Beetle swarm optimisation for solving investment portfolio problems

Tingting Chen1, Yongjian Zhu2, Jun Teng3

1Faculty of Science, Jiangsu University, Zhenjiang, Jiangsu, 212013, People’s Republic of China
2School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, 212013, People’s Republic of China
3Institute for Advanced Study, Nanchang University, Nanchang, Jiangxi 330031, People’s Republic of China
E-mail: xiaoduola93@gmail.com

Abstract: A portfolio model is established after analysing the investment environment of the artificial intelligence concept stocks in China. To reduce the risk of investment, the beetle swarm optimisation (BSO) is proposed. BSO, based on the beetle antennae search (BAS) and the standard particle swarm optimisation (PSO), is derived from the standard PSO but the update rules of each particle originate from BAS. In global searching, BSO, making the model get a lower value at risk, is more capable than standard PSO, which is easily trapped in local optimal defects. This study tries to solve portfolio model by using BSO algorithm. The results prove that BSO can do better in dealing with optimisation problems of constrained multi-dimensional functions.

1 Introduction

Artificial intelligence (AI) studies and develops theoretical methods for simulating and extending human intelligence. Due to the great success of deep learning, AI has recently attracted great attention from the scientific and industrial community. It is speculated that the combination of AI and traditional industries will trigger the fourth industrial revolution. The popularity of AI has effectively driven the capital market. Furthermore, the AI industry has become a hot spot for investment in recent years. The AI concept stocks of A-share, with a remarkable performance, are far more than the stock index gains. In 2017, the total financing amount of global AI investment reached US$8.9 billion. Financial research focuses on how to allocate capital effectively and efficiently in the market. How to optimise capital allocation in a complex system full of uncertainty is the issue of the investment portfolio [1]. The mean-variance model proposed by Markowitz in 1952 marked the advent of portfolio theory. However, for the investment portfolio problem with constraints, it is often difficult to find an effective solution using traditional numerical optimisation solutions.

Nowadays, the rapid development of AI has been very helpful in dealing with this kind of problems [2–7]. The application of the particle swarm algorithm is a good example [8–10]. However, the biggest drawback of the particle swarm algorithm is that it is easily trapped in the local optimal solution during the search process. To rectify this defect, combining beetle antennae search (BAS) with particle swarm optimisation (PSO) together, the beetle swarm optimisation (BSO) algorithm is established in this paper to solve the AI concept stock portfolio problem [11–13].

2 Model establishment

2.1 Introduction

A portfolio is the set of assets that people choose to invest in. For most investors, choosing the satisfactory optimal portfolio is a sophisticated process, which requires one predicts precisely the tendency of the assets under consideration and defines what the optimal is. To put it simply, the optimal portfolio is the one that maximises the return while minimising the fluctuation, especially a portfolio with the lowest fluctuation while having the highest return.

2.2 Establishment of a portfolio model

Suppose that the total investment is $S$; the proportion vector of investment of $n$ stocks held by the entire portfolio, $X = (x_1, x_2, \ldots, x_n)$; the proportion of investment in risk-free assets (assumed as national debt), $x_0$; the variance-covariance matrix of the $n$ AI concept stocks to be invested, $\Sigma$. The buying price per share of the $i$th stock is $p_i (i = 1, 2, \ldots, n)$, the rate of return of the $i$th stock is $y_i (i = 1, 2, \ldots, n)$. So, the objective function of Markowitz’s mean-variance model is (where the total risk of the entire portfolio is $\sigma$)

$$\min \sigma = X \Sigma X$$

In China’s stock market, the minimum trading unit of the stock is a single lot, and the number of trading volume is an integer. Suppose the turnover of the $i$th stock is $n_i$, with the restrictions $n_i \in I$, $I$ is a non-negative integer, $i = 1, 2, \ldots, n$.

Because China’s stock market does not allow short selling behaviour, funds can only be invested in $n$ types of stocks, and the remaining funds can be invested in risk-free assets. At the same time, transaction costs, including taxes and transaction fees, are important considerations in the financial market and necessary in trading. Take the Shanghai Stock Exchange as an example, transaction costs consist of the following expenses [15]:

(1) The entrustment fee, 5 yuan/time.
(2) Commissions. After the transaction is completed, the investor pays the broker a fee of 3% of the transaction amount, starting at five yuan.
(3) Transfer fee. After the stock is traded, the cost of changing the account name is paid. The transfer fee for the transaction is 0.3% of the number of shares traded. There is no minimum charge requirement.

Then all transaction costs of risk securities are

$$C_i(n_i) = k_0 + k_1(n_i) + k_2(n_i)$$

where
The transaction costs are $C_n$, $C(n_i)$ when the transaction costs for risk-free assets and transaction costs for the $i$th stock trading lots are $n$. Entrustment fee $k_i = 5$, when the $i$th stoke lots is $n_i$, the trading commission is $k(n_i)$. Commissions have a cost coefficient of $\mu = 0.3\%$, the transfer fee is $k(n_i)$, the cost coefficient is $\mu = 0.3\%$ per stoke.

In addition, the risk-free investment is a one-year National debt, the annual interest rate is $\gamma = 1.95\%$, and the transaction cost coefficient is $\mu = 0.1\%$. The transaction fee for national debt is

$$C_i = \mu x S$$

(3)

Based on it, the total transaction costs of the portfolio can be as follows:

$$C = C_i + \sum_{i=1}^{n} C_i(n_i)$$

(4)

The expected rate of return at a time $T$ is $Y_0$ (the higher the expected rate of return, the greater the risk) when the price of the $i$th stock is $p_i$, and the rate of return is $y_i$. Therefore, the constraint condition is

$$\sum_{i=1}^{n} 100n_i p_i + y x S - C \geq Y_0$$

(5)

Hence, we can finally get the portfolio model as

$$\min \ V = X^T \sigma X$$

(6)

$$\sum_{i=1}^{n} 100n_i p_i + y x S - C \geq Y_0$$

$$\sum_{i=1}^{n} 100n_i p_i + x S \leq S$$

$$x_i = \frac{100n_i p_i}{S} (i = 1, 2, \ldots, n)$$

$$\sum_{i=1}^{n} X_i = 1$$

$$x_i \geq 0 (i = 0, 1, \ldots, n)$$

$$n_i \in I, \text{ I} = \{\text{Isnot} - \text{negative}(i = 1, 2, \ldots, n)\}$$

2.3 PSO algorithm principle

The PSO algorithm, based on the social behaviour metaphor, was put forward in 1995 by Eberhart and Kennedy [12]. It originated from the research on predatory behaviour of flock. The basic concept is individuals in a group share the information so that the whole group's movement will evolve from disorder to order in the problem-solving process and, ultimately, obtain the optimal solution to the problem. The PSO algorithm includes some parameters that have great impact on the algorithm performance, often stated as the exploration–exploitation tradeoff: exploration is the ability to test various regions and, hopefully, find a global optimum in the problem space.

The PSO algorithm simulates a bird in a flock by designing a massless particle. The particle has two properties: velocity $v$ and position $x$, velocity represents the speed of movement, and position represents the direction of movement. Each particle individually searches for the optimal solution in the search space and records it as the extremum of the current individual $p_{best}$. Then, the individual extremum is shared with other particles in the entire particle group, and the optimal individual extremum is defined as the current global optimal solution of the entire particle swarm $G_{best}$. All particles in a particle swarm, based on their current individual extremum $p_{best}$ and the current global optimal solution $G_{best}$, share their own velocity and position variables [16–18]. The flowchart of the PSO is shown in Fig. 1.

2.4 BAS algorithm principle

The BAS algorithm, based on the foraging principle of the beetle, is a new technology to find the optimal solution proposed in 2017. When foraging, the beetle, not knowing the specific location of the food, uses two antennae to detect the odour of food and decides the direction of itself. Specifically, if the scent received by the left side of beetle is stronger than the right one, then, the beetle moves to the left, otherwise it moves to the right. Based on this simple principle, it can easily find food. Like the PSO, instead of knowing the specific formula, BAS uses an iterative approach to approximate the optimal solution step by step. In the BAS algorithm (see Fig. 2), there is only one individual.

The modelling steps are as follows:

1. The beetle head is assumed to be heading randomly toward any direction, so the direction of the vector from the right antenna to the left one must also be random. Accordingly, for an optimisation problem in an n-dimensional space, a random vector can be generated to represent and normalise it

$$b = \frac{\text{rands}(k, 1)}{\text{rands}(k, 1)}$$

(7)

where $k$ is spatial dimension and rand() is random function.

2. Thus, the relationship between the left and right antennae can be obtained as follows:

$$x_i - x_i = d_i \cdot \text{dir}$$

(8)

Moreover, $x_0, x_i$ can be represented by the centroid
where $x_l$ represents the left side of the search area, $x_r$ represents the right side.

(3) To determine the odour intensity of left and right antennae, substitute the left and right positions for $f(x_l)$ and $f(x_r)$, where $f(x)$ is the fitness function.

(4) In order to develop guidelines for the detection behaviour, this paper further generates the following iterative model, which detects odours by considering the search behaviour and iteratively updates the position of beetle

$$x^{t+1} = x^t - \delta \cdot b_{sign} \left( f(x^t_l) - f(x^t_r) \right)$$  \hspace{1cm} (10)

$x'$ represents the centroid coordinates at the $t$th iteration of the beetle, $x_l$ represents left antenna coordinate at the $t$th iteration, $x_r$ represents right antenna coordinate at the $t$th iteration. The step size at the $t$th iteration is $\delta$, $\text{sign}(x)$ represents symbolic functions.

### 2.5 Establishment of BSO algorithm principle

#### 2.5.1 Principle of BSO: As described in the previous two sections, BAS algorithm is aimed at individuals only and does not consider the connections between groups. PSO focuses on the impact of the population on a single particle, ignoring the particle's own judgment in the search process. Therefore, this paper proposes to integrate BAS and PSO models together, introducing the concept of BSO. Each particle in the PSO is characterised as a beetle and searches. The BSO constructed by this method can well overcome the problems of poor stability, tendency to fall into local optimum and others caused by the PSO algorithm [19].

The process of initial position and velocity of the beetle swarm is the same as that of the standard PSO. However, in the iterative process, the way of updating the position of the beetle swarm is no longer only relying on the historical best solution and the current global optimal solution of individual beetle, but adding the idea of Beetle Antennae Search. Individuals in the BSO will compare the fitness function values of their left and right sides during each iteration and compare the better values of the two, which can also be used to update the position of the beetle swarm. The updated formula for beetle swarm position can be expressed as follows [20]:

$$v_b^t = - \delta \cdot b \cdot \text{sign}(f(x_l^t) - f(x_r^t))$$  \hspace{1cm} (11)

$$v_{b_k}^{t+1} = v_{b_k}^{t+1} + c_1 \cdot \text{rand} \cdot \left( P_{b_k}^t - x_k^t \right)$$

$$+ c_2 \cdot \text{rand} \cdot \left( P_{g_k}^t - x_k^t \right) + c_3 \cdot \text{rand} \cdot v_{b_k}^t$$

$$x_k^{t+1} = x_k^t + v_{b_k}^{t+1}$$  \hspace{1cm} (12)

where $v_{b_k}^{t+1}$ represents the speed of the $k$th particle after the $t$th iteration, $x_k^{t+1}$ indicates the position after the $t$th iteration, $v_{b_k}$ represents the update rate generated by the BSO, and $c_1$ is the learning factor, $\text{sign}()$ is symbolic functions.

#### 2.5.2 Process of algorithm: The detailed steps are represented as follows:

1. Initialise the algorithm parameters, set the size of the PSO size $N$, the learning factors $c_1$, $c_2$, $c_3$ and the inertia weight $W$, as well as the distance $d_k$ between the two antennae of each beetle.
2. Randomly initialise the position $x$ and velocity $v$, calculate the fitness of each position, use the current position as the individual optimal solution $P_{best}$ and finally get the current global optimum $G_{best}$ by comparison.
3. Enter the iteration:

- Randomise the heads of the beetles dir. Calculate the left and right positions of each beetle $x_{left}$ and $x_{right}$ according to the position of dir and the beetle, and use this to calculate the fitness of each beetle’s left position $f_{left}$ and that of each beetle’s right position $f_{right}$. By comparing the two, the speed update rule generated by the left and right fitness of each beetle in the population is obtained

$$v_{b_k} = \delta \cdot b \cdot \text{sign}(f(x_{left}) - f(x_{right}))$$  \hspace{1cm} (13)

- By comparing the fitness of the current position of each beetle with $P_{best}$ and $G_{best}$, the speed update rule is generated by the current individual optimal solution and the global optimal solution.

- Combine the above two speed update rules to get the current update rules for speed of each antenna

$$v_{k}^{t+1} = v_{k}^{t+1} + c_1 \cdot \text{rand} \cdot \left( P_{b_k}^t - x_k^t \right)$$

$$+ c_2 \cdot \text{rand} \cdot \left( P_{g_k}^t - x_k^t \right) + c_3 \cdot \text{rand} \cdot v_{b_k}^t$$

$$x_k^{t+1} = x_k^t + v_{b_k}^{t+1}$$  \hspace{1cm} (14)

Current location update rules

- Updated learning factors and inertia weights $c_1$, $c_2$, $c_3$, $w$, updated individual optimal solutions and global optimal solutions $P_{best}$, $G_{best}$.

4. After completing the iteration, the global optimal solution $G_{best}$ and $f(G_{best})$ corresponding to the optimal solution position are obtained. The description is represented in Table 1.

#### 2.5.3 Empirical analysis: To test whether the BSO algorithm is more superior for solving the portfolio problem, this paper compares the BSO algorithm with the standard PSO algorithm and contrasts the value-at-risk obtained by the two in solving the portfolio and the number of iterations needed to achieve the convergence.
3 Experiment results and discussion

3.1 Materials and methodologies

3.1.1 Data source and index selection: To test the results of AI investment portfolio model by the BSO, the paper selects nine stocks from the AI concept stocks and the weekly rates of return from 23 June 2017 to 18 August 2017 (9 weeks). The calculated covariance matrix measures the risk of the nine stocks. Table 2 shows the covariance matrix. The yield and stock quotes select the weekly yield and price approximating the average of the 9-week return and the average price, which are shown in Table 3. The data used comes from the ‘Wind’ database.

At the same time, one-year national debt is used as risk-free assets (a yield of 1.95%) on the assumption that a total of 1 million assets can be used for investment and no securities have been purchased before investment [21, 22].

3.2 Parameter settings

In this paper, the trading lot number vector is used as the position variable of the particle and the beetle. PSO and BSO algorithms set the number of particles and beetles, a number of iterations set by the particle swarm and number of iterations set for the beetle swarm to 40, 8000 and 4000, respectively. The learning factors of PSO algorithm are set to \(c_1 = 0.5, c_2 = 2\); the learning factors used by BSO algorithm, \(c_1 = 0.5, c_2 = 2, c_3 = 3\); both inertia weights are the same, taking \(w = 0.9\). Both the learning factors are time-varying. Penalty method is adopted to deal with constraint conditions [23] (see Table 4).

3.3 Results and discussion

To verify the effectiveness and the algorithm's performance, based on the weekly returns and prices of the nine stocks, PSO and BSO are used, respectively, setting the expected portfolio returns to 0.06, 0.08 and 0.10, respectively. The two algorithms solve the investment proportion of risk-free investment and stock investment in each case, as well as the value-at-risk under this ratio respectively. All the experiments are carried out with MATLAB 9.2.0.538062 (R2017a) running on a PC with Intel (R) Core (TM) i7-7700 CPU 3.60 GHz and 8.00 GB RAM on Win 10 operation system.

### Table 1 Descriptions of BSO

| Algorithm | BSO algorithm |
|-----------|---------------|
| Input     | Initialise the algorithm parameters, get the initial individual optimal solution and the global optimal solution \(P_{best}, G_{best}\) |
| Iteration | While(End condition) |
|           | { 1. Calculate the position of each beetle, the fitness() of left and right antennae. 2. According to the calculated fitness, produce the updated rules for the speed and position of each beetle. 3. Update the velocity \(v\) and position \(x\) of each beetle. 4. Update the learning factor, inertia weight, current individual optimal solution \(P_{best}\) and the global optimal solution \(G_{best}\). } |
| Output    | The global optimal solution position \(G_{best}\), fitness function \(f(G_{best})\) |

### Table 2 Covariance matrix of selected stocks

| No | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     |
|----|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1  | 10.8  | 0.8   | 3.9   | 2.9   | -4.2  | 3.1   | -1.0  | 3.9   | -0.5  |
| 2  | 0.8   | 1.5   | 0.9   | -0.2  | -2.6  | 0.8   | -1.3  | 0.9   | 0.4   |
| 3  | 3.9   | 0.9   | 2.2   | 0.9   | -2.2  | 1.9   | -1.1  | 2.2   | 0.1   |
| 4  | 2.9   | -0.2  | 0.9   | 2.0   | 0.2   | 0.9   | 0.0   | 0.9   | -0.2  |
| 5  | -4.2  | -2.6  | -2.2  | 0.2   | 10.4  | -0.8  | -1.9  | -2.2  | -1.2  |
| 6  | 3.1   | 0.8   | 1.9   | 0.9   | -0.8  | 2.0   | -1.4  | 1.9   | 0.2   |
| 7  | -1.0  | -1.3  | -1.1  | 0.0   | -1.9  | -1.4  | 13.2  | -1.1  | -0.1  |
| 8  | 3.9   | 0.9   | 2.2   | 0.9   | -2.2  | 1.9   | -1.1  | 2.2   | 0.1   |
| 9  | -0.5  | 0.4   | 0.1   | -0.2  | 0.2   | -0.1  | 0.1   | 2.4   |

### Table 3 Weekly returns and prices of selected stock

| Stoke                  | Return | Price |
|------------------------|--------|-------|
| Iflytek Co. Ltd        | 0.0190 | 19.62 |
| SHE: 002230            | —      | —     |
| Dawning Information    | 0.1900 | 24.02 |
| SHA: 603019            | —      | —     |
| Inspur Software        | 0.0358 | 25.18 |
| SHA: 600756            | —      | —     |
| Siasun Robot & Auto.   | 0.0343 | 12.95 |
| SHE: 300024            | —      | —     |
| Insignia Technology    | 0.0466 | 24.56 |
| SHA: 600797            | —      | —     |
| Hangzhou Hikvision     | 0.2288 | 26.66 |
| SHE: 002415            | —      | —     |
| Beijing Kunlun Tech    | 0.0188 | 17.25 |
| SHE: 300418            | —      | —     |
| Inspur Electronic Info. | 0.0422 | 21.50 |
| SHE: 000977            | —      | —     |
| CSG Smart Sci & Tech   | 0.0679 | 23.47 |
| SHE: 300222            | —      | —     |
From the results in Table 5, it can be noticed that, for different expected return rates, the value-at-risk calculated by BSO is lower than that of PSO. In another word, BSO can obtain relatively better results at the same expected rate of return. This shows that the BSO algorithm has a stronger global search ability, and the algorithm is easier to find the global optimal solution. On the other hand, when the expected rate of return increases, the risk values obtained by both algorithms increase, and the proportion of risk-free investment drops significantly, and that of high-yield stocks rises significantly. It shows that the investment portfolio model constructed in this paper is in line with the actual market conditions.

Figs. 3 and 4 show the convergence of the two algorithms. From the perspective of the convergence of the algorithm, PSO algorithm converges around 3000 iterations, and BSO algorithm only 1500 iterations approximately to converge. It shows that, compared with the standard PSO, after the concept of the beetle group is introduced, each particle's updating method no longer only relies on its own historical optimal solution and the group's global optimal solution. Instead, the particle's own independent judgment is added during each iteration, which makes the particle's iterative method more flexible and more intelligent, making the number of iterations needed for convergence less, and the result better.

3.4 Summary

In this paper, we introduced the concept of beetle swarm in the particle swarm algorithm and proposed the BSO. BSO is used to solve the investment portfolio problem of today's AI concept stocks. After empirical analysis, BSO shows greater global searching ability than that of standard PSO and it is easier to find the global optimal solution when solving multi-dimensional constraints problems. In addition, in terms of algorithm convergence, by introducing the concept of BSO, the particle has its own judgment of the environment space in each iteration, so that the number of iterations required for the beetle to converge is less.

4 References

[1] Cohn, R.A., Lewellen, W.G., Lease, R.C., et al.: 'Individual investor risk aversion and investment portfolio composition', J. Financ., 1975, 30 (2), pp. 605-620

[2] Khairalla, M., Ning, X., Nashat, A.J.: 'Modelling and optimisation of effective hybridisation model for time-series data forecasting', J. Eng., 2018, 2, pp. 117-122
[3] Zhou, G., Li, Y., He, Y.C., et al.: ‘Artificial fish swarm based power allocation algorithm for MIMO-OFDM relay underwater acoustic communication’, IET Commun., 2018, 12, (9), pp. 1079–1085

[4] Labar, C., Garone, E., Kinnairt, M.: ‘Sub-optimal extremum seeking control for static maps’, IET Control Theory Appl., 2018, 12, pp. 745–752

[5] Voosen, K.: ‘Machine vision algorithms that learn’, Comput. Control Eng., 2004, 15, (5), pp. 30–31

[6] Jin, Z., Hou, Z., Yu, W., et al.: ‘Target tracking approach via quantum genetic algorithms’, IET Comput. Vis., 2017, 12, pp. 241–251

[7] Syahputra, R., Wiyagi, R.O., Suripto, S., et al.: ‘A novel fuzzy approach for multi-objective optimization of distribution network configuration in complex system’, Int. J. Appl. Eng. Res., 2018, 13, (2), pp. 1120–1127

[8] Chin, Y.H., Hsieh, Y.Z., Su, M.C., et al.: ‘Music emotion recognition using PSO-based fuzzy hyper-rectangular composite neural networks’, IET Signal Process., 2017, 11, (7), pp. 884–891

[9] Al-Saud, M.S.: ‘PSO of power cable performance in complex surroundings’, IET Gener. Transm. Distrib., 2018, 12, pp. 2452–2461

[10] Syahputra, R., Soesanti, I., Ashari, M.: ‘Performance enhancement of distribution network with DG integration using modified PSO algorithm’, J. Electr. Syst., 2016, 12, (1), pp. 1–19

[11] Jiang, X., Li, S.: ‘BAS: beetle antennae search algorithm for optimization problems’, Available at https://arxiv.org/pdf/1710.10724.pdf, accessed 30 October 2017

[12] Eberhart, R., Kennedy, J.: ‘A new optimizer using particle swarm theory’, Micro Mach. Human Sci., 1995, 95, pp. 39–43

[13] Solomonoff, R.J.: ‘The search for artificial intelligence’, Electron. Power, 1968, 14, (1), p. 8

[14] Markowicz, H.: ‘Portfolio selection’, J. Finance, 1952, 7, (1), pp. 77–91

[15] Ge, Y., He, Y., Hu, X., et al.: ‘Analysis method and empirical research on economic benefit of large-scale consumptive power grid investment’, J. Eng., 2017, 13, pp. 1285–1289

[16] Liu, J., Li, J., Wu, J., et al.: ‘Global MPPT algorithm with coordinated control of PSO and INC for roof top PV array’, J. Eng., 2017, 13, pp. 778–782

[17] Wu, H., Dong, P.: ‘PSO method for distribution network reconfiguration considering the stochastic uncertainty of wind turbine, photovoltaic and load’, J. Eng., 2017, 13, pp. 2032–2036

[18] Mostaghim, S., Teich, J.: ‘Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)’. Swarm Intelligence Symp., Indianapolis, USA, 2003, vol. 3, pp. 26–33

[19] Jiang, X., Li, S.: ‘Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization’. Available at https://arxiv.org/pdf/1711.02395.pdf, accessed 7 November 2017

[20] Garg, H.: ‘A hybrid PSO-GA algorithm for constrained optimization problems’, Appl. Math. Comput., 2016, 274, pp. 292–305

[21] Xia, X., Liu, J., Li, Y.: ‘Particle swarm optimization algorithm with reverse-learning and local-learning behavior’, J. Software, 2014, 9, (2), pp. 350–357

[22] Xu, S., Cui, J., Wang, H.: ‘Modified particle swarm optimization algorithms based on topology and particle mutation’, Control Decis., 2018, 33, (4), pp. 1–11, (in Chinese)

[23] Viswanathan, J., Grossmann, I.E.: ‘A combined penalty function and outer-approximation method for MINLP optimization’, Comput. Chem. Eng., 1990, 14, (7), pp. 769–782