


Article

A Multi-Strategy Adaptive Particle Swarm Optimization Algorithm for Solving Optimization Problem

Yingjie Song¹, Ying Liu², Huayue Chen^{3,*} and Wu Deng^{4,5,*} 

¹ School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China

² School of Statistics, Shandong Technology and Business University, Yantai 264005, China

³ School of Computer Science, China West Normal University, Nanchong 637002, China

⁴ Traction Power State Key Laboratory, Southwest Jiaotong University, Chengdu 610031, China

⁵ College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

* Correspondence: sunnyxiaoyue20@cwnu.edu.cn (H.C.); wdeng@cauc.edu.cn (W.D.)

Abstract: In solving the portfolio optimization problem, the mean-semivariance (MSV) model is more complicated and time-consuming, and their relations are unbalanced because they conflict with each other due to return and risk. Therefore, in order to solve these existing problems, multi-strategy adaptive particle swarm optimization, namely APSO/DU, has been developed to solve the portfolio optimization problem. In the present study, a constraint factor is introduced to control velocity weight to reduce blindness in the search process. A dual-update (DU) strategy is based on new speed, and position update strategies are designed. In order to test and prove the effectiveness of the APSO/DU algorithm, test functions and a realistic MSV portfolio optimization problem are selected here. The results demonstrate that the APSO/DU algorithm has better convergence accuracy and speed and finds the least risky stock portfolio for the same level of return. Additionally, the results are closer to the global Pareto front (PF). The algorithm can provide valuable advice to investors and has good practical applications.

Keywords: PSO; multi-strategy; dual-update strategy; mean-semivariance model; portfolio optimization



Citation: Song, Y.; Liu, Y.; Chen, H.; Deng, W. A Multi-Strategy Adaptive Particle Swarm Optimization Algorithm for Solving Optimization Problem. *Electronics* **2023**, *12*, 491. <https://doi.org/10.3390/electronics12030491>

Academic Editor: Young-Koo Lee

Received: 22 December 2022

Revised: 14 January 2023

Accepted: 16 January 2023

Published: 17 January 2023



Copyright: © 2023 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

The portfolio optimization problem (POP) aims to improve portfolio returns and reduce portfolio risk in the complex financial market. The mean-variance (MV) model was first proposed by economist Markowitz in 1952 to calculate the POP [1,2] and is a cornerstone of financial theory, providing a theoretical basis for investors to choose the optimal portfolio. However, there are significant limitations in its practical application. The use of variance to assess risk usually requires the calculation of a covariance matrix for all stocks, which is difficult to use in practice due to its computational complexity. Additionally, this risk measurement only considers the extent to which actual returns deviate from expected returns, whereas true losses refer to fluctuations below the mean of returns [3–8]. In order to be more in line with social reality, mean-semivariance portfolio models have been proposed and are widely used [9–12].

Traditional optimization algorithms for solving POPs require the application of many complex statistical methods and reference variables provided by experts, so solving large-scale POPs suffers from slow computational speed and poor solution accuracy, while heuristic algorithms can solve these problems well. In recent years, many scholars have used evolutionary computation algorithms to solve POPs, including the genetic algorithm (GA) [13], particle swarm optimization (PSO) [14,15], artificial bee colony algorithm (ABC) [16], and squirrel search algorithm (SSA) [17]. The particle swarm optimization algorithm (Eberhart & Kennedy, 1995) belongs to a class of swarm intelligence algorithms, which are designed by simulating the predatory behavior of a flock of birds [18–23]. Due

to its simple structure, fast convergence, and good robustness, it has been widely used in complex nonlinear portfolio optimization [24–29]. In addition, some new methods have also been proposed in some fields in recent years [30–39].

The improvement directions of the PSO algorithm are mainly divided into parameter improvement, update formula improvement, and integration with other intelligent algorithms. Setting the algorithm's parameters is the key to ensuring the reliability and robustness of the algorithm. With the determined population size and iteration time, the search capability of the algorithm is mainly decided by three core control parameters, namely the inertia weight (w), the self-learning factor (C_1), and the social-learning factor (C_2). To improve the performance of the algorithm, PSO algorithms based on the dual dynamic adaptation mechanism of inertia weights and learning factors have been proposed successively in recent years [40–42], considering that adjusting the core parameters alone weakens the uniformity of the algorithm evolution process and make it difficult to adapt to complex nonlinear optimization problems. Clerc et al. [43] proposed the concept of the shrinkage factor, and this method adds a multiplicative factor to the velocity formulation in order to allow the three core parameters to be tuned simultaneously, ultimately resulting in better algorithm convergence performance. Since then, numerous scholars have explored the full-parameter-tuning strategy to mix the three core parameters for tuning experiments. Zhang et al. [44] used control theory to optimize the core parameters of the standard PSO. Harrison et al. [45] empirically investigated the convergence behavior of 18 adaptive optimization algorithms.

The parameter improvement of PSO only involves improving the velocity update and does not consider the position update. Different position-updating strategies have different exploration and exploitation capabilities. In position updating, because the algorithm's convergence is highly dependent on the position weighting factor, a constraint factor needs to be introduced to control the velocity weight and reduce blindness in the search process. Liu et al. [46] proposed that the position weighting factor facilitates global algorithm exploration. The paper synthesizes the advantages of the two improvement methods and proposes a dual-update (DU) strategy. The method not only adjusts the core parameters of velocity update to make the algorithm more adaptable to nonlinear complex optimization problems, it also considers the position update formula and introduces a constraint factor to control the weight of velocity to reduce blindness in the search process and improve the convergence accuracy and convergence speed of the algorithm.

The main contributions of this paper are described as follows.

(1) This paper makes improvements based on fundamental particle swarm and proposes a multi-strategy adaptive particle swarm optimization algorithm, namely APSO/DU, to solve the portfolio optimization problem. Modern portfolio models are typically complex nonlinear functions, which are more challenging to solve.

(2) A dual-update strategy is designed based on new speed and position update strategies. The approach uses inertia weights to modify the learning factor, which can balance the capacity for learning individual particles and the capacity for learning the population and enhance the algorithm's optimization accuracy.

(3) A position update approach is also considered to lessen search blindness and increase the algorithm's convergence rate.

(4) Experimental findings show that the two strategies work better together than they do separately.

2. Multi-Strategy Adaptive PSO

2.1. Basic PSO Algorithm

The PSO algorithm is a population-based stochastic search algorithm in which the position of each particle represents a feasible solution to the problem to be optimized, and the position of the particle is evaluated in terms of its merit by the fitness value derived from the optimization function. The particle population is initialized randomly as a set of random candidate solutions in the PSO algorithm, and then each particle moves in the

search space with a certain speed, which is dynamically adjusted according to its own and its companion’s flight experience. The optimal solution is obtained after cyclic iterations until the convergence condition is satisfied.

Suppose a population $X = \{x_1, \dots, x_i, \dots, x_n\}$ of n particles without weight and volume in a D -dimensional search space, at the t th iteration, $x_i(t) = [x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t)]$ denotes the position of i th particle, $V_i(t) = [v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t)]$ denotes the velocity of i th particle. Up to generation t , $p_i(t) = [pbest_{i1}(t), pbest_{i2}(t), \dots, pbest_{iD}(t)]$ denotes the personal best position particle i has visited since the first-time step. g_{best} denote the best position discovered by all particles so far. In every generation, the evolution process of the i th particle is formulated as

$$v_i(t + 1) = wv_i(t) + c_1 \times rand() \times (p_i(t) - x_i(t)) + c_2 \times rand() \times (g_{best}(t) - x_i(t)) \quad (1)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2)$$

where $i = 1, 2, \dots, D$. w is the inertia weight. c_1 and c_2 are constants of the PSO algorithm with a value range of $[0, 2]$, while $rand()$ represents the random numbers in $[0, 1]$.

An iteration of PSO-based particle movement is demonstrated in Figure 1.

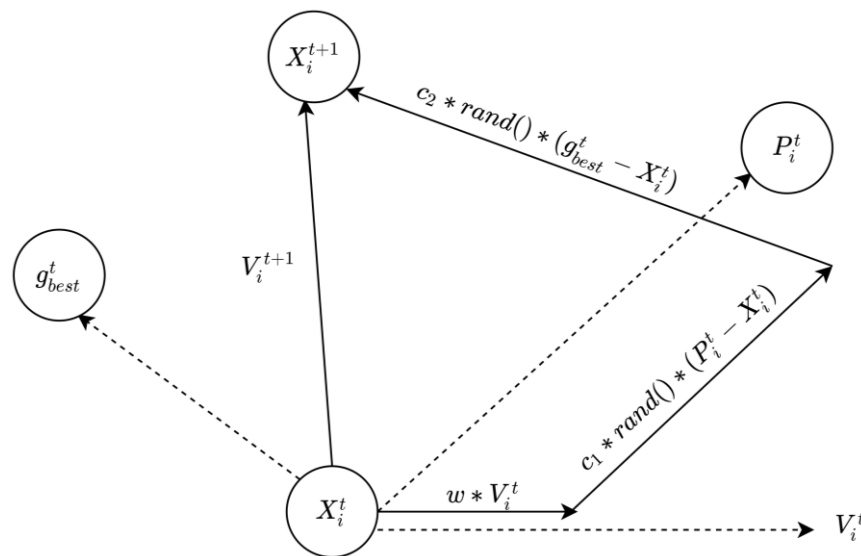


Figure 1. An iterative particle movement in PSO.

2.2. APSO/DU

PSO is an intelligent algorithm with global convergence, which requires fewer parameters to be adjusted. However, basic PSO has the problem of easily falling into local optimum and slow convergence. The APSO/DU algorithm can reduce the blindness of the search process and improve the convergence accuracy and speed of the algorithm, making it more adaptable to complex optimization problems. The APSO/DU algorithm can reduce the blindness of the search process and make the algorithm more adaptable to complex optimization problems.

2.2.1. Speed Update Strategy

The improvement strategies for inertia weights (w) and learning factors (c_1, c_2) can be classified as constant or stochastic, linear or nonlinear, and adaptive. The existing research on the dual dynamic adaptation mechanism has experimentally shown that using nonlinear decreasing weights is better than using linear decreasing weights. The functional relationship with nonlinear learning factors can be more adapted to complex optimization objectives. The strategy uses inertia weights to adjust the learning factors, which can balance the learning ability of individual particles and the group’s learning ability and

improve the algorithm's optimization accuracy. This paper uses a combination of the two with better results.

- Nonlinear Decreasing w

w is the core parameter that affects the performance and efficiency of the PSO algorithm. Smaller weights can strengthen the local search ability and improve convergence accuracy, while larger weights are beneficial to the global search and prevent the particles from falling into the optimal local position, but the convergence speed is slow. Most of the current improvements are related to the adjustment of w . In this paper, we use the nonlinear w exponential function decreasing way, and the formula is as follows.

$$w = w_{min} + (w_{max} - w_{min}) \times \exp \left[-20 \times \left(\frac{t}{T} \right)^6 \right] \quad (3)$$

where T is the maximum number of time steps, usually $w_{max} = 0.9$, $w_{min} = 0.4$.

- The learning factor (c_1 , c_2) varies according to w

c_1 and c_2 in the velocity update formula determine the size of the amount of learning of the particle in the optimal position. c_1 is used to adjust the amount of self-learning of the particle and c_2 is used to adjust the amount of social learning of the particle, and the change of the learning factor coefficient is used to change the trajectory of the particle. In this paper, referring to the previous summary, the adjustment strategy is better when the learning factor and inertia weights are a nonlinear function. The coefficient combination is $A = 0.5$, $B = 1$, $C = 0.5$, and the formula is described as follows.

$$\begin{aligned} C_1 &= Aw^2 + Bw + C \\ C_2 &= 2.5 - C_1 \end{aligned} \quad (4)$$

2.2.2. Position Update Policy

The convergence and convergence speed of the algorithm are greatly related to the position weighting factor, and the core parameter-tuning strategy only considers improving the velocity update without considering the position update. In order to control the influence of velocity on position, the constraint factor (α) is added to the position update formula, and α is introduced in order to achieve the weight of the control velocity to reduce blindness in the search process and improve the convergence rate.

- The Constraint Factors

In basic PSO, the new position of a particle is equal to its current position plus the current velocity, but the position vector and velocity vector cannot be added directly, so there must be a constraint factor between the two in the position update formula, and the constraint factor in the traditional PSO algorithm is equal to 1. α guides the particle to hover around the best position, and the improvement of α controls the influence of velocity on position so that the convergence of the algorithm is better improved. α based on w change is used in this paper which, in the early stage, is influenced by particle velocity and has strong exploration ability. In the later stage, it is less influenced by particle velocity and has strong local search ability.

$$\begin{aligned} x_{ij}(t+1) &= x_{ij}(t) + \alpha v_{ij}(t+1) \\ \alpha &= 0.1 + w \end{aligned} \quad (5)$$

2.2.3. Model of APSO/DU

The flow of the APSO/DU is shown in Figure 2.

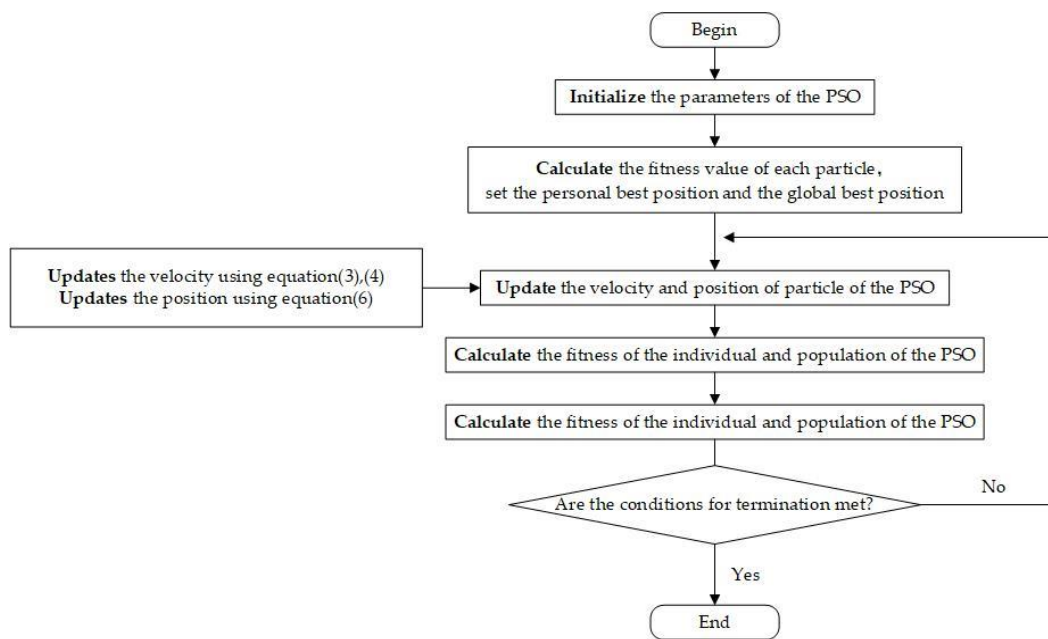


Figure 2. The flow of the APSO/DU.

2.3. Numerical Experiments and Analyses

In order to test the performance of the APSO/DU algorithm, three commonly used test functions were selected for the experiment. The test functions are shown in Table 1.

Table 1. Three test functions.

Selection	Function	Search Range	Global Optimum
Sphere	$f_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]$	$\min f_1 = 0$
Schwefel/sp2.22	$f_2 = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]$	$\min f_2 = 0$
Griewank	$f_3 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	$[-600, 600]$	$\min f_3 = 0$

- Contrast algorithms

The parameters of each PSO algorithm are shown in Table 2. To facilitate the comparison of the effectiveness of the APSO/DU algorithm, this paper chose to compare it with three classical adaptive improved PSO algorithms: PSO-TVIW; PSO-TVAC; and PSO-CF. The parameter settings summarized in the literature of Kyle Robert Harrison (2018) [45] were also used, where the time-varying inertia weight values of the PSO-TVIW algorithm are set according to the study in Harrison’s paper. The PSO-TVIW algorithm is also known as the standard particle swarm algorithm. The PSO-TVAC algorithm with time-varying acceleration coefficient adjusts the values of the w , c_1 , and c_2 parameters and introduces six additional control parameters. Clerc’s proposed PSO algorithm with shrinkage factor (PSOCF) has good convergence, but its computational accuracy is not high and its stability is not as good as that of standard PSO, so Eberhart proposed to limit the speed parameter $V_{max} = X_{max}$ of the algorithm so as to improve the convergence speed and search performance of the algorithm, and the PSOCF algorithm used this improved method for comparison experiments.

The new algorithm is based on a combination of two strategies. In order to verify whether the combination of two strategies is superior to one strategy, namely PSO/D (which updates only the core parameters), the formula and parameters are detailed in Section 2.2.1. Additionally, PSO/U, which only updates the velocity update formula, is

improved is by adding a constraint factor to the position formula, which needs to be combined with inertia weights. The basic particle swarm does not contain inertia weights, so the standard particle swarm algorithm (PSO-TVIW), by adding a constraint factor, can verify that the combination of update strategies proposed in this paper is superior.

Table 2. Parameter setting of each PSO algorithm.

Algorithm	w	c_1	c_2
PSO-TVAC	[0.4, 0.9]	[0.5, 2.5]	[0.5, 2.5]
PSO-TVIW	[0.4, 0.9]	1.49618	1.49618
PSOCF	0.729	2.8	1.3
PSO/D	[0.4, 0.9]	[0.695, 1.805]	[0.695, 1.805]
PSO/U	[0.4, 0.9]	1.49618	1.49618
APSO/DU	[0.4, 0.9]	[0.695, 1.805]	[0.695, 1.805]

In the experiments, to ensure fairness in the testing of each algorithm, different PSO algorithms were set with the same population size ($N = 30$), maximum number of iterations ($T_{\max} = 500$), and variable dimension ($D = 15$). Each algorithm was run 30 times, and the test results are shown in Table 3. The bold part of the text indicates the best optimization results.

- Test Results:

Table 3. Optimization results of six algorithms.

Function	Contrast Algorithms	f_{\min}	f_{mean}	f_{\max}	f_{sd}
F1	PSO – TVAC	2.26×10^{-3}	6.84×10^{-3}	1.37×10^{-2}	2.86×10^{-3}
	PSO – TVIW	1.22×10^{-3}	5.47×10^{-3}	1.16×10^{-2}	2.65×10^{-3}
	PSOCF	1.65×10^{-1}	1.90	6.08	1.69
	PSO/D	9.58×10^{-5}	4.92×10^{-3}	3.83×10^{-2}	8.08×10^{-3}
	PSO/U	3.36×10^{-3}	8.90×10^{-3}	2.10×10^{-2}	3.68×10^{-3}
	APSO/DU	4.57×10^{-5}	2.43×10^{-3}	1.37×10^{-2}	2.54×10^{-3}
F2	PSO – TVAC	2.0110	3.8211	7.2185	0.9937
	PSO – TVIW	1.7082	3.3503	5.0343	0.7761
	PSOCF	0.8047	2.5967	4.6286	0.9490
	PSO/D	0.4045	1.3387	3.0566	3.0566
	PSO/U	2.1516	3.6639	5.1408	5.1408
	APSO/DU	0.4246	1.3033	2.5632	0.5253
F3	PSO – TVAC	1.2174	1.8248	2.9463	4.03×10^{-1}
	PSO – TVIW	1.2724	1.9454	2.8534	4.06×10^{-1}
	PSOCF	1.0061	1.1614	1.5375	1.49×10^{-1}
	PSO/D	1.0018	1.0229	1.0959	2.57×10^{-2}
	PSO/U	1.5057	2.1557	3.3043	4.79×10^{-1}
	APSO/DU	0.8140	1.0119	1.0913	4.18×10^{-2}

It can be seen from Table 3 that APSO/DU outperforms the other algorithms overall. (i) The APSO/DU algorithm is compared with the classical adaptive algorithms (PSO-TVAC, PSO-TVIW, and PSOCF). APSO/DU takes the smallest optimal value in the three test functions and is closest to the optimal solution. The standard deviation is also the best among the three algorithms, which indicates that APSO/DU has a stable performance. (ii) To verify whether the combination of two strategies is better than one, the APSO/DU algorithm is compared with a single-strategy algorithm (PSO/D and PSO/U), and the results of PSO/U and APSO/DU are closer to each other. In the Griewank function, APSO/DU takes the smallest optimal value and is closest to the optimal solution with a standard deviation not much different from PSO/D. On balance, the APSO/DU algorithm outperforms the comparison algorithm.

In order to reflect more intuitively on the solution accuracy and convergence speed of each algorithm, the variation curves of the fitness values when each algorithm solves the three test functions are given in Figure 3. The horizontal coordinate indicates the number of iterations, and the vertical coordinate indicates the fitness value.

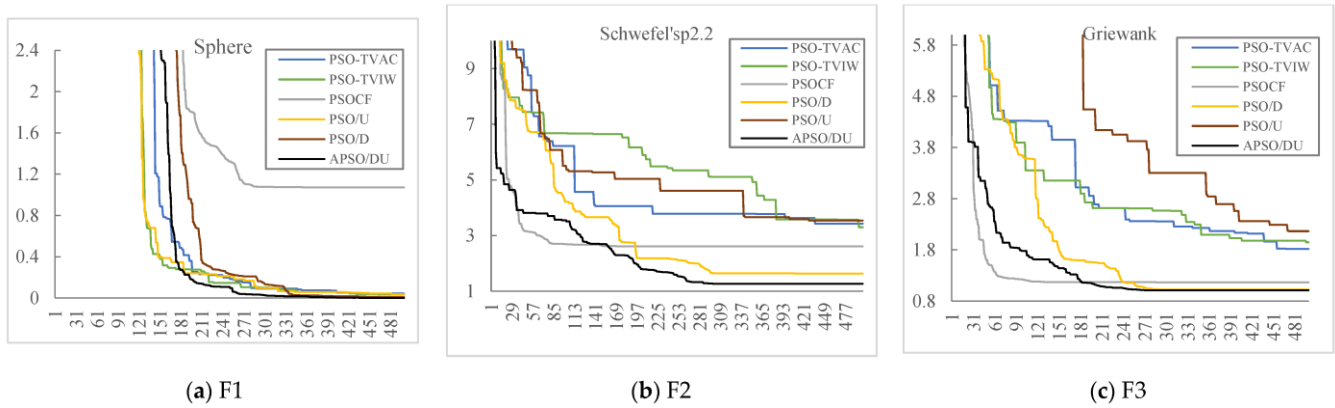


Figure 3. Curves of the convergence process of the benchmark test functions F1–F3.

The average convergence curves of each algorithm for the three tested functions are given in Figure 3. The single-peak test function shows whether the algorithm achieves the target value of the search accuracy. On single-peak functions F1 (sphere) and F2 (Schwefel’sp2.22), the relatively high convergence accuracy is achieved by the APSO/DU algorithm and the PSO/D algorithm, with PSO/CF easily falling into local optimality.

A multi-peaked test function can test the global searchability of an algorithm. In multi-peak function F3 (Griewank) optimization, the APSO/DU algorithm performs best, followed by the PSO/D algorithm and the PSO/CF algorithm, in that order. Among the different functions, APSO/DU has the fastest convergence speed and the highest convergence accuracy and, collectively, the APSO/DU algorithm is the best in terms of finding the best results and showing better stability.

3. Portfolio Optimization Problem

3.1. Related Definitions

The essential parameters in the POP are expected return and risk, and investors usually prefer to maximize return and minimize risk. Assuming a fixed amount of money to buy n stocks, the POP can be described as how to choose the proportion of investments that minimizes ρ the investor’s risk (variance or standard deviation) given a minimum rate of return, or how to choose the proportion of investments that maximizes the investor’s return given a level of risk.

The investor holds fixed assets invested in n stocks $A_i(i = 1, 2, \dots, m)$, let R_i be the return rate of A_i , which is a random variable. μ_i is the expected return on stock A_i . Let $E(R_i)$ denote the mathematical expectation of a random variable R . Define

$$\mu_i = E(R_i) \tag{6}$$

In a certain period, the stock return is the relative number of the difference between the opening and closing prices of that stock, where V_{ij} is the return of stock i in period t , as in Equation (7).

$$V_{ij} = \frac{p_{i,t} - p_{i,t-1}}{p_{i,t-1}}, \quad i = 1, 2, \dots, T \tag{7}$$

where $p_{i,t}$ and $p_{i,t-1}$ are the closing prices of stock i in periods t and $t - 1$, respectively. The expected return on the i th stock is given by Equation (8)

$$\mu_i = \frac{1}{T} \sum_{j=1}^T V_{ij} \tag{8}$$

3.2. Mean-Semivariance Model

A large number of empirical analysis results show that asset returns are characterized by spikes and thick tails, which contradicts the assumption that asset returns are normally distributed in the standard mean-variance model. Additionally, the variance reflects the degree of deviation between actual returns and expected returns, while actual losses (loss risk) are fluctuations below the mean of returns. Thus, the portfolio optimization model based on the lower half-variance risk function is more realistic. Equations (9)–(12) present the mean-semivariance model. Assume that the short selling of assets is not allowed.

$$\min f = \frac{1}{T} \sum_{t=1}^T [(\sum_{i=1}^m x_i r_{it} - \rho)^-]^2 \quad (9)$$

Subject to

$$E(\mu_p) = \sum_{i=1}^m \mu_i x_i \geq \rho \quad (10)$$

$$\sum_{i=1}^m 0 \leq x_i \leq 1, i = 1, 2, \dots, m \quad (11)$$

$$\sum_{i=1}^m x_i = 1 \quad (12)$$

where:

m is the number of stocks in the portfolio;

ρ is the rate of return required by the investor;

x_i is the proportion ($0 \leq x_i \leq 1$) of the portfolio held in assets i ($i = 1, 2, \dots, m$);

μ_i is the mean return of asset i in the targeted period;

μ_p is the mean return of the portfolio in the targeted period.

Equation (9) is the objective function of the model and represents minimizing the risk of the portfolio (the lower half of the variance); Equation (10) ensures that the return of the portfolio is greater than the investor's expected return ρ ; and Equations (11) and (12) indicate that the variables take values in the range $[0, 1]$, and the total investment ratio is 1.

4. Case Analysis

4.1. Experiment Settings

(1) Individual composition

The vector $X = (X_1, X_2, \dots, X_n)$ represents a portfolio strategy whose i th dimensional component x_i represents the allocation of funds to hold the i th stock in that portfolio, namely the weight of that asset in the portfolio.

(2) Variable constraint processing

Equation (10): the feasibility of the particle is checked after the initial assignment of the algorithm and the update of the position vector and if it does not work, the position vector of the particle is recalculated until it is satisfied before the calculation of the objective function is carried out.

Equation (11): the variables take values in the interval $[0, 1]$ and the iterative process uses the boundary to restrict within the interval.

Equation (12): variables on a non-negative constraint basis, sets $= x_1 + x_2 + \dots + x_n$ when $s = 0$, so that all variables in the portfolio are $\frac{1}{n}$; when $s \neq 0$, let $x_i = \frac{x_i}{n}$, $i = 1, 2, \dots, n$.

(3) Parameter values

The particle dimension D is the number of stocks included in the portfolio, and the number of stocks selected in this paper is 15, hence $D = 15$. The parameters of this experimental algorithm are set as described in Section 2.3. of this paper, and the results show the average of 30 independent runs of each algorithm. All PSO algorithms in this paper were written in Python and run on a Windows system for testing.

4.2. Sample Selection

Regarding the selection of stock data, firstly, recent stock data should be selected for analysis to have a certain practical reference value. Secondly, the number of shares is too small to be credible, and the number of shares is too large for the average investor to be distracted with at the same time. Finally, Markowitz's investment theory states that the risk of a single asset is fixed and cannot be reduced on its own, whereas investing in portfolio form diversifies risk without reducing returns. The lower the correlation between any two assets in a portfolio (preferably negative), the more significant the reduction in overall portfolio unsystematic risk [47]. Some methods can be used to solve this problem [48–52].

Based on the above considerations, 30 stocks from different sectors were selected from Choice Financial Terminal, with a time range of 1 January 2019 to 31 December 2021, for a total of 155 weeks of closing price data. Correlation analysis was conducted on the stock data, and 15 stocks with relatively low correlation coefficients were selected for empirical analysis. The price trend charts and correlation coefficients for the 15 stocks are given in Figures 4 and 5.

Figure 4 shows the weekly closing price trend for the 15 stocks data, which provides a visual indication of the trend in stock data. Stocks vary widely in price from one another, with 600612 being the most expensive. As shown in Figure 5, the fifteen stocks have low correlations, with only two portfolios having correlation coefficients greater than 0.5 for any two stocks. Stock 6 and Stock 8 have strong correlations with Stock 12, with correlation coefficients of 0.6 and 0.5, respectively, while all other correlations are below 0.5. Stock 4 and Stock 14 have the lowest correlation, with a correlation coefficient of -0.045 . After calculation, the correlation of the stock data in this paper is low, and the mean correlation coefficient is only 0.198. The lower the correlation between stocks is, the more effective the portfolio choice is in reducing unsystematic risk, thus indicating that investing with a portfolio strategy is effective in reducing risk.

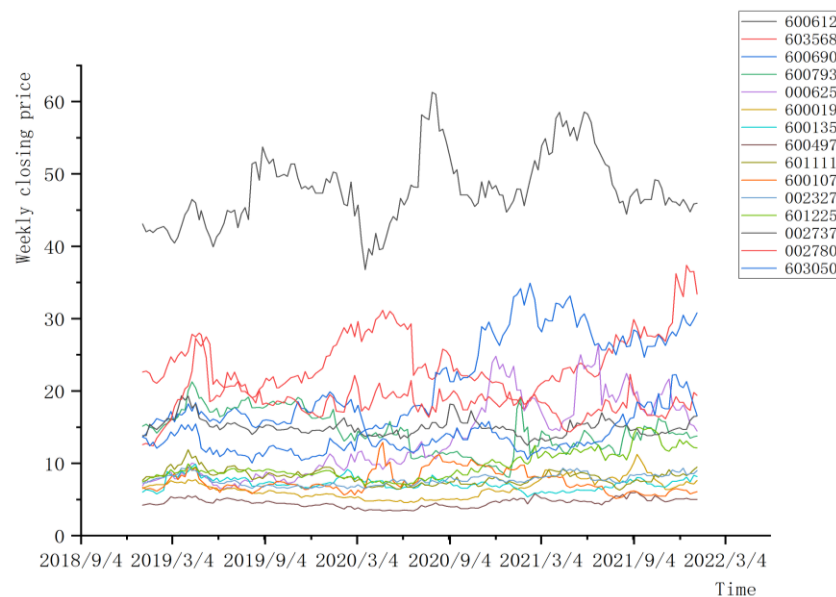


Figure 4. The weekly closing price trend for the 15 stocks data.

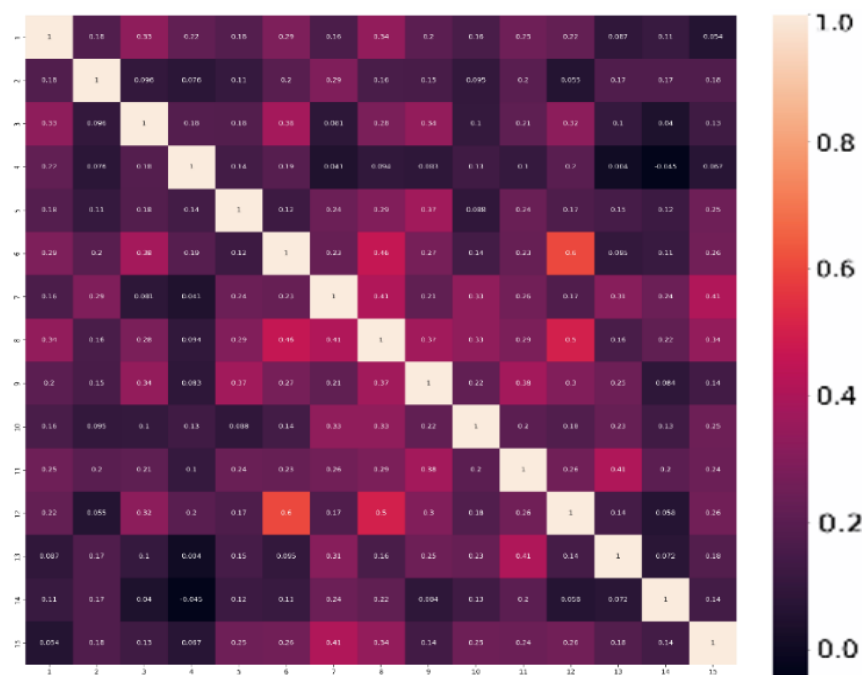


Figure 5. Correlation matrix of 15 stocks.

Table 4 gives the basic statistical characteristics of 15 stocks for 2019–2021, and the returns are the weekly averages of the relative number of closing prices of the stock data. The *p*-values for most of the stock returns in Table 4 are less than 0.05, which should reject the original hypothesis and indicates that the stock returns do not conform to a normal distribution at the 5% significance level. The *p*-values for 600793 and 600135 are greater than 0.05 at a level that does not present significance and cannot reject the original hypothesis, so the data satisfies a normal distribution.

Table 4. Basic characteristics and normality test of 15 stocks from 2019 to 2021.

NO.	Code	Price(yuan)	Return (%)	Std	Prob	Conclusion at the (5%) Level
1	600612	47.930	0.131	0.043	0.003 ***	Distribution not normally distributed
2	603568	24.329	0.408	0.055	0.000 ***	Distribution not normally distributed
3	600690	21.992	0.662	0.052	0.000 ***	Distribution not normally distributed
4	600793	14.396	0.288	0.087	0.060 *	Normality cannot be ruled out
5	000625	13.432	0.810	0.082	0.000 ***	Distribution not normally distributed
6	600019	6.526	0.207	0.046	0.000 ***	Distribution not normally distributed
7	600135	7.158	0.368	0.060	0.069 *	Normality cannot be ruled out
8	600497	4.558	0.253	0.053	0.030 **	Distribution not normally distributed
9	601111	8.095	0.259	0.049	0.000 ***	Distribution not normally distributed
10	600107	7.522	0.221	0.075	0.000 ***	Distribution not normally distributed
11	002327	7.704	0.208	0.038	0.000 ***	Distribution not normally distributed
12	601225	9.689	0.432	0.049	0.000 ***	Distribution not normally distributed
13	002737	14.959	0.204	0.042	0.000 ***	Distribution not normally distributed
14	002780	18.442	0.474	0.063	0.000 ***	Distribution not normally distributed
15	603050	13.506	0.304	0.060	0.000 ***	Distribution not normally distributed

Note: ***, **, and * represent the significance level of 1%, 5%, and 10%, respectively.

Figure 6 shows the histogram of the normality test for 15 stocks. If the normality plot is roughly bell-shaped (high in the middle and low at the ends), the data are largely accepted as normally distributed. It can be seen from the figure that the normal distribution plots of the 600793 and 600135 stock data roughly show a bell shape, which is consistent with normal distribution. However, the normal distribution of most stocks does not show a bell shape and does not conform to normal distribution.

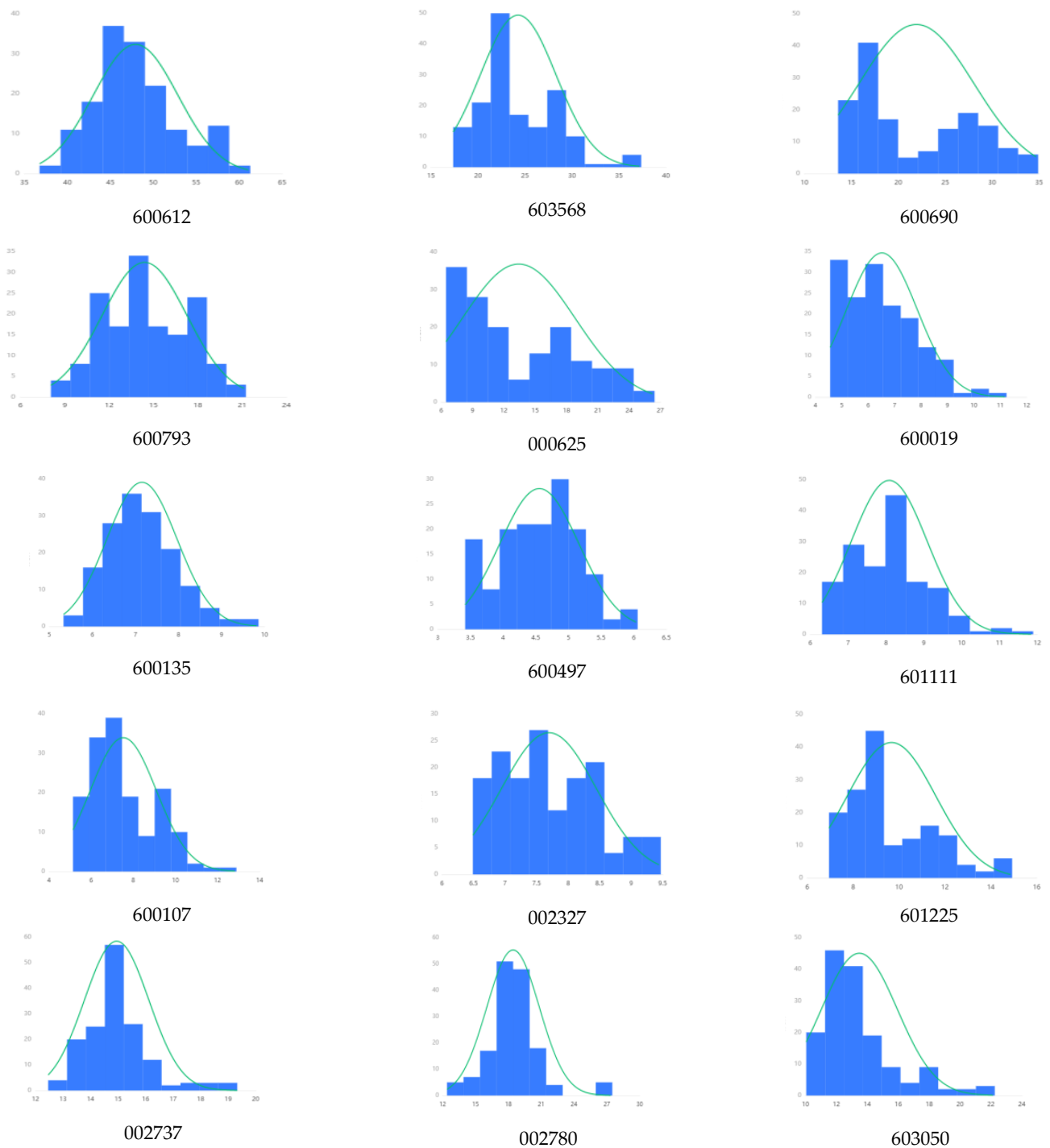


Figure 6. Histogram of normality test.

It is difficult for all the stock data to conform to the assumption that asset returns are normally distributed in MV. Secondly, the real loss refers to the fluctuation below the mean of returns; thus, the portfolio model based on the lower half-variance risk function is more realistic, so the MSV model is used for empirical analysis later in the paper.

4.3. Interpretation of Result

In order to verify the effectiveness of the semi-variance risk measure in practice, six different levels of return (0.005 to 0.0030) are set in this paper. Table 5 gives the risk values obtained by different algorithms at the same return level, and the best results are identified in bold font. A visualization of the Pareto frontier (PF) obtained by solving the

four algorithms is given in Figure 7. The optimal investment ratios derived from each algorithm solved at the expected return level of 0.03 are given in Table 6 to visually compare the effectiveness of the APSO/DU algorithm in solving the MSVPOP.

Table 5. Experimental results of five algorithms.

NO.	μ	MSV			
		PSO-TVIW	PSO-TVAC	PSOCF	APSO/DU
1	0.0030	3.70×10^{-4}	3.82×10^{-4}	3.63×10^{-4}	3.33×10^{-4}
2	0.0025	3.52×10^{-4}	3.69×10^{-4}	3.57×10^{-4}	3.16×10^{-4}
3	0.0020	3.34×10^{-4}	3.48×10^{-4}	3.37×10^{-4}	3.08×10^{-4}
4	0.0015	3.20×10^{-4}	3.37×10^{-4}	3.27×10^{-4}	3.02×10^{-4}
5	0.0010	3.16×10^{-4}	3.26×10^{-4}	3.10×10^{-4}	2.84×10^{-4}
6	0.0005	2.99×10^{-4}	3.04×10^{-4}	2.95×10^{-4}	2.78×10^{-4}

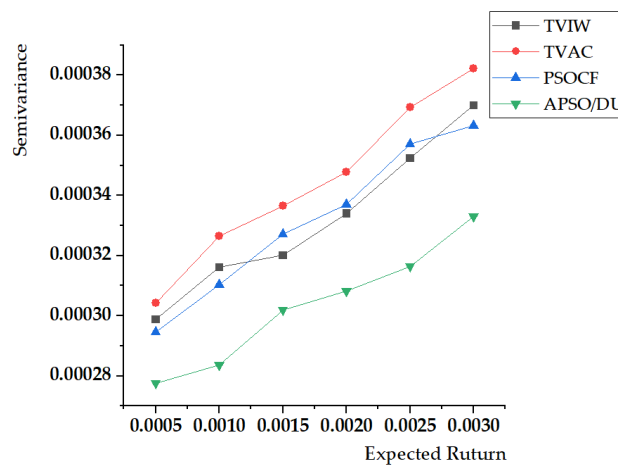


Figure 7. The obtained PF by five algorithms.

Table 6. The optimal investment ratio solved by each algorithm at $\mu = 0.03$.

	Code	PSO-TVIW	PSO-TVAC	PSOCF	APSO/DU
1	600612	0.1166	0.0782	0.0932	0.0729
2	603568	0.0000	0.0655	0.0817	0.1129
3	600690	0.1015	0.0861	0.0055	0.0277
4	600793	0.0000	0.0832	0.0062	0.0168
5	000625	0.0000	0.0186	0.0360	0.0078
6	600019	0.0767	0.0504	0.0091	0.0692
7	600135	0.0692	0.0317	0.1147	0.0038
8	600497	0.0000	0.0812	0.0000	0.0031
9	601111	0.0563	0.0802	0.0573	0.0611
10	600107	0.1311	0.0589	0.0518	0.0081
11	002327	0.1057	0.0760	0.1935	0.1773
12	601225	0.0276	0.0839	0.1324	0.0992
13	002737	0.1338	0.0718	0.0019	0.1273
14	002780	0.1322	0.0746	0.1378	0.0725
15	603050	0.0493	0.0596	0.0790	0.1402
	MSV	3.70×10^{-4}	3.82×10^{-4}	3.63×10^{-4}	3.33×10^{-4}

Table 5 and Figure 7 show that as returns increase, the portfolio’s risk also increases, in line with the law of high returns accompanied by high risk in the equity market. Taking the expected return $u = 0.003$ as an example, APSO/DU has the smallest value of risk (2.78×10^{-4}) and the PSO-TVAC algorithm has the largest value of risk (3.82×10^{-4}), so

the portfolio solved by the APSO/DU algorithm is chosen at the expected return level of 0.03, corresponding to the smallest value of risk. A sensible person should choose this portfolio. Similar to the other return levels analyzed, the APSO/DU algorithm proposed in this paper is always lower than the results calculated by the other algorithms. The APSO/DU algorithm calculates a lower value of risk than the three classical adaptive improved particle swarm algorithms when the expected returns are the same, indicating that the combination of improved particle swarm solutions obtains relatively better results at the same expected return, and APSO/DU has stronger global search capability and more easily finds the optimal global solution.

The optimal investment ratios derived from each algorithm solved at the expected return level of 0.03 are given in Table 6 to visually compare the effectiveness of the APSO/DU algorithm in solving the MSVPOP.

5. Conclusions

In order to cope with the POPMSV challenge well, a multi-strategy adaptive particle swarm optimization, namely APSO/DU, was developed, which has the following two advantages. Firstly, the variable constraint (1) is set to better represent the stock selection, and asset weights of the solution in the POP help to cope with the MSVPOP challenge efficiently. Secondly, an improved particle swarm optimization algorithm (APSO/DU) with adaptive parameters was proposed by adopting a dual-update strategy. It can adaptively adjust the relevant parameters so that the search behavior of the algorithm can match the current search environment to avoid falling into local optimality and effectively balance global and local search. The sole adjustment of w and c_1 and c_2 would weaken the uniformity of the algorithm's evolutionary process and make it difficult to adapt to complex nonlinear optimization, so a dual dynamic adaptation mechanism is chosen to adjust the core parameters. The APSO/DU algorithm is more adaptable to nonlinear complex optimization problems, improving solution accuracy and approximating the global PF. The results show that APSO/DU exhibits stronger solution accuracy than the comparison algorithm, i.e., the improved algorithm finds the portfolio with the least risk at the same level of return, more closely approximating PF. The above research results can be used for investors to invest in low-risk portfolios with valuable suggestions with good practical applications.

Author Contributions: Conceptualization, Y.S. and Y.L.; methodology, Y.S. and W.D.; software, Y.L.; validation, H.C. and Y.L.; resources, Y.S.; data curation, Y.S.; writing—original draft preparation, Y.S. and Y.L.; writing—review and editing, H.C.; visualization, Y.S.; supervision, H.C.; project administration, H.C.; funding acquisition, H.C. and W.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61976124, 61976125, U2133205), the Yantai Key Research and Development Program (2020YT06000970), Wealth management characteristic construction project of Shandong Technology and Business University (2022YB10), the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0536; and the Open Project Program of the Traction Power State Key Laboratory of Southwest Jiaotong University (TPL2203).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Markowitz, H. Portfolio selection. *J. Financ.* **1952**, *7*, 77–79.
2. Markowitz, H. *Portfolio Selection: Efficient Diversification of Investments*; Wiley: New York, NY, USA, 1959.
3. Xu, G.; Bai, H.; Xing, J.; Luo, T.; Xiong, N.N.; Cheng, X.; Liu, S.; Zheng, X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **2022**, *164*, 1–11. [[CrossRef](#)]

4. Yu, C.; Liu, C.; Yu, H.; Song, M.; Chang, C.-I. Unsupervised Domain Adaptation with Dense-Based Compaction for Hyperspectral Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12287–12299. [[CrossRef](#)]
5. Jin, T.; Yang, X. Monotonicity theorem for the uncertain fractional differential equation and application to uncertain financial market. *Math. Comput. Simul.* **2021**, *190*, 203–221. [[CrossRef](#)]
6. Li, N.; Huang, W.; Guo, W.; Gao, G.; Zhu, Z. Multiple Enhanced Sparse Decomposition for Gearbox Compound Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 770–781. [[CrossRef](#)]
7. Bi, J.; Zhou, G.; Zhou, Y.; Luo, Q.; Deng, W. Artificial Electric Field Algorithm with Greedy State Transition Strategy for Spherical Multiple Traveling Salesmen Problem. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 5. [[CrossRef](#)]
8. Zhong, K.; Zhou, G.; Deng, W.; Zhou, Y.; Luo, Q. MOMPA: Multi-objective marine predator algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *385*, 114029.
9. Venkataraman, S.V. A remark on mean: Emivariance behaviour: Downside risk and capital asset pricing. *Int. J. Financ. Econ.* **2021**. [[CrossRef](#)]
10. Kumar, R.R.; Stauvermann, P.J.; Samitas, A. An Application of Portfolio Mean-Variance and Semi-Variance Optimization Techniques: A Case of Fiji. *J. Risk Financial Manag.* **2022**, *15*, 190. [[CrossRef](#)]
11. Wu, Q.; Gao, Y.; Sun, Y. Research on Probability Mean-Lower Semivariance-Entropy Portfolio Model with Background Risk. *Math. Probl. Eng.* **2020**, *2020*, 2769617. [[CrossRef](#)]
12. Wu, X.; Gao, A.; Huang, X. Modified Bacterial Foraging Optimization for Fuzzy Mean-Semivariance-Skewness Portfolio Selection. In Proceedings of the International Conference on Swarm Intelligence, Cham, Switzerland, 13 July 2020; pp. 335–346. [[CrossRef](#)]
13. Ivanova, M.; Dospatliev, L. Constructing of an Optimal Portfolio on the Bulgarian Stock Market Using Hybrid Genetic Algorithm for Pre and Post COVID-19 Periods. *Asian-Eur. J. Math.* **2022**, *15*, 2250246. [[CrossRef](#)]
14. Sun, Y.; Ren, H. A GD-PSO Algorithm for Smart Transportation Supply Chain ABS Portfolio Optimization. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 6653051. [[CrossRef](#)]
15. Zhao, H.; Chen, Z.G.; Zhan, Z.H.; Kwong, S.; Zhang, J. Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. *Neurocomputing* **2021**, *430*, 58–70.
16. Deng, X.; He, X.; Huang, C. A new fuzzy random multi-objective portfolio model with different entropy measures using fuzzy programming based on artificial bee colony algorithm. *Eng. Comput.* **2021**, *39*, 627–649. [[CrossRef](#)]
17. Dhaini, M.; Mansour, N. Squirrel search algorithm for portfolio optimization. *Expert Syst. Appl.* **2021**, *178*, 114968. [[CrossRef](#)]
18. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1945–1950. [[CrossRef](#)]
19. Shi, Y.H.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
20. Xiao, Y.; Shao, H.; Han, S.; Huo, Z.; Wan, J. Novel Joint Transfer Network for Unsupervised Bearing Fault Diagnosis from Simulation Domain to Experimental Domain. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5254–5263. [[CrossRef](#)]
21. Yan, S.; Shao, H.; Xiao, Y.; Liu, B.; Wan, J. Hybrid robust convolutional autoencoder for unsupervised anomaly detection of machine tools under noises. *Robot. Comput. Manuf.* **2023**, *79*, 102441. [[CrossRef](#)]
22. Deng, W.; Zhang, L.; Zhou, X.; Zhou, Y.; Sun, Y.; Zhu, W.; Chen, H.; Deng, W.; Chen, H.; Zhao, H. Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem. *Inf. Sci.* **2022**, *612*, 576–593. [[CrossRef](#)]
23. Wei, Y.; Zhou, Y.; Luo, Q.; Deng, W. Optimal reactive power dispatch using an improved slime mould algorithm. *Energy Rep.* **2021**, *7*, 8742–8759. [[CrossRef](#)]
24. Song, Y.; Cai, X.; Zhou, X.; Zhang, B.; Chen, H.; Li, Y.; Deng, W.; Deng, W. Dynamic hybrid mechanism-based differential evolution algorithm and its application. *Expert Syst. Appl.* **2023**, *213*, 118834. [[CrossRef](#)]
25. Jin, T.; Zhu, Y.; Shu, Y.; Cao, J.; Yan, H.; Jiang, D. Uncertain optimal control problem with the first hitting time objective and application to a portfolio selection model. *J. Intell. Fuzzy Syst.* **2022**. [[CrossRef](#)]
26. Zhang, X.; Wang, H.; Du, C.; Fan, X.; Cui, L.; Chen, H.; Deng, F.; Tong, Q.; He, M.; Yang, M.; et al. Custom-Molded Offloading Footwear Effectively Prevents Recurrence and Amputation, and Lowers Mortality Rates in High-Risk Diabetic Foot Patients: A Multicenter, Prospective Observational Study. *Diabetes Metab. Syndr. Obesity Targets Ther.* **2022**, *15*, 103–109. [[CrossRef](#)]
27. Zhao, H.; Zhang, P.; Zhang, R.; Yao, R.; Deng, W. A novel performance trend prediction approach using ENBLS with GWO. *Meas. Sci. Technol.* **2023**, *34*, 025018. [[CrossRef](#)]
28. Ren, Z.; Han, X.; Yu, X.; Skjetne, R.; Leira, B.J.; Sævik, S.; Zhu, M. Data-driven simultaneous identification of the 6DOF dynamic model and wave load for a ship in waves. *Mech. Syst. Signal Process.* **2023**, *184*, 109422. [[CrossRef](#)]
29. Zhang, Z.; Huang, W.; Liao, Y.; Song, Z.; Shi, J.; Jiang, X.; Shen, C.; Zhu, Z. Bearing fault diagnosis via generalized logarithm sparse regularization. *Mech. Syst. Signal Process.* **2021**, *167*, 108576. [[CrossRef](#)]
30. Yu, Y.; Hao, Z.; Li, G.; Liu, Y.; Yang, R.; Liu, H. Optimal search mapping among sensors in heterogeneous smart homes. *Math. Biosci. Eng.* **2022**, *20*, 1960–1980. [[CrossRef](#)]
31. Chen, H.Y.; Fang, M.; Xu, S. Hyperspectral remote sensing image classification with CNN based on quantum genetic-optimized sparse representation. *IEEE Access* **2020**, *8*, 99900–99909. [[CrossRef](#)]
32. Zhao, H.; Yang, X.; Chen, B.; Chen, H.; Deng, W. Bearing fault diagnosis using transfer learning and optimized deep belief network. *Meas. Sci. Technol.* **2022**, *33*, 065009. [[CrossRef](#)]

33. Xu, J.; Zhao, Y.; Chen, H.; Deng, W. ABC-GSPBFT: PBFT with grouping score mechanism and optimized consensus process for flight operation data-sharing. *Inf. Sci.* **2023**, *624*, 110–127. [[CrossRef](#)]
34. Duan, Z.; Song, P.; Yang, C.; Deng, L.; Jiang, Y.; Deng, F.; Jiang, X.; Chen, Y.; Yang, G.; Ma, Y.; et al. The impact of hyperglycaemic crisis episodes on long-term outcomes for inpatients presenting with acute organ injury: A prospective, multicentre follow-up study. *Front. Endocrinol.* **2022**, *13*, 1057089. [[CrossRef](#)]
35. Chen, H.; Li, C.; Mafarja, M.; Heidari, A.A.; Chen, Y.; Cai, Z. Slime mould algorithm: A comprehensive review of recent variants and applications. *Int. J. Syst. Sci.* **2022**, *54*, 204–235. [[CrossRef](#)]
36. Liu, Y.; Heidari, A.A.; Cai, Z.; Liang, G.; Chen, H.; Pan, Z.; Alsufyani, A.; Bourouis, S. Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. *Neurocomputing* **2022**, *503*, 325–362. [[CrossRef](#)]
37. Dong, R.; Chen, H.; Heidari, A.A.; Turabieh, H.; Mafarja, M.; Wang, S. Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowl. Based Syst.* **2021**, *233*, 107529. [[CrossRef](#)]
38. Chen, M.; Shao, H.; Dou, H.; Li, W.; Liu, B. Data Augmentation and Intelligent Fault Diagnosis of Planetary Gearbox Using ILoFGAN Under Extremely Limited Samples. *IEEE Trans. Reliab.* **2022**. [[CrossRef](#)]
39. Tian, C.; Jin, T.; Yang, X.; Liu, Q. Reliability analysis of the uncertain heat conduction model. *Comput. Math. Appl.* **2022**, *119*, 131–140. [[CrossRef](#)]
40. Thakkar, A.; Chaudhari, K. A Comprehensive Survey on Portfolio Optimization, Stock Price and Trend Prediction Using Particle Swarm Optimization. *Arch. Comput. Methods Eng.* **2020**, *28*, 2133–2164. [[CrossRef](#)]
41. Harrison, K.R.; Engelbrecht, A.P.; Ombuki-Berman, B.M. Self-adaptive particle swarm optimization: A review and analysis of convergence. *Swarm Intell.* **2018**, *12*, 187–226. [[CrossRef](#)]
42. Boudt, K.; Wan, C. The effect of velocity sparsity on the performance of cardinality constrained particle swarm optimization. *Optim. Lett.* **2019**, *14*, 747–758. [[CrossRef](#)]
43. Clerc, M. *Particle Swarm Optimization*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
44. Zhang, W.; Jin, Y.; Li, X.; Zhang, X. A simple way for parameter selection of standard particle swarm optimization. In Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence, Berlin, Germany, 11–13 November 2011; pp. 436–443.
45. Huang, C.; Zhou, X.B.; Ran, X.J.; Liu, Y.; Deng, W.Q.; Deng, W. Co-evolutionary competitive swarm optimizer with three-phase for large-scale complex optimization problem. *Inf. Sci.* **2023**, *619*, 2–18. [[CrossRef](#)]
46. Liu, H.; Zhang, X.W.; Tu, L.P. A modified particle swarm optimization using adaptive strategy. *Expert Syst. Appl.* **2020**, *152*, 113353. [[CrossRef](#)]
47. Silva, Y.L.T.; Herthel, A.B.; Subramanian, A. A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems. *Expert Syst. Appl.* **2019**, *133*, 225–241. [[CrossRef](#)]
48. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
49. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
50. Yu, C.; Zhou, S.; Song, M.; Chang, C.-I. Semisupervised Hyperspectral Band Selection Based on Dual-Constrained Low-Rank Representation. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 5503005. [[CrossRef](#)]
51. Li, W.; Zhong, X.; Shao, H.; Cai, B.; Yang, X. Multi-mode data augmentation and fault diagnosis of rotating machinery using modified ACGAN designed with new framework. *Adv. Eng. Inform.* **2022**, *52*, 101552. [[CrossRef](#)]
52. He, Z.Y.; Shao, H.D.; Wang, P.; Janet, L.; Cheng, J.S.; Yang, Y. Deep transfer multi-wavelet auto-encoder for intelligent fault diagnosis of gearbox with few target training samples. *Knowl.-Based Syst.* **2019**, *191*, 105313. [[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.