Artificial Bee Colony Algorithm for Portfolio Optimization Problems

Zhen Wang*, Sanyang Liu, Xiangyu Kong
*Corresponding Author *Department of Mathematics, Xidian University, Xi'an, Shaanxi 710071, China, jen715@163.com*
Department of Mathematics, Xidian University, Xi'an, Shaanxi 710071, China, liusanyang@126.com
Department of Mathematics, Xidian University, Xi'an, Shaanxi 710071, China, kxywz08@163.com

Abstract

In this paper, a cardinality constrained mean-variance model is introduced for the portfolio optimization problems. This model is a mixed quadratic and integer programming problem for which efficient algorithms do not exist. The use of heuristic algorithms in this case is necessary. Some studies have investigated the cardinality constrained mean-variance model using heuristic algorithm. But almost none of these studies deal with artificial bee colony algorithm. The purpose of this paper is to use artificial bee colony algorithm to solve this model. The experimental results show that the proposed algorithm performs well for the portfolio optimization problem.

Keywords*: Artificial Bee Colony, Swarm Intelligence, Portfolio Optimization, Efficient Frontier*

1. Introduction

Portfolio optimization, which addresses the ideal assignment of assets, has been one of the important problems in modern financial mathematics. A fundamental research on this problem was given by Markowitz (see [1] and [2]), who proposed the mean-variance model. Although the problem proposed by Markowitz looks simple, there is more than one way to establish an optimum portfolio. Indeed, Markowitz himself suggested that a model based on semi-variance would be preferable. In [3], a mean absolute deviation portfolio selection model was proposed. It has been shown that this model has the similar results to the mean-variance model. Simaan [4] contented that the computational savings from the use of mean absolute deviation object function are outweighed by the loss of information from the covariance matrix. Samuelson [5] was the one who first noticed the asymmetry of returns, and considered the third order moment in portfolio optimization. Then, other authors did some researches on this problem, see [6-8].

On the other hand, the Markowitz mean-variance model has been expanded by introducing additional constraints, such as the no short selling constraint, transaction costs constraint and cardinality constraints. In this paper, we employ a modified mean-variance model called a "cardinality constrained mean-variance model", which is presented in [9]. In this model, we consider the cardinality constraints, which impose a limit on the number of assets in the portfolio and the quantity constraints which restrict the proportion of each asset in the portfolio to lie between lower and upper bounds.

Because of the cardinality constraints, the traditional algorithms are not available. There are some heuristic methods of solving the cardinality constrained mean-variance model, see [9-13]. These methods consist of genetic algorithms [9], tabu search [9], simulated annealing [9], neural networks [10] and particle swarm optimization [11].

In this study, we employ another heuristic method, which is artificial bee colony (ABC) algorithm (see [14-18]) to solve this problem. The ABC algorithm, introduced recently by Karaboga [14], is a swarm intelligence based algorithm which simulates the foraging behavior of a bee colony. From the previous research, it is clear that the ABC algorithm outperforms other heuristic methods for the high dimensional and multimodal problems. The portfolio optimization problems are always high in dimension. Furthermore, there are few studies on ABC algorithm in the literature, and almost none of them deals with portfolio optimization. Our experimental results show that the ABC algorithm is available for the cardinality constrained portfolio optimization problem.

The remainder of the paper is organized as follows. Section 2 presents the cardinality constrained mean-variance model for portfolio optimization problem. In Section 3, how to use ABC to solve the cardinality constrained mean-variance model is proposed. Section 4 gives some computational results. Some conclusions are given in the last section.

2. Problem Formulation

In this section, the cardinality constrained mean-variance (CCMV) model is introduced, which is derived from the well-known mean-variance (MV) model. The MV model for portfolio optimization problem is:

$$\begin{array}{lll} \min & \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij}\right] - (1-\lambda)\left[\sum_{i=1}^{N} x_i \mu_i\right] & \\ \text{subject to} & \sum_{i=1}^{N} x_i = 1, & \quad (1) \\ & 0 \le x_i \le 1, \qquad i = 1,\dots,N, & \end{array}$$

where N is the number of available assets, μ_i is the expected return of the i th asset ($i = 1,\dots,N$), σ_{ij} is the covariance between assets i and j ($i = 1,\dots,N;\ j = 1,\dots,N$), $\lambda \in [0,1]$ is the risk aversion parameter and x_i is the proportion held of the i th asset. When $\lambda = 0$, it means maximizing the portfolio mean return only and the optimal solution will be formed by the asset with the greatest mean return. When $\lambda = 1$, it means minimizing the total variance associated to the portfolio only. For each $\lambda \in [0,1]$, there is a best trade-off between mean return and risk for problem (1) which consists of the efficient frontier, which was called a standard efficient frontier by Fernandez and Gomez in [10].

In the CCMV model, two additional constraints have to be included in the MV model. The first one is the number of assets in the portfolio. The second one is the lower and the upper bounds of an included asset's proportion. Let K be the desired number of assets in the portfolio, ε_i and δ_i represent the lower and upper bounds of the i th ($i = 1,\dots,N$) asset, respectively, and if $z_i = 1$ asset i will be included in the portfolio. Then, the CCMV model is:

$$\begin{array}{lll} \min & \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij}\right] - (1-\lambda)\left[\sum_{i=1}^{N} x_i \mu_i\right] & \\ \text{subject to} & \sum_{i=1}^{N} x_i = 1, & \\ & \sum_{i=1}^{N} z_i = K, & \quad (2) \\ & \varepsilon_i z_i \le x_i \le \delta_i z_i, \quad i = 1,\dots,N, & \\ & z_i \in \{0,1\}, i = 1,\dots,N. & \end{array}$$

3. ABC Algorithm for the CCMV Model

It is clear that problem (2) is a mixed quadratic and integer programming problem for which efficient algorithms do not exist [10]. Therefore, we introduce the artificial bee colony (ABC) algorithm for solving the CCMV model.

The ABC algorithm, proposed by Karaboga, is a recently introduced optimization algorithm which simulates the foraging behavior of a bee colony [14]. The swarm in ABC consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony consists of employed bees, and the other half includes onlooker bees. Employed bees are responsible for exploiting the food source and giving the information to the onlooker bees in the hive. Onlooker bees wait in the hive and decide on a food source to exploit based on the information shared by the employed bees. The employed bee whose food source has been abandoned becomes a scout. In the ABC, the position of a food source represents a possible solution to the optimization problem and each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of solutions in the population. The basic steps of the ABC algorithm are given below.

Initialize.
Repeat:
Place the employed bees on the food source on the memory;
Place the onlooker bees on the food source on the memory;
Send the scouts to the search area for discovering new food source.
until requirements are satisfied.

3.1. Constraint satisfaction

Consider that there are N dimensions for each bee and each dimension represents an asset. Indeed, each bee includes proportional variables denoted by $x_{ij}(i=1,...,SN;\, j=1,...,N)$ and decision variables denoted by $z_{ij}(i=1,...,SN;\, j=1,...,N)$, where SN is the number of employed bees. Thus, the number of dimensions that a bee owns is $2\times N$. It is clear that bees represent candidate solutions, and each bee must be feasible and satisfy the constraints in (2). Inspired by the similar approach proposed in [11], the arrangement algorithm is presented to guarantee the feasibility of the solutions shown as follows.

Algorithm 1: Arrange (i **)**

i is the current solution and consist of:

Q : the set of K_i^* distinct assets in the current solution;
z_{ij} : the current decision value for asset j;
x_{ij} : the current proportion value for asset j.

While($K_i^* < K$)
 If random (0,1)<0.5 **then** j =randomly select an asset such that $j \notin Q$
 Else j =select the maximum c-valued asset such that $j \notin Q$
 End if
 $z_{ij}=1, Q=Q\cup[j], K_i^*=K_i^*+1$
End while
While($K_i^* > K$)
 If random (0,1)<0.5 **then** j =randomly select an asset such that $j \in Q$
 Else j =select the minimum c-valued asset such that $j \in Q$
 End if
 $z_{ij}=0, Q=Q-[j], K_i^*=K_i^*-1$
End while
While(true)
 $\chi=\sum_{j\in Q}x_{ij}, x_{ij}=x_{ij}/\chi, \eta=\sum_{j\in Q}\max(0, x_{ij}-\delta_j), \phi=\sum_{j\in Q}\max(0,\varepsilon_j-x_{ij})$
 If $\eta=0$ and $\phi=0$ **then** Exit arrangement algorithm
 For $j=1$ to N
  If $z_{ij}=1$ **then**
   If $x_{ij}>\delta_j$ **then** $x_{ij}=\delta_j$
   End if
   If $x_{ij}<\varepsilon_j$ **then** $x_{ij}=\varepsilon_j$
   End if
  End if
 End for
End while

Assume that Q is the set of assets held by bee i, and bee i has distinct K_i^* assets where $K_i^* = \sum_{j=1}^{N} z_{ij}$. If $K_i^* < K$, some assets must be added to Q; If $K_i^* > K$, some assets must be removed from Q until $K_i^* = K$. First, we consider the case of $K_i^* < K$. If a random number between 0 and 1 is less than 0.5, we select an asset which is not in Q and add it to Q. Otherwise, we select the maximum c-valued asset which is not in Q and add it to Q. The c-value gives the proportion between mean return and mean risk with respect to aversion parameter, which can be calculated as follows:

$$c_i = \frac{\theta_i + \theta}{\rho_i + \rho}, \qquad i = 1,..,N, \tag{3}$$

where $\theta_i = 1 + (1+\lambda)\mu_i$, $\theta = -\min(0, \theta_1, ..., \theta_N)$, $\rho_i = 1 + \lambda \sum_{j=1}^{N} \sigma_{ij} \big/ N$, $\rho = -\min(0, \rho_1, ..., \rho_N)$. When $K_i^* > K$, we select an asset in Q then remove it or select the minimum c-valued asset in Q then remove it with equal probabilities.

For the constraint "$\sum_{i=1}^{N} x_i = 1$", we set $\chi = \sum_{j \in Q} x_{ij}$ and let $x_{ij} = x_{ij} \big/ \chi$ for all $j \in Q$. Then, this constraint will be satisfied. Due to the constraint $\varepsilon_i z_i \le x_i \le \delta_i z_i$, $i = 1,..., N$, if $x_{ij} > \delta_j$ then $x_{ij} = \delta_j$; if $x_{ij} < \varepsilon_j$ then $x_{ij} = \varepsilon_j$.

3.2. Fitness function

In this study, we use the fitness function proposed in [16], i.e. ,

$$fitness_i = \begin{cases} 1/(1+f_i), & if\ f_i \ge 0, \\ 1 + abs(f_i), & if\ f_i < 0, \end{cases} \tag{4}$$

where f_i is the cost value of the solution represented as

$$f_i = \lambda \left[\sum_{j=1}^{N} \sum_{k=1}^{N} z_{ij} x_{ij} z_{ik} x_{ik} \sigma_{jk} \right] - (1-\lambda) \left[\sum_{j=1}^{N} z_{ij} x_{ij} \mu_j \right]. \tag{5}$$

3.3. Producing initial food source sites

Initial food source sites are produced randomly within the range of the boundaries of the parameters, which is

$$z_{ij} = round\left(\frac{1}{1 + e^{-z_j^{\min} + rand(0,1)(z_j^{\max} - z_j^{\min})}} - 0.06 \right), \tag{6}$$

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}), \tag{7}$$

In this phase, counters which store the numbers of trials of solutions are reset to 0. After initialization, the population of the food sources (solutions) is subjected to repeat cycles of the search process of the employed bees, the onlooker bees and the scout bees. A terminal criterion is reaching a maximum cycle number (MCN).

3.4. Sending employed bees to the food source

We have mentioned that an employed bee is associated with only one food source. Thus, the number of food sources is equal to the number of employed bees. An employed bee finds a neighboring food source and evaluates its quality. In ABC algorithm, this finding a neighboring food source is defined as

$$\tilde{z}_{ij} = round\left(\frac{1}{1+e^{-z_{ij}+\phi_{ij}(z_{ij}-z_{lj})}} - 0.06\right), \tag{8}$$

$$\tilde{x}_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{lj}), & if\ \tilde{z}_{ij} = 1, \\ x_{ij}, & otherwise, \end{cases} \tag{9}$$

where j is a random integer in $\{1, 2, ..., N\}$, $l \in \{1, 2, ..., SN\}$ is a randomly chosen index that has to be different to i, and ϕ_{ij} is a uniformly distributed real random number in the range [-1,1].

A greedy selection is applied between (z_i, x_i) and $(\tilde{z}_i, \tilde{x}_i)$; then the better one is selected depending on the value of cost function calculated by (5). If source at $(\tilde{z}_i, \tilde{x}_i)$ is superior to that of (z_i, x_i) in terms of profitability, the employed bee memorizes the new position and forgets the old one, otherwise the previous position is kept in memory. If (z_i, x_i) cannot be improved, its counter holding the number of trails is incremented by 1, otherwise the counter is reset to 0.

3.5. Selecting the food source by onlooker bees

After all employed bees complete their searches, they share their information of their sources with the onlooker bees. The onlooker bee evaluates the information and chooses a food source with a probability. In ABC algorithm, roulette wheel selection scheme is employed,

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}. \tag{10}$$

If the probability value p_i is greater than a random real number within the range [0,1], then the onlooker bee produces a modification on the position of this food source by using (8) and (9). Therefore, greedy selection is applied and the onlooker bee either memorizes the new position by replacing the old one or keeps the old one. If (z_i, x_i) cannot be improved, its counter holding the number of trails is incremented by 1, otherwise the counter is reset to 0.

3.6. Abandonment criteria

After all employed bees and onlooker bees complete their search, if a position cannot be improved further through a "limit", then that food source is assumed to be abandoned. If more than one counter exceeds the "limit" value, one of the maximum ones might be chosen. The abandoned food source can be replaced by a new one using (6) and (7).

All these units and interactions between them we have discussed until now are shown in ABC algorithm as follows.

Algorithm 2: ABC algorithm for portfolio optimization

While ($\lambda \le 1$ **)**

Initialize the population of solution z_{ij}, x_{ij}, $i = 1, ..., SN$, $j = 1, ..., N$ by using (6) and (7), $trail_i = 0$, $trail_i$ is the improvement number of the solution x_i used for abandonment.

Arrange (i **)** $i = 1, ..., N$.

Compute f_i $i = 1, ..., N$.

For $counter = 1$ to $\lfloor 1000N / SN \rfloor$

{-- -- Produce a new food source population for employed bees -- --}
For $i = 1$ to SN

$\tilde{z}_{ij} = round\left(\frac{1}{1+e^{-z_{ij}+\phi_{ij}(z_{ij}-z_{lj})}} - 0.06\right)$,

If $\tilde{z}_{ij} = 1$ **then** Produce a new food source $\tilde{x}_{ij}$ for the employed bee by using (9)
End if
Arrange (i **)**
Compute $\tilde{f}_i$
If $\tilde{f}_i < f_i$ **then** $z_{ij} = \tilde{z}_{ij}, x_{ij} = \tilde{x}_{ij}$
Else $trail_i = trail_i + 1$
End if
End for
Calculate the probability values p_i by (10) for the solutions using fitness values
{-- -- Produce a new food source population for onlooker bees -- --}
$t = 0, i = 1$
Repeat

If $random(0,1) < p_i$ **then** $\tilde{z}_{ij} = round\left(\frac{1}{1+e^{-z_{ij}+\phi_{ij}(z_{ij}-z_{lj})}} - 0.06\right)$,

If $\tilde{z}_{ij} = 1$ **then** Produce a new food source $\tilde{x}_{ij}$ for the employed bee by using (9)
End if
Arrange (i **)**
Compute $\tilde{f}_i$
If $\tilde{f}_i < f_i$ **then** $z_{ij} = \tilde{z}_{ij}, x_{ij} = \tilde{x}_{ij}$
Else $trail_i = trail_i + 1$
End if
$t = t + 1$
End if
Until ($t = SN$ **)**
{-- -- Determine Scout -- --}
If $\max(trail_i) > limit$ **then** Generate a new randomly solution by replacing the old one
End if
Memorize the best solution achieved so far
End for
$\lambda = \lambda + \Delta\lambda$
End while

4. Computational Results

In this section, the computational results of ABC algorithm for solving the CCMV model are presented. Test data are obtained from http://people.brunel.ac.uk/mastjjb/jeb/orlib/portinfo.html. These data correspond to different stock market indices: the Hong Kong HangSeng with 31 assets, the German Dax 100 with 85 assets, the British FTSE 100 with 89 assets, the US S&P 100 with 98 assets, and the Japanese Nikkei with 225 assets.

All the results were computed using the values $K = 10$, $\varepsilon_j = 0.01$ and $\delta_j = 1$, $(j = 1,..., N)$ for the problem formulation, and $\Delta\lambda = 0.02$ for the implementation of the algorithm. The number of different λ, denoted by ξ, is 51.

Inspire by the analysis in [11], we use mean Euclidian distance, variance of return error and mean return error for comparison of the standard efficient frontier and the corresponding heuristic efficient frontier. The standard efficient frontier is calculated taking 2000 different λ values by solving problem (1).

Table 1. Results for the cardinality constrained efficient frontier

Index		**GA**	**TS**	**SA**	**PSO**	**ABC**
HangSeng	Mean Euclidian distance	0.0040	0.0040	0.0040	0.0049	8.19e-005
	Variance of return error (%)	1.6441	1.6578	1.6628	2.2421	1.4805
	Mean return error (%)	0.6072	0.6107	0.6238	0.7427	0.5372
DAX100	Mean Euclidian distance	0.0076	0.0082	0.0078	0.0090	1.59e-004
	Variance of return error (%)	1.6441	1.6578	1.6628	2.2421	8.139
	Mean return error (%)	0.6072	0.6107	0.6238	0.7427	1.3511
FTSE100	Mean Euclidian distance	0.0020	0.0021	0.0021	0.0022	1.59e-004
	Variance of return error (%)	2.8660	4.0123	3.8205	3.0596	2.7629
	Mean return error (%)	0.3277	0.3298	0.3304	0.3640	0.3114
S&P100	Mean Euclidian distance	0.0041	0.0041	0.0041	0.0052	8.37e-005
	Variance of return error (%)	3.4802	5.7139	5.4247	3.9136	3.2092
	Mean return error (%)	1.2258	0.7125	0.8416	1.4040	0.6844
Nikkei	Mean Euclidian distance	0.0093	0.0010	0.0010	0.0019	0.6844e-005
	Variance of return error (%)	1.2056	1.2431	1.2017	2.4274	3.8087
	Mean return error (%)	5.3266	0.4207	0.4126	0.7997	0.9802

The results of above three measures for ABC algorithm are compared with which for the other four heuristic algorithms, as shown in Table 1. The other four heuristic algorithms are genetic algorithms (GA), tabu search (TS), simulated annealing (SA) and particle swarm optimization (PSO), and the results for these four heuristic algorithm are obtained from [11].

It is clear from Table 1 that the mean Euclidian distance of ABC algorithm is smaller than the other four algorithms for these five market index. For HangSeng, FTSE 100 and S&P 100, the ABC algorithm for solving the CCMV model is the best able to approximate the standard efficient frontier with the smallest variance of return error and mean return error.

Figure 1 shows the comparison of standard efficient frontiers and ABC efficient frontier. It is clear that the ABC efficient frontiers almost coincide with the standard efficient frontier for HangSeng, FTES100, S&P100 and Nikkei. For DAX 100, the coincidence of the efficient frontier happens with a low risk of investment.

The experimental results show that the ABC algorithm performs well for solving the CCMV model for portfolio optimization problem and gives better solutions than other heuristic algorithms. Economically, the ABC algorithm gives better portfolio strategies than other heuristic methods for an investor, especially for the risk aversion investors.

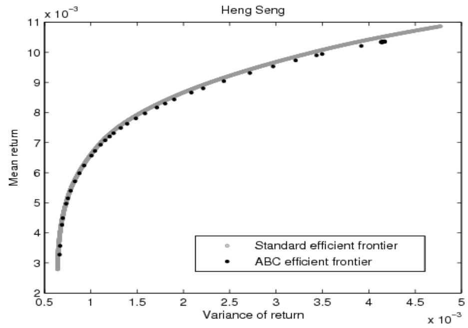
5. Conclusion

In this work, we have focused on solving the cardinality constrained portfolio optimization problem. This problem is a mixed quadratic and integer programming problem for which no computational efficient algorithms are known. We have used the ABC algorithm to solve the cardinality constrained portfolio optimization problem and traced out its efficient frontier. The results obtained have been compared to those obtained by four other heuristic algorithms, which are genetic algorithms, tabu search, simulated annealing and particle swarm optimization.

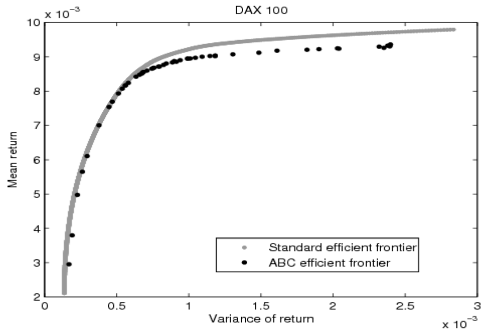
The experimental results have shown that the ABC algorithm obtained a set of solutions with higher quality than the solutions from the other four heuristic algorithms. In a word, the ABC algorithm presented in this paper is feasible for solving the cardinality constrained portfolio optimization problem.

6. Acknowledgements

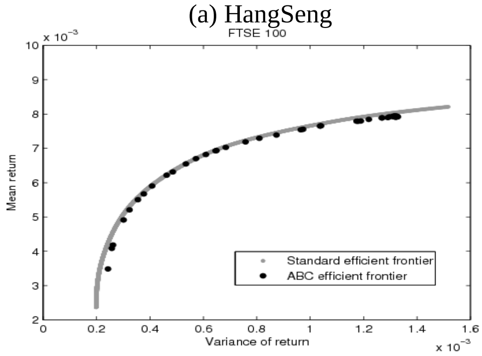
The research described in this paper was supported by National Natural Science Foundation Project of China (No.60974082).



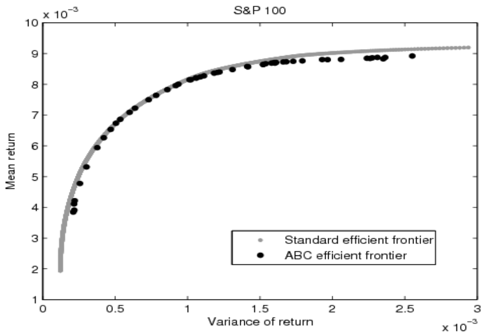
(a) HangSeng



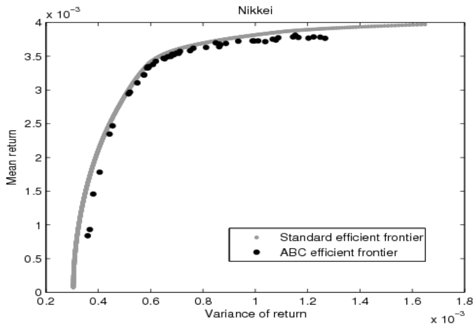
(b) Dax100



(c) FEST100



(d) S&P100



(e) Nekkei

Figure 1. Efficient frontier for five indices

7. References

[1] Harry Markowitz, “Portfolio selection”, The Journal of Finance, vol. 7, no. 1, pp. 77-91, 1952.

[2] Harry Markowitz, “Portfolio selection: Efficient diversification of investments”, John Wiley & Sons, New York, 1959.

[3] Hiroshi Konno, Hiroaki Yamazaki, “Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market”, Management Science, vol. 37, no. 5, pp. 519-531, 1991.

[4] Yusif Simaan, “Estimation risk in portfolio selection: the mean variance model versus the mean absolute deviation model”, Management Science, vol. 43, no. 10, pp. 1437-1446, 1997.

[5] Paul A. Samuelson, “The fundamental approximation theorem of portfolio analysis in terms of means, variances, and higher moments”, Review of Economic Studies, vol. 37, no. 4, pp. 537-542, 1970.

[6] Hiroshi Konno, Hiroshi Shirakawa, Hiroaki Yamazaki, “A mean-absolute deviation-skewness portfolio optimization model”, Annals of Operations Research, vol. 45, no. 1, pp. 205-220, 1993.

[7] Hiroshi Konno, Ken-ichi Suzuki, “A mean-variance-skewness portfolio optimization model”, Journal of the Operations Research Society of Japan, vol. 38, no. 2, pp. 173-187, 1995.

[8] Bruce R. Feiring, Wuilam Wong, Maychi Poon, Yanchong Chan, “Portfolio selection in downside risk optimization approach: application to the hong kong stock market”, International Journal of Systems Science, vol. 25, no. 11, pp. 1921-1929, 1994.
[9] Tien-Jye Chang, Nigel Meade, John E. Beasley, Yazid M. Sharaiha, “Heuristics for cardinality constrained portfolio optimization”, Computers and Operations Research, vol. 27, no. 13, pp. 1271-1302, 2000.
[10] Alberto Fernandez, Sergio Gomez, “Portfolio selection using neural networks”, Computers and Operations Research, vol. 34, no. 4, pp. 1177-1191, 2007.
[11] Tunchan Cura, “Particle swarm optimization approach to portfolio optimization”, Nonlinear Analysis: Real World Applications, vol. 10, no. 4, pp. 2396-2406, 2009.
[12] Kok Thim Chan, Voon Choong Yap, Hock Han Seah, "Optimizing Portfolio Construction Using Artificial Intelligence", International Journal of Advancements in Computing Technology, vol. 3, no. 3, pp. 168-175, 2011.
[13] Jianguo Zheng, Liang Zhou, "Multi-objective Model for Uncertain Portfolio Optimization Problems", International Journal of Advancements in Computing Technology, vol. 3, no. 8, pp. 138-145, 2011.
[14] Dervis Kraboga,” An idea based on honey bee swarm for numerical optimization”, Tech. rep., Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[15] Dervis Karaboga, Bahriye Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, 2007.
[16] Dervis Karaboga, Bahriye Basturk, “On the performance of artificial bee colony (ABC) algorithm”, Applied Soft Computing, vol. 8, no. 1, pp. 687-697, 2008.
[17] Dervis Karaboga, Bahriye Akay, “A comparative study of artificial bee colony algorithm”, Applied Mathematics and Computation, vol. 214, no. 1, pp. 108-132, 2009.
[18] Guoping Zhu, Sam Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization”, Applied Mathematics and Computation, vol. 217, no. 7, pp. 3166-3173, 2010.