POLITECNICO DI TORINO
Repository ISTITUZIONALE

A multi-objective Gaussian mixture-based estimation of distribution algorithm for structural design optimization

Original
A multi-objective Gaussian mixture-based estimation of distribution algorithm for structural design optimization / Siviero, M., Melchiorre, J., Rosso, M.M., Cirrincione, G., Marano, G.C.. - In: APPLIED SOFT COMPUTING. - ISSN 1568-4946. - ELETTRONICO. - 201:(2026), pp. 1-37. [10.1016/j.asoc.2026.115610]

Availability:
This version is available at: 11583/3015686 since: 2026-09-17T17:34:43Z

Publisher:
Elsevier

Published
DOI:10.1016/j.asoc.2026.115610

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

19 November 2026

Applied Soft Computing Journal 201 (2026) 115610

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multi-objective Gaussian mixture-based estimation of distribution algorithm for structural design optimization

Mattia Siviero [a], Jonathan Melchiorre [a,*], Marco M. Rosso [b], Giansalvo Cirrincione [c], Giuseppe C. Marano [a,d]

[a] DISEG, Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, 10129, Italy
[b] CAE, Department of Civil and Architectural Engineering, Aarhus University, Inge Lehmanns Gade 10, Navitas, Aarhus C, 8000, Denmark
[c] Laboratoire des Technologies Innovantes, Université de Picardie Jules Verne, 51 Boulevard de Châteaudun, Amiens, 80000, France
[d] School of Civil Engineering and Architecture, Henan University of Technology, 100 Lianhua St, Zhongyuan District, Zhengzhou, Henan, 450001, China

HIGHLIGHTS

- Introduction of MOEDA-GMM, a multi-objective optimizer using Gaussian Mixture Models (GMM).
- The method is a metaheuristic Estimation of Distribution Algorithm (EDA).
- MOEDA-GMM is specifically designed for constrained structural optimization problems.
- The method is tested against the state-of-the-art multi-objective optimization algorithms.
- A practical tool for structural optimization in early stage design exploration.

ARTICLE INFO

Keywords:
Structural optimization
Multi-objective optimization
Estimation of distribution algorithm (EDA)
Gaussian mixture model (GMM)
Metaheuristic algorithms
Evolutionary computation

ABSTRACT

Structural design optimization in the Architecture–Engineering–Construction (AEC) domain often involves complex, constrained and computationally intensive tasks. These problems are typically formulated as constrained multi-objective optimization tasks, where conflicting objectives, such as cost, sustainability, and constructability, must be balanced subject to structural performance requirements. In this context, population-based metaheuristics are effective for navigating black-box and non-convex design spaces. Among these, Estimation of Distribution Algorithms (EDAs) are well suited for this purpose, as they guide search through learned probability distributions over high-performing regions. This study introduces MOEDA-GMM, a multi-objective EDA that models the selected population using a Gaussian Mixture Model fitted via Expectation-Maximization. A power-law covariance decay schedule governs the transition from exploration to exploitation. The algorithm is evaluated on four constrained structural benchmarks and three analytical test functions, and compared against NSGA-II, SMS-EMOA, and RVEA under equal evaluation budgets. A factorial sensitivity analysis of the three hyperparameters and non-parametric statistical tests support the comparison. MOEDA-GMM shows rapid early-stage convergence and significantly outperforms NSGA-II on two structural problems, while established algorithms retain a final-convergence advantage on the analytical suite. The sensitivity analysis identifies a compact region of robust settings across problems, confirming the viability of mixture-model EDAs for constrained structural optimization under limited evaluation budgets.

1. Introduction

The Architecture, Engineering and Construction (AEC) industry has a substantial environmental footprint, accounting for approximately 34% of energy-related CO_2 emissions and 32% of final energy consumption worldwide [1]. According to projections, the global building floor area will double by 2050, driven by population growth and rising living standards in developing countries [2].

These pressures affect ecosystems, climate stability, and public health alike. Yet the AEC industry also plays a pivotal role in the economy, accounting for around 10% of Gross Domestic Product (GDP). Within the European Union alone, the sector provides more than 18

* Corresponding author.
Email address: jonathan.melchiorre@polito.it (J. Melchiorre).

https://doi.org/10.1016/j.asoc.2026.115610
Received 12 May 2025; Received in revised form 5 May 2026; Accepted 20 May 2026
Available online 5 June 2026
1568-4946/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

million direct jobs and contributes 9% to the total GDP [3]. The combined environmental and economic weight of the sector means that even incremental design improvements can yield considerable benefits at scale.

At the same time, the sector is characterized by inherently complex decision-making, where designers are required to carefully balance trade-offs between structural performance, safety, cost-efficiency, maintainability and sustainability. Design choices must satisfy multiple objectives and adhere to stringent requirements imposed by physical laws, regulatory codes and project specifications. In this context, structural analysis establishes the safety boundaries of the design space. It does so by defining inviolable constraints: any feasible solution must satisfy equilibrium conditions, material strength limits, stability criteria and serviceability requirements. These constraints shape the feasible region and control the admissibility of candidate solutions throughout the optimization process.

Verifying that a candidate design satisfies these constraints requires a full structural evaluation, which is rarely available in closed form. Analytical solutions are restricted to simplified configurations and boundary conditions, limiting their applicability to practical problems. Before the widespread adoption of numerical simulation, assessing structural behavior under varying loads, geometries, and material properties was largely infeasible. Consequently, early-stage design decisions relied on heuristic reasoning and approximate methods, often producing conservative or suboptimal outcomes.

The introduction of digital computing, followed by the development of numerical analysis techniques such as the Finite Element Method (FEM) [4,5], represented a critical milestone in the field. FEM enabled the modeling of complex structural systems [6,7] and the systematic evaluation of designs under realistic conditions, including non-uniform loading, material heterogeneity, complex boundary conditions and nonlinearities [8]. However, each FEM evaluation is computationally expensive, and iterative optimization requires many such evaluations across candidate solutions, severely limiting the number of configurations that can be explored within a given computational budget [9].

Modern structural design must also address multiple, often conflicting objectives simultaneously [10]: a design must not only be structurally efficient and compliant, but also cost-effective [9], environmentally sustainable, easy to construct with available technologies and materials [11], energy-efficient, compatible with the surrounding architectural context [12,13], amenable to monitoring [14] and maintenance throughout its service life [15], and adaptable to potential changes in use while complying with regulatory requirements and project-specific constraints [16].

Traditional optimization methods often prove inadequate in addressing the complexity and multi-objective nature of modern structural design problems. Most are limited to single-objective formulations and rely on restrictive assumptions, such as continuity, differentiability or linearity [17]. Gradient Descent [18,19] and Conjugate Gradient [20] methods require first-order derivatives, which may be unknown or undefined in structural optimization. They often perform poorly in constrained search spaces or in the presence of local minima. Linear Programming (LP) methods, including the Simplex algorithm [21], assume linearity in both the objective and constraint functions, which rarely hold in structural problems. Pattern search algorithms [22], while suited to black-box scenarios, are often inefficient in high-dimensional design spaces [23].

The limitations of traditional optimization techniques, coupled with the inherent multi-objective nature of AEC problems, have led to increasing adoption of population-based metaheuristic methods [24]. These approaches rely on pointwise evaluations of the objective functions and are considered effective in non-convex, non-smooth or constrained spaces [24,25].

Particle Swarm Optimization (PSO) [26] simulates search space trajectories of a population of candidate solutions (particles). Each particle updates its velocity through a weighted combination of the current value and a function of its current coordinates in relation to the personal and global best-known positions. PSO typically exhibits rapid convergence in smooth, continuous search spaces, but often performs poorly in discontinuous search spaces, which are common in constrained structural optimization [27].

Algorithm 1 NSGA-II: Non-dominated sorting genetic Algorithm II.

Require: Population size N, number of generations G

Initialize population P_0 of size N
Evaluate objective functions $F(x)$ for all $x \in P_0$
for $t = 0$ to $G - 1$ **do**
  $Q_t \leftarrow$ Generate offspring from P_t using:
    - Tournament selection (based on Pareto rank and crowding distance)
    - Crossover and mutation
  Evaluate $F(x)$ for all $x \in Q_t$
  $R_t \leftarrow P_t \cup Q_t$
  Perform non-dominated sorting on R_t to obtain fronts $F_0, F_1, \ldots$
  $P_{t+1} \leftarrow \emptyset$
  $i \leftarrow 0$
  while $|P_{t+1}| + |F_i| \leq N$ **do**
    Assign crowding distance to all solutions in F_i
    $P_{t+1} \leftarrow P_{t+1} \cup F_i$
    $i \leftarrow i + 1$
  end while
  Sort F_i by descending crowding distance
  Add the top $(N - |P_{t+1}|)$ individuals from F_i to P_{t+1}
end for
return Non-dominated solutions in the final population P_G (i.e., F_0)

Evolutionary algorithms (EAs) iteratively improve a population of candidate solutions by generating new candidates from the existing ones and selecting a subset of the combined pool that maximizes a chosen performance indicator. Since the previous population can always be retained, the indicator value is guaranteed to improve or remain constant over successive iterations.

Genetic Algorithms (GAs) [28,29] are a subset of EAs inspired by evolutionary biology. They implement an elitist selection by assigning individual fitness values to each solution and discarding the worst-performing. New candidate solutions (offspring) are generated by applying a binary recombination operator (crossover) to pairs of high-performing solutions (parents). Additional variability is introduced through stochastic mutation. Their ability to maintain a diverse population makes them well-suited for extension to multi-objective optimization (MOO) tasks [25,30].

A central concept in MOO is the Pareto front, which represents the set of non-dominated solutions, i.e., those for which no objective can be improved without degrading at least one other. To approximate this front, various evolutionary algorithms have been developed that guide population evolution by balancing convergence and diversity. For example, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [31] employs fast non-dominated sorting and crowding-distance mechanisms to maintain diversity across the front. Alternatively, the S-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) [32] focuses on maximizing the hypervolume indicator, directly promoting both convergence and diversity. Other methods, such as MCDM-GA [33], extend genetic algorithm frameworks by incorporating decision-maker preferences, often supported by trainable models like Radial Basis Function Networks (RBFNs).

Traditional genetic algorithms (GAs) rely heavily on recombination operators, particularly crossover, to combine previously evaluated candidate solutions to generate new ones. The offspring produced by

Algorithm 2 SMS-EMOA: S-Metric selection evolutionary multi-objective algorithm.

Require: Population size N, maximum number of iterations S, reference point r

Initialize population P with N random solutions
Evaluate objectives $F(x)$ for all $x \in P$
for $s = 1$ to S **do**
  Select two parents via binary tournament on Pareto dominance
  Generate offspring x_{new} by crossover and mutation
  Evaluate $F(x_{\text{new}})$
  $P \leftarrow P \cup \{x_{\text{new}}\}$
  Perform non-dominated sorting on P to obtain fronts $F_0, F_1, \ldots, F_L$
  Let F_L be the worst-ranked (last) front
  for each $x \in F_L$ **do**
    Compute contribution on the last front:

$$\Delta HV(x) = HV(F_L) - HV\left(F_L \setminus \{x\}\right)$$

  end for
  Remove

$$x^* = \arg\min_{x \in F_L} \Delta HV(x)$$

  from P
end for
return First Pareto front F_0 in the final population P

crossover typically remain within the region defined by the parent solutions in the search space. Mutation operators are introduced to provide stochastic variability and extend the search beyond the limits imposed by crossover. However, this exploration is mostly random. Estimation of Distribution Algorithms (EDAs) take a different approach as an alternative class of evolutionary methods [34–36]. Rather than relying on fixed genetic operators, EDAs explicitly model the distribution of high-quality solutions using probabilistic models defined over the entire search space. New candidate solutions are then sampled from this distribution, allowing for the exploration of both local and global regions according to their learned probability. The iterative refinement of the distribution guides the population toward high-performing regions. By modeling the solution space structure, EDAs capture variable interactions more effectively than standard Genetic Algorithms, which apply fixed recombination rules. The main downside is higher computational cost and the need for larger input samples.

Univariate EDAs represent candidate solutions by assuming independence among variables. This leads to probabilistic models expressed as products of independent distributions, making univariate EDAs particularly efficient for solving separable problems.

In contrast, multivariate EDAs are capable of capturing dependencies between variables, allowing them to address more complex and non-separable optimization tasks. These methods range from relatively simple pairwise models, such as the Bivariate Marginal Distribution Algorithm (BMDA) [37], to more expressive probabilistic graphical models. Notably, algorithms like the Hierarchical Bayesian Optimization Algorithm (hBOA) [38] employ Bayesian networks to represent higher-order and nonlinear dependencies in the objective landscape. While these models improve the ability of the algorithm to navigate complex search spaces, they usually require larger population sizes and involve higher computational costs.

Updating a single multivariate distribution can be effective when optimizing over smooth and unimodal landscapes. However, in multimodal settings, this approach often leads to premature convergence, as the model tends to focus on a single region of the search space. Mixture models [39] address this limitation by simultaneously updating multiple probabilistic components, enabling exploration across diverse regions of the solution space. This significantly enhances modeling flexibility, but also increases computational complexity. The effectiveness of such models depends on maintaining meaningful separation among components, which requires appropriate strategies to prevent mode collapse or redundancy.

Unimodal EDAs are particularly vulnerable in constrained optimization scenarios. As the search narrows, the learned distribution may extend into infeasible regions, making it difficult to accurately capture narrow feasible basins near the optimum. In contrast, multimodal EDAs offer greater adaptability to complex constraint landscapes and are therefore more suitable for irregular, fragmented, or disjoint feasible domains.

Multi-Objective EDAs (MOEDAs) must carefully balance convergence and diversity in order to approximate the Pareto front effectively. Several strategies have been proposed to address this challenge. Tian et al. [40] employ Markov network models to capture dependencies among decision variables and multiple objectives simultaneously. Yang et al. [41] introduce a hybrid approach that combines global exploration, guided by a probabilistic model, with local refinement using an incremental tournament-based local searcher.

A class of MOEDAs exploits a structural property of continuous multi-objective problems. Under mild smoothness conditions, the Karush-Kuhn-Tucker conditions [42] imply that the Pareto set is a piecewise continuous $(m-1)$-dimensional manifold in decision space, where m is the number of objectives. This is known as the regularity property of continuous multi-objective optimization. Algorithms in this class estimate the manifold from the current population, and then generate offspring by sampling points on the estimate and applying a small random perturbation, rather than fitting a probability density over the population itself. Zhang et al. [43] introduced this approach with RM-MEDA. The algorithm partitions the population through local principal component analysis and approximates each cluster by an affine $(m-1)$-dimensional patch of the manifold. New solutions are then drawn uniformly on the patches and perturbed by isotropic Gaussian noise.

Lu et al. [44] survey the resulting literature and organise it into three groups. The first group constructs a manifold approximation and samples offspring on or near it. The second group uses the manifold to define neighbourhoods of similar solutions and restricts mating to within those neighbourhoods. The third group uses the approximation to produce a small set of guiding solutions that steer the offspring operators of host algorithms such as differential evolution or particle-swarm optimization. Within the first group, Jiang et al. [45] replace the fixed Gaussian perturbation of RM-MEDA with a time-decaying schedule that is broad in the early stages of the search and contracts as the population converges. Wang and Zhou [46] unify the existing variants under the Regularity Evolution paradigm, in which every offspring is the sum of a structure vector that lies on the learned manifold and a perturbation vector that displaces it from the manifold. Within the second group, Pan et al. [47] approximate manifold distance through objective-space proximity and use it to restrict mating in otherwise standard MOEAs. Gao et al. [48] apply the same restriction strategy in both objective and decision space, defining objective-space neighbourhoods from idealised sub-Pareto fronts centred on knee and extreme points and decision-space neighbourhoods from elitist-seeded clustering.

Mixture models have also proven effective in maintaining population diversity when dealing with multimodal trade-off surfaces. Their ability to model multiple regions of interest within the search space is further enhanced when combined with diversity-preserving mechanisms that promote meaningful separation between components or clusters.

MOEDAs are particularly advantageous in problems involving complex variable interactions and conflicting objectives. However, their success depends critically on maintaining a balance between model

representativeness, computational overhead, and the population size required for reliable statistical learning.

This paper presents MOEDA-GMM, a Gaussian Mixture Model (GMM)-based MOEDA [39]. The algorithm integrates tournament-based parent selection [49] with probabilistic modeling via Expectation-Maximization (EM) [50]. A key feature of MOEDA-GMM is the implementation of a controlled covariance decay mechanism, which progressively reduces exploration and enhances exploitation over the course of the search. Unlike the regularity-based methods reviewed above, MOEDA-GMM does not assume a manifold approximation of the Pareto set; in this respect it is more closely related to the earlier mixture-based multi-objective EDA of Bosman and Thierens [51].

To preserve diversity, the selection process imposes a user-defined upper bound on the number of individuals retained from each Pareto front. This prevents overconcentration on the first front and promotes a more balanced distribution of solutions across multiple fronts. The use of GMM [39] allows for the modeling of disjoint or irregular Pareto regions and facilitates sampling near feasible boundaries, owing to the nonlocal support of Gaussian distributions. These characteristics support a more extensive exploration of the search space.

Given its probabilistic nature and modeling flexibility, MOEDA-GMM is particularly well suited for expensive structural optimization problems, where function evaluation cost constitutes the majority of the computational budget.

The proposed algorithm is benchmarked against NSGA-II, SMS-EMOA, and the Reference Vector Guided Evolutionary Algorithm (RVEA) on four structural optimization problems (welded beam, two-bar truss, four-bar truss, and Warren truss) and three analytical test functions (ZDT1–3). Results indicate that MOEDA-GMM offers competitive performance and effective diversity management, confirming the potential of probabilistic, mixture-based EDAs in structural multi-objective optimization. The study also identifies opportunities for improvement, particularly in diversity preservation and adaptive model control, which are relevant to ensuring robust convergence in constrained design spaces.

This work makes two main contributions. First, it presents an estimation-of-distribution approach (MOEDA-GMM) adapted to structural optimization problems characterized by high evaluation costs and limited sampling budgets. The algorithm employs an initial covariance inflation followed by a controlled power-law decay, allowing broad initial exploration that gradually concentrates sampling around high-performing regions as convergence proceeds. Second, it provides a comparative analysis against NSGA-II, SMS-EMOA, and RVEA on analytical and structural benchmarks. Four indicators are reported: hypervolume (HV), generational distance plus (GD^+), spread (Δ), and uniform distribution error (UDE). The discussion considers the implications for integration into computationally expensive structural optimization workflows.

This paper is structured as follows. Section 2 provides background on multi-objective optimization and genetic algorithms, followed by a description of the baseline methods: NSGA-II, SMS-EMOA and RVEA, in Section 2.1. Section 2.2 presents the proposed MOEDA-GMM algorithm in detail, including its overall structure, the EM procedure for model fitting and the minimum covariance scheduling strategy used to balance exploration and exploitation.

Section 3 reports the results of a comparative evaluation on analytical and structural optimization benchmarks, using four performance indicators. A real-world case study involving the optimization of a Warren truss is also included. Finally, Section 4 summarizes the key findings, discusses current limitations and outlines potential directions for future research.

2. Methodology

MOO aims to simultaneously optimize multiple conflicting objective functions over a common set of decision variables. A general MOO problem can be formally stated as follows:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{F}(\boldsymbol{x}) = \left[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\right] \\ \text{s.t.} \quad & \begin{cases} g_i(\boldsymbol{x}) \le 0, & i = 1, \ldots, I \\ h_j(\boldsymbol{x}) = 0, & j = 1, \ldots, J \\ \boldsymbol{x} \in \mathcal{X} \end{cases} \end{aligned} \tag{1}$$

Here, $\boldsymbol{x} \in \mathbb{R}^n$ denotes the decision vector within the design space $\mathcal{X} \subseteq \mathbb{R}^n$, and $\boldsymbol{F}(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^m$ is a vector of m scalar objective functions $f_k : \mathcal{X} \to \mathbb{R}$ to be minimized. The scalar inequality constraints $g_i(\boldsymbol{x}) \le 0$ and equality constraints $h_j(\boldsymbol{x}) = 0$ define the feasible region within the search space. The corresponding objective space is denoted by $\mathcal{Y} = \boldsymbol{F}(\mathcal{X})$.

Due to the presence of conflicting objectives, no single optimal solution generally exists. Instead, the goal is to approximate the Pareto front: the set of non-dominated solutions for which no objective can be improved without degrading at least one other.

Each design vector $\boldsymbol{x} \in \mathcal{X}$ is evaluated across multiple objectives, resulting in a point in a multi-dimensional objective space. When objectives are conflicting, their global minima occur at different values of $\boldsymbol{x}$, and cannot be simultaneously minimized. In such cases, optimization does not yield a single optimal solution, but rather a set of trade-offs, where improving one objective necessarily worsens at least one other.

To handle this, the optimization process aims to approximate a diverse set of high-quality trade-off solutions. This set is formally defined through the concept of Pareto dominance. Given two feasible solutions $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$, $\boldsymbol{x}_1$ is said to Pareto-dominate $\boldsymbol{x}_2$ (denoted $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$) if:

$$\begin{cases} f_i(\boldsymbol{x}_1) \le f_i(\boldsymbol{x}_2) \quad \forall i \in \{1, \ldots, m\}, \\ \exists j \in \{1, \ldots, m\} \quad \text{such that} \quad f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2) \end{cases} \tag{2}$$

A feasible solution is said to be Pareto-optimal if it is not dominated by any other feasible solution. The set of all such non-dominated solutions defines the Pareto-optimal set, and its image in the objective space forms the Pareto front.

The Pareto optimal set P^* is defined as the set of all Pareto-optimal solutions $\boldsymbol{x}_p$:

$$P^* = \left\{ \boldsymbol{x}_p \in \mathcal{X} \;\middle|\; \nexists\, \boldsymbol{x} \in \mathcal{X} \text{s.t. } \boldsymbol{x} \succ \boldsymbol{x}_p \right\} \tag{3}$$

This set consists of all non-dominated solutions. The corresponding objective space representation is the Pareto front $\boldsymbol{F}(P^*)$:

$$\boldsymbol{F}(P^*) = \left\{ \boldsymbol{F}(\boldsymbol{x}_p) \in \mathcal{Y} \;\middle|\; \boldsymbol{x}_p \in P^* \right\} \tag{4}$$

A central challenge in multi-objective optimization is to balance fast convergence toward the Pareto-optimal set with the need to obtain a diverse set of solutions that effectively covers the full extent of the Pareto front.

A common but limited strategy for generating Pareto-optimal solutions is scalarization through weighted sums. This approach reduces a multi-objective problem to a series of single-objective optimization problems by assigning fixed weights to each objective function. The scalarized problem can be formulated as:

$$\boldsymbol{x}_P = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{m} w_i f_i(\boldsymbol{x}) \tag{5}$$

where $w_i \ge 0$ and $\sum_{i=1}^{m} w_i = 1$. Different weight vectors lead to different trade-off solutions.

While simple to implement, this method presents several drawbacks. Each scalarized problem must be solved independently, making dense sampling of the Pareto front computationally inefficient. More critically, this approach cannot capture non-convex regions of the Pareto front, as it inherently favors convex combinations of objectives.

Genetic algorithms (GAs) mathematically model the process of natural selection, with design variables representing genetic traits and objective values corresponding to fitness. At each iteration, the original solutions are referred to as parent solutions, while the newly generated ones are termed offspring solutions. This approach follows the heuristic that the recombination of higher-fitness solutions within a population is more likely to produce offspring with improved performance. Additionally, a random mutation probability of traits introduces stochasticity, preventing premature convergence and allowing the algorithm to escape local optima. Typically, the initial population of candidate solutions is sampled randomly. Then, the main iteration loop alternates between two main steps. In the elitist selection step, objective values and constraints (if present) are evaluated and the worst-performing solutions are discarded. In the generation step, offspring are generated through binary recombination of high-performing parents and random mutation. These steps are repeated until a convergence criterion or iteration limit is reached.

In multi-objective optimization, a global ordering of the solutions based on a single metric is not possible and a fitness indicator must guide convergence while maintaining a diverse enough population to ensure representative coverage. NSGA-II [31] uses a fast non-dominated sorting algorithm to rank solutions based on their level of non-domination. Solutions in each front are ranked according to their crowding distance, a diversity-preserving criterion that favors more isolated solutions. SMS-EMOA [32] adds a single solution at each step through binary crossover of a selected parent pair. The solution with the lowest hypervolume contribution is then removed.

In constrained multi-objective optimization, algorithms must satisfy feasibility requirements while optimizing conflicting objectives. Constraints often fragment the search space and reshape the Pareto front, making constraint handling essential for finding valid, high-quality trade-offs.

Evolutionary algorithms employ several general strategies to address constraints. One common approach adds penalty terms to the objective function, discouraging violations [52]. Another prioritizes feasible individuals during selection; for example, NSGA-II always ranks any feasible solution above infeasible ones and compares infeasible solutions by degree of violation [31]. A third method uses repair operators that adjust infeasible candidates to satisfy constraints, preserving useful solution traits while restoring validity [52].

EDAs, which construct probabilistic models from high-quality solutions, require careful adaptation to constrained optimization. In particular, without explicit handling of constraints, the learned models may assign high probability density to infeasible regions of the search space. This motivates two requirements: the need to guide the search toward a fully feasible population when the initial samples are infeasible, and the importance of progressively reducing the probability of sampling infeasible solutions as the search converges toward a well-defined feasible region. The first requirement can be addressed through feasibility-aware model updates, in which samples are weighted based on their feasibility status. In this approach, feasible and near-feasible solutions are given higher influence during model construction, promoting convergence toward the feasible region. An alternative is to adopt a two-phase strategy: the first phase focuses on minimizing constraint violations to achieve feasibility, while the second shifts attention to optimizing the objective functions once a feasible region has been reached [35].

The second requirement focuses on limiting the generation of infeasible solutions during model-driven sampling. This can be addressed either through rejection sampling or repair mapping. In rejection sampling, candidate solutions that violate constraints are discarded and resampled, which is conceptually simple but may result in inefficiencies, especially when constraint violations are frequent and the total number of iterations is limited by a fixed computational budget. In contrast, repair mapping attempts to project infeasible samples back into the feasible domain. While this approach avoids wasting samples, it relies on an accurate approximation of the constraint boundary. For black-box problems, this is only efficient if constraint evaluations are substantially cheaper than objective function evaluations.

Shirazi et al. [53] adopted the repair-based approach in EDA++, enforcing feasibility at every generation by ensuring that all offspring remain within the admissible region throughout the optimization process.

Flexible probabilistic models can also contribute to constraint handling by reducing the likelihood of sampling infeasible solutions. In particular, GMMs are more effective than unimodal models in representing disjoint or irregular feasible regions, as they can assign separate components to distinct feasible zones [35]. Shirazi et al. [53] demonstrate this by incrementally increasing the number of Gaussian components until each component's centroid is located within a feasible subregion.

Effective constraint handling in EDAs depends not only on general evolutionary strategies but also on the modeling capacity of the probabilistic framework itself. Selecting an appropriate EDA model can significantly improve the efficiency of feasible region exploration.

Recent metaheuristic developments in multi-objective structural optimization have also explored new biologically inspired strategies. Several studies, all published between 2023 and 2025, propose algorithmic variants focusing on population interaction or hybrid selection mechanisms. For instance, the Adaptive Predator–Prey Algorithm [54] introduces a co-evolutionary approach for maintaining diversity in many-objective optimization; the Multi-Objective Elk Herd Optimization [55] and the Crested Porcupines Optimization Algorithm [56] apply similar principles to structural design contexts, demonstrating the versatility of swarm-inspired formulations. Complementary formulations include a weighted-average method for truss optimization aimed at improving convergence and front distribution [57], a comparative study on a 120-bar dome truss using three metaheuristics [58], and a thermal–structural optimization scheme based on Thermal Exchange Optimization [59]. Together, these works broaden the repertoire of bio-inspired and population-based algorithms tested on structural benchmarks. Within this landscape, MOEDA-GMM contributes to the EDA paradigm by coupling Gaussian mixture modeling with a decaying covariance schedule, enabling progressive narrowing of the search distribution as convergence proceeds—a capability particularly valuable when function evaluations are expensive and feasible regions may be fragmented.

2.1. Baseline models

Three well-established multi-objective evolutionary algorithms serve as baselines: NSGA-II (non-dominated sorting with crowding-distance diversity), SMS-EMOA (hypervolume-based selection), and RVEA (decomposition via reference vectors). Together they cover the three dominant algorithmic paradigms in multi-objective optimization, enabling a direct assessment of MOEDA-GMM's probabilistic modeling strategy across diverse selection mechanisms.

NSGA-II [31] is one of the most widely used genetic algorithms for multi-objective optimization. Its core mechanism is a fast non-dominated sorting procedure that partitions the population into hierarchical Pareto fronts. This is achieved with a time complexity of $O(MN^2)$, where M is the number of objectives and N is the population size.

During sorting, two auxiliary data structures are initialized for each solution x_i: the domination count n_i, which stores how many solutions dominate x_i, and the domination set d_i, which holds the indices of solutions dominated by x_i. Initially, all n_i values are set to zero. The algorithm iterates over all pairs of solutions (x_j, x_k) to determine dominance relationships. If $x_j \succ x_k$, then n_k is incremented and k is added to d_j. The first Pareto front is formed by collecting all solutions with $n_i = 0$. These are added to front F_0, and their domination counts are set to -1 to mark them as processed. Then, for each solution in F_0, the algorithm decrements the domination count of its dominated solutions. The process is repeated iteratively to construct subsequent fronts $F_1, F_2, \ldots$, until all solutions are assigned to a front.

Once sorting is complete, solutions within each front are ranked by crowding distance, which serves as a secondary criterion to promote diversity. For each objective, values are normalized to the $[0, 1]$ range, and the crowding distance of a solution is computed as the average distance between its nearest neighbors across all objectives. Boundary solutions are assigned an infinite crowding distance to ensure their retention. Finally, parent selection is performed using tournament selection based on dominance rank and crowding distance, introducing stochasticity to help prevent premature convergence.

SMS-EMOA [32] is a steady-state evolutionary algorithm that uses the hypervolume indicator as a selection criterion for multi-objective optimization. The hypervolume of a Pareto front approximation is defined as the measure of the region in the objective space that is weakly dominated by the solution set and bounded by a reference point.

After initializing a random population and evaluating its objective values, SMS-EMOA computes the hypervolume of the current solution set and the marginal contribution of each individual. The marginal contribution of a solution x is defined as the difference between the total hypervolume $HV(P)$ and the hypervolume of the set obtained by excluding x, $HV(P \setminus \{x\})$:

$$\Delta HV(x) = HV(P) - HV(P \setminus \{x\}), \tag{6}$$

where P is the current population.

At each iteration, binary tournament selection is used to select two parents based on Pareto dominance. A single offspring x_{new} is generated through crossover and mutation and added to the population. The hypervolume is then updated, and the contribution of the new solution is computed as:

$$\Delta HV(x_{\text{new}}) = HV(P \cup \{x_{\text{new}}\}) - HV(P). \tag{7}$$

Next, non-dominated sorting is performed to identify the Pareto fronts within the updated population. Among the solutions in the last front, the individual with the smallest hypervolume contribution is removed to maintain a fixed population size. This procedure continues until a predefined stopping criterion is met, such as a maximum number of iterations.

To maintain computational efficiency, hypervolume values are updated incrementally. In the bi-objective case, Klee's Measure Algorithm (KMA), also known as the sweep line algorithm, is used for efficient computation. In this method, solutions are first sorted by their x-coordinate in $O(n \log n)$ time. A single scan then computes both the total hypervolume and the marginal contribution of each point. During this process, dominated points are discarded, and only the first Pareto front (i.e., points with non-zero contribution) is retained for further updates. Insertion operations require $O(\log n)$ time, and updates can be performed in $O(n)$, resulting in an overall complexity of $O(Sn)$, where S is the number of iterations.

RVEA [60] is a decomposition-style evolutionary algorithm that guides the search using a set of unit reference vectors that partition the objective space into subregions. In each generation, candidate solutions are associated with their nearest reference vector (by acute angle), and one elitist per subregion is retained according to an angle-penalized distance (APD) metric that balances convergence toward the ideal point and alignment with the assigned vector. In this work, the bi-objective version ($M = 2$) is employed, where N evenly spaced reference vectors are placed in the first quadrant of the unit circle.

After evaluating the population, objective values are first translated so that the ideal point becomes the origin:

$$\mathbf{f}'(x) = \mathbf{f}(x) - \mathbf{z}_{\min}, \quad \mathbf{z}_{\min} = \min_{x \in P} \mathbf{f}(x), \tag{8}$$

ensuring that all translated points lie in the positive quadrant and are covered by the reference vectors. Each solution is then associated with the closest reference vector $\mathbf{v}_j$ by maximizing the cosine similarity

$$\cos\theta(x, \mathbf{v}_j) = \frac{\mathbf{f}'(x) \cdot \mathbf{v}_j}{\|\mathbf{f}'(x)\|}. \tag{9}$$

Within each subpopulation, the selection criterion is the Angle-Penalized Distance:

$$d(x|j) = \Big(1 + \mathcal{P}(\theta(x, \mathbf{v}_j))\Big) \, \|\mathbf{f}'(x)\|, \tag{10}$$

where the penalty term $\mathcal{P}(\theta)$ controls the trade-off between convergence and diversity:

$$\mathcal{P}(\theta) = M \left(\frac{t}{t_{\max}}\right)^p \frac{\theta}{\gamma_{\mathbf{v}_j}}, \tag{11}$$

with t denoting the generation index, p a user-defined exponent controlling how diversity pressure increases over time, and $\gamma_{\mathbf{v}_j}$ the minimum angle between $\mathbf{v}_j$ and its neighboring vectors. Early in the run, $\mathcal{P}(\theta) \approx 0$ emphasizes convergence, while near the end of the search ($t \approx t_{\max}$) the penalty term grows, encouraging a more even spread of solutions. In the bi-objective case ($M = 2$) the pymoo implementation uses an effective multiplier of 1 rather than M, so the active penalty reduces to $\mathcal{P}(\theta) = (t/t_{\max})^p \, \theta/\gamma_{\mathbf{v}_j}$. To account for different objective scales, RVEA periodically adapts the reference vectors using the current objective ranges:

$$\mathbf{v}_i^{(t+1)} = \frac{\mathbf{v}_i^{(0)} \circ (\mathbf{z}_{\max}^{(t+1)} - \mathbf{z}_{\min}^{(t+1)})}{\left\| \mathbf{v}_i^{(0)} \circ (\mathbf{z}_{\max}^{(t+1)} - \mathbf{z}_{\min}^{(t+1)}) \right\|}, \tag{12}$$

where $\circ$ denotes the Hadamard (elementwise) product. This adaptation is performed at a fixed frequency to maintain stability while compensating for objective scaling effects.

The per-generation computational cost is $O(MN^2)$, dominated by partitioning and APD evaluation. For bi-objective problems ($M = 2$), this reduces to $O(N^2)$. Offspring are generated through standard genetic operators (SBX crossover and polynomial mutation), and the number of reference vectors is typically set equal to the population size so that one elitist solution is selected per vector.

2.2. MOEDA-GMM

This study introduces MOEDA-GMM, a bi-objective EDA that employs a GMM to statistically model and explore the solution space [35,36].

The primary objective is to replace mutation and crossover with a tractable probabilistic model capable of capturing the complexity and multimodality of the design landscape.

Algorithm 3 RVEA: Reference vector guided evolutionary algorithm (Bi-Objective).

Require: Population size N, number of generations G, exponent p, adaptation frequency f_r

Generate N unit reference vectors $V_0 = \{\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_N^{(0)}\}$ uniformly in $[0, \pi/2]$
Initialize population P_0 of size N; evaluate $F(x)$ for all $x \in P_0$
for $t = 0$ to $G - 1$ **do**
    $Q_t \leftarrow$ Generate offspring from P_t via crossover and mutation
    Evaluate $F(x)$ for all $x \in Q_t$; set $R_t \leftarrow P_t \cup Q_t$
    Compute $\mathbf{z}_{\min} = \min_{x \in R_t} F(x)$ and translate $\mathbf{f}'(x) = F(x) - \mathbf{z}_{\min}$
    Assign each x to the closest reference vector $\mathbf{v}_j$ by maximal $\cos\theta(x, \mathbf{v}_j)$
    Compute $d(x|j)$ for each subpopulation using Eq. (10)
    Select one elitist (lowest $d(x|j)$) per subpopulation to form P_{t+1}
    if t/G is a multiple of f_r **then**
        Update reference vectors $\mathbf{v}_i$ using Eq. (12)
    end if
end for
return Non-dominated solutions in the final population P_G

The algorithm adopts the non-dominated sorting and crowding distance mechanism from NSGA-II for environmental selection. However, the generation step differs substantially, replacing direct recombination with a probabilistic modeling approach.

Rather than producing offspring through variable-level recombination, MOEDA-GMM estimates the distribution of selected parent solutions using a GMM. This model is fitted via the EM algorithm to maximize the likelihood of the parent set. New solutions are then sampled from the learned distribution, enabling exploration guided by the underlying statistical structure.

The MOEDA-GMM algorithm is governed by two parameters that control population management: the minimum population size n_p and the buffer fraction vector $S_f = \{s_{f,i}\},\ i = 1, \dots, n_f$, where $n_f = |S_f|$ is the number of buffered fronts. The former defines the minimum number of selected candidate solutions and the size of the offspring at each iteration. The latter specifies the upper bound on the population size near convergence and its distribution across the fronts. Each entry expresses the maximum number of candidate solutions retained for the corresponding front as a fraction of n_p, yielding $N_{f,i} = S_{f,i} \cdot n_p$ per front and a total capacity of $\sum_i N_{f,i}$. For instance, if $n_p = 100$ and $S_f = [1, 0.5]$, then up to 100 solutions are retained for the first front and 50 for the second, for a maximum population of 150.

During the selection process, the top n_f fronts are first filled up to their maximum size. If fewer than n_p individuals can be selected this way, the population is filled to its minimum with the best-performing among the remaining candidate solutions. If fewer than n_p feasible solutions are available, infeasible solutions are selected in increasing order of constraint violation.

This approach mitigates the loss of diversity caused by premature local convergence. A common cause of diversity loss in EAs is that solutions converging faster to one subregion of the Pareto front dominate individuals converging to a different region, which are then discarded when the first front fills the entire population. This strategy allows for keeping more solutions as the algorithm gets closer to convergence while ensuring that those solutions are distributed across the first n_f fronts.

The population P is initialized by providing or randomly generating n_p candidate solutions. The random generation is performed through the Latin Hypercube Sampling (LHS) [61] method. LHS partitions the range of each variable of the problem into equally spaced intervals and ensures a stratified, non-redundant sampling across all dimensions. For each sample, an interval is uniformly selected without replacement for each coordinate, and a point is then selected uniformly at random in the hyper-rectangle bounded by those intervals. The optimization loop proceeds as follows:

1. Evaluation. After adding them to the current population, the objective and constraint values of the new candidate solutions are evaluated.
2. Sorting. The population is sorted into Pareto fronts based on non-domination and crowding distance assignment (see Section 2.1).
3. Selection. Iterating over the first n_f fronts, up to N_i solutions are selected for the ith front in order of decreasing crowding distance. If fewer than n_p solutions can be selected this way, the population is completed by adding the best-performing among the remaining solutions, first selecting feasible solutions in order of rank and crowding distance, then, if necessary, the infeasible solutions with the lowest constraint violation.
4. Parent selection. One round of tournament selection identifies the parent population. If fewer than n_p individuals fall within the first n_f fronts, the tournament operates over the entire population. Otherwise, it is performed within each of those fronts independently, preserving the proportion of their population.
5. Estimation of distribution. A GMM is fit to the parent population using the EM algorithm, with initial centroids provided by k-means clustering.
6. Sampling. New candidate solutions are sampled from the estimated GMM.

The process is stopped upon exceeding the iteration limit or a maximum amount of computational resources. As the sorting algorithm populates the lower-rank (better) fronts first, it is possible to interrupt it after populating the first n_f fronts and reaching the minimum population size. In this work, the sorting and selection steps are decoupled. Given that the computational cost of non-dominated sorting is negligible compared to that of distribution estimation, this separation allows for a more modular implementation, improving both maintainability and scalability.

In the tournament selection process, pairs of candidate solutions are drawn randomly without replacement and the higher-ranked one receives a point. One or multiple rounds can be carried out and the parent population is selected from the higher score solutions, with rank as a tiebreaker. This scoring-based mechanism enhances elitism during early search phases while reducing over-exploitation compared to a rank-based cutoff: higher-ranked solutions are more likely to be selected, but lower-ranked individuals retain a non-zero probability that decreases gradually with rank, preserving diversity in the parent population. Developing more effective strategies to balance exploitation and diversity preservation remains an open research direction.

The GMM is governed by two critical parameters: the number K of Gaussian components and the lower bound on the covariance matrix elements $\Sigma_{\min}$. The latter is essential for ensuring numerical stability during the EM updates. Without a minimum variance threshold, the EM algorithm may converge to a degenerate solution in which a Gaussian component collapses onto a single data point, resulting in a near-zero variance and a divergent likelihood. Beyond ensuring stability, the choice of $\Sigma_{\min}$ directly affects the exploration-exploitation trade-off. A higher minimum variance promotes exploration by increasing the likelihood of sampling offspring farther from the current population clusters, thus enhancing the capacity of the algorithm to search unexplored regions of the design space.

Algorithm 4 MOEDA-GMM.

Initialization:
$P_{new} \leftarrow \{n_p$ random candidates via LHS$\}$
P $\leftarrow \emptyset$, F $\leftarrow \emptyset$, $C \leftarrow \emptyset$
R $\leftarrow \emptyset$
CD $\leftarrow \emptyset$
$t \leftarrow 0$
$N_f[i] \leftarrow \lceil n_p \cdot S_f[i] \rceil \quad \forall i$
while $t < T_{\max}$ **and** resourceLimitNotExceeded **do**
  $F_{\text{new}} \leftarrow [F_1(P_{\text{new}}), F_2(P_{\text{new}})]$
  $C_{\text{new}} \leftarrow \sum_{i=1}^{n} \max(0,\ g_i(P_{\text{new}}))$
  $P \leftarrow P \parallel P_{\text{new}}$
  $F \leftarrow F \parallel F_{\text{new}}$
  $C \leftarrow C \parallel C_{\text{new}}$
  $R, CD \leftarrow$ NondominatedSortingWithCD(P, F, C)
  $\text{id}_{\text{sel}} =$ SelectionStep(R, C, CD)
  $P \leftarrow P[\text{id}_{\text{sel}}]$
  $F \leftarrow F[\text{id}_{\text{sel}}]$
  $C \leftarrow C[\text{id}_{\text{sel}}]$
  $R \leftarrow R[\text{id}_{\text{sel}}]$
  $CD \leftarrow CD[\text{id}_{\text{sel}}]$
  $P_{\text{parents}} \leftarrow$ TournamentSelection(P, R, CD)
  initial_clusters $\leftarrow$ kMeans(P_{parents})
  GMM $\leftarrow$ FitGMM$(P_{\text{parents}}$, initial_clusters$)$
  $P_{\text{new}} \leftarrow$ Sample(GMM)
  $t \leftarrow t + 1$
end while
return Final population or best Pareto front

Algorithm 5 SelectionStep(R, C, CD).

$r_{\text{infeas}} = \max(R)$ ▷ rank equal to $n_{fronts} + 1$ for the infeasible solutions
$\text{id}_{\text{sel}} \leftarrow \emptyset$
for $f \in [1, n_f]$ **do**
  $\text{id}_{\text{curr}} \leftarrow \{i \in [1, |P|] \text{ s.t. } R[i] = f\}$
  $\text{id}_{\text{sel}} \parallel \text{id}_{\text{curr}}$
end for
if $|\text{id}_{\text{sel}}| < n_p$ **then**
  $n_{\text{remaining}} \leftarrow n_p - |\text{id}_{\text{sel}}|$
  $\text{r}_{\text{curr}} \leftarrow n_f$
  $\text{id}_{\text{curr}} \leftarrow \emptyset$
  while $n_{\text{remaining}} - |\text{id}_{\text{curr}}| > 0$ **do**
    $\text{id}_{\text{sel}} \parallel \text{id}_{\text{curr}}$
    $n_{\text{remaining}} \leftarrow n_{\text{remaining}} - |\text{id}_{\text{curr}}|$
    $\text{r}_{\text{curr}} \leftarrow \text{r}_{\text{curr}} + 1$
    $\text{id}_{\text{curr}} \leftarrow \{i \in [1, |P|] \text{ s.t. } R[i] = \text{r}_{\text{curr}}\}$
  end while
  if $\text{r}_{\text{curr}} = r_{\text{infeas}}$ **then**
    front_order $\leftarrow$ Argsort($C[\text{id}_{\text{curr}}]$, ascending)
  else
    front_order $\leftarrow$ Argsort($CD[\text{id}_{\text{curr}}]$, descending)
  end if
  $\text{id}_{\text{sel}} \parallel \text{id}_{\text{curr}}[\text{front_order}[1 : n_{\text{remaining}}]]$
end if
return id_{sel}

To dynamically balance exploration and exploitation throughout the search, MOEDA-GMM employs a scheduling strategy for the minimum covariance $\Sigma_{\min}$, which decays over time according to a power-law function. This schedule enables the model to explore broader regions of the search space in early iterations and gradually concentrate the search around high-performing regions as the optimization proceeds. The covariance floor $\Sigma_{\min}$ decays from its initial value α to a minimum $\alpha_{\min}$ over $t_{\max}$ iterations. The decay schedule is defined by:

$$\frac{\Sigma_{\min}}{\alpha} = 1 - \left(\frac{t}{t_{\max}}\right)^{\beta} \cdot \left(1 - \frac{\alpha_{\min}}{\alpha}\right), \tag{13}$$

where t represents the current iteration index and β controls the rate of decay. Larger values of β result in a steeper transition, allowing for more prolonged exploration before rapidly shifting to exploitation. Both α and β are user-defined parameters that can be tuned to adapt the convergence behavior of the algorithm to the specific characteristics of the optimization problem.

The GMM is fitted to the parent population using the EM algorithm [50], a two-step iterative procedure designed to compute maximum likelihood estimates for models with latent variables.

A latent variable model incorporates unobserved variables whose probability distribution depends on the model parameters. While the marginal likelihood can, in principle, be expressed and maximized directly, doing so is computationally intractable in most practical cases. EM circumvents this by iteratively estimating the expected value of the latent variables and optimizing the likelihood conditional on those estimates. This method applies to mixture models like the GMM.

A GMM is a probability distribution whose Probability Density Function (PDF) is defined as a weighted sum of K independent multivariate Gaussian distributions. Each component is parameterized by a mean vector μ_i, a covariance matrix Σ_i, and a mixing weight π_i. Sampling from a GMM consists of drawing a component index $z \sim \text{Multinomial}(\pi_1, \ldots, \pi_K)$, followed by sampling from the corresponding Gaussian distribution:

$$\begin{cases} P(z = i) = \pi_i, \quad \text{where } \sum_{i=1}^{K} \pi_i = 1 \\ P(x \mid z = i) = \mathcal{N}(\mu_i, \Sigma_i). \end{cases} \tag{14}$$

Marginalizing over z, the full distribution is expressed as:

$$p(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x \mid \mu_i, \Sigma_i) \tag{15}$$

Let $\{x_n\}_{n=1}^{N}$ be the dataset of N samples. A GMM with K components is parameterized by:

$$\theta^{(t)} = \left\{\mu_i^{(t)}, \Sigma_i^{(t)}, \pi_i^{(t)}\right\}_{i=1}^{K}, \tag{16}$$

where, for each component i, $\mu_i^{(t)}$ denotes its mean vector at iteration t, $\Sigma_i^{(t)}$ the corresponding covariance matrix, and $\pi_i^{(t)}$ the mixing weight, subject to the constraint $\sum_{i=1}^{K} \pi_i^{(t)} = 1$.

The latent variable $z \in \{1, \ldots, K\}$ indicates which component generated a sample.

The marginal density of a sample x_n under the GMM is:

$$p\left(x_n \mid \theta^{(t)}\right) = \sum_{j=1}^{K} \pi_j^{(t)} \mathcal{N}\left(x_n \mid \mu_j^{(t)}, \Sigma_j^{(t)}\right), \tag{17}$$

where $\mathcal{N}(x \mid \mu, \Sigma)$ denotes the multivariate normal PDF with mean μ and covariance Σ.

In the Expectation step, the responsibilities represent the posterior probability that the sample x_n was generated by component i, given the current parameter estimates. They are defined as:

$$P_i^{(t)}(x_n) = P\left(z = i \mid x_n, \theta^{(t)}\right), \tag{18}$$

Applying Bayes' rule, this expression becomes:

$$P_i^{(t)}(x_n) = \frac{\pi_i^{(t)} \mathcal{N}\left(x_n \mid \mu_i^{(t)}, \Sigma_i^{(t)}\right)}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{N}\left(x_n \mid \mu_j^{(t)}, \Sigma_j^{(t)}\right)}. \tag{19}$$

The effective sample count for component i is:

$$N_i^{(t)} = \sum_{n=1}^{N} P_i^{(t)}(x_n). \tag{20}$$

In the M-step, parameters are updated by maximizing the expected complete-data log-likelihood using the computed responsibilities:

$$\mu_i^{(t+1)} = \frac{1}{N_i^{(t)}} \sum_{n=1}^{N} P_i^{(t)}(x_n)\, x_n, \tag{21}$$

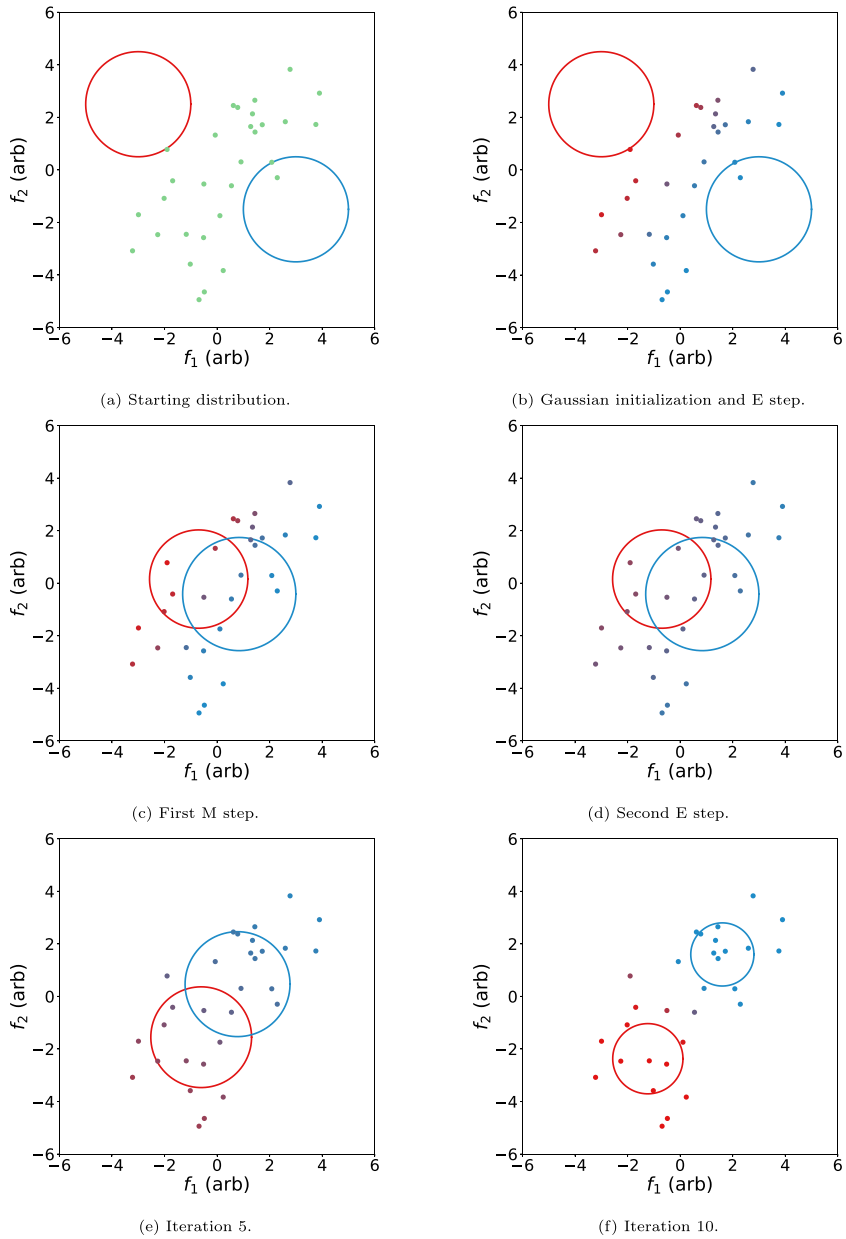
$$\Sigma_i^{(t+1)} = \frac{1}{N_i^{(t)}} \sum_{n=1}^{N} P_i^{(t)}(x_n)\, (x_n - \mu_i^{(t+1)})(x_n - \mu_i^{(t+1)})^{\top}, \tag{22}$$

$$\pi_i^{(t+1)} = \frac{N_i^{(t)}}{N}. \tag{23}$$

The updated weights are the normalized average responsibilities for each component.

The algorithm alternates between E and M steps until the relative change in log-likelihood falls below a predefined tolerance (10^{-3} by default) or 100 iterations are completed. All input samples are normalized to the unit hypercube based on their box constraints before fitting. This prevents numerical instability caused by different variable scales and improves convergence behavior.

An example of the EM iterations is illustrated in Fig. 1, showing how the Gaussian components adapt over the initial E and M steps and the convergence after 5 and 10 steps. Starting from a 2D data distribution and arbitrary initial Gaussian components, spherical in this example case, the responsibilities are computed in the first E step, with the red-blue mixing proportion indicating the fraction of responsibility. The following M step shows the recomputed Gaussians and the E step shows the updated responsibilities.



(a) Starting distribution.

(b) Gaussian initialization and E step.

(c) First M step.

(d) Second E step.

(e) Iteration 5.

(f) Iteration 10.

Fig. 1. Example of an EM process with spherical Gaussians. An intentionally suboptimal Initialization better highlights the changes occurring in the E- and M-steps.

3. Results and discussion

This section evaluates the performance of the proposed MOEDA-GMM algorithm by comparing it against three well-established multi-objective evolutionary algorithms: NSGA-II, SMS-EMOA, and RVEA. The comparison is carried out using three multi-objective analytical test functions (ZDT1, ZDT2, and ZDT3), three synthetic structural engineering benchmark problems (Welded Beam, Four-Bar Truss, and Two-Bar Truss), as well as an example real-world structural design involving a plane Warren Truss structure.

All algorithms were tested under identical evaluation budgets and population sizes, with results averaged over 50 independent runs. Performance was compared using the hypervolume (HV), generational distance plus (GD^+), spread (Δ), and uniform distribution error (UDE), which together quantify convergence and population distribution. Since the benchmark problems used here have very low evaluation costs, the computational overhead from model fitting and sampling is artificially emphasized. In realistic high-fidelity structural optimization tasks, where finite-element analyses dominate the runtime, this overhead is small by comparison. Efficiency is therefore reported in terms of the number of objective-function evaluations required to reach comparable solution quality rather than elapsed time. For each problem, the influence of α, β, and K was analyzed using the Kruskal–Wallis (KW) H test [62] on final indicator values (HV, GD^+, Δ, UDE), followed by Bonferroni-adjusted pairwise Mann–Whitney U tests [63]. Runs are treated as independent samples, as seed initialization does not create a blocking structure. To study trends and interactions across problems, these per-problem tests are complemented with factorial models. Where residuals are approximately normal, a Linear Mixed-Effects Model (LMM) [64] is used, treating problem variability as a random intercept. When normality assumptions are violated, the same factorial design is analyzed using an Aligned Rank Transform (ART) ANOVA [65], providing a distribution-free alternative that preserves interpretability of main and interaction effects. As a descriptive summary, per-problem configuration performance is visualized via decile-ranked heatmaps over the (β, α) grid, with separate panels for each K value and evaluation

timestep. Champion configurations are identified by weighted rank aggregation across indicators, validated through multi-method concordance (LMM, ART ANOVA, Friedman) and leave-one-problem-out jackknife stability (Section 3.5).

The same statistical framework is then applied to the baseline comparison: with the selected champion configuration, MOEDA is compared to NSGA-II, SMS-EMOA, and RVEA using per-problem Kruskal–Wallis and pairwise Mann–Whitney U tests (50 runs per algorithm). Cross-problem results use ART ANOVA with *algorithm* as fixed effect and *problem* as random intercept, complemented by Friedman ranks and Critical-Difference (CD) diagrams.

NSGA-II, SMS-EMOA, and RVEA are implemented using the pymoo [66] Python library, which also provides interfaces for the synthetic benchmark problems. RVEA uses a bi-objective reference-vector set with the number of vectors equal to the population size, APD selection, and the library defaults for vector adaptation frequency and penalty exponent; all other settings (population size, evaluation budget) match the baselines for comparability. The Warren Truss analysis is performed through a bespoke Python module integrated into the benchmark pipeline by wrapping it in the pymoo problem API.

For each structural problem, the constraints related to stress, buckling, and deflection are selected to reflect typical allowable limits found in structural design practice. Although strict compliance with specific design codes is not enforced, the adopted thresholds are consistent with conservative safety margins commonly used in engineering applications. These bounds are intended to ensure that all generated solutions maintain structural integrity under standard design conditions.

3.1. Experimental setup

MOEDA-GMM introduces three algorithm-specific parameters: the initial covariance floor α, the decay exponent β, and the number of Gaussian components K. To systematically assess the sensitivity of performance to these parameters and to identify configurations suitable for publication comparison, a full factorial grid was constructed as follows:

- $\beta \in \{0.5, 0.6, 0.7, 1.0, 1.5, 2.0\}$ (6 levels),
- $\alpha \in \{0.003, 0.005, 0.007, 0.008, 0.009, 0.010, 0.011\}$ (7 levels),
- $K \in \{2, 3, 4, 5, 7\}$ (5 levels),

yielding $6 \times 7 \times 5 = 210$ distinct configurations in a balanced factorial design. The balanced structure is essential for clean three-way factorial analyses of main effects and interactions ($\beta \times \alpha$, $\beta \times K$, $\alpha \times K$). The minimum covariance floor is fixed at $\alpha_{\min} = 10^{-5}$ across all configurations. Every configuration is evaluated on each of the seven benchmark problems over 50 independent runs to ensure statistical reliability. Each run uses a deterministic seed derived from a fixed base seed, so that all algorithms and configurations share identical initial populations for each run index, enabling full reproducibility. All algorithms, namely MOEDA-GMM, NSGA-II, SMS-EMOA, and RVEA, share the following common settings: population size $N = 50$ and maximum iterations $G = 200$. MOEDA-GMM additionally uses a parent fraction of 0.5 and a stratified buffer with fractions (1.0, 0.5, 0.5) for the first three non-dominated fronts, allowing the effective population to expand up to $2N = 100$ when beneficial. For performance evaluation, four complementary quality indicators are computed at each iteration: HV [67], GD$^+$ [68], Δ [31], and UDE [57], defined as the coefficient of variation of consecutive inter-point distances along the sorted front. For each problem, a global reference front is constructed by merging the non-dominated solutions from all algorithms, configurations, and timesteps. The ideal and nadir points are taken as the componentwise minimum and maximum of this reference front, ensuring tight normalization bounds for HV and GD$^+$ that remain consistent across all evaluation points.

3.2. Statistical framework

With 210 configurations, seven problems, and 50 runs each, the indicator distributions exhibit pronounced non-normality (Table 1).

Table 1
Distributional diagnostics for the four quality indicators (pooled across all configurations, problems, and runs). ICC (intra-class correlation) quantifies the proportion of variance attributable to between-problem differences in the LMM.

Indicator	Skewness	Excess Kurtosis	ICC (%)	Shapiro–Wilk p
HV	−6.4	49.9	84.8	$< 10^{-6}$
Δ	1.2	8.8	84.5	$< 10^{-6}$
UDE	2.9	19.2	66.0	$< 10^{-6}$
GD$^+$	11.0	144.1	≈ 0	$< 10^{-6}$

Normality is assessed via D'Agostino–Pearson [69], which tests whether skewness and excess kurtosis are jointly zero, and Shapiro–Wilk [70], which tests correlation with expected normal order statistics. Both reject normality at $p < 10^{-6}$ for all four metrics. The severe departures preclude the use of parametric ANOVA for comparing configurations; a non-parametric and rank-based framework is adopted throughout. Where parametric models (linear mixed models) are used as a supplementary perspective, the normality violation is reported explicitly.

Within each problem, the null hypothesis of distributional equality ($F_1 = F_2 = \cdots = F_k$) is tested via the Kruskal–Wallis H test, which detects any shift in location or shape among groups without assuming equal distributional form. Rejection indicates that at least one configuration is stochastically dominant. Pairwise Mann–Whitney U tests with Bonferroni correction then identify which specific pairs differ. This within-problem procedure is applied both during sensitivity analysis (comparing all 210 configurations) and during baseline comparison (comparing champion configurations against NSGA-II, SMS-EMOA, and RVEA). Across problems, the Friedman test assesses rank consistency by treating problems as blocks and configurations as treatments. Its statistic is based on mean ranks, but the hypothesis concerns the exchangeability of treatments, not average performance. Nemenyi [71] post-hoc tests with critical difference diagrams identify groups of statistically indistinguishable algorithms. To isolate the effect of individual design factors, three-way Kruskal–Wallis tests assess the main effects of β, α, and K as well as their pairwise interactions ($\beta \times \alpha$, $\beta \times K$, $\alpha \times K$) on each indicator. ART ANOVA [65] complements these tests by first stripping the effects of other factors before ranking, providing cleaner interaction estimates. Results are visualized as interaction heatmaps. Champion selection employs three independent ranking methods, each testing whether the choice of configuration has a significant effect on indicator values: Friedman (non-parametric, cross-problem rank consistency), ART ANOVA (non-parametric factorial via aligned ranks), and LMM (parametric mixed model testing whether $\tau_i = 0$ in $y_{ij} = \mu + \tau_i + u_j + \varepsilon_{ij}$ with random intercept $u_j \sim \mathcal{N}(0, \sigma_u^2)$). Friedman and ART require only ordinal data; the LMM additionally assumes normally distributed residuals and random effects, but provides the intra-class correlation ICC $= \sigma_u^2/(\sigma_u^2 + \sigma_\varepsilon^2)$ for variance decomposition and remains asymptotically valid for large samples. Each method produces a per-indicator ranking of all 210 configurations. Weighted rank aggregation across indicators (HV weight 0.4; Δ, UDE, GD$^+$ weight 0.2 each) yields a composite score, from which champion pools are defined as the top 10% of configurations. Agreement across methods with different assumptions strengthens the conclusions; the composite set is further validated by leave-one-problem-out jackknife stability.

HV shows the strongest problem grouping (ICC = 84.8%), confirming that problem identity is a major source of variability and motivating the cross-problem blocked designs. At the other extreme, GD$^+$ has ICC ≈ 0 and singular random effects in the LMM (between-problem variance converges to zero), so its variation is driven entirely by within-problem, between-configuration differences. LMM results for GD$^+$ should therefore be interpreted with caution, and ART ANOVA is used as the primary ranking method for that indicator.

Full statistical tables for the baseline comparison are reported in Appendix B; marginal sensitivity distributions are in Appendix C.

3.3. Benchmark problems

3.3.1. Welded beam problem

The welded beam problem (Fig. 2) involves the simultaneous minimization of fabrication cost and vertical deflection at the free end of a cantilevered steel beam welded to a rigid base. The beam has a rectangular cross-section with width b and height t, and it is attached to the base via a fillet weld of length l and thickness h. The beam extends a fixed length L beyond the weld, and a vertical load P is applied at its free end.

The design vector is defined as:

$$x = [x_1, x_2, x_3, x_4] = [h, l, t, b]. \tag{24}$$

The problem is subject to several structural and geometric constraints:

- Maximum shear stress on the weld: $\tau(x) \leq 13{,}600$ psi,
- Maximum normal stress in the weld: $\sigma_{NM} \leq 30{,}000$ psi,
- Buckling load resistance: $P_{\text{cr}}(x) \geq P$,
- Weld thickness lower than the beam width $h \leq b$.

The bounds on the design variables are: $x_1 = h \in [0.125, 5]$, $x_2 = l \in [0.1, 10]$, $x_3 = t \in [0.1, 10]$ and $x_4 = b \in [0.125, 5]$.

The problem can be mathematically formulated as:

$$\begin{aligned}
&given \quad \tau_1(x) = \frac{1}{\sqrt{2}x_1x_2}, \quad \tau_2(x) = \frac{(L + \frac{x_2}{2})\sqrt{x_2^2 + (x_1 + x_3)^2}}{2\sqrt{2}\,x_1x_2\left(\frac{x_2^2}{12} + \frac{(x_1+x_3)^2}{4}\right)} \\
&minimize \quad f_1(x) = 1.10471\,x_1^2x_2 + 0.04811\,x_3x_4(L + x_2) \\
&\qquad\quad f_2(x) = \frac{P}{x_4x_3^3} \times 3.6587 \times 10^{-4} \\
&s.t. \quad P\sqrt{\tau_1(x)^2 + \tau_2(x)^2 + \frac{2\tau_1(x)\tau_2(x)x_2}{\sqrt{x_2^2 + (x_1 + x_3)^2}}} - 13{,}600 < 0 \\
&\qquad\quad \frac{6PL}{x_4x_3^2} - 30{,}000 < 0; \quad x_1 - x_4 < 0 \\
&\qquad\quad 6000 - 64{,}746(1 - 0.02823x_3)x_3x_4^3 < 0
\end{aligned} \tag{25}$$

where $P = 6000$ lbs and $L = 14$ in.

3.3.2. Four-bar truss optimization problem

The four-bar truss problem [72,73] involves a planar truss composed of four structural members, supporting a vertical load F applied at a free node corresponding to joint C, as represented in Fig. 3. The optimization objective is to simultaneously minimize the total structural volume and the vertical deflection at the loaded joint.

The design vector consists of the cross-sectional areas of the four members:

$$x = [x_1, x_2, x_3, x_4]. \tag{26}$$

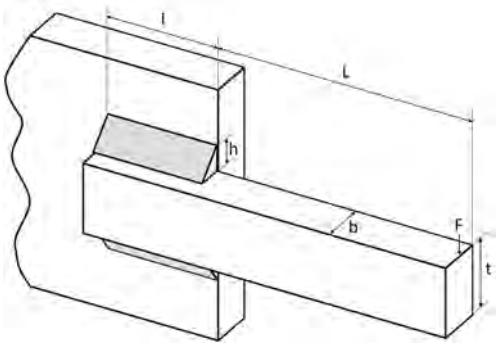


Fig. 2. Welded beam optimization benchmark.

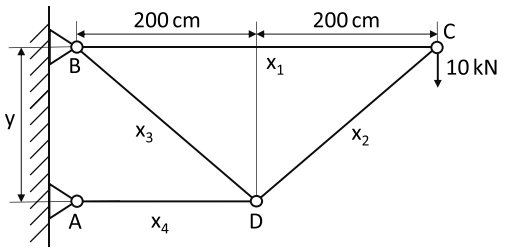


Fig. 3. Four-bar truss optimization benchmark.

The problem is subject to stress constraints on each structural member, which are implicitly handled by bounding the design variables within feasible ranges.

The design variable bounds are defined as: $x_1 \in [1, 3]$, $x_2 \in [\sqrt{2}, 3]$, $x_3 \in [\sqrt{2}, 3]$ and $x_4 \in [\sqrt{2}, 3]$.

The formal problem formulation is:

$$\begin{aligned}
\text{minimize} \quad & f_1(x) = 200\left(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4\right) \\
& f_2(x) = 10^{-3}F\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right)
\end{aligned} \tag{27}$$

where $F = 10$ kN is the applied load.

3.3.3. Two-bar truss optimization problem

The two-bar truss problem [72,73], illustrated in Fig. 4, features a simple triangular truss in which two angled members support a vertical load of 100 kN applied to the last node of the cantilever. The objective is to simultaneously minimize the total material volume of the structure and the maximum axial stress experienced by the members.

The design variables include the cross-sectional areas of the two bars, denoted x_1 and x_2, and the vertical distance y between the base supports and the loaded joint. The design vector is defined as:

$$x = [x_1, x_2, x_3] = [x_1, x_2, y]. \tag{28}$$

The system is constrained to a maximum member stress below 10^5 kPa.

The design variable bounds are: $x_1 \in [10^{-4}, 0.01]$, $x_2 \in [10^{-4}, 0.01]$ and $x_3 \in [1, 3]$.

The optimization problem is formally defined as:

$$\begin{aligned}
&minimize \quad f_1(x) = x_1\sqrt{16 + x_3^2} + x_2\sqrt{1 + x_3^2} \\
&\qquad\quad f_2(x) = \max\left(\frac{20\sqrt{16 + x_3^2}}{x_3x_1}, \frac{80\sqrt{1 + x_3^2}}{x_3x_2}\right) \\
&s.t. \quad \max\left(\frac{20\sqrt{16 + x_3^2}}{x_3x_1}, \frac{80\sqrt{1 + x_3^2}}{x_3x_2}\right) \leq 10^5
\end{aligned} \tag{29}$$

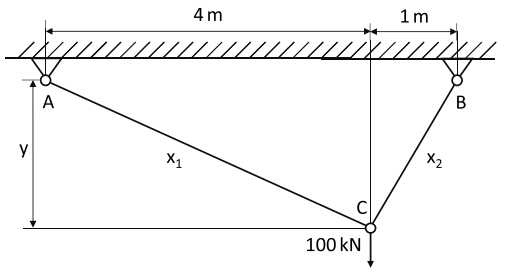


Fig. 4. Two-bar truss optimization benchmark.

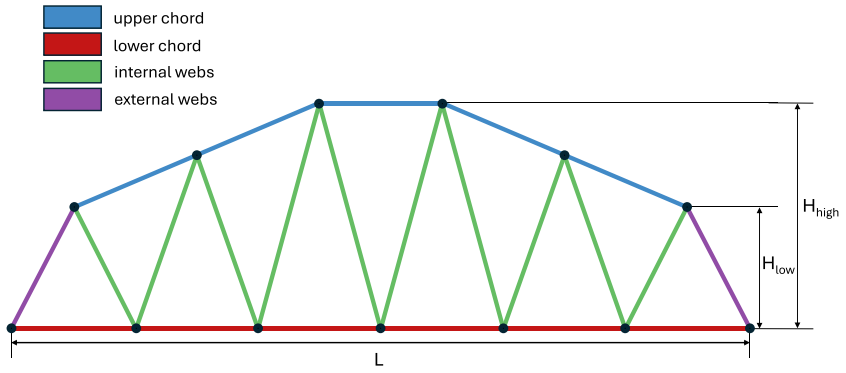


Fig. 5. Diagram of the Warren truss under study. Each color represents beams sharing identical cross-sectional dimensions B and web thickness s.

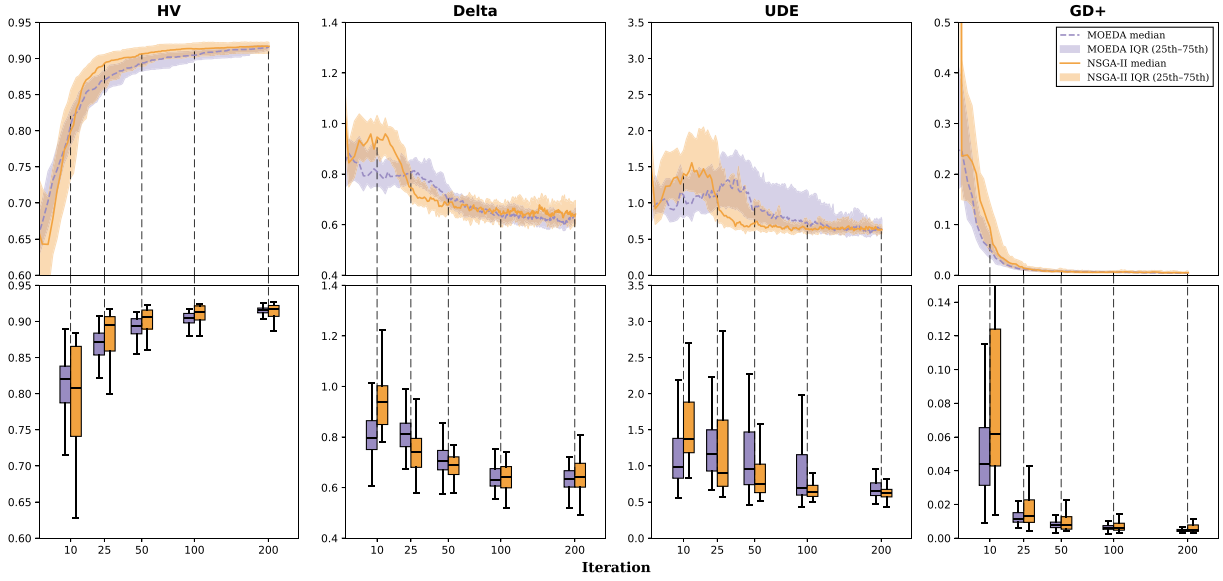


Fig. 6. Welded Beam: MOEDA-GMM champion vs NSGA-II.

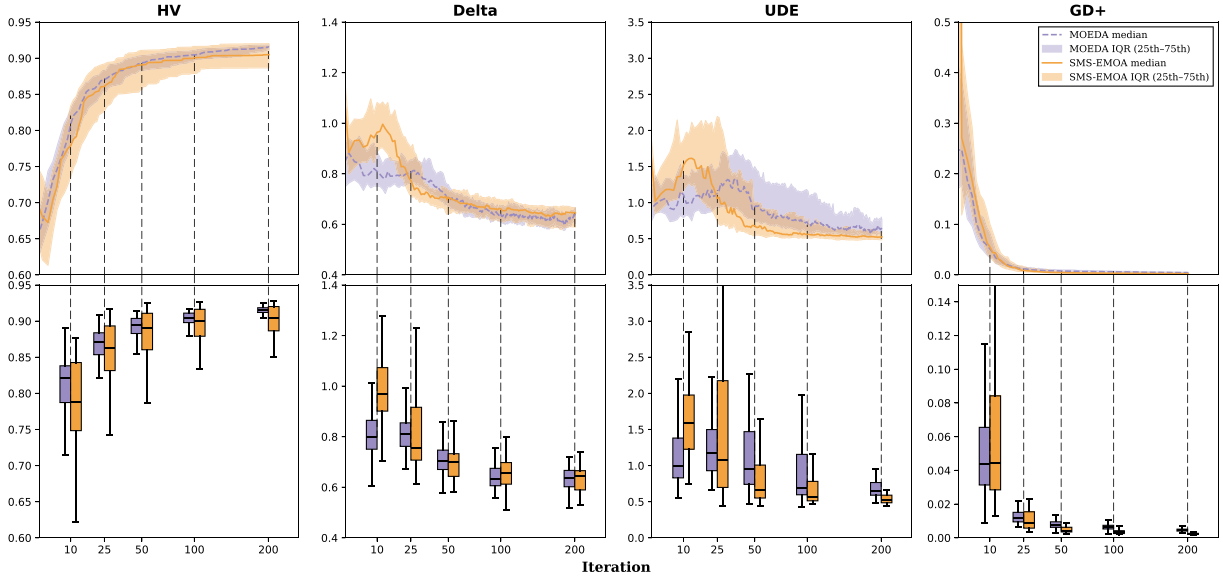


Fig. 7. Welded Beam: MOEDA-GMM champion vs SMS-EMOA.

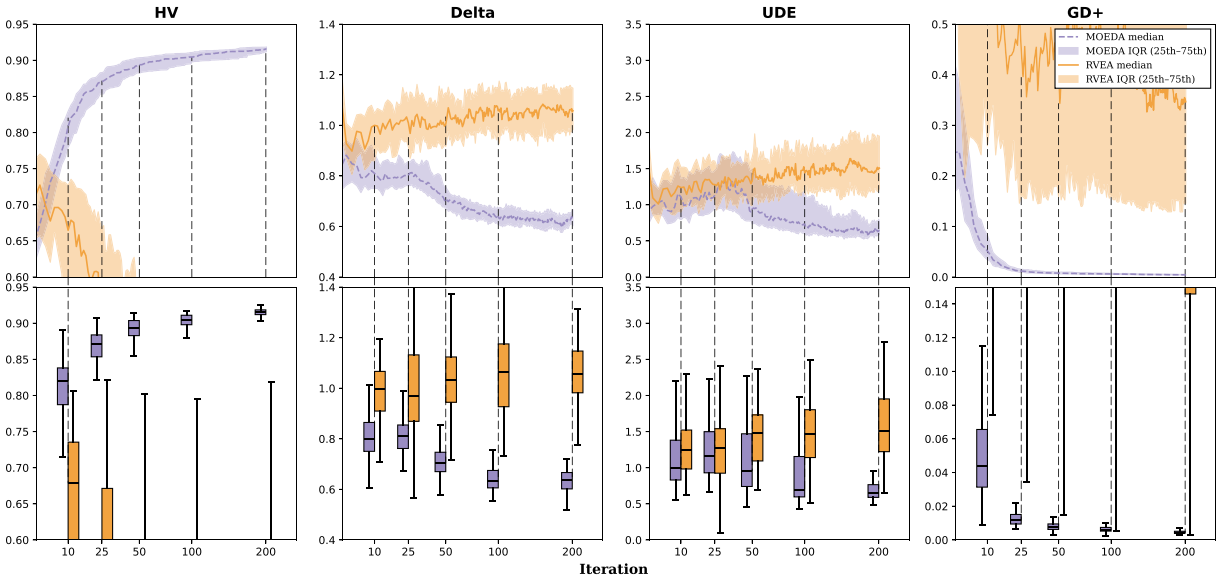


Fig. 8. Welded Beam: MOEDA-GMM champion vs RVEA.

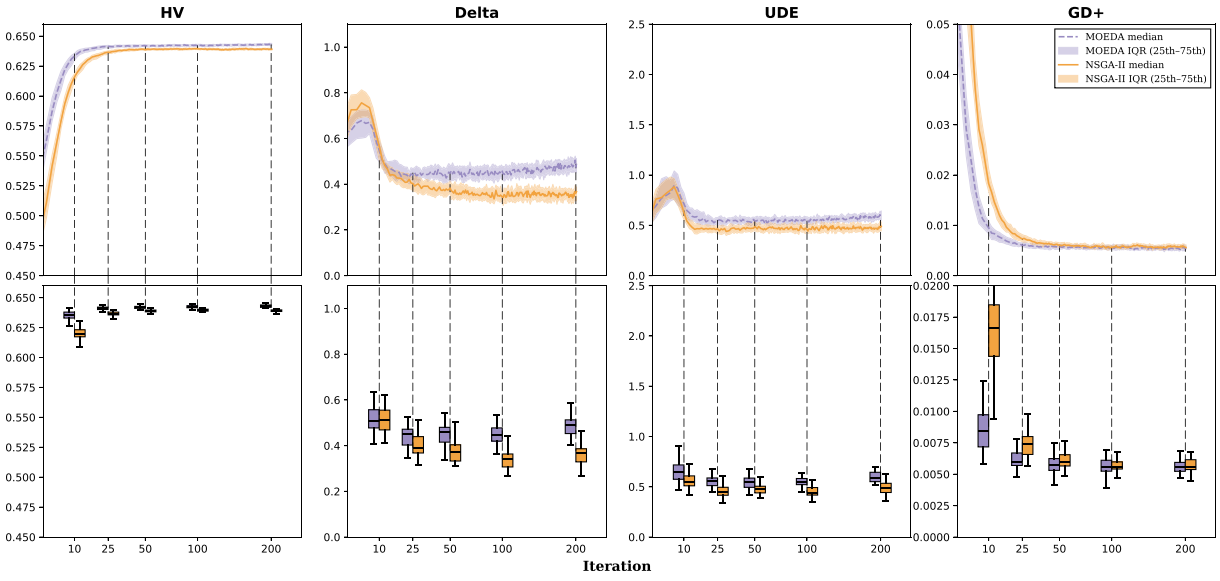


Fig. 9. Four-Bar Truss: MOEDA-GMM champion vs NSGA-II.

3.3.4. ZDT benchmark problems

To complement the structural benchmarks, the evaluation includes three problems from the ZDT benchmark family [74], a set of analytical bi-objective test functions widely used to assess convergence and diversity in multi-objective algorithms. These problems provide controlled Pareto front geometries (convex, concave, and discontinuous), allowing validation of algorithmic behavior under idealized but diverse conditions. All ZDT problems are continuous, unconstrained bi-objective optimization tasks, originally defined with $n = 30$ decision variables. In this work, each problem is implemented with $n = 10$ decision variables to ensure stable covariance estimation in the GMM and to avoid overfitting when training up to $K = 7$ Gaussian components under the sampling budgets available at population size $N = 50$. All decision variables satisfy $x_i \in [0, 1]$ for $i = 1, \ldots, n$. The general form of these problems is:

$$\begin{aligned} f_1(x) &= x_1, \\ f_2(x) &= g(x)\, h(f_1, g), \end{aligned} \tag{30}$$

where

$$g(x) = 1 + 9\,\frac{\sum_{i=2}^{n} x_i}{n-1}. \tag{31}$$

The function $h(f_1, g)$ defines the shape of the Pareto-optimal front and varies across ZDT1–3. For ZDT1 (convex front):

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}. \tag{32}$$

The Pareto-optimal front ($g = 1$) forms a continuous convex curve between $(f_1, f_2) = (0, 1)$ and $(1, 0)$, serving as a baseline for assessing convergence smoothness and uniform spread. For ZDT2 (concave front):

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2. \tag{33}$$

The inward-curved trade-off surface increases the difficulty of maintaining even population coverage. For ZDT3 (discontinuous front):

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1). \tag{34}$$

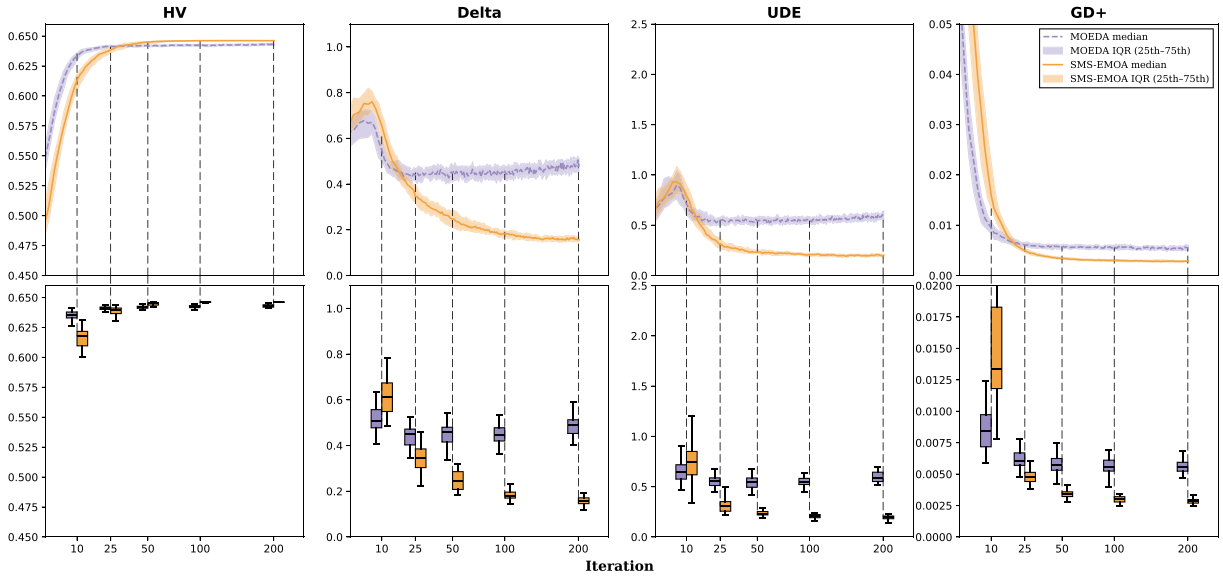


Fig. 10. Four-Bar Truss: MOEDA-GMM champion vs SMS-EMOA.

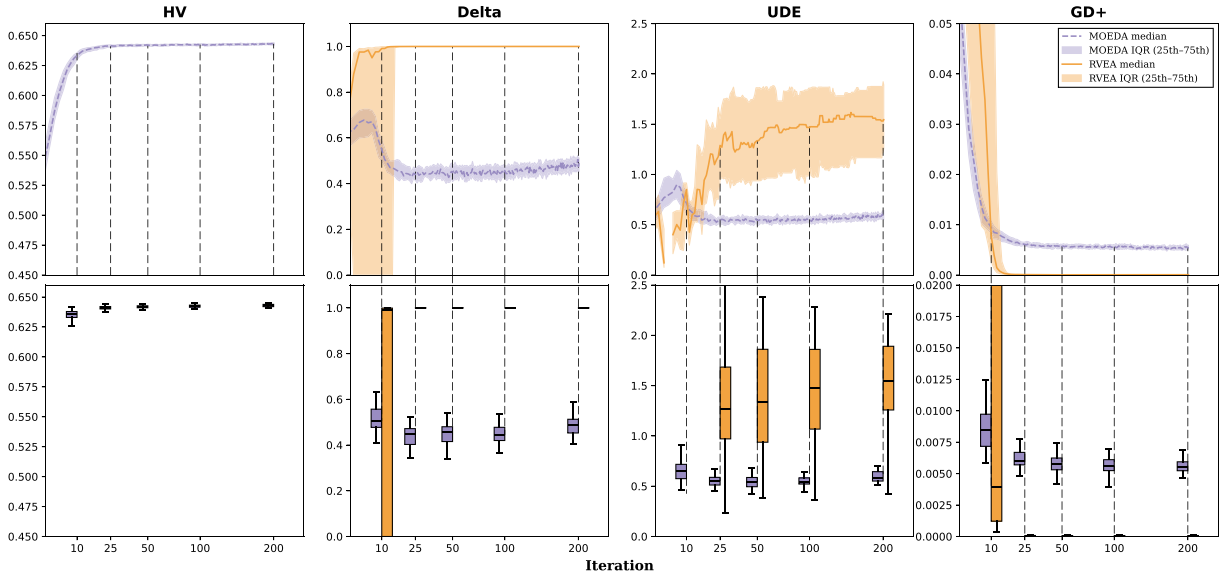


Fig. 11. Four-Bar Truss: MOEDA-GMM champion vs RVEA.

The Pareto-optimal front ($g = 1$) consists of five disjoint convex segments, requiring the algorithm to maintain diversity across isolated regions. Together, ZDT1–3 form a compact synthetic test suite covering three complementary Pareto front geometries, validating algorithmic behavior before extending the evaluation to constrained structural optimization problems.

3.3.5. *Warren truss system*

The final evaluation is conducted on a real-world structural design problem involving a planar Warren truss, illustrated in Fig. 5. The structure features a symmetric configuration, with the lower chord divided into six equal-length segments, forming a triangulated lattice between the intrados and extrados. The geometry is tapered, with the truss height increasing toward the center, enhancing structural efficiency under vertical loading. The truss is composed of welded hollow-core steel members with square cross-sections, which offer high resistance to buckling while minimizing the overall structural mass.

Design variables include the cross-section dimension B and wall thickness s for each of four beam types (upper chord, lower chord, internal and external webs), along with two height parameters: $H_{\min}$ (external webs) and $H_{\max}$ (central node). This yields a total of 10 continuous decision variables: $x = [B_1, B_2, B_3, B_4, s_1, s_2, s_3, s_4, H_{\min}, H_{\max}]$.

The constants are the span length of 6 m, the safety factors for maximum stresses $\gamma_s = 1.1$, and the material (reinforcement steel) properties. For the material properties we assume characteristic yield $f_{yk} = 355$ MPa, Young's modulus $E = 210$ GPa, and density $\rho = 7.7 \times 10^{-8}$ kN/mm^3.

The structure is modeled with pinned supports and is subjected to self-weight. The bi-objective problem aims to simultaneously minimize the total structural weight and the vertical displacement of the central lower chord node under self-load.

These objectives are inherently conflicting: minimizing structural mass tends to reduce stiffness, thereby increasing displacement. For example, the deflection δ of a simply supported beam under uniform

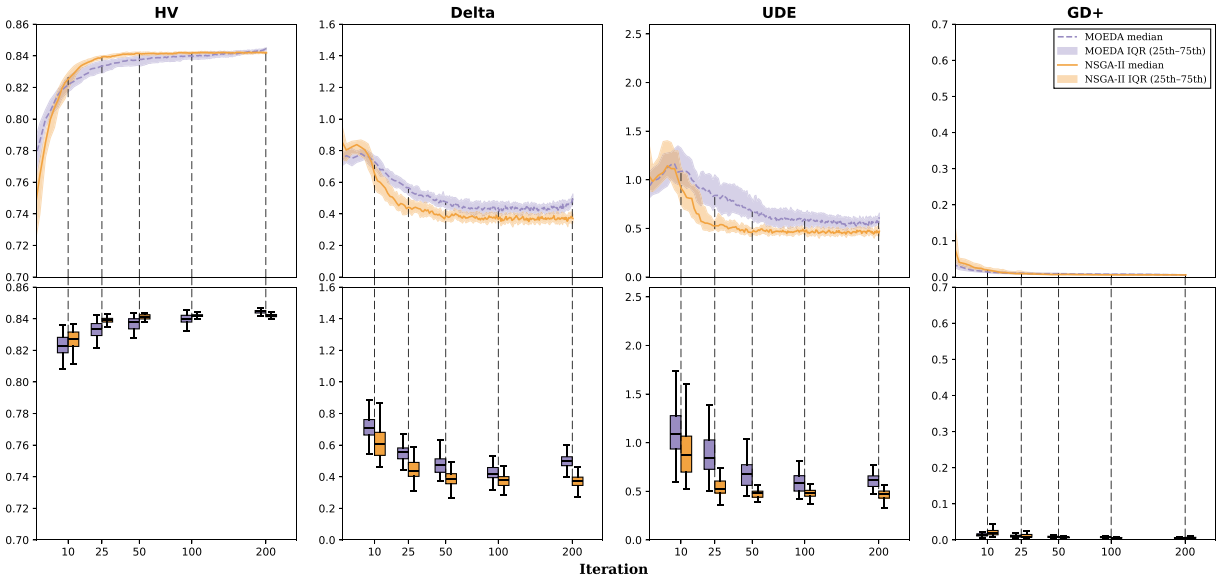


Fig. 12. Two-Bar Truss: MOEDA-GMM champion vs NSGA-II.

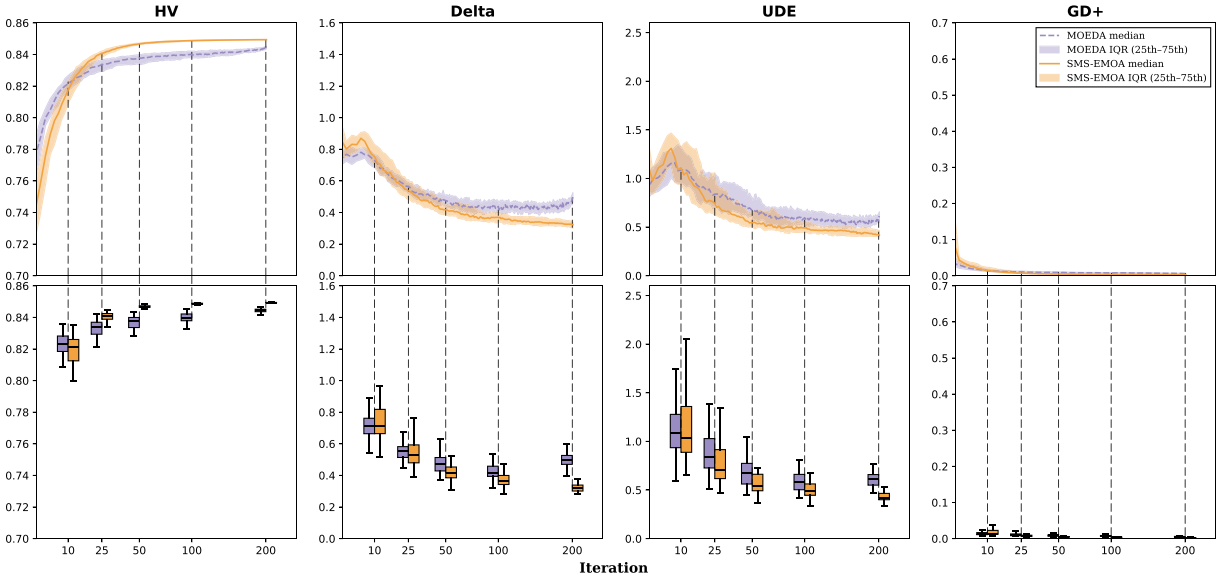


Fig. 13. Two-Bar Truss: MOEDA-GMM champion vs SMS-EMOA.

load scales as $\mathcal{O}(W L^3/bd^3)$, where W is the applied load, L is the span length, and b and d are the cross-sectional width and depth, respectively. Reducing the cross-sectional area lowers weight quadratically but increases deflection at the same rate, illustrating the trade-off between material efficiency and structural performance.

With a complex structure like a plane truss, the scaling laws change, but as shown by the experiments, the two objectives remain conflicting.

The structural solver is implemented as a bespoke Python module wrapped in the pymoo problem API. The solver uses a linear approximation to compute nodal displacements and beam stresses under gravitational loading.

The feasible design space is bounded by box constraints: $B_i \in [36, 300]$ mm, $s_i \in [5, 30]$ mm, $50\text{ mm} \leq H_{\min} \leq H_{\max} \leq 10^4$ mm

Inequality constraints are a maximum stress in all beams lower than the design yield strength $f_{yd} = f_{yk}/\gamma_s$ and a maximum permissible node displacement lower than $L/500$.

3.3.6. Benchmark optimization results

Each algorithm is executed for 200 iterations with a population size of 50, repeated across 50 independent runs. In the *pymoo* implementation, each iteration evaluates a number of offspring equal to the population size, resulting in a total of 10,050 function evaluations per run (200×50 offspring evaluations plus the initial population of 50).

NSGA-II, SMS-EMOA, and RVEA are executed using their default configurations as provided by the *pymoo* library. For MOEDA-GMM, all 210 configurations from the factorial grid (Section 3.1) are first evaluated across all benchmarks to produce a comprehensive sensitivity analysis of the three algorithm-specific parameters (Section 3.4). From these results, a small set of "champion" configurations is then selected via cross-problem weighted rank aggregation over all four quality indicators; the selection procedure and its validation are detailed in Section 3.5. For each benchmark, the best-performing champion from this pool is then compared head-to-head against the baseline algorithms.

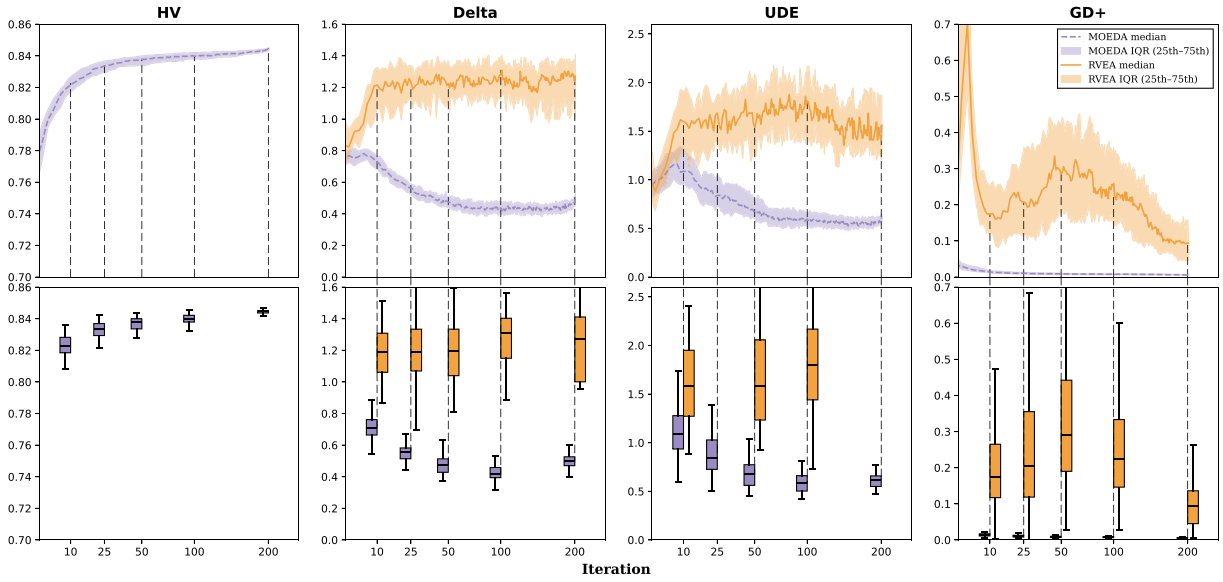


Fig. 14. Two-Bar Truss: MOEDA-GMM champion vs RVEA.

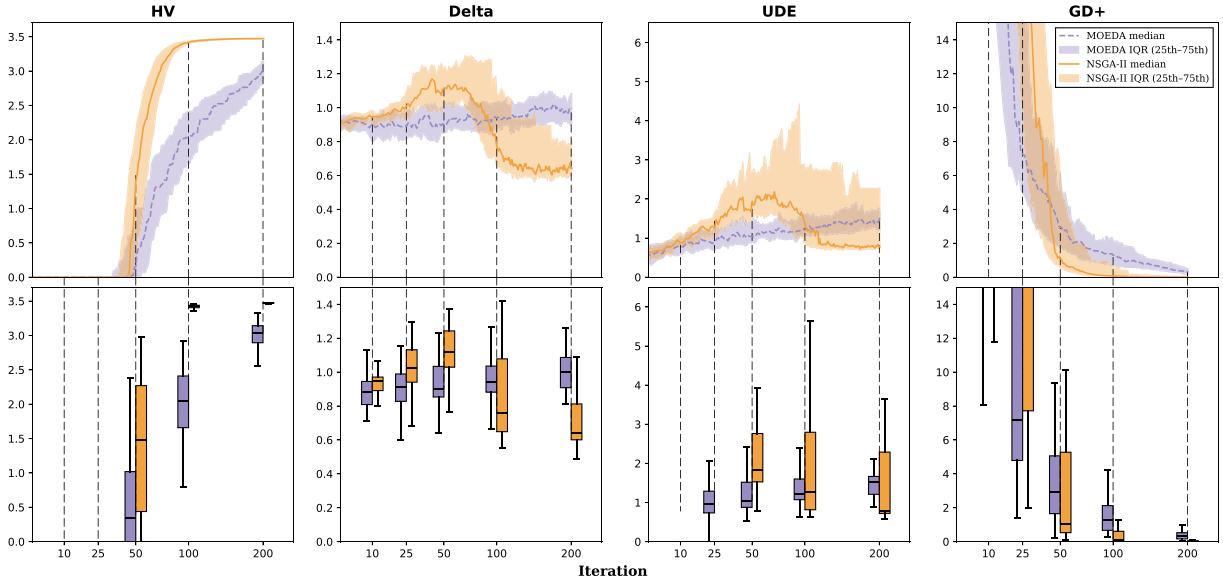


Fig. 15. Warren Truss: MOEDA-GMM champion vs NSGA-II.

Figs. 6–26 display the convergence behavior of the overall top-ranked MOEDA champion (β=1.0, α=0.009, K=2) against each baseline individually across the four quality indicators (HV, Δ, UDE, GD$^+$). Each panel shows median convergence with interquartile-range bands (top row) and boxplot distributions at selected iterations (bottom row) over 50 independent runs. On the structural benchmarks (Figs. 6–17), MOEDA-GMM demonstrates rapid early convergence on WeldedBeam and Truss2DFourbar, achieving competitive HV within the first 25 iterations. On Truss2D2bar the champion reaches near-final HV early but NSGA-II and SMS-EMOA close the gap by $t = 100$. RVEA exhibits front collapse on the three actively constrained structural problems (WeldedBeam, Truss2D2bar, WarrenTruss; HV $\to 0$). On the ZDT benchmarks (Figs. 18–26), SMS-EMOA achieves the best final HV, while MOEDA remains competitive with NSGA-II on ZDT1 and ZDT2. Early- and mid-stage pairwise tests (Appendix B.1) show that MOEDA achieves a significant advantage on Truss2DFourbar from $t = 10$, with further constrained problems followed by $t = 50$. The Friedman rank progression (Table B.20) indicates that rankings stabilize from $t = 100$ onwards. On the ZDT benchmarks and on WarrenTruss at later stages, SMS-EMOA and NSGA-II maintain or extend their lead, suggesting that MOEDA's probabilistic model benefits early exploration but does not consistently translate into long-term convergence advantages. The multi-component Gaussian mixture provides a mechanism for diversity maintenance, as distinct components can represent different regions of the objective space. However, without adaptive control the number of components remains fixed, which may limit coverage in later stages. Adaptive mechanisms such as entropy-based component revival or dynamic covariance inflation are identified as future extensions (Section 4). To quantify these observations statistically, cross-problem Friedman tests and per-problem Mann–Whitney U tests (Bonferroni-corrected) are applied at $t = 199$. The

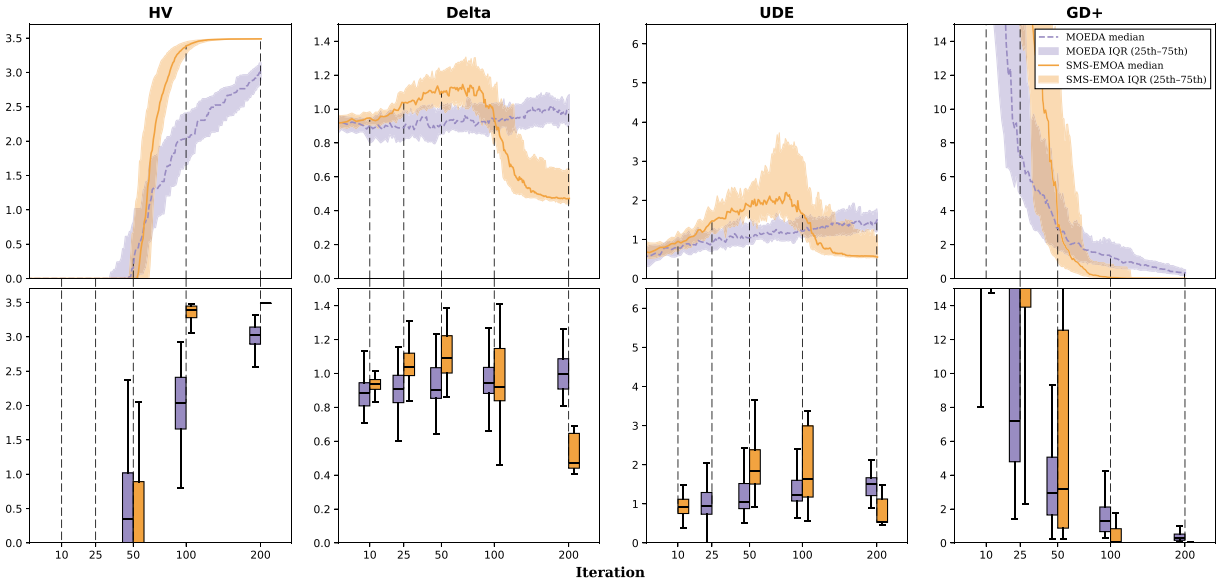


Fig. 16. Warren Truss: MOEDA-GMM champion vs SMS-EMOA.

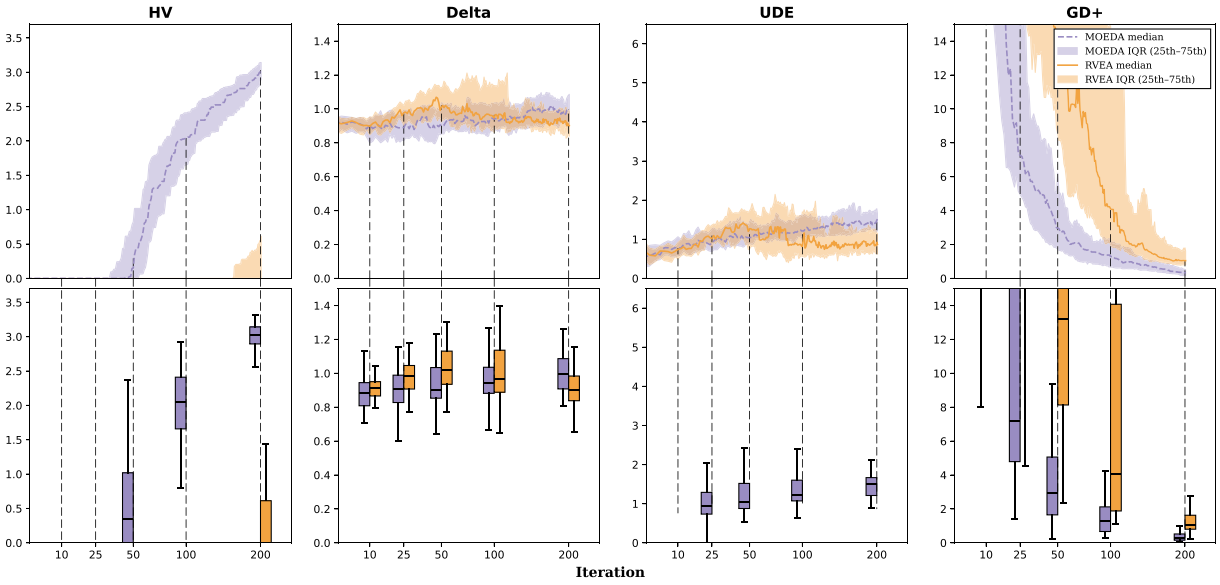


Fig. 17. Warren Truss: MOEDA-GMM champion vs RVEA.

final cross-problem comparison restricts to the top-5 MOEDA champions plus 3 baselines (8 algorithms). On HV, all three omnibus methods detect significant differences ($p < 10^{-4}$), with SMS-EMOA holding the top Friedman rank (1.86) and two MOEDA champions ranking ahead of NSGA-II. Per-problem pairwise tests show that MOEDA significantly outperforms RVEA on all seven benchmarks, partly due to RVEA's front collapse on the constrained problems. Against NSGA-II, MOEDA wins on Truss2D2bar and Truss2DFourbar (both $p < 0.001$) but is outperformed on the ZDT problems and WarrenTruss. Against SMS-EMOA, MOEDA wins on WeldedBeam ($p = 0.042$) while SMS-EMOA prevails on ZDT1, ZDT3, Truss2D2bar, and WarrenTruss. Full ranking tables and pairwise test results are reported in Appendix B.

3.4. Parameter sensitivity analysis

To understand how each hyperparameter influences algorithm performance, a three-way factorial Kruskal–Wallis analysis is conducted on the full 210-configuration grid. For each of the four indicators and at each of the nine evaluation timesteps, the test assesses significance of the main effects (β, α, K) and their pairwise interactions ($\beta \times \alpha$, $\beta \times K$, $\alpha \times K$).

3.4.1. Main effects

Table 2 reports, for each parameter and indicator, the number of benchmark problems on which the factorial KW test rejects H_0 at $p < 0.05$ at the final evaluation point ($t = 199$).

The decay exponent β is the most universally influential parameter, reaching significance on all seven problems for every indicator at $t = 199$. The number of Gaussian components K is nearly as pervasive (significant on 6–7 out of 7 problems, depending on the metric), while the initial covariance floor α shows the most variability, particularly for Δ (5/7). The WarrenTruss problem stands out as the most parameter-insensitive benchmark: α fails to reach significance on three of the four indicators (HV $p = 0.20$; Δ $p = 0.87$; UDE $p = 0.14$), and K is marginally

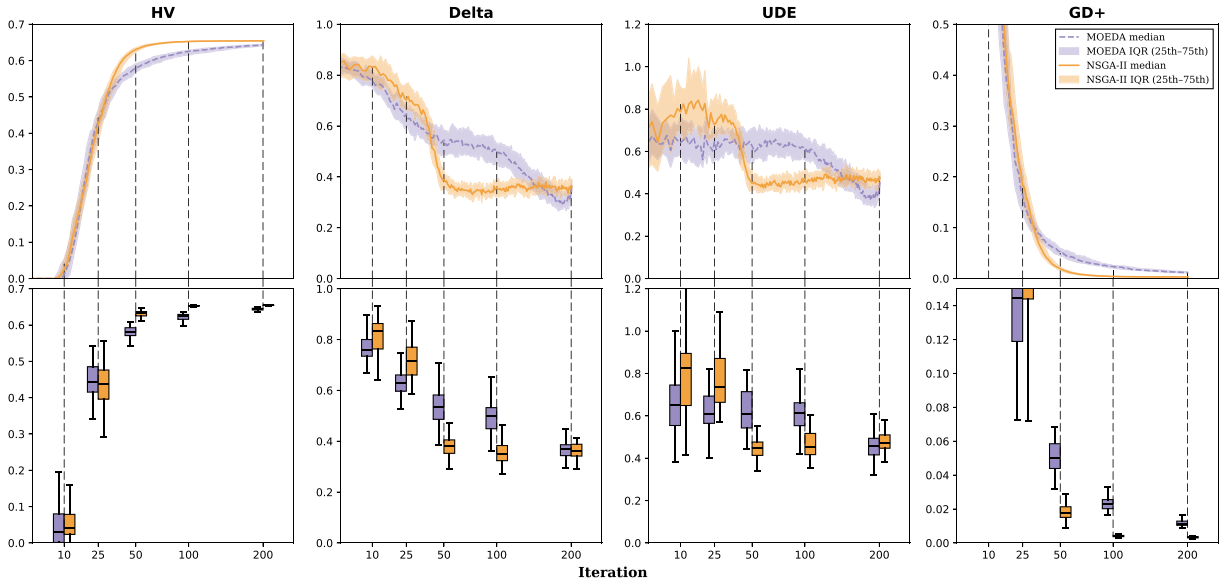


Fig. 18. ZDT1: MOEDA-GMM champion vs NSGA-II.

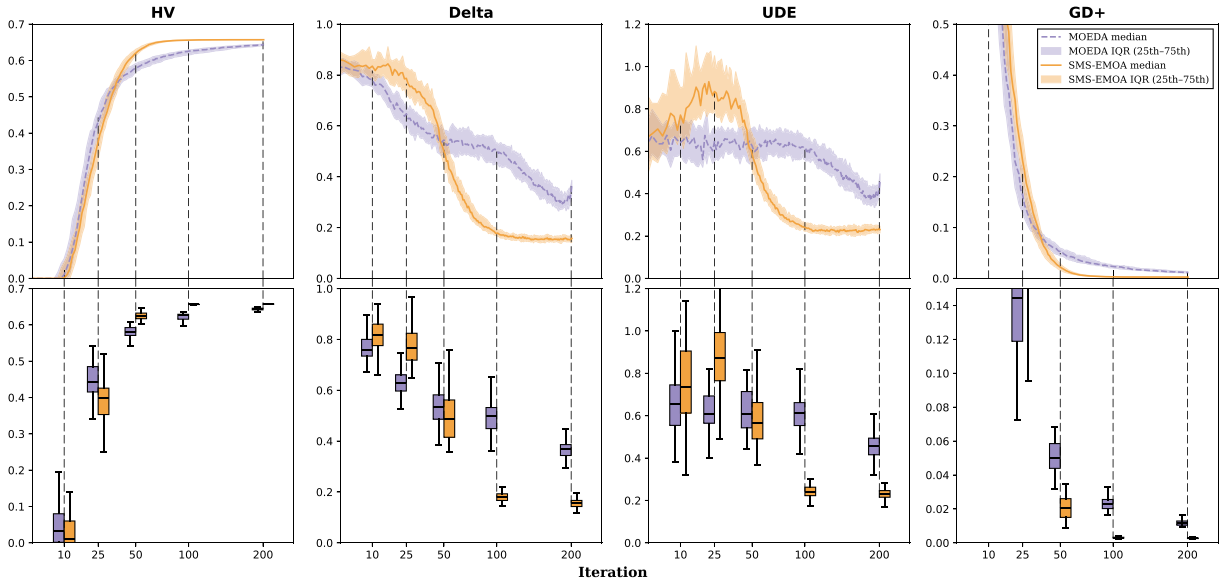


Fig. 19. ZDT1: MOEDA-GMM champion vs SMS-EMOA.

non-significant on UDE ($p = 0.06$). WeldedBeam also shows reduced α sensitivity on Δ ($p = 0.09$). The temporal evolution of these effects reveals a pattern consistent with the covariance decay mechanism. At the earliest evaluation point ($t = 10$), β is significant on only 3 out of 7 problems for HV, whereas α and K already reach 6/7. By $t = 50$, all three parameters achieve near-universal significance. This lag for β is expected: the decay schedule $\Sigma_{\min} = \alpha\,(1 - t^{\beta}\,(1 - \alpha_{\min}/\alpha))$ means that β primarily controls the shape of the later decay trajectory, so its effect on solution quality only becomes measurable once the optimization has progressed beyond the initial exploration phase.

3.4.2. *Interaction effects*

Beyond main effects, two-way interactions reveal whether the influence of one parameter depends on the level of another. Fig. 27 illustrates these interactions in the WarrenTruss case study. The reduced α sensitivity in this problem (Table 2) makes the $\beta \times K$ interaction structure particularly prominent. Each cell displays the decile-ranked HV of one (β, α) configuration; columns advance through four evaluation points to show how the sensitivity landscape evolves during the optimization. All 210 configurations are ranked together per-problem and per-timestep; green denotes the top decile (best-performing) and red the bottom decile (worst). The ranking accounts for each metric's optimization direction (higher is better for HV and Δ; lower is better for GD$^+$ and UDE). All three pairwise interactions are broadly significant at $t = 199$. The $\alpha \times K$ interaction reaches significance on all seven problems for every indicator, indicating that the optimal covariance floor depends on the number of mixture components: with fewer components, each Gaussian covers a broader region of the objective space, so a higher α is needed to maintain adequate initial exploration.

The $\beta \times K$ interaction is significant on 6–7/7 problems (non-significant only for UDE on WarrenTruss, $p = 0.15$), suggesting a compensatory relationship between decay speed and model complexity:

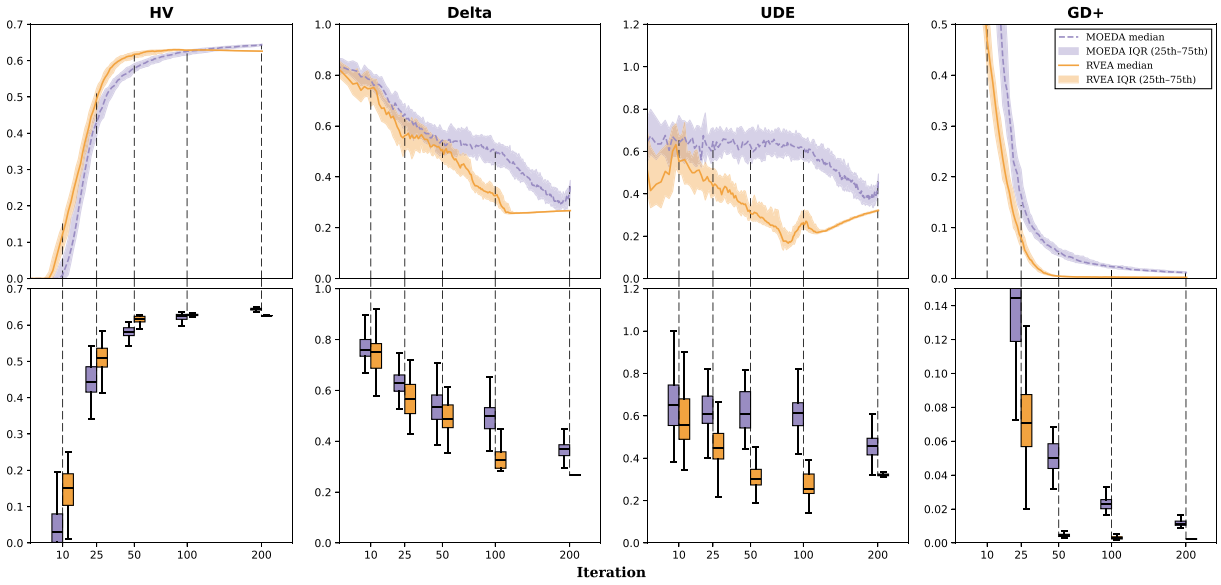


Fig. 20. ZDT1: MOEDA-GMM champion vs RVEA.

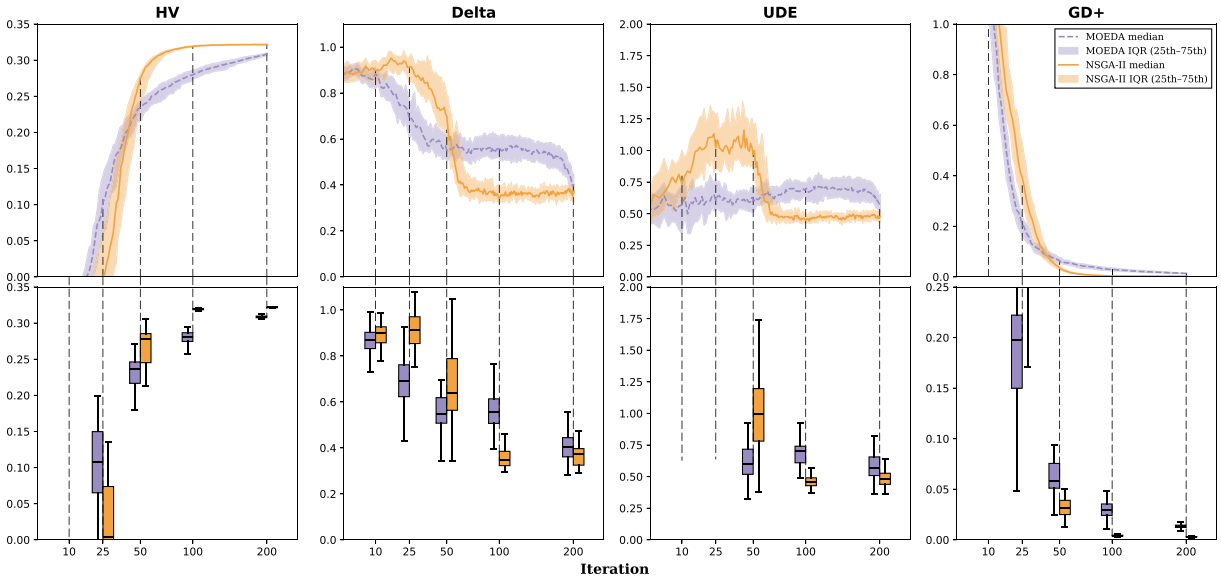


Fig. 21. ZDT2: MOEDA-GMM champion vs NSGA-II.

aggressive decay ($\beta \geq 1.0$) paired with more components, or gentle decay with fewer components, can yield comparable results. The $\beta \times \alpha$ interaction is significant on 6–7/7 problems, with non-significant cases limited to WeldedBeam (Δ $p = 0.24$; UDE $p = 0.08$) and WarrenTruss (Δ and UDE at mid-phase timesteps, $t = 75$–150). From a practitioner perspective, these interactions imply that the three MOEDA-GMM hyperparameters should not be tuned independently: the optimal α shifts depending on K, and the appropriate β depends on both.

3.4.3. Cross-problem robustness

To complement the per-problem factorial analysis, a cross-problem view ranks all 210 configurations by their Friedman rank across the seven benchmarks. Fig. 28 displays the HV ranking at $t = 199$ as a percentile-colored heatmap on the $\beta \times \alpha$ grid, with separate panels for each K value.

The composite heatmap reveals that the $K = 3$ panel contains a broad high-performing region: configurations with low-to-moderate decay ($\beta \in \{0.5, 0.6, 0.7\}$) and upper-range covariance floor ($\alpha \in \{0.009, 0.011\}$) consistently fall within the top two deciles across all problems and metrics. The $K = 2$ panel shows a comparably strong but narrower region concentrated at slightly higher β values. By contrast, panels with $K \geq 5$ display more fragmented high-performing zones, suggesting that overly complex mixture models reduce robustness at this population size. These patterns directly motivate the champion selection procedure in the next section: a robust recommendation should be drawn from the stable plateau identified here.

Detailed marginal performance distributions are given in Appendix C; per-problem interaction heatmaps for all four indicators are collected in Appendix A.

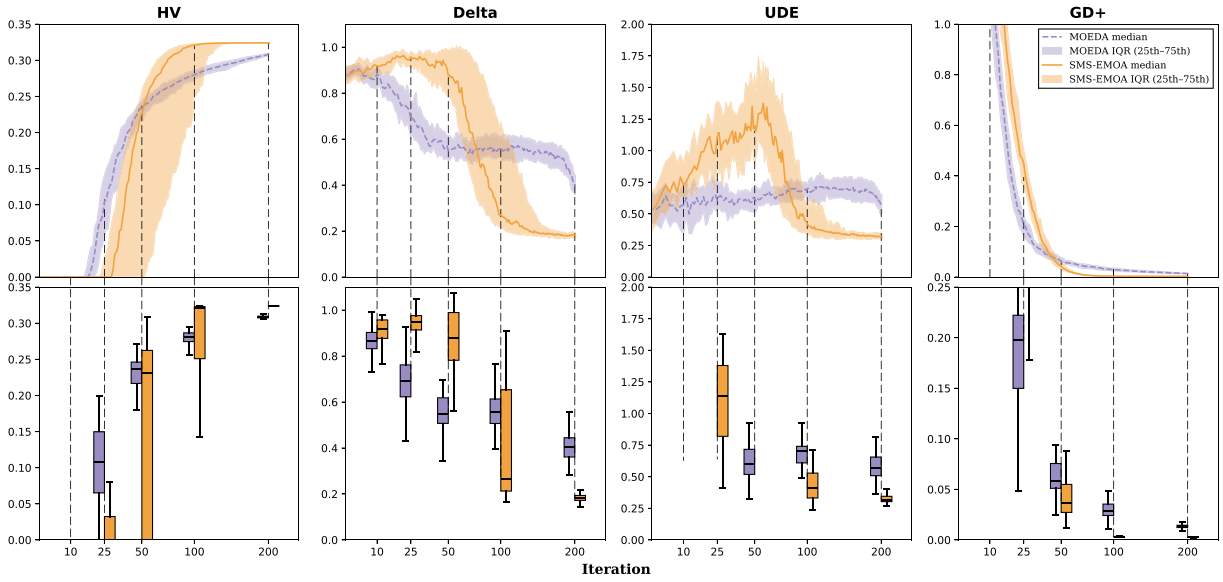


Fig. 22. ZDT2: MOEDA-GMM champion vs SMS-EMOA.

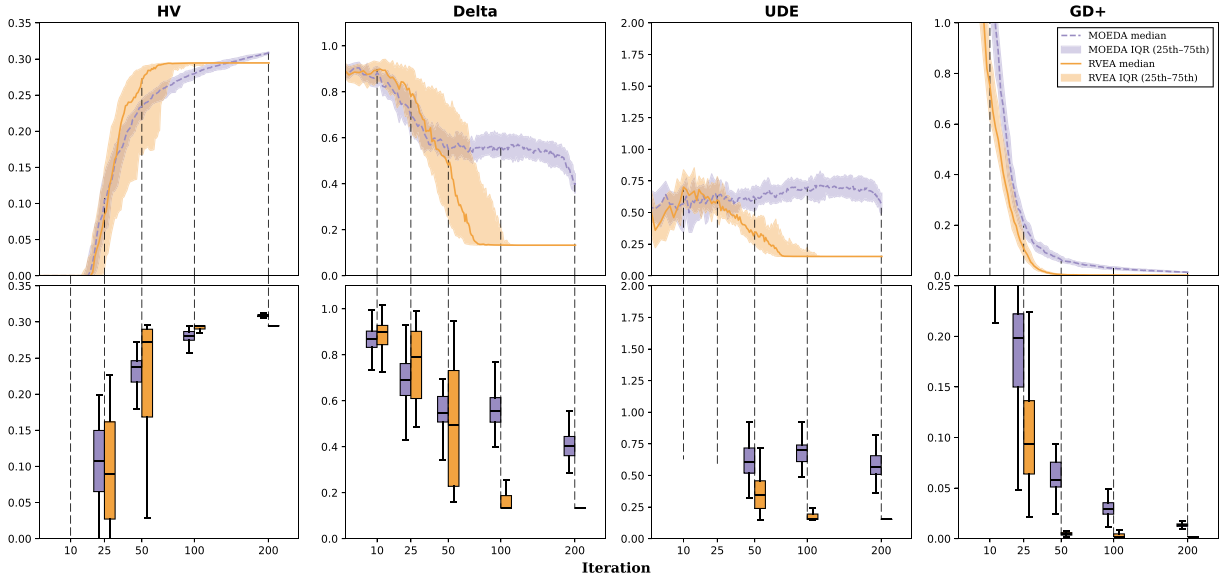


Fig. 23. ZDT2: MOEDA-GMM champion vs RVEA.

3.5. Champion configuration selection

From the sensitivity analysis, a small set of champion configurations is identified through a multi-step procedure. For each of the four indicators, all 210 configurations are ranked by their cross-problem estimated marginal mean (LMM) at the final evaluation point ($t = 199$). A weighted composite score aggregates the four indicator ranks (HV weight 0.4; Δ, UDE, GD$^+$ weight 0.2 each), reflecting the primacy of hypervolume coverage while accounting for diversity and convergence quality. The composite scores are sorted and a gap analysis identifies a natural cutoff separating a compact champion set from the remainder. To confirm that the observed ranking differences are statistically meaningful, three independent significance tests are applied to the same data: LMM [64] with *problem* as random intercept, ART ANOVA [65] as a distribution-free alternative, and a Friedman test [75] as a non-parametric blocked-design check. Robustness is verified through two additional validation steps: (i) a leave-one-problem-out jackknife confirms that the top-ranked configurations remain stable when any single problem is excluded, and (ii) multi-method concordance checks that the same configurations appear in the top-ranked sets produced by all three methods independently. A configuration is termed a consensus champion when it appears in the independently selected top-ranked pools of all three methods. The three methods show genuine methodological diversity: Friedman and ART, both non-parametric rank-based approaches, correlate most strongly (Spearman $\rho_{\text{Friedman–ART}} = 0.78$), while the parametric LMM diverges from both ($\rho_{\text{LMM–ART}} = 0.52$, $\rho_{\text{LMM–Friedman}} = 0.20$). Consensus therefore requires agreement across three distinct ranking criteria, providing a stringent robustness filter. Gap analysis on the sorted composite scores identifies the top 10% of configurations (21 out of 210) as the champion pool. Within this pool, a score discontinuity separates the top two configurations (composite scores 4.0 and 6.4) from the third (13.8), with a secondary gap between positions 6 and 7. These discontinuities suggest a two-tier structure: an elite pair and a supporting group.

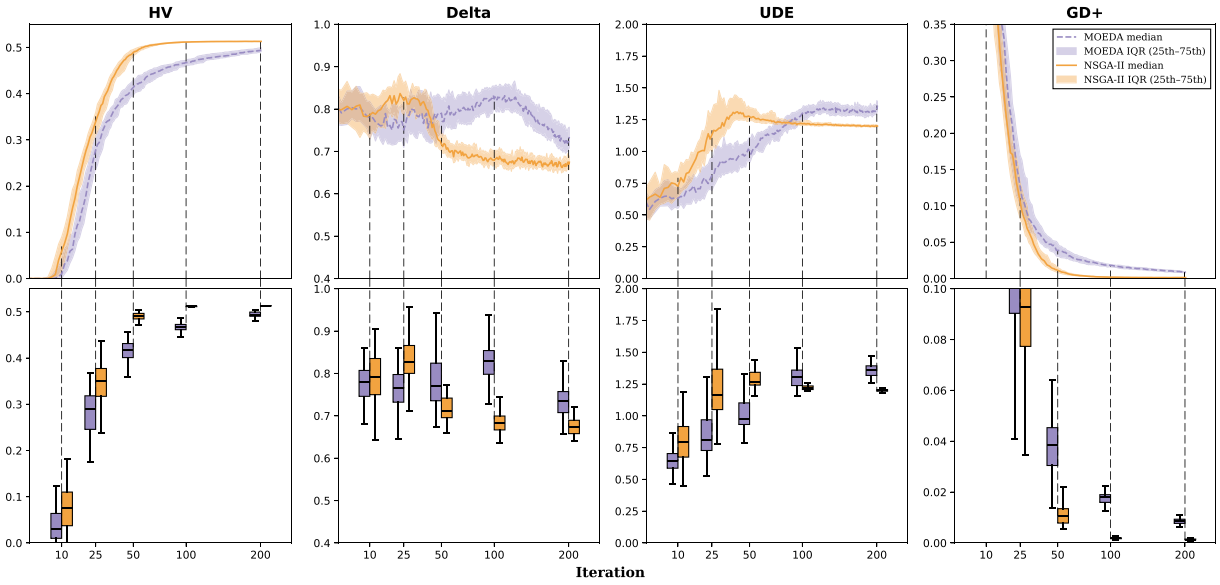


Fig. 24. ZDT3: MOEDA-GMM champion vs NSGA-II.

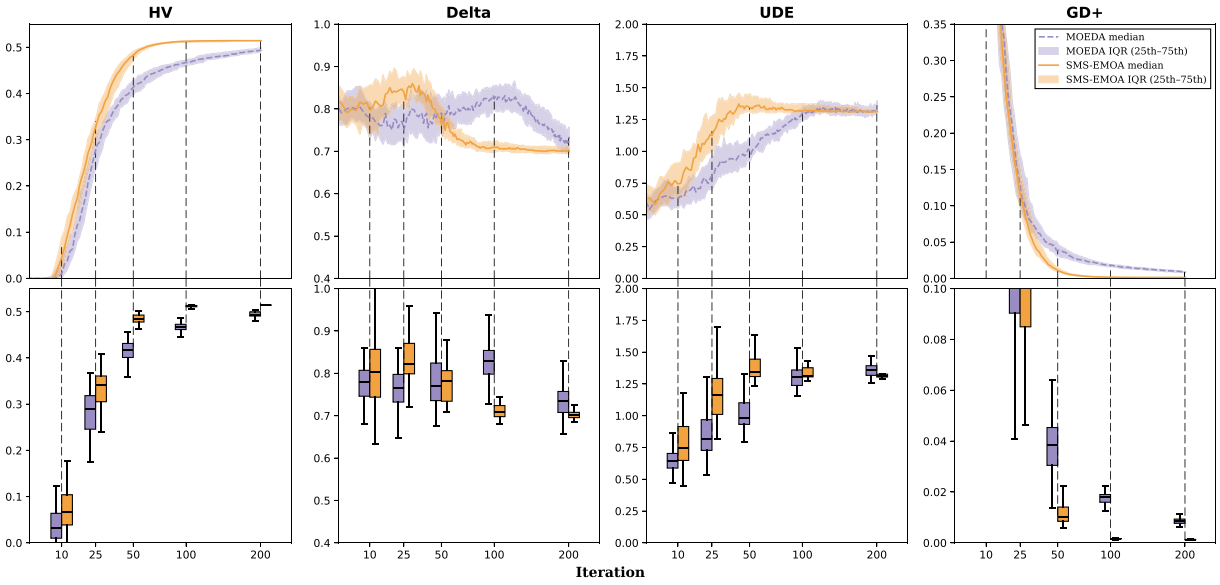


Fig. 25. ZDT3: MOEDA-GMM champion vs SMS-EMOA.

Table 3 lists the top 10 of these 21 champions together with their per-metric ranks and consensus status; the remaining 11 are omitted for brevity. Three of the 21 champions are consensus configurations, all within the top 10. The champion pool exhibits consistent parameter preferences. Among the 21 selected configurations, $K = 3$ is the most frequent (11 out of 21), followed by $K = 2$ (8 out of 21), indicating that the GMM benefits from parsimonious mixture models at this population size. Lower decay exponents are favored: $\beta \leq 1.0$ accounts for 18 of 21 champions. Since lower β produces steeper early decay of the covariance floor, these results suggest that rapid transition to exploitation benefits convergence quality at this population size. The initial covariance floor concentrates in the upper range, with $\alpha \in \{0.009, 0.011\}$ covering 15 of 21 champions. The two elite configurations separated by the primary gap exhibit complementary strengths. The overall top-ranked configuration ($\beta = 1.0$, $\alpha = 0.009$, $K = 2$; composite 4.0) achieves strong ranks across all four metrics simultaneously but does not appear in Friedman's top-ranked pool, because Friedman's cross-problem blocking favors K=3 configurations with moderate β that rank consistently across all seven problems, and therefore lacks the consensus label. The second-ranked configuration ($\beta = 0.6$, $\alpha = 0.011$, $K = 3$; composite 6.4) is a consensus champion that appears in every method pool, making it the most robust single recommendation. Leave-one-problem-out jackknife analysis confirms that both remain in the top 5 regardless of which problem is excluded.

4. Conclusions

This study introduces MOEDA-GMM, a multi-objective estimation of distribution algorithm that models the selected population using a GMM trained via EM. The algorithm was specifically developed for constrained structural optimization tasks, where the high computational

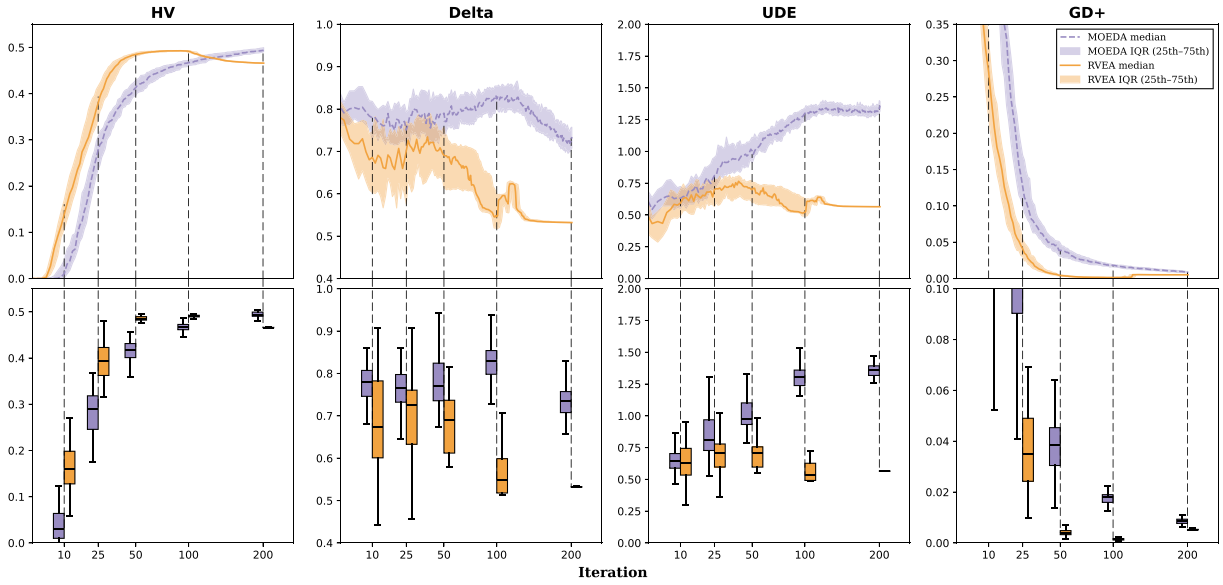


Fig. 26. ZDT3: MOEDA-GMM champion vs RVEA.

Table 2
Main effects significance at $t = 199$: number of problems (out of 7) for which the Kruskal–Wallis test rejects H_0 at $p < 0.05$.

Parameter	HV	Δ	UDE	GD^+
K (GMM components)	7/7	7/7	6/7	7/7
β (decay exponent)	7/7	7/7	7/7	7/7
α (initial covariance)	6/7	5/7	6/7	7/7

cost of function evaluations, often driven by finite element simulations and complex constraint handling, demands strategies that can extract useful information from a limited number of evaluations. To support this, MOEDA-GMM incorporates a power-law covariance scheduling mechanism, which promotes broader sampling in early iterations and focuses the search progressively as the optimization converges.

The algorithm was evaluated across seven benchmark problems: three tests from the ZDT suite and four constrained structural design tasks (welded beam, two-bar truss, four-bar truss, and Warren truss), using a 210-configuration sensitivity grid and three baseline algorithms (NSGA-II, SMS-EMOA, RVEA). MOEDA-GMM is most competitive in the initial stages of the search, where it identifies high-performing design regions and preserves diversity. This was most evident on the welded beam and four-bar truss problems, where MOEDA-GMM achieved equal or higher HV than the baselines during the first 50 iterations. On the ZDT benchmarks, SMS-EMOA and NSGA-II achieve superior final convergence. RVEA is significantly outperformed on all seven problems, with front collapse observed on three constrained structural tasks (welded beam, two-bar truss, and Warren truss). On the two-bar and Warren truss problems, MOEDA-GMM's convergence rate decreases in later stages.

These outcomes highlight both the strengths and current limitations of the method. The use of a fixed number of mixture components and a static population structure limits the flexibility of the model in highly multimodal or irregular feasible domains. Moreover, the absence of local exploitation or component merging strategies may hinder long-term convergence, particularly when the feasible region is narrow or fragmented.

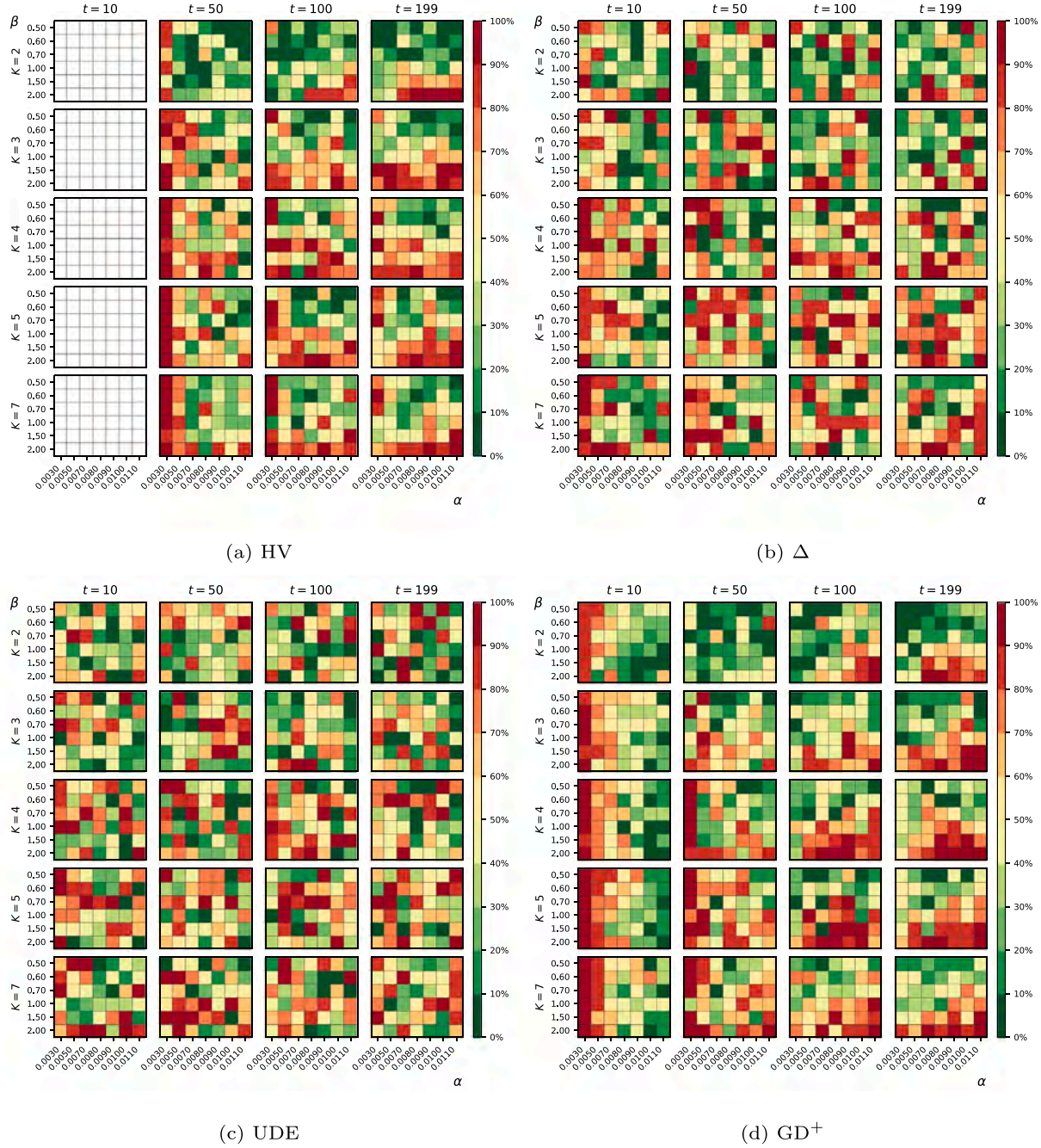
The present implementation is intentionally restricted to bi-objective problems (M=2), which correspond to the most common structural design scenarios, such as weight–deflection or cost–performance trade-offs. This choice is consistent with the algorithmic design: MOEDA-GMM is not explicitly tailored to address many-objective problems ($M \geq 3$), where the dominance relation used for ranking becomes increasingly weak and a large fraction of solutions tend to be mutually non-dominated. In such settings, effective search typically requires alternative ranking criteria (e.g., reference-vector or indicator-based methods) and enhanced diversity management. Adapting the proposed probabilistic framework to those conditions would involve larger populations for reliable mixture estimation, adaptive component control, and modified selection strategies to maintain search pressure. These extensions are identified as a direction for future work.

Future work will focus on several directions aimed at improving the scalability, robustness, and long-term convergence performance of the algorithm. One direction is to develop adaptive mixture control mechanisms that allow the number of Gaussian components to be adjusted dynamically based on the evolving population structure. This could enable the model to capture multimodal landscapes more effectively and reduce redundancy during sampling. Another direction involves the integration of feasibility-aware modeling strategies. By incorporating constraint violation information directly into the estimation process, the algorithm could improve its ability to navigate narrow or irregular feasible regions, which are common in structural optimization tasks.

Further improvements could be achieved through hybrid optimization frameworks that combine the strengths of MOEDA-GMM in early-stage exploration with the reliable convergence properties of exploitation-oriented methods, such as NSGA-II or SMS-EMOA. These hybrid approaches could be implemented either sequentially or within a single run, enabling a more effective balance between exploration and exploitation.

Model robustness could also be improved through regularization techniques. This involves exploring methods for merging and pruning components, as well as using Bayesian estimation frameworks to mitigate overfitting and improve generalization to areas of the search space that have not been explored before.

Finally, future research will focus on extending the applicability of the algorithm to high-fidelity structural design scenarios. This will include real-world systems characterized by non-linear material behavior, complex loading conditions, and multi-disciplinary constraints, where evaluation costs are particularly high and traditional optimization approaches may struggle to scale.



(a) HV (b) Δ

(c) UDE (d) GD^+

Fig. 27. Per-configuration sensitivity heatmaps for WarrenTruss across four timesteps ($t = 10, 50, 100, 199$), shown on the $\beta \times \alpha$ grid with separate rows for each K. Green = top decile; red = bottom decile. Equivalent plots for all seven problems are given in Appendix A.

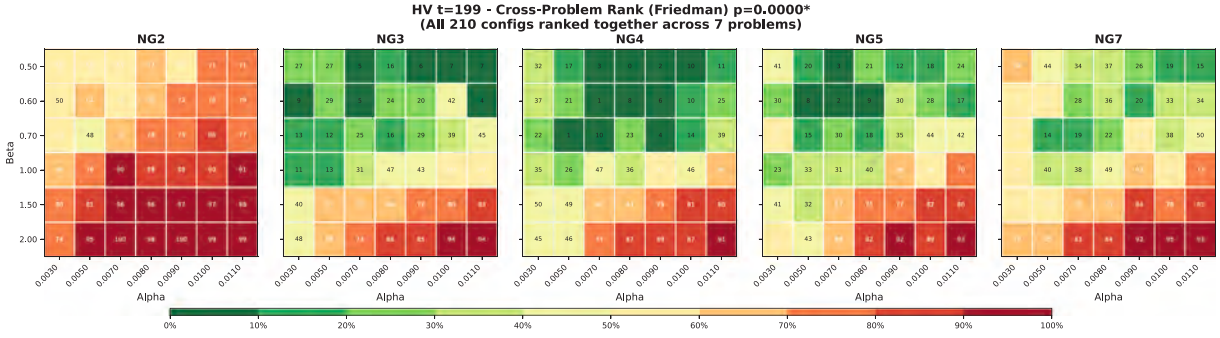


Fig. 28. Cross-problem HV rank at $t = 199$ (Friedman) shown as percentiles on the $\beta \times \alpha$ grid, with separate panels for each K. Cell values are overall ranks (1 = best of 210); color encodes the percentiles (green = top decile, red = bottom decile).

Table 3
Top 10 of the 21 champion configurations ($\leq$10th percentile) selected by weighted rank aggregation at $t = 199$. Per-metric ranks are out of 210 configurations (cross-problem LMM). Composite = weighted rank aggregate (HV = 0.4; Δ, UDE, $GD^+ = 0.2$ each). A checkmark in the Consensus column indicates that the configuration was independently selected by all three ranking methods (LMM, ART, Friedman).

Rank	Configuration	Composite	HV	Δ	UDE	GD^+	Consensus
1	β=1.0, α=0.009, K=2	4.0	5	5	3	2	–
2	β=0.6, α=0.011, K=3	6.4	1	4	22	4	✓
3	β=0.5, α=0.011, K=3	13.8	23	3	11	9	✓
4	β=0.5, α=0.011, K=2	14.6	30	6	4	3	–
5	β=0.6, α=0.011, K=2	19.0	19	25	14	18	–
6	β=0.7, α=0.011, K=3	20.0	16	2	13	53	–
7	β=0.7, α=0.011, K=2	23.6	7	10	58	36	–
8	β=0.7, α=0.008, K=2	25.4	17	35	31	27	–
9	β=0.7, α=0.008, K=3	26.2	27	20	50	7	✓
10	β=0.6, α=0.009, K=3	27.8	10	29	47	43	–

CRediT authorship contribution statement

Mattia Siviero: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Jonathan Melchiorre:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Conceptualization. **Marco M. Rosso:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Giansalvo Cirrincione:** Writing – review & editing, Validation, Supervision, Project administration, Methodology. **Giuseppe C. Marano:** Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

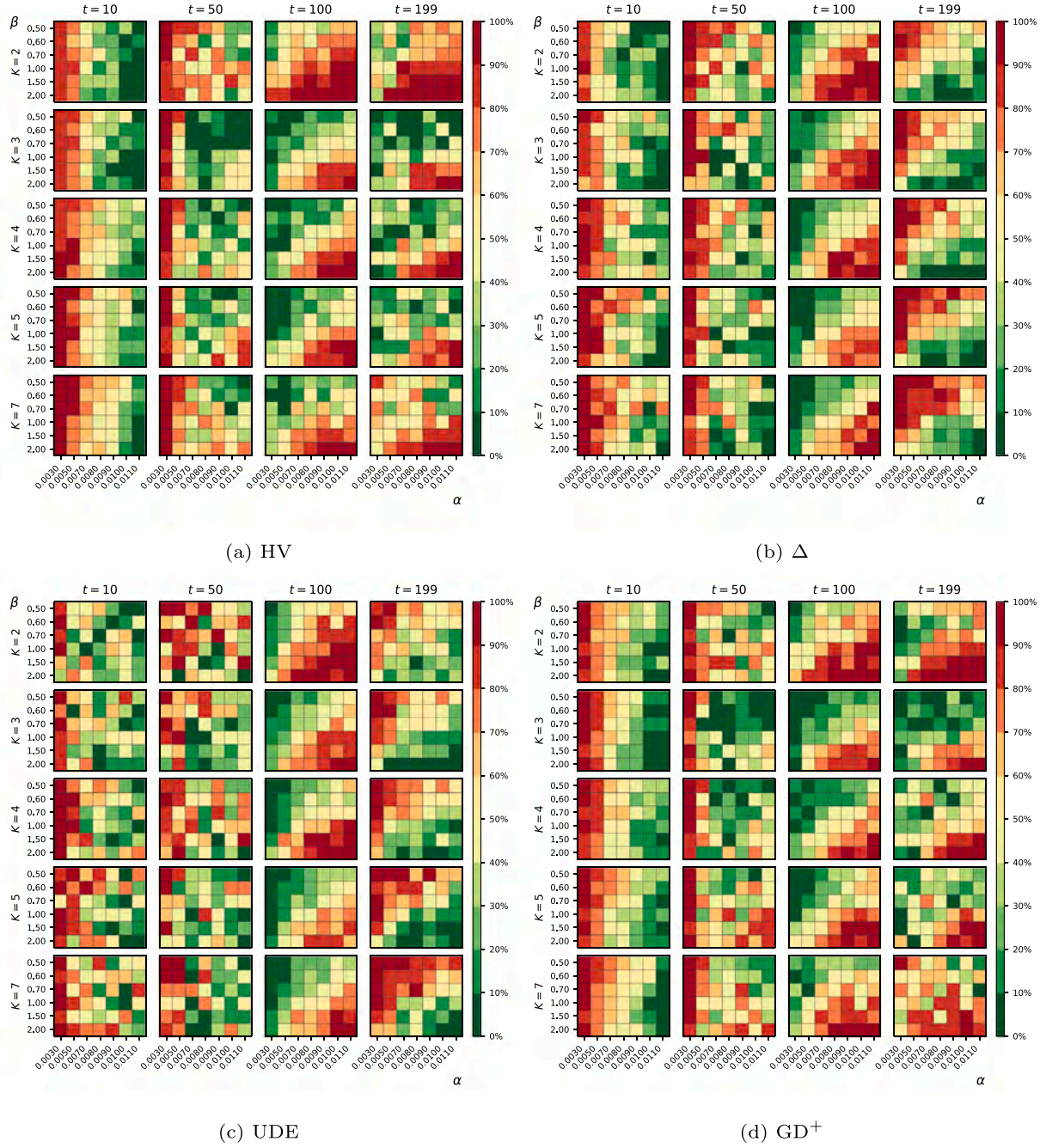
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was carried out within the FAIR – Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Appendix A. Per-problem sensitivity heatmaps

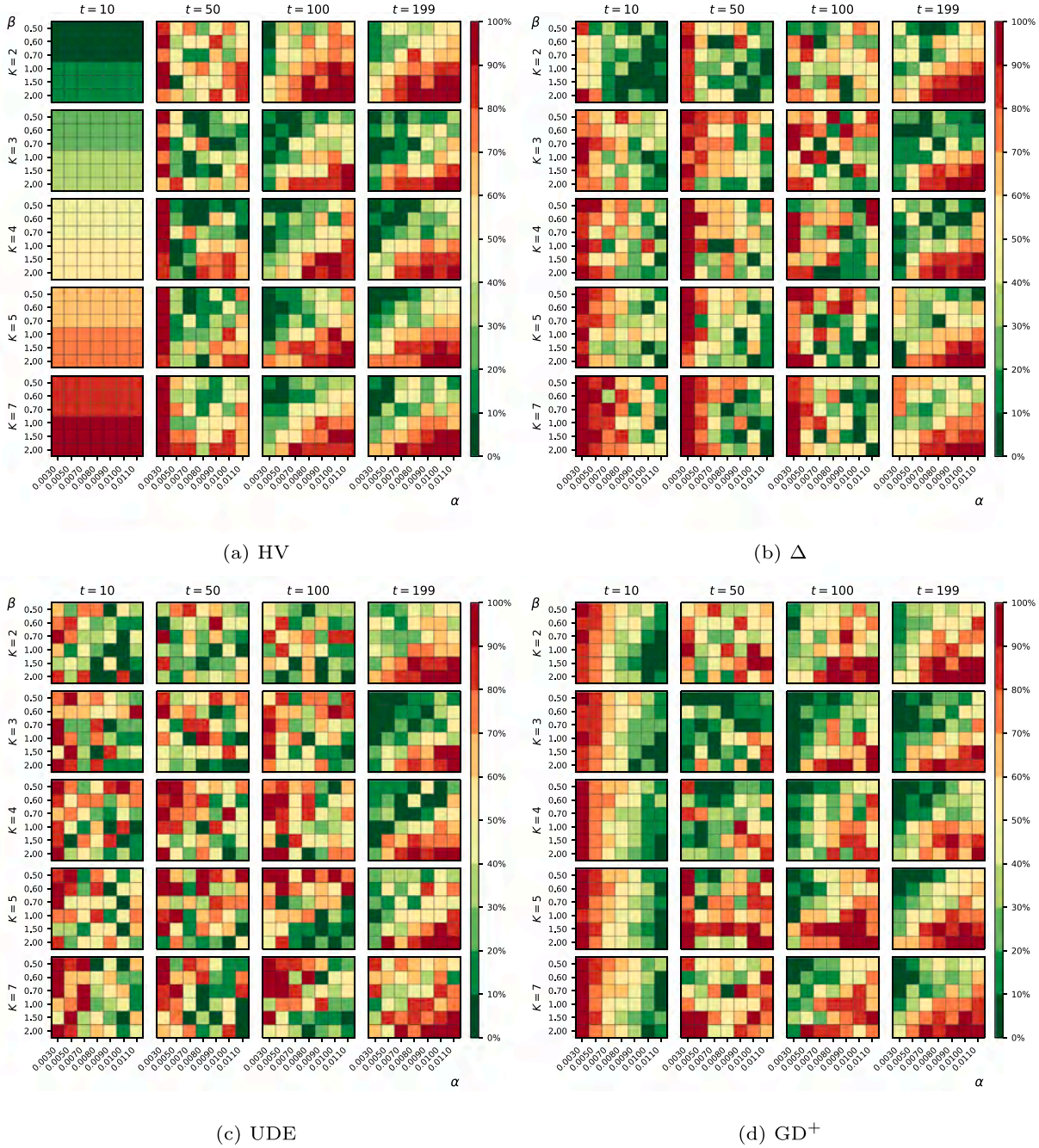
This appendix collects the per-configuration decile-ranked heatmaps for the six problems not shown in the main text (Fig. 27 presents WarrenTruss). Each figure displays four indicators (HV, Δ, UDE, GD^+) as subfigures, with columns corresponding to evaluation timesteps ($t = 10, 50, 100, 199$) and rows to the number of Gaussian components ($K = 2, 3, 4, 5, 7$). Cells encode the percentile rank of each (β, α) configuration among all 210 configurations evaluated on the same problem and timestep; green denotes the top decile and red the bottom decile (Figs. A.29–A.34).



(a) HV (b) Δ

(c) UDE (d) GD^+

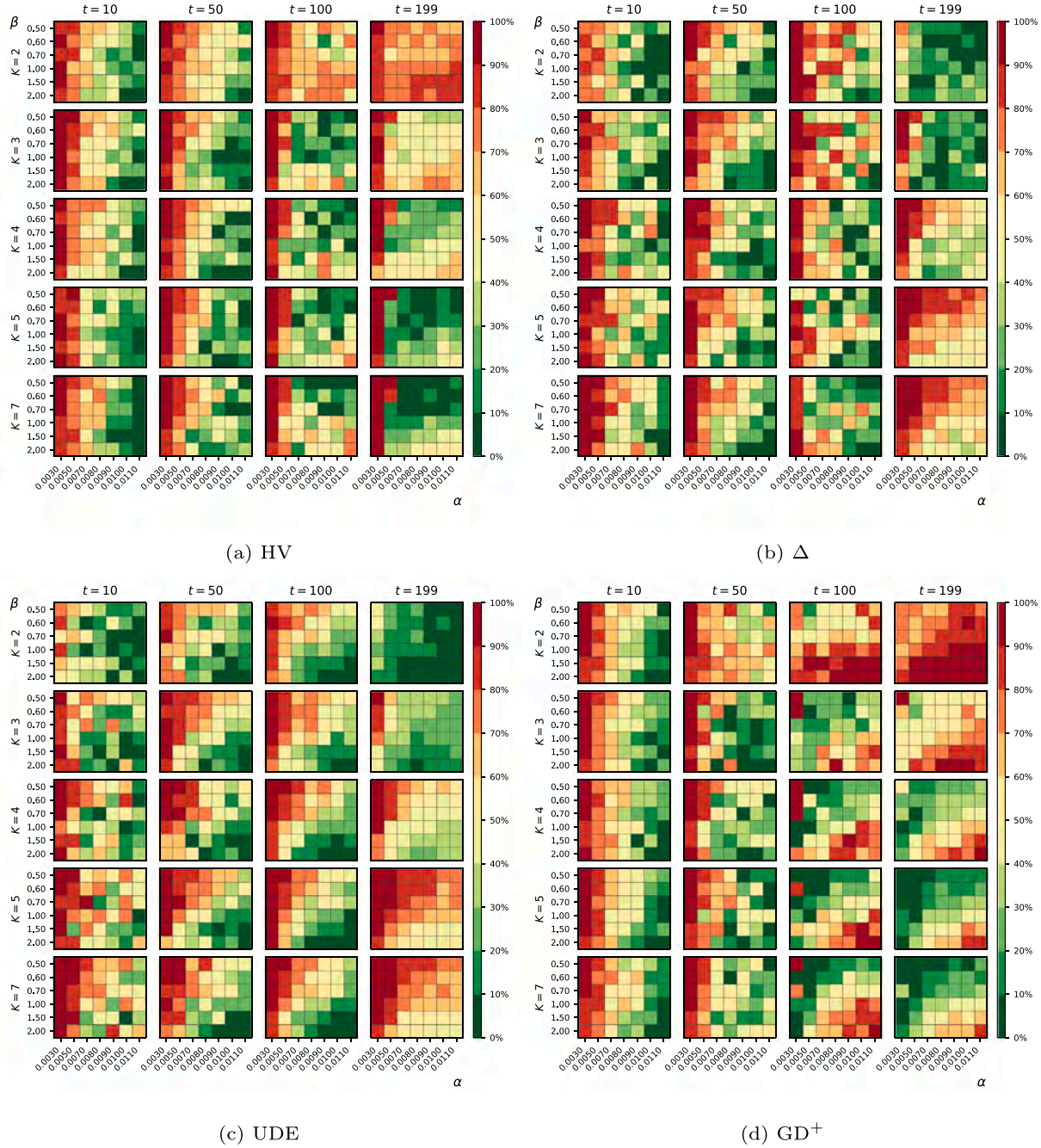
Fig. A.29. ZDT1: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).



(a) HV (b) Δ

(c) UDE (d) GD^+

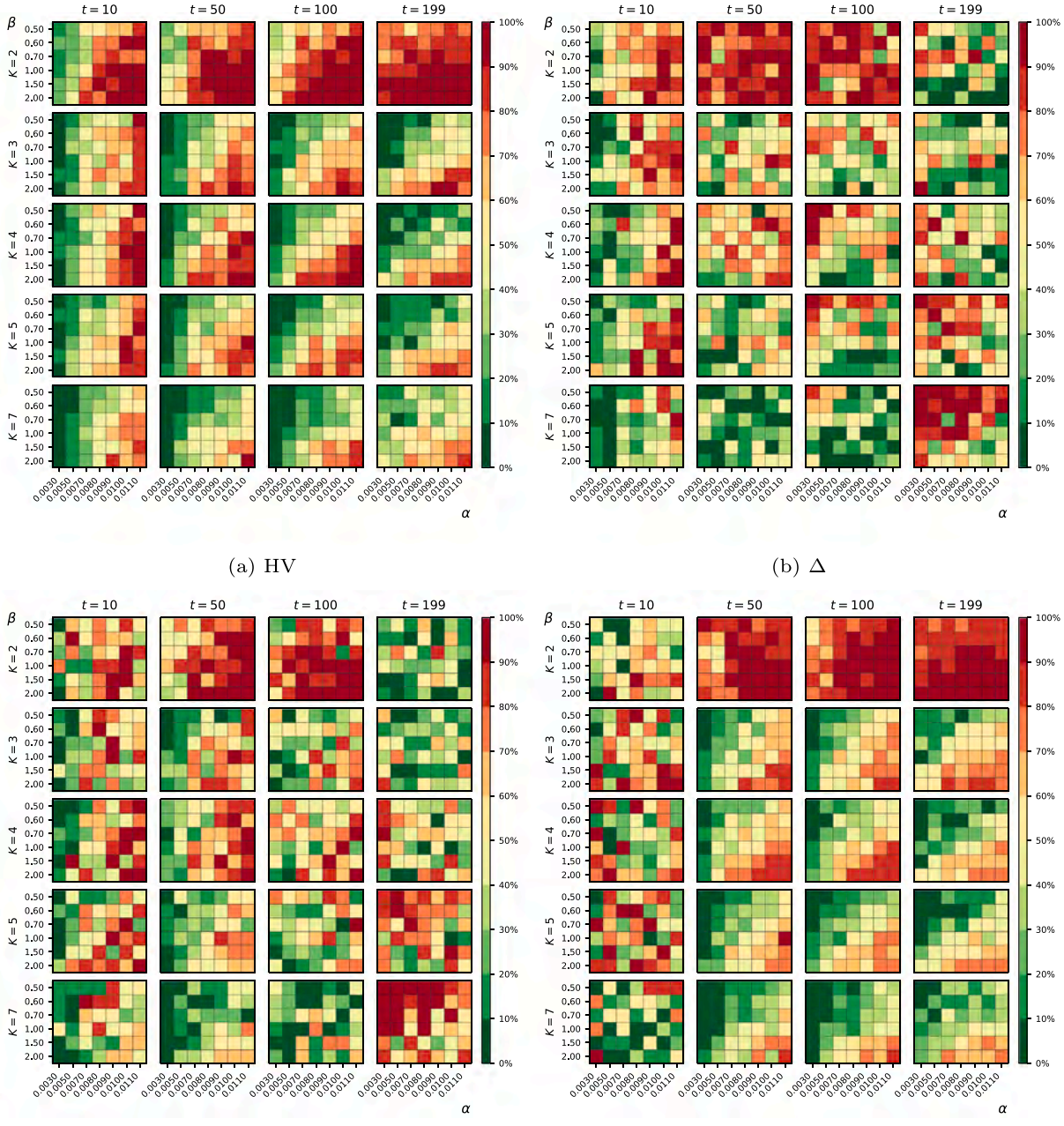
Fig. A.30. ZDT2: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).



(a) HV (b) Δ

(c) UDE (d) GD$^+$

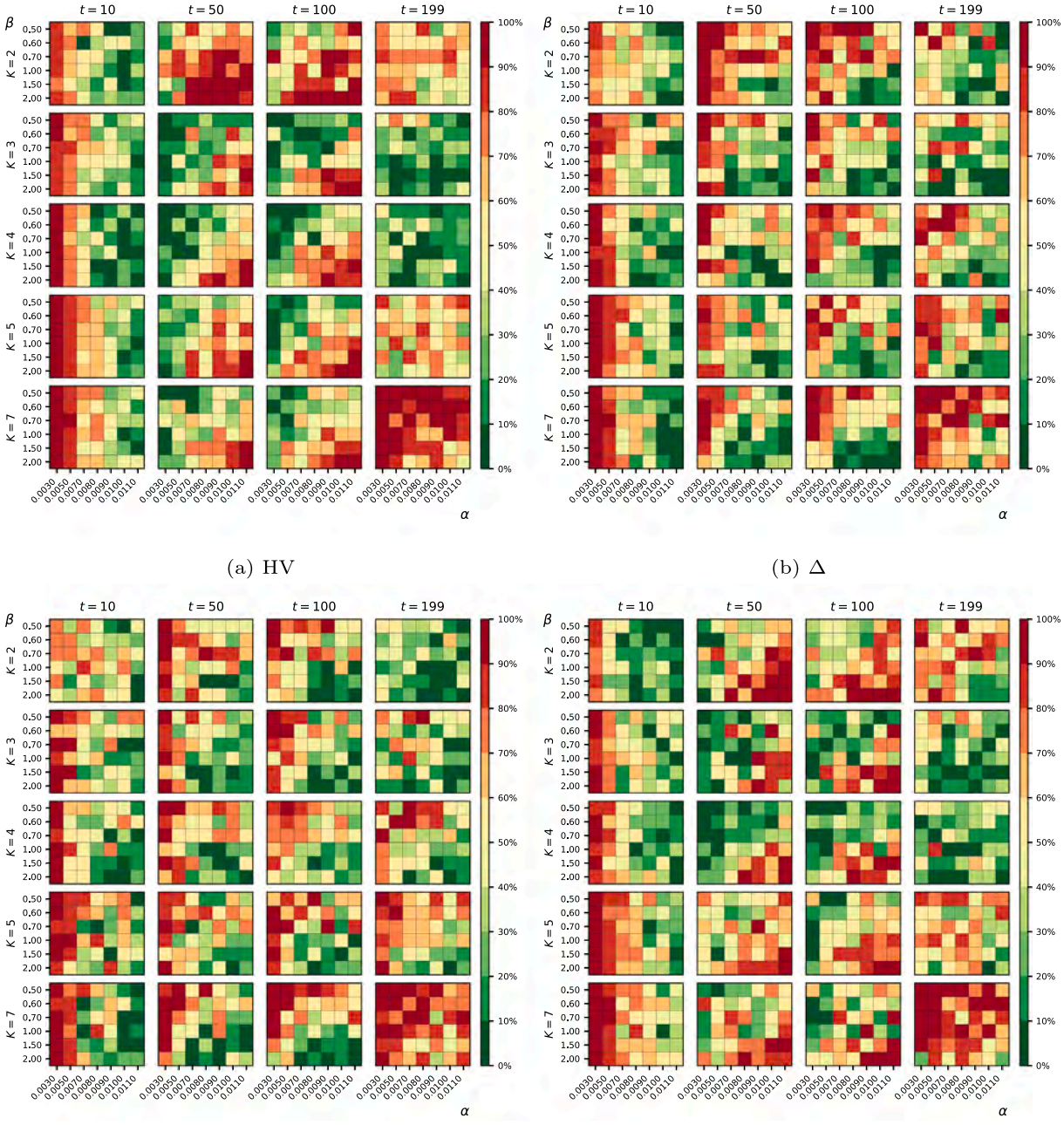
Fig. A.31. ZDT3: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).



(a) HV (b) Δ

(c) UDE (d) GD^+

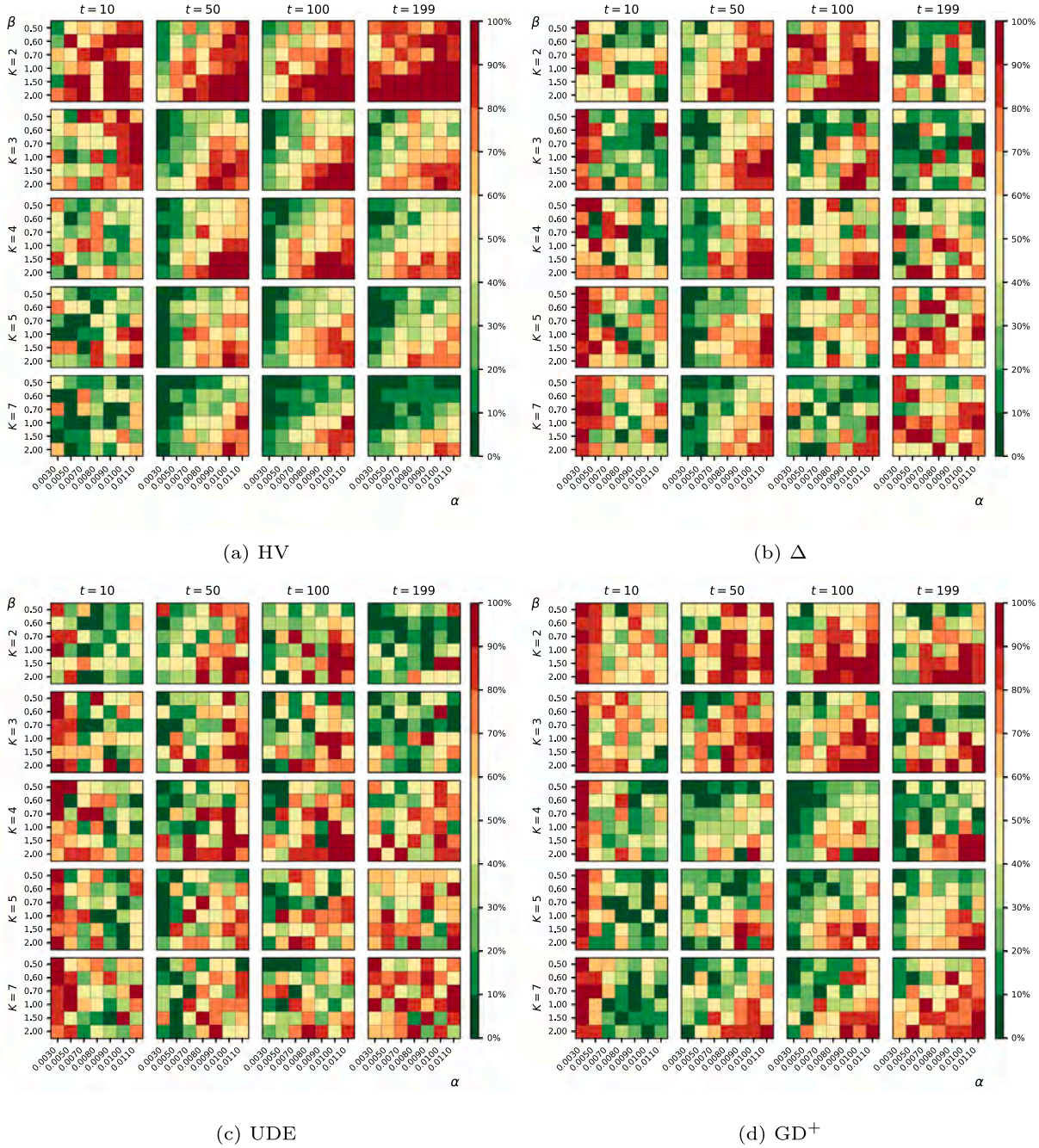
Fig. A.32. Two-Bar Truss: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).



(a) HV (b) Δ

(c) UDE (d) GD^+

Fig. A.33. Four-Bar Truss: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).



(a) HV (b) Δ

(c) UDE (d) GD^+

Fig. A.34. Welded Beam: per-configuration sensitivity heatmaps for all four indicators. Rows = K values; columns = evaluation timesteps. Cell color encodes the decile rank (green = top 10%, red = bottom 10%).

Appendix B. Non-parametric statistical analysis

This appendix reports the statistical tests underpinning the baseline comparison in Section 3.5. All p-values are Bonferroni-adjusted for multiple comparisons. For each problem, the per-problem best MOEDA configuration (by mean HV at the given timestep) is compared against SMS-EMOA, NSGA-II, and RVEA using Mann–Whitney U tests ($n = 50$ runs per algorithm). Kruskal–Wallis omnibus tests were significant for all seven problems at $t = 199$ (all $p < 0.001$); pairwise results are reported at four milestones ($t = 10, 50, 100, 199$).

B.1. Within-problem pairwise tests

Tables B.5–B.7 report the same Mann–Whitney U pairwise tests as Table B.4, evaluated at three earlier milestones ($t = 10$, $t = 50$, $t = 100$). For each problem, the per-problem best MOEDA champion (by mean HV at that timestep) is compared against each baseline. These tables provide statistical backing for the convergence-trajectory observations in Section 3.

Table B.4
Per-problem pairwise significance for HV at $t = 199$. Each cell shows whether MOEDA wins (+), loses (−), or is not significant (n.s.). Mann–Whitney U, Bonferroni-adjusted, n=50 runs.

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	− (p≈0)	+ (p≈0)
ZDT2	− (p≈0)	− (p≈0)	+ (p≈0)
ZDT3	− (p≈0)	− (p≈0)	+ (p≈0)
Truss2D2bar	− (p≈0)	+ (p≈0)	+ (p≈0)
Truss2DFourbar	− (p≈0)	+ (p≈0)	+ (p≈0)
WeldedBeam	+ (p=0.042)	n.s.	+ (p≈0)
WarrenTruss	− (p≈0)	− (p≈0)	+ (p≈0)

Table B.5
Per-problem pairwise significance for HV at $t = 10$ (550 FE). +/− = MOEDA wins/loses; n.s. = not significant.

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	n.s.	n.s.	− (p<0.001)
ZDT2	n.s.	n.s.	n.s.
ZDT3	n.s.	n.s.	− (p<0.001)
Truss2D2bar	n.s.	n.s.	+ (p<0.001)
Truss2DFourbar	+ (p<0.001)	+ (p<0.001)	+ (p<0.001)
WeldedBeam	n.s.	n.s.	+ (p<0.001)
WarrenTruss	n.s.	n.s.	n.s.

Table B.6
Per-problem pairwise significance for HV at $t = 50$ (2 550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p<0.001)	− (p<0.001)	− (p<0.001)
ZDT2	n.s.	− (p=0.003)	n.s.
ZDT3	− (p<0.001)	− (p<0.001)	− (p<0.001)
Truss2D2bar	− (p<0.001)	n.s.	+ (p<0.001)
Truss2DFourbar	− (p<0.001)	+ (p<0.001)	+ (p<0.001)
WeldedBeam	n.s.	n.s.	+ (p<0.001)
WarrenTruss	n.s.	n.s.	+ (p<0.001)

Table B.7
Per-problem pairwise significance for HV at $t = 100$ (5 050 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p<0.001)	− (p<0.001)	n.s.
ZDT2	n.s.	− (p<0.001)	− (p<0.001)
ZDT3	− (p<0.001)	− (p<0.001)	− (p<0.001)
Truss2D2bar	− (p<0.001)	n.s.	+ (p<0.001)
Truss2DFourbar	− (p<0.001)	+ (p<0.001)	+ (p<0.001)
WeldedBeam	n.s.	n.s.	+ (p<0.001)
WarrenTruss	− (p<0.001)	− (p<0.001)	+ (p<0.001)

At $t = 10$ (550 function evaluations), most comparisons are non-significant. MOEDA is significantly better than all three baselines on Truss2DFourbar and better than RVEA on two additional constrained benchmarks. RVEA outperforms MOEDA on ZDT1 and ZDT3, consistent with effective decomposition-based search on unconstrained problems at early iterations.

By $t = 50$, MOEDA is significantly better than RVEA on 4 of 7 problems (all constrained benchmarks and WarrenTruss). On ZDT benchmarks, SMS-EMOA and NSGA-II are significantly better than MOEDA. Against NSGA-II, MOEDA is significantly better on Truss2DFourbar ($p <$ 0.001).

At $t = 100$, the pattern is consistent with $t = 199$ (Table B.4): MOEDA remains significantly better than NSGA-II on Truss2DFourbar and better than RVEA on all constrained benchmarks, while SMS-EMOA is significantly better on ZDT1 and ZDT3. Compared with $t = 50$, the main change is on WarrenTruss, where SMS-EMOA and NSGA-II become significantly better than MOEDA.

MOEDA achieves significantly better front spread (Δ) than SMS-EMOA on 5 of 7 problems, and better than NSGA-II on 4 of 7. RVEA, however, achieves superior Δ on the two truss benchmarks where it does not suffer front collapse. On ZDT benchmarks, MOEDA's spread is competitive or superior. WarrenTruss is the only problem where MOEDA significantly outperforms both SMS-EMOA and NSGA-II on this indicator.

UDE combines convergence and diversity into a single scalar; lower is better. MOEDA loses to SMS-EMOA on 6 of 7 problems and to

Table B.8
Per-problem pairwise significance for Δ (spread) at $t = 199$. +/− = MOEDA wins/loses; n.s. = not significant. Mann–Whitney U, Bonferroni-adjusted, n=50 runs.

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	+ (p≈0)	n.s.	+ (p≈0)
ZDT2	+ (p≈0)	n.s.	+ (p≈0)
ZDT3	n.s.	+ (p<0.001)	+ (p≈0)
Truss2D2bar	+ (p≈0)	+ (p≈0)	− (p≈0)
Truss2DFourbar	+ (p≈0)	+ (p≈0)	− (p≈0)
WeldedBeam	n.s.	n.s.	− (p≈0)
WarrenTruss	+ (p=0.001)	+ (p=0.016)	n.s.

Table B.9
Per-problem pairwise significance for UDE at $t = 199$. +/− = MOEDA wins/loses; n.s. = not significant. Mann–Whitney U, Bonferroni-adjusted, n=50 runs.

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	+ (p<0.001)	− (p≈0)
ZDT2	− (p≈0)	n.s.	− (p≈0)
ZDT3	n.s.	− (p≈0)	− (p≈0)
Truss2D2bar	− (p≈0)	− (p≈0)	+ (p≈0)
Truss2DFourbar	− (p≈0)	− (p<0.001)	+ (p<0.001)
WeldedBeam	− (p<0.001)	n.s.	+ (p≈0)
WarrenTruss	− (p=0.023)	n.s.	− (p<0.001)

Table B.10
Per-problem pairwise significance for GD^+ at $t = 199$. +/− = MOEDA wins/loses; n.s. = not significant. Mann–Whitney U, Bonferroni-adjusted, n=50 runs.

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	− (p≈0)	− (p≈0)
ZDT2	− (p≈0)	− (p≈0)	− (p≈0)
ZDT3	− (p≈0)	− (p≈0)	− (p≈0)
Truss2D2bar	− (p≈0)	+ (p<0.001)	+ (p≈0)
Truss2DFourbar	− (p≈0)	n.s.	− (p≈0)
WeldedBeam	− (p<0.001)	n.s.	+ (p≈0)
WarrenTruss	+ (p<0.001)	+ (p<0.001)	+ (p<0.001)

Table B.11
Per-problem pairwise significance for Δ at $t = 10$ (550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p<0.001)	− (p=0.003)	n.s.
ZDT2	− (p<0.001)	− (p<0.001)	− (p=0.048)
ZDT3	n.s.	n.s.	+ (p=0.041)
Truss2D2bar	n.s.	n.s.	− (p≈0)
Truss2DFourbar	− (p<0.001)	n.s.	n.s.
WeldedBeam	− (p≈0)	− (p<0.001)	− (p<0.001)
WarrenTruss	n.s.	n.s.	n.s.

Table B.12
Per-problem pairwise significance for UDE at $t = 10$ (550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	+ (p<0.001)	+ (p<0.001)	n.s.
ZDT2	+ (p<0.001)	+ (p<0.001)	n.s.
ZDT3	+ (p<0.001)	+ (p<0.001)	n.s.
Truss2D2bar	n.s.	n.s.	+ (p<0.001)
Truss2DFourbar	n.s.	− (p<0.001)	n.s.
WeldedBeam	+ (p<0.001)	+ (p<0.001)	n.s.
WarrenTruss	n.s.	n.s.	n.s.

Table B.13
Per-problem pairwise significance for GD^+ at $t = 10$ (550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	+ (p<0.001)	+ (p<0.001)	− (p≈0)
ZDT2	+ (p≈0)	+ (p<0.001)	− (p<0.001)
ZDT3	+ (p=0.011)	n.s.	− (p≈0)
Truss2D2bar	n.s.	n.s.	+ (p≈0)
Truss2DFourbar	+ (p<0.001)	+ (p≈0)	n.s.
WeldedBeam	n.s.	+ (p<0.001)	+ (p≈0)
WarrenTruss	+ (p=0.005)	+ (p=0.023)	+ (p<0.001)

Table B.14
Per-problem pairwise significance for Δ at $t = 50$ (2 550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	n.s.	+ (p≈0)	n.s.
ZDT2	− (p<0.001)	n.s.	n.s.
ZDT3	n.s.	+ (p<0.001)	+ (p<0.001)
Truss2D2bar	n.s.	+ (p<0.001)	− (p≈0)
Truss2DFourbar	+ (p≈0)	+ (p<0.001)	− (p≈0)
WeldedBeam	n.s.	n.s.	− (p≈0)
WarrenTruss	− (p<0.001)	− (p<0.001)	− (p=0.015)

NSGA-II on 3 of 7, reflecting weaker final convergence on ZDT benchmarks. However, MOEDA significantly outperforms RVEA on both truss benchmarks and WeldedBeam, consistent with RVEA's front collapse on constrained problems.

GD^+ measures convergence quality toward the reference front. On ZDT benchmarks, all three baselines significantly outperform MOEDA, reflecting SMS-EMOA's and NSGA-II's convergence advantage. WarrenTruss is the only problem where MOEDA significantly outperforms all three baselines on this indicator. On constrained structural problems, results are mixed: MOEDA wins on Truss2D2bar vs NSGA-II and RVEA but loses to SMS-EMOA, consistent with the overall pattern across metrics.

Tables B.8–B.10 report pairwise tests at $t = 199$ (10000 FE) for Δ, UDE, and GD^+ Tables B.11–B.19 report pairwise tests at $t = 10$ (550 FE), $t = 50$ (2550 FE) and $t = 100$ (5050 FE) for the same metrics. The corresponding HV tables are given in Tables B.4–B.7. These results support the observation that MOEDA-GMM is most competitive in the early stages of the search.

B.2. Cross-problem analysis

The cross-problem analyses compare the top-5 MOEDA champions (by weighted LMM composite, Section 3.5) against the three baselines, for a total of 8 algorithms. Restricting the treatment count improves

Table B.15
Per-problem pairwise significance for UDE at $t = 50$ (2 550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	n.s.	− (p<0.001)	− (p≈0)
ZDT2	+ (p<0.001)	+ (p<0.001)	− (p<0.001)
ZDT3	+ (p≈0)	+ (p≈0)	− (p≈0)
Truss2D2bar	n.s.	− (p<0.001)	+ (p≈0)
Truss2DFourbar	− (p≈0)	− (p<0.001)	+ (p<0.001)
WeldedBeam	n.s.	n.s.	+ (p<0.001)
WarrenTruss	+ (p<0.001)	+ (p<0.001)	n.s.

Table B.16
Per-problem pairwise significance for GD^+ at $t = 50$ (2 550 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	− (p≈0)	− (p≈0)
ZDT2	n.s.	− (p<0.001)	− (p≈0)
ZDT3	− (p≈0)	− (p≈0)	− (p≈0)
Truss2D2bar	− (p<0.001)	n.s.	+ (p≈0)
Truss2DFourbar	− (p≈0)	+ (p<0.001)	− (p≈0)
WeldedBeam	− (p=0.002)	n.s.	+ (p≈0)
WarrenTruss	n.s.	n.s.	+ (p≈0)

Table B.17
Per-problem pairwise significance for Δ at $t = 100$ (5 050 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	+ (p≈0)	+ (p=0.021)	+ (p<0.001)
ZDT2	n.s.	+ (p<0.001)	+ (p≈0)
ZDT3	+ (p≈0)	+ (p≈0)	+ (p≈0)
Truss2D2bar	n.s.	n.s.	− (p≈0)
Truss2DFourbar	+ (p≈0)	+ (p<0.001)	− (p≈0)
WeldedBeam	n.s.	n.s.	− (p≈0)
WarrenTruss	n.s.	n.s.	n.s.

Table B.18
Per-problem pairwise significance for UDE at $t = 100$ (5 050 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	n.s.	− (p≈0)
ZDT2	− (p<0.001)	− (p≈0)	− (p≈0)
ZDT3	+ (p<0.001)	n.s.	− (p≈0)
Truss2D2bar	n.s.	− (p<0.001)	+ (p≈0)
Truss2DFourbar	− (p≈0)	− (p<0.001)	+ (p≈0)
WeldedBeam	n.s.	n.s.	+ (p<0.001)
WarrenTruss	n.s.	n.s.	n.s.

Table B.19
Per-problem pairwise significance for GD^+ at $t = 100$ (5 050 FE).

Problem	vs SMS-EMOA	vs NSGA-II	vs RVEA
ZDT1	− (p≈0)	− (p≈0)	− (p≈0)
ZDT2	− (p≈0)	− (p≈0)	− (p≈0)
ZDT3	− (p≈0)	− (p≈0)	− (p≈0)
Truss2D2bar	− (p<0.001)	n.s.	+ (p≈0)
Truss2DFourbar	− (p≈0)	n.s.	− (p≈0)
WeldedBeam	− (p<0.001)	n.s.	+ (p≈0)
WarrenTruss	− (p<0.001)	− (p<0.001)	+ (p<0.001)

the statistical power of the Friedman test, which loses sensitivity as the number of treatments grows relative to the number of blocks. At $t = 10$, no significant differences are detected ($p = 0.56$). By $t = 50$, differences become significant ($p = 0.016$) with NSGA-II ranked first (2.00) and the best MOEDA champion at rank 3.71. From $t = 100$, SMS-EMOA takes rank 1 (2.00) and MOEDA β=0.5, α=0.011, K=3 reaches rank 3.14. At $t = 199$, SMS-EMOA retains rank 1 (1.86) while NSGA-II returns to 3.57. RVEA moves from 5.00 at $t = 10$ to 8.00 (last) at $t = 199$, consistent with its front deterioration on constrained problems .

Table B.20
Friedman average rank progression for HV across 7 problems (8 algorithms: 5 MOEDA champions + 3 baselines). Lower rank = better. Boldface marks the best rank at each timestep.

Algorithm	t=10 (550 FE)	t=50 (2 550 FE)	t=100 (5 050 FE)	t=199 (10 050 FE)
MOEDA β=0.6, α=0.011, K=3	4.57	4.29	4.43	3.00
MOEDA β=0.5, α=0.011, K=3	4.14	3.71	3.14	3.14
MOEDA β=0.5, α=0.011, K=2	**3.57**	4.86	5.71	4.86
MOEDA β=0.6, α=0.011, K=2	4.43	5.86	5.71	5.43
MOEDA β=1.0, α=0.009, K=2	5.29	6.29	6.57	6.14
SMS-EMOA	5.43	3.36	**2.00**	**1.86**
NSGA-II	**3.57**	**2.00**	2.43	3.57
RVEA	5.00	5.64	6.00	8.00
Friedman p	0.56 (n.s.)	0.016	<0.001	<0.001

Table B.21
Cross-problem omnibus p-values at $t = 199$ for all four indicators (8 algorithms). All tests are significant at $\alpha = 0.05$.

Indicator	Friedman	ART ANOVA	LMM
HV	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
Δ	0.046	$< 10^{-4}$	$< 10^{-4}$
UDE	$< 10^{-2}$	$< 10^{-4}$	$< 10^{-4}$
GD^+	$< 10^{-2}$	$< 10^{-4}$	$< 10^{-4}$

Table B.22
Friedman average ranks: HV.

Rank	Algorithm	Avg. Friedman Rank
1	SMS-EMOA	1.86
2	β=0.6, α=0.011, K=3	3.00
3	β=0.5, α=0.011, K=3	3.14
4	NSGA-II	3.57
5	β=0.5, α=0.011, K=2	4.86
6	β=0.6, α=0.011, K=2	5.43
7	β=1.0, α=0.009, K=2	6.14
8	RVEA	8.00

Table B.23
Friedman average ranks: Δ.

Rank	Algorithm	Avg. Friedman Rank
1	SMS-EMOA	2.14
2	NSGA-II	2.86
3	RVEA	4.43
4	β=0.6, α=0.011, K=2	4.86
5	β=0.5, α=0.011, K=3	5.00
6	β=0.5, α=0.011, K=2	5.29
7	β=0.6, α=0.011, K=3	5.57
8	β=1.0, α=0.009, K=2	5.86

Table B.21 reports the cross problem omnibus P values for all three statistical tests on the four indicators. All three methods detect significant differences at $\alpha = 0.05$. The Friedman test yields the weakest signal on Δ ($p = 0.046$); ART ANOVA and LMM confirm significance on all four indicators ($p < 10^{-4}$).

Table B.24
Friedman average ranks: UDE.

Rank	Algorithm	Avg. Friedman Rank
1	SMS-EMOA	1.43
2	NSGA-II	3.00
3	RVEA	4.43
4	β=0.6, α=0.011, K=2	4.86
5	β=1.0, α=0.009, K=2	5.43
6	β=0.5, α=0.011, K=2	5.57
6	β=0.6, α=0.011, K=3	5.57
8	β=0.5, α=0.011, K=3	5.71

Table B.25
Friedman average ranks: GD^+.

Rank	Algorithm	Avg. Friedman Rank
1	SMS-EMOA	1.43
2	β=0.5, α=0.011, K=3	3.57
3	β=0.6, α=0.011, K=3	3.71
4	NSGA-II	4.29
4	RVEA	4.29
6	β=0.5, α=0.011, K=2	5.86
7	β=0.6, α=0.011, K=2	6.43
7	β=1.0, α=0.009, K=2	6.43

Tables B.22–B.25 report Friedman's average ranks at $t = 199$ across 7 problems for the 8 algorithms (5 MOEDA champions + 3 baselines).

SMS-EMOA ranks first for HV. Two MOEDA champions (K=3 variants) rank second and third, ahead of NSGA-II (rank 3.57). RVEA ranks last (8.00).

SMS-EMOA ranks first on all four indicators. For HV and GD^+, MOEDA champions rank immediately after SMS-EMOA and ahead of NSGA-II. For Δ and UDE, NSGA-II ranks second, with MOEDA champions in positions 4–8. These rankings depend on the aggregation method. Friedman assigns within-problem ranks before averaging, weighting each problem equally regardless of indicator scale. When the same 8 algorithms are ranked by LMM marginal means (which average raw indicator values), NSGA-II and SMS-EMOA occupy the top two HV positions because both achieve near-optimal HV on the three ZDT benchmarks, whose larger absolute values dominate the cross-problem mean. On the structural problems, where absolute HV values are smaller, MOEDA performs comparably. Friedman's rank-based aggregation treats each problem as an equally weighted block, making it appropriate when problem-level equity is desired.

Figs. B.35–B.38 display the Critical Difference diagrams comparing the selected MOEDA champions against the baselines through a Nemenyi post hoc test.

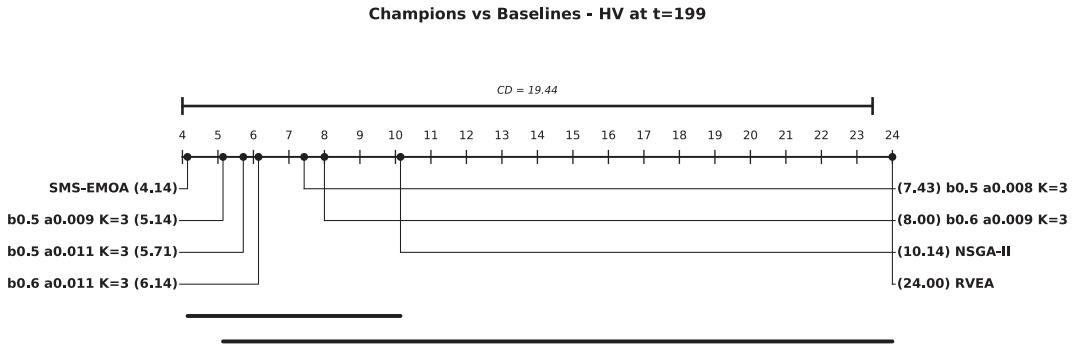


Fig. B.35. Critical Difference diagram for HV at $t = 199$ (Nemenyi post-hoc, $\alpha = 0.05$). Five MOEDA champions plus three baselines. Thick horizontal bars connect algorithms whose rank difference does not exceed the CD threshold.

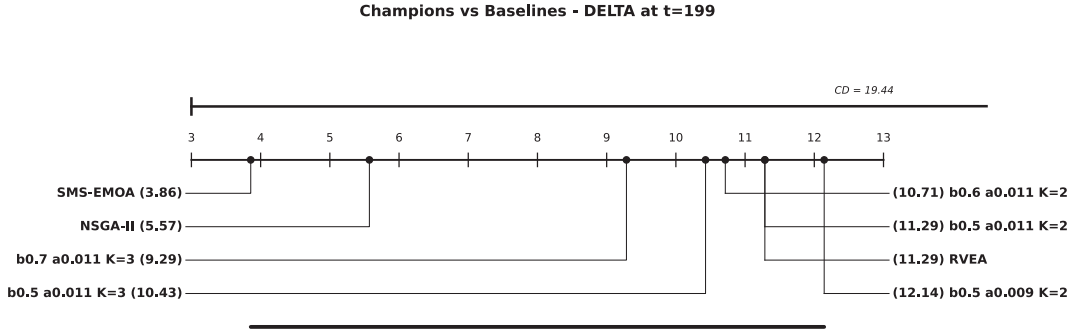


Fig. B.36. Critical Difference diagram for Δ (spread) at $t = 199$ (Nemenyi post-hoc, $\alpha = 0.05$). Five MOEDA champions plus three baselines.

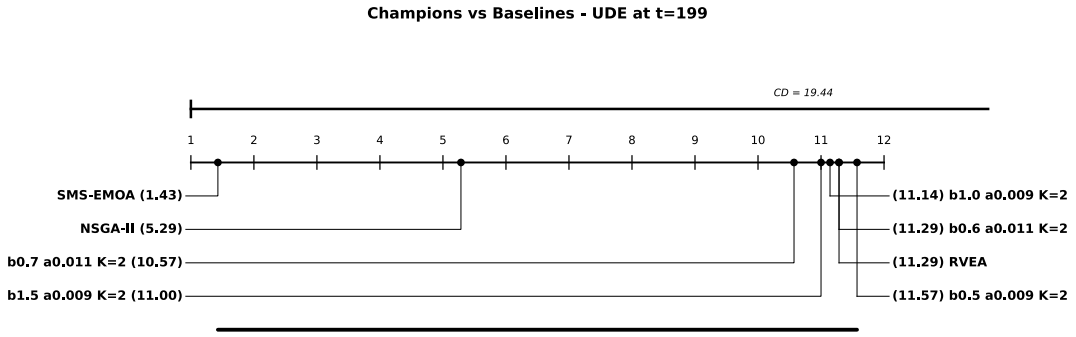


Fig. B.37. Critical Difference diagram for UDE at $t = 199$ (Nemenyi post-hoc, $\alpha = 0.05$). Five MOEDA champions plus three baselines.

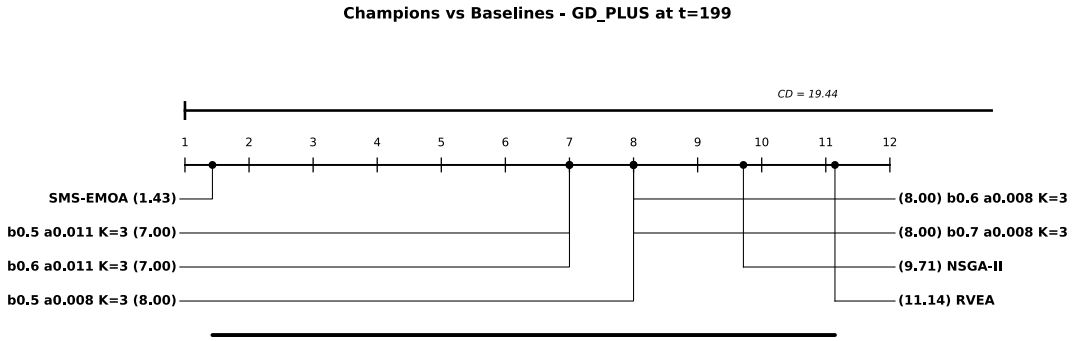


Fig. B.38. Critical Difference diagram for GD^+ at $t = 199$ (Nemenyi post-hoc, $\alpha = 0.05$). Five MOEDA champions plus three baselines.

Appendix C. Marginal sensitivity distributions

We assess sensitivity to the three MOEDA-GMM-specific hyperparameters: α (initial covariance lower bound), β (covariance decay exponent), and the number of Gaussian components K in the mixture model. All 210 configurations from the factorial grid (Section 3.1) are evaluated; the figures below report marginal performance averaged over the other two parameters. Reported values are mean±std across 50 runs.
Figs. C.39–C.42 present marginal distributions as violin plots at $t = 199$, marginalizing over the other two parameters. Red triangles mark grid-edge values.

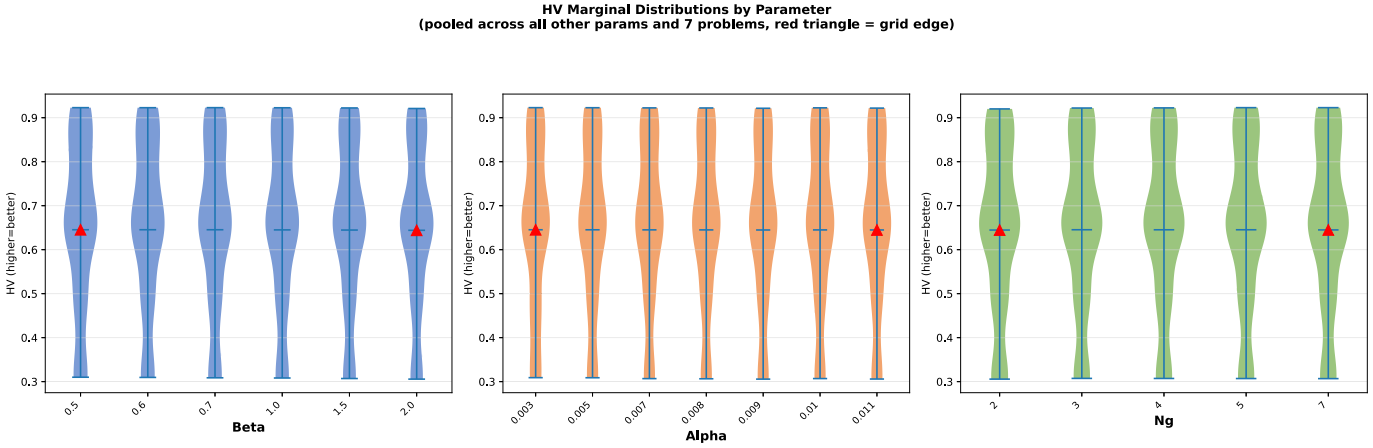


Fig. C.39. Marginal HV distributions by parameter level.

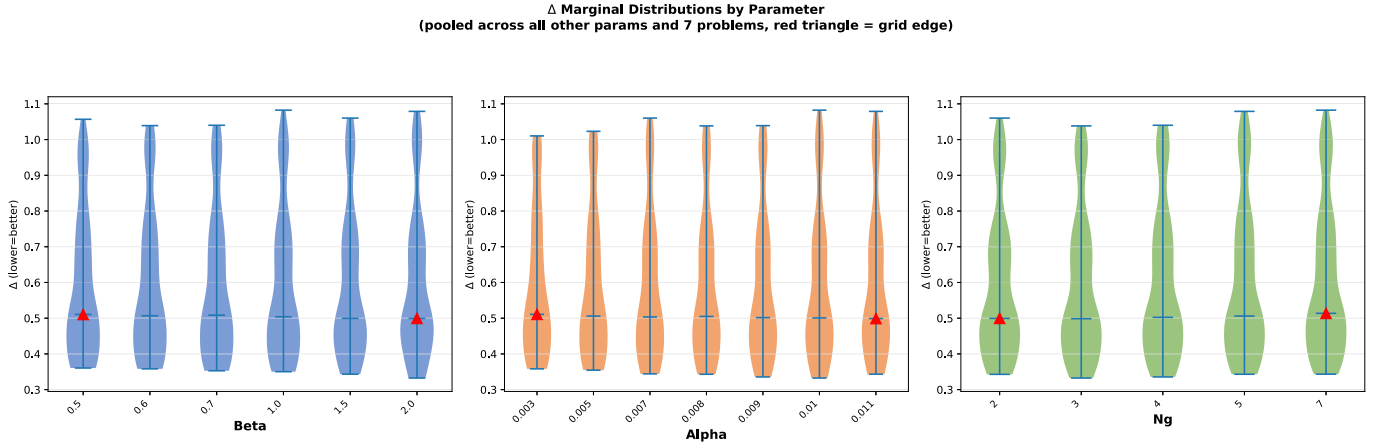


Fig. C.40. Marginal Δ distributions by parameter level.

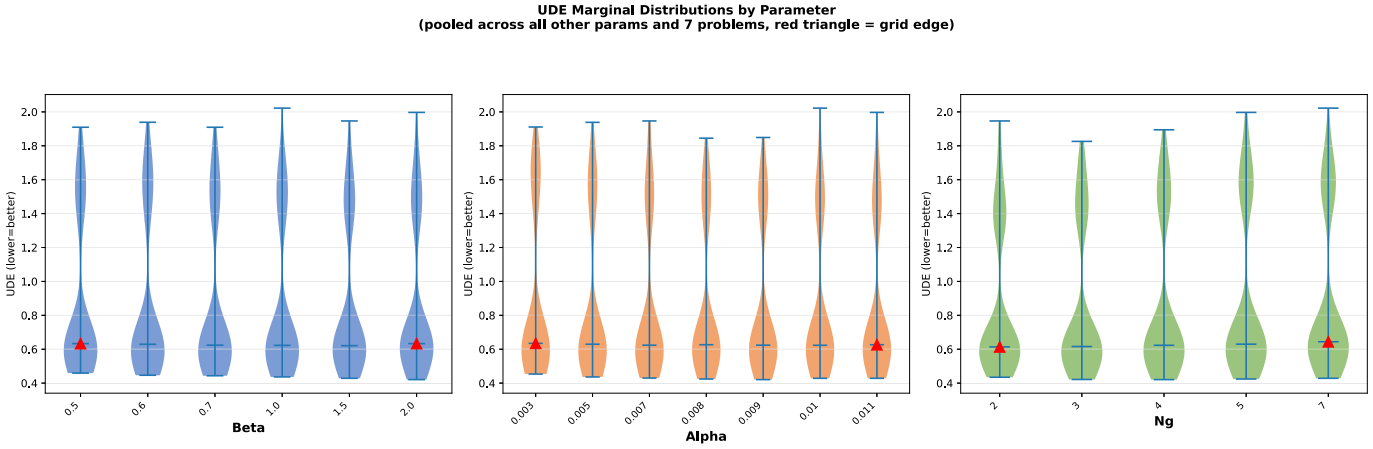


Fig. C.41. Marginal UDE distributions by parameter level.

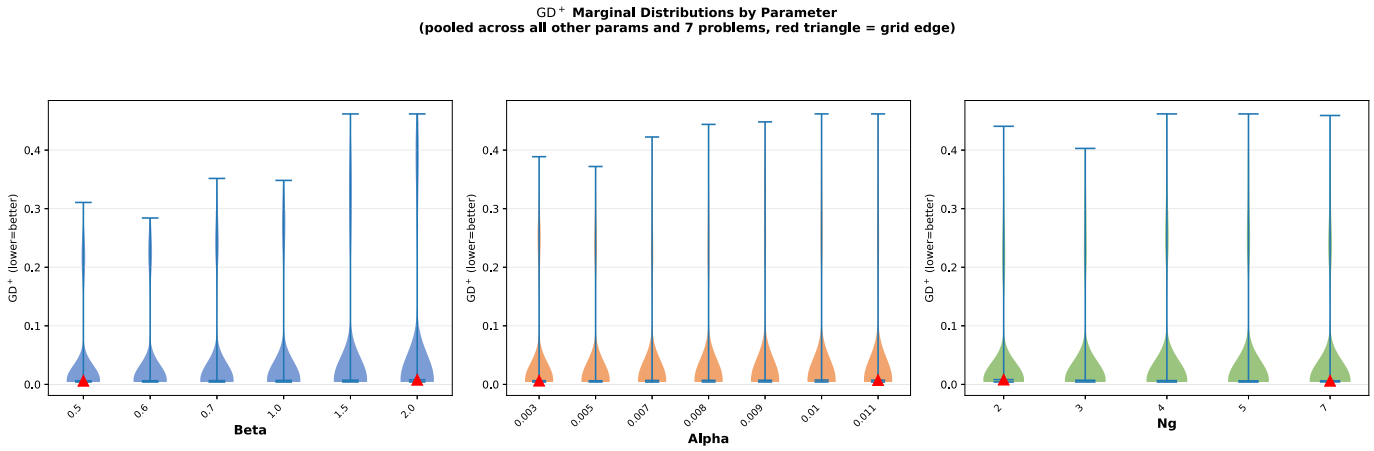


Fig. C.42. Marginal GD⁺ distributions by parameter level.

Data availability

No data was used for the research described in the article.

References

[1] United Nations Environment Programme, Global status report for buildings and construction 2024/2025, Tech. rep.Accessed:, United Nations Environment Programme, 2025, https://www.unep.org/resources/report/global-status-report-buildings-and-construction-20242025 (2 March 2026).

[2] N.D. Lagaros, The environmental and economic impact of structural optimization, Struct. Multidiscip. Optim. 58 (4) (2018) 1751–1768, https://doi.org/10.1007/s00158-018-1998-z

[3] E. Commission, European commission: internal market, industry, entrepreneurship and SMEs, 2024. Accessed:, https://single-market-economy.ec.europa.eu/sectors/construction_en (15 April 2024).

[4] R.W. Clough, The finite element method in plane stress analysis, in: Proceedings of the 2nd ASCE Conference on Electronic Computation, 1960, pp. 345–378.

[5] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, Elsevier, 2005.

[6] R.E. International, STAAD: a structural analysis and design program for AEC applications, 1981.

[7] C. Chapman, Finite element analysis in AutoCAD: a new era for CAD engineers, Eng. Comput. 10 (1) (1994) 45–56.

[8] J. Melchiorre, A. Manuello Bertetto, S. Invernizzi, G.C. Marano, F. Bazzucchi, Dakar mosque gridshell: exploring the benefits of the improved multibody rope approach through postbuckling analysis, Structures 61 (2024) 106059, https://doi.org/10.1016/j.istruc.2024.106059, https://www.sciencedirect.com/science/article/pii/S235201242400211X.

[9] J. Melchiorre, A. Manuello, F. Marmo, S. Adriaenssens, G.C. Marano, Differential formulation and numerical solution for elastic arches with variable curvature and tapered cross-sections, European Journal of Mechanics - A/Solids 97 (2023) 104757.

[10] J. Melchiorre, S. Soutiropoulos, A. Manuello Bertetto, G.C. Marano, F. Marmo, Grid-shell multi-step structural optimization with improved multi-body rope approach and multi-objective genetic algorithm, in: S. Gabriele, A. Manuello Bertetto, F. Marmo, A. Micheletti (Eds.), Shell and Spatial Structures, Springer Nature Switzerland, Cham, 2024, pp. 62–72.

[11] A.M. Bertetto, J. Melchiorre, G.C. Marano, Improved multi-body rope approach for free-form gridshell structures using equal-length element strategy, Autom. Constr. 161 (2024) 105340, https://doi.org/10.1016/j.autcon.2024.105340, https://www.sciencedirect.com/science/article/pii/S0926580524000761.

[12] J. Melchiorre, A. Manuello Bertetto, G.C. Marano, S. Adriaenssens, Human-in-the-loop structural optimization: a paradigm shift in structural and architectural design, in: International Association for Shell and Spatial Structures (IASS), 2024.

[13] J. Melchiorre, A. Manuello Bertetto, G.C. Marano, S. Adriaenssens, AI-driven human-in-the-loop structural optimization of gridshells, Comput.-aided Civ. Infrastruct. Eng., 2026.

[14] J. Melchiorre, L. D'Amato, F. Agostini, A.M. Rizzo, Acoustic emission onset time detection for structural monitoring with U-Net neural network architecture, Dev. Built Environ. 18 (2024) 100449.

[15] M.M. Rosso, A. Aloisio, J. Parol, G.C. Marano, G. Quaranta, Intelligent automatic operational modal analysis, Mech. Syst. Signal Process. 201 (2023) 110669.

[16] G.C. Marano, M.M. Rosso, J. Melchiorre, Optimization as a tool for seismic protection of structures, in: Proc. World Conference on Seismic Isolation, 2022, pp. 100–113.

[17] J. Nocedal, S.J. Wright, Numerical Optimization, Springer New York, New York, NY, 1999.

[18] A. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, C. R. Acad. Sci. Paris 25 (1847) 536–538.

[19] H. Robbins, S. Monro, A stochastic approximation method, The Annals of Mathematical Statistics 22 (3) (1951) 400.

[20] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (2) (1964) 149–154.

[21] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, 1963.

[22] R. Hooke, T.A. Jeeves, "Direct search" solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229, https://doi.org/10.1145/321062.321069

[23] V. Torczon, On the convergence of pattern search algorithms, SIAM J. Optim. 7 (1) (1997) 1–25.

[24] A.R. Kashani, C.V. Camp, M. Rostamian, K. Azizi, A.H. Gandomi, Population-based optimization in structural engineering: a review, Artif. Intell. Rev. (2022) 1–108.

[25] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction, in: Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, Springer London, 2011, pp. 3–34.

[26] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[27] M.M. Rosso, R. Cucuzza, A. Aloisio, G.C. Marano, Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator, Appl. Sci. 12 (5) (2022) 2285.

[28] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (1988) 95–99, https://doi.org/10.1023/A:1022602019183

[29] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.

[30] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: Proc. Int. Conf. on Genetic Algorithms (ICGA), 93, 1993, pp. 416–423.

[31] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[32] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669.

[33] M. Tanaka, H. Watanabe, Y. Furukawa, T. Tanino, GA-based decision support system for multicriteria optimization, in: Proc. 1995 IEEE Int. Conf. Syst, Man, Cybern.: Intelligent Systems for the 21st Century, 2, 1995, pp. 1556–1561.

[34] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, vol. 2, Springer Science & Business Media, 2001.

[35] S. Gao, C.W. de Silva, Estimation distribution algorithms on constrained optimization problems, Appl. Math. Comput. 339 (2018) 323–345.

[36] M.M. Rosso, J. Melchiorre, R. Cucuzza, A. Manuello, G.C. Marano, Estimation of distribution algorithm for constrained optimization in structural design, in: Pursuing the Infinite Potential of Computational Mechanics, International Centre for Numerical Methods in Engineering, CIMNE, 2022.

[37] M. Pelikan, H. Mühlenbein, The bivariate marginal distribution algorithm, in: Advances in Soft Computing: Engineering Design and Manufacturing, Springer, 1999, pp. 521–535.

[38] M. Pelikan, M. Pelikan, Hierarchical Bayesian Optimization Algorithm, Springer, 2005.

[39] D.A. Reynolds, Gaussian mixture models, Encyclopedia of Biometrics 741 (2009) 659–663.

[40] J. Tian, X. Hao, T. Murata, Markov network based multi-objective EDA and its application for resource constrained project scheduling, IEEJ Transactions on Electronics, Information and Systems 136 (3) (2016) 290–298.

[41] K. Yang, L. Mu, D. Yang, F. Zou, L. Wang, Q. Jiang, Multiobjective memetic estimation of distribution algorithm based on an incremental tournament local searcher, The Scientific World Journal 2014 (1) (2014) 836272.

[42] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: J. Neyman (Ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1951, pp. 481–492.

[43] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63, https://doi.org/10.1109/TEVC.2007.894202
[44] Y. Lu, X. Guo, J. Geng, S. Wang, Z. Wang, A. Zhou, J. Sun, X. Si, X. Sun, H. Zhang, Regularity-based evolutionary multi-objective optimization review, Swarm Evol. Comput. 97 (2025) 101999, https://doi.org/10.1016/j.swevo.2025.101999
[45] Q. Jiang, J. Cui, L. Wang, Y. Lin, Y. Wu, X. Hei, A regularity model-based multiobjective estimation of distribution memetic algorithm with auto-controllable population diversity, Memetic Comput. 15 (2023) 45–70, https://doi.org/10.1007/s12293-023-00387-y
[46] S. Wang, A. Zhou, Regularity evolution for multiobjective optimization, IEEE Trans. Evol. Comput. 28 (5) (2024) 1470–1483, https://doi.org/10.1109/TEVC.2023.3306523
[47] L. Pan, L. Li, R. Cheng, C. He, K.C. Tan, Manifold learning-inspired mating restriction for evolutionary multiobjective optimization with complicated Pareto sets, IEEE Trans. Cybern. 51 (6) (2021) 3325–3337, https://doi.org/10.1109/TCYB.2019.2952881
[48] X. Gao, H. Zhang, S. Song, A regularity property-driven evolutionary algorithm for multiobjective optimization, Swarm Evol. Comput. 78 (2023) 101258, https://doi.org/10.1016/j.swevo.2023.101258
[49] H. Xie, M. Zhang, Parent selection pressure auto-tuning for tournament selection in genetic programming, IEEE Trans. Evol. Comput. 17 (1) (2012) 1–19.
[50] T.K. Moon, The expectation-maximization algorithm, IEEE Signal Process. Mag. 13 (6) (1996) 47–60.
[51] P.A.N. Bosman, D. Thierens, Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms, Int. J. Approx. Reason. 31 (2002) 259–289, https://doi.org/10.1016/S0888-613X(02)00090-7
[52] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (11–12) (2002) 1245–1287.
[53] A. Shirazi, J. Ceberio, J.A. Lozano, EDA + +: estimation of distribution algorithms with feasibility conserving mechanisms for constrained continuous optimization, IEEE Trans. Evol. Comput. 26 (5) (2022) 1144–1156.
[54] N. Mashru, K. Kalita, L. Čepová, P. Patel, Arpita, P. Jangir, Adaptive predator prey algorithm for many objective optimization, Sci. Rep. 15 (1) (2025) 12690.
[55] P. Patel, D. Adalja, N. Mashru, P. Jangir, Arpita, R. Jangid, Arpita, M. Khishe, Multi objective elk herd optimization for efficient structural design, Sci. Rep. 15 (1) (2025) 11767.
[56] D. Adalja, P. Patel, N. Mashru, P. Jangir, Arpita, R. Jangid, G. Gulothungan, M. Khishe, A new multi objective crested porcupines optimization algorithm for solving optimization problems, Sci. Rep. 15 (1) (2025) 14380.
[57] D. Adalja, K. Kalita, L. Čepová, P. Patel, N. Mashru, P. Jangir, et al., Advancing truss structure optimization–a multi-objective weighted average algorithm with enhanced convergence and diversity, Results Eng. 25 (2025) 104241.
[58] N. Mashru, G.G. Tejani, P. Patel, Many-objective optimization of a 120-bar 3D dome truss structure using three metaheuristics, in: International Conference on Advanced Engineering Optimization Through Intelligent Techniques, Springer, 2023, pp. 231–239.
[59] N. Mashru, P. Patel, G.G. Tejani, A. Kaneria, Multi-objective thermal exchange optimization for truss structure, in: Advanced Engineering Optimization Through Intelligent Techniques: Select Proceedings of AEOTIT 2022, Springer, 2023, pp. 139–146.
[60] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[61] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[62] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, J. Am. Stat. Assoc. 47 (260) (1952) 583–621.
[63] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, The annals of mathematical statistics (1947) 50–60.
[64] D. Bates, M. Mächler, B. Bolker, S. Walker, Fitting linear mixed-effects models using lme4, J. Stat. Softw. 67 (2015) 1–48.
[65] J.O. Wobbrock, L. Findlater, D. Gergle, J.J. Higgins, The aligned rank transform for nonparametric factorial analyses using only anova procedures, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2011, pp. 143–146.
[66] J. Blank, K. Deb, pymoo: multi-objective optimization in Python, IEEE Access 8 (2020) 89497–89509.
[67] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point, in: Proc. 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA), ACM, 2009, pp. 87–102.
[68] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Modified distance calculation in generational distance and inverted generational distance, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 110–125.
[69] R. D'Agostino, E.S. Pearson, Tests for departure from normality. Empirical results for the distributions of b 2 and $\sqrt{b}$, Biometrika 60 (3) (1973) 613–622.
[70] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (3–4) (1965) 591–611.
[71] P.B. Nemenyi, Distribution-Free Multiple Comparisons, Princeton University, 1963.
[72] C.A. Coello, G.T. Pulido, Multiobjective structural optimization using a microgenetic algorithm, Struct. Multidiscip. Optim. 30 (2005) 388–403.
[73] A.A. El-Sawy, M.A. Hussein, E.S.M. Zaki, A.M.A. Allah, Local search-inspired rough sets for improving multiobjective evolutionary algorithm, Appl. Math. 2014 (2014).
[74] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[75] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (200) (1937) 675–701.