^2 \leq Q_p(v),$$

and by Lemma 5,

$$\mathbb{E}\|\boldsymbol{X}\| \geq m_p(v).$$

Hence

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq Q_p(v) - m_p(v)^2.$$

Since $0 \leq \|\boldsymbol{X}\| \leq \sqrt{p}$, Popoviciu's variance inequality shows

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq \frac{(\sqrt{p}-0)^2}{4} = \frac{p}{4}.$$

Combining the two bounds,

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq \min\left\{\frac{p}{4},\, Q_p(v) - m_p(v)^2\right\}.$$

□

Lemma 14. *Let $\boldsymbol{K} \subseteq [0,1]^p$ be convex with $|\boldsymbol{K}| = v$, where $0 < v \leq 1$, and let $\boldsymbol{X} \sim \mathrm{Unif}(\boldsymbol{K})$. Then*

$$\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p} \leq \mathbb{E}\|\boldsymbol{X}\| \leq \sqrt{p} - \frac{p}{2(p+1)\sqrt{p}}(p!v)^{1/p}.$$

Moreover,

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq p - \frac{p}{p+1}(p!v)^{1/p} - \left[\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}\right]^2.$$

Consequently,

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq \min\left\{\frac{p}{4},\, p - \frac{p}{p+1}(p!v)^{1/p} - \left[\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}\right]^2\right\},$$

where $\kappa_p = \pi^{p/2}/\Gamma(p/2+1)$ *denotes the volume of the unit ball in* $\mathbb{R}^p$.

Proof The lower bound for the expectation follows from Lemma 8, and the upper bound for the expectation follows from Lemma 10.

For the variance, we again use

$$\mathrm{Var}(\|\boldsymbol{X}\|) = \mathbb{E}\|\boldsymbol{X}\|^2 - \left(\mathbb{E}\|\boldsymbol{X}\|\right)^2.$$

By Lemma 11,

$$\mathbb{E}\|\boldsymbol{X}\|^2 \leq p - \frac{p}{p+1}(p!v)^{1/p}.$$

By Lemma 8,

$$\mathbb{E}\|\boldsymbol{X}\| \geq \frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}.$$

Therefore

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq p - \frac{p}{p+1}(p!v)^{1/p} - \left[\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}\right]^2.$$

Combining this bound of Popoviciu's inequality

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq \frac{p}{4}$$

gives the stated minimum bound. □

A.3 Proofs of Theorems

Proof of Theorem 1

Proof By Lemma 1, $R_i \overset{i.i.d}{\sim} \mathcal{N}\left(\mu_p, \sigma_p^2\right)$ regardless of sample size n. By standardization, $\sigma_p^{-1}(R_i - \mu_p) \overset{i.i.d}{\sim} \mathcal{N}(0,1)$. Let $R_{min} = min\{R_1, R_2, \cdots, R_n\}$, and it is obvious that $R_{min} = \mu_p + \sigma_p \mathbb{E} Z_{(1)}$, where $Z_{(1)}$ is the minimum normal random variable, i.e. $Z_{(1)} = \min(Z_1, Z_2, \cdots, Z_n)$ and $Z_i \overset{i.i.d}{\sim} \mathcal{N}(0,1)$.

Take expectation and apply Lemma 2, we have

$$
\begin{aligned}
\mathbb{E}R_{min} &= \mu_p + \sqrt{\sigma_p^2}\mathbb{E}Z_{(1)} \\
&\approx \mu_p + \sigma_p \Phi^{-1}\left(\frac{1}{n+1}\right) \\
&= \mu_p + \sigma_p \left(-\sqrt{2\log n} + \mathcal{O}\left(\frac{\log\log n}{\sqrt{\log n}}\right)\right) \\
&= \mu_p - \sigma_p\sqrt{2\log n} + \mathcal{O}\left(\frac{\log\log n}{\sqrt{\log n}}\right),
\end{aligned}
$$

where $\Phi(\cdot)$ is the cdf of standard normal distribution. Hence $R_{min} = \mu_p + \sigma_p Z_{(1)}$ implies $R_{min} = \mu_p - \sigma_p\sqrt{2\log n} + \mathcal{O}\left(\frac{\log\log n}{\sqrt{\log n}}\right) + \mathcal{O}_p(log^{-1/2}n)$ since, by Lemma 2, $\mathrm{Var}Z_{(1)} = \mathcal{O}(\log^{-1} n)$. □

A.3.1 Proof of Theorem 2

Proof Similarly to the procedure in Lemma 1,

$$
\begin{aligned}
\mathbb{E}R^2 &= \sum_{i=1}^{p} VarX_i + \sum_{i=1}^{p}(a_i - \mathbb{E}X_i)^2 \\
&= \sum_{i=1}^{p}(a_i - \mathbb{E}X_i)^2 + \mathcal{O}\left(pS^2\right) \\
&= \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\|^2 + \mathcal{O}\left(pS^2\right),
\end{aligned}
$$

and

$$
\mathrm{Var}(R^2) = \sum_{i=1}^{p}\left\{\mathbb{E}\left(X_i^{(2)} - a_i\right)^4 - \left[\mathbb{E}\left(X_i^{(2)} - a_i\right)^2\right]^2\right\} = \mathcal{O}\left(pB + pS^4\right),
$$

implying $\mathbb{E}R = \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| + \mathcal{O}\left(p^{-1/2}S\right)$, and $\mathrm{Var}R = \mathcal{O}\left(B^{1/2} + S^2\right)$. Further $R^2 = \mathbb{E}R^2 + \mathcal{O}_p\left(B^{1/4} + S\right)$.

By Theorem 1 and notations in Lemma 1,

$$
\begin{aligned}
R_{min} =& r_p + \sigma_p\mathbb{E}Z_{(1)} \\
=& \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| + \mathcal{O}\left(p^{\frac{1}{2}}S\right) - \mathcal{O}\left(p^{\frac{1}{2}}(B + S^4)^{\frac{1}{2}}\log^{\frac{1}{2}} n\right) \\
&+ \mathcal{O}_p\left((B + S^4)^{\frac{1}{4}}\log^{-\frac{1}{2}} n\right) \\
=& \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| - \mathcal{O}\left(S^2 p^{\frac{1}{2}}\log^{\frac{1}{2}} n\right) + \mathcal{O}_p\left(S\log^{-\frac{1}{2}} n\right)
\end{aligned}
$$

$p^{-1}\|\mathbb{E}\boldsymbol{X}\|$, $p^{-1}\|\mathbb{E}\boldsymbol{X}^2\| = \mathcal{O}(S^2)$, and $p^{-1}\|\mathbb{E}\boldsymbol{X}^4\| = \mathcal{O}(B)$ are bounded, so it implies S and B are both at constant level, which means

$$
R_{min} = \|\mathbb{E}\boldsymbol{X} - \boldsymbol{a}\| - \mathcal{O}\left(p^{\frac{1}{2}}\log^{\frac{1}{2}} n\right) + \mathcal{O}_p\left(\log^{-\frac{1}{2}} n\right)
$$

□

A.3.2 Proof of Theorem 3

Proof By Lemma 12,

$$\kappa_p^{1/p} \asymp p^{-1/2}.$$

Therefore

$$\left(\frac{2^p v}{\kappa_p}\right)^{1/p} = 2v^{1/p}\kappa_p^{-1/p} \asymp \sqrt{p}\, v^{1/p}.$$

Using the closed-form lower bound from Lemma 8, we obtain $\mathbb{E}\|\boldsymbol{X}\| \gtrsim \sqrt{p}\, v^{1/p}$. Again by Lemma 12, $(p!)^{1/p} \asymp p$. Hence

$$\frac{(p!v)^{1/p}}{\sqrt{p}} \asymp \sqrt{p}\, v^{1/p}.$$

Using the upper bound from Lemma 10, we have, for some universal constant $c > 0$, $\mathbb{E}\|\boldsymbol{X}\| \leq \sqrt{p} - c\sqrt{p}\, v^{1/p}$. For the variance, we have

$$\mathrm{Var}(\|\boldsymbol{X}\|) \leq p - \frac{p}{p+1}(p!v)^{1/p} - \left[\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}\right]^2.$$

By Lemma 12, we have

$$\frac{p}{p+1}(p!v)^{1/p} \asymp pv^{1/p},$$

and

$$\left[\frac{p}{p+1}\left(\frac{2^p v}{\kappa_p}\right)^{1/p}\right]^2 \asymp pv^{2/p}.$$

Therefore

$$Var\|\boldsymbol{X}\| \lesssim \mathcal{O}(p) - \mathcal{O}\left(pv^{1/p}\right) - \mathcal{O}\left(pv^{2/p}\right).$$

□

A.3.3 Proof of Theorem 4

Proof By definition, $\boldsymbol{C}_i(\boldsymbol{K}) = \{\boldsymbol{c} \in \boldsymbol{K} \mid \forall j \neq i, \|\boldsymbol{c} - \boldsymbol{X}_i\| \leq \|\boldsymbol{c} - \boldsymbol{X}_j\|\}$.

Denote its volume as $V_i = \mathrm{vol}_{[rn]}(\boldsymbol{C}_i(\boldsymbol{K}))$. Because $\boldsymbol{a} \notin \partial\boldsymbol{K}$, samples are around a spherical shell (Li 2021), similarly as shown in Lemma 1.

Under high dimension, asymptotic distribution of $\frac{[rn]V_i}{|K|}$ only relies on dimensions p, so denote $[rn]V_i/|\boldsymbol{K}| \to T_p$, where $\mathbb{E}T_p = 1$ and $Var(T_p) \leqslant 6\left(\frac{3}{4}\right)^{p/2}$. See Devroye et al. (2017) for a detailed proof. Thus we can write the expression as $[rn]V_i/|\boldsymbol{K}| = T_p + \mathcal{O}_p(1)$. Further,

$$\begin{aligned} V_i &= \frac{|\boldsymbol{K}|}{[rn]}T_p + \mathcal{O}_p\left(\frac{|\boldsymbol{K}|}{[rn]}\right) \\ &= \frac{|\boldsymbol{K}|}{[rn]} + \frac{|\boldsymbol{K}|}{[rn]}(T_p - 1) + \mathcal{O}_p\left(\frac{|\boldsymbol{K}|}{[rn]}\right) \\ &= \frac{|\boldsymbol{K}|}{[rn]} + \mathcal{O}_p\left(\frac{|\boldsymbol{K}|}{[rn]}\left(\frac{3}{4}\right)^{p/4}\right) + \mathcal{O}_p\left(\frac{|\boldsymbol{K}|}{[rn]}\right). \end{aligned}$$

For any small ξ, the maximum of the volume M satisfies

$$\begin{aligned} \max_{1\leq i\leq [rn]} V_i &= \frac{|\boldsymbol{K}|}{[rn]} + \frac{|\boldsymbol{K}|}{[rn]}\mathcal{O}_p\left(\left(\frac{3}{4}\right)^{p/4}\sqrt{\log[rn]}\right) + \mathcal{O}_p\left(\frac{|\boldsymbol{K}|}{[rn]}\sqrt{\log[rn]}\right) \\ &= \frac{|\boldsymbol{K}|}{[rn]} + \mathcal{O}_p\left(n^{-1-\xi}\right) \end{aligned}$$

In VEGA, we generate a elitism region after each generation where the augmented children are located. Considering the elite rate r, each elite region has an augmentation size of rn with only one elite individual. Hence, we could use the volume upper bound as order bound $v = |\boldsymbol{K}|/rn + \mathcal{O}_p\left(n^{-1-\xi}\right)$ in Theorem 3.

As the initial space is a high dimensional cube, which is convex, so elitism region is also convex. What's more by induction and the assumption that fitness $f(\boldsymbol{X}_1) > f(\boldsymbol{X}_2)$ almost surely for candidates closer to true value, $\boldsymbol{a}$ is always in the elitism region which is still a convex set. Notice the fact that the $\boldsymbol{a}$ is always in one of the Voronoi cells.

In the Voronoi cell which contain the true maximizer $\boldsymbol{a}$, there are $[rn]$ augmented samples in crossover step. Hence, we further consider this specific Voronoi cell $\boldsymbol{K}^{(2)}$. Applying the conditions at the first generation, the minimum distance $R^{(2)}_{min}$ between new samples to $\boldsymbol{a}$ satisfies the format of $R^2_{min} = \mu_p + \sigma_p Z_{(1)}$. In high dimensional scenarios, we could approximately take the Voronoi cells as hyper-cones, so the order in Theorem 3 reduces to

$$\sqrt{p}\, v^{1/p} \asymp \mathbb{E}\|\boldsymbol{X}\|,$$

and

$$Var\|\boldsymbol{X}\| \lesssim \mathcal{O}(p) - \mathcal{O}\left(pv^{1/p}\right) - \mathcal{O}\left(pv^{2/p}\right),$$

where $v = \mathcal{O}(n^{-1}) + \mathcal{O}_p(n^{-1-\xi})$. Hence we have

$$\begin{aligned} R^{(2)}_{min} \lesssim & \mathcal{O}(p^{\frac{1}{2}} n^{-\frac{(1+\xi)}{p}} - p^{\frac{1}{2}} g(m) \log^{\frac{1}{2}} n) \\ & + \mathcal{O}_p(g^{\frac{1}{2}}(m) \log^{-\frac{1}{2}} n). \end{aligned}$$

The final minimum distance, considering the elites, is $R_{min} = \min\{R^{(2)}_{min}, R^{(1)}_{min}\}$. Since the samples are generated independently,

$$\begin{aligned} R_{min} \lesssim & \mathcal{O}(p^{\frac{1}{2}} n^{-\frac{(1+\xi)}{p}} - p^{\frac{1}{2}} g(m) \log^{\frac{1}{2}} (n + rn) \\ & + \mathcal{O}_p(g^{\frac{1}{2}}(m) \log^{-\frac{1}{2}} (n + rn)). \end{aligned}$$

As generation goes to m, it further becomes, by induction,

$$\begin{aligned} R^{(m)}_{min} \lesssim & \mathcal{O}(p^{\frac{1}{2}} n^{-\frac{(1+\xi)}{p}} - p^{\frac{1}{2}} g(m) \log^{\frac{1}{2}} (n + mrn) \\ & + \mathcal{O}_p(g^{\frac{1}{2}}(m) \log^{-\frac{1}{2}} (n + mrn)). \end{aligned}$$

□

Appendix B Pseudo Codes

Algorithm 2 GA Framework

```
procedure GeneticAlgorithm(island)
    Set Population size, Island number, Elite ratio and tolerance = tol
    Set max_iter = 100, mutation rate for Controls
    Initialize
    for g = 1 to max_iter do
        if island = false then
            best_before ← best fitness in generation
            generation ← Evolve(generation)
            best_after ← best fitness in generation
        else
            best_before ← min best fitness across islands
            if g mod island_crossover = 0 then
                pool ← flatten all islands into one population
                pool ← Evolve(pool)
                Shuffle pool and redistribute evenly across islands
            else
                Apply Evolve independently within each island
            end if
            best_after ← min best fitness across islands
        end if
        Δ ← (best_before − best_after)/best_before
        if Δ < tol then
            break
        end if
    end for
end procedure
```

[1] The boundaries of $\boldsymbol{S}_j$ could be complicated after several generations, a possible remedy could be replacing the exact diagram by the circumsphere or circumcube Das et al. (2021).

[2] Top candidates (elites) could be kept for monotonicity

Algorithm 3 Control 0: Evolution Operator

procedure EVOLVE(`batch`)
    $m \leftarrow |\texttt{batch}|$
    Initialize empty population $\texttt{batch}'$
    for $i = 1$ to m **do**
        Select two parents uniformly at random from `batch`
        Generate offspring via single-point crossover
        Apply additive Gaussian mutation to each gene with fixed probability
        Clip offspring to $[0, 1]$
        Append offspring to $\texttt{batch}'$
    end for
    return `batch`' sorted by fitness
end procedure

Algorithm 4 Control 1: Evolution Operator

procedure EVOLVE(`batch`)
    Let $m \leftarrow |\texttt{batch}|$
    Let `elites` $\leftarrow$ top $\lfloor rm \rfloor$ individuals from `batch`
    Compute selection probabilities over `batch` proportional to reciprocal fitness
    Initialize empty list `children`
    for $i = 1$ to $m - |\texttt{elites}|$ **do**
        Sample two parents from `batch` using the fitnessbased probabilities
        Generate offspring via singlepoint crossover
        Apply mutation to each gene with fixed probability (scaled by parent difference with a minimum step size)
        Clip offspring to $[0, 1]$
        Append offspring to `children`
    end for
    return (`elites` + `children`) sorted by fitness
end procedure

Algorithm 5 Voronoi Partition Search

Require: Fitness function $\ell(\boldsymbol{Z}, \boldsymbol{\theta}; \boldsymbol{X})$, Max search area $\boldsymbol{S}$

Initialize Population $= n$, MaxIter $= m$
Generate n multidimensional uniform random vector $\boldsymbol{Z}$ on $\boldsymbol{S}$
for i in 1:m **do**
  for j in 1:n **do**
    Derive $\widehat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \quad \ell(\boldsymbol{Z}_j, \boldsymbol{\theta}; \boldsymbol{X})$
    Encode $\boldsymbol{Z}_j$ to Chromosome(j)
  end for
  Sort Chromosome descendingly by $\ell(\boldsymbol{Z}_j, \widehat{\boldsymbol{\theta}}_j; \boldsymbol{X})$
  Keep r(%) with top Fitness
  By each kept $\boldsymbol{Z}_j$'s, **Construct** Voronoi Partitioned area as new $\boldsymbol{S}_j$

$$\boldsymbol{S}_j = \boldsymbol{S}_{1,j} \times \boldsymbol{S}_{2,j} \times \cdots \times \boldsymbol{S}_{p,j}$$

  for k in 1:p **do**
    Generate $\frac{1}{r}$ uniform random number within $\boldsymbol{S}_j$
    Encode as Chromosomes
  end for
  Crossover, **Mutation** within each partition
  Amalgamate all Chromosome together and **Decode**
  Start next loop with each $\boldsymbol{S}_j$ as S [1]
end for
Sort Chromosome descendingly by $\ell(\boldsymbol{Z}_i, \widehat{\boldsymbol{\theta}}_i; \boldsymbol{X})$ [2]
Return $\boldsymbol{Z}_i, \widehat{\boldsymbol{\theta}}_i$ corresponding to best Chromosome

Algorithm 6 VEGA: Voronoi-Augmented Evolution Operator

procedure EVOLVE(`batch`)
Let $m \leftarrow |\texttt{batch}|$
Let `elites` $\leftarrow$ top $\lfloor rm \rfloor$ individuals from `batch`
Construct rectangular Voronoi bounds $[\texttt{low}_i, \texttt{high}_i]$ around each elite
Initialize empty list `augmented`
for each elite i **do**
Sample a small set of candidate points uniformly from the hyperrectangle $[\texttt{low}_i, \texttt{high}_i]$
if any candidate has finite fitness **then**
Add the first finitefitness candidate to `augmented`
else
Add the elite itself to `augmented`
end if
end for
`pool` $\leftarrow$ `elites` + `augmented`
Compute selection probabilities over `pool` proportional to reciprocal fitness
Initialize empty list `children`
for $i = 1$ to $m - |\texttt{elites}|$ **do**
Sample two parents from `pool` using the fitness-based probabilities
Generate offspring via single-point crossover
Apply mutation to each gene with fixed probability (scaled by parent difference with a minimum step size)
Clip offspring to $[0, 1]$
Append offspring to `children`
end for
return (`elites` + `children`) sorted by fitness
end procedure

Appendix C Additional Figures

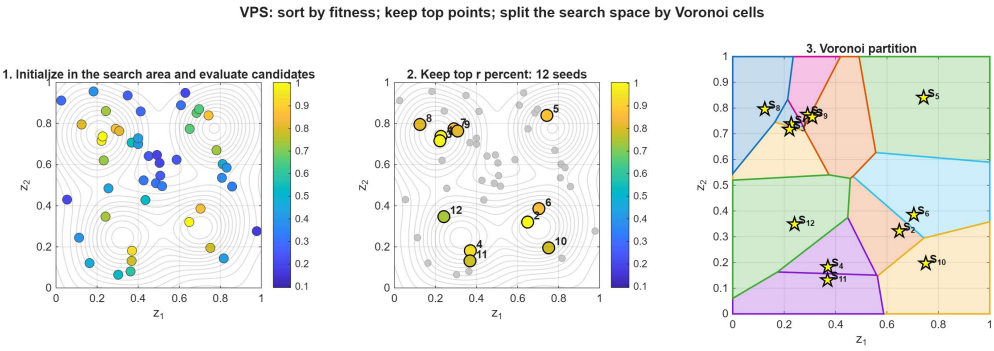


Fig. C1: VPS initialization and partitioning: candidates are uniformly sampled and evaluated in the search space, the top r(%) high-fitness points are retained as seeds, and these seeds define Voronoi cells that partition the space into region-specific search domains.

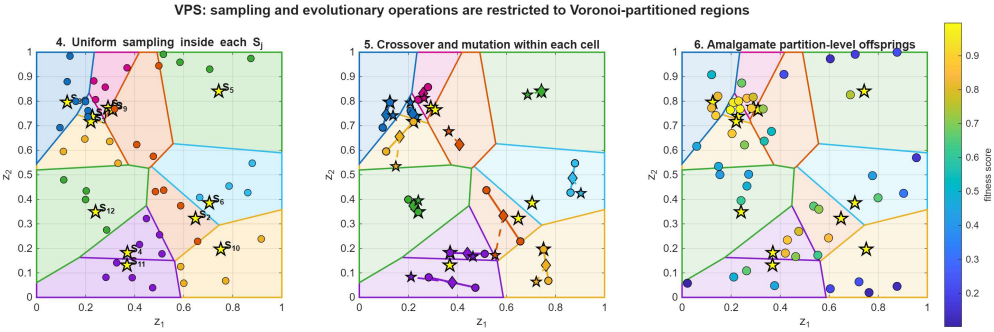


Fig. C2: VPS region-restricted evolution: within each Voronoi cell, new candidates are uniformly sampled and evolved by crossover and mutation using only cell-local individuals. Offspring from all cells are then amalgamated into the next population, preserving partition-guided exploration.

Appendix D Some Potential Applications

D.1 Quantile Regression

Considering n observations with K groups, suppose there're $\boldsymbol{n}_k$ observations in the k-th group, while $\sum_{k=1}^{K} n_k \geq n$, so there could be overlappings. For the i-th data point, we observe the response y_i, a set of the index of groups s_i =

$\{k|$the i-th data point is in the k-th group$\} \subseteq \{1, \ldots, K\}$, i.e., the group membership is known, and covariate vector $\boldsymbol{X}_i$. We model the quantile regression by minimizing

$$L(\tau, \boldsymbol{\beta}) = \sum_{i=1}^{n} \sum_{k \in s_i} \left[f(\tau_k)(y_i - \boldsymbol{X}_i \boldsymbol{\beta}_k)^+ + g(\tau_k)(-y_i + \boldsymbol{X}_i \boldsymbol{\beta}_k)^+ \right],$$

where $(x)^+ = max\{x, 0\}$ and (differentiable) functions f and g. We denote $\tau = (\tau_1, \tau_2, \cdots, \tau_K)$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_k)$ as the quantile and the regression parameters for the k-th group respectively..

We call this method quantile regression because it gives an inner perspective of how to model the data quantile by τ. Although we usually given τ at a level of 0.5, it can be taken as a parameter, i.e., for some specific dataset, 75-th quantile model could be better than a median model for robustness and better prediction. Our goal is to get

$$(\widehat{\tau}, \widehat{\boldsymbol{\beta}}) = \underset{\tau, \boldsymbol{\beta}}{\operatorname{argmin}} \, L_\omega(\tau, \boldsymbol{\beta}).$$

The total weighted loss can be easily derived by adding weight to $L_\omega(\tau, \boldsymbol{\beta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \omega_k L_{k,i}$, where ω_k is given weight of the k-th group and $\sum_{k=1}^{K} \omega_k = 1$, and solve

$$(\widehat{\tau}, \widehat{\boldsymbol{\beta}}) = \underset{\tau, \boldsymbol{\beta}}{\operatorname{argmin}} \, L_\omega(\tau, \boldsymbol{\beta})$$

for the parameter estimate. It is obvious that $\partial L / \partial \tau$ exists with closed form but $\partial L / \partial \boldsymbol{\beta}$ doesn't exist due to non-differentiability, so τ plays as $\boldsymbol{\theta}$, and $\boldsymbol{\beta}$ plays as $\boldsymbol{Z}$.

As an illustrative example, let's consider the case when $K = 1$ with $\omega_K = \omega_1 = 1$, i.e., only one group with unknown quantile parameter $\tau_1 := \tau$, i.e. the obejctive function reduces to

$$L(\tau, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left[f(\tau)(y_i - \boldsymbol{X}_i \boldsymbol{\beta})^+ + g(\tau)(-y_i + \boldsymbol{X}_i \boldsymbol{\beta})^+ \right],$$

for some (differentiable) functions f and g. For example, setting $f(\tau) = e^\tau / (1 + e^\tau)$ and $g(\tau) = \Phi(\tau)$, where $\Phi(x)$ is the cdf of standard normal distribution. We aim to find a quantile that fits the data best without assuming the underlying noise structure for robustness. Note that we are not going to consider the sub-gradient of $\boldsymbol{\beta}$. Moreover, the above problem could also be generalized to the joint quantile regression model (Sangnier et al. 2016), which models multiple quartiles simultaneously. Another generalization is modeling the joint conditional quartile of multivariate response (Petrella and Raponi 2019).

D.2 Continuous Change Point Detection

A more challenging problem is change point model with continuous expectation. Here is the model:

$$Y_i(t) = \boldsymbol{X}_i(t) \boldsymbol{\beta}_t + \epsilon_i,$$

where $\boldsymbol{\beta}_t = \boldsymbol{\beta}_k$ when $\tau_k \leq t \leq \tau_{k+1}$, $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ with $-\infty := \tau_1 < \tau_2 < \cdots < \tau_K := \infty$ and K unknown. Since we have sampling over time, we have to impose additional conditions allowing change point at continuous scale. The nature choice is continuous expectation at change points, i.e.,

$$\boldsymbol{X}_i(\tau_{k+1})\boldsymbol{\beta}_k = \boldsymbol{X}_i(\tau_{k+1})\boldsymbol{\beta}_{k+1},$$

for all i and k. Indeed, the parameter estimation given τ's would still require constrained (numerical) optimization techniques. The choice of K would also required some information criterion, such as the Bayesian information criterion (BIC) and Minimum Description Length (MDL), see Zhang et al. (2023) for details about information criteria.

References

Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. Mathematical Programming **137**(1–2), 91–129 (2013) https://doi.org/10.1007/s10107-011-0484-9

Asakawa, K., Hirano, Y., Tan, K.-T., Ogasawara, T.: Bio-inspired study of stiffener arrangement in composite stiffened panels using a Voronoi diagram as an indicator. Composite Structures **327**, 117640 (2024) https://doi.org/10.1016/j.compstruct.2023.117640

Almeida, L.S., Rios, P.R.: Insights on the use of genetic algorithm to tessellate Voronoi structures in materials science. Journal of Materials Research and Technology **34**, 449–462 (2025) https://doi.org/10.1016/j.jmrt.2024.12.063

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. NIPS'11, pp. 2546–2554. Curran Associates Inc., Red Hook, NY, USA (2011)

Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American Statistical Association **112**(518), 859–877 (2017) https://doi.org/10.1080/01621459.2017.1285773 https://doi.org/10.1080/01621459.2017.1285773

Bai, J., Perron, P.: Estimating and testing linear models with multiple structural changes. Econometrica **66**(1), 47–78 (1998) https://doi.org/10.2307/2998540

Cauchy, A.: Méthode générale pour la résolution des systèmes d'équations simultanées. Comptes Rendus de l'Académie des Sciences **25**, 536–538 (1847)

Cerf, R.: Asymptotic convergence of genetic algorithms. Advances in Applied Probability **30**(2), 521–550 (1998) https://doi.org/10.1239/aap/1035228082

Cai, Y., Ma, T.F.: Maximum Ideal Likelihood Estimation: A Unified Inference Framework for Latent Variable Models (2025). https://arxiv.org/abs/2410.01194

Chen, M., Wu, X., Liu, Q., He, T., Ong, Y.-S., Jin, Y., Lao, Q., Yu, H.: Voronoi-grid-based pareto front learning and its application to collaborative federated learning. In: Proceedings of the 42nd International Conference on Machine Learning. ICML'25, pp. 9334–9353 (2025)

Devroye, L., Györfi, L., Lugosi, G., Walk, H.: On the measure of voronoi cells. Journal of Applied Probability **54**(2), 394–408 (2017) https://doi.org/10.1017/jpr.2017.7

Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(61), 2121–2159 (2011)

Das, S., Nandy, A., Sarvottamananda, S.: Radius, diameter, incenter, circumcenter, width and minimum enclosing cylinder for some polyhedral distance functions. Discrete Applied Mathematics **305**, 311–328 (2021) https://doi.org/10.1016/j.dam.2020.10.021

Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-44874-8

Fontalvo García, J.S., Moya Olivares, M.J., Gomez, A., Villalba Morales, J.D.: A genetic algorithm-based methodology for the structural optimization of Voronoi flat roofs. Applied Soft Computing **171**, 112742 (2025) https://doi.org/10.1016/j.asoc.2025.112742

Freidlin, M.I., Wentzell, A.D.: Random Perturbations of Dynamical Systems. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25847-3

Ghafour, P., Tavasoli, S., Shojaei, M.R.: VEGA: Voids identification using genetic algorithm. Journal of Cosmology and Astroparticle Physics **2025**(12), 001 (2025) https://doi.org/10.1088/1475-7516/2025/12/001 arXiv:2501.16431 [astro-ph.IM]

Hansen, L.P.: Large sample properties of generalized method of moments estimators. Econometrica **50**(4), 1029–1054 (1982) https://doi.org/10.2307/1912775

Huber, P.J.: Robust estimation of a location parameter. The Annals of Mathematical Statistics **35**(1), 73–101 (1964) https://doi.org/10.1214/aoms/1177703732

Koenker, R., Bassett, G.J.: Regression quantiles. Econometrica **46**(1), 33–50 (1978) https://doi.org/10.2307/1913643

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International

Conference on Learning Representations (2015). arXiv:1412.6980

Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review **45**(3), 385–482 (2003) https://doi.org/10.1137/S003614450242889

Li, S.: Volume decay and concentration of high-dimensional euclidean balls – a pde and variational perspective. Analysis **41**(1), 25–29 (2021) https://doi.org/10.1515/anly-2020-0035

Ling, S.: Estimation of change-points in linear and nonlinear time series models. Econometric Theory **32**(2), 402–430 (2016)

Lin, C.-C., Tung, C.-C., Chuang, Y.-Y., Chen, P.-Y.: Bio-inspired structural optimization of three-dimensional Voronoi structures using genetic algorithms: Inspirations from avian wing bones. Materials & Design **248**, 113501 (2024) https://doi.org/10.1016/j.matdes.2024.113501

Manski, C.F.: Maximum score estimation of the stochastic utility model of choice. Journal of Econometrics **3**(3), 205–228 (1975) https://doi.org/10.1016/0304-4076(75)90032-9

Ma, T.F., Yau, C.Y.: A pairwise likelihood-based approach for changepoint detection in multivariate time series models. Biometrika **103**(2), 409–421 (2016) https://doi.org/10.1093/biomet/asw002

Petrella, L., Raponi, V.: Joint estimation of conditional quantiles in multivariate linear regression models with an application to financial distress. Journal of Multivariate Analysis **173**, 70–84 (2019) https://doi.org/10.1016/j.jmva.2019.02.008

Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization **56**(3), 1247–1293 (2013) https://doi.org/10.1007/s10898-012-9951-y

Richter, W.-D., Schicker, K.: Simulation of polyhedral convex contoured distributions. Journal of Statistical Distributions and Applications **4**(1), 1 (2017) https://doi.org/10.1186/s40488-017-0055-6

Sangnier, M., Fercoq, O., d'Alché-Buc, F.: Joint quantile regression in vector-valued rkhss. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16, pp. 3700–3708. Curran Associates Inc., Red Hook, NY, USA (2016)

Shimosaka, H., Hiroyasu, T., Miki, M.: Voronoi model-building genetic algorithm. In: IEEE Conference on Cybernetics and Intelligent Systems, 2004., vol. 1, pp. 584–5891 (2004). https://doi.org/10.1109/ICCIS.2004.1460481

Suzuki, A., Nakatani, H., Nakagawa, S., Kobashi, M., Tsuji, Y.: Architected materials informatics: Construction and application to cellular-structured heat sink optimization. Acta Materialia **283**, 120557 (2025) https://doi.org/10.1016/j.actamat.2024.120557

Tian, G.-L., Fang, H.-B., Tan, M., Qin, H., Tang, M.-L.: Uniform distributions in a class of convex polyhedrons with applications to drug combination studies. Journal of Multivariate Analysis **100**(8), 1854–1865 (2009) https://doi.org/10.1016/j.jmva.2009.02.011

Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996) https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Tung, C.-C., Lai, Y.-Y., Chen, Y.-Z., Lin, C.-C., Chen, P.-Y.: Optimization of mechanical properties of bio-inspired Voronoi structures by genetic algorithm. Journal of Materials Research and Technology **26**, 3813–3829 (2023) https://doi.org/10.1016/j.jmrt.2023.08.210

Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Applications **109**(3), 475–494 (2001) https://doi.org/10.1023/A:1017501703105

Wu, C., Liang, K., Sang, H., Ye, Y., Pan, M.: A low-sample-count, high-precision Pareto front adaptive sampling algorithm based on multi-criteria and Voronoi. Soft Computing **28**, 5881–5897 (2024) https://doi.org/10.1007/s00500-023-09464-3

Xie, K., Cai, C., Yang, Z., Pan, J.: Application of improved genetic algorithm based on Voronoi partitioning in pseudolite deployment for tunnel positioning systems. Sensors **26**(9), 2596 (2026) https://doi.org/10.3390/s26092596

Ypma, T.J.: Historical development of the newton–raphson method. SIAM Review **37**(4), 531–551 (1995) https://doi.org/10.1137/1037125

Zeiler, M.D.: ADADELTA: An adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)

Zhang, J., Yang, Y., Ding, J.: Information criteria for model selection. WIREs Computational Statistics **15**(5), 1607 (2023) https://doi.org/10.1002/wics.1607