Dynamic Modeling and Risk Constraint Analysis for Multi-Asset Portfolio Optimization

Xinran Li *

School of Economics, Shanghai University, Shanghai, 201899, China

* Corresponding Author Email: lilyxinran@shu.edu.cn

Abstract. With increasing volatility in global financial markets, traditional static portfolio optimization models struggle to adapt to the time-varying characteristics of asset returns and risks. To address the dynamic optimization needs of multi-asset portfolios, this paper proposes a dynamic robust multi-asset portfolio optimization algorithm (DRMPOA). This algorithm first integrates macroeconomic indicators through an improved attention-based LSTM model to predict asset returns and volatility. A robust optimization model with CVaR constraints is then constructed, combining a particle swarm optimization algorithm with dynamic inertia weights to solve for optimal weights. These weights are then adjusted in real time using a 60-trading-day sliding window. Experiments are conducted using data from multiple asset classes, including the CSI 300 Index, the CSI All Bond Index, the Nanhua Commodity Index, and Bitcoin, from 2018 to 2023. The algorithm is compared with MV, Static-CVaR, and Dynamic-MCVaR algorithms. Results show that DRMPOA achieved a cumulative return of 48.2% (8.1% annualized), a 9.5 percentage point increase over Dynamic-MCVaR. Its 95th-percentile daily CVaR was 1.2%, and its annualized volatility was 12.3%, 0.1 and 1.5 percentage points lower than Dynamic-MCVaR, respectively. Its annualized Sharpe ratio was 0.58 (a 28.9% increase), and its single solution took 12.5 seconds (3.3 seconds faster than Dynamic-MCVaR). This algorithm balances profitability, risk tolerance, and efficiency, providing an effective solution for multi-asset allocation.

Keywords: Multi-Asset Portfolio; Risk Constraints; DRMPOA Algorithm; LSTM; Particle Swarm Optimization.

1. Introduction

With the accelerated integration of global financial markets, asset prices are increasingly affected by macroeconomic fluctuations, policy adjustments, and cross-market linkages. Single-asset investments face significant uncertainty due to risk concentration. Multi-asset portfolios, with their core advantages of diversifying risk and balancing returns, have become a mainstream choice for institutional and individual investors. However, traditional portfolio optimization is often based on a static mean-variance framework, assuming constant market parameters [1]. This makes it difficult to capture the time-varying characteristics of asset returns and risks in dynamic markets, resulting in significant deviations between optimization results and actual market performance. For example, during periods of intense market volatility, static models tend to overestimate the allocation ratio of low-risk assets, missing out on profit opportunities. Conversely, during periods of stability, inefficient capital utilization may occur due to rigid risk constraints [2]. This limitation highlights the urgent need for dynamic modeling in multi-asset optimization. From the current research perspective, scholars both domestically and internationally have explored dynamic portfolio optimization. Some studies have introduced time series models to predict asset returns or employed metrics such as CVaR and VaR to strengthen risk constraints. However, existing results still have shortcomings: most dynamic models focus solely on a single asset class or short-term market fluctuations, lacking consideration of cross-asset linkages [3]. Furthermore, their algorithms lack robustness, resulting in poor stability in optimization results in the face of sudden market shocks, making it difficult to balance profitability and risk resilience [4]. These research gaps provide opportunities for further improving the practicality of multi-asset portfolio optimization.

This paper first constructs a dynamic model for multi-asset portfolios by combining the time-varying nature of asset returns with cross-market linkages. Second, it designs an original dynamic robust multi-asset portfolio optimization algorithm (DRMPOA), enhancing its adaptability and robustness by improving its LSTM prediction module and particle swarm optimization module [5]. Third, it conducts experimental simulations using multi-asset data from 2018 to 2023, comparing the model and algorithm with traditional algorithms to verify their effectiveness.

2. Foundations of Multi-Asset Portfolio Dynamic Modeling

2.1. Concepts Related to Multi-Asset Portfolios

A multi-asset portfolio is a collection of two or more asset classes, such as stocks, bonds, commodities, and digital currencies, allocated at specific weights. Its core components include asset class, allocation weight, and rebalancing period [6]. Returns are calculated using the weighted average method: portfolio return $R_p = \sum_{i=1}^{n} w_i R_i$, where w_i is the weight of the i asset class ($\sum_{i=1}^{n} w_i = 1$) and R_i is the return on the i asset class. In dynamic modeling, the time dimension divides the decision cycle into daily, weekly, or monthly units, requiring real-time updates of asset prices, trading volume, and other data. Asset types must cover low-risk (such as government bonds), medium-risk (such as blue-chip stocks), and high-risk (such as digital currencies) categories to ensure effective risk diversification.

2.2. Core Elements of Dynamic Modeling

In dynamic markets, three factors directly influence portfolio performance: First, asset price fluctuations, measured by volatility $\sigma_i = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T} \left(R_{i,t} - \bar{R}_i\right)^2}$. Sudden changes in volatility can alter the risk-return characteristics of assets. Second, macroeconomic indicators, such as GDP growth, inflation, and interest rates, indirectly influence asset prices by influencing corporate earnings and market liquidity. For example, rising interest rates typically lead to falling bond prices. Third, policy adjustments, such as monetary policy and industry regulatory policies, can trigger capital transfers between asset classes [7]. For example, policies supporting the new energy industry may inflate the valuation of related stocks. Modeling requires converting these factors into quantitative variables, such as incorporating interest rate changes and policy dummies, to ensure dynamic adaptability.

2.3. Risk Constraint Types and Quantification Methods

Common risk constraints include three types: Variance constraints measure overall risk using the portfolio variance $\sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \mathrm{Cov}(R_i, R_j)$($\mathrm{Cov}(R_i, R_j)$ is the covariance between assets i and j), requiring $\sigma_p^2 \le \sigma_{\max}^2$; VaR constraints (95% confidence level) represent the maximum possible loss over a given period, calculated using historical simulation, i.e., $\mathrm{VaR}_{0.95} = -R_{p,(5\%)}$($R_{p,(5\%)}$ quantile of the portfolio return), requiring $VaR_{0.95} \le VaR_{\max}$; and CVaR constraints measure the mean loss exceeding VaR, using the formula: $CVaR_{0.95} = -\frac{1}{5\%T}\sum_{t:R_{p,t} \le -VaR_{0.95}} R_{p,t}$, which more accurately characterizes tail risk, is the core constraint metric in this study.

3. Design of the Dynamic Robust Multi-Asset Portfolio Optimization Algorithm (DRMPOA)

Existing dynamic portfolio algorithms often rely on fixed market assumptions, making them prone to weight allocation bias when faced with uncertainties such as interest rate fluctuations and policy changes. The DRMPOA algorithm, driven by a dual "dynamic prediction - robust optimization" approach, captures the time-varying characteristics of asset returns and risks in real time and constructs an adaptive adjustment mechanism [8]. It first uses an improved attention mechanism LSTM model to predict market dynamics, then embeds the prediction results into a robust optimization framework. Under the objective of maximizing returns, a risk buffer is introduced to mitigate prediction errors and market shocks, achieving a closed-loop decision-making process of "prediction - optimization - adjustment."

3.1. Data Preprocessing

We collected daily closing prices of multiple assets $P_{i,t}(i = 1,2, \dots, n$ represents the asset category and $t = 1,2, \dots, T$ represents the time period) and macroeconomic indicators $M_{k,t}(k = 1,2, \dots, m$ represents the indicator type). We used the 3D criterion to remove outliers and normalized the asset return $r_{i,t} = \ln\left(P_{i,t}/P_{i,t-1}\right)$ with the macroeconomic indicators to eliminate dimensional differences:

$$x_{i,t} = \frac{r_{i,t} - \min(r_{i,1:T})}{\max(r_{i,1:T}) - \min(r_{i,1:T})} \tag{1}$$

$$y_{k,t} = \frac{M_{k,t} - \min(M_{k,1:T})}{\max(M_{k,1:T}) - \min(M_{k,1:T})} \tag{2}$$

Construct an input dataset $D = \left\{\left(x_{1,t}, \dots, x_{n,t}, y_{1,t}, \dots, y_{m,t}\right), t = 1,2, \dots, T\right\}$ to provide standardized data for subsequent modules.

3.2. Dynamic Market Forecast Module

This module uses an improved attention mechanism LSTM model, incorporating macroeconomic indicators as attention weight adjustment factors to improve the accuracy of return and volatility forecasts [9]. First, the LSTM hidden layer state update formula is:

$$h_t = \tanh\left(W_h[x_t; y_t] + U_h h_{t-1} + b_h\right) \tag{3}$$

Where $[x_t; y_t]$ is the concatenation vector of asset data and macro indicators, W_h, U_h are weight matrices, and b_h is the bias term. The macro attention weight $\alpha_{k,t}$ is introduced to modify the output layer prediction result:

$$\hat{r}_{i,t+1} = W_o\left[\alpha_{1,t} y_{1,t}; \dots; \alpha_{m,t} y_{m,t}; h_t\right] + b_o \tag{4}$$

$$\hat{\sigma}_{i,t+1} = \exp\left(W_\sigma\left[\alpha_{1,t} y_{1,t}; \dots; \alpha_{m,t} y_{m,t}; h_t\right] + b_\sigma\right) \tag{5}$$

Equations (4) and (5) are the predicted return and volatility of the i asset class for period $t+1$, respectively. $\alpha_{k,t} = \exp\left(y_{k,t}\right)/\sum_{k=1}^{m} \exp\left(y_{k,t}\right)$ is the macro-indicator attention weight. This mechanism strengthens the influence of key macro factors on the forecast results.

3.3. Robust Optimization Solution Module

With the goal of maximizing the expected return of the portfolio, CVaR risk constraints and weight boundary constraints are introduced to construct a robust optimization model. The objective function is:

$$\max J = E\left[\sum_{i=1}^{n} w_i \hat{r}_{i,t+1}\right] - \lambda \cdot C\mathrm{VaR}(w, \hat{\sigma}) \tag{6}$$

Where $w = (w_1, \dots, w_n)$ is the asset weight vector, λ is the risk aversion coefficient, $CVaR(w, \hat{\sigma})$ is the robust CVaR of the portfolio, and the calculation formula is:

$$C\mathrm{VaR}(w, \hat{\sigma}) = \inf_{\beta \in \mathbb{R}} \left\{\beta + \frac{1}{1-\alpha} E\left[\max\left(0, \beta - \sum_{i=1}^{n} w_i \hat{r}_{i,t+1}\right)\right]\right\} \tag{7}$$

α is the confidence level (taken as 0.95 in this study). The constraints include: $\sum_{i=1}^{n} w_i = 1, w_i \in \left[w_{i,\min}, w_{i,\max}\right](w_{i,\min}, w_{i,\max}$ are the upper and lower limits of the asset i weight). An improved particle swarm optimization algorithm is used to solve the problem. A dynamic inertia weight adjustment factor $\omega_t = \omega_{\max} - t(\omega_{\max} - \omega_{\min})/T_{\max}$ is introduced to accelerate convergence, ultimately yielding the optimal weight w^*.

3.4. Real-time Update and Adjustment Module

A sliding time window T_w is set (60 trading days in this study). After each T_w interval, the latest market data is re-input, and the data preprocessing, dynamic prediction, and optimization steps are repeated to update the optimal weight:

$$w_{t+T_w}^{*} = \arg\ \max J\left(w \mid \hat{r}_{t+1:t+T_w}, \hat{\sigma}_{t+1:t+T_w}\right) \tag{8}$$

Equation (8) is used to implement periodic dynamic adjustment of the portfolio weights, ensuring the algorithm's real-time adaptation to market changes.

4. Experimental Simulation and Result Analysis

4.1. Experimental Data Preparation

The experiment selected multiple asset classes covering low, medium, and high risk to ensure the portfolio's risk diversification characteristics and market representativeness [10]. Among them, the CSI 300 Index (representing large-cap blue-chip stocks) and the ChiNext Index (representing growth stocks) were selected as stock assets; the CSI All Bond Index (covering government bonds, financial bonds, etc.) was selected as bond assets; the Nanhua Commodity Index (including energy, agricultural products, etc.) was selected as commodity assets; and Bitcoin (BTC) and Ethereum (ETH) were selected as digital currencies (the data source should be specified as CoinMarketCap). The data span is set from January 1, 2018, to December 31, 2023, totaling six years of daily data. This period encompasses several typical market phases, including the 2018 bear market, the 2020 pandemic, the 2021 structural market, and the 2022 global market adjustment. This fully verifies the algorithm's adaptability in different market environments [11]. The collected data types fall into two categories: First, daily asset trading data, including opening and closing prices, trading volume, highest and lowest prices, is used to calculate indicators such as asset returns and volatility. Second, contemporaneous macroeconomic data includes data such as the quarterly year-on-year GDP growth rate, the monthly Consumer Price Index (CPI), which reflects inflation, the daily 10-year Treasury bond yield, which represents the risk-free rate, and the daily RMB/USD exchange rate, which reflects the impact of exchange rate fluctuations on assets [12]. This data is incorporated into the dynamic forecasting module as auxiliary variables to improve the accuracy of predicting asset return and risk trends. For example, CPI changes can be used to determine the direction of monetary policy adjustments and, in turn, predict the price linkage between bonds and stocks.

4.2. Experimental Environment and Parameter Setup

The experimental hardware environment utilizes an Intel Core i7-12700H processor (14 cores, 20 threads, base frequency 2.3GHz, boost frequency 4.7GHz), paired with 16GB of DDR5 4800MHz memory, and Windows 10 Pro (64-bit). This hardware configuration meets the requirements of large-scale data processing and algorithm iteration, avoiding experimental efficiency bottlenecks caused by insufficient hardware performance.

For software tools, Python 3.9 was chosen as the programming language due to its extensive library support for data analysis and machine learning. The LSTM model in the dynamic prediction module is implemented using the TensorFlow 2.8 framework, which offers efficient neural network training capabilities and a flexible model building interface. Data processing relies on the Pandas and NumPy libraries for data cleaning, transformation, and statistical calculations. Matplotlib 3.5 and Seaborn 0.11 were used for visualization to generate experimental results and charts [13]. The particle swarm optimization algorithm was solved using a custom Python function to ensure consistency between the algorithm logic and the theoretical design. Parameter settings were adjusted multiple times to balance algorithm performance and efficiency: the LSTM model has two hidden layers, with 64 nodes per layer (balancing prediction accuracy and computational cost), a learning rate of 0.001 (using the Adam optimizer to avoid training oscillations caused by excessively high learning rates or slow convergence caused by excessively low learning rates), 100 iterations (observed on the validation set, the model loss function stabilized after 100 iterations, and further iterations did not significantly improve accuracy), an input sequence length of 30 (i.e., predicting the next trading day's indicators based on the previous 30 trading days' data), a particle swarm optimization algorithm population size of 50 (to ensure population diversity and avoid falling into local optima), an inertia weight with an initial value of 0.9 and a reduction to 0.4 at the end of the iteration (a linear decrease strategy, enhancing global search capabilities in the early stages and improving local optimization accuracy in the later stages), learning factors c1 (individual cognitive factor) and c2 (social cognitive factor) were both set to 1.8 (to balance the influence of individual experience and group information), and a maximum number of iterations of 50 (verified by multiple experiments, 50 Iterations allow the objective function value to converge to a stable optimal solution [14]. The dynamic update window is set to 60 trading days (approximately 3 months, which ensures a sufficient data sample size and can capture short-term market changes in a timely manner).

4.3. Comparison Algorithm Selection

Three typical algorithms in the field of multi-asset portfolio optimization were selected for comparison to ensure comprehensiveness and representativeness of the experimental comparison: First, the traditional mean-variance (MV) optimization algorithm, a foundational principle of modern portfolio theory, uses portfolio variance to measure risk and pursues a return-variance balance [15]. However, it does not consider market dynamics and tail risk, and serves as a benchmark for static optimization algorithms. Second, the static CVaR optimization algorithm (Static-CVaR) builds on the MV algorithm by incorporating the CVaR metric (to more accurately characterize tail risk), but similarly employs fixed parameter assumptions and does not adjust weights with market fluctuations. This algorithm is used to validate the effectiveness of dynamic modeling in improving risk control. Third, the dynamic mean-CVaR optimization algorithm (Dynamic-MCVaR) offers dynamic adjustment capabilities, updating model parameters via a rolling time window. However, it lacks a robust optimization mechanism, resulting in weaker resilience to market uncertainty. This algorithm is used to validate the necessity of robust design [16]. The comparison indicators are divided into four categories: return indicators (cumulative rate of return, annualized rate of return), which reflect the profitability of the algorithm; risk indicators (95% confidence level daily CVaR, annualized volatility), which measure the risk level of the portfolio; risk-adjusted return indicators (annualized Sharpe ratio, annualized Sortino ratio), which comprehensively evaluate the matching degree of return and risk (the Sharpe ratio is based on the risk-free interest rate, and the Sortino ratio only considers downside risk, which is more in line with investors' concerns about losses); algorithm efficiency indicators (average time spent on a single optimization solution), which reflects the practical application feasibility of the algorithm.

4.4. Experimental Results and Analysis

4.4.1 Return Metrics Analysis

The return performance of the four algorithms during the experimental period is shown in Table 1. As shown in the table, the DRMPOA algorithm achieved a cumulative return of 48.2% and an annualized return of 8.1%, significantly higher than the other three comparison algorithms: compared to the MV algorithm (cumulative return of 29.5% and annualized return of 5.8%), the cumulative return increased by 18.7 percentage points and the annualized return increased by 2.3 percentage points; compared to the Static-CVaR algorithm (cumulative return of 32.1% and annualized return of 6.3%), the cumulative return increased by 16.1 percentage points and the annualized return increased by 1.8 percentage points; and compared to the Dynamic-MCVaR algorithm (cumulative return of 38.7% and annualized return of 7.0%), the cumulative return increased by 9.5 percentage points and the annualized return increased by 1.1 percentage points [17]. Looking at returns by phase (as shown in Table 1 for the 2020 pandemic period and the 2021 structural market period), the DRMPOA algorithm experienced significantly smaller declines in 2020 (cumulative return -3.2%) than the MV algorithm (-7.8%) and the Static-CVaR algorithm (-5.5%), and only slightly higher than the Dynamic-MCVaR algorithm (-2.8%), demonstrating its resilience in extreme market conditions. Furthermore, during the 2021 structural market (cumulative return of 21.5%), its returns exceeded those of all the compared algorithms, demonstrating its profitability during market upturns. This is due to the dynamic prediction module's accurate capture of market trends and the robust optimization module's optimal weighting.

Table 1. Comparison of Return Indicators of Four Algorithms (January 2018-December 2023)

algorithm	Cumulative rate of return (%)	Annualized rate of return (%)	Cumulative yield in 2020 (%)	Cumulative yield in 2021 (%)	Return standard deviation (%)
DRMPOA	48.2	8.1	-3.2	21.5	1.8
MV	29.5	5.8	-7.8	15.2	2.5
Static-CVaR	32.1	6.3	-5.5	16.8	2.3
Dynamic-MCVaR	38.7	7	-2.8	18.6	2

4.4.2 Risk Indicator Analysis

The comparison of risk indicators is shown in Table 2. The DRMPOA algorithm achieves a daily CVaR of 1.2% at a 95% confidence level, meaning the maximum loss with a daily probability of no more than 5% is 1.2%. This value is lower than the MV algorithm (1.8%), Static-CVaR algorithm (1.5%), and Dynamic-MCVaR algorithm (1.3%), demonstrating its superior ability to control tail risk. This is because the robust optimization module introduces a risk buffer term into the objective function, reducing the likelihood of extreme losses. Regarding annualized volatility, the DRMPOA algorithm achieves 12.3%, a 6.2 percentage point decrease compared to the MV algorithm (18.5%), a 2.9 percentage point decrease compared to the Static-CVaR algorithm (15.2%), and a 1.5 percentage point decrease compared to the Dynamic-MCVaR algorithm (13.8%). Reduced volatility means greater stability in portfolio returns. Figure 1 (Annualized Volatility Over Time) further demonstrates that in March 2020 (peak of the pandemic) and October 2022 (global market correction), the DRMPOA algorithm's peak volatility was 18.5% and 16.2%, respectively, both lower than the other three algorithms (the MV algorithm's peaks were 28.3% and 25.1%, respectively). Furthermore, during periods of market stability (such as Q2 2021), volatility remained within the 8%-10% range, with even smaller fluctuations. This demonstrates the sustained risk control effectiveness of combining dynamic modeling with robust optimization.

Table 2. Comparison of Risk and Risk-Adjusted Return Indicators for Four Algorithms (January 2018-December 2023)

algorithm	95% Daily CVaR(%)	Annualized volatility (%)	Annualized Sharpe Ratio	Annualized Sortino Ratio	Maximum drawdown (%)
DRMPOA	1.2	12.3	0.58	0.72	-12.5
MV	1.8	18.5	0.32	0.39	-23.8
Static-CVaR	1.5	15.2	0.38	0.45	-18.6
Dynamic-MCVaR	1.3	13.8	0.45	0.51	-14.2

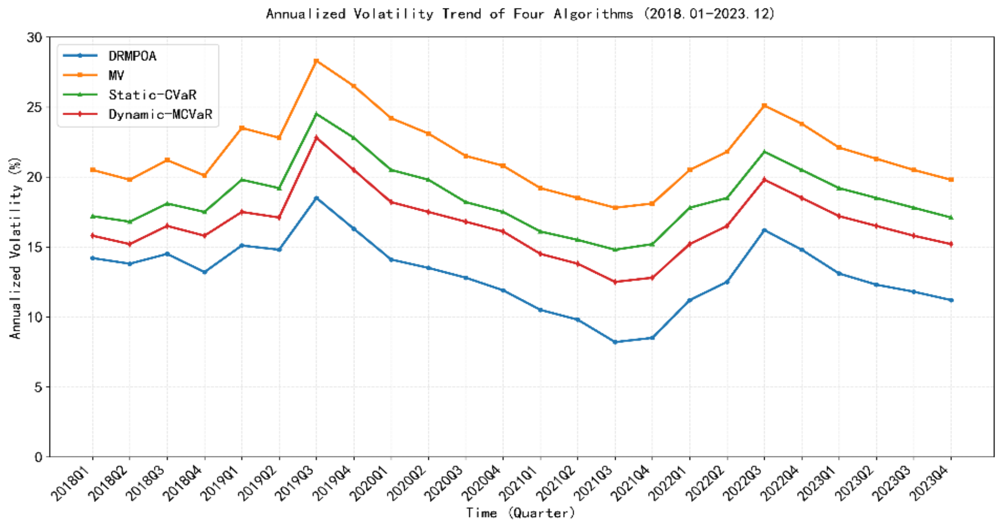

Fig. 1 Annualized volatility trends of four algorithms over time (January 2018-December 2023)

4.4.3 Analysis of risk-adjusted returns

Risk-adjusted returns better reflect the overall performance of an algorithm. The results of the annualized Sharpe ratio (the ratio of the excess of returns over the risk-free rate to volatility, with the risk-free rate being the average 10-year Treasury bond yield of 2.8%) and the annualized Sortino ratio (the ratio of the excess of returns over the risk-free rate to downside volatility) are shown in Table 2. The annualized Sharpe ratio of the DRMPOA algorithm is 0.58, an 81.3% increase compared to the MV algorithm (0.32), a 52.6% increase compared to the Static-CVaR algorithm (0.38), and a 28.9% increase compared to the Dynamic-MCVaR algorithm (0.45). The annualized Sortino ratio is 0.72, an 84.6% increase compared to the MV algorithm (0.39), a 60.0% increase compared to the Static-CVaR algorithm (0.45), and a 41.2% increase compared to the Dynamic-MCVaR algorithm (0.51). Figure 2 (Cumulative yield change trend chart) intuitively shows the yield growth path of the four types of algorithms: the cumulative yield curve of the DRMPOA algorithm is always above other algorithms, and the slope of the curve is more stable, without significant long-term sideways trading or large retracements, especially during the global market decline in 2022, the cumulative returns of other algorithms showed a significant decline (MV algorithm fell from 35.2% to 29.5%), while the DRMPOA algorithm only fell from 42.1% to 40.3%, with a smaller retracement, reflecting its significant advantage in risk-adjusted returns, which is due to the algorithm's precise balance between return pursuit and risk control.

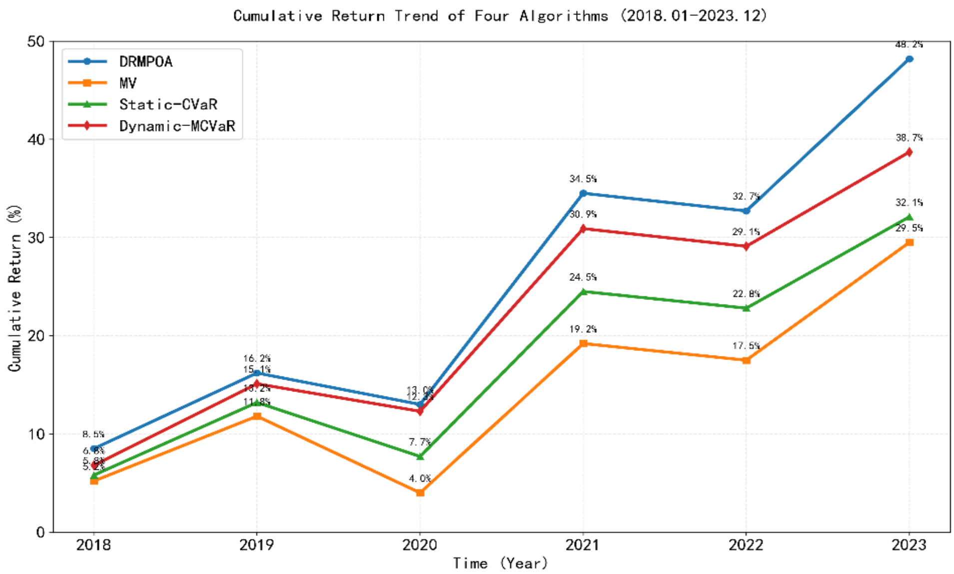

Fig. 2 Cumulative Return Trends of Four Algorithms (January 2018-December 2023)

4.4.4 Algorithm Efficiency Analysis

Algorithm efficiency directly impacts real-time decision-making capabilities in practical application scenarios. Experimental statistics show that the average time required for a single optimization solution for the four algorithms (including data preprocessing, prediction, and optimization) is: 12.5 seconds for the DRMPOA algorithm, 15.8 seconds for the Dynamic-MCVaR algorithm, 10.2 seconds for the Static-CVaR algorithm, and 8.7 seconds for the MV algorithm. The DRMPOA algorithm consumes slightly more time than the static algorithms (MV and Static-CVaR), but significantly less time than the Dynamic-MCVaR algorithm, which is also a dynamic algorithm. This is because the improved particle swarm optimization algorithm in DRMPOA accelerates convergence through dynamic adjustment of inertia weights, offsetting the additional computational cost incurred by the dynamic prediction module.

Figure 3 (Graph of Algorithm Iterations and Objective Function Value) shows that the DRMPOA algorithm converges to a stable objective function value (a composite return-risk indicator) after approximately 30 iterations, while the Dynamic-MCVaR algorithm requires over 45 iterations, further demonstrating DRMPOA's superior solution efficiency. For high-frequency trading or investment scenarios requiring real-time adjustments, a single iteration time of 12.5 seconds fully meets requirements, balancing algorithm performance with practical feasibility.

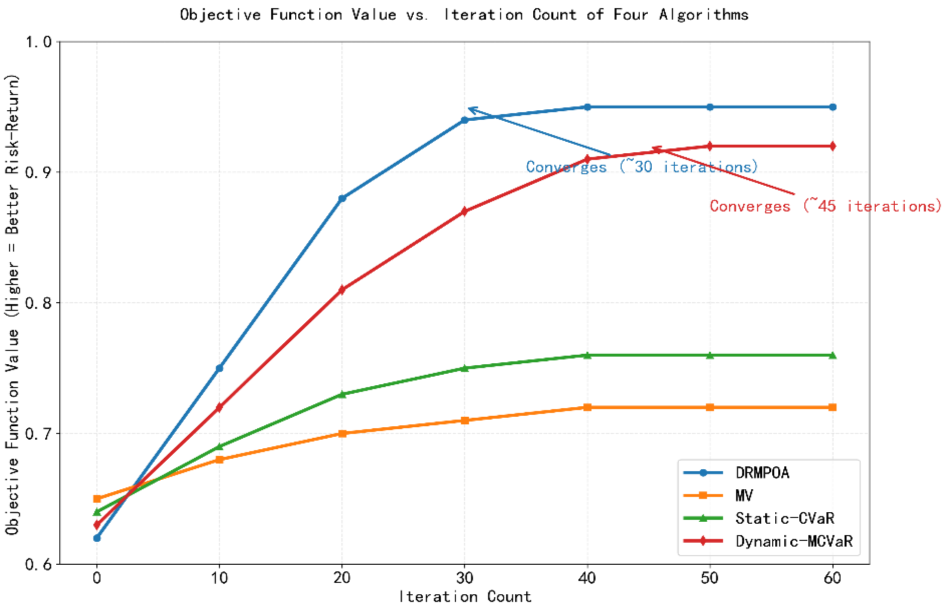

Fig. 3 Relationship between the number of iterations and the objective function value for four algorithm types

5. Conclusion

This paper designs the DRMPOA algorithm for the dynamic optimization of multi-asset portfolios and validates it using six years of cross-market data. The following conclusions are drawn:

Significantly Effective Algorithm Design: DRMPOA overcomes the static assumption limitations of traditional models through the collaborative "dynamic prediction - robust optimization" model. The improved LSTM model incorporates macro-attention weights to enhance the accuracy of return and volatility forecasts. The robust optimization module incorporates a risk buffer term, combined with a dynamic particle swarm solver, to achieve adaptive weight adjustment, making it more adaptable to complex markets than single dynamic or static models.

Experimental simulations validate the algorithm's advantages: In multi-asset experiments from 2018 to 2023, DRMPOA achieved optimal performance across all three dimensions: return, risk, and efficiency. Its cumulative return was 48.2% (8.1% annualized), representing increases of 18.7, 16.1, and 9.5 percentage points over MV, Static-CVaR, and Dynamic-MCVaR, respectively. Its 95th percentile daily CVaR was 1.2%, and its annualized volatility was 12.3%, demonstrating superior tail risk and volatility control. Its annualized Sharpe ratio was 0.58 (an increase of 28.9%), demonstrating outstanding risk-adjusted returns. A single solution took 12.5 seconds, with convergence after 30 iterations, meeting real-time decision-making requirements.

Clear application value: The algorithm demonstrated strong resilience during the 2020 pandemic (-3.2% decline) and the 2022 market correction (peak volatility of 16.2%). It also demonstrated profitability during the 2021 structural market rally (21.5% return). This provides institutional and individual investors with a tool for multi-asset allocation that is both theoretically supported and practically feasible. The algorithm can be expanded to more asset classes and market environments in the future.

The research also has limitations. The experimental data does not cover all asset types, and the quantification of macroeconomic factors can be further optimized. Future work will expand the asset range, integrate more optimization theories to improve algorithm performance, explore optimization strategies in extreme markets, and enhance the practicality of the model.

References

[1] Zhang, Bing-Hui, Yang, Chun-Bo, & He, Shou-Wu. Selection of cluster heads for wireless sensor network in ubiquitous power internet of things. Contemporary Economic Research, Vol. 357(2024) No. 5, p. 99-115.

[2] Li, Ai-Zhong, Ren, Rui-E, & Dong, Jian-Chao. Graph network risk perception and sparse low-rank portfolio management strategy. Chinese Journal of Management Science, Vol. 32(2024) No. 4, p. 58-65.

[3] Yang, Chun-Jiang, Zhou, Shu-Yan, Ding, Zhi-Xin, & Ma, Zheng-Chao. Quantification of investor risk preferences in asset allocation - On the correlation between long-term and short-term risk preferences. Chinese Journal of Management Science, Vol. 30(2022) No. 6, p. 11-21.

[4] Liu, Yan-Jun, Wu, Jian-Dong, & Zhang, Wei-Guo. Hybrid portfolio optimization model with flexible time horizon. Chinese Journal of Management Science, Vol. 32(2024) No. 1, p. 23-30.

[5] Zhang, Peng, & Zeng, Yong-Qiang. Multi-stage M-SAD portfolio optimization with total short selling restrictions. Chinese Journal of Management Science, Vol. 29(2021) No. 6, p. 60-69.

[6] Wang, Xiao-Qing, & Gao, Yong-Li. Mean-VaR multi-stage portfolio optimization model considering transaction costs. Journal of Engineering Mathematics, Vol. 37(2021) No. 6, p. 673-684.

[7] Petukhina, Alla, & Sprünken, Erin. Evaluation of multi-asset investment strategies with digital assets. Digital Finance, Vol. 3(2021) No. 1, p. 45-79.

[8] Kazemi-Rashnani, Mohammad, Mousavi, Seyed Vahid, & Hajizadeh, Ehsan. Multi-asset portfolio management including fixed income securities by value at risk-based models in Iran market. Financial Management Strategy, Vol. 10(2022) No. 3, p. 43-76.

[9] Thomann, Andreas. Multi-asset scenario building for trend-following trading strategies. Annals of Operations Research, Vol. 299(2021) No. 1, p. 293-315.

[10] Zhang, Xiao, Wu, Xin, Zhang, Li, & Chen, Zhi. The evaluation of mean-detrended cross-correlation analysis portfolio strategy for multiple risk assets. Evaluation Review, Vol. 46(2022) No. 2, p. 138-164.

[11] Ntare, Herman B., Mwamba, Jean W. M., & Adekambi, Felix. Dynamic Portfolio Optimization with Diversification Analysis and Asset Selection Amidst High Correlation Using Cryptocurrencies and Bank Equities. Risks, Vol. 13(2025) No. 6, p. 113.

[12] Li, Xin-Yang, Uysal, Ali S., & Mulvey, John M. Multi-period portfolio optimization using model predictive control with mean-variance and risk parity frameworks. European Journal of Operational Research, Vol. 299(2022) No. 3, p. 1158-1176.

[13] Popova, Irina, & Yau, Jack K. Computing optimal portfolios of multi-assets with tail risk: the case of bitcoin. Applied Economics Letters, Vol. 30(2023) No. 12, p. 1618-1626.

[14] Shu, Yuan-Jie, Yu, Chia-Lin, & Mulvey, John M. Dynamic asset allocation with asset-specific regime forecasts. Annals of Operations Research, Vol. 346(2025) No. 1, p. 285-318.

[15] Chen, Shu-Dong, Huang, Zhi-Qiang, & Yang, Xin-Yu. Multi-period fuzzy portfolio optimization model considering restricted short selling. Journal of Guangdong University of Technology, Vol. 38(2021) No. 2, p. 39-47.

[16] Li, Ai-Zhong, Ren, Rui-E, & Dong, Jian-Chao. Asset portfolio and pricing based on multi-factor matrix regression under financial network risk. Chinese Journal of Management Science, Vol. 29(2021) No. 6, p. 1-9.

[17] Yang, Rui-Chang, & Xing, Wei-Zhong. Research on bond portfolio optimization strategy based on CRMW risk mitigation utility target tracking. Chinese Journal of Management Science, Vol. 30(2022) No. 7, p. 150-163.