mathematics

MDPI

Article

CoEGAN-BO: Synergistic Co-Evolution of GANs and Bayesian Optimization for High-Dimensional Expensive Many-Objective Problems

Jie Tian [1,*], Hongli Bian [2], Yuyao Zhang [3], Xiaoxu Zhang [4] and Hui Liu [1]

[1] School of Data and Computer Science, Shandong Women's University, No. 2399 University Road, Jinan 250300, China
[2] School of Computer Science and Engineering, Shandong Normal University, No. 1 University Road, Jinan 250014, China
[3] School of Computer Science and Technology, Shandong Jianzhu University, No. 1000 Fengming Road, Jinan 250101, China
[4] School of Information Science and Engineering, University of Jinan, No. 336 Nanxinzhuang West Road, Jinan 250022, China
* Correspondence: 29052@sdwu.edu.cn

Abstract

Bayesian optimization (BO) struggles with data scarcity and poor scalability in high-dimensional many-objective optimization problems. To address this, we propose Co-Evolutionary GAN–Bayesian Optimization (CoEGAN-BO), a novel framework that synergizes generative adversarial networks (GANs) with Bayesian co-evolutionary search for data-driven optimization. The GAN module generates synthetic samples conditioned on promising regions identified by BO, while a co-evolutionary mechanism maintains two interacting populations: one explores the GAN's latent space for diversity, and the other exploits BO's probabilistic model for convergence. A bi-stage infilling strategy further enhances efficiency: early iterations prioritize exploration via L_p-norm-based candidate selection, later switching to a max–min distance criterion for Pareto refinement. Experiments on expensive multi/many-objective benchmarks show that CoEGAN-BO outperforms four state-of-the-art surrogate-assisted algorithms, achieving superior convergence and diversity under limited evaluation budgets.

Keywords: high-dimensional expensive many-objective problems; Bayesian optimization; co-evolution; generative adversarial networks; infill criterion

MSC: 68T20

check for updates

Academic Editor: Víctor Yepes

Received: 17 September 2025
Revised: 14 October 2025
Accepted: 15 October 2025
Published: 29 October 2025

Citation: Tian, J.; Bian, H.; Zhang, Y.; Zhang, X.; Liu, H. CoEGAN-BO: Synergistic Co-Evolution of GANs and Bayesian Optimization for High-Dimensional Expensive Many-Objective Problems. *Mathematics* **2025**, *13*, 3444. https://doi.org/10.3390/math13213444

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Evolutionary computation has increasingly focused on complex engineering problems characterized by multiple conflicting objectives [1,2] in recent years. These problems, prevalent in domains such as aerodynamic design optimization [3], control engineering systems [4], and drug design optimization [5], pose significant challenges due to their high-dimensional and computationally expensive nature. Each objective requires expensive computational costs, rendering the problem an expensive multi-objective optimization problem (EMOP) [6–8] as follows:

$$\min F(\boldsymbol{x}) = \min(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \quad (1)$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_D) \in \Omega^D$ is a D-dimensional variable vector in the decision space Ω^D, while m and D represent the number of objectives and the dimensions of the decision variables, respectively. When each objective function $f_i(\boldsymbol{x})$ requires exorbitant computational overhead, highlighting a scenario where $D \geq 30$ and $m \geq 3$, $F(\boldsymbol{x})$ is indicative of high-dimensional expensive many-objective optimization problems (HEMaOPs) [9–12].

Bayesian optimization algorithms (BOs) [13–15] have emerged as the predominant methods for addressing expensive optimization problems. BOs significantly economize fitness evaluations, which are an intensive and time-consuming aspect of traditional evolutionary algorithms (EAs) [16,17]. By utilizing surrogate models constructed from preprocessed historical data to approximate true models, BOs efficiently pre-screen candidate solutions, substantially reducing the need for expensive original fitness evaluations [13,14]. In essence, only those solutions predicted to be promising undergo expensive computational or physical resource consumption, while the rest benefit from less resource-intensive surrogate models [18]. This strategic approach effectively addresses expensive optimization problems by saving time and resources that would otherwise be spent on extensive fitness evaluations [19].

Furthermore, when traditional optimization problems encounter expensive, high-dimensional, multi-objective problems, the difficulty of optimization is further amplified.

- Constructing a high-fidelity model from limited sample data poses substantial challenges, encountering issues such as reduced optimization efficiency and compromised model precision [20]. Moreover, the time required for model construction escalates exponentially with rising dimensionality, further complicating the optimization process [21].
- As the dimension of the objective space increases, the number of non-dominated solutions tends to burgeon exponentially, thereby intensifying selection pressure throughout the evolutionary process [22].
- Inherently, the sparsity of samples in the high-dimensional decision spaces contributes to a significant level of uncertainty in the value of each predictive sample [23]. This uncertainty complicates the selection of the most advantageous solutions, which are critical for enhancing the precision of the model while concurrently providing more effective guidance for the population's search trajectory.

There are currently three main approaches to addressing issues encountered when using BO. We provide a more detailed discussion in Table 1.

- Model-Construction-Based BO: In pursuit of refining the surrogate-assisted model's fidelity, hybrid surrogate modeling has become a key technique [24]. This method capitalizes on the synergy of multiple models to better fit the real objective function, thereby reducing overreliance on the predictive power of any single model. Surrogate models such as the Radial Basis Function (RBF) model [25], Artificial Neural Networks (ANNs) [26], Gaussian process [27,28], and Support Vector Machine (SVM) models [29] are utilized for their respective strengths in various optimization contexts. This illustrates the tailored application of surrogates to different optimization problems.
- Model-Management-Based BO: As the complexity of the objective function increases, not only does the prediction error of the surrogate model intensify, but it also exacerbates the challenge of approaching the Pareto front, making the means of distinguishing between better individuals more difficult. These challenges can be addressed by adopting a more efficient infill criterion.
- Novel Search-Strategy-Based BO: Environment selection is an important part of multi-objective evolutionary algorithms [30]. Individuals with better fitness are selected as the next initial population from the mixed population. Existing methods still use the predicted value and uncertainty of the model to select the next initial population.

However, the predicted value and uncertainty variance given by the surrogate model decrease rapidly with the increase in the dimensions. Therefore, the effectiveness of the evolutionary algorithm is also reduced.

Table 1. Summary table of Bayesian optimization (BO)-related methods.

Method Category	Algorithm	Summary of Algorithm Ideas
Model-Construction-Based BO	MOEA/D-EGO [31]	Adopt fuzzy clustering-based modeling in the decision space; construct multiple local surrogate models for each objective function.
	KTA2 [32]	Combine two point-insensitive models with the original Gaussian process to adaptively approximate each objective function, minimizing outlier impacts.
	CLBO [33]	Build several Gaussian processes; impose protocol constraints on all/part of them during training to ensure reasonable and accurate predictions.
Model-Management-Based BO	EI (Expected Improvement) [24,34]	A common infill criterion for selecting promising candidates. Loses effectiveness with increasing dimension (uncertainty differences shrink) and fails to optimize all problems.
	PI (Probability of Improvement) [35,36]	Another widely used infill criterion. Similar to EI: dimension growth reduces uncertainty distinction, leading to invalidation; not optimal for all problems.
	LCB (Lower Confidence Bound) [37,38]	A mainstream infill criterion. Faces the same issues as EI/PI: dimension increase weakens uncertainty measurement, and a single criterion cannot fit all scenarios.
Novel Search-Strategy-Based BO	NSGA-III [39]	Integrate Pareto dominance and decomposition strategies; balance convergence and diversity effectively in high-dimensional problems.
	Two-Arch2 [40]	Establish two updated archives (via metrics and Pareto dominance). Performs well in high dimensions by considering both convergence and diversity.

Building on the insights gained from existing research and aiming to address the optimization bottlenecks inherent in high-dimensional expensive problems, this paper pivots towards enhancing the algorithm's convergence capabilities when faced with numerous non-dominant solutions. Recognizing the necessity for a more efficient approach, we present a novel optimization framework—the Bi-Stage Infill Criterion Bayesian Co-evolutionary Optimization Algorithm (`CoEGAN-BO`). This algorithm is specifically tailored for many-objective, high-dimensional, and expensive optimization challenges. `CoEGAN-BO` leverages a cutting-edge generative adversarial network to adeptly select potential solutions that significantly propel the algorithm's convergence. By doing so, it not only minimizes the need for expensive fitness evaluations but also accelerates the convergence rate, culminating in the attainment of better optimal solutions. The key contribution of our work lies in the selection mechanism following this secondary assessment. Specifically, we extract those individuals that the GAN identifies as "real" from the pool of predicted samples. This approach allows us to isolate and prioritize the high-fidelity samples, which

are deemed by the trained GAN to closely resemble the distribution of the actual computed samples. By integrating the GAN-based selection process, we are able to mitigate the risks associated with the surrogate model's inaccuracies. This is particularly critical in optimization tasks where the search path could be significantly altered due to misleading predictions. Thus, our approach ensures that the optimization process remains robust against such deviations, maintaining the integrity of the search trajectory toward the global optimum. The major contributions of this paper can be summarized as follows:

(1) The proposed method adeptly extracts useful information from both real and predicted samples. It employs a secondary assessment of the current model's predictive outcomes. This is designed to mitigate the deterioration in predictive accuracy that can arise due to the trajectory deviation of the surrogate model in the context of high-dimensional expensive optimization challenges.
(2) We introduce an efficient model management strategy. This strategy synergistically combines the L_p-norm with a convergence criterion based on the extremities of distances (maximum and minimum). The main purpose is to significantly reduce computational complexity and to be able to accurately determine which individuals have high "potential".
(3) The proposed `CoEGAN-BO` exhibits considerable advantages when addressing the "curse of dimensionality" challenges in HEMaOPs. The `CoEGAN-BO` algorithm accelerates population evolution by conducting a secondary selection of more competitive predicted solutions from the non-dominated solutions by reducing non-dominated selection pressure.
(4) The proposed `CoEGAN-BO` demonstrates outstanding performance; the experimental results clearly demonstrate that this method is highly competitive in solving high-dimensional problems of 50D and 100D, as well as many-objective problems with 3, 4, 6, 8, and 10 objectives.

The remainder of this paper is organized as follows: Section 2 provides a concise overview of the key techniques utilized in this research: Gaussian processes and generative adversarial networks. Within this context, Gaussian processes are employed for model construction, while generative adversarial networks are applied for the secondary estimation of predicted samples. Section 3 gives a detailed description of the proposed algorithm. The experimental results and analysis are presented in Section 4. The conclusions and future prospects are provided in Section 5.

2. Preliminaries

2.1. Gaussian Process

The Gaussian process (GP) [27,28] is a method renowned for its efficacy in enhancing the forecasting efficiency of engineering problems. Due to its effectiveness, the GP is extensively utilized across diverse engineering disciplines. Also referred to as the kriging model, this method reflects the spatial correlation between known and unknown points by making assumptions. Predictions at interpolation points are made using the GP, which is formulated as follows:

$$\widehat{y}(x) = \mu + \varepsilon(x) \tag{2}$$

where $\widehat{y}(x)$ represents the predicted output at input location x; μ denotes the constant mean parameter, representing the global average of the process; and $\varepsilon(x)$ signifies a zero-mean Gaussian process with the covariance function $k(x, x')$, where $\varepsilon(x) \sim \mathcal{N}(0, \delta^2)$.

Given N training points, the predictive distribution at any test point x is Gaussian, with the mean given by

$$\widehat{f}(x) = \widehat{\mu} + p^T P^{-1}(f - l\widehat{\mu}) \tag{3}$$

where $\widehat{f}(x)$ is the predicted mean value at point x; $\widehat{\mu}$ is the estimated global mean, computed as $\widehat{\mu} = (l^T P^{-1} l)^{-1} l^T P^{-1} f$; p is an $N \times 1$ correlation vector between the test point x and training points, with elements $p_i = \text{corr}(x, x_i)$; P is an $N \times N$ correlation matrix among training points, where $P_{ij} = \text{corr}(x_i, x_j)$; f is an $N \times 1$ vector of observed outputs at training points, where $f = [y_1, y_2, \ldots, y_N]^T$; and l is an $N \times 1$ vector of ones, where $l = [1, 1, \ldots, 1]^T$.

The prediction uncertainty is quantified by the variance:

$$s^2(x) = \widehat{\delta}^2 \left[1 - p^T P^{-1} p + \frac{(1 - l^T P^{-1} p)^2}{l^T P^{-1} l} \right] \tag{4}$$

where $s^2(x)$ represents the prediction variance at point x, measuring uncertainty; and $\widehat{\delta}^2$ is the estimated process variance, typically computed as $\widehat{\delta}^2 = \frac{1}{N}(f - l\widehat{\mu})^T P^{-1} (f - l\widehat{\mu})$. The term $p^T P^{-1} p$ accounts for the reduction in uncertainty due to proximity to training points. The fraction term $\frac{(1 - l^T P^{-1} p)^2}{l^T P^{-1} l}$ represents additional uncertainty from estimating the mean parameter.

The Gaussian process framework assumes that the covariance between any two points is related through the correlation matrix P and correlation vector p. This spatial correlation structure enables the GP to provide both accurate predictions and reliable uncertainty estimates, making it particularly valuable for engineering applications requiring risk assessment and decision-making under uncertainty.

2.2. Generative Adversarial Networks

Generative adversarial networks (GANs) [41,42] offer a powerful framework for learning data representations even in the absence of extensive labeled training datasets. This innovative approach circumvents the need for explicit modeling of the probability density function and has achieved widespread application across various domains, including unsupervised representation learning, image synthesis, and image super-resolution enhancement.

The foundational structure of the GAN is depicted in Figure 1. At its core, the GAN is composed of two neural networks that are trained concurrently: the Generator (G) and the Discriminator (D). The Generator's role is to approximate the distribution of real data by generating synthetic samples that are indistinguishable from actual data, thereby attempting to deceive the Discriminator into accepting the fabricated data as authentic. On the other hand, the Discriminator functions as a classifier that discerns whether a sample originates from the real dataset or has been concocted by the Generator.

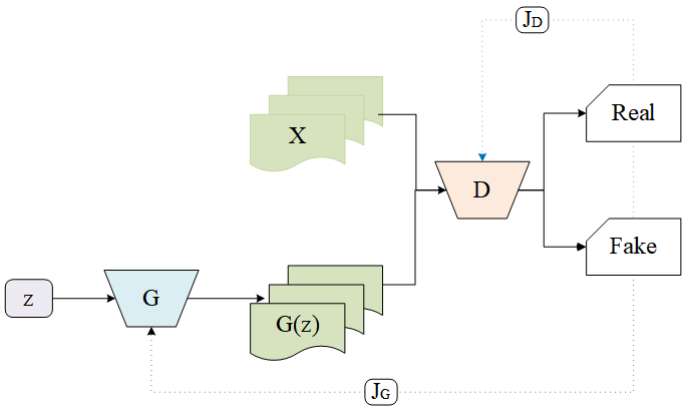


Figure 1. The architecture of the generative adversarial network (GAN).

Engaged in a continuous adversarial contest, both networks employ backpropagation to update their weights, thereby progressively honing their respective capabilities. The Generator strives to maximize the likelihood of creating convincing forgeries, while the

Discriminator aims to minimize the chances of erroneously classifying fake data as real. Through this adversarial training process, both networks incrementally escalate their competencies, learning from each other's weaknesses and thereby advancing the model's overall performance.

The Discriminator within a generative adversarial network bears the crucial task of discerning the authenticity of data. It acts as a judge, evaluating each sample to ascertain whether it is genuine or a counterfeit produced by the Generator. The objectives of the Discriminator are twofold:

Accuracy in Classification: The Discriminator must accurately distinguish between real and generated (fake) samples, minimizing the likelihood of misclassification.

Adversarial Improvement: Through iterative training, the Discriminator seeks to refine its ability to identify subtle cues that differentiate authentic data from imitations, thereby providing a robust challenge to the Generator. These goals ensure that the Discriminator evolves in tandem with the Generator, maintaining a delicate balance where each component improves in response to the other's advancements, driving the overall system towards more sophisticated data representation and generation capabilities.

$$\max E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z}[\log(1 - D(G(z)))] \tag{5}$$

In Equation (5), the first part is to make the Generator judge whether the real sample is real with high probability. The second part is to minimize the probability of the Generator discriminating the fake sample as true.

GANs learn the distribution of the given data by adversarial means. This makes the Generator fit the distribution close to the real data. It is the distribution of the PS (Pareto set), which provides convenience for us to find the true PF [43]. At the same time, the trained discriminator completes the binary classification task, making it possible to distinguish what is generator-generated data, in order to complete an effective control solution. GMOEA [44] proposes dividing the population into two different datasets based on the fitness evaluation criterion. It uses two datasets to train the GAN model. The GAN and genetic operators are used to generate some offspring to complete the optimization work. EGANS [45] proposes an evolutionary generator architecture search strategy. It employs a many-to-one adversarial training strategy to evolutionarily search for the optimal Generator. In this way, a generative network with good adaptability and stability is designed. EWSGAN [46] trains a weight-sharing super-network through GANs and identifies the optimal subnet. It can be seen that the GAN has great potential in the optimization field. We propose CoEGAN-BO, an efficient GAN-based environmental selection strategy to guide population evolution.

In this study, we leverage the power of generative adversarial networks (GANs) by utilizing samples that have undergone actual computation as the "real" samples while treating the predicted samples as the "generated" ones. We train the GAN in such a way that it becomes capable of conducting a secondary assessment of the predicted samples. Through this process, the GAN evaluates the authenticity of each predicted sample, akin to distinguishing between the genuine and the imitative.

3. The Proposed `CoEGAN-BO` Algorithm

In this work, we introduce `CoEGAN-BO`, a novel strategy tailored for tackling HEMaOPs. Mirroring the structure of existing SAEAs, `CoEGAN-BO` is comprised of a triad of core components: an optimizer, a surrogate model, and an adaptive sampling criterion. Initially, this section outlines the overarching architecture of `CoEGAN-BO`. This is followed by an exposition on the innovative use of generative adversarial networks (GANs) for environmental selection, which plays a pivotal role in guiding the optimization process. Subsequently,

we introduce a distinctive infilling criterion that holistically considers both the convergence and diversity of the algorithm, ensuring a balanced exploration and exploitation throughout the optimization process.

The computational framework of the proposed `CoEGAN-BO` is given in Algorithm 1. The algorithm adopts an individual-based model management strategy. This means that `CoEGAN-BO` will select several individuals for expensive fitness evaluation. These selected individuals are then utilized to iteratively refine both the Gaussian process and the GAN model with each successive generation. The flowchart in Figure 2 presents a comprehensive overview of an evolutionary algorithm that incorporates the GAN for environmental selection and uses surrogate models to facilitate the optimization process, especially in scenarios where evaluations are computationally expensive. Throughout the process, the database (DB) plays a central role by providing the necessary data for rebuilding the surrogate and GAN models, ensuring that they are kept up-to-date with the latest insights from the search space. Using the GAN for environment selection introduces a novel approach to guide the evolutionary process by selecting individuals that are not only high-performing according to surrogate models but also resemble authentic data points, potentially leading to more efficient and effective optimization.

Algorithm 1 Framework of `CoEGAN-BO`

Require: $FE_{\max}$ (maximum number of fitness evaluations); NI (number of initialized solutions); N (population size); $\omega_{\max}$ (maximum generations before Gaussian process update); η (number of fitness evaluations in each generation).
Ensure: PS (Pareto set).
1: Initialization: Employ LHS to generate design samples and evaluate them; archive all of the evaluated solutions into the DB, set $FE = NI$;
2: Establish Gaussian process and GAN for each objective function using the DB samples;
3: From the initial design points, select NP superior candidates to form the initial population P;
4: **while** $FE <= FE_{\max}$ **do**
5: **while** $\omega <= \omega_{\max}$ **do**
6: Select mating parents from P^{ω} and generate offspring O^{ω} via each optimizer;
7: Estimate the offspring by the surrogate-assisted models;
8: Select NP best individuals to be the next population $P^{\omega+1}$;
9: Increment generation counter:$\omega = \omega + 1$;
10: **end while**
11: Employ an infill criterion to identify η optimal offspring;
12: Execute original fitness evaluations for these candidates;
13: Update FE accordingly: $FE = FE + \eta$;
14: Update surrogate models with new evaluated data;
15: **end while**

The operational specifics of `CoEGAN-BO` are delineated in Algorithm 1. An initial sampling of candidates is generated via Latin hypercube sampling, with the evaluated samples stored in the database (DB). These samples form the foundation for constructing objective-wise surrogate-assisted and GAN models.

Within the `CoEGAN-BO` framework, popular multi-objective evolutionary algorithms such as MOEA/D-DE [47] and RVEA [48] are integrated as the baseline optimization engines. The algorithm employs genetic operators to spawn offspring, while a surrogate model is utilized to estimate the fitness value of each objective function.

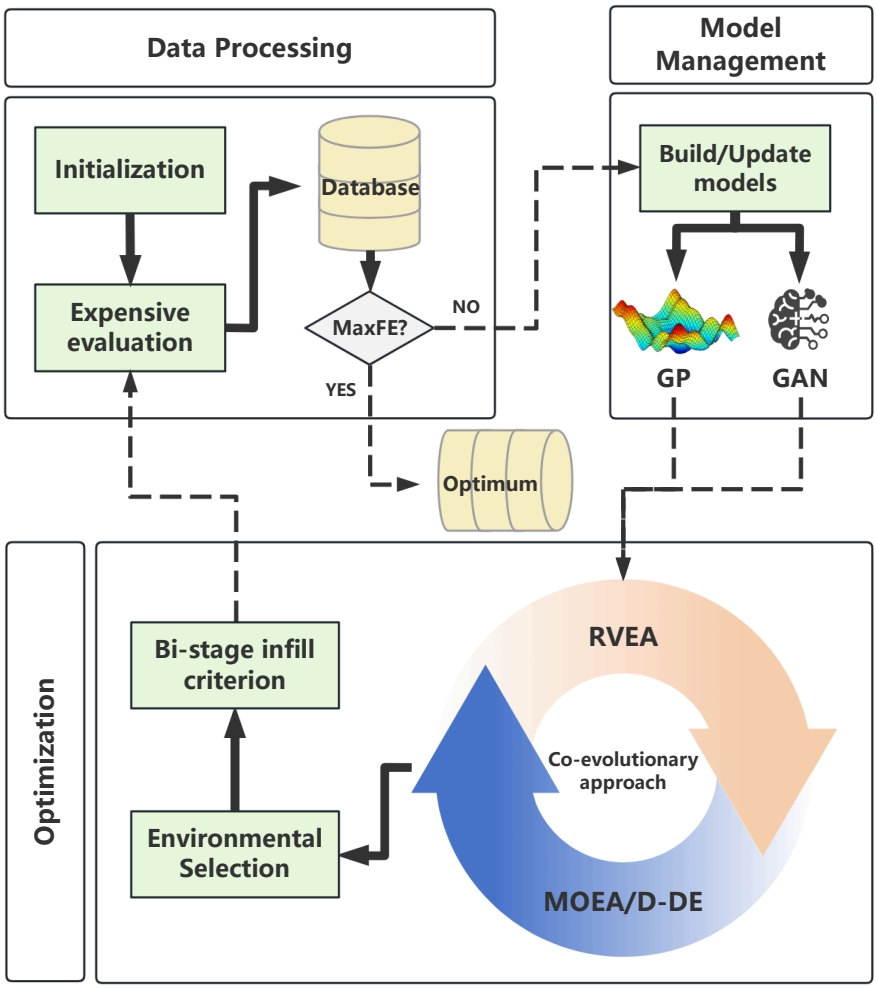


Figure 2. The flowchart of the proposed `CoEGAN-BO`.

Subsequently, selection for the next generation is conducted using the procedure outlined in Algorithm 2, which continues until the predefined population iteration limit is achieved. Integral to the process is Algorithm 3, which selectively pinpoints the most promising subset of solutions and subjects them to expensive fitness evaluations.

Algorithm 2 Environmental selection strategy using GAN

Require: N (population size); $\omega_{\max}$ (maximum iterations of population evolution); O_1^{ω} (offspring of the first population); O_2^{ω} (offspring of the second population); GAN (trained generative adversarial network).
Ensure: $P^{\omega+1}$ (the next population).

while $\omega <= \omega_{\max}$ **do**
  Perform non-dominated sorting on O_1^{ω} and O_2^{ω};
  $PF1_1 \leftarrow$ Select individuals from the first Pareto front of O_1^{ω};
  $PF1_2 \leftarrow$ Select individuals from the first Pareto front of O_2^{ω};
  if $|NP| > N$ **then**
    $P^{\omega+1} \leftarrow$ Select top N individuals most likely to be authentic using GAN;
  else
    $\Lambda \leftarrow$ Select $N - |NP|$ individuals most likely to be authentic using GAN from the remaining candidates by discriminator prediction probability;
    $P^{\omega+1} \leftarrow NP \cup \Lambda$;
  end if
  $\omega \leftarrow \omega + 1$;
end while

Algorithm 3 Bi-stage infill sampling criterion

Require: K (number of clusters for infill sampling); N (population size); γ (bi-stage division threshold); FE_{max} (maximum number of fitness evaluations).
Ensure: Λ (candidate solutions for expensive re-evaluation).
1: Initialize set Λ as empty;
2: Perform K-means clustering on the solutions to form K clusters;
3: **for** $i = 1$ to K **do**
4: **if** $FE < \gamma \times FE_{max}$ **then**
5: Select the solution with the minimum L_p-norm within the cluster;
6: **else**
7: Calculate MD;
8: Select the solution with the maximum MD from these solutions in the cluster;
9: **end if**
10: Add the selected solution to set Λ;
11: **end for**

The iterative process of model refinement and solution evaluation via `CoEGAN-BO` strategically inserts new update points for expensive fitness evaluation. This systematic approach propels the algorithm towards identifying the optimal solution, ensuring that each update maximizes the efficacy of the expensive fitness evaluation process until the budget for such fitness evaluations is exhausted.

In the following sections, we will explore the details of the co-evolutionary algorithm and the new sampling criterion.

3.1. Environmental Selection Strategy Based on Generative Adversarial Network

For complex optimization challenges, one of the most efficacious approaches is to harness diverse heuristic search strategies to enhance the efficiency of identifying optimal solutions. By integrating additional search tactics and employing various operators, the rate of population convergence is expedited. Therefore, this study adopts a co-evolutionary guidance approach for populations to address such complex optimization issues. Our proposed algorithm operates on the basis of two distinct populations, and these populations evolve independently. Each population employs a specific optimizer, with one using MOEA/D-DE and the other utilizing RVEA. Evolution within each population is steered by disparate environment selection, crossover, and mutation techniques. The two populations have the same population size and generate different offspring to ensure the diversity of the offspring. Co-evolutionary dynamics between the two populations enhance the efficacy of the search process. Inter-population information exchange occurs via non-dominated solutions and those identified as promising by the GAN. The optimization process is guided by a secondary GAN-based screening that selects predictive samples that are more closely aligned with the distribution of the real sample (which is evaluated by the original expensive function) space. Both populations excel at identifying superior candidates within their local vicinities, thereby sustaining elevated search efficiency. The experimental results presented in Section 4.3 of this paper indicate that this search strategy not only effectively enhances population diversity but also improves the algorithm's search capability and optimization speed.

The detailed procedure of Algorithm 2, which describes an environmental selection strategy using the GAN, is outlined below. The algorithm operates on offspring populations O_1^ω and O_2^ω. Each iteration begins by performing non-dominated sorting on these offspring populations to identify the fittest individuals in the first Pareto front from each population.

The first Pareto fronts $PF1_1$ and $PF1_2$ are then merged into a single population NP. If the size of NP is greater than the desired population size N, the GAN's discriminative network is used to select the top N individuals that are most likely to be authentic. These

individuals are chosen based on their estimated probabilities of resembling authentic samples, thereby ensuring the reliability of the selected population.

In the case where the number of individuals in NP is less than N, additional individuals are chosen from the remaining candidates by the GAN to make up the shortfall. Once N individuals have been selected, they form the new parent population $P^{\omega+1}$ for the next generation.

This process iterates until the maximum number of generations $\omega_{\max}$ has been reached. The goal throughout is to maintain a population size of N, filled with individuals that exhibit high fitness and authenticity, thus facilitating effective convergence of the population.

The rationale behind integrating a GAN within Algorithm 2 for environment selection lies in the Discriminator's acute ability to discern the authenticity of samples, effectively evaluating the likelihood of a sample being representative of the real data distribution. By harnessing the Discriminator's judgments, the algorithm preferentially selects individuals that are most likely to be authentic. This method seeks to minimize the deviation in the optimization trajectory caused by potential prediction errors, directing the evolutionary process more precisely toward the desired outcomes. Furthermore, this approach is robust even in scenarios like high-dimensional expensive multi-objective optimization problems (HEMaOPs), where sparse samples could potentially diminish model accuracy. The GAN provides valuable secondary information that can be utilized to refine the selection process, thus bolstering the algorithm's ability to navigate the optimization landscape effectively.

Figure 3 illustrates the environmental selection process using a GAN model within an evolutionary algorithm framework. The overall goal of this process, as depicted in the diagram, is to maintain a population of individuals that not only exhibit high fitness levels but also align well with the authentic data distribution as determined by the GAN's Discriminator. This approach seeks to enhance the evolutionary algorithm's ability to converge towards optimal solutions effectively.

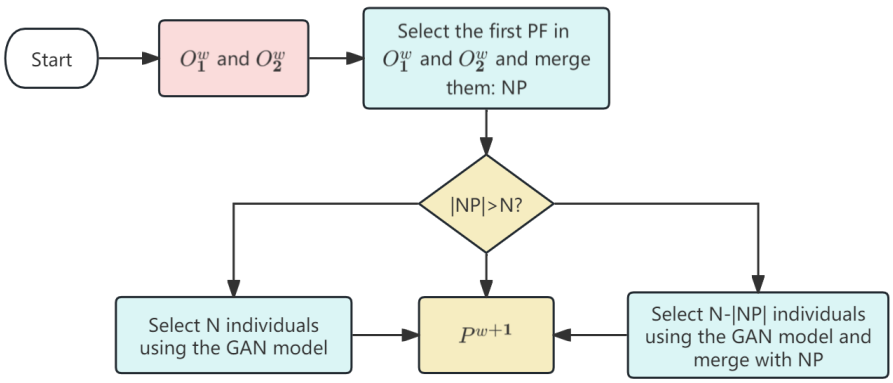


Figure 3. The flowchart of the environment selection strategy based on the GAN.

3.2. Bi-Stage Infill Criterion

Our method introduces a bi-stage infill criterion that utilizes both the L_p-norm distance and $max - min$ distances to identify more promising candidate solutions. This criterion efficiently leverages the surrogate model's feedback. During the optimization process, sampling is performed according to the infill criterion, with the surrogate model undergoing continuous updates and optimization until the maximum number of evaluations is reached. The selection of individuals for costly fitness evaluations is pivotal within BO. The efficacy of this selection bears directly on the precision of the model post-update. The inherent uncertainty in model predictions sways the algorithm's search trajectory, thereby influencing the solution quality of the algorithm. In this study, `CoEGAN-BO` introduces an infill sampling criterion propelled by both the L_p-norm distance [49] and max–min

distances [50]. This criterion meticulously parses the feedback from the surrogate model to pinpoint more propitious samples. The proposed infill criterion is detailed in Algorithm 3. K-means [51] clustering arranges the candidate solutions into K groups, placing similar individuals within the same cluster. In the first stage, if the number of costly evaluations is below the threshold $\gamma \times FE_{max}$, the candidate solution with the minimum L_p-norm distance within each cluster is selected for re-evaluation. In the second stage, after the threshold is met, we calculate the minimum Euclidean distance (MD) in the objective space between each candidate and those already expensively evaluated. The candidate with the maximum MD from the evaluated set in each cluster is then selected.

The impact of dimensionality on distance metrics is scrutinized [49]. In high-dimensional spaces, the disparity between the most distant and nearest individuals to the search probe is negligible, indicating the erosion of distance functions' effectiveness as dimensionality escalates. Compared to the Manhattan distance (L_1-norm) [52] and the Euclidean distance (L_2-norm) [53], the Lp-norm distance demonstrates superior performance in high-dimensional spaces. A static distance may not be universally applicable across optimization problems with varying objective counts. Hence, experimentally, $p = 1/m$ is set [40], where m is the number of objectives. This method takes into account the impact of dimensionality on distance metrics, which becomes particularly pertinent in high-dimensional spaces, where traditional distance measures may lose effectiveness. An experimental L_p-norm with $p = 1/m$, where m is the number of objectives, is utilized to maintain the efficacy of distance measurement across different objective counts. Moreover, in the early stages of optimization in high-dimensional spaces, when candidate solutions are sparse, relying solely on the max–min distance criterion can reduce the population's local search capability. Our bi-stage infill criterion addresses this issue.

The strategy enables potential solutions to oscillate between regions, circumventing local optima. Each update point is selected with the utmost deliberation, thoroughly assimilating the surrogate model's feedback. By enabling the selection of multiple promising candidates near the current best solution and in regions with fewer samples, this method significantly improves both the accuracy of the surrogate model and the efficiency of the optimization process. It is well-suited for surrogate models that lack variance estimation, overcoming the limitations of a single sampling criterion and accelerating convergence.

3.3. Computational Complexity Analysis

The proposed CoEGAN-BO algorithm integrates multiple computational components, whose complexity is analyzed as follows:

The algorithm sequentially performs GP and GAN training in each generation. The computational complexity for training these two models is $O(n^2 \cdot d)$ for the GP training and $O(E \cdot n \cdot M^2)$ for the GAN, where n is the population size, d is the dimension of the decision space, E is the number of training epochs, and M is the number of objectives.

Following the surrogate modeling, the hybrid optimization combining MOEA/D-DE and RVEA is executed. The computational complexity of the hybrid optimization is $O(N \cdot T \cdot d \cdot \omega_{max} + N_1^2 \cdot M)$, where N is the number of weight vectors in MOEA/D-DE, T is the neighborhood size, ω_{max} is the maximum iterations, and N_1 is the RVEA population size.

Finally, K-means clustering is applied for solution selection with complexity $O(K_e \cdot Q \cdot d)$, where K_e is the number of clusters and $Q \approx O(N)$ is the candidate solution size.

The main optimization loop runs for T generations, where T is determined by the function evaluation budget. Therefore, the overall computational complexity of CoEGAN-BO is

$$T \cdot \left[O(n^2 \cdot d + E \cdot n \cdot M^2) + O(N \cdot T \cdot d \cdot \omega_{max} + N_1^2 \cdot M) + O(K_e \cdot N \cdot d) \right] \tag{6}$$

Considering that $N, N_1 \approx O(n)$ and other parameters are constants, the overall complexity can be simplified to

$$T \cdot O(n^2 \cdot (d + M)) \tag{7}$$

4. Experimental Evaluation

This section assesses the effectiveness of our proposed method, referred to as `CoEGAN-BO`, in the context of many-objective optimization problems. An empirical comparison is made with notable SAEAs such as MOEA/D-EGO [31], K-RVEA [24], KTA2 [32], REMO [54], and EDN-ARMOEA [55]. All of the referenced algorithms have been realized within the PlatEMO framework [56]. Our experimental setup included instances from the DTLZ and MaF suite targeting 3, 4, 6, 8, and 10 objectives. Across all DTLZ cases, we set the count of decision variables to 50 and 100, while the MaF problem was on 50 dimensions. Importantly, every experimental setup underwent 20 independent runs.

To evaluate the performance of the SAEAs under comparison, we adopted the Inverted Generational Distance (IGD) metric [57], ensuring a comprehensive assessment of both convergence and diversity. Furthermore, we applied the Wilcoxon rank-sum (WRS) test for a statistical comparison between `CoEGAN-BO` and the other four SAEAs, establishing a significance threshold of 0.05. In our results, the symbols "+", "−", and "=" symbolize the statistical significance in superiority, inferiority, or approximate equivalence of a given algorithm relative to `CoEGAN-BO`, respectively.

Our GAN architecture employs lightweight feed-forward neural networks for both the Generator and Discriminator, each containing three hidden layers with 5 neurons, epoch = 100, learning rate = 0.001, and batch size equal to the number of training samples, and using the cross-entropy loss function. The number of actual samples and predicted samples is equal during training, and normalization is performed before training. This makes the GAN efficient in training, and it achieves good performance.

4.1. Experimental Settings

In our comparative experiment, all of the compared algorithms shared common parameter settings, which are outlined as follows:

1. The initial population size was uniformly set to $100 + D/10$ for consistency across all algorithms;
2. The maximum allowable number of expensive function evaluations was limited to 300, ensuring a level playing field for evaluation;
3. Parameters governing reproduction, including crossover and mutation, had values of $pc = 1.0$, $\eta c = 20$, $pm = 1.0$, and $\eta m = 20$ across all algorithms;
4. To maintain uniformity, the maximum number of generations before updating the Gaussian process was uniformly established at $w_{\max} = 20$ for all considered algorithms.

4.2. Sensitivity Analysis Based on Bi-Stage Division Threshold

In the experimental section, we emphasize the significance of the parameter γ, which is integral to the bi-stage filling criterion detailed in Algorithm 3. The selection of the γ value is crucial, as it directly influences the efficacy of `CoEGAN-BO`. For the purpose of optimizing and refining the parameters of `CoEGAN-BO`, we chose DTLZ1 as the principal test case. DTLZ1 is renowned for its testing complexity, particularly due to the slow convergence that it typically presents. Success in optimizing DTLZ1 suggests that `CoEGAN-BO` is likely to perform well across a diverse array of test scenarios. To this end, we rigorously examined the sensitivity of γ within the `CoEGAN-BO` by conducting a series of targeted experiments on the DTLZ1 benchmark.

To ascertain the impact of varying γ values on performance, we tested a spectrum of γ values within the set $\gamma \in 1/3, 0.5, 0.7, 0.9, 1$. The comparative results for the DTLZ1 problems with 3, 4, 6, 8, and 10 objectives are depicted in Figure 4. The results indicate that CoEGAN-BO performs optimally when γ is set to 0.5, striking an effective balance between convergence and diversity.

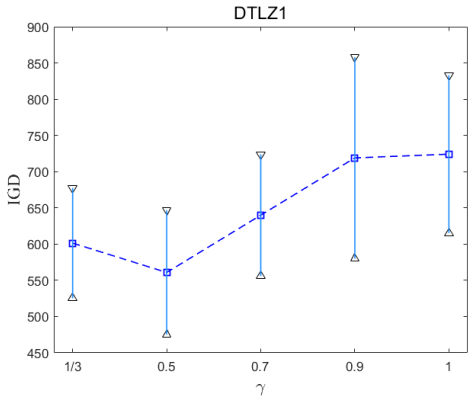


Figure 4. Average IGD profile plots of parameter γ for different parameter values on a DTLZ1 problem with 3 objectives.

The value $\gamma = 1/3$ represents the lower bound of the parameter's range. At this setting, the model relies exclusively on the max–min distance for sample selection. Despite enhancing the diversity of solutions, this setting leads to suboptimal results due to compromised convergence.

At the other extreme, $\gamma = 1$ denotes the upper limit and involves the exclusive use of the Lp-norm distance for sample selection. However, this setting yields results that are inferior, even when compared to the $\gamma = 1/3$ setting. The likely reason for this decline in performance is the diminished efficacy of the Lp-norm distance in high-dimensional spaces.

A discernible trend is observed, where performance gradually deteriorates as γ increases from 0.5 to 1. A larger γ value potentially causes the algorithm to converge more slowly. Consequently, we adopted $\gamma = 0.5$ as the default value for all test instances, given its general effectiveness.

4.3. Impact of GAN-Based Environmental Selection

In our comparative analysis, we examine two distinct variants of CoEGAN-BO to isolate the impacts of different environmental selection strategies. The variant named CoEGAN-BO(N_p) adopts only the non-dominated sorting technique, eschewing the proposed GAN-based environmental selection strategy. Conversely, CoEGAN-BO(GAN) exclusively utilizes the generative adversarial network, bypassing the traditional environmental selection approach. Table 2 shows the average Inverted Generational Distance (IGD) results for both CoEGAN-BO(N_p) and CoEGAN-BO(GAN), as well as the standard CoEGAN-BO, with the best-performing results emphasized.

Table 2. The statistical results of IGD values obtained after 300 function evaluations. The best results are highlighted in bold.

Problem	CoEGAN-BO (GAN)	CoEGAN-BO (N_p)	CoEGAN-BO
DTLZ1	6.45×10^{2}	6.01×10^{2}	$\mathbf{5.61 \times 10^{2}}$
DTLZ2	1.62×10^{0}	1.67×10^{0}	$\mathbf{1.57 \times 10^{0}}$
DTLZ3	1.24×10^{3}	1.32×10^{3}	$\mathbf{1.20 \times 10^{3}}$
DTLZ5	1.65×10^{0}	1.62×10^{0}	$\mathbf{1.49 \times 10^{0}}$
DTLZ6	$\mathbf{2.20 \times 10^{1}}$	2.21×10^{1}	2.24×10^{1}
DTLZ7	9.19×10^{0}	9.09×10^{0}	$\mathbf{8.59 \times 10^{0}}$

The empirical results indicate that the standard CoEGAN-BO, which integrates both non-dominated sorting and the GAN-based strategy, outperforms these variants on the DTLZ problems. This finding highlights the algorithm's significant advantage in effectively identifying non-dominated solutions, which are instrumental in generating the next population.

Therefore, it is effective to propose the environment selection strategy based on a GAN to select effective non-dominant solutions. Firstly, the CoEGAN-BO(N_p) version exclusively uses the non-dominated sorting technique, foregoing the proposed GAN-based environmental selection strategy. Non-dominated sorting is an important tool in traditional multi-objective and many-objective optimization problems, as it helps maintain solution diversity and guides the population towards Pareto front evolution. However, as the dimensionality of the problem increases and the limitations of surrogate models become apparent, non-dominated sorting may amplify these inadequacies, because it relies entirely on model predictions to evaluate the fitness of individuals. On the other hand, the CoEGAN-BO(GAN) version combines the advantages of non-dominated sorting with the GAN-based environmental selection strategy. In high-dimensional problems, the constraints of surrogate models imply that using model predictions for non-dominated sorting could lead to cumulative errors affecting the direction of population search and could even cause search bias. The introduction of GANs becomes particularly important at this juncture, as they improve the selection process by learning to differentiate between real samples and those generated by the surrogate models. In other words, GANs help CoEGAN-BO reduce the search bias that could arise due to model inaccuracies.

4.4. Comparison Results on DTLZ Test Problems

To discuss the performance of the proposed algorithm, CoEGAN-BO was compared with several other algorithms, including K-RVEA, MOEA/D-EGO, KTA2, REMO, and EDN-ARMOEA, on a set of 30 problems with varying numbers of objectives (3, 4, 6, 8, and 10). The comparison was based on the Inverted Generational Distance (IGD) metric, with the results summarized in Tables 3 and 4 for 50-dimensional and 100-dimensional problems, respectively, derived from 20 independent runs. A Wilcoxon rank-sum (WRS) test was performed to statistically compare the performance of CoEGAN-BO against the other four algorithms. CoEGAN-BO consistently achieved the best IGD values among the 30 test instances for both 50- and 100-dimensional problems, with the highest number of best results (25 out of 30), followed by K-RVEA and KTA2. MOEA/D-EGO and ParEGO did not secure the best results on the DTLZ test suite.

DTLZ1 and DTLZ3 are known for their slow convergence characteristics. DTLZ1 has a linear Pareto-optimal front, while DTLZ3 challenges an algorithm's ability to converge to the global Pareto front. The other algorithms may struggle with DTLZ1 and DTLZ3 due to these properties. However, CoEGAN-BO outperforms the competition on these problems, with REMO and MOEA/D-EGO following behind. DTLZ2 evaluates an algorithm's computational power as the number of objectives increases. CoEGAN-BO shows strong performance on DTLZ2, particularly for problems with 8 and 10 objectives, highlighting the importance of maintaining diversity for this problem. The other algorithms also demonstrate satisfactory results for DTLZ2.

DTLZ5 and DTLZ6 test an algorithm's ability to converge to a curved Pareto front, with DTLZ6 presenting a more complex challenge than DTLZ5. Overall, CoEGAN-BO outperforms the other algorithms for DTLZ1-6, while KTA2 shows the weakest performance, possibly due to its convergence and diversity mechanisms.

Table 3. Statistical IGD results of the four compared algorithms and CoEGAN-BO on 30 DTLZ test instances by Wilcoxon signed-rank test (the significance level is 0.05). The best results for each instance of 50-dimensional problems are highlighted in bold.

Problem	M	K-RVEA	MOEA/D-EGO	KTA2	REMO	EDN-ARMOEA	CoEGAN-BO
DTLZ1	3	1.1630×10^{3} (7.89×10^{1}) −	9.8024×10^{2} (8.76×10^{1}) −	1.1909×10^{3} (1.30×10^{2}) −	8.9557×10^{2} (8.73×10^{1}) −	1.1999×10^{3} (5.50×10^{1}) −	**5.6089×10^{2} (8.58×10^{1})**
	4	1.0674×10^{3} (5.90×10^{1}) −	8.7119×10^{2} (5.75×10^{1}) −	9.6466×10^{2} (1.63×10^{2}) −	8.8421×10^{2} (8.66×10^{1}) −	1.0493×10^{3} (6.44×10^{1}) −	**5.5394×10^{2} (1.22×10^{2})**
	6	9.0109×10^{2} (7.09×10^{1}) −	7.7411×10^{2} (6.08×10^{1}) −	8.0940×10^{2} (3.22×10^{1}) −	7.3680×10^{2} (7.20×10^{1}) −	9.0269×10^{2} (6.09×10^{1}) −	**5.3822×10^{2} (9.80×10^{1})**
	8	7.9409×10^{2} (7.25×10^{1}) −	7.0443×10^{2} (5.89×10^{1}) −	7.6470×10^{2} (1.41×10^{2}) −	6.6895×10^{2} (7.95×10^{1}) −	8.4174×10^{2} (4.86×10^{1}) −	**5.4340×10^{2} (9.26×10^{1})**
	10	6.5519×10^{2} (8.55×10^{1}) −	6.0232×10^{2} (6.57×10^{1}) −	7.9799×10^{2} (9.15×10^{1}) −	5.6624×10^{2} (7.74×10^{1}) −	7.6990×10^{2} (5.12×10^{1}) −	**4.1843×10^{2} (1.20×10^{2})**
DTLZ2	3	3.0168×10^{0} (1.26×10^{-1}) −	2.0717×10^{0} (2.20×10^{-1}) −	2.5237×10^{0} (3.08×10^{-1}) −	1.6689×10^{0} (2.74×10^{-1}) =	2.7616×10^{0} (1.19×10^{-1}) −	**1.5716×10^{0} (4.69×10^{-1})**
	4	3.0025×10^{0} (1.45×10^{-1}) −	2.4457×10^{0} (2.16×10^{-1}) −	2.6122×10^{0} (3.66×10^{-1}) −	2.1026×10^{0} (2.43×10^{-1}) −	2.7210×10^{0} (1.26×10^{-1}) −	**1.5617×10^{0} (2.91×10^{-1})**
	6	2.9962×10^{0} (1.18×10^{-1}) −	2.6470×10^{0} (2.11×10^{-1}) −	2.5842×10^{0} (3.72×10^{-1}) −	2.1645×10^{0} (2.02×10^{-1}) −	2.7891×10^{0} (1.09×10^{-1}) −	**1.5619×10^{0} (2.61×10^{-1})**
	8	2.8051×10^{0} (1.87×10^{-1}) −	2.6686×10^{0} (1.31×10^{-1}) −	2.7268×10^{0} (3.22×10^{-1}) −	2.0794×10^{0} (1.65×10^{-1}) −	2.7682×10^{0} (9.51×10^{-2}) −	**1.5613×10^{0} (2.98×10^{-1})**
	10	2.7647×10^{0} (1.67×10^{-1}) −	2.6465×10^{0} (2.01×10^{-1}) −	2.5056×10^{0} (2.99×10^{-1}) −	2.0173×10^{0} (1.99×10^{-1}) −	2.6952×10^{0} (9.71×10^{-2}) −	**1.5378×10^{0} (1.87×10^{-1})**
DTLZ3	3	3.5193×10^{3} (2.18×10^{2}) −	3.0694×10^{3} (2.69×10^{2}) −	3.2725×10^{3} (6.62×10^{2}) −	2.8104×10^{3} (2.54×10^{2}) −	3.7080×10^{3} (1.52×10^{2}) −	**1.1958×10^{3} (2.25×10^{2})**
	4	3.3967×10^{3} (2.32×10^{2}) −	3.1343×10^{3} (2.51×10^{2}) −	2.7809×10^{3} (9.35×10^{2}) −	2.8618×10^{3} (2.67×10^{2}) −	3.6016×10^{3} (1.57×10^{2}) −	**1.1265×10^{3} (1.79×10^{2})**
	6	3.1608×10^{3} (2.26×10^{2}) −	3.1859×10^{3} (2.10×10^{2}) −	1.7952×10^{3} (6.63×10^{2}) −	2.6210×10^{3} (2.79×10^{2}) −	3.4518×10^{3} (1.41×10^{2}) −	**9.5895×10^{2} (2.79×10^{2})**
	8	2.9514×10^{3} (1.95×10^{2}) −	3.0723×10^{3} (2.00×10^{2}) −	1.8546×10^{3} (6.86×10^{2}) −	2.6058×10^{3} (2.66×10^{2}) −	3.2543×10^{3} (1.27×10^{2}) −	**9.5564×10^{2} (2.43×10^{2})**
	10	2.7507×10^{3} (2.14×10^{2}) −	2.9255×10^{3} (2.29×10^{2}) −	2.0252×10^{3} (7.83×10^{2}) −	2.3794×10^{3} (2.73×10^{2}) −	3.0955×10^{3} (1.63×10^{2}) −	**9.8153×10^{2} (2.27×10^{2})**
DTLZ5	3	2.8829×10^{0} (1.62×10^{-1}) −	2.1090×10^{0} (1.75×10^{-1}) −	2.4662×10^{0} (4.68×10^{-1}) −	1.7606×10^{0} (1.59×10^{-1}) −	2.6462×10^{0} (1.05×10^{-1}) −	**1.4994×10^{0} (3.39×10^{-1})**
	4	2.8584×10^{0} (1.49×10^{-1}) −	2.3209×10^{0} (1.82×10^{-1}) −	2.4170×10^{0} (4.70×10^{-1}) −	2.0165×10^{0} (3.34×10^{-1}) −	2.6011×10^{0} (1.22×10^{-1}) −	**1.4336×10^{0} (3.57×10^{-1})**
	6	2.6142×10^{0} (1.79×10^{-1}) −	2.5474×10^{0} (1.50×10^{-1}) −	2.5377×10^{0} (3.62×10^{-1}) −	1.8241×10^{0} (3.11×10^{-1}) −	2.4741×10^{0} (1.40×10^{-1}) −	**1.2413×10^{0} (2.84×10^{-1})**
	8	2.3461×10^{0} (1.33×10^{-1}) −	2.4404×10^{0} (1.21×10^{-1}) −	2.2628×10^{0} (1.98×10^{-1}) −	1.6780×10^{0} (1.97×10^{-1}) −	2.3209×10^{0} (9.32×10^{-2}) −	**1.1171×10^{0} (2.92×10^{-1})**
	10	2.1783×10^{0} (1.38×10^{-1}) −	2.3175×10^{0} (1.63×10^{-1}) −	2.0085×10^{0} (3.10×10^{-1}) −	1.6264×10^{0} (2.26×10^{-1}) −	2.2470×10^{0} (6.59×10^{-2}) −	**9.1949×10^{-1} (2.55×10^{-1})**
DTLZ6	3	3.7862×10^{1} (6.27×10^{-1}) −	3.8309×10^{1} (9.45×10^{-1}) −	2.9367×10^{1} (3.59×10^{0}) −	3.8616×10^{1} (1.44×10^{0}) −	4.1882×10^{1} (2.91×10^{-1}) −	**2.2441×10^{1} (2.59×10^{0})**
	4	3.7501×10^{1} (1.23×10^{0}) −	3.8952×10^{1} (7.66×10^{-1}) −	2.8591×10^{1} (3.71×10^{0}) −	3.8753×10^{1} (1.16×10^{0}) −	4.1156×10^{1} (1.09×10^{-1}) −	**2.5876×10^{1} (2.89×10^{0})**
	6	3.6577×10^{1} (7.35×10^{-1}) −	3.8246×10^{1} (4.90×10^{-1}) −	2.7463×10^{1} (2.95×10^{0}) −	3.7040×10^{1} (1.10×10^{0}) −	3.9301×10^{1} (2.42×10^{-1}) −	**2.4404×10^{1} (1.82×10^{0})**
	8	3.5115×10^{1} (8.54×10^{-1}) −	3.6498×10^{1} (7.45×10^{-1}) −	2.5257×10^{1} (4.64×10^{0}) −	3.5939×10^{1} (9.76×10^{-1}) −	3.7511×10^{1} (2.38×10^{-1}) −	**2.2870×10^{1} (2.42×10^{0})**
	10	3.3429×10^{1} (1.01×10^{0}) −	3.4783×10^{1} (5.58×10^{-1}) −	2.1562×10^{1} (3.07×10^{0}) =	3.4070×10^{1} (9.54×10^{-1}) −	3.5672×10^{1} (2.05×10^{-1}) −	**2.0519×10^{1} (1.72×10^{0})**
DTLZ7	3	**5.4355×10^{-1} (7.98×10^{-2}) +**	5.6601×10^{0} (7.65×10^{-1}) +	9.3943×10^{0} (8.12×10^{-1}) −	5.4656×10^{0} (9.92×10^{-1}) +	9.2824×10^{0} (5.35×10^{-1}) −	8.5884×10^{0} (7.91×10^{-1})
	4	**8.6090×10^{-1} (9.32×10^{-2}) +**	9.1196×10^{0} (1.44×10^{0}) +	1.1873×10^{1} (9.73×10^{-1}) =	8.9490×10^{0} (8.38×10^{-1}) +	1.2503×10^{1} (5.41×10^{-1}) −	1.1532×10^{1} (1.52×10^{0})
	6	**1.4746×10^{0} (3.04×10^{-1}) +**	1.5391×10^{1} (1.75×10^{0}) +	1.9032×10^{1} (1.66×10^{0}) −	1.5532×10^{1} (1.36×10^{0}) +	1.9573×10^{1} (9.33×10^{-1}) −	1.7448×10^{1} (2.20×10^{0})
	8	**2.7182×10^{0} (1.20×10^{0}) +**	2.2466×10^{1} (2.36×10^{0}) =	2.5169×10^{1} (2.93×10^{0}) −	2.2078×10^{1} (1.84×10^{0}) =	2.6114×10^{1} (1.49×10^{0}) −	2.3463×10^{1} (2.54×10^{0})
	10	**5.2089×10^{0} (1.87×10^{0}) +**	2.8486×10^{1} (3.95×10^{0}) =	3.2039×10^{1} (2.65×10^{0}) −	2.7666×10^{1} (2.82×10^{0}) =	3.2454×10^{1} (2.07×10^{0}) −	2.9126×10^{1} (4.30×10^{0})
+/−/=		5/25/0	3/25/2	0/28/2	3/24/3	0/29/1	

Table 4. Statistical IGD results of the four compared algorithms and CoEGAN-BO on 30 DTLZ test instances by Wilcoxon signed-rank test (the significance level is 0.05). The best results for each instance are highlighted for 100-dimensional problems in bold.

Problem	M	K-RVEA	MOEA/D-EGO	KTA2	REMO	EDN-ARMOEA	CoEGAN-BO
DTLZ1	3	2.7400×10^3 (9.95×10^1) −	2.3905×10^3 (1.58×10^2) −	2.7076×10^3 (1.33×10^2) −	2.2310×10^3 (9.09×10^1) −	2.6023×10^3 (5.46×10^1) −	**1.3536×10^3 (4.04×10^2)**
	4	2.4209×10^3 (1.10×10^2) −	2.1237×10^3 (1.25×10^2) −	2.4253×10^3 (1.00×10^2) −	2.0944×10^3 (1.46×10^2) −	2.2831×10^3 (6.71×10^1) −	**1.3482×10^3 (2.03×10^2)**
	6	1.9867×10^3 (6.60×10^1) −	1.8757×10^3 (1.09×10^2) −	2.0993×10^3 (1.35×10^2) −	1.8480×10^3 (1.15×10^2) −	1.9940×10^3 (1.07×10^2) −	**1.2423×10^3 (1.95×10^2)**
	8	1.8175×10^3 (1.09×10^2) −	1.7335×10^3 (9.42×10^1) −	2.0243×10^3 (1.15×10^2) −	1.7154×10^3 (1.42×10^2) −	1.8348×10^3 (8.79×10^1) −	**1.0897×10^3 (3.27×10^2)**
	10	1.7690×10^3 (1.06×10^2) −	1.5889×10^3 (1.37×10^2) −	1.9312×10^3 (1.48×10^2) −	1.6480×10^3 (1.27×10^2) −	1.7663×10^3 (9.13×10^1) −	**9.4570×10^2 (2.76×10^2)**
DTLZ2	3	6.5568×10^0 (2.95×10^{-1}) −	5.5056×10^0 (3.00×10^{-1}) −	6.4772×10^0 (3.25×10^{-1}) −	4.9357×10^0 (4.93×10^{-1}) =	6.1443×10^0 (1.78×10^{-1}) −	**4.6614×10^0 (9.00×10^{-1})**
	4	6.4980×10^0 (3.25×10^{-1}) −	6.0500×10^0 (3.00×10^{-1}) −	6.5618×10^0 (2.50×10^{-1}) −	5.3265×10^0 (3.68×10^{-1}) −	6.0854×10^0 (2.08×10^{-1}) −	**4.7588×10^0 (8.49×10^{-1})**
	6	6.5986×10^0 (1.79×10^{-1}) −	6.1751×10^0 (2.37×10^{-1}) −	6.5517×10^0 (2.89×10^{-1}) −	5.1044×10^0 (3.36×10^{-1}) −	6.1155×10^0 (1.56×10^{-1}) −	**4.1289×10^0 (1.06×10^0)**
	8	6.5548×10^0 (2.69×10^{-1}) −	6.1776×10^0 (3.83×10^{-1}) −	6.5600×10^0 (2.36×10^{-1}) −	5.2862×10^0 (4.16×10^{-1}) −	6.0762×10^0 (1.99×10^{-1}) −	**4.5706×10^0 (9.46×10^{-1})**
	10	6.3792×10^0 (3.22×10^{-1}) −	6.0196×10^0 (2.06×10^{-1}) −	6.4229×10^0 (3.74×10^{-1}) −	5.1812×10^0 (3.58×10^{-1}) −	6.0105×10^0 (1.60×10^{-1}) −	**4.4632×10^0 (9.27×10^{-1})**
DTLZ3	3	8.3362×10^3 (3.61×10^2) −	7.8510×10^3 (2.23×10^2) −	8.2749×10^3 (4.88×10^2) −	7.1383×10^3 (4.10×10^2) −	8.4466×10^3 (1.83×10^2) −	**2.6985×10^3 (1.36×10^2)**
	4	8.3606×10^3 (2.67×10^2) −	8.0465×10^3 (3.08×10^2) −	8.1113×10^3 (9.15×10^2) −	7.3300×10^3 (4.70×10^2) −	8.2661×10^3 (2.00×10^2) −	**2.4775×10^3 (2.79×10^2)**
	6	8.2088×10^3 (2.68×10^2) −	8.0250×10^3 (3.06×10^2) −	7.4869×10^3 (9.23×10^2) −	7.1297×10^3 (2.84×10^2) −	7.9735×10^3 (2.59×10^2) −	**2.1852×10^3 (5.92×10^2)**
	8	8.0873×10^3 (2.09×10^2) −	7.8899×10^3 (2.56×10^2) −	6.8763×10^3 (1.58×10^3) −	7.0811×10^3 (5.28×10^2) −	7.8185×10^3 (2.58×10^2) −	**1.9900×10^3 (8.30×10^2)**
	10	7.7497×10^3 (3.33×10^2) −	7.5194×10^3 (3.35×10^2) −	6.8331×10^3 (1.51×10^3) −	6.8932×10^3 (4.77×10^2) −	7.7091×10^3 (2.35×10^2) −	**2.3608×10^3 (7.12×10^2)**
DTLZ5	3	6.4983×10^0 (2.24×10^{-1}) −	5.5380×10^0 (3.18×10^{-1}) −	6.4745×10^0 (2.86×10^{-1}) −	4.6592×10^0 (4.69×10^{-1}) =	5.9920×10^0 (1.61×10^{-1}) −	**4.4818×10^0 (9.80×10^{-1})**
	4	6.5053×10^0 (2.27×10^{-1}) −	5.6927×10^0 (3.17×10^{-1}) −	6.3412×10^0 (3.41×10^{-1}) −	5.4709×10^0 (4.64×10^{-1}) −	5.9715×10^0 (1.24×10^{-1}) −	**4.5460×10^0 (1.17×10^0)**
	6	6.3855×10^0 (3.00×10^{-1}) −	5.9243×10^0 (2.10×10^{-1}) −	6.3203×10^0 (2.52×10^{-1}) −	5.1026×10^0 (4.31×10^{-1}) −	5.8540×10^0 (2.80×10^{-1}) −	**4.1627×10^0 (1.03×10^0)**
	8	6.1416×10^0 (2.44×10^{-1}) −	5.9718×10^0 (3.16×10^{-1}) −	6.0392×10^0 (5.27×10^{-1}) −	5.0663×10^0 (3.82×10^{-1}) −	5.7729×10^0 (1.65×10^{-1}) −	**3.7756×10^0 (9.71×10^{-1})**
	10	6.1396×10^0 (1.96×10^{-1}) −	5.9373×10^0 (2.38×10^{-1}) −	5.7663×10^0 (5.84×10^{-1}) −	4.7344×10^0 (4.74×10^{-1}) −	5.6017×10^0 (1.35×10^{-1}) −	**3.9932×10^0 (1.12×10^0)**
DTLZ6	3	8.2975×10^1 (1.05×10^0) −	8.4323×10^1 (8.88×10^{-1}) −	7.5523×10^1 (6.66×10^0) −	8.2203×10^1 (1.63×10^0) −	8.6402×10^1 (2.76×10^{-1}) −	**4.6775×10^1 (4.12×10^0)**
	4	8.2596×10^1 (8.57×10^{-1}) −	8.4379×10^1 (6.46×10^{-1}) −	7.8968×10^1 (5.13×10^0) −	8.3351×10^1 (9.80×10^{-1}) −	8.5521×10^1 (3.77×10^{-1}) −	**4.7666×10^1 (4.08×10^0)**
	6	8.1410×10^1 (7.52×10^{-1}) −	8.2948×10^1 (5.82×10^{-1}) −	7.5985×10^1 (6.10×10^0) −	8.1505×10^1 (1.23×10^0) −	8.3769×10^1 (2.57×10^{-1}) −	**4.6297×10^1 (3.95×10^0)**
	8	8.0194×10^1 (8.38×10^{-1}) −	8.1350×10^1 (8.09×10^{-1}) −	7.5311×10^1 (6.77×10^0) −	7.9948×10^1 (1.04×10^0) −	8.1799×10^1 (3.99×10^{-1}) −	**4.4825×10^1 (4.25×10^0)**
	10	7.8081×10^1 (8.25×10^{-1}) −	7.9716×10^1 (6.16×10^{-1}) −	7.0222×10^1 (1.05×10^1) −	7.7877×10^1 (7.50×10^{-1}) −	8.0155×10^1 (2.29×10^{-1}) −	**4.3439×10^1 (3.77×10^0)**
DTLZ7	3	**4.2957×10^0 (7.69×10^{-1}) +**	9.4168×10^0 (4.29×10^{-1}) +	1.0478×10^1 (4.20×10^{-1}) −	6.9445×10^0 (4.93×10^{-1}) +	9.9256×10^0 (3.47×10^{-1}) =	1.0015×10^1 (6.16×10^{-1})
	4	**4.8264×10^0 (1.41×10^0) +**	1.2636×10^1 (9.37×10^{-1}) +	1.3982×10^1 (9.53×10^{-1}) =	1.0970×10^1 (7.67×10^{-1}) +	1.3366×10^1 (5.16×10^{-1}) =	1.3532×10^1 (9.69×10^{-1})
	6	**7.3726×10^0 (1.57×10^0) +**	2.0451×10^1 (1.10×10^0) =	2.1300×10^1 (1.02×10^0) =	1.9530×10^1 (7.65×10^{-1}) +	2.0726×10^1 (5.75×10^{-1}) =	2.1114×10^1 (1.35×10^0)
	8	**9.3152×10^0 (1.38×10^0) +**	2.8779×10^1 (1.34×10^0) =	2.9089×10^1 (1.53×10^0) =	2.7428×10^1 (1.09×10^0) +	2.8210×10^1 (1.08×10^0) =	2.8402×10^1 (1.46×10^0)
	10	**1.1131×10^1 (1.62×10^0) +**	3.5965×10^1 (1.51×10^0) =	3.6868×10^1 (1.21×10^0) −	3.4709×10^1 (1.19×10^0) +	3.5519×10^1 (1.13×10^0) =	3.6030×10^1 (1.31×10^0)
+/−/=		5/25/0	2/25/3	0/27/3	5/23/2	0/25/5	

CoEGAN-BO achieves significantly better results on the DTLZ test suite, especially on the simpler problems such as DTLZ2, DTLZ5, and DTLZ7, compared to the more challenging DTLZ1, DTLZ3, and DTLZ6. Its superior exploratory capabilities allow CoEGAN-BO to excel on these problems, achieving the best IGD values for DTLZ1-6. CoEGAN-BO and REMO rank as the top two algorithms for these problems. Furthermore, as the number of decision variables and objectives increases, CoEGAN-BO's performance continues to improve. For DTLZ7, K-RVEA achieves the best IGD value, while CoEGAN-BO ranks fourth.

Despite the challenges with problems like DTLZ1 and DTLZ3 that are difficult to converge, CoEGAN-BO manages to obtain the best convergence solution sets. However, it demonstrates poor diversity in solving DTLZ7, with its solutions tending to cluster in specific regions, which affects its ability to address problems with a discontinuous Pareto-optimal front.

4.5. Comparison Results on MaF Problems

We extended our evaluation to assess the performance of CoEGAN-BO on the complex ultra multi-objective optimization MaF benchmarks (Table 5). It is imperative to note that not all of the algorithms yielded results across the full spectrum of MaF instances. Specifically, algorithms such as MOEA/D-EGO and ParEGO, incorporated within the PlatEMO framework, faced challenges when confronting some of the more intricate MaF scenarios, resulting in incomplete datasets in our comparative analysis. We use "N/A" to indicate missing data.

MaF1 is designed to test an algorithm's ability to address inverted Pareto fronts (PFs). CoEGAN-BO exhibited superior performance on the 3-, 4-, 6-, and 8-objective variants of the MaF1 problem, but its efficacy diminished at the 10-objective level, where MOEA/D-EGO prevailed. MaF2, known for its irregularity and incomplete coverage of the objective space, saw CoEGAN-BO triumphing at 3 and 4 objectives, while MOEA/D-EGO outperformed the rest at 6, 8, and 10 objectives, likely due to its more effective convergence function in such contexts.

MaF3, characterized by its convex PF and multitude of local fronts, was another domain where CoEGAN-BO demonstrated best-in-class performance for 6-, 8-, and 10-objective settings. K-RVEA, however, stood out for the 3- and 4-objective cases. This may be attributed to CoEGAN-BO's environmental selection strategy, which adeptly tackles the convex PF challenges in many-objective problems.

MaF4's high multimodality, comprising numerous local PFs, tests an algorithm's ability to navigate through severe PF landscapes. Here, CoEGAN-BO proved to be a robust solver, particularly in the 8- and 10-objective scenarios, showcasing significant improvements over its peers. CoEGAN-BO emerged as the preeminent solver for MaF4, with no other algorithm matching its performance.

On MaF5, where each objective function is scaled to distinctively different ranges, CoEGAN-BO did not emerge as the top contender. This suboptimal performance might be due to the limitations imposed by its bi-stage infilling criterion, which hinders its ability to effectively scale diverse objective functions. K-RVEA demonstrated the best performance on MaF5's three objectives.

MaF6 features a degenerate PF and serves as a litmus test for an algorithm's proficiency in tackling such challenges. It is evident that CoEGAN-BO has a commendable capacity to address degenerate PFs effectively. MaF7, which evaluates an algorithm's prowess in managing disconnected PFs, highlighted K-RVEA as the superior solver, potentially owing to an environmental selection strategy that favors continuous local optima. In conclusion, K-RVEA, CoEGAN-BO, and REMO stand out as the leading algorithms for resolving the diverse range of challenges presented by the MaF suite.

Table 5. Statistical IGD results of the four compared algorithms and CoEGAN-BO on 35 MaF test instances by the Wilcoxon signed-rank test (the significance level is 0.05). The best results for each instance are highlighted for 100-dimensional problems in bold.

Problem	M	K-RVEA	MOEA/D-EGO	KTA2	REMO	EDN-ARMOEA	CoEGAN-BO
MaF1	3	3.0630×10^{0} (1.67×10^{-1}) −	2.0202×10^{0} (2.83×10^{-1}) −	2.5657×10^{0} (4.97×10^{-1}) −	1.7246×10^{0} (2.73×10^{-1}) −	3.0720×10^{0} (1.94×10^{-1}) −	**1.0977×10^{0} (1.90×10^{-1})**
	4	3.5592×10^{0} (3.55×10^{-1}) −	2.3167×10^{0} (4.56×10^{-1}) −	3.3756×10^{0} (7.06×10^{-1}) −	2.0812×10^{0} (3.14×10^{-1}) −	3.9069×10^{0} (2.59×10^{-1}) −	**1.4582×10^{0} (3.63×10^{-1})**
	6	4.4377×10^{0} (4.79×10^{-1}) −	2.8709×10^{0} (4.03×10^{-1}) −	4.6643×10^{0} (7.89×10^{-1}) −	2.6182×10^{0} (2.19×10^{-1}) =	5.1277×10^{0} (2.72×10^{-1}) −	**2.4861×10^{0} (5.48×10^{-1})**
	8	4.7617×10^{0} (6.96×10^{-1}) −	3.2560×10^{0} (6.56×10^{-1}) =	5.4752×10^{0} (7.56×10^{-1}) −	3.0219×10^{0} (4.76×10^{-1}) =	5.7178×10^{0} (3.20×10^{-1}) −	**3.2295×10^{0} (6.66×10^{-1})**
	10	5.0680×10^{0} (6.58×10^{-1}) −	**3.2540×10^{0} (4.78×10^{-1}) +**	6.5188×10^{0} (7.06×10^{-1}) −	3.0217×10^{0} (5.44×10^{-1}) +	6.1669×10^{0} (2.90×10^{-1}) −	3.6704×10^{0} (7.31×10^{-1})
MaF2	3	2.6263×10^{-1} (7.31×10^{-3}) −	2.0954×10^{-1} (1.12×10^{-2}) −	2.3592×10^{-1} (1.09×10^{-2}) −	2.0265×10^{-1} (1.65×10^{-2}) −	2.4247×10^{-1} (5.19×10^{-3}) −	**1.7090×10^{-1} (1.09×10^{-2})**
	4	2.2340×10^{-1} (7.35×10^{-3}) −	2.0269×10^{-1} (5.54×10^{-3}) −	2.1383×10^{-1} (8.81×10^{-3}) −	2.0096×10^{-1} (9.29×10^{-3}) −	2.0679×10^{-1} (3.01×10^{-3}) −	**1.9284×10^{-1} (9.25×10^{-3})**
	6	2.7449×10^{-1} (1.60×10^{-2}) =	**2.0713×10^{-1} (7.43×10^{-3}) +**	2.6282×10^{-1} (1.40×10^{-2}) +	2.7039×10^{-1} (1.58×10^{-2}) =	2.3129×10^{-1} (6.94×10^{-3}) +	2.7770×10^{-1} (1.27×10^{-2})
	8	3.1911×10^{-1} (1.68×10^{-2}) =	**2.2804×10^{-1} (8.30×10^{-3}) +**	3.1711×10^{-1} (2.36×10^{-2}) =	3.2468×10^{-1} (2.07×10^{-2}) =	2.7502×10^{-1} (1.15×10^{-2}) +	3.2753×10^{-1} (2.52×10^{-2})
	10	3.5788×10^{-1} (2.58×10^{-2}) =	**2.3443×10^{-1} (1.09×10^{-2}) +**	3.4783×10^{-1} (2.28×10^{-2}) =	3.6012×10^{-1} (2.22×10^{-2}) =	3.0043×10^{-1} (1.09×10^{-2}) +	3.5505×10^{-1} (2.44×10^{-2})
MaF3	3	**1.5968×10^{7} (1.45×10^{6}) +**	2.9779×10^{7} (5.41×10^{6}) −	N/A	1.6799×10^{7} (7.00×10^{6}) +	2.1325×10^{7} (2.21×10^{6}) =	2.0445×10^{7} (6.14×10^{6})
	4	**1.6168×10^{7} (2.10×10^{6}) +**	2.6638×10^{7} (5.80×10^{6}) −	N/A	1.5377×10^{7} (3.47×10^{6}) +	2.0561×10^{7} (2.90×10^{6}) =	1.9783×10^{7} (6.77×10^{6})
	6	1.5305×10^{7} (2.07×10^{6}) −	2.3244×10^{7} (5.64×10^{6}) −	N/A	1.8147×10^{7} (4.41×10^{6}) −	1.8280×10^{7} (2.29×10^{6}) −	**9.9529×10^{6} (5.80×10^{6})**
	8	1.3950×10^{7} (2.52×10^{6}) −	2.1341×10^{7} (4.55×10^{6}) −	N/A	1.8048×10^{7} (6.65×10^{6}) −	1.7721×10^{7} (2.91×10^{6}) −	**7.7932×10^{6} (4.69×10^{6})**
	10	1.2835×10^{7} (2.32×10^{6}) −	2.1700×10^{7} (3.85×10^{6}) −	N/A	1.5644×10^{7} (7.46×10^{6}) −	1.4193×10^{7} (2.76×10^{6}) −	**5.4190×10^{6} (2.50×10^{6})**
MaF4	3	1.1714×10^{4} (7.59×10^{2}) −	1.0856×10^{4} (1.12×10^{3}) −	N/A	8.7103×10^{3} (8.20×10^{2}) −	1.1913×10^{4} (5.38×10^{2}) −	**7.2791×10^{3} (1.83×10^{3})**
	4	2.4369×10^{4} (1.81×10^{3}) −	2.5217×10^{4} (1.86×10^{3}) −	N/A	1.8465×10^{4} (2.37×10^{3}) −	2.4869×10^{4} (1.58×10^{3}) −	**1.6300×10^{4} (3.33×10^{3})**
	6	9.2871×10^{4} (5.95×10^{3}) −	1.0140×10^{5} (9.25×10^{3}) −	N/A	7.0090×10^{4} (9.57×10^{3}) −	9.4063×10^{4} (6.03×10^{3}) −	**6.0758×10^{4} (1.34×10^{4})**
	8	3.5739×10^{5} (2.59×10^{4}) −	3.8744×10^{5} (3.11×10^{4}) −	N/A	2.6748×10^{5} (4.12×10^{4}) −	3.7234×10^{5} (1.79×10^{4}) −	**2.1231×10^{5} (4.17×10^{4})**
	10	1.2808×10^{6} (1.17×10^{5}) −	1.4273×10^{6} (1.25×10^{5}) −	N/A	9.7701×10^{5} (1.08×10^{5}) −	1.3940×10^{6} (7.71×10^{4}) −	**8.3542×10^{5} (1.60×10^{5})**
MaF5	3	9.1984×10^{0} (5.39×10^{-1}) =	9.5013×10^{0} (2.21×10^{0}) =	1.0741×10^{1} (2.91×10^{0}) =	**6.3104×10^{0} (1.97×10^{0}) +**	1.0056×10^{1} (9.10×10^{-1}) =	1.0048×10^{1} (2.18×10^{0})
	4	1.0502×10^{1} (8.87×10^{-1}) +	1.1610×10^{1} (2.67×10^{0}) =	1.2108×10^{1} (2.31×10^{0}) =	**8.3278×10^{0} (1.13×10^{0}) +**	1.3135×10^{1} (1.66×10^{0}) −	1.1784×10^{1} (1.86×10^{0})
	6	**1.7812×10^{1} (2.67×10^{0}) +**	2.8493×10^{1} (7.96×10^{0}) −	2.5577×10^{1} (6.56×10^{0}) =	1.9600×10^{1} (3.15×10^{0}) +	2.8780×10^{1} (6.34×10^{0}) −	2.3089×10^{1} (2.94×10^{0})
	8	**4.6440×10^{1} (6.74×10^{0}) +**	6.0658×10^{1} (1.04×10^{1}) −	5.7401×10^{1} (1.22×10^{1}) =	5.2089×10^{1} (9.93×10^{0}) =	8.0599×10^{1} (2.57×10^{1}) −	5.3179×10^{1} (6.30×10^{0})
	10	**1.5650×10^{2} (1.73×10^{1}) +**	1.8395×10^{2} (4.56×10^{1}) =	1.6005×10^{2} (2.10×10^{1}) +	1.8052×10^{2} (2.02×10^{1}) =	2.3848×10^{2} (4.50×10^{1}) −	2.1777×10^{2} (9.91×10^{1})
MaF6	3	2.7284×10^{2} (2.66×10^{1}) −	1.6807×10^{2} (2.61×10^{1}) −	N/A	**1.3929×10^{2} (2.58×10^{1}) =**	2.4823×10^{2} (1.99×10^{1}) −	1.3966×10^{2} (4.21×10^{1})
	4	2.6540×10^{2} (1.65×10^{1}) −	2.1893×10^{2} (2.74×10^{1}) −	N/A	1.4843×10^{2} (2.54×10^{1}) =	2.4619×10^{2} (1.58×10^{1}) −	**1.4091×10^{2} (3.31×10^{1})**
	6	2.4028×10^{2} (2.04×10^{1}) −	2.2333×10^{2} (2.28×10^{1}) −	N/A	1.5078×10^{2} (2.87×10^{1}) −	2.2341×10^{2} (1.12×10^{1}) −	**1.2670×10^{2} (3.65×10^{1})**
	8	2.1140×10^{2} (1.78×10^{1}) −	2.1633×10^{2} (1.99×10^{1}) −	N/A	1.3969×10^{2} (2.67×10^{1}) −	2.1538×10^{2} (1.37×10^{1}) −	**1.1838×10^{2} (3.60×10^{1})**
	10	2.0077×10^{2} (1.70×10^{1}) −	2.0177×10^{2} (1.70×10^{1}) −	N/A	1.2808×10^{2} (2.75×10^{1}) −	2.0416×10^{2} (1.56×10^{1}) −	**9.6546×10^{1} (5.06×10^{1})**
MaF7	3	**5.9342×10^{-1} (9.20×10^{-2}) +**	6.0456×10^{0} (7.44×10^{-1}) +	8.7173×10^{0} (1.20×10^{0}) =	5.5385×10^{0} (9.61×10^{-1}) +	9.4068×10^{0} (4.46×10^{-1}) −	8.8519×10^{0} (8.05×10^{-1})
	4	**8.9847×10^{-1} (1.21×10^{-1}) +**	8.5943×10^{0} (1.72×10^{0}) +	1.1882×10^{1} (1.72×10^{0}) =	9.1095×10^{0} (1.33×10^{0}) +	1.2849×10^{1} (4.94×10^{-1}) −	1.1682×10^{1} (9.83×10^{-1})
	6	**1.6050×10^{0} (4.26×10^{-1}) +**	1.5978×10^{1} (2.03×10^{0}) +	1.6023×10^{1} (3.35×10^{0}) =	1.5290×10^{1} (1.12×10^{0}) +	1.9694×10^{1} (8.66×10^{-1}) −	1.7763×10^{1} (2.96×10^{0})
	8	**2.7957×10^{0} (1.15×10^{0}) +**	2.2782×10^{1} (2.69×10^{0}) =	2.3852×10^{1} (2.34×10^{0}) =	2.2239×10^{1} (1.32×10^{0}) =	2.5877×10^{1} (7.13×10^{-1}) −	2.4003×10^{1} (2.83×10^{0})
	10	**4.6937×10^{0} (1.63×10^{0}) +**	2.8617×10^{1} (3.45×10^{0}) =	3.0500×10^{1} (2.86×10^{0}) =	2.8646×10^{1} (2.56×10^{0}) =	3.2563×10^{1} (1.30×10^{0}) −	2.8807×10^{1} (4.67×10^{0})
+/−/=		11/20/4	7/22/6	2/7/11	9/15/11	3/29/3	

4.6. Runtime

We compared the runtime performance of seven algorithms on DTLZ problems with 3 objectives and 50 decision variables. Table 6 presents the average computation time (in seconds) across 20 independent runs for each algorithm. Even though `CoEGAN-BO` does not execute rapidly, it is still in the same order of magnitude as the rest of the algorithms and fits into the range of expensive optimization.

Table 6. Average runtime (seconds) of all seven algorithms on DTLZ test problems.

Problem	K-RVEA	MOEA/D-EGO	KTA2	REMO	EDN-ARMOEA	`CoEGAN-BO`
DTLZ1	1.4427×10^2	3.7621×10^2	3.5105×10^3	3.7793×10^2	1.3358×10^1	5.1477×10^2
DTLZ2	1.5611×10^2	4.5886×10^2	8.7350×10^2	2.3010×10^2	1.3099×10^1	5.0543×10^2
DTLZ3	1.8730×10^2	4.3973×10^2	9.0765×10^2	4.4563×10^2	1.6600×10^1	5.0490×10^2
DTLZ5	1.3001×10^2	4.5142×10^2	9.2378×10^2	2.9397×10^2	1.5870×10^1	4.8810×10^2
DTLZ6	1.3371×10^2	4.6383×10^2	8.8606×10^2	3.1500×10^2	1.4933×10^1	4.6461×10^2
DTLZ7	1.4691×10^2	4.4943×10^2	8.5553×10^2	1.8000×10^2	1.5293×10^1	4.7112×10^2

Additionally, we evaluated the runtime of the algorithm's BO and GAN modules with 3 objectives and 50 decision variables. In Table 7, Avg. Train Time denotes the duration of a single training run, Total Train Time represents the cumulative runtime until the maximum number of genuine evaluations is exhausted, and Total Runtime indicates the overall execution time of the entire algorithm.

Table 7. BO and GAN operational efficiency comparison results (seconds).

Problem	Bayesian optimization (BO)		Generative Adversarial Network (GAN)		Total Runtime
	Avg. Train Time	Total Train Time	Avg. Train Time	Total Train Time	
DTLZ1	1.7787×10^1	6.9370×10^2	6.5291×10^{-2}	2.5464×10^0	5.1477×10^2
DTLZ2	1.6424×10^1	6.4054×10^2	7.6485×10^{-2}	2.9829×10^0	5.0543×10^2
DTLZ3	1.6385×10^1	6.3900×10^2	6.7070×10^{-2}	2.6157×10^0	5.0490×10^2
DTLZ5	1.5958×10^1	6.2237×10^2	6.8467×10^{-2}	2.6702×10^0	4.8810×10^2
DTLZ6	1.6909×10^1	6.5947×10^2	6.7778×10^{-2}	2.6433×10^0	4.6461×10^2
DTLZ7	1.7023×10^1	6.6392×10^2	7.5646×10^{-2}	2.9502×10^0	4.7112×10^2

5. Conclusions

In this work, we have developed a cutting-edge bi-stage Bayesian co-evolutionary optimization algorithm underpinned by generative adversarial networks specifically designed to tackle HEMaOPs. Our proposed method, dubbed `CoEGAN-BO`, employs distinct search strategies to expedite convergence, allowing two independent populations to evolve while facilitating the exchange of superior individuals at strategic intervals to enhance the convergence speed. This approach ensures that the optimal solution is a composite of the individual optimal solutions, thereby refining the overall solution quality within the population. A novel bi-stage infill criterion is also introduced, ensuring that solution diversity is not solely reliant on the measure of uncertainty. Moreover, by fine-tuning the balance factor, our algorithm adeptly navigates the trade-off between diversity and convergence. In the realm of high-dimensional optimization, where sample sparsity is an issue, our algorithm adeptly sidesteps the local optimum problem, advancing towards a more precise identification of the global optimum. The efficacy and superior performance of our

algorithm, with its integrated strategy, have been rigorously tested on a suite of established benchmark problems. The empirical results highlight our algorithm's remarkable superiority over other comparative algorithms across most of the test scenarios. The algorithm is competitive in solving most of the test problems used in the experiments. However, the algorithm converges slowly in solving DTLZ7 test problems. This shows that `CoEGAN-BO` is not capable enough when the real PF is a degenerate curve and discontinuous. Future research should consider selecting more effective prediction samples from the optimization process of irregular PFs to guide the population evolution in the right direction.

Author Contributions: Conceptualization, J.T. and H.B.; methodology, J.T. and H.B.; software, H.B. and Y.Z.; validation, H.B. and Y.Z.; formal analysis, H.B.; investigation, X.Z.; data curation, H.L.; writing—original draft preparation, J.T. and H.B.; writing—review and editing, J.T.; visualization, X.Z.; supervision, H.L.; project administration, J.T.; funding acquisition, J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This Research was supported, in part, by the National Natural Science Foundation of China under Grant 62006143, and by the Natural Science Foundation of Shandong Province under Grants ZR2025MS1012 and ZR2020MF152.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shu, X.; Liu, Y.; Liu, J.; Yang, M.; Zhang, Q. Multi-objective particle swarm optimization with dynamic population size. *J. Comput. Des. Eng.* **2023**, *10*, 446–467. [CrossRef]
2. Stewart, R.; Palmer, T.S.; Bays, S. An agent-based blackboard system for multi-objective optimization. *J. Comput. Des. Eng.* **2022**, *9*, 480–506. [CrossRef]
3. Sheikh, H.M.; Lee, S.; Wang, J.; Marcus, P.S. Airfoil optimization using design-by-morphing. *J. Comput. Des. Eng.* **2023**, *10*, 1443–1459. [CrossRef]
4. Tariq, R.; Recio-Garcia, J.A.; Cetina-Quiñones, A.J.; Orozco-del Castillo, M.G.; Bassam, A. Explainable Artificial Intelligence twin for metaheuristic optimization: Double-skin facade with energy storage in buildings. *J. Comput. Des. Eng.* **2025**, *12*, qwaf015. [CrossRef]
5. Nicolaou, C.A.; Brown, N. Multi-objective optimization methods in drug design. *Drug Discov. Today Technol.* **2013**, *10*, e427–e435. [CrossRef]
6. Machairas, V.; Tsangrassoulis, A.; Axarli, K. Algorithms for optimization of building design: A review. *Renew. Sustain. Energy Rev.* **2014**, *31*, 101–112. [CrossRef]
7. Liu, D.; Ouyang, H.; Li, S.; Zhang, C.; Zhan, Z.H. Hyperparameters optimization of convolutional neural network based on local autonomous competition harmony search algorithm. *J. Comput. Des. Eng.* **2023**, *10*, 1280–1297. [CrossRef]
8. El-Shorbagy, M.; Alhadbani, T.H. Monarch butterfly optimization-based genetic algorithm operators for nonlinear constrained optimization and design of engineering problems. *J. Comput. Des. Eng.* **2024**, *11*, 200–222. [CrossRef]
9. Soltani, S.; Murch, R.D. A Compact Planar Printed MIMO Antenna Design. *IEEE Trans. Antennas Propag.* **2015**, *63*, 1140–1149. [CrossRef]
10. del Valle, Y.; Venayagamoorthy, G.K.; Mohagheghi, S.; Hernandez, J.C.; Harley, R.G. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Trans. Evol. Comput.* **2008**, *12*, 171–195. [CrossRef]
11. Dai, R.; Jie, J.; Wang, Z.; Zheng, H.; Wang, W. Automated surrogate-assisted particle swarm optimizer with an adaptive parental guidance strategy for expensive engineering optimization problems. *J. Comput. Des. Eng.* **2025**, *12*, 145–183. [CrossRef]
12. Zhu, S.; Zeng, L.; Cui, M. Symmetrical Generalized Pareto Dominance and Adjusted Reference Vector Cooperative Evolutionary Algorithm for Many-Objective Optimization. *Symmetry* **2024**, *16*, 1484. [CrossRef]
13. Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm Evol. Comput.* **2011**, *1*, 61–70. [CrossRef]
14. Li, J.; Wang, P.; Dong, H.; Shen, J.; Chen, C. A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization. *Knowl.-Based Syst.* **2022**, *242*, 108416. [CrossRef]

15. Park, J.; Kang, N. BMO-GNN: Bayesian mesh optimization for graph neural networks to enhance engineering performance prediction. *J. Comput. Des. Eng.* **2024**, *11*, 260–271. [CrossRef]
16. Wang, X.; Jin, Y.; Schmitt, S.; Olhofer, M. Recent advances in Bayesian optimization. *ACM Comput. Surv.* **2023**, *55*, 1–36. [CrossRef]
17. He, C.; Zhang, Y.; Gong, D.; Ji, X. A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Syst. Appl.* **2023**, *217*, 119495. [CrossRef]
18. Zhan, D.; Xing, H. Expected improvement for expensive optimization: A review. *J. Glob. Optim.* **2020**, *78*, 507–544. [CrossRef]
19. Li, J.Y.; Zhan, Z.H.; Zhang, J. Evolutionary computation for expensive optimization: A survey. *Mach. Intell. Res.* **2022**, *19*, 3–23. [CrossRef]
20. Chandra, N.K.; Canale, A.; Dunson, D.B. Escaping the curse of dimensionality in Bayesian model-based clustering. *J. Mach. Learn. Res.* **2023**, *24*, 1–42. [CrossRef]
21. Lanthaler, S.; Stuart, A.M. The curse of dimensionality in operator learning. *arXiv* **2023**, arXiv:2306.15924. [CrossRef]
22. Gao, R. Finite-sample guarantees for Wasserstein distributionally robust optimization: Breaking the curse of dimensionality. *Oper. Res.* **2023**, *71*, 2291–2306. [CrossRef]
23. Altman, N.; Krzywinski, M. The curse(s) of dimensionality. *Nat. Methods* **2018**, *15*, 399–400. [CrossRef] [PubMed]
24. Chugh, T.; Jin, Y.; Miettinen, K.; Hakanen, J.; Sindhya, K. A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *22*, 129–142. [CrossRef]
25. Gutmann, H.M. A radial basis function method for global optimization. *J. Glob. Optim.* **2001**, *19*, 201–227. [CrossRef]
26. Wang, S.C. Artificial neural network. In *Interdisciplinary Computing in Java Programming*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 81–100. [CrossRef]
27. Liu, B.; Zhang, Q.; Gielen, G.G. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. Evol. Comput.* **2013**, *18*, 180–192. [CrossRef]
28. Wang, B.; Yan, L.; Duan, X.; Yu, T.; Zhang, H. An integrated surrogate model constructing method: Annealing combinable Gaussian process. *Inf. Sci.* **2022**, *591*, 176–194. [CrossRef]
29. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121. [CrossRef]
30. Cheng, J.; Yen, G.G.; Zhang, G. A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Trans. Evol. Comput.* **2015**, *19*, 592–605. [CrossRef]
31. Zhang, Q.; Liu, W.; Tsang, E.; Virginas, B. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. Evol. Comput.* **2009**, *14*, 456–474. [CrossRef]
32. Song, Z.; Wang, H.; He, C.; Jin, Y. A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Trans. Evol. Comput.* **2021**, *25*, 1013–1027. [CrossRef]
33. Guo, Z.; Ong, Y.S.; He, T.; Liu, H. Co-Learning Bayesian Optimization. *IEEE Trans. Cybern.* **2022**, *52*, 9820–9833. [CrossRef]
34. Jones, D.R.; Schonlau, M.; Welch, W.J. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [CrossRef]
35. Jones, D.R. A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.* **2001**, *21*, 345–383. [CrossRef]
36. Couckuyt, I.; Deschrijver, D.; Dhaene, T. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *J. Glob. Optim.* **2014**, *60*, 575–594. [CrossRef]
37. El-Beltagy, M.A.; Keane, A.J. Evolutionary optimization for computationally expensive problems using gaussian processes. *Proc. Int. Conf. Artif. Intell.* **2001**, *1*, 708–714.
38. Dennis, J.; Torczon, V. Managing approximation models in optimization. *Multidiscip. Des. Optim. State Art* **1997**, *5*, 330–347.
39. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
40. Wang, H.; Jiao, L.; Yao, X. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 524–541. [CrossRef]
41. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
42. Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.Y. Generative adversarial networks: Introduction and outlook. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 588–598. [CrossRef]
43. Qian, W.; Xu, H.; Chen, H.; Yang, L.; Lin, Y.; Xu, R.; Yang, M.; Liao, M. A Synergistic MOEA Algorithm with GANs for Complex Data Analysis. *Mathematics* **2024**, *12*, 175. [CrossRef]
44. He, C.; Huang, S.; Cheng, R.; Tan, K.C.; Jin, Y. Evolutionary Multiobjective Optimization Driven by Generative Adversarial Networks (GANs). *IEEE Trans. Cybern.* **2021**, *51*, 3129–3142. [CrossRef]
45. Chen, S.; Chen, S.; Hou, W.; Ding, W.; You, X. EGANS: Evolutionary Generative Adversarial Network Search for Zero-Shot Learning. *IEEE Trans. Evol. Comput.* **2024**, *28*, 582–596. [CrossRef]

46. Xue, Y.; Tong, W.; Neri, F.; Chen, P.; Luo, T.; Zhen, L.; Wang, X. Evolutionary Architecture Search for Generative Adversarial Networks Based on Weight Sharing. *IEEE Trans. Evol. Comput.* **2024**, *28*, 653–667. [CrossRef]
47. Li, H.; Zhang, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2008**, *13*, 284–302. [CrossRef]
48. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 773–791. [CrossRef]
49. Morgan, R.; Gallagher, M. Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements. *IEEE Trans. Evol. Comput.* **2013**, *18*, 456–461. [CrossRef]
50. Zhan, D.; Cheng, Y.; Liu, J. Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Trans. Evol. Comput.* **2017**, *21*, 956–975. [CrossRef]
51. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]
52. Kwak, N. Principal component analysis based on L1-norm maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1672–1680. [CrossRef]
53. Danielsson, P.E. Euclidean distance mapping. *Comput. Graph. Image Process.* **1980**, *14*, 227–248. [CrossRef]
54. Hao, H.; Zhou, A.; Qian, H.; Zhang, H. Expensive multiobjective optimization by relation learning and prediction. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1157–1170. [CrossRef]
55. Guo, D.; Wang, X.; Gao, K.; Jin, Y.; Ding, J.; Chai, T. Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 2084–2097. [CrossRef]
56. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
57. Bosman, P.A.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 174–188. [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.