Stock Price Prediction based on the Improved Flower Pollination Algorithm Optimizing BP Neural Network

Pengkai Wang*

International College, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd, Beijing, China 100876;

*Corresponding author: wpk@bupt.edu.cn

Abstract. This research introduces a predictive model, IFPA-BP, for stock price forecasting that optimizes the BP neural network weights and biases using the Improved Flower Pollination Optimization Algorithm (IFPA). We addressed the traditional inflexibility between global and local searches by introducing the concepts of adaptive conversion probability and temperature. To tackle the issue of population diversity, a chaotic reverse initialization strategy was employed, significantly reducing the local optimum challenges common with conventional flower pollination algorithms. The efficacy of IFPA was demonstrated using five benchmark functions. We subsequently used the IFPA-BP model to forecast the stock prices of Guoxin Securities. Notably, the IFPA-BP's MSE, MAPE, MAE, and RMSE metrics outperformed those of the traditional BP model, suggesting superior forecasting ability and providing valuable insights for financial investments.

Keywords: Improved Flower Pollination Optimization Algorithm, BP Neural Network, Stock Price, Prediction

1 Introduction

The conventional Flower Pollination Algorithm (FPA) has been developed based on the study of pollen swarm intelligence behavior, drawing inspiration from the natural process of flower pollination, and it typically employs mathematical models for optimization [1]. The FPA has demonstrated its efficacy in various fields including algorithm optimization, data processing, and model training. However, with ongoing in-depth research and increasing real-world applications, certain limitations of the traditional FPA, such as issues with local optima, convergence speed, and parameter selection, have become increasingly noticeable. In response to these issues, this paper proposes improvements to the conventional algorithm in areas such as parameter selection, population diversity, and location changes of algorithmic components. By applying five distinct test functions to the improved algorithm, experimental analyses demonstrate its commendable performance in aspects of convergence speed, optimal fitness, and escaping local optima. We applied the Improved FPA (IFPA) to optimize a Backpropagation (BP) neural network prediction model (referred to as IFPA-BP) to

C. Chen et al. (eds.), *Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, Atlantis Highlights in Computer Sciences 17,
https://doi.org/10.2991/978-94-6463-304-7_26

forecast the stock prices of Guoxin Securities. Comparative experiments with the traditional BP neural network prediction model reveal superior outcomes achieved by the IFPA-BP model, underscoring its value as a reference for stock price prediction.

2 Basic Flower Pollination Algorithm

The FPA algorithm draws inspiration from the natural process of flower pollination, simulating two primary behaviors found in nature: self-pollination and cross-pollination. Pollinators in this process are classified into two categories: biological and non-biological. Specifically, biological pollination, typically carried out by animals or insects, is viewed as cross-pollination, facilitating wide-ranging dissemination. In contrast, non-biological pollination, mainly propagated through the wind, is considered self-pollination. To leverage FPA for solving optimization problems, the algorithm presumes that each plant bears only one flower, and each flower contains just one pollen embryo, with each embryo corresponding to a solution within the optimization problem. Moreover, the algorithm mandates adherence to the following four stipulations.

1. The process of cross-pollination in plants is analogous to the global search phase in the algorithm. During this phase, the global search is conducted using biological entities as carriers, following the pattern of Levy flight.
2. The non-biological self-pollination process equates to the local search phase of the algorithm. Here, local search is facilitated through wind-driven pollination among plants of the same species.
3. The probability of reproduction is related to the specific traits of the flowers. The similarity and connection between two flowers (or two individual entities within the algorithm) are proportionate.
4. The probability transition parameter p, with a range of [0,1], controls the interchange between global search (or global pollination) and local search (or local pollination) within the FPA algorithm. Influenced by factors like location and wind, the likelihood of local pollination typically outweighs that of global pollination during the entire process.

Using the rules, we formulated the subsequent mathematical models.

1. The formula representing pollen's pollination during its global phase [2]

$$X_i^{t+1} = X_i^t + L(X_i^t - g_{best}) \tag{1}$$

In this context, X_i^{t+1} and X_i^t signify the solutions for the t+1 and t generations, g_{best} represents the global optimum achieved in a single iteration. The step size L, adheres to the Lévy distribution, and its computational formula is presented below

$$L \approx \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}} \quad S > 0 \tag{2}$$

Within this expression: λ serves as a scaling factor, with reference setting $\lambda=1.5$ [3]; $\Gamma(\lambda)$ denotes the standard gamma function; and s represents the step length.

2. The formula for updating position in the local pollination phase [4]

$$X_i^{t+1} = X_i^t + \varepsilon\left(X_j^t - X_k^t\right) \quad (3)$$

Within the given equation: ε represents a random number uniformly distributed between [0,1]; X_j^t and X_k^t correspond to the pollens of two distinct flowers, both from the same species.

3. The mechanism dictating the transition between global and local pollination

The switch between global and local pollination can be modulated by tuning the conversion probability, p, which lies in the interval [0,1]. Extensive simulation experiments demonstrate that setting p to 0.8 allows the algorithm to realize its best search performance [5].

3 Enhancements to the Flower Pollination Algorithm

3.1 Initializing the Pollen Population

The conventional flower pollination algorithm, given its random approach to pollen position initialization, struggles with local optimization issues. Addressing this, we propose a chaotic reverse initialization approach. Drawing from chaotic mapping and reverse learning principles, our method synergizes Logistics mapping, Tent mapping, and reverse learning. Introducing a reverse chaotic sequence, we weave the hybrid mutation control strategy into the established flower pollination algorithm. On one side, the Logistics [6] and Tent mappings guarantee uniform population distribution, individual association, and broadened diversity, importing the chaotic mapping's exploratory, random, and overarching stability attributes into the classic algorithm. Conversely, by leveraging the reverse learning strategy, we amplify population diversity and uniformity, broadening the algorithm's search spectrum and embedding reverse learning's explorative, random, and holistic adaptiveness [7].

Addressing the traditional flower pollination algorithm's limited divergence and stability, Logistics mapping ensures an enhanced, uniform population distribution, underscoring its inherent diversity. The objective is for the initial solution to be distributed consistently throughout the search domain. The corresponding formula is presented below.

$$x_{n+1} = 4x_n(1 - x_n) \quad x_n \in (0,1) \quad (4)$$

Further, to counteract the uneven distribution of the initializing population within the search space, we utilize Tent mapping for a refined population mapping. This not only ensures uniform population distribution but also articulates the strong interrelation between individuals. Consequently, the exploratory and stochastic traits of chaotic mapping come to the fore, amplifying the algorithm's convergence rate and precision. The respective formula is presented below.

$$x_{n+1} = \begin{cases} x_n/a & x_n \in [0,a) \\ (1 - x_n)/(1 - a) & x_n \in [a.1) \end{cases} \quad (5)$$

In this context, a represents the Tent mapping parameter, set at 0.5.

Lastly, in response to the challenges of confined search space, inadequate uniformity, and diminished diversity during population initialization, we employ a reverse learning approach, superseding the secondary chaotic mapping sequence. This step further bolsters both the diversity and uniformity of the population while introducing an augmented mutation mechanism. The corresponding formula is illustrated below.

$$x_{n+1} = lb + ub - (ub + lb) \times x_n \tag{6}$$

Here, lb denotes the lower bound, while ub signifies the upper bound.

Leveraging the chaotic reverse initialization strategy, this study markedly reinforces the diversity and uniformity of the initial population. This method effectively mitigates the tendency of traditional algorithms to succumb to local optima, leading to improved convergence rates and computational precision.

3.2 Striking a Balance Between Global and Local Search

In conventional flower pollination algorithms, the consistent fixed value of the switch probability p results in a limited adaptability between global and local pollination, hampering the algorithm's versatility and efficiency. To counter this, our study introduces an adaptive switch probability enhancement method for equitably balancing exploration and exploitation [8]. This approach incorporates a temperature parameter. When the overall quality of the population is nearing the global optimum, the temperature is reduced to diminish global search's probability p, favoring a more localized search to pinpoint the optimum solution. On the other hand, when the population quality is sub-optimal, the temperature increases, boosting the probability of a global search p, thereby augmenting the diversity and scope of exploration.

The core steps of the algorithm include.

1. Iteration Verification: Initially, ascertain if it's the maiden iteration. If affirmative, refrain from altering the temperature T due to the absence of a previous optimal fitness benchmark.
2. Temperature Reduction: If the prevailing optimal fitness falls short of its predecessor, the algorithm deduces a proximation to the global optimum and reduces the temperature T by multiplying it with a decay coefficient alpha.
3. Temperature Elevation: Conversely, if the current optimal fitness parallels or exceeds its predecessor, the algorithm identifies potential stagnation at a local optimum. This necessitates an increase in temperature T, achieved by dividing it with the decay coefficient alpha.
4. Switch Probability Computation: Based on the ongoing temperature T and iteration trajectory, determine the global search switch probability p.

$$\mathrm{p} = 0.8 \times \mathrm{T} \times \left(\frac{\mathrm{t}}{\mathrm{N}}\right)^{\frac{1}{\mathrm{t}}} \tag{7}$$

Herein, t symbolizes the iteration count, while N denotes the aggregate iteration tally.

4 Results and Discussion

4.1 Design of the Experiment

The designated environment for this study operated on an Intel Core™ i7-11800H processor, clocked at 2.22GHz, bolstered by 16GB of RAM, under the Windows 11 64-bit OS. MATLAB R2023a was the chosen simulation platform.

The primary objective was to ascertain the effectiveness of the enhanced IFPA strategy. Comparative simulation experiments were executed across five foundational test functions, focusing on the areas of local refinement and expansive global probing. Details pertaining to these test functions and their global optima are tabulated in Table 1. For a holistic perspective, our enhanced flower pollination algorithm was juxtaposed against its traditional counterpart and the particle swarm optimization algorithm. Consistency was maintained by initializing the population size at 50, conducting 30 runs, and setting the iteration limit to 100.

4.2 Analysis of the Results

From Figures 1 through 5, it's evident that the convergence curve of IFPA undergoes distinct iterative shifts. After a short phase of iteration and stabilization, these shifts quickly transition into a fresh stage. Such transitions emphasize the algorithm's capacity to leverage the discrete traits inherent in individual solutions. Impressively, the algorithm converges swiftly after just a few iterations. Not only has its convergence rate been markedly enhanced, but its optimization prowess also notably outstrips that of the FPA and PSO algorithms. The test functions are shown in Table 1.

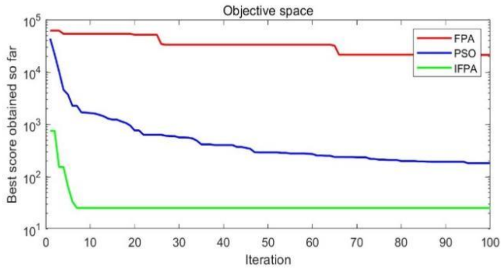

Fig. 1. Test Function 1

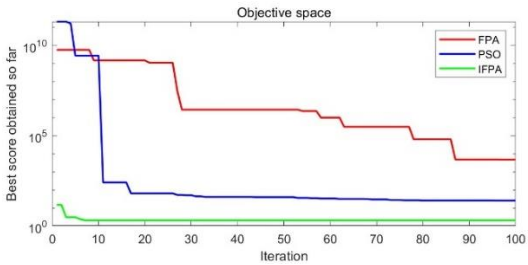

Fig. 2. Test Function 2

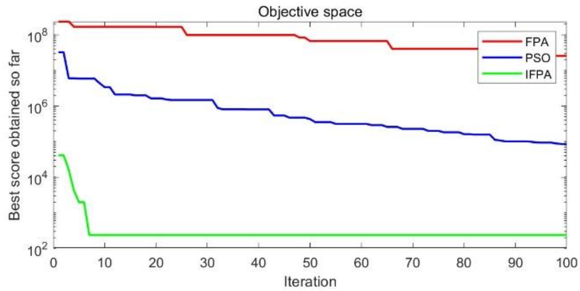

Fig. 3. Test Function 3

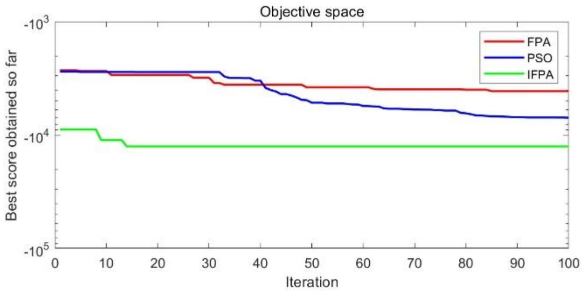

Fig. 4. Test Function 4

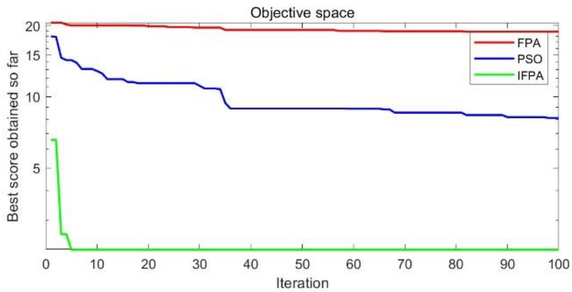

Fig. 5. Test Function 5

Table 1. Test Function

Function	Function Name	Formula
F1	Sum of Squares Function	$f(x) = \sum x_i^2$
F2	Sum and Product of Absolute Values Function	$f(x) = \sum \lvert x_i \rvert + \prod \lvert x_i \rvert$
F3	Rosenbrock Function	$f(x) = \sum (100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2)$

F4	Sum of Negative Sine Transformations Function	$f(x) = \sum -x_i \times \sin\sqrt{\lvert x_i \rvert}$
F5	Ackley Function	$f(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$

5 Stock Price Forecasting using IFPA and bp Neural Networks

5.1 Dataset Preparation

This study employs a dataset obtained from Tonghuashun, which encompasses Guoxin Securities stock data from January to June 2023. The dataset features variables like opening price, peak price, bottom price, closing price, price increase percentage, fluctuation amplitude, and turnover rate. The predictive goal is the following day's closing price.

For model training, the initial 94 data batches were designated as the training set, and the subsequent 24 batches as the test set. Given the disparities in magnitude and numerical ranges of the chosen feature variables, normalization was applied to the dataset before training and testing to mitigate potential model biases.

Linear normalization (Min-Max Scaling) was the chosen method, linearly transforming data values into the [0, 1] range. The corresponding formula is:

$$\hat{S} = \frac{S - \min(S)}{\max(S) - \min(S)} \tag{8}$$

Here, $\hat{S}$ signifies the normalized data, S represents the original data sample, min(S) is the sample's minimum value, and max(S) is its maximum value.

5.2 IFPA-Enhanced Neural Network Prediction Model

This research introduces the IFPA-BP model, which integrates the improved flower pollination algorithm (IFPA) with a BP neural network for predictions [9]. The input layer features 7 neuron nodes, while the output layer has a single node. The hidden layer comprises 15 nodes. The BP neural network has a combined total of 136 parameters for both weights and biases. Seven stock price indicators form the input, and the subsequent day's closing stock price constitutes the output. A logistic sigmoid function (logsig) activates transitions from the input to the hidden layer, while a pure linear function (pureline) is used from the hidden to the output layer. The network's weights and biases are treated as IFPA individuals. The fitness function is defined as the sum of absolute discrepancies between predicted and real values. Through iterative optimization, the ideal individual with the minimal fitness function value emerges. After training on the dataset, the weights and biases of the IFPA-BP structure are finalized for predictions. The objective of this approach is to bolster the model's predictive efficiency and augment stock price forecasting accuracy.

BP Neural Network Prediction Workflow [10].

1. Network Initialization: Set the initial weights and biases.
2. Calculation of Hidden Layer Output: The hidden layer's activation function, the logistic sigmoid function (logsig) is applied.
3. Calculation of Output Layer: A linear transfer function (purelin) is used.
4. Error Calculation.
5. Update Weights & Biases: Following error identification, backpropagation based on this error adjusts the network's weights and biases until reaching either the maximum iteration count or the targeted error, resulting in final weight and bias values.
6. Forecasting: Using the derived weights and biases, the BP neural network makes its predictions.

In this study, the IFPA-BP model's steps are

1. Parameter Initialization - This includes setting the pollen count, their initial positions, and the iteration limit.
2. Each pollen entity is associated with the BP neural weights and biases. The IFPA fitness function uses these, with the actual and estimated values.
3. Apply the IFPA method, using formulas to adjust the pollen's positions.
4. If the end conditions aren't met, revert to Step 2. Otherwise, continue to Step 5.
5. The best entity is mapped to the BP neural weights and biases, followed by neural network training and testing.

5.3 IFPA-Enhanced Neural Network Prediction Model

Within the IFPA framework, we set the total population size to 50 and capped iterations at 100. To assess the predictive performance of our IFPA-BP model, it was juxtaposed against the standard BP neural network. The comparative predictive results are displayed in Figure 6. For a comprehensive evaluation, we employed metrics such as MSE (Mean Square Error), MAPE (Mean Absolute Percentage Error), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error), with the respective error data outlined in Table 2. Our findings underscore that integrating the IFPA methodology effectively fine-tunes the BP neural network parameters, rendering the IFPA-BP model adept at more precise stock price predictions.

Table 2. Comparison of Prediction Errors for Stock Price Among Various Models

Model	MSE	MAPE/%	MAE	RMSE
IFPA-BP	0.018503	1.0833%	0.096467	0.13603
BP	0.10496	2.5411%	0.22303	0.32398

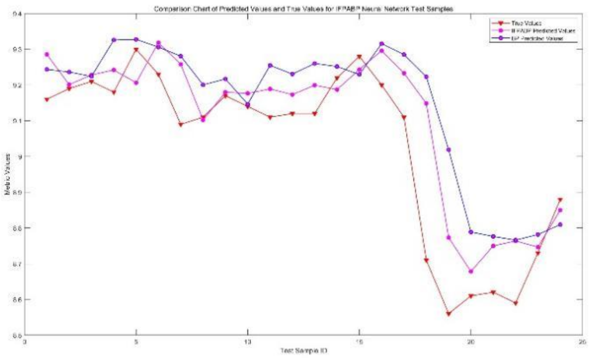

Fig. 6. Comparison Chart of Predicted Values and True Values for IFPABP Neural Network Test Samples

The experimental findings demonstrate that the IFPA algorithm effectively optimizes the parameters of the BP neural network, enabling the constructed IFPA-BP model to yield more precise stock price predictions.

6 Conclusion

This study scrutinized the limitations of the conventional Flower Pollination Algorithm (FPA) and implemented enhancements from both optimization and applicability standpoints. These modifications addressed two distinct aspects of the traditional algorithm. By employing five test functions in conjunction with pertinent data, various experimental results were obtained. Across multiple dimensions of assessment, the algorithm introduced in this study outperforms its traditional counterpart, demonstrating commendable convergence and robustness. Concurrently, an IFPA-BP predictive model was constructed, leveraging the synergistic integration of IFPA with BP neural networks, to forecast the stock prices of Cathay Securities. The empirical outcomes affirm that the devised IFPA-BP model excels in prediction precision, stability, and demonstrates notable generalization capabilities.

In subsequent research endeavors, there is potential for further refinement of the FPA algorithm from different angles, aimed at elevating its performance. Furthermore, its integration with additional machine learning strategies can be explored to address a diverse array of predictive challenges, ultimately seeking to bolster both the accuracy and stability of such forecasts.

Acknowledgment

This research was supported by the Shijiazhuang Municipal Science and Technology Plan Project (Project No: 236240267A), the Key Research and Development Plan

Project of Hebei Province (Project No: 20313701D), and the University-Enterprise Collaboration Project (Project No: A2023028).

References

1. Yang, X. S., "Flower Pollination Algorithm for Global Optimization," J. Springer, (2012). doi:10.1007/978-3-642-32894-7_27.
2. Zhou, Y., Wang, R., and Luo, Q., "Elite opposition-based flower pollination algorithm," Neurocomputing 188, 294-310 (2015). doi:10.1016/j.neucom.2015.01.110.
3. Zhao, C., "Research and Application of Flower Pollination Algorithm," Ph.D. thesis, Guangxi University for Nationalities, [2023-08-15]. doi:CNKI:CDMD:2.1017.074174.
4. Yang, X. S., Karamanoglu, M., and He, X., "Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization," Engineering Optimization 46, 1222-1237 (2014). doi:10.1080/0305215X.2013.832237.
5. Jamil, M., and Yang, X. S., "A Literature Survey of Benchmark Functions For Global Optimization Problems," International Journal of Mathematical Modelling & Numerical Optimisation 4, 150-194 (2013). doi:10.1504/IJMMNO.2013.055204.
6. Ning, J. Q., and He, Q., "Flower Pollination Algorithm with t-Distribution Perturbation Strategy and Mutation Strategy," Journal of Small and Micro Computer Systems, (2021).
7. Zhang, H. Y., Yu, H. Y., and Chen, L. L., "Adaptive Indoor Visible Light Positioning Based on a Reverse Learning Strategy of Flower Pollination Algorithm with Received Signal Strength Indicator," Chinese Journal of Lasers 48(1), 11 (2021). doi:10.3788/CJL202148.0106001.
8. Niu, P. F., Li, J. B., Liu, N., et al., "Boiler NOx Emission Optimization Based on Improved Flower Pollination Algorithm and Extreme Learning Machine," Journal of Power Engineering, 38(10), 15-20 (2018). DOI: CNKI:SUN:DONG.0.2018-10-002.
9. Zhang, L., and Yu, S., "COVID-19 Epidemic Prediction Based on Particle Swarm Optimization BP Neural Network," Software Guide, 21(4), 6 (2022).
10. Qi, H., Hu, H., Bai, Y., et al., "COVID-19 Epidemic Prediction Based on Improved Goose Algorithm Optimized BP Neural Network," Journal of Shanxi University (Natural Science Edition), (2023), pp. 1-10. Available: https://doi.org/10.13451/j.sxu.ns.2023025. Accessed: 2023-08-16.