Application of Multi-Armed Bandit Algorithm in Quantitative Finance

Chengxun Chen[1], *Xuanyuan Liu*[2], *Yanyan Ma*[3], and *Xiaole Zuo*[4*]

[1]International Digital Economy College, Minjiang University, 350108 Fuzhou, Fujian Province, China
[2]School of Mathematics and Statistics, Zhengzhou University, 450001 Zhengzhou, Henan Province, China
[3]Bachelor of Arts, Dalhousie University, B3H 4R2 Halifax, Nova Scotia, Canada
[4]School of Mathematics, South China University of Technology, 510641Guangzhou, Guangdong Province, China

Abstract. The volatility and diversity of financial markets make it challenging for a single portfolio achieve better returns, therefore, adjustable portfolios based on the risk tolerance of clients are highly demanded. However, traditional portfolio strategies cannot meet this requirement. Regarding this issue, the paper combines Fuzzy C-means (FCM) with the Upper Confidence Bound (UCB) algorithm, Genetic Algorithm (GA) optimizing UCB parameters (GA-UCB) and UCB redefining the fitness of GA (UCB-GA) to construct an investment portfolio strategy that can be dynamically adjusted. The research methodology is as follows: the assets are grouped by FCM, using UCB to find the best cluster among the groups; UCB, UCB-GA, and GA-UCB are used to refine the weight distribution of the best cluster. The result shows that the cumulative return of the cluster recommended by the UCB is significantly higher than that recommended by FCM, the Sortino Ratio is improved by 1.18, and the Maximum Drawdown is reduced by 8%. In terms of the weights of the optimal cluster; the portfolio strategy from GA-UCB has the highest cumulative return of approximately 250% in algorithms. The Sortino Ratio of the GA-UCB is the largest at 3.23, which is 1.5 and 1.63 higher than the UCB and the UCB-GA, respectively. In addition, the Maximum Drawdown of the GA-UCB is 26%, which is 1% lower than UCB-GA and 3% lower than UCB. Combining FCM and GA-UCB can improve investment return and stability by adjusting the portfolio weight, which leads to better return risk ratios.

1 Introduction

In recent years, quantitative investment has been widely applied in the financial field, particularly investment strategies based on multi-factor models, which have gradually become mainstream [1]. A multi-factor model predicts future performance by analyzing various economic and market factors affecting asset returns, such as market capitalization, value, price-to-earnings ratio, momentum, and other macroeconomic indicators. Its core idea

* Corresponding author: mailto:202130322407@mail.scut.edu.cn

is that different factors provide important information about potential asset returns, helping investors build more effective portfolios. Mean-Variance Model optimizes portfolios by balancing expected returns with risk (variance of returns). However, this static strategy often underperforms in dynamic market environments due to rapidly changing conditions that affect asset return distributions and volatility. In this context, dynamic decision models like the Multi-Armed Bandit (MAB) algorithm significantly enhance strategy adaptability. The MAB algorithm continuously learns and adjusts investment strategies in response to changing market conditions, evaluating the return potential of various assets. This enables quick responses to new information, effectively reducing risks from market fluctuations. Additionally, combining the MAB algorithm with Reinforcement Learning (RL) can further optimize investment decisions in complex environments [2]. This study aims to combine FCM with the MAB algorithm, using GA to optimize model parameters, improve overall return performance, and verify effectiveness through actual data experiments [3].

2 Related work

In recent years, the MAB algorithm has attracted much attention in the field of quantitative finance due to its excellent dynamic optimization capabilities and efficient exploration and exploitation trade-offs. The mainstream MAB algorithms include the Explore Then Commit (ETC) algorithm, the UCB algorithm, and the Thompson Sampling (TS) algorithm, which can flexibly respond to market changes by continuously updating selection strategies. The ETC algorithm emphasizes extensive evaluation of various investment options in the initial stage to make better decisions in the subsequent stages; The UCB algorithm balances exploration and utilization by calculating the confidence limits of each option, ensuring the best balance between risk and return; The TS algorithm uses probability models to evaluate the potential benefits of different options, achieving more accurate decision-making [4, 5]. These adaptive algorithms can respond to market changes in real time, quickly adjust strategies to achieve better return risk ratios in a fiercely competitive market. In practical applications, the MAB algorithm has demonstrated its flexibility. For example, in a significantly volatile market, the MAB algorithm can quickly shift towards more stable assets, effectively reducing the overall risk of the portfolio. In addition, researchers are also exploring the combination of FCM and MAB algorithms to achieve more detailed investment classification and selection. FCM can classify assets into different categories based on characteristics, helping investors identify potential investment opportunities. After combining the MAB algorithm, investment strategies could be dynamically adjusted based on market feedback, thereby enhancing adaptability in complex market environments and increasing return. At the same time, the techniques of RL and Deep Reinforcement Learning (DRL) are gradually being integrated into quantitative finance [6, 7]. The self-learning ability of RL enables models to dynamically adjust in constantly changing markets, further improving portfolio returns and reducing risks. DRL utilizes deep neural networks to model complex market environments, demonstrating stronger predictive and decision-making abilities when dealing with high-dimensional data. Combining the MAB algorithm with these technologies, investors can more accurately grasp market trends and develop more targeted investment strategies. These innovative research trends indicate that portfolio management strategies are gradually transitioning towards more intelligent and adaptable dynamic models for complex markets.

3 Methodology

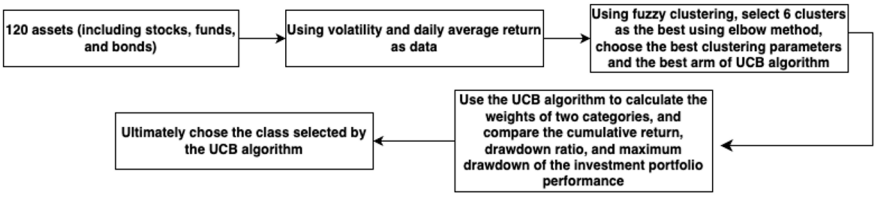

Fig. 1. Overall Experimental Framework

Figure 1 presents the overall experimental framework. This study proposes a portfolio strategy based on FCM analysis and the MAB algorithm. The dataset, sourced from Yahoo Finance, comprises 120 assets, including stocks, funds, and bonds.

$$\underset{(Volatility)}{V_o(R,W) = \sqrt{W^\top R^\top W}} \tag{1}$$

Formula 1 represents the entire portfolio's volatility as a standard deviation of the portfolio's returns and serves as a measure of the portfolio's risk. $W^\top$ represents the transpose of the weight vector, where W is the vector of weights assigned to each asset within the portfolio. The matrix R denotes the matrix of all of the asset daily returns.

$$MD = \max_{i\in[1,T]}\left(\frac{CW_i - \min_{j\in[i,T]} C\,W_j}{CW_i}\right) \tag{2}$$

Formula 2 measures the maximum drawdown of a portfolio over a specific period, capturing the largest peak-to-trough decline in the portfolio's value. Maximum drawdown is a crucial risk metric, reflecting the greatest potential loss a portfolio might face over a given period. CW_i denotes the portfolio's value at time i. And $\min_{j\in[i,T]} C\,W_j$ represents the minimum cumulative wealth between the time i and T.

$$PR(W,R) = \sum_{i\in stocks} W_i \cdot R_i \tag{3}$$

Formula 3 calculating the portfolio's daily return is calculated as the sum of the weighted daily returns of each asset. Each term $W_i \cdot R_i$ corresponds to the weight of a specific asset multiplied by its return.

$$SR = \frac{Portfolio\ Return\text{-}Risk\text{-}Free\ Rate}{Downside\ Deviation} \times \sqrt{252} \tag{4}$$

Formula 4 is a metric used to measure the risk-adjusted returns of a portfolio, focusing solely on downside risk. The term $\sqrt{252}$ represents the annualization factor because stock markets typically operate around 252 trading days per year [8].

$$CWF(W,R) = \prod_{i=1}^{t}\left(1 + \sum_{a \in stocks} W_a \cdot R_{ai}\right) \tag{5}$$

Formula 5 is used to calculate the cumulative wealth.

$$CW(R_a) = \prod_{i=1}^{t}(1 + R_{ai}) \tag{6}$$

$CW(R_a)$ in Formula 6 represents the cumulative wealth of a single asset a, generated by sequential product the daily wealth growth rate over the period.

3.1 FCM

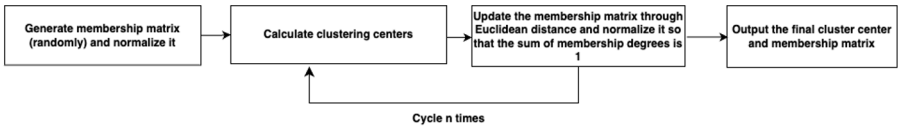

Fig. 2. The Rough Process of FCM

Figure 2 illustrates the rough process of FCM. In the stock pools, FCM is applied using the average daily returns and volatility of stocks as input data 9. Before clustering, the data is standardized to prevent overflow caused by excessively large or small values. The following Formula 7 is used for standardization:

$$\overline{x_i} = \frac{x_i - \mu}{\sigma} \tag{7}$$

x_i represents the ith number point, μ is the average of the entire data set, σ measures the dispersion of the data. $\overline{x_i}$ shows the standardization of x_i. FCM involves several key Formulas:

$$v_j = \frac{\sum_{i=1}^{N}(u_{ij})^m x_i}{\sum_{i=1}^{N}(u_{ij})^m} \tag{8}$$

In Formula 8, v_j means the center of the jth cluster, u_{ij} is the membership degree of the ith sample to the jth cluster. m represents the fuzzy exponent, a parameter controlling the fuzziness of the clustering. In this case, m is set to 2. In addition, the feature vector of the ith sample is x_i and N indicates the total number of samples.

$$d_{ij} = \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^2 \right)^{\frac{1}{2}} \tag{9}$$

In Formula 9, d_{ij} is the Euclidean distance between the ith sample and the jth cluster center. x_{ik} and x_{jk} are the values of the kth feature for the ith sample and the jth cluster center, respectively.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}} \tag{10}$$

In Formula 10, C is the number of clusters, and $\frac{2}{m-1}$ is the adjustment factor of weight, indicating the degree of influence of distance in membership calculation.

$$u_{ij} = \frac{u_{ij}}{\sum_{j=1}^{C} u_{ij}} \tag{11}$$

Formula 11 is used to update and normalize each element in the membership matrix. In order to ensure that the sum of membership degrees for each sample across all clusters equals 1.

3.2 UCB

The core idea of the UCB algorithm is to balance exploration and exploitation at each time step. Figure 3 shows the pseudocode of the UCB algorithm.

UCB Algorithm

Input: A set of arms k with unknown reward distributions.
Output: Selected arm kt at each time step t and its cumulative reward.

for k in 1 to K:
  $N_k \leftarrow 0$
  $S_k \leftarrow 0$
end for
for t in 1 to T:
  if $N_k > 0$:
    $\mu_k \leftarrow \frac{S_k}{N_k}$
    $UCB_k \leftarrow \mu_k + \sqrt{\frac{2 \log t}{N_k}}$
    $k_t \leftarrow \arg\max_k \left(\mu_k + \sqrt{\frac{2 \log t}{N_k}} \right)$
  end if:
  $N_{k_t} \leftarrow N_{k_t} + 1$
  $S_{k_t} \leftarrow S_{k_t} + r_t$
end for
return 0

Fig. 3. Pseudocode of UCB

K is the total number of arms; N_k is the number of times arm k is selected. The cumulative reward of arm k is symbolized S_k. T and t mean the total number of time steps

and the current time step, respectively. μ_k and UCB_k represent the average reword and the UCB value of arm k, respectively. The immediate reward obtained after selecting arm k_t at time step t is r_t in pseudocode. Moreover, $R_k(t)$ is a function that returns the reward for selecting arm k_t at time t [10].

UCB could be also used to calculate the weights of different stocks in a portfolio. This thesis uses rewards for all the arms to normalize and then gets the weights of the arms.

3.3 GA optimization

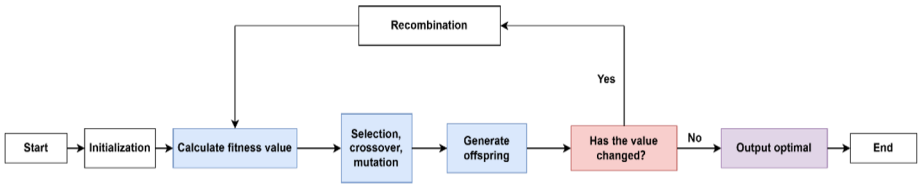

Fig. 4. GA Overall Process

Figure 4 simply shows how to compile a Genetic Algorithm, which is not a key in the thesis. How to use GA optimizing UCB will be shown in the following.

3.3.1 UCB-GA

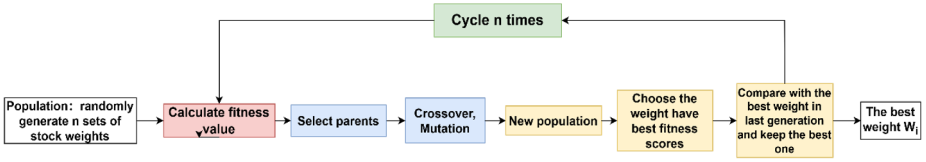

Fig. 5. UCB-GA Overall Process

Figure 5 depicts the overall process of UCB-GA. In this algorithm, y_i is the ith individual of the population, and y_i means the weight vector of different assets in the portfolio, which is generated randomly. Formula 12 is the reward equation:

$$reward_i = \mu_i + \sqrt{\frac{c \cdot ln(n)}{T_{n_i}}} \tag{12}$$

μ_i represents the average award for arm i, Represents the number of times the arm i is selected at time n. Formula 13 is the fitness function:

$$f(y_i) = PR(y_i, R) - \kappa \cdot V_o(R, y_i) \tag{13}$$

From Formula 13, $PR(y_i, R)$ represents the portfolio's total return, κ represents the risk-adjusted parameter, and $V_o(R, y_i)$ represents the volatility function.

$$k(y_i) = \frac{f(y_i)}{\sum_{i=1}^{n} f(y_i)} \tag{14}$$

$$p(i) = \frac{e^{\beta f(y_i)}}{k(y_i)} \tag{15}$$

Formula 14 and Formula 15 show a process to generate probabilities for each asset to select parents [3]. And β helps higher fitness get the higher probability of being chosen.

$$p(i) = \frac{p(i)}{\sum_{i=1}^{n} p(i)} \tag{16}$$

Formula 16 ensures the sum of selection probabilities for all the population to be 1. Formula 17 is the cross function:

$$H = \frac{R + 2\sqrt{r}}{3R} \tag{17}$$

H denotes the crossover point, R represents the total number of iterations, and r is the current iteration number [3]. Figure 6 shows an intersection simulation image.

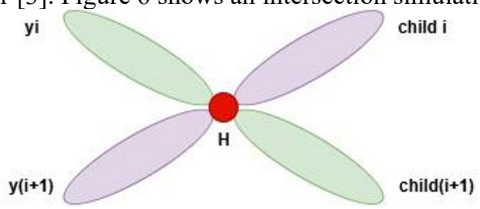

Fig. 6. Intersection Simulation Image

Formula 18 is the mutation function:

$$\mathrm{A} = \frac{1}{1 + \mathrm{H}} \tag{18}$$

A represents population similarity. If A>0.5, then child_i is randomly mutated and normalized to be a weights vector [3].

The entire time horizon is first divided into multiple Minimum Formation Periods (MFP). For each MFP, the UCB-GA is applied to optimize the portfolio by determining the optimal weights for each asset during that specific period. Afterward, the weights from all MFP are aggregated to form a comprehensive portfolio that accounts for the optimal weight distributions across different time periods. In this case, MFP is defined as 47, with a penalty coefficient of 0.8 and a UCB parameter of 0.8.

3.3.2 GA-UCB

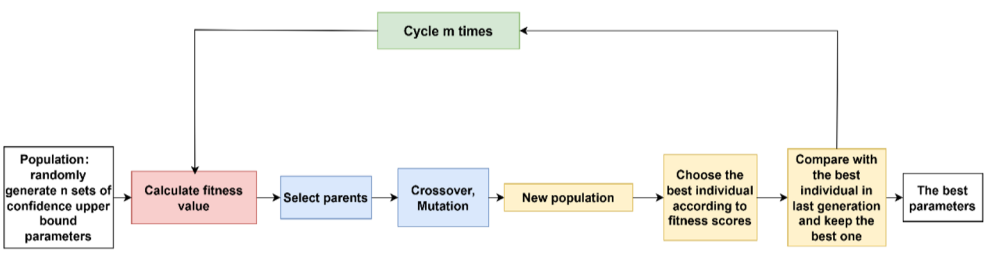

Fig. 7. GA-UCB General Process

Figure 7 describes the general process of GA-UCB. Formula 19 ensures that the range is controlled between [0.1,5.0].

$$\text{reward}(i, c_i) = \mu_i + c_i \sqrt{\frac{\ln(n)}{Tn_i}} \tag{19}$$

In this case, different exploration coefficients c_i are set for arms firstly, allowing arms with higher rewards to be explored more frequently, rather than using a uniform value for exploration as Formula 12. Secondly, the UCB parameter c_i is moved outside the square root in the confidence upper bound Formula. This adjustment prevents the confidence upper bound from changing too slowly when c_i is small, thereby avoiding local solutions, leading to bad performance in the model. As a result of this adjustment, fitness changes more stably, reducing the number of iterations and accelerating convergence. These modifications are aimed at enhancing the global optimization capability of the algorithm.

$$w(i, c_i) = \frac{\text{reward}(i, c_i)}{\sum_{i \in \text{arms}} \text{reward}(i, c_i)} \tag{20}$$

$$W(c) = \{w(i, c_i)\}_{i \in \text{arms}}$$

In Formula 20, $w(i, c_i)$shows weight for stock i, and $W(c)$ means a set including all the weights for the portfolio. In Formula 21, the loss function is defined as:

$$l(W, R, \kappa) = CWF(W, R) - \kappa \cdot V_o(R, W) \tag{21}$$

In Formula 22, chromosome y_i represents the other j parameter groups in n populations, and y_{ji} represents the upper confidence bound parameter of arm i for the jth parameter group [11].

$$f(y_i) = -l(W(y_i), R, \kappa) \tag{22}$$

For the selection function and cross function, see Formula 14-17. Mutation function is shown in Formula 23, $\mathcal{N}\left(0, \left(\frac{1}{1+H}\right)^2\right)$means a random number.

$$\text{child}_i = \text{child}_i + \mathcal{N}\left(0, \left(\frac{1}{1+H}\right)^2\right) \tag{23}$$

Formula 24 keeps the child individual satisfied with $\text{child}_i \in [0.1, 5]$.

$$\text{child}_i = \min(\max(\text{child}_i, 0.1), 5) \tag{24}$$

As the mutation in UCB-GA, if $A > 0.5$ is satisfied, the mutation occurs.

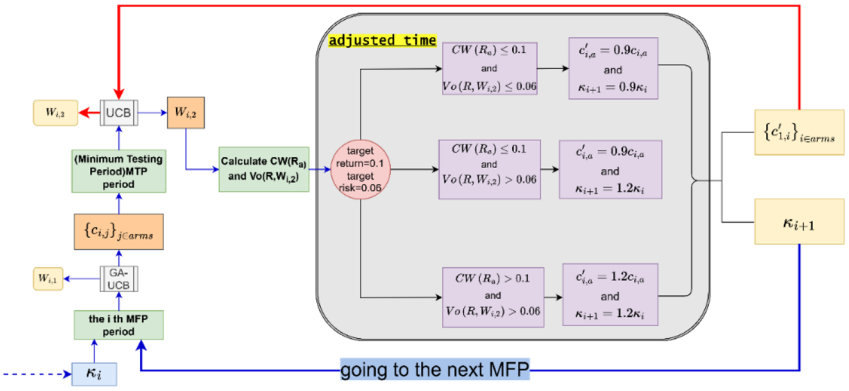

Fig. 8. The Process of Adjusting the Correlation Coefficient

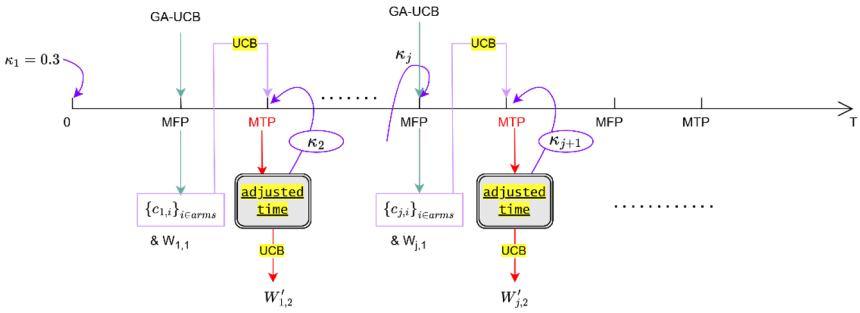

Fig. 9. GA-UCB MFP and MTP Rough Process

In Figure 8 and Figure 9, $W_{i,1}$, $W_{i,2}$ mean the weights in the ith MFP and the weight in the ith Minimum Time Period (MTP) respectively. It is ensured that MFP is always greater than MTP. Initially, a set of confidence parameters is generated during the MFP phase, at the same time $W_{i,1}$ is also generated by GA-UCB and then added to the weight list. In the MTP phase, UCB uses the parameters generated in MTP to get $W_{i,2}$, leading to cumulative portfolio returns, and subsequently, daily returns for each stock and maximum drawdown for the portfolio are calculated. The UCB weights ($W'_{i,2}$) are recalculated and added to the weight list after parameter adjustments, iterating this MFP+MTP process continuously until all weights are obtained.

4 Experiment

4.1 Preprocessing

4.1.1 FCM

Firstly, the elbow method is used to determine the optimal cluster numbers. The specific operation of the elbow method includes drawing a graph of the relationship between the sum of squared errors (SSE) and the number of clusters and determining the 'elbow point' where the marginal gain begins to significantly decrease, as a reference for the optimal number of clusters.

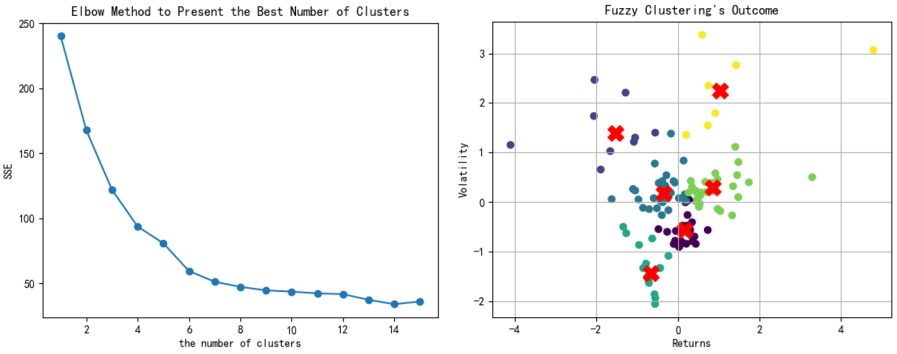

Fig. 10. Elbow Method Curve (left) Cluster Distribution (right)

Figure 10 (left) illustrates the negative relationship between SSE and the number of clusters. The result shows the decline of SSE gradually slows down after the 6th point, which means 6 clusters as the optimal point. Then used FCM to group the data into clusters, with each cluster representing a portfolio of stocks with distinct characteristics.

Using the above result, Figure 10 (right) displays the cluster distribution. From the clustering distribution plot, different colored points represent stocks from different clusters. Red crosses denote the centroid positions of the clusters, namely cluster centers. The plot illustrates the distribution of stocks within each cluster in terms of returns and volatility. Typically, stocks within the same cluster exhibit high similarity, meaning that their returns and volatility are relatively close to one another when being standardized, while there are distinct differences between different clusters.

Table 1. Cluster group

Cluster		1	2	3	4	5	6
Centroid	Return	-0.3532	-0.6645	0.8591	-1.5251	0.1663	1.0354
	Volatility	0.1899	-1.4383	0.2875	1.3916	-0.5585	2.2554

Table 1 lists the centroid return and volatility for all clusters. According to Fig. 11 and the explanation of the cluster center's equation, Cluster 3 should have a good performance, owing to its higher centroid return and lower volatility.

4.1.2 UCB Selection

Using the UCB algorithm, high-performing clusters were identified. However, the cluster selected by the UCB algorithm is Cluster 6 but not Cluster 3. This inconsistency suggests that the UCB algorithm can adjust its focus on clusters based on their real-time performance instead of relying on fixed average values. This dynamic prioritization allows the algorithm to be more responsive and effective in changing market conditions.

4.2 Portfolio optimization with UCB

Figure 11 compares different investment portfolios. The portfolio was further optimized using the UCB algorithm. UCB algorithm dynamically adjusted the investment weights to achieve higher returns with lower risks.

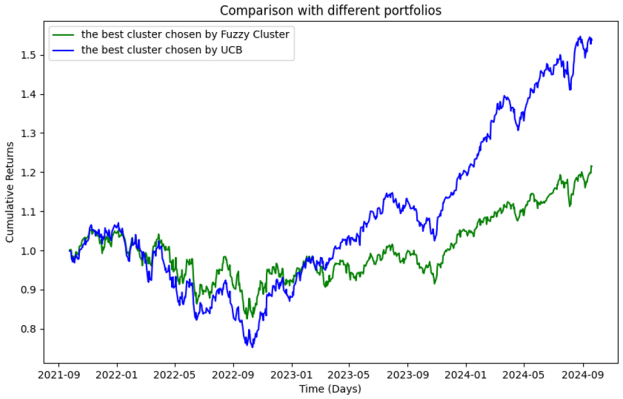

Fig. 11. Comparison of Different Portfolios

Table 2 shows that the optimal clustering result has a Sortino Ratio of -0.24 and a Maximum Drawdown of -0.22, while the UCB-selected cluster has a Sortino Ratio of 1.60 and a Maximum Drawdown of -0.30. Although Sortino Ratios alone do not determine superiority, the UCB selection shows clear performance advantages. However, the optimal clustering result's negative Sortino Ratio indicates poor returns under downside risk, whereas the UCB's positive ratio signifies strong risk-adjusted returns. UCB effectively balances risk and return across market conditions, providing more stable investment choices. Furthermore, Fig. 11 shows the higher cumulative return for the cluster chosen by UCB than that of FCM. Overall, UCB outperforms the best clustering result, confirming it as a superior selection method.

Table 2. Indicators corresponding to the optimal cluster investment portfolio of UCB

Cluster	Sortino Ratio	Maximum Drawdown
Cluster selected by the best clustering results (cluster 3)	-0.24	-0.22
Cluster selected by UCB (cluster 6)	1.60	-0.30

4.3 Comparison of GA-UCB, UCB-GA, and UCB

In this part, the paper compared the performance of GA-UCB, UCB-GA, and UCB algorithms in portfolio optimization. Figure 12 compares the cumulative weighted returns of GA-UCB, UCB-GA, and UCB. The red line represents the cumulative return trend of the GA-UCB investment portfolio. The initial phase, extending until mid-2022, displays fluctuations and a period of decline. Cumulative returns have a marked and steady increase from early 2023, peaking at approximately 2.5. The blue line shows the result of UCB-GA. UCB-GA has a similar trend as GA-UCB. While the peak of cumulative return is lower than that generated by GA-UCB. In addition, the green curve is UCB, and the trend of UCB is very similar to that of UCB-GA. There is a period of overlap in early 2024 between UCB and UCB-GA, but its peak value of UCB is the least in algorithms.

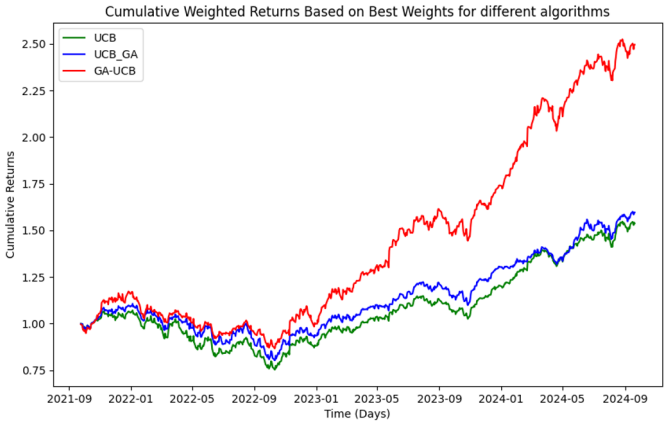

Fig. 12. Comparison of Three Algorithms

Table 3. Comparison of UCB-GA, GA-UCB, and UCB indicators

	Sortino Ratio	Maximum Drawdown
UCB-GA	1.7	-0.27
GA-UCB	3.23	-0.26
UCB	1.60	-0.30

In the table 3, the Sortino Ratio of GA-UCB is 3.23, highlighting its exceptional performance in achieving higher excess returns for unit downside risk, making it stand out in risk-adjusted returns. Its Maximum Drawdown of -0.26 indicates strong risk resistance. In comparison, UCB-GA has a Sortino Ratio of 1.73, which, while still good, demonstrates less effectiveness in managing downside risk. Compared with GA-UCB, the Maximum Drawdown for UCB-GA is -0.27, showcasing solid, but slightly lower, risk resistance during market fluctuations. The UCB expresses the worst performance of the Sortino Ratio and Maximum Drawdown in algorithms, which are 1.6 and -0.30, respectively. Overall, GA-UCB clearly outperforms in both excess returns and risk management.

5 Conclusion

In the financial market, investment portfolio adjustment is the key to avoiding risks and maximizing profits. Thus, introducing a suitable dynamic decision-making model to adapt to market volatility is an important research field for optimizing investment portfolio strategy. This paper proposes an adjustable investment portfolio strategy with FCM and UCB algorithm, improving UCB parameters by GA. The research clusters assets by FCM and selects the optimal investment portfolio by the UCB algorithm. The portfolio weights are derived from UCB and the parameter of the UCB algorithm by GA to enhance the performance. Comparing UCB and UCB-GA models in the past real stock data to verify the efficiency of the GA-UCB model. The results show that the cumulative return of the investment portfolio chosen by the UCB is obviously higher than FCM; the Sortino Ratio is 1.18 higher than FCM. The maximum Drawdown from the UCB recommendation cluster is 8% less than FCM. In terms of the investment portfolio strategy, the cumulative return of GA-UCB is about 250%, which is clearly higher than UCB-GA and UCB. The Sortino Ratio are 3.23, 1.73, and 1.6 respectively for GA-UCB, UCB-GA, and UCB. Furthermore, the Maximum Drawdown of the GA-UCB is 26%, which is outperforming the 27% for UCB-GA and the 30% for UCB. The reason for the better performance of GA-UCB is that the

weights generated by the UCB algorithm are closer to the optimal solution, whereas UCB-GA requires a sufficiently large sample size and a large number of iterations to achieve the solution as good as UCB due to random generation of weights by GA. Compared with the UCB algorithm, GA-UCB can assign more weights to well-performing stocks by optimizing the UCB upper bound parameters through GA. In conclusion, the combination of FCM and GA-UCB can enhance the stability of the investment portfolio; also, significantly increasing the cumulative return and improving investment efficiency. However, this study does not consider realistic factors that may affect investment returns, such as transaction costs. Moreover, the data is divided into multiple sets to form MFPs and MTPs, which results in insufficiently large data for each period and may lead to poor performance of the model in different markets. In the future, the study could add more parameters and enough data to improve the adaptability and accuracy of the model in different scenarios, thus providing more informative investment portfolio strategy recommendations to investors.

Authors Contribution
All the authors contributed equally and their names were listed in alphabetical order.

References

1. Kabir M A, Liping Y, Sarker S K, et al. Portfolio optimization and valuation capability of multi-factor models: an observational evidence from Dhaka stock exchange[J]. Frontiers in Applied Mathematics and Statistics, 2023, 9: 1271485.(P7-8)
2. Aboussalah A M, Lee C G. Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization[J]. Expert Systems with Applications, 2020, 140: 112891.(P11)
3. Guo Z, Kang G. Financial Investment Optimization by Integrating Multifactors and GA Improved UCB Algorithm[J]. Informatica, 2024, 48(13).(P116-125)
4. Ni H, Xu H, Ma D, et al. Contextual combinatorial bandit on portfolio management[J]. Expert Systems with Applications, 2023, 221: 119677.(P8-10)
5. Zhu M, Zheng X, Wang Y, et al. Adaptive portfolio by solving multi-armed bandit via thompson sampling[J]. arXiv preprint arXiv:1911.05309, 2019.(P3-4)
6. Charpentier A, Elie R, Remlinger C. Reinforcement learning in economics and finance[J]. Computational Economics, 2021: 1-38.(P430-431)
7. Liu X Y, Yang H, Chen Q, et al. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance[J]. arXiv preprint arXiv:2011.09607, 2020.(P3-4)
8. Rollinger T N, Hoffman S T. Sortino: a ‘sharper’ratio[J]. Chicago, Illinois: Red Rock Capital, 2013.(P3-8)
9. Song Zongxiang. Application of Fuzzy C-Means Clustering in Stock Investment [D]. Northeast Petroleum University [2014-10-21] DOI:CNKI:CDMD:2.1017.096721.(P13-17)
10. Manome N, Shinohara S, Chung U. Simple modification of the Upper Confidence Bound algorithm by generalized weighted wverages. arXiv preprint arXiv:2308.14350, 2023.(P2)
11. Almulla H, Gay G. Learning how to search: generating effective test cases through adaptive fitness function selection[J]. Empirical Software Engineering, 2022, 27(2): 38.(P2)