

Article

# A Synergistic Multi-Objective Evolutionary Algorithm with Diffusion Population Generation for Portfolio Problems

Mulan Yang <sup>1,†</sup>, Weihua Qian <sup>2,†</sup>, Lvqing Yang <sup>2,\*</sup>, Xuehan Hou <sup>3</sup>, Xianghui Yuan <sup>1</sup> and Zhilong Dong <sup>1</sup> 

<sup>1</sup> School of Economics and Finance, Xi'an Jiaotong University, Xi'an 710000, China; mlyang@stu.xjtu.edu.cn (M.Y.); xhyuan@xjtu.edu.cn (X.Y.); zldong@xjtu.edu.cn (Z.D.)

<sup>2</sup> School of Informatics, Xiamen University, Xiamen 361000, China; clydeqian@stu.xmu.edu.cn

<sup>3</sup> School of Software, Xi'an Jiaotong University, Xi'an 710000, China; houxuehan@stu.xjtu.edu.cn

\* Correspondence: lqyang@xmu.edu.cn

† These authors contributed equally to this work.

**Abstract:** When constructing an investment portfolio, it is important to maximize returns while minimizing risks. This portfolio optimization can be considered as a multi-objective optimization problem that is solved by means of multi-objective evolutionary algorithms. The use of multi-objective evolutionary algorithms (MOEAs) provides an effective approach for dealing with the complex data involved in multi-objective optimization problems. However, current MOEAs often rely on a single strategy to obtain optimal solutions, leading to premature convergence and an insufficient population diversity. In this paper, a new MOEA called the *Synergistic MOEA with Diffusion Population Generation* (DPG-SMOEA) is proposed to address these limitations by integrating MOEAs with diffusion models. To train the diffusion model, a mixed memory pool strategy is optimized, which collects improved solutions from the MOEA/D-AEE, an optimized MOEA, as training samples. The trained model is then used to generate offspring. Considering the cold-start mechanism of the diffusion model, particularly during the training phase where it is not suitable for generating initial offspring, this paper adjusts and optimizes the collaborative strategy to enhance the synergy between the diffusion model and MOEA/D-AEE. Experimental validation of the DPG-SMOEA demonstrates the advantages of using diffusion models in low-dimensional and relatively continuous data analysis. The results show that the DPG-SMOEA performs well on the low-dimensional Hang Seng Index test dataset, while achieving average performance on other high-dimensional datasets, consistent with theoretical predictions. Overall, the DPG-SMOEA achieves better results compared to MOEA/D-AEE and other multi-objective optimization algorithms.

**Keywords:** multi-objective optimization; evolutionary algorithm; diffusion; portfolio problems

**MSC:** 68W50



**Citation:** Yang, M.; Qian, W.; Yang, L.; Hou, X.; Yuan, X.; Dong, Z. A Synergistic Multi-Objective Evolutionary Algorithm with Diffusion Population Generation for Portfolio Problems. *Mathematics* **2024**, *12*, 1368. <https://doi.org/10.3390/math12091368>

Academic Editor: Ekaterina Orlova

Received: 30 March 2024

Revised: 24 April 2024

Accepted: 25 April 2024

Published: 30 April 2024



**Copyright:** © 2024 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

Since the introduction of the mean-variance model by Markowitz [1], the maximization of returns while minimizing risk to the greatest extent possible has become a prominent issue in constructing investment portfolios. Balancing returns and risk represents a typical multi-objective optimization problem (MOP), especially as the scale and diversity of financial investments have grown in recent years. The emergence of multi-objective evolutionary algorithms (MOEAs) [2] provides an effective approach for handling complex data in MOPs.

MOEAs aim to identify a series of solutions that achieve optimal balance across multiple objectives, resulting in a set of Pareto optimal solutions—improving one subobjective unavoidably deteriorates other subobjectives—commonly referred to as the Pareto frontier. In the context of portfolio optimization, this Pareto frontier represents a series of combinations of returns and risks. To explore the solution space more effectively, MOEAs generate

a population of candidate solutions and then apply genetic operators such as mutation, crossover, and selection to evolve better solutions over several generations.

Classical MOEAs can face certain challenges such as avoiding premature convergence and maintaining population diversity. Among numerous multi-objective evolutionary algorithms, MOEA/D stands out significantly due to its unique problem decomposition strategy, making it easier to integrate with other optimization techniques to enhance its performance. Zhang et al. previously proposed the combination of the NBI-style Chebyshev method with MOEA/D to decompose subproblems in portfolio management [3]. MOEA/D-AEE, a variant of MOEA/D proposed by Qian et al. [4], is as a typical approach for exploring the solution space through stochastic sampling. By introducing a jumping Lévy strategy as a genetic operator, it enhances comprehensive search capabilities. However, its limitations persist, particularly in generating high-quality and diversified offspring, especially when dealing with complex data, thus impacting its efficiency in addressing modern portfolio optimization problems.

Diffusion models [5–7] have garnered wide attention as one of the generative models in recent years. By simulating the diffusion process of data, they gradually construct complex data distributions from Gaussian distributions. The core idea of diffusion models originates from Brownian motion in statistical physics, generating data by gradually applying noise and then reversing this process. A major advantage of diffusion models lies in their learning and generalization capabilities. Through training, diffusion models can learn the underlying patterns and structures of optimization problems from historical data, thereby guiding the generation of high-quality solutions more effectively. Once trained, the network can be rapidly applied to similar optimization problems without the need for a complete search process as with genetic algorithms. Furthermore, as diffusion models calculate the probability density function (PDF) explicitly, typically expressed through mathematical formulas, they demonstrate clear advantages in handling data with good continuity and low dimensions. Diffusion models can also be combined with other optimization techniques, such as hybrid algorithms, to further enhance the solution efficiency and quality.

Traditional genetic algorithms (GAs) have long been a popular choice for solving multi-objective optimization problems due to their flexibility in exploring the solution space and adapting to complex problems. However, GAs have certain limitations, particularly in terms of efficiency in solution quality and computational cost. With each run, GAs need to start searching from scratch, potentially leading to high computational costs and a low time efficiency. In contrast, using diffusion models to address multi-objective optimization problems offers new possibilities.

As a typical generative algorithm explicitly calculating PDFs, this paper focuses on the performance of diffusion models in low-dimensional data analysis. Due to their ability to generate complex data more accurately by gradually adding and removing noise, diffusion models hold an advantage in capturing complex data distributions. Considering the potential of diffusion models in generating and handling complex problems, their application in multi-objective optimization algorithms becomes particularly meaningful. Diffusion models may offer more stable and diversified solutions for MOPs, bringing new perspectives and innovative approaches to this field.

Based on the considerations above, this paper designs a new optimization algorithm, DPG-SMOEA, a synergistic MOEA with diffusion population generation, and applies it to multi-objective optimization problems in investment portfolios across low, medium, and high-dimensional data.

The main contributions of this work include:

1. A novel cooperative diffusion model generative algorithm (DPG-SMOEA) is proposed for explicit PDF solving, demonstrating evident advantages in low-dimensional data analysis (Hang Seng Index test dataset) and providing comparable results in the analysis of other high-dimensional data (such as the Nikkei 225 Index test dataset), consistent with theoretical conjecture.

2. The DPG-SMOEA establishes a mixed-memory optimization pool, storing high-quality solutions generated by the MOEA/D-AEE and utilizing samples from this mixed-memory optimization pool for pretraining the diffusion model. The trained model shows significant effectiveness in generating high-quality offspring, addressing the high uncertainty that traditional random sampling methods cannot handle.
3. Diffusion models exhibit typical cold start characteristics, especially in the initialization and pretraining stages. This paper proposes a new cooperative strategy, utilizing the MOEA/D-AEE to generate high-quality solutions in the early stages and employing the diffusion model to generate offspring in the later stages, thereby avoiding the disadvantage of diffusion models' poor performance during the cold start phase.

## 2. Related Work

### 2.1. Overview of Multi-Objective Evolutionary Algorithms

Multi-objective optimization refers to optimization problems involving a large number of objectives, generally greater than four, that we need to optimize simultaneously [8,9]. However, most MOPs do not lend themselves to a single solution, but rather to a set of solutions. Such solutions are actually “trade-offs” or good compromises between goals [10]. Since they are an evolutionary algorithm (EA) because of their population-based nature, multiobjective EAs (MOEAs) have been very popular for solving MOPs [11,12]. The classical multi-objective optimization algorithms for solving multi-objective optimization problems are NSGA II [13], NSGA III [14], SPEA2 [15], MOEA/D [16], MOGLS [17], and C-MOGA [18]. Among them, NSGA II, NSGA III and SPEA2 are dominance-based MOEAs, while MOEA/D, MOGLS and C-MOGA are decomposition-based MOEAs. Dominance-based approaches optimize the MOP by simultaneously optimizing all objectives [19], using a Pareto-based fitness allocation strategy to identify all non-dominated individuals from the current evolutionary population as elite individuals for generating the next generation. Decomposition-based methods decompose the approximation of a PF into many single-objective optimization subproblems [20], aggregating the objectives of the original multi-objective problem in a linear or nonlinear manner, i.e., multiple objectives are aggregated into a single one, and then approaching the global Pareto-optimal solution continuously through the evolution of the population.

### 2.2. Overview of Portfolio Optimization Problem

Markowitz's mean-variance (MV) model is the forerunner of modern portfolio theory [1]. The model considers the construction of a portfolio in terms of both the return and risk dimensions of an asset, and there are two objectives for investors, namely maximising return and minimising risk. However, these two goals are often in conflict, so the model often ends up with a set of solutions that trade-off between return and risk, i.e., the Pareto frontier.

However, some assumptions of this model are overly idealistic, leading to its limited ability to construct portfolios in the real world. Therefore, scholars have proposed a series of constraints to make the model more closely aligned with real-world situations. The constraints proposed by scholars include:

1. Cardinality constraints (CCs), which restrict the number of assets to be invested in to a specific number or within a range, allowing asset managers to more conveniently track assets and reduce trading costs;
2. Floor and ceiling constraints (FCs), which stipulate that the weight of each asset must fall within a certain range, reflecting investors' preferences for specific assets;
3. Round-lot constraints (RL), which specify that asset purchases must be made in multiples of a certain quantity, making transactions more similar to real-world occurrences;
4. Pre-assignment constraints (PA), which determine in advance whether a certain group of assets must be invested in, also reflecting the preferences of investors for specific assets.

Introducing these constraints will increase the complexity of the mean-variance (MV) model, making the model an NP-hard problem. For example, the introduction of cardinality constraints would transform the original quadratic programming problem into a mixed-integer quadratic programming problem, making solving the MV model an NP-hard problem [21,22]. At this point, heuristic algorithms such as multi-objective evolutionary algorithms (MOEAs) can obtain feasible efficient frontiers within a reasonable computational time, providing specific advantages for solving large-scale portfolio optimization problems. Arnone et al. (1993) first utilized genetic algorithms (GAs) to obtain solutions for the MV model [23]. However, due to their definition of risk as tail risk, the results were difficult to obtain through quadratic programming. Anagnostopoulos and Mamanis (2011) compared the performance of five MOEA algorithms on MV models with cardinality constraints, operating on a publicly available dataset containing 2196 assets [24]. Lwin et al. (2014) proposed an efficient learning-guided hybrid multi-objective evolutionary algorithm to solve MV models with four types of constraints and compared it with existing multi-objective evolutionary algorithms [25].

### 2.3. Standardized Test Functions for Multi-Objective Evolutionary Algorithms

Standard test functions for multi-objective optimization algorithms are mathematical functions used to evaluate and compare the performance of multi-objective optimization algorithms. These functions usually have the following characteristics:

1. **Realism:** These functions are designed to simulate real-world problems to better reflect actual multi-objective optimization challenges.
2. **Nonlinear and multimodal:** standard test functions are usually nonlinear and may contain multiple local optimal solutions (multimodal), which can be used to optimize the performance of the algorithm.
3. **Scalability:** these functions can often be scaled to meet the needs of different dimensions, thus providing a good understanding of its ability to evaluate the performance of the algorithm in high-dimensional spaces.

Common standard test functions for MOEAs include:

1. **DTLZ (Deb–Thiele–Laumanns–Zitzler)** functions are a set of commonly used benchmarking functions that are scalable and nonlinear, and are often used to evaluate the effectiveness of an algorithm when dealing with multiple objectives.
2. **ZDT (Zitzler–Deb–Thiele)** functions are another set of widely used test functions, which are usually used to evaluate the effectiveness of multi-objective optimization algorithms. They involve both linear and nonlinear relationships and are designed to test the robustness of the algorithm against different types of functions.
3. **UFs (unconstrained optimization test functions)** are used to evaluate the robustness of an algorithm to non-linear relationships that may be encountered in real-world problems.

These functions are often used to test the convergence, equilibrium, and extensibility of an algorithm in solving multi-objective optimization problems. These functions have been widely used to evaluate and compare the performance of multi-objective optimization algorithms and to help researchers and developers understand the strengths and weaknesses of different algorithms.

### 2.4. Diffusion Model

Diffusion models [26] have received extensive attention in recent years as a type of generative model. These models construct complex data distributions by simulating the diffusion process of data, gradually moving away from the Gaussian distribution. The core idea of diffusion models originates from Brownian motion in statistical physics, where data are generated by progressively adding noise and then reversing the process to recover the original data. The mathematical foundation of this approach lies in a deep understanding and application of stochastic processes and probability theory. Existing

diffusion models can be classified into three categories: Denoising Diffusion Probabilistic Models (DDPMs) [5], Fractional-based Generative Models (SGMs) [27], and Stochastic Differential-Equation-based Generative Models (SDEs) [28].

1. Denoising Diffusion Probabilistic Models involve two processes [26]: a forward noisy process and a reverse denoising process. In the forward process, noise is gradually added to the original data, with each step's data depending on the previous step's result until, at step  $T$ , the data become pure Gaussian noise. The reverse process involves removing noise points step by step to recover the original data.
2. The core of Fractional-based Generative Models lies in the concept of Stein fractions (also known as fractional or score functions) [29]. Given a probability density function  $p(x)$ , its score function is defined as the gradient of the logarithm of the probability density,  $\nabla_x * \log p(x)$ . Here, the Stein fraction is a function of the data  $x$ , rather than a function of model parameters  $\sigma$ . It represents a vector field indicating the direction of the maximum growth rate for the probability density function.
3. Stochastic Differential-Equation-based Generative Models utilize stochastic differential equations for noise perturbation and sample generation. Additionally, the denoising process requires estimating the score function of the noise data distribution. This score function controls the diffusion process that perturbs the data into noise according to the following stochastic differential Equation (1):

$$dx = f(x, t)dt + g(t)dw \quad (1)$$

where  $f(x, t)$  and  $g(t)$  represent the diffusion and drift functions of the SDE, respectively, and  $w$  is a standard Wiener process (also known as Brownian motion). The forward process in DDPMs and SGMs discretizes these SDEs.

Diffusion models generally avoid the common problem of model collapse in generative adversarial networks (GANs), and the generated images are clearer and more detailed than those produced by variational autoencoders (VAEs). However, the training process of diffusion models is often complex and computationally expensive. Moreover, there are still challenges in dealing with high-dimensional and complex data, especially concerning the time efficiency and resource consumption when applying these models to practical problems. Currently, researchers are exploring ways to improve the efficiency of diffusion models, including reducing the training time and optimizing the generation process. Additionally, applying diffusion models to domains beyond image generation is an important direction for future research.

Compared to GANs, diffusion models employ a different approach to generate data. They control the generation process more accurately by gradually adding and removing noise, making them particularly suitable for generating complex data. Considering the potential of diffusion models in data generation and handling complex problems, their application to multi-objective optimization algorithms is particularly meaningful. Diffusion models may provide more stable and diverse solutions for multi-objective optimization problems and bring new perspectives and innovative approaches to the field.

### 3. Algorithm Implementation

In this section, the core issues targeted by this research are first clearly defined, including the background of the problem and key elements, as well as its importance and challenges within the current field. After defining the problem explicitly, a specific algorithm framework is proposed to address the defined problem through innovative approaches. This algorithm framework integrates the latest technological advancements and theoretical research to optimize the effectiveness of the solutions.

#### 3.1. Problem Definition

In multi-objective problems (MOPs), optimizing a single objective may come at the cost of sacrificing other objectives, while achieving multiple optimal objectives may require iden-

tifying the optimal constraints among them [30]. Classical portfolio optimization problems must simultaneously satisfy the highest return and lowest risk. However, in reality, returns and risks are often mutually constrained; achieving higher returns may lead to higher risks, while reducing risks may result in lower returns. Classical multi-objective evolutionary algorithms (MOEAs) have provided good solutions for MOPs. By considering the weights of each objective, a set of optimal solutions that satisfy the problem constraints, known as the Pareto solution set, can be identified. In MOEA/D-AEE [31], the authors combined dynamic search and cooperative development techniques, solving the self-optimization problem of low- to mid-dimensional data and enhancing the algorithm's adaptability and global search capabilities. However, effective solutions have yet to be found for large and highly complex MOPs.

Under the MOEA framework, this paper attempts to combine the diffusion model with genetic evolution algorithms, initially relying on their learning capabilities and adaptability to complex problems to generate diverse and high-quality descendant populations, thereby enhancing the overall optimization process. Thus, in this paper, the DPG-SMOEA based on MOEA/D-AEE is introduced to handle MOPs with complex data.

To evaluate the effectiveness of the proposed algorithm, experiments were conducted using financial datasets from the OR library, which includes five datasets comprising low-dimensional Hang Seng data and 225-dimensional high-dimensional Nikkei data. The portfolio optimization problem can be formulated as maximizing returns, as shown in Equation (2), and minimizing risk, as shown in Equation (3):

$$\max_{\omega} RETURN = \sum_{i=1}^n r_i \omega_i \quad (2)$$

$$\min_{\omega} RISK = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} \omega_i \omega_j \quad (3)$$

subject to,

$$\sum_{i=1}^n \omega_i = 1 \quad (4)$$

$$0 \leq \omega_i \leq 1 \quad (5)$$

where  $\omega = (w_1, w_2, \dots, w_n)$  is a vector representing the invest ratios of  $n$  assets.  $r_i$  is the return rate of the  $i$ -th asset, and  $\sigma_{ij}$  is the co-variance between the return of the  $i$ -th asset and the  $j$ -th asset.

The aim of the MOEA is solving portfolio optimization problems by determining the Pareto solution set and Pareto front. The Pareto front, which consists of all feasible solutions in the problem domain that are not dominated by any other solution, embodies this concept of multi-objective optimization.

This paper explores the possibility of integrating diffusion models with MOEAs within the framework of multi-objective optimization. By leveraging the advanced data-generation capabilities of diffusion models and the efficient solution space exploration of genetic algorithms, the aim is to generate a more diverse and higher-quality offspring population when addressing complex optimization problems. The combination of the powerful learning capabilities of diffusion models and the natural selection mechanism of genetic algorithms can provide a broader exploration of the solution space and more precise solution filtering during the optimization process. This integration offers new approaches and methodologies for effectively handling low-dimensional and complex objective functions. By combining these two techniques, it becomes possible to effectively leverage the advantages of diffusion models in understanding and simulating complex data distributions, while also utilizing the global search capability of genetic algorithms to seek more comprehensive and diversified optimization solutions. Ultimately, this innovative approach enhances the efficiency and effectiveness of the overall optimization process,

providing a fresh perspective for tackling complex multi-objective optimization problems. The following are three key challenges:

- Challenge 1: How can we coordinate diffusion networks?
- Challenge 2: How can we address the issue of training data for diffusion models?
- Challenge 3: How can we introduce offspring generated by diffusion models?

The specific solutions are as follows:

1. Based on the cold-start characteristic of diffusion models, an optimized collaborative strategy is designed in this paper.
2. Establish a mixed pool training pool, optimize the population using the MOEA/D-AEE in the early stage, and store good solutions in the mixed pool as training data.
3. Calculate the relevant information of the offspring and replace the original solutions in the population, making the generated offspring more applicable for addressing the random uncertainty issues of traditional sampling methods.

### 3.2. DPG-SMOEA Framework

In a portfolio optimization problem, it is necessary to balance returns and reduce risks simultaneously. The dimensions of the investment portfolio optimization problem vary, ranging from the low-dimensional Hang Seng Index to the high-dimensional Nikkei Index. Transforming the multi-objective optimization problem (MOP) into matrix-vector form, the corresponding solutions are represented by a weight vector consisting of floating-point numbers between 0 and 1. This paper utilizes the MOEA/D-AEE as an enhanced version of the basic MOEA/D, incorporating the Lévy flight strategy as a genetic operator. As in [32], the variation formula is expressed as follows in Equation (6):

$$y(\varepsilon) = \varepsilon x^i + \alpha_0(x^i - x^j) \oplus Levy(\beta) \tag{6}$$

where  $x^i$  and  $x^j$  represent two parents used for offspring reproduction,  $y$  denotes the generated offspring,  $\oplus$  denotes element-wise multiplication,  $\alpha_0$  is the scaling factor used, and  $Levy(\beta)$  is the vector generated by the Mantegna algorithm (MA). The *Levy* flight mutation operator adopted in this study, resembling the differential evolution (DE) mutation operator, adjusts different individuals. The distinctive feature lies in the proportion factor used for *Levy* flight mutation, derived from a heavy-tailed distribution rather than a fixed value. Moreover, the *Levy* flight mutation typically involves only two parents, instead of three. The *Levy* flight mutation formula is represented as follows in Equation (7).

$$Levy(\beta) \sim \frac{u}{|v|^{\frac{1}{\beta}}} \tag{7}$$

$$u \sim N(0, \sigma_u^2), v_s \sim N(0, \sigma_v^2) \tag{8}$$

$$\sigma_v = \left[ \frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(1 + \frac{\beta}{2}\right) \cdot \beta \cdot 2^{\beta - \frac{1}{2}}}\right]^{\frac{1}{\beta}}, \sigma_v = 1 \tag{9}$$

where  $\beta$  is a parameter of the *Levy* mutation algorithm.  $u$  represents a random variable with a mean of 0 and a variance of  $\sigma_u^2$ .  $v$  is another random variable with a mean of 0 and a variance of  $\sigma_v^2$ . The expressions for  $\sigma_v$  and  $\sigma_u$  are provided in Equation (8).  $\Gamma$  denotes a function. Due to the limitations of the global search capability and adaptive control in the *Levy* flight algorithm, it exhibits a lower frequency of generating superior offspring in the later stages of evolution, thereby reducing the effectiveness of portfolio optimization. To address these challenges, MOEA/D-AEE introduces a dynamic coefficient to globally explore the objective space, enhancing the search capability. Additionally, MOEA/D-AEE utilizes joint coefficient adaptive control for parent selection exploration and exploitation to ensure algorithm convergence. This is achieved by randomly selecting a number between

0 and 1, then selecting a parent node randomly. If the number is less than 0, a parent node is chosen from the neighborhood; otherwise, a parent is selected from the initial population. Furthermore, with a probability of  $1 - \epsilon$ , another parent is selected from the initial population. The formula of this algorithm [4] is as follows in Equation (10):

$$y(\epsilon) = \epsilon x^i + \alpha_0(1 - \epsilon)(x^i - x^j) \oplus \text{Levy}(\beta) \quad (10)$$

The experimental results indicate that the MOEA/D-AEE [4] demonstrates robust global search and adaptive capabilities. The performance indicators of MOEA/D-AEE significantly outperform classical multi-objective optimization algorithms such as NSGA-II, MOEA/D-Lévy, and MOEA/D-DEM. The GAN in MOEA/D-AEE and diffusion in the DPG-SMOEA are both generative models, and the two algorithms have similarities. In this paper, the DPG-SMOEA is further improved based on MOEA/D-AEE. Algorithm A1 provides the detailed pseudocode. Here,  $t$  represents the number of iterations, iterating through each individual in the population,  $g$  controls when to utilize the DPG-SMOEA, acting as a threshold,  $i$  iterates through each individual in the population, and  $b$  represents the current individual's neighbor.  $M$  and  $V$  denote the [return, risk] of individuals in the population. The parameter *count* signifies a single update counter, while *nr* sets the upper limit for updating neighbors. In the offspring generation process, if the number of iterations is below the threshold, polynomial mutation is employed; otherwise, the DPG-SMOEA is trained using the population from recent iterations to generate new solutions.

The steps of the algorithm are as follows:

1. Read the relevant information from the dataset and set the parameters of the problem, including the number of assets, the statistics of returns and risks, and other related information such as the covariance matrix. Also, set the initial parameters of the algorithm, such as the population size, number of neighbors, etc.
2. Define neighbors, generate the initial population, and calculate the [return, risk] values and Pareto frontier extreme points for each individual.
3. Instantiate the diffusion model.
4. Iterate in a loop, where each iteration represents a complete search process, including selection, mutation, crossover, replacement, and other operations.
5. Iterate through each individual in the population in each iteration. This step involves performing mutation and crossover operations on each individual to generate new offspring.
6. Determine the mating pool for the offspring generated by the mutation operator.
7. Decide the reproduction method for the offspring. For the first  $g$  generations, use polynomial mutation to generate offspring. After the  $g$ -th generation, use the saved better solutions in the population to train the diffusion model, and then use the diffusion model for inference to generate offspring. Calculate the returns, risks, reference points, and other information for the offspring.
8. Update the extreme points of the Pareto frontier, calculate the reference points and weight vectors, determine whether to replace the parent nodes, and then update the population.
9. Set counters, iterate through neighbors, update the population if the offspring is better than the parent node, and exit the loop when the iteration is complete or the counter reaches the upper limit.

The offspring generated by the hybrid operators (AEE operator and diffusion model) may not adhere to constraints, necessitating a repair operation on the offspring. The repair steps are outlined as follows:

1. Iterate through the weight of all assets in the offspring, setting negative values to 0.
2. Sum the weights of all assets to obtain a total sum ( $s$ ).
3. If ( $s$ ) is not equal to 0, scale all weight vectors to ensure their sum equals 1; if ( $s$ ) equals 0, use a generated solution satisfying the offspring constraints.

In the experiments, the number of iterations is set to 1500, and the population size is set to 100. Considering the cold start mode of the diffusion model, optimal solutions are carefully selected as training samples, with (g) set to 1000 in this study.

### 3.3. Synergy in Diffusion Models

The diffusion model is a generative model, different from other generative networks such as variational autoencoders (VAE) and generative adversarial networks (GANs). In the forward stage, noise is gradually applied to the original data until they are completely degraded into Gaussian noise, and then in the reverse stage, the model learns how to restore the original image from the Gaussian noise. The diffusion model consists of two processes: the forward noise addition process and the reverse denoising process.

In the forward noise addition process, noise is gradually added to the original data, and each step of the data obtained is related to the result of the previous step, until the data at step T become pure Gaussian noise. The reverse process is a continuous process of removing noise points, achieving the restoration of the original data through step-by-step denoising. The diffusion model can be used for generating continuous signals such as in speech synthesis, image generation, and super-resolution.

This paper introduces the stable diffusion model into the framework of the MOEA/D, investigating the collaborative methods of explicitly solving the probability density function (PDF) with generative algorithms and the MOEA/D. The specific descriptions of the diffusion model’s relevant theory and the methods used in this paper are as follows.

Assuming that the sample is drawn from the data distribution  $q(x_0)$ , the model needs to learn a distribution  $p_\theta(x_0)$  to approximate the original data distribution  $q(x_0)$ , facilitating the process of sampling data from the learned distribution  $p_\theta(x_0)$  to complete the data generation. The denoising diffusion model is a latent variable model, expressed in the form of Equation (11).

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T} \tag{11}$$

where  $p_\theta(x_{0:T})$  can be represented as Equation (12):

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^T p_\theta^{(t)}(x_{t-1}|x_t) \tag{12}$$

The inference distribution on the latent variables can be represented as Equation (13).

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \tag{13}$$

The forward process of the denoising diffusion model is a fixed process, and the latent variables are high-dimensional. Its forward process needs to conform to the Markov chain assumption, with parameters  $\alpha_{1:T} \in (0, 1]$ . The forward noise addition process can be represented as Equation (14), where it should be ensured that there are positive terms on the covariance matrix’s diagonal.

$$q(x_t | x_{t-1}) = \mathcal{N}\left(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}x_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)I\right) \tag{14}$$

When  $T$  is sufficiently large,  $x_T \sim N(0, I)$ . Assuming the accurate inverse distribution is  $q(x_{t-1}|x_t)$ , it is possible to sample  $x_T$  from the distribution  $N(0, I)$  and obtain data that conform to the distribution  $q(x_0)$  through the inverse distribution. Since the inverse distribution  $q(x_{t-1}|x_t)$  depends on the entire data distribution, this paper can approximate it with a neural network  $p_\theta(x_{t-1}|x_t)$ , expressed specifically as in Equation (15).

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1} : \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{15}$$

Intuitively, the forward process gradually adds noise to the observed values, and the generation process gradually removes the noise from the observed values. This process needs to satisfy a special property, as shown in Equation (16).

$$q(x_t | x_0) = \int q(x_{1:t} | x_0) dx_{1:(t-1)} = \mathcal{N}(x_t: \sqrt{\alpha_t}x_0, (1 - \alpha_t)I) \tag{16}$$

Therefore, this paper can represent  $t$  as a linear combination of  $x_0$  and the noise variable, specifically as shown in Equation (17).

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon \tag{17}$$

where  $\epsilon \sim \mathcal{N}(0, 1)$ .

When setting  $\alpha_T$  very close to 0, for all  $x_0$ ,  $q(x_t|x_0)$ , a standard Gaussian distribution will be approximated; hence, it is natural to set  $p\theta(x_T) = N(0, I)$ . If all conditional distributions are modeled as Gaussian distributions, with the mean of the Gaussian distribution being learnable and the variance being fixed, the training objective function can be simplified to Equation (18).

$$L(\epsilon_\theta) = \sum_{t=1}^T \mathbb{E}_{x_0 \sim q(x_0), \epsilon_t \sim \mathcal{N}(0, I)} \left[ \left\| \epsilon_\theta^{(t)} \left( \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon_t \right) - \epsilon_t \right\|_2^2 \right] \tag{18}$$

where  $\epsilon_\theta = \left\{ \epsilon_\theta^{(t)} \right\}_{t=1}^T$  and the model's parameters differ at different times  $t$ . For a well-trained model,  $x_0$  is sampled from  $x_T$ , and by iterative sampling,  $x_0$  is eventually obtained.

The pseudocode of the diffusion model algorithm used in the DPG-SMOEA is shown in Appendix A.1.

### 4. Experiments and Analysis

#### 4.1. Test Data and Evaluation Indexes

##### 4.1.1. Test Data

In order to verify the effectiveness of the algorithm, the experiment used the financial test dataset in the OR library, which is mainly used for portfolio optimization. The dataset contains five files collected from the financial indices of Hang Seng (Hong Kong), DAX (Germany), FSET (UK), S&P (US) and Nikkei (Japan).

The weekly price data over five years were obtained for the stocks in these indices. Stocks with missing values were dropped. We had 291 values for each stock, from which we calculated the (weekly) returns and covariances, and the size of our five test problems ranged from  $N = 31$  (Hang Seng) to  $N = 225$  (Nikkei).

Therefore, the dimensionality (i.e., the number of targets or assets) of these five datasets is 31, 85, 89, 98, and 225, respectively, and the datasets include low-, medium-, and high-dimensional data. The algorithm in this paper needs to reweight the assets based on the information in the datasets to find high-return or low-risk portfolios as early as possible. The dataset can help to validate the effectiveness of the algorithm on complex and low-dimensional data and to validate the DPG-SMOEA's effectiveness in different dimensions. The dataset includes the number of assets, the average return and return variance for each asset, and the correlation between assets. Table 1 gives the relevant information about the dataset.

**Table 1.** A summary of the datasets used in experiments.

Dataset	Region	Dimensions
Hangsheng	Hongkong	31
DAX100	Germany	85
FTSE100	U.K.	89
S&P100	U.S.	98
Nikkei	Japan	225

#### 4.1.2. Evaluation Indexes

Evaluation metrics for multi-objective evolutionary algorithms can be divided into three main categories: convergence metrics, distribution metrics and comprehensive metrics. The convergence metrics, such as the generation distance (GD) and inverse generation distance (IGD), for evaluating multi-objective evolutionary algorithms can be divided into three main categories: convergence metrics, distribution metrics and comprehensive metrics. Convergence metrics, such as GD and IGD, are used to evaluate how close the solution set is to the Pareto optimal surface. Distribution metrics are used to assess the diversity and homogeneity of the solution set and include the spacing spatial evaluation method, the diversity metric (Delta) and the maximum spread (MS). Integration metrics combine the convergence and distribution of the solution set, such as the HyperVolume (HV) metric. In the study in this chapter, six metrics, GD, IGD, Spacing, Delta, HV and MS, are used, where GD and Spacing are used to evaluate the convergence of the algorithms, MS and Delta are used to evaluate the distribution of the algorithms, and IGD and HV are used to evaluate the integrated performance of the algorithms.

#### 4.2. Comparison Algorithm

This experiment compares five multi-objective optimization algorithms (e.g., NSGA-II, MOEA/D-AEE, MOEA/DGA, MOEA/D-D-DEM, and MOEA/D-DE). NSGA-II uses non-dominance sorting and the congestion distance to maintain population diversity. MOEA/D-D-D-DE and MOEA/D-DEM use a decomposition approach to transform the multi-objective problem into multiple single-objective subproblems and solve these subproblems using differential evolutionary algorithms. MOEA/D-GA uses a genetic algorithm to solve the multi-objective problem. All five algorithms use cross-variance for progeny propagation and all use a single operator.

#### 4.3. Parameter Settings

For each dataset, in this paper, the experiment was repeated 51 times according to APG-SMOEA [4], for a total of 255 experiments. The population size was set to 100 and the neighborhood size was set to 20, the number of children generated in each iteration of the algorithm was 1500, and the maximum number of individual neighbors updated in each iteration was 2. The parameters of the AEE algorithm in the hybrid algorithm were set according to the literature [4],  $\sigma$  was set to 0.9 to determine the source of the parents, and the scaling factor was set to 1.1. For the parameters of the diffusion model, the learning rate was set to 0.0001, and the total number of iterations was set to 200. For the parameters of the diffusion model, the learning rate was 0.0001, and the total number of iterations was set to 200. In this paper, we chose the previous generation to update the population with the traditional algorithm, and then we trained the diffusion model with the better solutions saved in the previous generation, generating 100 individuals each time and then performing fast non-dominated sorting to select the top 20 individuals to replace the parent generation. The experimental results are obtained from the last generation of the population.

#### 4.4. Analysis of Results

In this section, a total of two comparison experiments were conducted. Experiment one adopted the synergistic method of the previously proposed APG-SMOEA [32] to adaptively control whether to adopt the diffusion model for offspring generation by means of population entropy. The results of the experiment show that because of the effect of the diffusion model's cold-start mechanism, the effect is not good, and the performance is not outstanding compared with the other algorithms, which is basically what was predicted in this paper. Experiment two is a synergistic approach using the DPG-SMOEA, and the experimental results show excellent performance on low-dimensional datasets. The specific experiments are summarized as follows.

#### 4.4.1. Experiments on Synergistic Methods of Population Entropy

In this section, referring to previous studies [33], experiments on the synergistic approach of APG-SMOEA (hereinafter referred to as Experiment I) are carried out, which specifically employ the synergistic approach of APG-SMOEA to design the DPG-SMOEA and carry out the related experiments. Experiment I adopts the adaptive control strategy of population entropy [34] to control the probability of generating offspring of the diffusion model to achieve the effect of adaptive control. The similarity between individual  $P_i$  and other individuals  $P_j$  is derived by calculating the sum of Euclidean distances between individual  $P_i$  and every other individual  $P_j$  in the population, as shown in the equation below.

$$P_i = \sum_{j=1}^{\text{len}(P)} (|P_i - P_j|^2) \quad (19)$$

$P_i, P_j$  are individuals in the population, and the expression is converted to a probability using a Gaussian distribution, as shown in the equation below:

$$P_i = \exp\left\{\frac{-P_i}{2 \cdot \sigma(P_i)}\right\} \quad (20)$$

where  $\sigma(P_i)$  is the variance of the individuals in the population.

After normalizing the probabilities, the entropy of the population is calculated by the following equation:

$$E = - \sum_{i=1}^{\text{len}(P)} (P_i * \log_2 P_i) \quad (21)$$

After obtaining the entropy of the population, this paper will deflate the ratio of the generation of offspring from the adversarial network and the AEE operator by comparing the current population entropy with the historical average population entropy, i.e., the value of  $\alpha$ . The specific formula is shown in the equation below.

$$\alpha = \begin{cases} \max\left(\frac{\alpha_{list[-1]}}{\text{scaling\_factor}}, 0\right), & E < \text{avg\_E} \\ \min(\alpha_{list[-1]} \cdot \text{scaling\_factor}, 1), & E \geq \text{avg\_E} \end{cases} \quad (22)$$

where  $\alpha_{list[-1]}$  is the initial alpha value,  $\text{scaling\_factor}$  is the scaling factor, and  $\text{avg\_E}$  is the historical average population entropy. Figures 1 and 2 show the variation in IGD metrics in Hang Seng and DAX datasets, respectively, using the synergistic method of population entropy control. From the analysis of the figures, it can be seen that the differences between the algorithms are subtle, while this synergistic method does not perform well on both the low-dimensional Hang Seng and high-dimensional DAX datasets. It performs slightly poorer than the APG-SMOEA and the MOEA/D-AEE in terms of early convergence, but perform better when generations exceed 100.

Tables A1 and A2 in the Appendix show the results of the synergistic approach designed in Experiment 1 on the Hang Seng and DAX datasets, respectively, which more clearly display the performance of the algorithms. The results show that this synergistic approach is ineffective, and only one of the six indexes of HV (i.e., the optimum, median, and standard deviation of the HVs in the two tables) is tied for first place, and the other indexes do not exhibit outstanding changes. Its performance is far inferior to that of the MOEA/D-AEE. The plot of HV is not displayed as the difference in HV is rather small and is clearer in the form of tables. The experimental results also fit the theoretical inference that the adaptive control of generating offspring by population entropy is not applicable to the synergistic method of explicitly solving PDFs. The diffusion model with its cold-start approach does not have an advantage in the pre-iteration period, and it is necessary to carry out the next step of the experiment to design a synergistic approach that can introduce a diffusion model to generate offspring in the late iteration period.

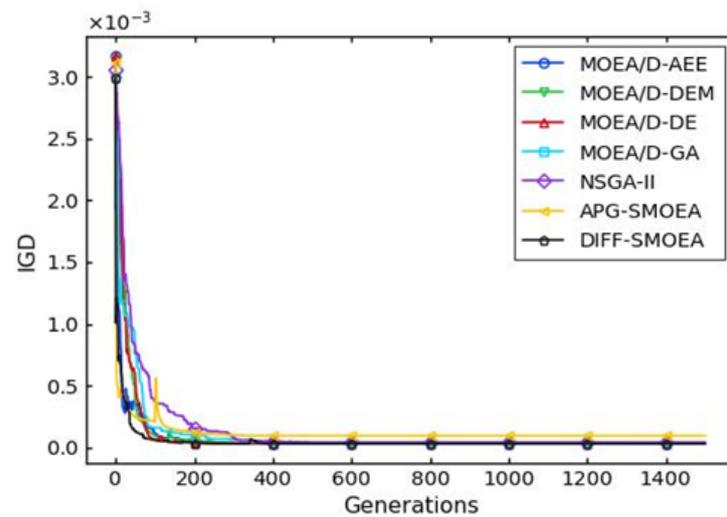


Figure 1. Experiment 1: IGD metrics on Hang Seng dataset.

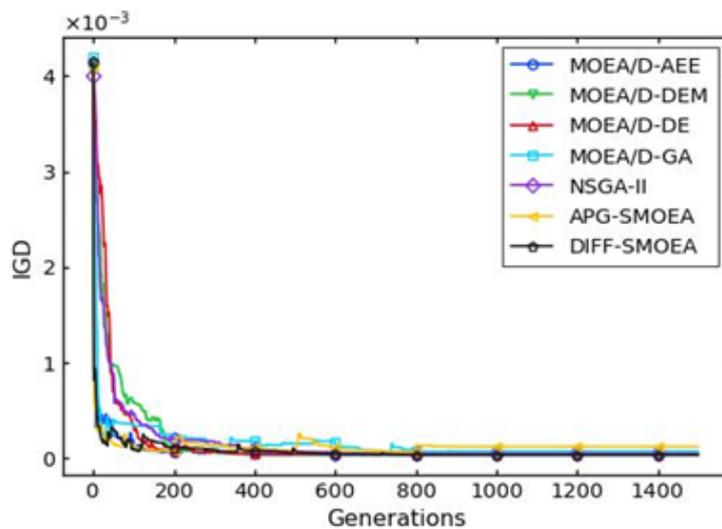


Figure 2. Experiment 1: IGD metrics on DAX.

#### 4.4.2. Collaborative Experiments Adapted to the Cold-Start Mechanism

This section further designs a collaborative experiment adapted to the cold-start mechanism (hereafter referred to as Experiment II). Based on the analysis of the results of Experiment I, it is more clear that there is a need to use a new synergetic approach for synergetic method research on explicitly solving PDFs. This experiment two involved a higher iteration number  $g$ . Although a comparison of experiments showed that when  $g$  is 1000, the performance is better, because of space reasons, this section does not show the iteration parameter  $g$  from the comparison of experimental results. Figure 3 shows the change in the IGD of different algorithms on the low-dimensional Hang Seng dataset, and it can be seen that the DPG-SMOEA can converge quickly and it has an excellent diversity and convergence, which also shows that the DPG-SMOEA is better in terms of comprehensive performance. Figure 4 shows the population distribution of different algorithms on the high-dimensional Nikkei dataset. Although the diffusion model is not applicable to the analysis of high-dimensional data, the optimization of the collaborative approach makes its population distribution comparable to that of the MOEA/D-AEE, especially when using the trained diffusion model for the generation of offspring at the late stage of iterations, which enables an exploration of the development of the solution space continuously. There are no problems in the process of converging to the Pareto front. The Pareto front does not stopping the exploration of the solution set, and the obtained solution set conforms to the

distribution of the Pareto front perfectly. This also shows that the DPG-SMOEA has an excellent population diversity even in a high-dimensional data analysis, suggesting that the diffusion model with its fine-grained generation can solve the problem of uncertainty in random sampling methods well.

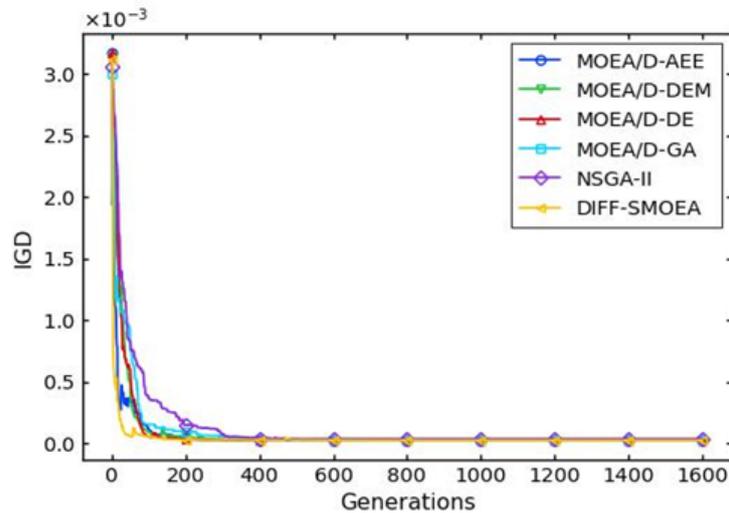


Figure 3. Experiment 2: IGD metrics on the Hang Seng dataset.

Table A3 shows the comparative experimental results of Experiment 2 on the 31-dimensional Hang Seng dataset, Table A4 shows the comparative experimental results of Experiment 2 on the 89-dimensional FTSE dataset, and Table A5 shows the comparative experimental results of Experiment 2 on the 225-dimensional Nikkei dataset. Through an analysis, it is found that for the six experimental test metrics, the DPG-SMOEA does not achieve any optimal test metrics on the highest-dimensional Nikkei dataset, the DPG-SMOEA achieves one optimal metric with HV as the co-optimal one on the intermediate-dimensional FTSE dataset, and it achieves four optimal metrics on the low-dimensional Hang Seng dataset. It can be clearly concluded from Experiment 2 that the DPG-SMOEA performs poorly on high-dimensional datasets and performs increasingly better as the dimensionality decreases, especially on the 31-dimensional Hang Seng dataset, where it obtains four optimal metrics. The experimental results fully demonstrate that the DPG-SMOEA, a synergistic method based on explicitly solving PDFs, is effective and clear regarding the analysis of low-dimensional data.

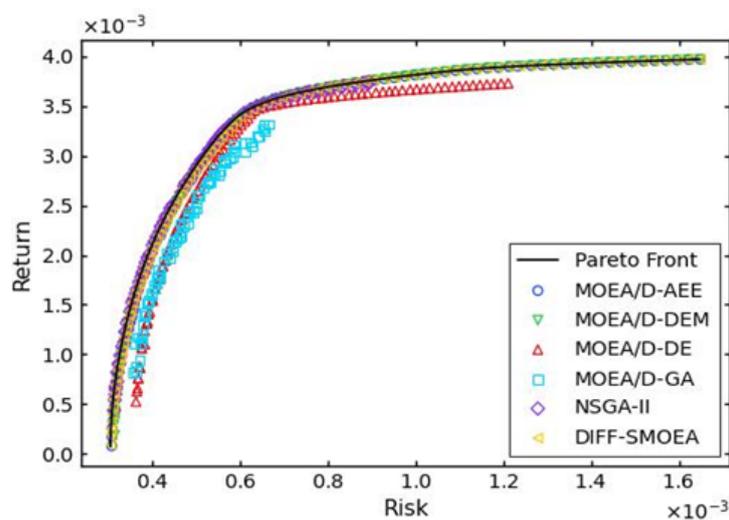


Figure 4. Experiment 2: Final population distribution for the Nikkei index.

## 5. Conclusions

In this paper, the AEE algorithm was first used to optimize the population, accumulating some high-quality solutions that were stored in the mixed optimization pool. Subsequently, an attempt was made to use these solutions to train a diffusion model, aiming to further enhance the quality of the solutions through this advanced generative model. However, the results did not meet expectations. Experimental observations revealed that the solutions generated after training with the diffusion model did not exceed the quality of the best solutions obtained through the AEE method. Upon analyzing the reasons for this, it was noted that in high-dimensional datasets, due to certain dimensions having small values, the noise introduced by the diffusion model during the generation process may cause significant disturbances to these small values, which could be one of the reasons for the suboptimal performance of the model. This paper also delved into the applicability of the diffusion model in handling multi-objective evolutionary algorithms (MOEAs). In traditional algorithms, solutions are generated by selecting two solutions as parents and then producing offspring through genetic crossover and mutation operations. While this method is simple, it does not always guarantee better solutions. On the contrary, the working mechanism of the diffusion model focuses more on learning and imitating the patterns of existing solutions. In MOEA problems, even minor differences can lead to significant variations in results.

Due to the cold-start mechanism of the diffusion model, the collaborative approach that involved the diffusion model too early in the iterative process did not leverage the advantages of the diffusion model but instead highlighted its significant disadvantages. Therefore, subsequent designs of new collaborative patterns involved minimal use of the diffusion model in the early stages, optimizing the strategy of the mixed memory pool and collecting high-quality solutions as much as possible to train the diffusion model. The diffusion model was introduced in later iterations to generate offspring. The final experimental results demonstrated an outstanding performance on the low-dimensional Hang Seng dataset and a moderate performance on the high-dimensional Nikkei dataset, with a clear trend of a better performance at lower dimensions, which fully aligns with the theoretical inference of this paper. The related research has been fruitful.

The article primarily focuses on the enhancements of MOEAs, solely employing the unconstrained standard Markowitz problem, without addressing its NP-hard variants. The research in this area is still relatively limited. From a theoretical perspective, the inclusion of constraints enables algorithms to provide effective solutions to the NP-hard problem of the extended mean-variance model. From a practical standpoint, the addition of these constraints brings the model closer to the real world, making it more effective in solving real-world portfolio problems.

The effectiveness of the algorithm is validated in this paper. Subsequent validation of the algorithm can be conducted using A-share data, and the financial attributes of the actual asset portfolios generated by the algorithm can be analyzed, including objectives such as annual dividends, the probability that the portfolio will deliver positive returns relative to a benchmark, and the Sharpe ratio over a period of time.

This algorithm estimates risk using traditional volatility, which penalizes both excessively high and low returns and lacks consistency, among other issues (Artzner et al., 1999) [35]. Subsequent approaches could involve using metrics such as the conditional value at risk (cVaR) and the value at risk (VaR) to measure risk.

The algorithm assumes that all the population is normally distributed, which is an idealized case. The diffusion model needs to be optimized for different distributions.

**Author Contributions:** Conceptualization, W.Q.; Methodology, L.Y.; Software, W.Q., L.Y. and X.H.; Validation, W.Q.; Formal analysis, M.Y. and X.H.; Investigation, M.Y.; Resources, X.Y.; Data curation, M.Y.; Writing—original draft, M.Y.; Writing—review & editing, M.Y. and W.Q.; Visualization, W.Q.; Supervision, L.Y. and Z.D.; Project administration, L.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper is supported the National Natural Science Foundation of China((No.11801433): Research on sparse stochastic optimization theory and algorithms induced by financial system risk issues, 2023 Horizontal project of Xi'anJiaotong University: Dynamic estimation and optimization method of investor preference in intelligent investment, The 2021 Fujian Foreign Cooperation Project(No. 202110001): Research on Human Behavior Recognition Based on RFID and Deep Learning and Fundamental Research Funds for the Central Universities (No.20720210047): Intelligent Theory and Key Technologies of Optimized Operation of Regional Energy Internet Based on Edge Cloud Collaboration.

**Data Availability Statement:** Data are contained within the article.

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

## Appendix A. Algorithms

### Appendix A.1. The DPG-SMOEA Framework

---

#### Algorithm A1: DPG-SMOEA Framework

---

**Input:**  $T$  is the maximum number of iterations,  $n_r$  is the maximum number of offspring updates,  $g$  is the threshold for collaborative strategy;

**Initialize:**  $T = 1500$ ,  $n_r = 2$ ;

- 1: Read relevant parameters from the dataset
- 2: Randomly initialize the population
- 3: Calculate the initial parameters
- 4: Initiate the network model
- 5: **while**  $t < T$  **do**
- 6:   **for**  $x_i$  in population **do**
- 7:     Determine the mating pool  $b$  for generating offspring using the mutation operator
- 8:     **if**  $t < g$  **then**
- 9:       Perform AEE operation to generate offspring in (Equation (6))
- 10:    **else**
- 11:     Construct a mixed optimization pool and select better solutions using AEE as training samples
- 12:     Use the diffusion model for offspring generation (refer to Appendix A.2)
- 13:    **end if**
- 14:    Repair offspring
- 15:    Calculate various information of offsprings (e.g., return, risk, reference points)
- 16:    Calculate and update the initial parameters
- 17:    **for**  $b_i$  in neighbor of  $b$  **do**
- 18:     **if** the M,V of offspring is better than that of the parent  $b_i$  **then**
- 19:       Update populations by replace  $b_i$  with  $y$
- 20:        $update\_count + 1$
- 21:     **end if**
- 22:     **if**  $update\_count \geq n_r$  **then**
- 23:       Break;
- 24:     **end if**
- 25:    **end for**
- 26:    **end for**
- 27: **end while**

---

## Appendix A.2. Generating Offspring via the Diffusion Model

**Algorithm A2:** Generating Offspring via the Diffusion Model

**Input:** High-quality solutions from MOEA/D-AEE stored in the mixed optimization pool as training samples.

- 1: **for** epoch in total\_epoch **do**
- 2:   Apply forward noise addition to the solutions in the population (refer to Equation (14))
- 3:   Record the top ten individuals in the population
- 4:   Feed the noised solutions into the network for training
- 5:   Calculate the loss function
- 6:   Update the network parameters using backward gradient descent (refer to Equation (18))
- 7: **end for**
- 8: Use the trained network for inference to generate new offspring

## Appendix B. Tables

**Table A1.** Comparative experimental results of experiment 1 on Hang Seng dataset.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
GD	Best	$4.02 \times 10^{-6}$	$4.21 \times 10^{-6}$	$3.49 \times 10^{-6}$	$1.74 \times 10^{-6}$	$9.06 \times 10^{-6}$	$4.17 \times 10^{-6}$
	Median	$5.91 \times 10^{-6}$	$7.26 \times 10^{-6}$	$7.98 \times 10^{-6}$	<b><math>2.29 \times 10^{-6}</math></b>	$1.18 \times 10^{-5}$	$6.41 \times 10^{-6}$
	Std.	$1.13 \times 10^{-6}$	$2.22 \times 10^{-6}$	$1.18 \times 10^{-4}$	$2.60 \times 10^{-7}$	$1.34 \times 10^{-6}$	$6.00 \times 10^{-6}$
Spacing	Best	$1.65 \times 10^{-5}$	$1.50 \times 10^{-5}$	$8.51 \times 10^{-6}$	$9.38 \times 10^{-6}$	$3.94 \times 10^{-5}$	$1.60 \times 10^{-5}$
	Median	$2.05 \times 10^{-5}$	$2.34 \times 10^{-5}$	$1.80 \times 10^{-5}$	<b><math>1.53 \times 10^{-5}</math></b>	$4.89 \times 10^{-5}$	$2.23 \times 10^{-5}$
	Std.	$6.25 \times 10^{-6}$	$9.07 \times 10^{-6}$	$6.76 \times 10^{-6}$	$5.71 \times 10^{-6}$	$4.18 \times 10^{-6}$	$6.95 \times 10^{-6}$
MaxSpread	Best	$9.10 \times 10^{-3}$	$9.23 \times 10^{-3}$	$9.00 \times 10^{-3}$	$8.92 \times 10^{-3}$	$9.06 \times 10^{-3}$	$9.13 \times 10^{-3}$
	Median	<b><math>8.96 \times 10^{-3}</math></b>	$8.89 \times 10^{-3}$	$8.54 \times 10^{-3}$	$8.25 \times 10^{-3}$	$8.63 \times 10^{-3}$	$8.89 \times 10^{-3}$
	Std.	$9.91 \times 10^{-5}$	$1.98 \times 10^{-4}$	$9.93 \times 10^{-4}$	$3.16 \times 10^{-4}$	$2.18 \times 10^{-4}$	$4.48 \times 10^{-4}$
Delta	Best	$2.44 \times 10^{-1}$	$2.53 \times 10^{-1}$	$2.33 \times 10^{-1}$	$2.61 \times 10^{-1}$	$4.48 \times 10^{-1}$	$2.46 \times 10^{-1}$
	Median	<b><math>2.60 \times 10^{-1}</math></b>	$2.87 \times 10^{-1}$	$2.87 \times 10^{-1}$	$2.80 \times 10^{-1}$	$4.96 \times 10^{-1}$	$2.77 \times 10^{-1}$
	Std.	$2.37 \times 10^{-2}$	$3.99 \times 10^{-2}$	$8.20 \times 10^{-2}$	$1.40 \times 10^{-2}$	$3.50 \times 10^{-2}$	$5.72 \times 10^{-2}$
IGD	Best	$2.86 \times 10^{-5}$	$2.99 \times 10^{-5}$	$3.15 \times 10^{-5}$	$2.98 \times 10^{-5}$	$3.92 \times 10^{-5}$	$2.88 \times 10^{-5}$
	Median	<b><math>3.12 \times 10^{-5}</math></b>	$3.50 \times 10^{-5}$	$6.03 \times 10^{-5}$	$7.54 \times 10^{-5}$	$5.01 \times 10^{-5}$	$3.27 \times 10^{-5}$
	Std.	$2.29 \times 10^{-6}$	$8.54 \times 10^{-6}$	$2.44 \times 10^{-4}$	$3.97 \times 10^{-5}$	$1.55 \times 10^{-5}$	$5.38 \times 10^{-5}$
HV	Best	$1.18 \times 10^{-2}$					
	Median	<b><math>1.18 \times 10^{-2}</math></b>					
	Std.	$1.56 \times 10^{-8}$	$3.36 \times 10^{-8}$	$9.66 \times 10^{-4}$	$6.49 \times 10^{-7}$	$4.12 \times 10^{-6}$	$1.67 \times 10^{-7}$

**Table A2.** Comparative experimental results of experiment 1 on DAX dataset.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
GD	Best	$6.29 \times 10^{-6}$	$7.11 \times 10^{-6}$	$7.32 \times 10^{-6}$	$1.83 \times 10^{-6}$	$5.37 \times 10^{-6}$	$6.26 \times 10^{-6}$
	Median	$8.07 \times 10^{-6}$	$9.53 \times 10^{-6}$	$1.65 \times 10^{-5}$	<b><math>2.78 \times 10^{-6}</math></b>	$8.04 \times 10^{-6}$	$8.77 \times 10^{-6}$
	Std.	$9.03 \times 10^{-7}$	$1.81 \times 10^{-6}$	$9.27 \times 10^{-5}$	$6.36 \times 10^{-7}$	$1.99 \times 10^{-6}$	$1.94 \times 10^{-6}$
Spacing	Best	$2.59 \times 10^{-5}$	$2.34 \times 10^{-5}$	$1.64 \times 10^{-5}$	$1.59 \times 10^{-5}$	$2.27 \times 10^{-5}$	$2.73 \times 10^{-5}$
	Median	$3.38 \times 10^{-5}$	$3.24 \times 10^{-5}$	$2.98 \times 10^{-5}$	<b><math>2.48 \times 10^{-5}</math></b>	$4.34 \times 10^{-5}$	$3.45 \times 10^{-5}$
	Std.	$6.04 \times 10^{-6}$	$7.37 \times 10^{-6}$	$6.61 \times 10^{-6}$	$5.68 \times 10^{-6}$	$6.95 \times 10^{-6}$	$5.97 \times 10^{-6}$
MaxSpread	Best	$8.11 \times 10^{-3}$	$8.12 \times 10^{-3}$	$8.36 \times 10^{-3}$	$7.37 \times 10^{-3}$	$7.83 \times 10^{-3}$	$8.19 \times 10^{-3}$
	Median	<b><math>7.78 \times 10^{-3}</math></b>	$7.71 \times 10^{-3}$	$7.40 \times 10^{-3}$	$6.04 \times 10^{-3}$	$7.20 \times 10^{-3}$	$7.68 \times 10^{-3}$
	Std.	$2.12 \times 10^{-4}$	$2.31 \times 10^{-4}$	$7.26 \times 10^{-4}$	$3.70 \times 10^{-4}$	$5.20 \times 10^{-4}$	$3.00 \times 10^{-4}$

Table A2. Cont.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
Delta	Best	$3.95 \times 10^{-1}$	$4.07 \times 10^{-1}$	$3.60 \times 10^{-1}$	$4.54 \times 10^{-1}$	$5.70 \times 10^{-1}$	$3.91 \times 10^{-1}$
	Median	<b><math>4.14 \times 10^{-1}</math></b>	$4.49 \times 10^{-1}$	$4.36 \times 10^{-1}$	$5.81 \times 10^{-1}$	$6.68 \times 10^{-1}$	$4.29 \times 10^{-1}$
	Std.	$2.18 \times 10^{-2}$	$3.79 \times 10^{-2}$	$7.72 \times 10^{-2}$	$3.65 \times 10^{-2}$	$5.08 \times 10^{-2}$	$3.26 \times 10^{-2}$
IGD	Best	$3.41 \times 10^{-5}$	$3.62 \times 10^{-5}$	$4.40 \times 10^{-5}$	$7.20 \times 10^{-5}$	$4.14 \times 10^{-5}$	$3.33 \times 10^{-5}$
	Median	<b><math>4.16 \times 10^{-5}</math></b>	$4.90 \times 10^{-5}$	$9.48 \times 10^{-5}$	$1.54 \times 10^{-4}$	$6.55 \times 10^{-5}$	$4.67 \times 10^{-5}$
	Std.	$1.42 \times 10^{-5}$	$1.60 \times 10^{-5}$	$9.39 \times 10^{-5}$	$3.75 \times 10^{-5}$	$3.34 \times 10^{-5}$	$2.54 \times 10^{-5}$
HV	Best	$1.87 \times 10^{-2}$	$1.87 \times 10^{-2}$	$1.87 \times 10^{-2}$	$1.87 \times 10^{-2}$	$1.87 \times 10^{-2}$	$1.87 \times 10^{-2}$
	Median	<b><math>1.87 \times 10^{-2}</math></b>	<b><math>1.87 \times 10^{-2}</math></b>	<b><math>1.87 \times 10^{-2}</math></b>	$1.75 \times 10^{-2}$	$1.83 \times 10^{-2}$	<b><math>1.87 \times 10^{-2}</math></b>
	Std.	$1.46 \times 10^{-8}$	$3.08 \times 10^{-8}$	$9.62 \times 10^{-4}$	$4.47 \times 10^{-4}$	$5.68 \times 10^{-4}$	$4.79 \times 10^{-8}$

Table A3. Comparative experimental results of experiment 2 on Hang Seng dataset.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
GD	Best	$4.02 \times 10^{-6}$	$4.21 \times 10^{-6}$	$3.49 \times 10^{-6}$	$1.74 \times 10^{-6}$	$9.06 \times 10^{-6}$	$1.81 \times 10^{-6}$
	Median	$5.91 \times 10^{-6}$	$7.26 \times 10^{-6}$	$7.98 \times 10^{-6}$	<b><math>2.29 \times 10^{-6}</math></b>	$1.18 \times 10^{-5}$	$5.26 \times 10^{-6}$
	Std.	$1.13 \times 10^{-6}$	$2.22 \times 10^{-6}$	$1.18 \times 10^{-4}$	$2.60 \times 10^{-7}$	$1.34 \times 10^{-6}$	$9.89 \times 10^{-7}$
Spacing	Best	$1.65 \times 10^{-5}$	$1.50 \times 10^{-5}$	$8.51 \times 10^{-6}$	$9.38 \times 10^{-6}$	$3.94 \times 10^{-5}$	$1.63 \times 10^{-5}$
	Median	$2.05 \times 10^{-5}$	$2.34 \times 10^{-5}$	$1.80 \times 10^{-5}$	<b><math>1.53 \times 10^{-5}</math></b>	$4.89 \times 10^{-5}$	$1.95 \times 10^{-5}$
	Std.	$6.25 \times 10^{-6}$	$9.07 \times 10^{-6}$	$6.76 \times 10^{-6}$	$5.71 \times 10^{-6}$	$4.18 \times 10^{-6}$	$5.85 \times 10^{-6}$
Max Spread	Best	$9.10 \times 10^{-3}$	$9.23 \times 10^{-3}$	$9.00 \times 10^{-3}$	$8.92 \times 10^{-3}$	$9.06 \times 10^{-3}$	$9.16 \times 10^{-3}$
	Median	$8.96 \times 10^{-3}$	$8.89 \times 10^{-3}$	$8.54 \times 10^{-3}$	$8.25 \times 10^{-3}$	$8.63 \times 10^{-3}$	<b><math>8.99 \times 10^{-3}</math></b>
	Std.	$9.91 \times 10^{-5}$	$1.98 \times 10^{-4}$	$9.93 \times 10^{-4}$	$3.16 \times 10^{-4}$	$2.18 \times 10^{-4}$	$7.85 \times 10^{-5}$
Delta	Best	$2.44 \times 10^{-1}$	$2.53 \times 10^{-1}$	$2.33 \times 10^{-1}$	$2.61 \times 10^{-1}$	$4.48 \times 10^{-1}$	$2.44 \times 10^{-1}$
	Median	$2.60 \times 10^{-1}$	$2.87 \times 10^{-1}$	$2.87 \times 10^{-1}$	$2.80 \times 10^{-1}$	$4.96 \times 10^{-1}$	<b><math>2.55 \times 10^{-1}</math></b>
	Std.	$2.37 \times 10^{-2}$	$3.99 \times 10^{-2}$	$8.20 \times 10^{-2}$	$1.40 \times 10^{-2}$	$3.50 \times 10^{-2}$	$1.87 \times 10^{-2}$
IGD	Best	$2.86 \times 10^{-5}$	$2.99 \times 10^{-5}$	$3.15 \times 10^{-5}$	$2.98 \times 10^{-5}$	$3.92 \times 10^{-5}$	$2.80 \times 10^{-5}$
	Median	$3.12 \times 10^{-5}$	$3.50 \times 10^{-5}$	$6.03 \times 10^{-5}$	$7.54 \times 10^{-5}$	$5.01 \times 10^{-5}$	<b><math>3.02 \times 10^{-5}</math></b>
	Std.	$2.29 \times 10^{-6}$	$8.54 \times 10^{-6}$	$2.44 \times 10^{-4}$	$3.97 \times 10^{-5}$	$1.55 \times 10^{-5}$	$1.59 \times 10^{-6}$
HV	Best	$2.64 \times 10^{-5}$	$2.64 \times 10^{-5}$	$2.64 \times 10^{-5}$	$2.64 \times 10^{-5}$	$2.63 \times 10^{-5}$	$2.64 \times 10^{-5}$
	Median	<b><math>2.64 \times 10^{-5}</math></b>	$2.63 \times 10^{-5}$	$2.63 \times 10^{-5}$	$2.64 \times 10^{-5}$	$2.63 \times 10^{-5}$	<b><math>2.64 \times 10^{-5}</math></b>
	Std.	$1.22 \times 10^{-8}$	$2.64 \times 10^{-8}$	$2.21 \times 10^{-6}$	$1.38 \times 10^{-8}$	$1.31 \times 10^{-8}$	$1.07 \times 10^{-8}$

Table A4. Comparative experimental results of experiment 2 on FTSE dataset.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
GD	Best	$5.27 \times 10^{-6}$	$6.32 \times 10^{-6}$	$7.01 \times 10^{-6}$	$2.84 \times 10^{-6}$	$7.38 \times 10^{-6}$	$4.54 \times 10^{-6}$
	Median	$7.05 \times 10^{-6}$	$9.58 \times 10^{-6}$	$1.83 \times 10^{-5}$	<b><math>5.05 \times 10^{-6}</math></b>	$9.25 \times 10^{-6}$	$7.72 \times 10^{-6}$
	Std.	$7.28 \times 10^{-7}$	$2.41 \times 10^{-6}$	$1.55 \times 10^{-4}$	$1.60 \times 10^{-6}$	$9.50 \times 10^{-7}$	$8.27 \times 10^{-7}$
Spacing	Best	$1.71 \times 10^{-5}$	$1.38 \times 10^{-5}$	$9.87 \times 10^{-6}$	$1.14 \times 10^{-5}$	$2.39 \times 10^{-5}$	$1.69 \times 10^{-5}$
	Median	$2.07 \times 10^{-5}$	$2.01 \times 10^{-5}$	<b><math>1.72 \times 10^{-5}</math></b>	$1.97 \times 10^{-5}$	$2.99 \times 10^{-5}$	$2.05 \times 10^{-5}$
	Std.	$3.76 \times 10^{-6}$	$4.54 \times 10^{-6}$	$3.34 \times 10^{-6}$	$5.10 \times 10^{-6}$	$2.24 \times 10^{-6}$	$5.04 \times 10^{-6}$
Max Spread	Best	$5.96 \times 10^{-3}$	$5.79 \times 10^{-3}$	$5.74 \times 10^{-3}$	$5.47 \times 10^{-3}$	$5.67 \times 10^{-3}$	$5.84 \times 10^{-3}$
	Median	<b><math>5.56 \times 10^{-3}</math></b>	$5.40 \times 10^{-3}$	$5.15 \times 10^{-3}$	$4.91 \times 10^{-3}$	$5.45 \times 10^{-3}$	$5.42 \times 10^{-3}$
	Std.	$1.59 \times 10^{-4}$	$1.93 \times 10^{-4}$	$5.08 \times 10^{-4}$	$4.19 \times 10^{-4}$	$1.69 \times 10^{-4}$	$1.63 \times 10^{-4}$
Delta	Best	$4.01 \times 10^{-1}$	$4.25 \times 10^{-1}$	$4.21 \times 10^{-1}$	$4.27 \times 10^{-1}$	$5.47 \times 10^{-1}$	$4.01 \times 10^{-1}$
	Median	<b><math>4.33 \times 10^{-1}</math></b>	$4.72 \times 10^{-1}$	$4.51 \times 10^{-1}$	$5.05 \times 10^{-1}$	$6.06 \times 10^{-1}$	$4.53 \times 10^{-1}$
	Std.	$2.02 \times 10^{-2}$	$3.06 \times 10^{-2}$	$6.30 \times 10^{-2}$	$6.88 \times 10^{-2}$	$3.34 \times 10^{-2}$	$3.58 \times 10^{-2}$

Table A4. Cont.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
IGD	Best	$2.30 \times 10^{-5}$	$2.90 \times 10^{-5}$	$4.26 \times 10^{-5}$	$4.07 \times 10^{-5}$	$3.22 \times 10^{-5}$	$2.14 \times 10^{-5}$
	Median	<b><math>3.71 \times 10^{-5}</math></b>	$5.31 \times 10^{-5}$	$9.09 \times 10^{-5}$	$8.76 \times 10^{-5}$	$4.74 \times 10^{-5}$	$5.02 \times 10^{-5}$
	Std.	$1.14 \times 10^{-5}$	$2.16 \times 10^{-5}$	$1.47 \times 10^{-4}$	$4.33 \times 10^{-5}$	$1.38 \times 10^{-5}$	$1.35 \times 10^{-5}$
HV	Best	$1.37 \times 10^{-5}$	$1.37 \times 10^{-5}$	$1.37 \times 10^{-5}$	$1.37 \times 10^{-5}$	$1.37 \times 10^{-5}$	$1.37 \times 10^{-5}$
	Median	<b><math>1.37 \times 10^{-5}</math></b>	<b><math>1.37 \times 10^{-5}</math></b>	$1.36 \times 10^{-5}$	$1.34 \times 10^{-5}$	<b><math>1.37 \times 10^{-5}</math></b>	<b><math>1.37 \times 10^{-5}</math></b>
	Std.	$5.08 \times 10^{-9}$	$1.64 \times 10^{-8}$	$1.36 \times 10^{-6}$	$6.19 \times 10^{-7}$	$8.35 \times 10^{-8}$	$6.01 \times 10^{-9}$

Table A5. Comparative experimental results of experiment 2 on Nikkei dataset.

Metric		MOEA/D-AEE	MOEA/D-DEM	MOEA/D-DE	MOEA/D-GA	NSGA-II	DPG-SMOEA
GD	Best	$5.32 \times 10^{-6}$	$5.10 \times 10^{-6}$	$5.93 \times 10^{-5}$	$9.95 \times 10^{-6}$	$2.54 \times 10^{-6}$	$4.51 \times 10^{-6}$
	Median	$7.36 \times 10^{-6}$	$8.06 \times 10^{-6}$	$1.45 \times 10^{-4}$	$3.32 \times 10^{-5}$	<b><math>4.24 \times 10^{-6}</math></b>	$8.33 \times 10^{-6}$
	Std.	$9.19 \times 10^{-7}$	$1.66 \times 10^{-4}$	$7.09 \times 10^{-5}$	$2.08 \times 10^{-4}$	$2.01 \times 10^{-6}$	$1.34 \times 10^{-6}$
Spacing	Best	$1.42 \times 10^{-5}$	$1.28 \times 10^{-5}$	$5.99 \times 10^{-6}$	$0.00 \times 10^0$	$1.05 \times 10^{-5}$	$1.17 \times 10^{-5}$
	Median	$1.79 \times 10^{-5}$	$2.08 \times 10^{-5}$	<b><math>1.15 \times 10^{-5}</math></b>	$1.92 \times 10^{-5}$	$1.45 \times 10^{-5}$	$1.87 \times 10^{-5}$
	Std.	$4.87 \times 10^{-6}$	$5.77 \times 10^{-6}$	$4.53 \times 10^{-6}$	$9.72 \times 10^{-6}$	$1.87 \times 10^{-6}$	$8.82 \times 10^{-6}$
Max Spread	Best	$4.17 \times 10^{-3}$	$4.23 \times 10^{-3}$	$4.29 \times 10^{-3}$	$2.63 \times 10^{-3}$	$3.36 \times 10^{-3}$	$4.11 \times 10^{-3}$
	Median	$3.93 \times 10^{-3}$	<b><math>3.94 \times 10^{-3}</math></b>	$2.96 \times 10^{-3}$	$2.20 \times 10^{-3}$	$2.88 \times 10^{-3}$	$3.80 \times 10^{-3}$
	Std.	$1.21 \times 10^{-4}$	$5.39 \times 10^{-4}$	$5.48 \times 10^{-4}$	$4.65 \times 10^{-4}$	$2.54 \times 10^{-4}$	$1.67 \times 10^{-4}$
Delta	Best	$3.87 \times 10^{-1}$	$3.99 \times 10^{-1}$	$3.17 \times 10^{-1}$	$8.40 \times 10^{-1}$	$6.09 \times 10^{-1}$	$3.78 \times 10^{-1}$
	Median	<b><math>4.42 \times 10^{-1}</math></b>	$4.81 \times 10^{-1}$	$5.58 \times 10^{-1}$	$9.34 \times 10^{-1}$	$6.81 \times 10^{-1}$	$4.87 \times 10^{-1}$
	Std.	$6.09 \times 10^{-2}$	$9.05 \times 10^{-2}$	$1.00 \times 10^{-1}$	$3.55 \times 10^{-2}$	$2.94 \times 10^{-2}$	$1.04 \times 10^{-1}$
IGD	Best	$1.72 \times 10^{-5}$	$1.90 \times 10^{-5}$	$7.90 \times 10^{-5}$	$1.77 \times 10^{-4}$	$4.64 \times 10^{-5}$	$1.63 \times 10^{-5}$
	Median	<b><math>2.47 \times 10^{-5}</math></b>	$2.73 \times 10^{-5}$	$2.23 \times 10^{-4}$	$2.41 \times 10^{-4}$	$9.69 \times 10^{-5}$	$3.51 \times 10^{-5}$
	Std.	$7.05 \times 10^{-6}$	$4.09 \times 10^{-4}$	$6.95 \times 10^{-5}$	$5.29 \times 10^{-4}$	$3.67 \times 10^{-5}$	$1.59 \times 10^{-5}$
HV	Best	$8.31 \times 10^{-6}$	$8.29 \times 10^{-6}$	$7.96 \times 10^{-6}$	$7.87 \times 10^{-6}$	$8.19 \times 10^{-6}$	$8.32 \times 10^{-6}$
	Median	<b><math>8.29 \times 10^{-6}</math></b>	$8.26 \times 10^{-6}$	$7.23 \times 10^{-6}$	$7.54 \times 10^{-6}$	$7.94 \times 10^{-6}$	$8.27 \times 10^{-6}$
	Std.	$1.06 \times 10^{-8}$	$9.52 \times 10^{-7}$	$3.43 \times 10^{-7}$	$1.20 \times 10^{-6}$	$1.08 \times 10^{-7}$	$1.75 \times 10^{-8}$

References

1. Markowitz, H. Portfolio Selection. *J. Financ.* **1952**, *7*, 77–91.
2. Metaxiotis, K.; Liagkouras, K. Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Syst. Appl.* **2012**, *39*, 11685–11698. [CrossRef]
3. Zhang, Q.; Li, H.; Maringer, D.; Tsang, E. MOEA/D with NBI-style Tchebycheff approach for portfolio management. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–8. [CrossRef]
4. Qian, W.; Liu, J.; Lin, Y.; Yang, L.; Zhang, J.; Xu, H.; Liao, M.; Chen, Y.; Chen, Y.; Liu, B. An improved MOEA/D algorithm for complex data analysis. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6393638. [CrossRef]
5. Nichol, A.; Dhariwal, P. Improved denoising diffusion probabilistic models. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: Cambridge MA, USA, 2021; pp. 8162–8171.
6. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
7. Song, J.; Meng, C.; Ermon, S. Denoising diffusion implicit models. *arXiv* **2020**, arXiv:2010.02502.
8. Ming, F.; Gong, W.; Wang, L.; Gao, L. A Constraint-Handling Technique for Decomposition-Based Constrained Many-Objective Evolutionary Algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7783–7793. [CrossRef]
9. Asafuddoula, M.; Ray, T.; Sarker, R. A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 445–460. [CrossRef]
10. Coello, C.A.C. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–36. [CrossRef]
11. Zhu, H.; Chen, Q.; Ding, J.; Zhang, X.; Wang, H. Parameter-Adaptive Paired Offspring Generation for Constrained Large-Scale Multiobjective Optimization Algorithm. In Proceedings of the 2023 IEEE Symposium Series on Computational Intelligence (SSCI), Mexico City, Mexico, 5–8 December 2023; pp. 470–475.
12. Zhang, S.; Yang, T.; Liang, J.; Yue, C. A Novel Adaptive Bandit-Based Selection Hyper-Heuristic for Multiobjective Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7693–7706. [CrossRef]

13. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
14. 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.* **2014**, *18*, 577–601. [[CrossRef](#)]
15. Shui, Y.; Li, H.; Sun, J.; Zhang, Q. The Combination of MOEA/D and WOF for Solving High-Dimensional Expensive Multiobjective Optimization Problems. In Proceedings of the 2023 IEEE Congress on Evolutionary Computation (CEC), Chicago, IL, USA, 1–5 July 2023; pp. 1–8.
16. He, L.; Shang, K.; Nan, Y.; Ishibuchi, H.; Srinivasan, D. Relation Between Objective Space Normalization and Weight Vector Scaling in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2023**, *27*, 1177–1191. [[CrossRef](#)]
17. Salih, A.; Moshaiov, A. Promoting Transfer of Robot Neuro-Motion-Controllers by Many-Objective Topology and Weight Evolution. *IEEE Trans. Evol. Comput.* **2023**, *27*, 385–395. [[CrossRef](#)]
18. Zheng, W.; Sun, J.; Zhang, Q.; Xu, Z. Continuous Encoding for Overlapping Community Detection in Attributed Network. *IEEE Trans. Cybern.* **2023**, *53*, 5469–5482. [[CrossRef](#)] [[PubMed](#)]
19. Trivedi, A.; Srinivasan, D.; Sanyal, K.; Ghosh, A. A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition. *IEEE Trans. Evol. Comput.* **2017**, *21*, 440–462. [[CrossRef](#)]
20. Zhao, S.Z.; Suganthan, P.N.; Zhang, Q. Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes. *IEEE Trans. Evol. Comput.* **2012**, *16*, 442–446. [[CrossRef](#)]
21. Bienstock, D. Computational study of a family of mixed-integer quadratic programming problems. *Math. Program.* **1996**, *74*, 121–140. [[CrossRef](#)]
22. Shaw, D.X.; Liu, S.; Kopman, L. Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optim. Methods Softw.* **2008**, *23*, 411–420. [[CrossRef](#)]
23. Arnone, S.; Loraschi, A.; Tettamanzi, A. A genetic approach to portfolio selection. *Neural Netw. World—Int. J. Neural MassParallel Comput. Inf. Syst.* **1993**, *3*, 597–604.
24. Anagnostopoulos, K.P.; Mamanis, G. The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Syst. Appl.* **2011**, *38*, 14208–14217. [[CrossRef](#)]
25. Khin, L.; Qu, R.; Kendall, G. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Appl. Soft Comput.* **2014**, *24*, 757–772.
26. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.H. Diffusion Models: A comprehensive survey of methods and applications. *ACM Comput. Surv.* **2023**, *56*, 1–39. [[CrossRef](#)]
27. Song, Y.; Ermon, S. Generative modeling by estimating gradients of the data distribution. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1415–1428.
28. Song, Y.; Durkan, C.; Murray, I.; Ermon, S. Maximum likelihood training of score-based Diffusion Models. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1415–1428.
29. Hyvärinen, A.; Dayan, P. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.* **2005**, *6*, 695–709.
30. Siddique, N.; Adeli, H. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*; John Wiley, Sons: Hoboken, NJ, USA, 2013.
31. Yu, G.; Chai, T.; Luo, X. Two-level production plan decomposition based on a hybrid MOEA for mineral processing. *IEEE Trans. Autom. Sci. Eng.* **2012**, *10*, 1050–1071. [[CrossRef](#)]
32. 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](#)]
33. Orlova, E. Decision-Making Techniques for Credit Resource Management Using Machine Learning and Optimization. *Information* **2020**, *11*, 144. [[CrossRef](#)]
34. Zhang, J.; Liang, C.; Lu, Q. A novel small-population Genetic Algorithm based on adaptive mutation and population entropy sampling. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; IEEE: Piscataway, NJ, USA, 2008.
35. Artzner, P.; Delbaen, F.; Eber, J.; Heath, D. Coherent measures of risk. *Math. Financ.* **2011**, *38*, 14208–14217.

**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.