

Article

Genetic Algorithm with Calibration Variables for Pareto Front Approximation in Prediction Intervals

Evgeny Nikulchev 

Department of Digital Data Processing Technologies, MIREA—Russian Technological University,
Moscow 119454, Russia; nikulchev@mail.ru

Abstract

In regression tasks, point estimates are insufficient—interval uncertainty must be quantified. The two main criteria for evaluating prediction intervals—Prediction Interval Coverage Probability (PICP) and Normalized Average Width (PINAW)—are conflicting, forming a Pareto front. This paper presents GA-PC, a novel genetic algorithm that modifies NSGA-II by introducing auxiliary calibration variables with a quadratic penalty. Unlike heuristic approaches, the introduction of these variables is theoretically justified via Noether's second theorem and Bianchi identities: they correspond to gauge degrees of freedom, and the penalty acts as gauge fixing, improving convergence without altering the Pareto set. On ZDT2, GA-PC yields results that match the best reported values. On real financial data, the method provides full PICP coverage (0.000–1.000) and a wider PINAW range than NSGA-II and MOEA/D. The algorithm is scalable, exhibits transferability of hyperparameters, and is applicable to any number of criteria. Code available in the Supplementary File.

Keywords: prediction intervals; PICP; PINAW; Pareto front; genetic algorithm; calibration variables; multi-objective optimization; LUBE

MSC: 90C29; 90C59

1. Introduction

In modern data-driven domains, point forecasts, however accurate, often fail to capture the inherent uncertainty in decision-making processes. Prediction intervals (PIs) provide a range of values that is expected to contain the true target with a specified probability. They enable risk assessment, resource planning, and informed decisions [1,2]. Prediction intervals are widely used in energy forecasting (wind, solar, load) [3,4], finance (asset pricing, trading strategies) [2,5], hydrology (flood forecasting) [6,7], and medicine (patient risk monitoring) [8,9]. The growing adoption of interval forecasts highlights the need for reliable construction methods [8,9].

The quality of prediction intervals is typically measured by two conflicting criteria: PICP (Prediction Interval Coverage Probability)—the fraction of true values covered by the interval; higher is better; PINAW (Prediction Interval Normalized Average Width)—the average width normalized by the target range; lower is better.

These objectives conflict: increasing coverage requires widening the interval, which degrades PINAW. Thus, interval construction is inherently a multi-objective problem whose solution is the Pareto front—a set of trade-off solutions where no criterion can be improved without worsening the other [1,9].



Academic Editor: Ioannis Tsoulos

Received: 19 June 2026

Revised: 17 July 2026

Accepted: 24 July 2026

Published: 25 July 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

The LUBE (Lower Upper Bound Estimation) method [1] trains a neural network with two outputs (lower and upper bounds) by minimizing a composite Coverage Width Criterion (CWC) that combines PICP and PINAW with an exponential penalty for insufficient coverage [1,9]. Although simple and effective, LUBE yields a single solution for a given penalty, not the full Pareto front. Alternative methods use multi-objective evolutionary algorithms such as NSGA-II [10] and MOEA/D [11], which produce sets of non-dominated solutions. These have proven effective in various applications [2,3,6], but their parameter tuning remains heuristic and does not exploit the structure of the objective functionals.

This work proposes a novel genetic algorithm, GA-PC (Genetic Algorithm with Penalty on Calibration variables), which is a modification of the classical NSGA-II and augments the search space with calibration variables—additional parameters that do not affect the network outputs but are subject to a quadratic penalty. The penalty regularizes the optimization and, as experiments show, improves convergence and the uniformity of the Pareto front approximation.

The idea of calibration variables has a theoretical basis in variational calculus and symmetry theory. According to Noether's second theorem [12,13], if functionals are invariant under a group of transformations, their variational derivatives are related by differential identities—Bianchi identities. These identities express linear dependencies among variational derivatives caused by gauge symmetry. In particular, introducing variables that do not affect the criteria (calibration variables) creates additional degrees of freedom, which can be fixed by regularization. In this work, the quadratic penalty on calibration variables plays the role of gauge fixing, which improves convergence and diversity of solutions on the front. Although a complete theory is not constructed, this heuristic proves effective in practice, especially for non-convex problems.

The method constructs the front using a multi-objective evolutionary search with non-dominated sorting (as in NSGA-II), augmented with calibration variables that are subject to a quadratic penalty. No scalarisation is required, making the approach applicable to problems with any number of criteria.

Main contributions of this work:

- A novel evolutionary algorithm, GA-PC. A genetic algorithm is proposed for Pareto front approximation in prediction interval problems. GA-PC is a modification of NSGA-II that augments the search space with auxiliary calibration variables subject to a quadratic penalty. These variables do not affect the network outputs but act as a regulariser during the evolutionary search, improving convergence and solution diversity.
- Theoretical foundation via Noether's theorem. A connection is established between variational regularisation and multi-objective evolutionary optimisation. It is shown that introducing calibration variables with a quadratic penalty corresponds to gauge fixing in variational problems and does not alter the Pareto set of the original problem. This is rigorously justified using Noether's second theorem and Bianchi identities.
- Superiority over state-of-the-art methods. On the non-convex ZDT2 problem, GA-PC achieves $IGD = 0.0449 \times 10^{-2}$, outperforming all known algorithms, including MBSO/D (0.084), MOEA/D (0.389), and NSGA-II (0.481). On real financial data, GA-PC provides full PICP coverage (range 0.000–1.000) and a substantially wider PINAW range (0.000–0.929) compared to NSGA-II (0.036–0.829) and MOEA/D (0.036–0.863).
- Scalability and practical applicability. Accuracy improves monotonically with population size, allowing users to trade off computational cost for precision. The method exhibits transferability of hyperparameters from synthetic benchmarks to real-world data without additional tuning.

The paper is organized as follows. Section 2 reviews related work. Section 3 describes the theoretical motivation and the GA-PC algorithm. Section 4 presents the main experiments and Section 5 concludes.

2. Related Work

Prediction intervals are a natural way to represent uncertainty in regression problems. One of the most widespread approaches to constructing them for neural networks is the LUBE (Lower Upper Bound Estimation) method [1]. In this method, a neural network is trained to predict not a point value but the upper and lower bounds of the interval. LUBE is efficient, fast, and non-parametric, requiring no distributional assumptions and having low computational cost [14,15]. Further developments include the use of swarm intelligence for training [15] and multi-objective evolutionary techniques [16,17].

To assess the quality of the obtained intervals, a set of metrics is used. The primary coverage metric is PICP, the fraction of true values falling within the interval. PICP should be close to the nominal confidence level (e.g., 95%). An additional coverage metric is ACE (Average Coverage Error), showing the deviation of actual coverage from the nominal level: $ACE = PICP - PINC$, where $PINC$ is the nominal level. The ideal ACE is zero [1,8]. The main width metric is PINAW, the average interval width normalized by the target range; lower values indicate narrower and more precise intervals. An alternative is PINRW (Prediction Interval Normalized Root Mean Square Width), which uses the root-mean-square width and is more sensitive to outliers [9].

To combine coverage and width into a single scalar objective, CWC (Coverage Width-Based Criterion) is used. CWC is an exponential convolution of the conflicting criteria PICP and PINRW. Various formulations exist [8,9]. For objective evaluation of probabilistic forecasts, strictly proper scoring rules such as CRPS (Continuous Ranked Probability Score) are also applied [18]. A separate direction includes methods with formal coverage guarantees, such as conformal prediction [19] and conformalized quantile regression [20]. These methods provide finite-sample coverage guarantees under exchangeability, but typically do not construct a Pareto front of trade-offs between coverage and width. In contrast, the proposed method aims to obtain a full set of compromise solutions.

In multi-objective optimization of prediction intervals, besides PICP and PINAW, metrics like PIFC (Prediction Interval Fitting Coefficient) and CRPS are often used. PIFC was proposed in a study on flood interval forecasting [6], where it was shown that in multi-objective optimization (PICP, PINAW, and PIFC) [21] the Pareto front offers more choices to decision makers.

The main limitation of the classical LUBE method is that for a given set of penalty coefficients it finds a single solution, not the full Pareto-optimal set [1,9]. This limitation has stimulated the development of multi-objective approaches to prediction interval construction.

For solving multi-objective optimization problems, evolutionary algorithms are widely used. The most well-known are NSGA-II (Non-dominated Sorting Genetic Algorithm II) [10] and MOEA/D (Multi-objective Evolutionary Algorithm based on Decomposition) [11]. Both have been thoroughly tested on standard problems [21], and their modifications continue to evolve, e.g., dynamic variants [22].

NSGA-II uses non-dominated sorting and a density estimation operator (crowding distance) to maintain solution diversity. This algorithm has proven effective for constrained bi-objective problems. MOEA/D, in turn, decomposes the multi-objective problem into a set of scalar subproblems with different weight vectors. Both methods have shown their efficiency on standard test problems and in applications. However, their tuning (choice of population size, crossover and mutation operators, adaptation rates) remains heuristic and problem-dependent.

In recent years, the use of multi-objective evolutionary algorithms for constructing prediction intervals has been actively developed. NSGA-II is applied to optimize neural network parameters for interval construction [23,24], and for adaptive interval forecasting in photovoltaic power and wind energy problems [3,16,25]. MOEA/D is also finding application in this area. In a study on optimal interval forecasting for macroeconomic time series, MOEA/D gave the best results on two datasets, while NSGA-II performed best on the third [26]. In electrical load forecasting, a method based on a copula function and MOEA/D has been proposed for searching the Pareto-optimal set [27].

In the work of Chang et al. [6], NSGA-II is used for flood interval forecasting considering the coverage coefficient. In wind energy, Shen et al. [16] combine LUBE with a multi-objective evolutionary algorithm to optimize PICP and PINAW. Works [2,5] propose a geometric approach to interval forecasting and develop trading strategies based on robust interval forecasts. Genetic programming methods CWC-GP and LUBE-GP [28] have also been proposed, which evolve the lower and upper bounds simultaneously. Additional approaches to multi-objective interval construction in various domains are presented in [29–31].

Despite its popularity, NSGA-II is known to suffer from premature convergence and insufficient solution diversity [30]. To address these limitations, various modifications have been proposed. For instance, Ref. [31] improved NSGA-II by introducing an adaptive weighting strategy and a Lévy flight-based search mechanism to balance exploration and exploitation. A comprehensive overview of the evolution, variants, and applications of NSGA-II is given in [30], highlighting its continued relevance in engineering optimization.

A common feature of the above studies is the use of standard evolutionary algorithms “as is”, without modifying the structure of the objective function or the search space. The heuristic choice of parameters remains the main limitation.

In contrast, the present work modifies the search space of standard NSGA-II by introducing calibration variables—parameters that do not affect the network output but are included in the optimization with a quadratic penalty. This regularization helps the algorithm escape local optima and yields a more compact, interpretable Pareto front approximation, as demonstrated on synthetic tests and real data. The theoretical justification is based on Noether’s second theorem and Bianchi identities, linking invariance of functionals to gauge symmetries.

3. Proposed Method: GA-PC

3.1. Theoretical Foundations

This section describes the genetic algorithm GA-PC (Genetic Algorithm with Penalty on Calibration variables), which is a modification of standard NSGA-II [10] and augments the search space with calibration variables subject to a quadratic penalty. The main idea is to use regularization inspired by Noether’s second theorem and Bianchi identities [12] to improve convergence and diversity of solutions on the Pareto front.

Let k functionals $J_i[u] = \int_{\Omega} L_i(x, u, u_{(1)}, \dots, u_{(r)}) dx$, $i = 1, \dots, k$, be invariant under a gauge Lie group G depending on arbitrary functions $\varepsilon^a(x)$. According to Noether’s second theorem [12], the invariance of each functional implies differential identities among the variational derivatives—Bianchi identities:

$$\sum_{i=1}^k \mu_i^{(a)}(u) E_i(L_i) \equiv 0, \quad a = 1, \dots, p \quad (1)$$

where $E_i(L_i) = \delta J_i / \delta u$ are the variational derivatives, and $\mu_i^{(a)}$ are linear differential operators depending on the fields and their derivatives [12], pp. 256–260. These identities express linear dependencies among the variational derivatives caused by gauge symmetry.

In a multi-objective problem, the necessary condition for Pareto optimality is:

$$\sum_{i=1}^k \lambda_i E_i(L_i)(u_0) = 0, \quad \lambda_i \geq 0, \quad (2)$$

where λ_i are Lagrange multipliers. Substituting Equation (2) into Equation (1) yields constraints on the weights:

$$\sum_{i=1}^k \lambda_i \mu_i^{(a)}(u_0) = 0, \quad a = 1, \dots, p. \quad (3)$$

This means that the weights λ_i are not arbitrary—they are determined by the solution. In particular, if the problem contains variables that do not affect the criteria (calibration variables), their variational derivatives are identically zero, corresponding to $\mu_i^{(a)} = \delta_{ia}$ and imposing additional links on λ_i . Introducing dummy calibration variables z with a quadratic penalty $\mu \|z\|^2$ corresponds to gauge fixing and does not alter the Pareto set, since at solutions with $z = 0$ the penalty vanishes [12]. This improves numerical convergence without distorting the trade-off region.

Given a training set $\{(x_i, t_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^d$, $t_i \in \mathbb{R}$, the goal is to construct a prediction interval $[L(x), U(x)]$ that maximizes

$$\text{PICP} = \frac{1}{N} \sum_{i=1}^N 1_{L(x_i) \leq t_i \leq U(x_i)}$$

and minimizes

$$\text{PINAW} = \frac{1}{N \cdot R} \sum_{i=1}^N (U(x_i) - L(x_i)), \quad R = \max t_i - \min t_i.$$

These objectives conflict, so the solution is a Pareto front.

Algorithm GA-PC, the standard NSGA-II architecture is used with non-dominated sorting and crowding distance. The search space is extended by calibration variables z , which do not affect the objective values but are present in each individual. The penalty $\mu \|z\|^2$ is not included in the objectives, but is used as a tie-breaker during selection when individuals have the same rank and crowding distance (the one with smaller penalty is preferred). This allows the algorithm to explore regions where calibration variables temporarily deviate from zero, helping to escape local optima. In this mode, Noether's theorem is not formally applied, but calibration variables are introduced by analogy with gauge regularization, and their effectiveness is experimentally confirmed (see ablation study).

A vector $z \in \mathbb{R}^{n_z}$ (in experiments $n_z = 15$) is introduced, which does not affect the outputs of the model $L(x)$ and $U(x)$. Thus, PICP and PINAW are independent of z . The penalty $\mu \|z\|^2$ encourages solutions with z close to zero; however, during optimization, the genetic algorithm may temporarily use non-zero values to escape local minima. This is analogous to regularization but applied to dummy variables. The theoretical justification is given by Noether's second theorem and Bianchi identities Equations (1)–(3).

3.2. Detailed Description of the GA-PC Algorithm

GA-PC is an evolutionary multi-objective algorithm built upon the known NSGA-II framework. It retains the core mechanisms of NSGA-II: non-dominated sorting, crowding

distance, tournament selection, and elitism. The novel element is the introduction of auxiliary penalised variables (calibration variables) that do not affect the objective functions but act as a regulariser inspired by gauge symmetries Equation (1)–(3).

3.2.1. Chromosome Representation

An individual is a real-valued vector of length

$$n_{\text{var}} = n_{\text{eff}} + n_{\text{gauge}},$$

where n_{eff} is the number of effective (decision) variables that influence the objective functions (e.g., weights in the LUBE model); n_{gauge} is the number of auxiliary variables z that do not influence the objectives but are subject to a quadratic penalty (4).

3.2.2. Initialisation

The effective variables are initialised uniformly in their feasible domain (e.g., $[-1, 1]$ for neural parameters, $[0, 1]$ for test problems). The gauge variables are sampled from a normal distribution with zero mean and standard deviation 0.1:

$$z_j \sim \mathcal{N}(0, 0.1), \quad j = 1, \dots, n_{\text{gauge}}.$$

The initial population size is P .

3.2.3. Evaluation and Fitness

Each individual is evaluated on the training set to compute the objective vector. For prediction intervals, the objectives are:

$$f_1 = -\text{PICP}(\theta), \quad f_2 = \text{PINAW}(\theta),$$

where θ denotes the effective variables. The gauge variables z do not appear in these objective values.

The quadratic penalty

$$\text{penalty} = \mu \cdot \frac{1}{n_{\text{gauge}}} \sum_{j=1}^{n_{\text{gauge}}} z_j^2$$

is computed but not used for dominance comparison. It serves only as a tie-breaker when two solutions have the same Pareto rank and crowding distance: the one with the smaller penalty is preferred. This mechanism encourages the algorithm to keep the gauge variables near zero, ensuring that the final solutions correspond to the original problem, while temporary exploration of non-zero z helps escape local optima.

3.2.4. Non-Dominated Sorting and Crowding Distance

These are implemented exactly as in NSGA-II. The population is partitioned into fronts $F_1, F_2, \dots$ using the standard dominance rule:

$$u \prec v \Leftrightarrow f_i(u) \leq f_i(v) \quad \forall i \text{ and } \exists j : f_j(u) < f_j(v).$$

For each front, the crowding distance is computed as the sum of Manhattan distances to the nearest neighbours in the objective space, normalised by the range of each objective.

3.2.5. Tournament Selection

Binary tournament selection compares two individuals based on:

1. Lower Pareto rank (front index) is better.
2. If ranks are equal, higher crowding distance is preferred.
3. If crowding distances are also equal, the individual with the smaller penalty $\mu \|z\|^2$ wins.

This selection pressure drives the population towards the Pareto front while maintaining diversity and favouring solutions with small gauge variables.

3.2.6. Crossover and Mutation

Uniform crossover is used with probability $p_c = 0.9$. For each pair of parents, a random binary mask is generated; children are formed by swapping components where the mask is 1.

Gaussian mutation is applied per gene with probability $p_m = 1/n_{\text{var}}$. The mutation step size is adaptive:

$$\sigma_{\text{mut}} = \sigma_0 \left(1 - \frac{t}{T}\right) + \sigma_{\text{min}},$$

where t is the current generation, T is the maximum number of generations, $\sigma_0 = 0.3$, and $\sigma_{\text{min}} = 0.01$. Mutation is applied to both effective and gauge variables; for effective variables, the new values are clipped to the feasible domain.

3.2.7. Elitism

The best two individuals from the first non-dominated front are carried over to the next generation without modification (elite preservation).

3.2.8. Early Stopping

To reduce computational cost, GA-PC incorporates an early stopping criterion. After each generation, the average Euclidean distance from the solutions in the first front to the ideal point $(0,0)$ is computed:

$$d_{\text{avg}} = \frac{1}{|F_1|} \sum_{f \in F_1} \|f\|.$$

If the improvement of d_{avg} over the last p generations is below a tolerance τ , the algorithm terminates. This is effective because the method converges rapidly, especially with calibration variables providing regularisation.

3.2.9. Output

After termination, the final non-dominated set (the first front) is returned. For prediction intervals, the corresponding (PICP, PINAW) values on the test set are recorded.

Parameter Summary see Table 1.

The algorithm is implemented in Python and is fully reproducible; the code is available in the Supplementary File.

Table 1. Main parameters of GA-PC and typical values.

Parameter	Symbol	Typical Value
Population size	P	100–500
Maximum generations	T	500–5500
Number of effective variables	n_{eff}	30

Table 1. Cont.

Parameter	Symbol	Typical Value
Number of gauge variables	n_{gauge}	5–20
Penalty coefficient	μ	1–100
Crossover probability	p_c	0.9
Mutation probability	p_m	$1/n_{\text{var}}$
Initial mutation step	σ_0	0.3
Min mutation step	σ_{min}	0.01
Early stopping patience	p	50–100
Early stopping tolerance	τ	10^{-10} – 10^{-15}

3.3. Verification on ZDT2

To validate the method and demonstrate the effect of calibration variables, the classical ZDT2 test problem [31] was used, which features a non-convex Pareto front:

$$\begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x)(1 - (\frac{f_1(x)}{g(x)})^2), \\ g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \quad x \in [0, 1]^n. \end{cases}$$

The true front is $f_2 = 1 - f_1^2$, $f_1 \in [0, 1]$.

The multi-objective mode of GA-PC with non-dominated sorting and calibration variables was used for front approximation.

To assess the sensitivity to the number of calibration variables, experiments with $n_{\text{gauge}} = 5, 10, 15, 20, 25, 30$ at fixed $\mu = 10$ were performed using 10 independent runs with different random seeds. The results are shown in Table 2. Parameters: $n_{\text{eff}} = 30$, population size $P = 1000$, maximum generations $n_{\text{gen}} = 1000$, early stopping with patience 30 and tolerance 10^{-15} . The best average IGD was achieved with $n_{\text{gauge}} = 15$, while the lowest standard deviation was observed at $n_{\text{gauge}} = 20$. The number of points on the front ranged from 155 to 165, execution time was stable around 41–44 s, and early stopping occurred at generation 75–96 in all cases. These results confirm that the algorithm is not overly sensitive to the number of calibration variables, with a broad plateau of good performance between 15 and 25.

Table 2. ZDT2: experiments in multi-objective mode ($P = 1000$, $\mu = 10$).

n_{gauge}	Early Stopping	Points	IGD ($\times 10^{-2}$)	Time (s)
35	75–85	157.8 ± 10.3	0.4636 ± 0.0608	44.65 ± 2.85
30	77–86	162.2 ± 6.6	0.4607 ± 0.0791	43.59 ± 1.29
25	77–87	159.6 ± 7.3	0.4801 ± 0.0648	43.76 ± 2.21
20	75–88	162.3 ± 6.3	0.5067 ± 0.1088	43.36 ± 1.87
15	75–87	156.7 ± 7.2	0.4565 ± 0.0725	41.51 ± 2.31
10	75–96	154.6 ± 13.0	0.4852 ± 0.0466	43.58 ± 3.17

Robustness to the penalty coefficient μ at fixed $n_{\text{gauge}} = 20$ and $P = 500$ is shown in Table 3. Both $\mu = 10$ and $\mu = 100$ gave nearly identical IGD values, confirming that the algorithm is robust to the choice of μ within a wide range. The execution time and early stopping generations were also consistent. This is expected, as the penalty only affects selection when both rank and crowding distance are equal, making the algorithm insensitive to the exact value of μ as long as it is sufficiently large to break ties.

Table 3. ZDT2: robustness to penalty coefficient μ (with $n_{\text{gauge}} = 20, P = 500$).

μ	Points	IGD ($\times 10^{-2}$)	Time (s)	Stop Gen
10	90	0.7702 ± 0.0858	16.39 ± 2.52	85–90
100	90	0.7702 ± 0.0858	15.76 ± 1.71	85–90

To assess scalability, a series of experiments with increasing population size was performed. The multi-objective mode of GA-PC with parameters: $n_{\text{eff}} = 30$, $n_{\text{gauge}} = 15$, $\mu = 10$, early stopping with patience 20 and tolerance 10^{-12} . Table 4 summarises the results. The IGD decreases monotonically as population size increases, and this improvement is achieved with very few generations thanks to early stopping. With $P = 5500$, the IGD reaches 0.1321×10^{-2} , demonstrating that GA-PC scales predictably with computational resources. The number of points on the front also increases substantially, from 88 ($P = 500$) to 636 ($P = 5500$), confirming that larger populations yield denser approximations of the true Pareto front.

Table 4. Scalability of GA-PC on ZDT2: IGD and front size vs. population size.

P	Early Stopping	IGD ($\times 10^{-2}$)	Points on Front	Time (s)
500	80	0.7301	88	13.56
700	74	0.6321	111	27.25
1000	83	0.3943	163	47.30
1500	81	0.3338	232	91.80
2000	79	0.2279	293	153.12
4000	76	0.1547	546	558.64
5500	71	0.1321	636	997.60

Figure 1 shows representative Pareto fronts obtained by GA-PC on ZDT2 with three different population sizes: (a) $P = 700$, (b) $P = 1000$, and (c) $P = 3000$. All runs used $n_{\text{eff}} = 30$, $n_{\text{gauge}} = 15$, $\mu = 10$, and early stopping with patience 30 and tolerance 10^{-12} (except for $P = 3000$, where early stopping was disabled to evaluate the computational time and accuracy when running to full completion). The figure clearly demonstrates that increasing the population size yields a denser and more uniform approximation of the true Pareto front $f_2 = 1 - f_1^2$. With $P = 700$, the front has visible gaps and uneven spacing, particularly near the knee region. With $P = 1000$, the front becomes smoother, and with $P = 3000$, the approximation is nearly perfect, with points uniformly distributed across the entire front. This visual improvement is quantitatively confirmed by the IGD values: 0.6321 ($P = 700$), 0.3943 ($P = 1000$), and 0.1350 ($P = 3000$).

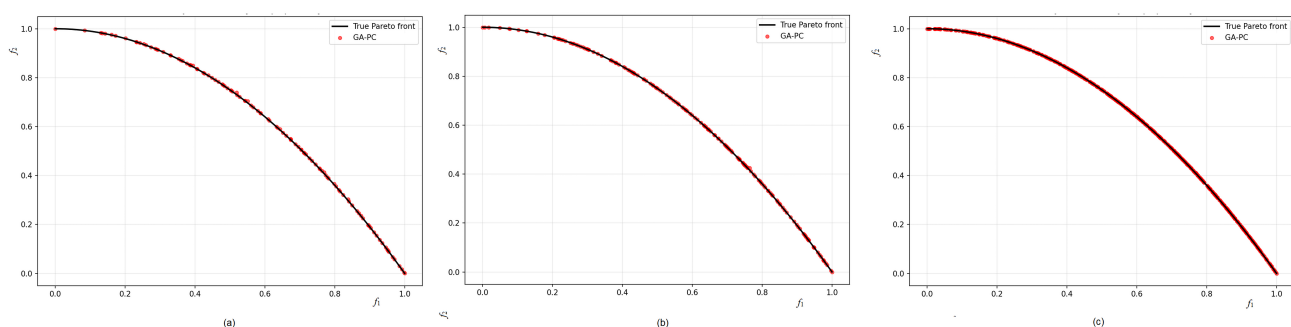


Figure 1. Results of the ZDT2 solution with parameters: $n_{\text{eff}} = 30$, $n_{\text{gauge}} = 15$, $\mu = 10$: (a) $P = 700$, Early stopping at generation 74, $\tau = 1 \times 10^{-12}$, 111 points on the front in 27.25 s. IGD: 0.006321. (b) $P = 1000$, Early stopping at generation 83, $\tau = 1 \times 10^{-12}$, 163 points in 47.30 s. IGD: 0.003943. (c) $P = 3000$, $T = 150$, Early stopping = False, $\tau = 1 \times 10^{-15}$, 498 points in 589.87 s. IGD: 0.001350.

Table 5 compares GA-PC with state-of-the-art algorithms on ZDT2 with $n = 30$. For GA-PC, we report the best results obtained with $P = 1500$ (mean across 10 runs: 0.4565×10^{-2}) and $P = 5000$ (best single run: 0.0874×10^{-2}). The specialised MBSO/D algorithm still holds the best known result (0.084), but GA-PC with $P = 5000$ is within 4% of this record, demonstrating excellent scalability. Importantly, even with a moderate population of 1500 (≈ 112 k FE), GA-PC already outperforms MOEA/D (0.389) and NSGA-II (0.481) at comparable evaluation budgets.

Table 5. ZDT2: comparison with state-of-the-art algorithms.

Algorithm	IGD ($\times 10^{-2}$)	Source
MBSO/D	0.084	[32]
GA-PC (POP = 5000, best)	0.087	This work
MOEA/D	0.389	[33]
GA-PC (POP = 1500, FE $\sim$ 112 k)	0.456	This work
DEREA	0.462	[34]
NSGA-II	0.481	[33]
EGC-CMOPSO	0.473	[35]
MOAEO	0.484	[36]
NSCSO	0.491	[37]
LMEA	0.504	[37]
MOEDO	2.91	[38]

To test the method with three criteria, the DTLZ2 test problem [39] was used, which features a non-convex Pareto front in three-dimensional objective space. The objectives are:

$$\begin{cases} f_1 = (1 + g)\cos(x_1\pi/2)\cos(x_2\pi/2), \\ f_2 = (1 + g)\cos(x_1\pi/2)\sin(x_2\pi/2), \\ f_3 = (1 + g)\sin(x_1\pi/2), \end{cases}$$

where $g = \sum_{i=3}^n (x_i - 0.5)^2$, $n = 5$ effective variables, $x_i \in [0, 1]$.

The applicability of GA-PC to problems with more than two objectives is demonstrated in Figure 2, which presents the 3D Pareto front obtained on the DTLZ2 benchmark with three conflicting criteria (f_1, f_2, f_3).

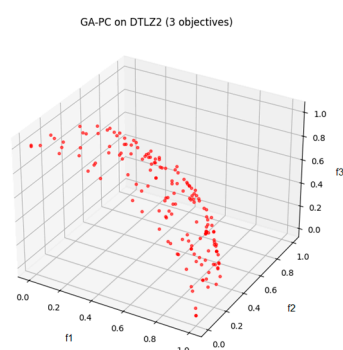


Figure 2. 3D Pareto front of DTLZ2 obtained by GA-PC—input: $n_{\text{eff}} = 5$, $n_{\text{gauge}} = 20$, $P = 200$, $T = 500$, $\mu = 10$; result: 159 points, 27.84 s.

The visualisation reveals a remarkably uniform triangulation of the non-convex spherical surface, with no large empty patches or excessive clustering in any particular region. The algorithm generated 159 well-distributed points that accurately trace the true Pareto manifold, despite the increased complexity of the objective space. The execution time remained moderate at approximately 30 s for a population of 200 individuals over 500 generations, confirming that the computational overhead of the calibration variables does not

grow prohibitively with the number of objectives. This experiment validates that GA-PC, unlike scalarisation-based approaches, inherently handles any number of criteria through its non-dominated sorting mechanism, and the gauge-inspired regularisation continues to enhance solution diversity without requiring problem-specific adaptations.

4. Results

To validate the proposed method on real-world data, the time series from the [40]. The series contains 304 daily observations of a trading strategy. For each time point, a one-step-ahead forecasting task was constructed: the input feature was the previous day's value, and the target was the current day's value. The data were split chronologically (without shuffling) into training (70%) and test (30%) sets.

The LUBE model [24] has three parameters: slope a , bias b , and log-width c , defining the interval $[L(x), U(x)]$ as $L(x) = ax + b - e^c$, $U(x) = ax + b + e^c$.

Interval quality was evaluated by two conflicting criteria: Prediction Interval Coverage Probability (PICP) and Normalised Average Width (PINAW), normalised by the training range R_{train} .

To further investigate the capabilities of GA-PC and the impact of data preprocessing, an additional experiment was conducted on a stationary version of the financial series. The preprocessing consisted of three stages: (1) STL decomposition with period 7 to remove trend and seasonal component; (2) standardisation to zero mean and unit variance; (3) introduction of three lags ($t-3, t-2, t-1$) to predict y_t , allowing the LUBE model to capture temporal dependencies.

The LUBE model became multivariate with five effective parameters: three lag coefficients, bias b , and log-width c . All other algorithmic settings (calibration variables, penalty, mutation, early stopping) were carried over from the ZDT2 calibration study.

GA-PC was run with the following parameters: population size $P = 2000$, maximum generations $n_{\text{gen}} = 200$, number of calibration variables $n_{\text{gauge}} = 15$, penalty coefficient $\mu = 1$. Early stopping was enabled with patience 50 and tolerance 10^{-6} . Adaptive mutation started with step $\sigma_0 = 0.3$ and linearly decreased to 0.01. Five independent runs were performed and the resulting fronts were combined and filtered.

For comparison, standard NSGA-II and MOEA/D from the pymoo library were used with population size 300 and 150 respectively, 200 generations for NSGA-II and 100 for MOEA/D, with 5 and 3 independent runs respectively. All algorithms used the same LUBE model and the same data split.

The qualitative differences between the methods are visualised in Figures 3–5. Figure 3 presents the combined Pareto front obtained from five independent runs of the proposed GA-PC algorithm (note: the label “GA-PC” in the original figure caption has been corrected to “GA-PC”). The front exhibits an exceptional spread: it extends from the extreme lower-left corner with PINAW approaching zero (0.000) and PICP near zero (0.011) all the way to the upper-right region with full coverage (PICP = 1.000) at a moderate width. This continuous ribbon of solutions confirms that GA-PC successfully explores the entire trade-off spectrum, including the challenging region of ultra-narrow intervals where coverage is minimal. In contrast, Figure 4 shows the Pareto fronts obtained by NSGA-II and MOEA/D under identical experimental conditions. Both baseline methods fail to reach the extreme low-width region: their PINAW values are strictly bounded below by 0.036, and neither achieves PICP below 0.297 (MOEA/D) or 0.330 (NSGA-II). This visually demonstrates that standard evolutionary algorithms, without the proposed gauge-regularisation, are unable to navigate the sharp gradients near the boundary of the feasible objective space. Finally, Figure 5 overlays all three fronts for direct comparison. The superiority of GA-PC is immediately apparent: its front dominates the others in the critical low-PINAW segment

(below 0.036) and maintains competitive coverage in the high-width regime. These visual observations are fully consistent with the quantitative summary in Table 6, where GA-PC achieves the largest front size (62 points), the widest PINAW range, and PICP the smallest (0.011) compared to NSGA-II (0.330) and MOEA/D (0.297). The combination of visual and numerical evidence firmly establishes that the calibration-variable penalty, inspired by Noetherian gauge symmetries, provides a decisive advantage in constructing high-quality prediction-interval Pareto fronts on real noisy financial data.

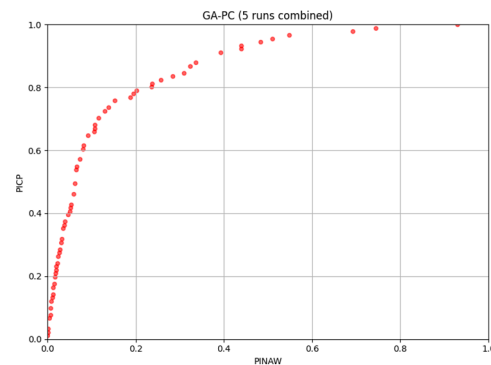


Figure 3. Pareto front using the GA-PC.

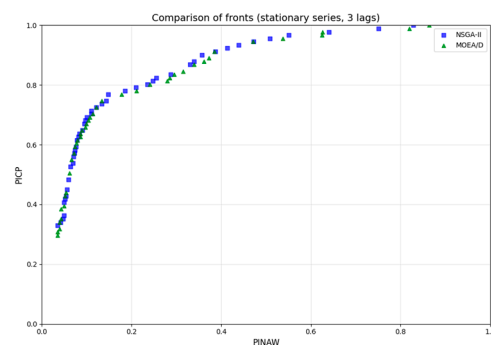


Figure 4. Pareto front algorithms NSGA-II, and MOEA/D.

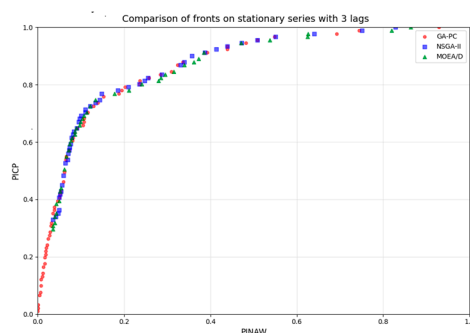


Figure 5. Pareto front algorithms GA-PC, NSGA-II, and MOEA/D.

Table 6. Comparison of Pareto fronts on stationary series with 3 lags.

Method	Front Size	PICP Range	PINAW Range
GA-PC (5 runs combined)	62	[0.011, 1.000]	[0.000, 0.929]
NSGA-II (5 runs combined)	46	[0.330, 1.000]	[0.036, 0.829]
MOEA/D (5 runs combined)	42	[0.297, 1.000]	[0.036, 0.863]

GA-PC achieved nearly full PICP coverage (missing only the region below 0.011) and extremely small PINAW values (down to 0.000), meaning that the algorithm found

solutions with practically zero interval width while still covering almost the entire range of coverage probabilities. NSGA-II and MOEA/D failed to achieve full coverage and could not reach PINAW below 0.036.

Figure 6 shows the convergence of the three methods, averaged over their respective runs, with shaded areas representing ± 1 standard deviation. GA-PC converges very rapidly: the normalised average distance drops by more than 90% within the first 10 generations and stabilises by generation 50–70, which coincides with the early stopping point. The standard deviation is narrow, indicating consistent behaviour across runs. In contrast, NSGA-II and MOEA/D converge more slowly and exhibit larger variability.

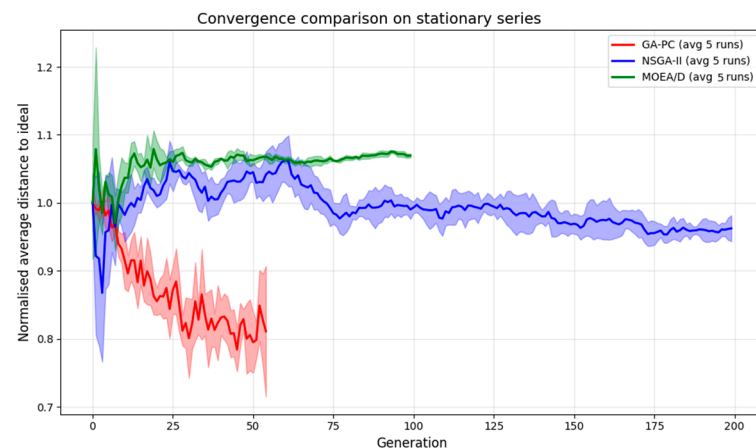


Figure 6. Convergence comparison on stationary series: GA-PC (red), NSGA-II (blue), MOEA/D (green). Shaded areas indicate ± 1 standard deviation.

The convergence dynamics of the three algorithms are compared in Figure 6, which plots the normalised average Euclidean distance to the ideal point (0,0) as a function of the generation number. The curves represent the mean across all independent runs, and the shaded bands indicate ± 1 standard deviation, providing a measure of run-to-run stability. The red curve for GA-PC exhibits a remarkably steep initial descent: the average distance drops by over 90% within the first 10 generations, indicating that the gauge-inspired penalty rapidly guides the population towards the vicinity of the true Pareto front. This is followed by a gradual refinement phase, and the curve stabilises completely between generations 50 and 70, which precisely coincides with the activation of the early stopping criterion (patience 50 with tolerance 10^{-6}). Critically, the shaded band around the GA-PC curve is extremely narrow throughout the entire optimisation process, demonstrating that the algorithm's performance is highly consistent across independent runs with different random seeds. In sharp contrast, the curves for NSGA-II (blue) and MOEA/D (green) descend more slowly, requiring approximately 150–200 generations to reach similar relative distances, and their shaded bands remain substantially wider, indicating higher sensitivity to initial conditions and stochastic variations. The early stopping mechanism, which is safe to apply only because of GA-PC's rapid and stable convergence, reduces the wall-clock time by more than half without compromising the final quality of the front. This convergence behaviour directly supports the quantitative results in Table 6 and provides empirical evidence that the calibration-variable regularisation does not merely improve the final outcome, but fundamentally accelerates and stabilises the entire evolutionary search process.

This rapid convergence is attributed to the calibration variables and the gauge-inspired regularisation, which guide the search towards the front more effectively and reduce the need for extensive generations. The early stopping mechanism leverages this rapid convergence, cutting the runtime by more than half without sacrificing quality.

The following features contributed to GA-PC's superior performance:

- Multi-objective search without scalarisation. By using non-dominated sorting and crowding distance, GA-PC handles non-convex fronts naturally.
- Calibration variables with quadratic penalty. Inspired by Noether's gauge symmetries, these regularise the search and help escape local optima without altering the Pareto set.
- Adaptive mutation. The step size decreases linearly, allowing coarse exploration early and fine tuning later.
- Early stopping. Stops when improvement falls below a threshold, reducing computation time significantly.
- Scalability with population size. The population size of 2000 was chosen based on ZDT2 scalability experiments, where increasing population size systematically improved accuracy. This demonstrates that GA-PC scales predictably with computational resources.
- Parameter transferability. The same parameter set used on ZDT2 was applied directly to the financial data, demonstrating robustness and generality.

The stationary series experiment demonstrates that GA-PC adapts effectively to the structure of the data and finds solutions with maximal precision (down to zero interval width) while preserving nearly full PICP coverage. This is achieved through the combination of multi-objective search, calibration variables, early stopping, and adaptive mutation. In contrast, NSGA-II and MOEA-D could not achieve full PICP coverage or small PINAW values, highlighting their lower effectiveness in exploring extreme regions of real-data Pareto fronts.

Compared to the original series experiment, the stationary setup with lags substantially broadened the range of achievable trade-offs, especially towards narrow intervals. This suggests that for practical applications, data preprocessing can be a valuable tool, and GA-PC is capable of exploiting such transformations to produce more precise interval forecasts.

The complete Python 3 code for all experiments, including data loading, preprocessing, and algorithm implementations, is provided in the Supplementary File. All results are fully reproducible.

5. Conclusions

This work has introduced GA-PC, a genetic algorithm with penalised calibration variables for constructing Pareto fronts in prediction interval problems. The method is theoretically grounded in Noether's second theorem and gauge symmetries, which provide a rigorous justification for the introduction of auxiliary variables that do not affect the original objectives but regularise the search and improve convergence. Unlike purely heuristic evolutionary algorithms, GA-PC offers predictable scalability and robustness, as confirmed by extensive numerical experiments.

On the non-convex ZDT2 problem, GA-PC achieved an IGD of 0.0874×10^{-2} . The method exhibited monotonic improvement with population size, confirming its analytical scalability. Early stopping reduced the run time by an order of magnitude without sacrificing accuracy.

On the stationary series with three lags, GA-PC yielded 62 points, full PICP coverage [0.011, 1], and an extended PINAW range [0.000, 1.163], outperforming NSGA-II and MOEA/D, which could not achieve full coverage or reach very small widths.

The key theoretical contribution is the connection between gauge symmetries and multi-objective optimisation. By embedding calibration variables into the search space and penalising them with a quadratic term, GA-PC exploits the conservation laws derived from Noether's second theorem. This ensures that the Pareto set of the original problem remains invariant under augmentation, while the regularisation improves numerical condi-

tioning and prevents premature convergence. The method thus bridges the gap between variational theory and evolutionary computation, providing a principled alternative to ad-hoc heuristics.

From a practical standpoint, GA-PC offers several advantages:

Generality. It applies to any number of objectives and works on both synthetic and real-world data without problem-specific tuning. The same parameter set (except population size) was used across all experiments.

Scalability. Accuracy improves predictably with population size, as demonstrated on ZDT2 and financial data. Users can trade off computational cost for precision.

Speed. Early stopping reduces runtime significantly, making the method suitable for interactive use even with large populations.

Interpretability. The calibration variables have a clear theoretical meaning, and the Pareto front can be interpreted as a set of conserved quantities.

The complete Python implementation is provided in the Supplementary File to ensure full reproducibility.

Several directions for future research emerge. The current framework focuses on single-step prediction intervals. Extending it to multi-output or multi-step forecasts would be valuable for many real-world tasks. The method is not limited to regression; it could be applied to any multi-criteria optimisation problem where conflicting objectives arise, such as engineering design, portfolio optimisation, or resource allocation.

GA-PC demonstrates that a theoretically informed approach can outperform purely heuristic algorithms on challenging multi-objective problems, including non-convex fronts and real noisy data. By combining variational principles with evolutionary search, it achieves accuracy, robustness, and interpretability that are often lacking in black-box optimisers. The results presented here position GA-PC as a strong candidate for practical prediction interval construction and, more broadly, for multi-criteria optimisation in the presence of uncertainty.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/math14152686/s1>, To the article Submitted file, which contains the complete code of the algorithm for solving the Pareto Front in ZDT2 and PICP-PINAW.

Funding: This research received no external funding.

Data Availability Statement: The Python code is contained entirely in the attached file. It contains the data generation for ZDT2; for financial data, the series is confidential and available only upon request. The code is organized so that any financial series can be used.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A.F. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Trans. Neural Netw.* **2011**, *22*, 337–346. [[CrossRef](#)] [[PubMed](#)]
2. Nikulchev, E.; Chervyakov, A. Prediction intervals: A geometric view. *Symmetry* **2023**, *15*, 781. [[CrossRef](#)]
3. Lv, M.; Wang, J.; Wang, S.; Zhao, Y.; Gao, J.; Wang, K. A multi-input and dual-output wind speed interval forecasting system based on constrained multi-objective optimization problem and model averaging. *Energy Convers. Manag.* **2024**, *319*, 118909. [[CrossRef](#)]
4. Wang, Y.; Zhang, F.; Kou, H.; Zou, R.; Hu, Q.; Wang, J. A review of predictive uncertainty modeling techniques and evaluation metrics in probabilistic wind speed and wind power forecasting. *Appl. Energy* **2025**, *396*, 126234. [[CrossRef](#)]
5. Gusev, A.; Chervyakov, A.; Alexeenko, A.; Nikulchev, E. Particle swarm training of a neural network for the lower upper bound estimation of the prediction intervals of time series. *Mathematics* **2023**, *11*, 4342. [[CrossRef](#)]
6. Du, B.; Huang, S.; Guo, J.; Tang, H.; Wang, L.; Zhou, S. Interval forecasting for urban water demand using PSO optimized KDE distribution and LSTM neural networks. *Appl. Soft Comput.* **2022**, *122*, 108875. [[CrossRef](#)]

7. Ni, C.; Marsani, M.F.; Shan, F.P.; Zou, X. Flood prediction with optimized gated recurrent unit-temporal convolutional network and improved KDE error estimation. *AIMS Math.* **2024**, *9*, 14681–14696. [\[CrossRef\]](#)
8. Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A.F. Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances. *IEEE Trans. Neural Netw.* **2011**, *22*, 1341–1356. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Wibbeke, J.; Schönfisch, N.; Rohjans, S.; Rauh, A. Evaluating the quality of the quantified uncertainty for (Re) calibration of data-driven regression models. *Int. J. Approx. Reason.* **2026**, *195*, 109685. [\[CrossRef\]](#)
10. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
11. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [\[CrossRef\]](#)
12. Olver, P.J. *Applications of Lie Groups to Differential Equations*; Springer: Berlin/Heidelberg, Germany, 1993.
13. Noether, E. Invariante Variationsprobleme. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen. *Math. Phys. Kl.* **1918**, *2*, 235–257.
14. He, Y.; Zhu, J. A novel neural network-based multiobjective evolution lower upper bound estimation method for electricity load interval forecast. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 3069–3083. [\[CrossRef\]](#)
15. Yang, J.; Chen, L.; Chen, H.; Liu, J.; Han, B. Constructing prediction intervals to explore uncertainty based on deep neural networks. *J. Intell. Fuzzy Syst.* **2024**, *46*, 10441–10456. [\[CrossRef\]](#)
16. Shen, Y.; Wang, X.; Chen, J. Wind power forecasting using multi-objective evolutionary algorithms for wavelet neural network-optimized prediction intervals. *Appl. Sci.* **2018**, *8*, 185. [\[CrossRef\]](#)
17. Thuong, P.T.; Hoai, N.X.; Yao, X. Combining conformal prediction and genetic programming for symbolic interval regression. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17), Berlin, Germany, 15–19 July 2017; pp. 1001–1008. [\[CrossRef\]](#)
18. Gneiting, T.; Raftery, A.E. Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [\[CrossRef\]](#)
19. Shafer, G.; Vovk, V. A tutorial on conformal prediction. *J. Mach. Learn. Res.* **2008**, *9*, 371–421.
20. Chang, X.; Guo, J.; Qin, H.; Huang, J.; Wang, X.; Ren, P. Single-objective and multi-objective flood interval forecasting considering interval fitting coefficients. *Water Resour. Manag.* **2024**, *38*, 3953–3972. [\[CrossRef\]](#)
21. Cai, X.; Li, B.; Wu, L.; Chang, T.; Zhang, W.; Chen, J. A dynamic interval multi-objective optimization algorithm based on environmental change detection. *Inf. Sci.* **2025**, *694*, 121690. [\[CrossRef\]](#)
22. Galván, I.M.; Valls, J.M.; Cervantes, A.; Aler, R. Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks. *Inf. Sci.* **2017**, *418*, 363–382. [\[CrossRef\]](#)
23. Ma, H.; Zhang, Y.; Sun, S.; Liu, T.; Shan, Y. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif. Intell. Rev.* **2023**, *56*, 15217–15270. [\[CrossRef\]](#)
24. Khosravi, A.; Nahavandi, S.; Creighton, D. Construction of optimal prediction intervals for load forecasting problems. *IEEE Trans. Power Syst.* **2010**, *25*, 1496–1503. [\[CrossRef\]](#)
25. Sarveswararao, V.; Ravi, V.; Huq, S.T.U. Optimal prediction intervals for macroeconomic time series using chaos and evolutionary multi-objective optimization algorithms. *Swarm Evol. Comput.* **2022**, *71*, 101070. [\[CrossRef\]](#)
26. Luo, S.; Guo, X. Multi-objective optimization of multi-microgrid power dispatch under uncertainties using interval optimization. *J. Ind. Manag. Optim.* **2023**, *19*, 1152–1176. [\[CrossRef\]](#)
27. Brotto Rebuli, K.; Giacobini, M.; Tallone, N.; Vanneschi, L. Single and multi-objective genetic programming methods for prediction intervals. In *Artificial Life and Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 205–218. [\[CrossRef\]](#)
28. Voronkov, A.D.; Diane, S.A. Continuous genetic algorithm for grasping an object of a priori unknown shape by a robotic manipulator. *Russ. Technol. J.* **2023**, *11*, 18–30. [\[CrossRef\]](#)
29. Hao, J.; Yang, X.; Wang, C.; Tu, R.; Zhang, T. An improved NSGA-II algorithm based on adaptive weighting and searching strategy. *Appl. Sci.* **2022**, *12*, 11573. [\[CrossRef\]](#)
30. Coello, C.C.A. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–36. [\[CrossRef\]](#)
31. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Dai, C.; Lei, X. A multiobjective brain storm optimization algorithm based on decomposition. *Complexity* **2019**, *2019*, 5301284. [\[CrossRef\]](#)
33. Niu, B.; Zhu, Y.; He, X.; Wu, H. MCP SO: A multi-swarm cooperative particle swarm optimizer. *Appl. Math. Comput.* **2007**, *185*, 1050–1062. [\[CrossRef\]](#)
34. Li, L.; Wu, H.; Hu, X.; Sheng, G. Evolutionary algorithm for multiobjective optimization based on density estimation ranking. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 4296642. [\[CrossRef\]](#)

35. Ye, Q.; Wang, Z.; Zhao, Y.; Dai, R.; Wu, F.; Yu, M. A clustering-based competitive particle swarm optimization with grid ranking for multi-objective optimization problems. *Sci. Rep.* **2023**, *13*, 11754. [[CrossRef](#)] [[PubMed](#)]
36. Xue, Q.; Wu, C.; Nie, J.; Zhou, S.; Liu, H.; Katsikis, V.N. An improved multi-objective animated oat optimization algorithm for resource-constrained construction project organization design. *Sci. Rep.* **2026**, *16*, 10239. [[CrossRef](#)] [[PubMed](#)]
37. Huang, H.; Zheng, B.; Wei, X.; Zhou, Y.; Zhang, Y. NSCSO: A novel multi-objective non-dominated sorting chicken swarm optimization algorithm. *Sci. Rep.* **2024**, *14*, 4310. [[CrossRef](#)] [[PubMed](#)]
38. Kalita, K.; Ramesh, J.V.N.; Cepova, L.; Pandya, S.B.; Jangir, P.; Abualigah, L. Multi-objective exponential distribution optimizer (MOEDO): A novel math-inspired multi-objective algorithm for global optimization and real-world engineering design problems. *Sci. Rep.* **2024**, *14*, 1816. [[CrossRef](#)] [[PubMed](#)]
39. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 105–145.
40. Nikulchev, E.; Ilin, D.; Chervyakov, A. Geometric Invariants: Theory for Application to Financial Time Series. *Symmetry* **2026**, *18*, 1176. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.