Portfolio Optimization with Investment Constrains Based on Modified Cuckoo Search Algorithm

Xiangli Kong
School of Economics and Management, Xiamen University of Technology, Xiamen 361024, China

Abstract
Faced with complex and volatile securities market, investors constantly seek various investment strategies to maximize investment profit and minimize the risks. Classic portfolio theory believes that multi-stage diversified investment is the most optimized investment and risk should be avoided by taking diversified investment portfolio. In 1952, the U.S. famous economist Markowitz proposed a Mean-Variance Mode. It used the variance of yields to measure investment risks and promoted the development of the modern investment theory. But Markowitz’s mean-variance model put returns on assets as symmetric distribution and it regarded the part of returns that were higher than mean returns as risks. This is obviously not complied with the reality. Some scholars proposed robust portfolio optimization with value-at-risk-adjusted (Deng and Geng, et al., 2013) and portfolio optimization with mean-absolute deviation model in Malaysia stock market, it think the model to minimize the portfolio risk at certain rate of return in portfolio optimization(Lam and Lam, 2015), Portfolio optimization using a credibility mean-absolute semi-deviation model (Enriqueta and José, 2015), other scholars proposed that Random fuzzy mean-absolute deviation models with hybrid uncertainty (Zhang, 2016) and using asymmetry robust mean absolute deviation model (Li and Han, Xia, 2016). However, these models established by these studies are still NP-complete problems, and the traditional mathematical programming methods cannot find the effective solution of NP-complete problem. Currently, genetic algorithms that have been used to solve portfolio problems include particle swarm optimization, ant colony algorithm, neural network algorithm, simulated annealing algorithm and other intelligent algorithms. Some presented the algorithm based on the particle swarm optimization and improved SVDD for the personal credit rating(Pang and Li, 2014), other put forward Hybrid discriminant neural networks for bankruptcy prediction and risk scoring (Azayite and Achchab, 2016) and modal parameters estimation using ant colony optimization algorithm (Piotr Sitarz, Bartosz Powałka, 2016). Research shows that, these genetic algorithms have advantages such as strong robustness and global searching ability, but they also have a tendency to converge towards local optima or lead to premature convergence. Ant colony algorithm (ACO) has strong robustness and global searching ability, but it has slow convergence and is prone to generate stagnation. Particle swarm optimization (PSO) has fast convergence, but in later iterations, its searching ability is not strong enough and there are problems such as early convergence. As for neural networks algorithm, it has strong self-organizing and self-adapting ability. What’s more, it has strong fault tolerance and can do parallel processing on a large scale. However, it does not converge into global optimum. Simulated annealing has strong ability to find optimum but the optimization process is too long, so its efficiency is low.

Cuckoo search algorithm (CS) is a new type of optimization algorithm developed by Yang from Cambridge and other people in 2009. It is inspired by two strategies, including Nest parasitism of some cuckoo species and Levy flight. The bird uses random-walk style search to get an optimal nest of other host birds, in order to lay its own eggs. This is very efficient in seeking optimal solutions (Yang and Deb, 2010). Compared with particle swarm optimization, ant colony optimization, neural network algorithm, simulated annealing algorithm and other algorithms, Cuckoo search algorithm has comprehensive advantages in terms of parameters, global searching ability, generality and robustness. Besides, in Cuckoo search, cuckoo birds perform highly random
levy flights. However, during the process, they do not communicate with and study from each other. It is unfavorable for the algorithm to evolve and explore more efficient searching paths. In later period, the algorithm converges slowly with barely satisfied accuracy. Some scholars have proposed some methods to improve Cuckoo Search algorithm. Zhang Y, Valian E and other scholars improved the algorithm by including variables in the algorithm: probability of the eggs being abandoned $P_a$ and the step size control factor $\alpha$. The variables of $P_a$ and $\alpha$ are dynamically changed with the number of generations. This self-adapting method improves the speed of convergence and its accuracy (Zhang, Wang and Wu, 2012; Valian, Tavakoli and Mohanna, 2013). In the CS algorithm, some have added global optimization and self-adapting strategies and come up with modified cuckoo search (MCS). MCS algorithm has more powerful global search ability, and higher convergence accuracy. It is also applied to solve the optimum design problem of multiple-effect evaporation system (Jiang and Ruan, 2014). Hamming distance used to update the current position of bird nests and improved the CS algorithm (Sun and Zhang, 2015). Having studied the theories above, this paper takes many constraints into consideration, such as the transaction costs and trading units and on short selling and long selling. Then the paper establishes a mean-semi absolute deviation asset portfolio model with investment constraints; it includes local searching strategies, which significantly improve local search ability, and it also takes self-adaptive strategy, which allows step size factors to adapt along with the growing algorithm processes; it also adopts the learning strategy that strengthens information exchanges between cuckoo individuals. CS algorithm is modified and proposes the MCS (Modified Cuckoo Search, MCS). MCS is applied to solve portfolio optimization problems.

2. MEAN-SEMI ABSOLUTE DEVIATION RISK MODEL FOR THE STOCK-BOND INVESTMENT PORTFOLIO

This paper supposes the investor’s overall investment is $I$ and he chooses $n$ types of stocks, $m$ types of national bonds and one risk-free asset as investment. The stocks and bonds without considering investment lots can be marked as:

$$x = \left( x_{a1}, x_{a2}, \cdots, x_{an}, y_{b1}, y_{b2}, \cdots, y_{bn} \right)$$

$p_{ai}$ is price of stock $i$ for every lot. $p_{bj}$ is the price of $j$ for every lot. For exchange-traded securities, a lot represents the minimum quantity of the trading securities, so the lots of stock $i$ is $x_i = \left\lfloor \frac{x_i \cdot I}{p_{ai}} \right\rfloor$, where $\left\lfloor \cdot \right\rfloor$ is integer-valued function. Therefore, the real investment ratio of the stock $i$ is $\left\lfloor \frac{x_i \cdot I}{p_{ai}} \right\rfloor \cdot \frac{p_{ai}}{I}$. Similarly, the lots of stock number $j$ is $y_j = \left\lfloor \frac{y_j \cdot I}{p_{bj}} \right\rfloor$, and the real investment ratio of the stock is $\left\lfloor \frac{y_j \cdot I}{p_{bj}} \right\rfloor \cdot \frac{p_{bj}}{I}$.

Therefore, the semi-absolute deviation risk function of the stock-bond portfolio is:

$$\omega = \frac{1}{T} \sum_{t=1}^{T} \min \left\{ 0, \sum_{j=1}^{n} (a_{it} - a_t) \frac{p_{ai} x_i}{I} + \sum_{j=1}^{m} (b_{jt} - b_t) \frac{p_{bj} y_j}{I} \right\}$$

Where, $a_{it}$ stands for the historical earnings rate of the stock $i$ at the period of $t$; $b_{jt}$ stands for the historical earnings rate of the bond $j$ at the period of $t$ ($t = 1, 2, \cdots, T$); $a_t$ is the expected yield of stock $i$, $b_j$ is the expected yield of bond $j$, which does not have the risks of default or non-redeemption; $x_{ai}$ is the investment ratio of the stock $i$ ($i = 1, 2, \cdots, n$); $y_{bj}$ is the investment ratio of the bond $j$ ($j = 1, 2, \cdots, m$).

Besides, $a_t$, the expected yields of the stock $i$, can be counted based on historic records of period $T$, which is $a_t = \sum_{i=1}^{n} \frac{1}{T} a_{it}$.

Under normal circumstances, the expected yield of interest rates at the period of $T + 1$ equals to its original yield. For bonds that do not have risk of defaults or redemption risks $j (j = 1, 2, \cdots, m)$, their expected yields $b_j$ equal to its original yields, that is $b_j = b_{j1}$. $b_{j1}$ is the first period’s yield of $j$. The transaction cost is an important factor. Hereby, we have $c_i$ as the transaction costs of the stock $i$. $c_j$ is the transaction cost of the stock $j$. $k_a$ is the transaction ratio of the stock. $k_b$ is the transaction ratio of the bond. $X = (x_{a1}, x_{a2}, \cdots, x_{an}, y_{b1}, y_{b2}, \cdots, y_{bn})$
is the bond-stock portfolio’s original investment lots. (where \(x_i^0\) represents the original investment lots of the stock \(i\), and \(y_j^0\) represents the original investment lots of the bond \(j\)). Therefore, the transaction costs of the securities exchange are as follows:

\[
c_i = k_a \cdot p_{a_i} \cdot |x_i - x_i^0| \\
\overline{c}_i = k_b \cdot p_{b_i} \cdot |y_j - y_j^0|
\]

China’s securities market stipulates that buying long or selling short is forbidden and the minimum trading unit is one lot. Therefore, the remaining capital of investment on stocks and bonds is invested in risk-free assets. In this paper, \(r_o\) is the yield of risk-free assets and \(\rho\) is the minimum rate of return. As the acquired stock-bond portfolio must have the ratio of minimum return to investment as one, the constraints are as follows:

\[
\sum_{i=1}^{n} \left( a_i \cdot \frac{p_{a_i}}{I} \cdot c_i \right) + \sum_{j=1}^{m} \left( b_j \cdot \frac{p_{b_j}}{I} \cdot \overline{c}_j \right) + \left[ I - \sum_{i=1}^{n} (x_i + c_i) - \sum_{j=1}^{m} (y_j + \overline{c}_j) \right] \frac{y}{I} \geq \rho;
\]

\[
\sum_{i=1}^{n} x_i + \sum_{j=1}^{m} y_j = 1, x_i \geq 0, y_j \geq 0;
\]

\[
x_i = \left[ \frac{x_i I}{p_{a_i}} \right],
\]

\[
y_j = \left[ \frac{y_j I}{p_{b_j}} \right],
\]

\[
a_i = \sum_{i=1}^{n} \frac{1}{I} a_i;
\]

\[
b_j = b_j;
\]

\(t = 1, 2, \ldots, T; \)

\(i = 1, 2, \ldots, n; \)

\(j = 1, 2, \ldots, m; \)

Therefore, the stock-bond portfolio’s mean-semi absolute deviation risk model is:

\[
\min_{\omega} \frac{1}{T} \sum_{t=1}^{T} \left[ \sum_{i=1}^{n} (a_i - a) \frac{p_{a_i}}{I} + \sum_{j=1}^{m} (b_j - b) \frac{p_{b_j}}{I} \right]
\]

\(s.t.
\]

\[
\sum_{i=1}^{n} \left( a_i \cdot \frac{p_{a_i}}{I} \cdot c_i \right) + \sum_{j=1}^{m} \left( b_j \cdot \frac{p_{b_j}}{I} \cdot \overline{c}_j \right) + \left[ I - \sum_{i=1}^{n} (x_i + c_i) - \sum_{j=1}^{m} (y_j + \overline{c}_j) \right] \frac{y}{I} \geq \rho;
\]

\[
\sum_{i=1}^{n} x_i + \sum_{j=1}^{m} y_j = 1, x_i \geq 0, y_j \geq 0;
\]

\[
x_i = \left[ \frac{x_i I}{p_{a_i}} \right],
\]

\[
y_j = \left[ \frac{y_j I}{p_{b_j}} \right],
\]

\[
a_i = \sum_{i=1}^{n} \frac{1}{I} a_i;
\]

\[
b_j = b_j;
\]

\(t = 1, 2, \ldots, T; \)

\(i = 1, 2, \ldots, n; \)

\(j = 1, 2, \ldots, m; \)

\(1\)

3. CUCKOO SEARCH

CS is based on three idealized rules:

(1) Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;

(2) The best nests with high quality of eggs will carry over to the next generation;

(3) The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability \(P_{a_x} = [0,1]\)

Based on these three rules, the basic steps and location of the Cuckoo searching for hosts following the pattern of Levy Flights can be summarized as the equation shown Formula (2).

\[
x_{i}^{k+1} = x_{i}^{k} + \alpha_{g} \oplus \text{Levy}_{y} (\lambda)
\]

(2)
which few optimization problems that have negative minimum values:

fitness value of the generation

optimization problems, and by improving the CS algorithm’s simulation results, we come up with an adaptive
adjust the searching step length, by changing the step
solutions. To enable the searching step size and allow it to adapt itself along with the growing algorit
adapt itself to changes of the algorithm. This significantly lowers the algorithm’s ability to find the optimal

4
appropriate.

to

algorithm. For example, we have introd
To improve the efficiency and accuracy of optimization

1
probably remains weak. Its convergence is not accurate, and there are other problems such as slow convergence.

1

-1


4. MODIFIED CUCKOO SEARCH ALGORITHM (MCS)

4.1. Local Search Strategy

CS algorithm has powerful global searching ability and optimum searching paths, but its local searching
ability remains weak. Its convergence is not accurate, and there are other problems such as slow convergence.
To improve the efficiency and accuracy of optimization, we introduce a strategy of local searching in the CS
algorithm. For example, we have introduced a random number \( \xi \) to enable cuckoo searching near the location of

\( x^{\text{best},i}_{\text{best},l} \) with high-accuracy searching in small areas. When CS finds better location than \( x^{k}_{\text{best},l} \), and we name
it \( x^{k}_{\text{best},l} \), we use \( x^{k}_{\text{best},l} \) to replace \( x^{k-1}_{\text{best},l} \), and then the searching process ends up as

\[
x_{\text{best},l}^{k+1} = x_{\text{best},l}^{k} + \xi
\]

\( x_{\text{best},l}^{k+1} \) is the neighborhood solution of \( x_{\text{best},l}^{k} \). \( \xi \) is the correction, and in \( \xi \in [-\delta, \delta] \), \( \delta \) changes according

to \( x_{\text{best},l}^{k} \), which is

\[
\delta = \eta \times x_{\text{best},l}^{k}
\]

\( \eta \) is coefficient, and \( \eta \in (0,1) \). Based on numerous simulated experiments, we believe that \( \eta = 0.1 \) is more
appropriate.

4.2. Self-adapting Strategy

In the CS algorithm, Levy flight decides the step size randomly. However, the searching step size cannot
adapt itself to changes of the algorithm. This significantly lowers the algorithm’s ability to find the optimal
solutions. To enable the searching step size and allow it to adapt itself along with the growing algorithm we
adjust the searching step length, by changing the step-size control factor. We select several minimum-value
optimization problems, and by improving the CS algorithm’s simulation results, we come up with an adaptive
strategy for \( \alpha_{l} \), the step size control factor.

\[
\alpha_{l} = \begin{cases} 
\left( \frac{h_{\min,1}}{h_{\max,1}} \right)^{\beta}, & h_{\max,1} > h_{\min,1} \\
\left( \frac{h_{\max,1}}{h_{\min,1}} \right)^{\beta}, & h_{\max,1} < h_{\min,1}
\end{cases}
\]

\( h_{\max,1} \) is the worst fitness value in the first generation of algorithm for the original group. \( h_{\min,1} \) is the best
fitness value of the generation \( k \), and \( \beta \) is the exponent. For most optimization problems that have non-
negative minimum values: \( h_{\max,1} > h_{\min,1} \), and the value of \( h_{\max,1} \) decreases as algorithm algebra \( k \) increases. For
few optimization problems that have negative minimum values, we have \( h_{\max,1} < h_{\min,1} \), \( |h_{\max,1}| < |h_{\min,1}| \), during
which \( |h_{\min,1}| \) increases as the algorithm algebra increases. For rare cases that have negative minimum values, its
fitness value will grow from a small negative number into a negative optimal value; where $h_{\text{max},j} < h_{\text{min},k} < 0$, $\|h_{\text{min},j}\| > \|h_{\text{max},k}\|$. The value of $\|h_{\text{min},j}\|$ decreases as $k$ increases. $\beta$ is a positive exponent that adjusts the changes of $\alpha_j$. Usually, $\beta = 0.2$.

4.3. Learning Strategy

In evolution, individuals pay great attention to mutual learning. However, during the searching process of cuckoos, individuals lack efficient information exchanges and the opportunities to learn from each other. It is often used as a performance test problem for optimization algorithms.

4.4. Basic steps of MCS

Step1: Initialization of the nest population, including the number of nests, the initial location of nests and algorithm parameters;

Step2: Calculating the fitness value of nest locations, and find the optimal location of the nest (i.e. investment lots) and highest fitness value (i.e. minimum objective function and investment risks), worst nest location and its fitness value.

Step3: The cuckoo with the worst adaptability learns from cuckoo with the best adaptability, based on the formula (7). In this way, it moves closer to the current optimal nest location.

Step4: By updating the nest location through the equation (2), (3) and (6) of Lery Flights, we have a new set of locations on the birds’ nests.

Step5: Calculating the corresponding fitness value of each nest location, and updating the historical optimal location of the nest after comparison.

Step6: By comparing the random number $r \in [0,1]$, which follows a uniform distribution, with $P \alpha_j$, we retain the nest location with smallest probability of being found. Meanwhile, we randomly change the nest location that has a large probability of being found. In this way, we work out a new nest location.

Step7: Calculating the corresponding fitness value of each new nest location, in order to determine the new optimal nest location, the worst nest location and its fitness value;

Step8: Based on equation (4) and (5), the cuckoos search carefully near the current optimal nest location and obtain the optimal nest location. After comparison, we update the historical optimal location of the nests;

Step9: Determining the termination conditions of the algorithm. If satisfied, results will be obtained. Otherwise, Step 3—Step 9 will be repeated.

5. SIMULATION EXPERIMENT AND EMPIRICAL ANALYSIS

5.1. Simulation Experiment

Simulation Experiment uses the software Matlab 2011b and the experiment is performed on the platform of Windows7( Intercore i7, 2.8GHz, 4.00GB).

It adopts Rastrigin function and Sphere function to test and compare the performance of CS algorithm and MCS algorithm. Rastrigin function is $f(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10)$. It is a multi-dimensional, multi-model function that has local minima. It is often used as a performance test problem for optimization algorithms; Sphere function $f(x) = \sum_{i=1}^{D} x_i^2$ is continuous, convex and unimodal, often being used to test the accuracy of convergence in an algorithm. The two functionss domain is $x \in [-5.12,5.12]$. The parameters of CS and MCS algorithms take the number of nests as $n = 25$ . Step-size control factor is $\alpha = 1$; elimination probability is $P \alpha_j = 0.25$; and the convergence accuracy is $\varepsilon \leq 10^{-7}$. The maximum searching algebra is 15000. The accuracy of convergence has a domain of $e = |f - f_{\text{opt}}|$, where $f$ is the fitness value when the function ends. $f_{\text{opt}}$ is the optimal fitness value. For different testing functions, we allow both CS and MCS algorithm to operate independently for 50 times. We record the algorithm’s mean convergence $\bar{T}$, standard deviation $s$, successful convergence rate $\eta$, operation
time $t$ and the highest searching convergence accuracy, as the algorithms evolve into their best form. Results of the experiments are shown in table 2 and 3. The optimal fitness values of the 2 testing functions change with the evolution, and the algebra changing curves are shown in Figure 1 and Figure 2.

**Table 1.** Convergence rate comparison of CS and MCS

<table>
<thead>
<tr>
<th>Functions (dimension)</th>
<th>CS $k$ $\pm s$</th>
<th>$\eta$</th>
<th>$\overline{t}$</th>
<th>MCS $k$ $\pm s$</th>
<th>$\eta$</th>
<th>$\overline{t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rastrigin(l=10)</td>
<td>4102$\pm$354</td>
<td>100%</td>
<td>4.582</td>
<td>1538$\pm$248</td>
<td>100%</td>
<td>2.421</td>
</tr>
<tr>
<td>Sphere(l=100)</td>
<td>6078$\pm$267</td>
<td>100%</td>
<td>10.08</td>
<td>2775$\pm$83</td>
<td>100%</td>
<td>5.385</td>
</tr>
</tbody>
</table>

**Table 2.** Convergence accuracy comparison of CS and MCS

<table>
<thead>
<tr>
<th>Functions (dimension)</th>
<th>Maximum Search algebra</th>
<th>CS accuracy</th>
<th>$\eta$</th>
<th>MCS accuracy</th>
<th>$\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rastrigin(l=10)</td>
<td>3500</td>
<td>3.0592</td>
<td>0</td>
<td>1.9928E-12</td>
<td>100%</td>
</tr>
<tr>
<td>Sphere(l=100)</td>
<td>6000</td>
<td>3.8514E-03</td>
<td>0</td>
<td>1.4962E-19</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Figure 1.** Searching curves for Rastrigin function

**Figure 2.** Searching curves for Sphere function

From the results of the simulation experiments, we can see that, in Rastrigin function and Sphere function, both algorithms’ convergence rates are 100%. It shows that MCS algorithm inherits the powerful global searching ability of CS algorithm. As we can see from the results of two testing functions, MCS algorithm has lower mean convergence algebra, lower standard deviation, and shorter mean operation time than those of CS. (which can be seen in table 1). MCS meets the requirement on convergence accuracy of $\varepsilon \leq 10^{-7}$ and its convergence rates are 100%. In comparison, CS does not meet the requirement and its convergence rates are 0 (which can be see in table 2). From Figure 1 and 2, we can infer that, for the two testing functions, the results of MCS in the evolution process are obviously better than CS algorithm. Therefore, MCS inherits the powerful global searching ability and has faster convergence, higher convergence accuracy and better robustness.

5.2. Empirical Analysis

We use the data of the real securities market to verify the effectiveness of the model and the improved algorithm. The data of this paper is from Dazhihui software (2010.12.1 ~ 2016.12.1), and we treat the monthly average yield as the historic yield. We regard the closing price on December the first as the price of stocks and bonds. We randomly select 6 types of stocks from the A-shares market in Mainland China, (including China
Merchants Bank Co., Ltd 600036, SAIC Motor Corporation Limited 600104, TBEA Co., Ltd 600089, Xiamen C&D Inc. (600153), Xinjiang Western Animal Husbandry Co Ltd 300106, and Jiangxi Copper Co Ltd (600362) and two bonds (21 government bonds 010107 and 02-government bonds 010213) to form investment portfolio.

We suppose that an investor has 1 million yuan for investment. The initial investment lot is 0, and we take banks’ 3-month fixed term deposit rates as the risk-free yield, \( r_0 = 0.0025 \). We suppose the commission rate is \( k_i = 0.003 \), the stamp tax \( k_t = 0.001 \), and the scrip fee is \( k_s = 0.1 \). Based on the data, we are able to count the expected yields and risk of these 6 stocks and 2 bonds. (As what can be seen in Table 3).

### Table 3. The expected return rate and investment risk of 6 stock and 2 bonds

<table>
<thead>
<tr>
<th>Stock code</th>
<th>Bond code</th>
<th>Expected return rate</th>
<th>Investment risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>600036</td>
<td>010107</td>
<td>0.2085</td>
<td>0.0702</td>
</tr>
<tr>
<td>600104</td>
<td>010213</td>
<td>0.1783</td>
<td>0.1087</td>
</tr>
<tr>
<td>600089</td>
<td></td>
<td>0.1202</td>
<td>0.1442</td>
</tr>
<tr>
<td>6000153</td>
<td></td>
<td>0.2155</td>
<td>0.3157</td>
</tr>
<tr>
<td>300106</td>
<td></td>
<td>0.0605</td>
<td>0.3803</td>
</tr>
<tr>
<td>600362</td>
<td></td>
<td>0.0965</td>
<td>0.2876</td>
</tr>
<tr>
<td>010107</td>
<td></td>
<td>0.0426</td>
<td>0.0029</td>
</tr>
<tr>
<td>010213</td>
<td></td>
<td>0.0026</td>
<td>0.0054</td>
</tr>
</tbody>
</table>

We use MCS algorithm to verify the effectiveness of stock-bond portfolio investment and mean-semi absolute deviation risk model. The number of iteration is 6000. We observe that when the minimum return rate is 0.002, 0.004, 0.006, 0.008, 0.010 respectively, the 6 stocks and 2 bonds have minimum investment risk best investment lots and real yields. (as can be seen in the table 4). As for MCS algorithm, we use the optimal location of the bird nest as investment lots and the optimal fitness value (i.e. minimum objective function value) to represent investment risk.

### Table 4. The different return rate of investment risk and investment hand

<table>
<thead>
<tr>
<th>Return rate</th>
<th>Investment risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.002</td>
<td>0.00306</td>
</tr>
<tr>
<td>0.004</td>
<td>0.00378</td>
</tr>
<tr>
<td>0.006</td>
<td>0.00512</td>
</tr>
<tr>
<td>0.008</td>
<td>0.00689</td>
</tr>
<tr>
<td>0.010</td>
<td>0.00853</td>
</tr>
</tbody>
</table>

Based on Table 3 and 4, we conclude that Xinjiang Western Animal Husbandry Co Ltd (300106) is the most risky investment with fairly low yields, so we do not suggest investing this stock; when the required rate of return is 0.002, Jiangxi Copper Co Ltd (600362) has low expected rate of return and high risk. The 02-government bond (010213) has lowest risk but its expected yield is also low. So this paper does not suggest investing the two securities above. when the required rate of return is 0.004, Xiamen C&D Inc. (600153) has highest expected yields but is too risky to invest. If the required ROA is 0.006, 21-government bond (010107) has low risk but its expected yield is not high enough for investment; No choice, when the required ROA is 0.008, TBEA Co., Ltd (600089) has not expected risk but the yield is not ideal, So do not choose investment. If the required ROA is 0.010, the expected yields of both China Merchants Bank Co., Ltd (600036) and SAIC Motor Corporation Limited (600104 are higher and their investment risk is relatively low. So we could opt to increase investment lots of China Merchants Bank Co., Ltd (600036) and SAIC Motor Corporation Limited (600104).

### 6. CONCLUSIONS

This paper applies Markowitz model into various risk investment situations. It fully considers Chinese securities markets’ constraints on buying long and selling short, transaction costs and trading units. It also establishes mean-semi absolute deviation risk model for stock-bond investment portfolio. It improves CS algorithm by drawing on advantages of local searching strategy, self-adaptive mechanism, intelligent learning strategies and other strategies. Finally, the Modified Cuckoo Searching (MCS) comes into being. MCS greatly improves the overall performance of CS algorithm and is successfully applied to solve problems during the optimization of asset portfolio that has investment constraints. Through simulation experiment and empirical analysis, this paper verifies the accuracy and effectiveness of mean-semi absolute deviation risk model.

### ACKNOWLEDGEMENTS

This work was supported by 71502154 (The National Natural Science Foundation of China).

...
REFERENCES


