Physica A 429 (2015) 125–139

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem

Wei Chen *

School of Information, Capital University of Economics and Business, Beijing, 100070, PR China

HIGHLIGHTS

- A possibilistic semi-absolute deviation model with real-world constraints is proposed.
- A modified artificial bee colony (MABC) algorithm is developed to solve the proposed model.
- Real-world constraints have great influence on optimal strategies making.
- MABC algorithm outperforms several heuristic algorithms.

ARTICLE INFO

Article history:
Received 7 September 2014
Received in revised form 4 January 2015
Available online 20 February 2015

Keywords:
Fuzzy portfolio optimization
Semiabsolute deviation
Cardinality constraint
Artificial bee colony algorithm
Heuristics

ABSTRACT

In this paper, we discuss the portfolio optimization problem with real-world constraints under the assumption that the returns of risky assets are fuzzy numbers. A new possibilistic mean-semiabsolute deviation model is proposed, in which transaction costs, cardinality and quantity constraints are considered. Due to such constraints the proposed model becomes a mixed integer nonlinear programming problem and traditional optimization methods fail to find the optimal solution efficiently. Thus, a modified artificial bee colony (MABC) algorithm is developed to solve the corresponding optimization problem. Finally, a numerical example is given to illustrate the effectiveness of the proposed model and the corresponding algorithm.

1. Introduction

Portfolio selection discusses the problem of how to allocate a certain amount of investors' wealth among different assets and form a satisfying portfolio. The mean–variance (M–V) model proposed by Markowitz [1] has become the foundation of the modern finance theory since 1950s. It combines probability with optimization techniques to model the behavior investment under uncertainty. The key principle of the M–V model is to use the expected return of a portfolio as the investment return and to use the variance of a portfolio as the risk measure. After Markowitz's work, a lot of work has been done to improve and extend the standard M–V model in three directions: (i) the simplification of the type and amount of the input data; (ii) the introduction of alternative measures of risk; and (iii) the inclusion of the real-world constraints. In this study we concentrate on the second and third directions.

In the original Markowitz model [1], the risk is measured by the standard deviation or variance. However, as pointed out by Grootveld and Hallerbach in Ref. [2], the distinguished drawback is that variance treats high returns as equally unde-

* Tel.: +86 1083952388; fax: +86 1083951017.
E-mail address: chenwei@cueb.edu.cn.

http://dx.doi.org/10.1016/j.physa.2015.02.060
0378-4371/

sirable as low returns because high returns will also contribute to the extreme of variance. In particular, when probability distribution of security returns are asymmetric, variance becomes a deficient measure of investment risk because it may have a potential danger to sacrifice too much expected return in eliminating both low and high return extremes. Thus, some researchers have proposed various types of portfolio models based on alternative risk measures, e.g., safety-first model [3], mean–semivariance model [4], mean absolute deviation (MAD) model [5], mean semiabsolute deviation models [6,7], maximim model [8], etc. More researches can be found in Refs. [9–11].

In practical financial applications, the portfolio optimization problem has to take into account real features such as transaction costs, cardinality constraint imposing a limit on the number of assets in the portfolio, quantity constraints restricting the proportion of each asset in the portfolio to lie between lower and upper bounds, and minimum transaction lots. Such realistic constraints form mixed integer nonlinear programming problem which falls into the class of considerably more difficult NP-hard problems [12]. Therefore, several researches have focused on the heuristic algorithms for complex constrained portfolio optimization problems. Chang et al. [13] proposed a cardinality constrained mean–variance (CCMV) model, and employed three heuristic algorithms based upon a genetic algorithm (GA), tabu search (TS) and simulated annealing (SA) to solve it. Ehrgott et al. [14] applied four different heuristic solution methods including GA, TS, SA and local search (LS) for the cardinality constrained portfolio problem. Soleimani et al. [15] proposed an improved GA for Markowitz's model with minimum transaction lots, cardinality constraints and market capitalization. Later, Anagnostopoulos and Mamanis [16] presented a computational comparison of five multi-objective evolutionary algorithms, i.e., NSGA-II, SPEA2, NPGA2, PESA and e-MOEA, on the tri-objective portfolio optimization problem. In their study, beyond risk and return they also considered an additional objective which minimizes the number of assets in a portfolio. Apart from the above mentioned, other heuristic algorithms have also been used to solve the constrained portfolio optimization problems, including ant colony optimization (ACO) algorithm [17], neural network (NN) [18], particle swarm optimization (PSO) algorithm [19,20], differential evolution (DE) algorithm [21,22], bacteria foraging optimization (BFO) algorithm [23,24].

All the above literatures assume that the security returns are random variables. However, if there is not enough historical data, it is more reasonable to assume them as fuzzy variables. Since Zadeh introduced the fuzzy set theory, many researchers have studied portfolio selection problem in fuzzy environments, such as Tanaka and Guo [25], Carlsson et al. [26], Vercher et al. [27], Smimou et al. [28], Chen et al. [29], Sadjadi et al. [30], Wang et al. [31], Kamdem et al. [32], Tsaur [33]. Though great progress has been made in fuzzy portfolio selection problems, none of the above-cited papers incorporate real-world constraints such as transaction costs and cardinality constraints. Recently, various fuzzy portfolio models with complex constraints have been proposed, and solved by heuristic algorithms. Li and Xu [34] applied GA algorithm to solve a fuzzy portfolio selection model with cardinality and buy-in thresholds constraints. Lin and Liu [35] used GA for the fuzzy multiobjective portfolio selection problem with minimum transaction lots. Chen and Zhang [36] applied PSO algorithm for the admissible portfolio selection problem with transaction costs. Bermúdez et al. [37] applied GA algorithm for a bi-objective fuzzy portfolio selection problem with cardinality constraints. Later, Gupta et al. [38] developed a hybrid intelligent algorithm integrating fuzzy simulation and genetic algorithm to solve a tri-objective fuzzy portfolio model with cardinality constraints. Zhang et al. in 2014 [39] presented an improved DE algorithm for a multi-period fuzzy lower semi-deviation model with risk control. Liu and Zhang [40] applied GA algorithm to solve a multi-period fuzzy portfolio optimization model which includes transaction costs, cardinality constraint and minimum transaction lots.

ABC algorithm is a relatively new meta-heuristic algorithm first developed by Karaboga [41], which is inspired by the intelligent behavior of honey bee swarm. Some researches [42–44] demonstrate that ABC is simple in concept, few in parameters, easy for implementation and more effective than some other population-based algorithms such as GA, PSO, ACO and DE. Therefore, ABC algorithm has aroused much interest and has been successfully used in different fields, such as function optimization, neural networks training, image processing, data mining, scheduling, and engineering design. For a detailed list of ABC usage please see recent comprehensive surveys on ABC [45–47]. In addition to the above mentioned applications, nowadays, some researchers have applied ABC algorithm for portfolio optimization problems. For example, Wang et al. [48] proposed an improved ABC algorithm for solving CCMV model and compared the results with those obtained by Chang et al. [13]. Later, for the same model, Chen et al. [49] and Tuba and Bacanin [50] used standard ABC algorithm and improved ABC algorithm by hybridization with firefly algorithm (FA) to solve it, respectively. Recently, Hsu [51] developed an integrated approach based on data envelopment analysis (DEA), ABC and genetic programming (GP) for the portfolio selection problem. Chen et al. [52] proposed an improved ABC algorithm to solve a fuzzy mean–variance portfolio selection model under four kinds of transaction costs, and compared it with standard ABC and GA.

In summary, there is few research on constructing fuzzy portfolio selection model by using semiabsolute deviation as risk measurement under the real-world constraints, and then solving the corresponding model by ABC algorithm.

The purpose of this paper is to construct a mean-semiabsolute deviation portfolio model based on the possibility theory and to develop an efficient heuristic approach based on ABC algorithm for solving proposed model. The main contributions of our paper can be summarized as follows. We propose a possibilistic mean-semiabsolute deviation portfolio model including transaction costs, cardinality and quantity constraints, in which for a given return level, the investor penalizes the negative semiabsolute deviation that is defined as a risk. Meanwhile, we present a MABC algorithm for the solution, in which chaotic initialization based on logistic map is used to produce initial population, and a hybridization method of ABC and PSO is presented to further improve the performance of ABC. Finally, we give a numerical example to illustrate the idea of our model and demonstrate the effectiveness of the designed algorithm.

The rest of the paper is organized as follows. In Section 2, we present a possibilistic semiabsolute deviation model for portfolio selection. In Section 3, our proposed ABC algorithm is introduced. After that, an example is given to illustrate the effectiveness of the proposed model and algorithm in Section 4. Section 5 presents conclusions.

2. The possibilistic semi-absolute deviation model

Suppose that an investor considers portfolio selection with n risky assets. In order to describe it conveniently, we use the following notations:

x_j, the proportion invested in asset j;
r_j, the random return rate of asset j;
σ_{ij}, the covariance between the return of assets i and j;
k_j, the constant rate of transaction cost for the risky asset j;
m, the maximum number of assets in the portfolio, $1 \le m \le n$;
ε_j, the minimum proportion that must be held of asset j if any of asset j is held;
δ_j, the maximum proportion that must be held of asset j if any of asset j is held;
z_j, the binary variable indicating whether asset j is included in the portfolio or not. $z_j = 1$, if asset j is included in the portfolio, and $z_j = 0$ otherwise;
$i, j = 1, 2, \ldots, n$.

Transaction costs are inevitably present in practical applications of portfolio selection and can be used to capture a number of costs such as brokerage fees, bid–ask spreads, taxes, or even fund loads [53]. The transaction costs yield a decrease in the portfolio return. Thus, there is a practical need to include such costs in optimization models for portfolio selection. Here, we assume that the transaction cost is a V-shaped function of differences between a new portfolio $x = (x_1, x_2, \ldots, x_n)$ and the existing portfolio $x_0 = (x_1^0, x_2^0, \ldots, x_n^0)$. That is to say, the transaction cost of jth asset can be expressed as

$$c_j = k_j|x_j - x_j^0|, \quad j = 1, 2, \ldots, n.$$

Hence, the total transaction costs of n risky assets are

$$\sum_{j=1}^{n} c_j = \sum_{j=1}^{n} k_j|x_j - x_j^0|.$$

Furthermore, the net return on the portfolio after paying transaction costs is given by

$$\sum_{j=1}^{n} r_j x_j - \sum_{j=1}^{n} k_j|x_j - x_j^0|.$$

For a new investor, it can be taken that $x_j^0 = 0,\ j = 1, 2, \ldots, n$.

It is known that very small weighting of an asset will have no distinct influence on the portfolio's return, but the administrative and monitoring costs will increased. Similarly, very high weighting in any asset will cause investors to suffer from a larger risk. Thus, quantity constraints have to be included in the portfolio model. Specifically, a minimum ε_j and a maximum δ_j for each asset j are given, and we impose that either $x_j = 0$ or $\varepsilon_j \le x_j \le \delta_j$. In addition, in order to facilitate management, some investors may wish to limit the number of assets held in their portfolios. Introducing for asset j a binary variable z_j ($z_j = 1$ if asset j is in the portfolio and 0 otherwise), the cardinality constraint can be expressed as $\sum_{j=1}^{n} z_j = m$. This constraint can also be defined in inequality form, imposing that the portfolio must contain no more ($\le$) than m assets.

Following Markowitz's idea, we quantify investment return by the expected value of a portfolio, and risk by the variance. Then, an optimal portfolio should be the one with minimal investment risk for a given level of return. Thus, the constrained portfolio selection problem in the M–V framework can be formulated as:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} E(r_j)x_j - \sum_{j=1}^{n} k_j|x_j - x_j^0| \ge \mu, \\
& \sum_{j=1}^{n} x_j = 1, \\
& \sum_{j=1}^{n} z_j = m, \\
& \varepsilon_j z_j \le x_j \le \delta_j z_j, \quad j = 1, 2, \ldots, n, \\
& z_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n, \\
& x_j \ge 0, \quad j = 1, 2, \ldots, n,
\end{aligned} \tag{1}$$

where $E(r_j)$ is the expected returns of asset j, μ is the given level of return.

By solving the above optimization problem continuously with a different μ each time, a set of efficient points is traced out. This efficient set is called the efficient frontier, and it is a curve that lies between the global minimum risk portfolio and the maximum return portfolio. In other words, the portfolio selection problem is to find all the efficient portfolios along this frontier.

In order to enrich the model, a weighting parameter $\lambda (0 \le \lambda \le 1)$ is introduced to reflect different investor risk attitudes. With this new parameter, the model (1) can be described as follows:

$$\begin{aligned}
\min \quad & \lambda \left[\sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \sigma_{ij} \right] - (1-\lambda) \left[\sum_{j=1}^{n} E(r_j) x_j - \sum_{j=1}^{n} k_j |x_j - x_j^0| \right] \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_j = 1, \\
& \sum_{j=1}^{n} z_j = m, \\
& \varepsilon_j z_j \le x_j \le \delta_j z_j, \quad j = 1, 2, \ldots, n, \\
& z_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n, \\
& x_j \ge 0, \quad j = 1, 2, \ldots, n.
\end{aligned} \tag{2}$$

Obviously, the greater the factor λ is, the more risk averse the investor is. When $\lambda = 1$, the investor will be extremely conservative because in this case only the risk is considered and no attention is paid to the returns. Conversely, $\lambda = 0$ means that the investor is extremely aggressive in pursuing the returns, completely ignoring the investment risk. Most investors' risk preference will lie somewhere between these two extremes.

In order to use the model (2), it is necessary to estimate the probability distribution of the portfolio return. It is well-known that the returns of risky assets are in a fuzzy economic environment and vary from time to time, the future states of returns and risks of risky assets cannot be predicted accurately. Moreover, by using fuzzy approaches, it is better to handle the vagueness and ambiguity in the investment environment and the investors' subjective opinions can be better integrated. Due to these facts, it is useful and meaningful to discuss the portfolio problem under the assumption that the returns of the assets are fuzzy numbers. Similar to the possibilistic approach introduced by Carlsson et al. [26], and Liu and Zhang [40], we assume that the returns of assets are trapezoidal fuzzy numbers. Trapezoidal possibilistic distribution is only considered because it can easily be generalized to the case of possibility distribution of type LR or triangle.

In the following section, we assume that r_j is a trapezoidal fuzzy number with tolerance interval $[a_j, b_j]$, left width α_j and right width β_j, $j = 1, 2, \ldots, n$, i.e., $r_j = (a_j, b_j, \alpha_j, \beta_j)$. r_j can be described with the following membership function:

$$r_j(t) = \begin{cases} 1 - \dfrac{a_j - t}{\alpha_j}, & \text{if } a_j - \alpha_j \le t \le a_j, \\ 1, & \text{if } a_j \le t \le b_j, \\ 1 - \dfrac{t - b_j}{\beta_j}, & \text{if } b_j \le t \le b_j + \beta_j, \\ 0, & \text{otherwise.} \end{cases}$$

Based on Liu [54], we know that the sum of independent trapezoidal fuzzy variables $\xi = (a_1, a_2, \alpha_1, \beta_1)$ and $\eta = (b_1, b_2, \alpha_2, \beta_2)$ is also a trapezoidal fuzzy variable, i.e., $\xi + \eta = (a_1 + b_1, a_2 + b_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)$. Moreover, the product of a trapezoidal fuzzy variable $\xi = (a_1, a_2, \alpha_1, \beta_1)$ and a scalar number $\rho \ge 0$ is also a trapezoidal fuzzy variable, i.e., $\rho\xi = (\rho a_1, \rho a_2, \rho\alpha_1, \rho\beta_1)$. Therefore, the total return of a portfolio $x = (x_1, x_2, \ldots, x_n)(x_j \ge 0,\ j = 1, 2, \ldots, n)$ is still a trapezoidal fuzzy variable in the form of

$$\begin{aligned}
P &= \sum_{j=1}^{n} r_j x_j \\
&= \left(\sum_{j=1}^{n} a_j x_j, \sum_{j=1}^{n} b_j x_j, \sum_{j=1}^{n} \alpha_j x_j, \sum_{j=1}^{n} \beta_j x_j \right) \\
&= (P_l(x), P_u(x), C(x), D(x)).
\end{aligned}$$

It is easy to obtain the γ-level set of P. That is,

$$[P]^\gamma = [P_l(x) - C(x)(1-\gamma), P_u(x) + D(x)(1-\gamma)].$$

Carlsson and Fullér [55] introduced the lower and upper possibilistic mean values of fuzzy number A with γ-level set $[A]^\gamma = [a_1(\gamma), a_2(\gamma)]$ $(\gamma > 0)$ as

$$M_*(A) = 2\int_0^1 \gamma a_1(\gamma)d\gamma,$$
$$M^*(A) = 2\int_0^1 \gamma a_2(\gamma)d\gamma.$$

Furthermore, the interval-valued and crisp possibilistic mean values of fuzzy number A are defined as (see Ref. [55])

$$M(A) = [M_*(A), M^*(A)],$$
$$\overline{M}(A) = \frac{M_*(A) + M^*(A)}{2}.$$

According to the above definitions, we easily obtain the lower and upper possibilistic means, the interval-valued and crisp possibilistic mean values of the total fuzzy return as follows,

$$M_*(P) = 2\int_0^1 \gamma(P_l(x) - C(x)(1-\gamma))d\gamma = P_l(x) - \frac{1}{3}C(x),$$
$$M^*(P) = 2\int_0^1 \gamma(P_u(x) + D(x)(1-\gamma))d\gamma = P_u(x) + \frac{1}{3}D(x),$$
$$M(P) = \left[P_l(x) - \frac{1}{3}C(x), P_u(x) + \frac{1}{3}D(x)\right],$$
$$\overline{M}(P) = \frac{1}{2}(P_l(x) + P_u(x)) - \frac{1}{6}(C(x) - D(x)).$$

The following theorem can be found in Ref. [27].

Theorem 1. *Let $r_j = (a_j, b_j, \alpha_j, \beta_j)$ be n trapezoidal return of asset j, $j = 1, \ldots, n$, and let $P = (P_l(x), P_u(x), C(x), D(x))$ be the total return of a portfolio $x = (x_1, x_2, \ldots, x_n)$, then:*

(a) $\max\{0, M(P) - P\} = (0, P_u(x) - P_l(x) + D(x)/3, 0, C(x))$,
(b) $\omega(P) = M(\max\{0, M(P) - P\}) = [0, P_u(x) - P_l(x) + (C(x) + D(x))/3]$.

In this paper, we will use possibilistic semiabsolute deviation to measure risk. Based on the probabilistic theory, Speranza [6] introduced the semiabsolute deviation as

$$E\left(\left|\min\left\{0, \sum_{j=1}^n r_j x_j - E(\sum_{j=1}^n r_j x_j)\right\}\right|\right) = E\left(\max\left\{0, E\left(\sum_{j=1}^n r_j x_j\right) - \sum_{j=1}^n r_j x_j\right\}\right).$$

Furthermore, the possibilistic semiabsolute deviation can be defined as

$$\omega(P) = M(\max\{0, M(P) - P\}).$$

Based on Theorem 1, the interval-valued possibilistic semiabsolute deviation is represented as follows:

$$\omega(P) = [0, P_u(x) - P_l(x) + (C(x) + D(x))/3].$$

Furthermore, we can obtain the crisp possibilistic semiabsolute deviation of the return associated with the portfolio $x = (x_1, x_2, \ldots, x_n)$ as

$$\overline{\omega}(P) = \frac{P_u(x) - P_l(x)}{2} + \frac{C(x) + D(x)}{6}$$
$$= \frac{1}{2}\sum_{j=1}^n \left(b_j - a_j + \frac{1}{3}(\alpha_j + \beta_j)\right)x_j.$$

Moreover, the possibilistic mean value of the return associated with the portfolio $x = (x_1, x_2, \ldots, x_n)$ is given by

$$\overline{M}(P) = \frac{1}{2}(P_l(x) + P_u(x)) - \frac{1}{6}(C(x) - D(x))$$
$$= \sum_{j=1}^n \frac{1}{2}\left(a_j + b_j + \frac{\beta_j - \alpha_j}{3}\right)x_j.$$

Using the possibilistic mean to measure the portfolio return, and the possibilistic semiabsolute deviation to measure the portfolio risk, the portfolio model (2) can be formulated as follows:

$$\begin{aligned}
\min\quad & \frac{\lambda}{2}\sum_{j=1}^{n}\left[b_j - a_j + \frac{1}{3}(\alpha_j+\beta_j)\right]x_j - (1-\lambda)\left\{\sum_{j=1}^{n}\frac{1}{2}\left[a_j+b_j+\frac{1}{3}(\beta_j-\alpha_j)\right]x_j - \sum_{j=1}^{n}k_j|x_j - x_j^0|\right\}\\
\text{s.t.}\quad & \sum_{j=1}^{n}x_j = 1,\\
& \sum_{j=1}^{n}z_j = m,\\
& \varepsilon_j z_j \le x_j \le \delta_j z_j,\quad j=1,2,\ldots,n,\\
& z_j\in\{0,1\},\quad j=1,2,\ldots,n,\\
& x_j \ge 0,\quad j=1,2,\ldots,n.
\end{aligned}\tag{3}$$

By solving the problem for a set of values of λ ($\lambda \in [0, 1]$) it is possible to obtain an optimal solution for the problem (3).

3. Modified ABC algorithm for portfolio optimization problem

The proposed model (3) is classified as a mixed integer nonlinear programming model necessitating the use of efficient heuristic algorithms to find the solution. In this paper, we propose a heuristic based on ABC algorithm. In the following, standard ABC algorithm is first reviewed and then, modified ABC algorithm to solve the model (3) is presented.

3.1. Standard ABC algorithm

This section outlines the standard ABC algorithm briefly which can be referred to Refs. [42–44]. In the ABC algorithm, the bees are divided into three types: employed bees, onlookers and scouts. Half of the colony of artificial bees is employed bees and the other is onlooker bees and scout bees. Employed bees are responsible for exploiting available food sources, namely, a possible solution for the problem under consideration, and gathering required information. They also share the information with the onlookers, and the onlookers select existing food sources to be further explored. When the food source is abandoned by its employed bee, the employed bee becomes a scout and starts to search for a new food source in the vicinity of the hive. The abandonment happens when the quality of the food source is not improved after performing a maximum allowable number of iterations.

The onlooker bees probabilistically choose food sources. Food sources of higher fitness have a larger chance of being selected by onlooker bees than the ones of lower fitness. The probability value p_i that a food source will be selected can be calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i},\tag{4}$$

where fit_i is the fitness value of solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i. SN is the number of food sources which is equal to the number of employed bees or onlooker bees. In this way, the employed bees exchange their information with the onlookers.

In order to produce a candidate food position $v_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,D}]$ from the old one $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}]$ in memory, the ABC uses the following expression,

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}),\tag{5}$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. k is different from i. D is the number of variables (problem dimension). $\phi_{i,j}$ is randomly generated between $[-1, 1]$. The above explanation implies that the other components of v_i except for dimension j are the same as the ones of x_i. That is, in each iteration, only one dimension of each position is changed.

After each candidate source position is produced and evaluated by the artificial bee, its performance is compared with that of the old one. If the new food source v_i has an equal or better quality than the old source x_i, the old one is replaced by the new one. Otherwise, the old one is retained.

In the ABC algorithm, if a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned by its employed bee. The value of predetermined number of cycles is an important control parameter in the ABC algorithm, which is called "*limit*" for abandonment. These employed bees become scout bees. Assume that the abandoned food source is x_i and $j \in \{1, 2, \ldots, D\}$, then the scout discovers a new food source to be replaced with x_i. This operation can be defined as

$$x_{i,j} = x_j^{\min} + \text{rand}(0, 1)(x_j^{\max} - x_j^{\min}),\tag{6}$$

Table 1
Chaotic initialization based on logistic equation.

Set the population size SN, the dimension D
Randomly initialize the first chaotic variable $cx_{1,j}, j \in [0, D)$
for $j = 1 : D$
  $cx_{1,j} = rand(0, 1)$
end for
// The iterative process of chaotic initialization
for $i = 1 : SN$
  for $j = 1 : D$
    $x_{i,j} = x_j^{\min} + cx_{i,j} \times (x_j^{\max} - x_j^{\min})$
    $cx_{i+1,j} = 4 \times cx_{i,j} \times (1 - cx_{i,j})$
  end for
end for

where $x_{i,j}$ is the jth component of i solution vector, $x_j^{\min}$ and $x_j^{\max}$ are the minimum and maximum values of j components for the x_i vector. rand(0, 1) is a uniformly distributed random number in the range of (0, 1).

Based on the above discussions, we formally describe the main steps of the standard ABC algorithm as follows,

Initialize the population of solutions $x_{i,j}$, $i = 1, 2, \ldots, SN$, $j = 1, 2, \ldots, D$.
Evaluate the population
cycle = 1
repeat
  Produce new solution $v_{i,j}$ for the employed bees by using (5) and evaluate them
  Apply the greedy selection process for the employed bees
  Calculate the probability values $p_{i,j}$ for the solutions $x_{i,j}$ by using (4)
  Generate the new solutions $v_{i,j}$ for the onlookers from the solutions $x_{i,j}$ selected depending on $p_{i,j}$ and evaluate them
  Apply the greedy selection process for the onlookers
  Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution x_i by using (6)
  Memorize the best solution achieved so far
  cycle = cycle + 1
Until cycle = Maximum Cycle Number

3.2. *The proposed MABC algorithm*

3.2.1. *Initial population*

Population initialization is a crucial task for heuristic algorithms. If no information about the solution is available, then random initialization is the most commonly used method to generate candidate solutions. However, this method may affect the algorithm performance on global convergence and convergence speed. Recently, chaotic sequences have been adopted instead of random sequences and some good results have been shown in many applications [56,57]. They have also been used together with some heuristic optimization algorithms to express optimization variables [58,59]. The choice of chaotic sequences is justified theoretically by their unpredictability, i.e., by their spread-spectrum characteristic, non periodic, complex temporal behavior, and ergodic properties. In this paper, chaos-based initialization method is employed to improve the convergence characteristics and to prevent the ABC from getting stuck on local solutions. The pseudo-code of the proposed chaotic initialization is given in Table 1.

3.2.2. *New search method*

In a robust search process, exploration and exploitation process should be well balanced. The exploration is the ability to investigate the various unknown regions in the problem space to discover a good optimum, hopefully the global one. While, the exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely. The exploitation and exploration contradict each other and the two abilities should be well balanced in order to achieve good optimization performance.

The main challenging problem of the ABC algorithm is that its convergence speed is typically lower than those of representative population-based algorithms such as DE and PSO when handling unimodal problems and it can easily get trapped in the local optima when solving complex multimodal problems [44]. Moreover, some researches [60,61] also show that ABC algorithm is good at exploration but poor at exploitation. Therefore, a number of variant ABC algorithms have been developed to achieve the two abilities [60–66].

PSO algorithm is one of the recent swarm intelligence methods which is based on natural flocking and swarming behavior of birds and proposed by Kennedy and Eberhart [67]. In PSO algorithm, the position of each particle is updated based on its

Table 2
MABC algorithm for constraints satisfaction.

```
for i = 1 : SN      /*the number of food sources equals SN */
    for j = 1 : D      /*the number of parameters of the problem to be optimized*/
        s_j = x_{i,j}      /*each vector assigned to s_j*/
    end for
    s_j = |x_{i,j} + φ_{i,j}(x_{i,j} − x_{k,j}) + ψ_{i,j}(g_j − x_{i,j})|      /*neighbor of current solution */
    Count the number of new solution, K
    if (K > m)
        Delete the smallest asset
    else
        while (K < m)
          Renew s_j = |x_{i,j} + φ_{i,j}(x_{i,j} − x_{k,j}) + ψ_{i,j}(g_j − x_{i,j})|
        end while
    end if
    L = Σ_{j∈Q} s_j      /*L is the current solution of s_j, Q is the set of current K assets */
    F = 1 − Σ_{j∈Q} ε_j      /*F is the free proportions */
    s_j = ε_j + s_j F/L, ∀j∈Q      /*calculate proportions to satisfy s_j and sum to 1*/
    R = ∅      /*R is a set of j whose proportions are fixed at δ_j*/
    while there exists a j∈(Q − R) with s_j > δ_j
        for all j∈(Q − R)
            if s_j > δ_j R = R ∪ [j]      /*if s_j exceeds δ_j, add to R*/
            end if
        end for
        L = Σ_{j∈(Q−R)} s_j
        F = 1 − Σ_{j∈(Q−R)} ε_j + Σ_{j∈R} δ_j
        s_j = ε_j + s_j F/L, ∀j∈(Q − R)
        s_j = δ_j, ∀j∈R
    end while
end for
```

current search position, the best position ever found by itself called *pbest*, and the global best position ever found by all the particles called *gbest*. In other words, a particle moves towards its best previous position and towards the best particle. However, in ABC algorithm, a new candidate solution is generated according to Eq. (5), which only considers its previous solution value $x_{i,j}$, and the difference between $x_{i,j}$ and its neighboring solution $x_{k,j}$. That is to say, $v_{i,j}$ is modified from $x_{i,j}$ based on comparison with the randomly selected position from its neighboring solution $v_{k,j}$. However, the randomly chosen dimension j may not always guarantee a better quality solution, so leads to slow convergence speed or even makes the search easily fall into undesired local optimum. Therefore, inspired by the PSO exploitation mechanism, a new neighborhood search strategy is introduced for an employed bee or an onlooker bee to update the new candidate food source, which takes advantage of the information of the global best ($g$best) solution. This operation can be defined as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) + \psi_{i,j}(g_j - x_{i,j}), \tag{7}$$

where g_j is jth component of the global best solution, $\psi_{i,j}$ is a uniform random number in $[0, 1]$.

3.2.3. Handling of the constraints

Inspired by the ideas in Ref. [13], two distinct parts, namely, a set Q of m distinct assets and m real numbers s_i $(\varepsilon_i \le s_i \le \delta_i)$, $i \in Q$, are introduced. Therefore, to ensure that s_i meets its ε_i and δ_i, and that the m real numbers satisfy $\sum_{i\in Q} s_i = 1$, a procedure dealing with these constraints is given, which is listed in Table 2.

It should be noted that, an employed bee generates a new food source position v_i $(i = 1, 2, \ldots, SN)$, according to Eq. (5), but it is possible that there are more than m components of v_i greater than 0. Such case also exists in the situation that onlooker bee renews its solutions. Therefore, to make sure that the desired number of assets in the portfolio be m, we keep the m largest values of v_i and set all other v_i to zero. Moreover, considering that optimal solution must satisfy nonnegative for problem (3), a new food source position v_i is produced by the following formula:

$$v_{i,j} = |x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) + \psi_{i,j}(g_j - x_{i,j})|. \tag{8}$$

4. Numerical experiments

This section is organized in three parts: first the parameters of the model and algorithms are set, second to illustrate the impacts of real-world constraints on optimal investment decision, third to evaluate the performance of MABC algorithm by comparing results with those of GA, SA, PSO, DE, and standard ABC.

4.1. Parameters setting

We consider a practical example introduced by Ref. [68]. It should be noted that the historical data of the 30 stocks were collected from the Shanghai Stock Exchange from November 2004 to November 2005. Based on the historical data and the

Table 3
The possibilistic distributions of returns of 30 stocks.

No.	a_i	b_i	α_i	β_i	No.	a_i	b_i	α_i	β_i
1	0.009	0.017	0.008	0.018	2	0.012	0.025	0.01	0.023
3	0.009	0.017	0.0082	0.017	4	0.010	0.019	0.008	0.018
5	0.013	0.027	0.010	0.019	6	0.010	0.025	0.008	0.018
7	0.010	0.020	0.009	0.019	8	0.008	0.016	0.007	0.015
9	0.005	0.015	0.0045	0.017	10	0.007	0.017	0.006	0.016
11	0.011	0.019	0.0090	0.018	12	0.009	0.024	0.008	0.025
13	0.007	0.016	0.0064	0.018	14	0.013	0.032	0.011	0.030
15	0.008	0.015	0.0076	0.017	16	0.011	0.026	0.009	0.028
17	0.012	0.031	0.011	0.030	18	0.006	0.031	0.005	0.026
19	0.011	0.050	0.0095	0.047	20	0.008	0.025	0.006	0.026
21	0.010	0.023	0.008	0.021	22	0.011	0.031	0.010	0.030
23	0.012	0.046	0.010	0.043	24	0.009	0.026	0.008	0.019
25	0.007	0.027	0.0065	0.025	26	0.010	0.036	0.009	0.028
27	0.011	0.029	0.010	0.026	28	0.008	0.043	0.007	0.035
29	0.007	0.035	0.006	0.034	30	0.006	0.031	0.0045	0.029

Table 4
Portfolio parameters.

No.	x_i^0	ε_i	δ_i	k_i	No.	x_i^0	ε_i	δ_i	k_i
1	0.02	0.01	0.35	0.0000	2	0.03	0.01	0.35	0.0000
3	0.00	0.01	0.35	0.0000	4	0.00	0.01	0.35	0.0000
5	0.05	0.01	0.35	0.0010	6	0.00	0.02	0.35	0.0010
7	0.00	0.02	0.35	0.0015	8	0.15	0.02	0.35	0.0015
9	0.00	0.02	0.35	0.0015	10	0.07	0.02	0.35	0.0015
11	0.00	0.025	0.50	0.0000	12	0.00	0.025	0.50	0.0000
13	0.08	0.025	0.50	0.0020	14	0.00	0.025	0.50	0.0020
15	0.15	0.025	0.50	0.0020	16	0.00	0.03	0.50	0.0020
17	0.00	0.03	0.50	0.0020	18	0.05	0.03	0.50	0.0025
19	0.00	0.03	0.50	0.0025	20	0.15	0.03	0.50	0.0025
21	0.00	0.04	0.65	0.0025	22	0.00	0.04	0.65	0.0030
23	0.05	0.04	0.65	0.0030	24	0.00	0.04	0.65	0.0030
25	0.05	0.04	0.65	0.0040	26	0.00	0.05	0.65	0.0040
27	0.00	0.05	0.65	0.0050	28	0.10	0.05	0.65	0.0050
29	0.00	0.05	0.65	0.0055	30	0.05	0.05	0.65	0.0055

corporations' financial reports, the parameters of membership functions for 30 asset returns are listed in Table 3. Moreover, some parameters of the model (3) are listed in Table 4.

In the MABC algorithm, there are three main parameters: the number of food sources (*SN*), the value of *limit* and the maximum cycle number (*MCN*). In the following experiments, *SN* and *limit* are set to 20, and 100, respectively, as in Ref. [61]. In addition, results of repeated experiments by varying *MCN* show that the modified ABC algorithm is convergent and has a good stability when $MCN = 2500$. Thus, *MCN* is set to 2500. Furthermore, in order to make a fair comparison, the values of population size and maximum cycle number are chosen same as the values of MABC algorithm. Such as we select the population size 40, maximum cycle/generation number 2500. Other values of control parameters employed for GA, SA, DE, PSO and standard ABC are presented below.

GA settings: The crossover probability p_c and the mutation probability p_m are set to 0.8 and 0.08, respectively. The selection method is roulette wheel and the crossover method is one-point crossover.

SA settings: The initial temperature $T_0 = 100$, the coefficient controlling the cooling scheme $\alpha = 0.95$.

PSO settings: The inertia weight factor ω is 0.5, the learning factors, c_1 and c_2 are both set to 2.

DE settings: Differential evolution factor $F = 0.5$, crossover rate *CR* is 0.9.

Standard ABC settings: The values of *SN*, *limit* and *MCN* are the same as those of modified ABC.

In addition, a total of 20 runs for each experimental setting are conducted and the average results are given. All algorithms have been implemented in Java language and run on a personal computer having a 2.90 GHz processor and 4 GB RAM.

4.2. Effects of real-world constraints on optimal portfolio selection

Let the desired number of assets in the portfolio be $m = 10$. Using MABC algorithm, we can obtain some efficient portfolios by varying the risk tolerance parameter λ. Some results are listed in Tables 5–7. We can see that, if the investor chooses an aggressive strategy he/she will obtain a higher portfolio return than by choosing a conservative strategy, and stand a higher risk. On the other hand, if the investor prefers a conservative strategy he/she will obtain a lower portfolio return than by choosing an aggressive strategy, and stand a lower portfolio risk. Moreover, the results show that different λ produce different optimal strategies, that is, the investors' subjective opinions have great impact on the optimal portfolio strategy.

Table 5
Possibilistic efficient portfolios with $\lambda = 0$.

Security	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Proportions	0	0.0321	0	0	0	0.0623	0	0	0	0
Security	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Proportions	0	0	0	0.0780	0	0	0.0935	0.0836	0.1576	0.0935
Security	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
Proportions	0	0	0.1249	0	0.1186	0	0	0.1559	0	0

Corresponding investment return and risk are 0.0241 and 0.0402, respectively.

Table 6
Possibilistic efficient portfolios with $\lambda = 0.5$.

Security	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Proportions	0	0.0345	0	0	0.0346	0	0	0.0683	0	0
Security	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Proportions	0.1642	0	0	0	0.0860	0.1024	0.1018	0	0	0.1026
Security	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
Proportions	0	0	0.1366	0	0	0	0	0.1690	0	0

Corresponding investment return and risk are 0.0205 and 0.0311, respectively.

Table 7
Possibilistic efficient portfolios with $\lambda = 1$.

Security	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Proportions	0	0	0.0445	0	0	0	0.0895	0.0895	0.0895	0.0895
Security	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Proportions	0.1118	0	0.1118	0	0.1118	0	0	0	0	0
Security	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
Proportions	0.1789	0	0	0.0831	0	0	0	0	0	0

Corresponding investment return and risk are 0.0125 and 0.0187, respectively.

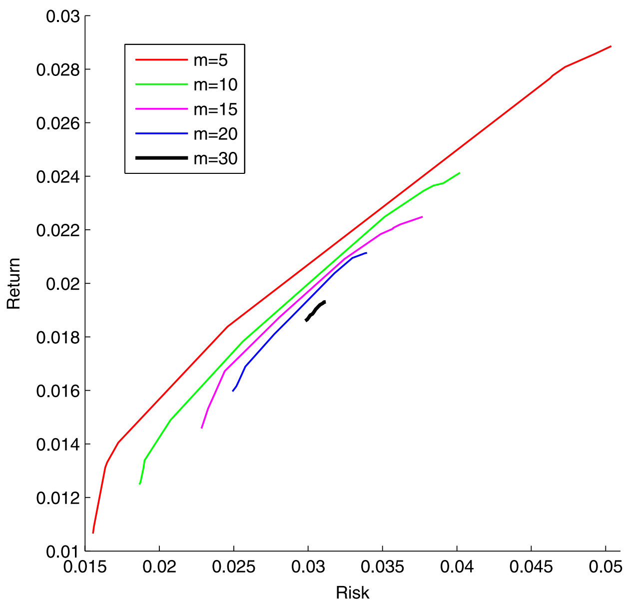

Fig. 1. The possibilistic efficient frontiers.

To demonstrate the effects of the cardinality constraint on the portfolio selection, the possibilistic efficient frontiers with $m = 5, 10, 15, 20, 30$ are constructed by varying λ from 0 to 1 with 0.1 step, which are depicted in Fig. 1. It should be noted that other parameter settings of the portfolio are the same as Table 4. We can see that the possibilistic efficient frontiers become shorter with the increase of the m value, which is consistent with the conclusion obtained by Chang et al. [13].

It must be emphasized that the introduction of cardinality constraints may result in a discontinuous efficient frontier [13]. The discontinuities imply that there are certain combinations of return and risk which are "undefined" for a rational investor, since an alternative portfolio with both a higher return and lower risk exists. However, there are no signs of

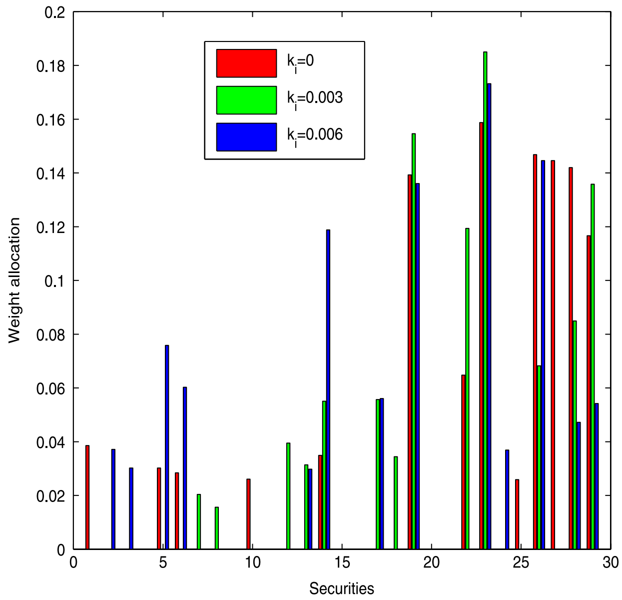

Fig. 2. Effect of different transaction costs.

Table 8
Possibilistic efficient portfolios under different lower bounds ($\delta_j = 0.55$).

Security	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Case 1	0	0	0	0	0.0621	0	0.0513	0	0	0
Case 2	0.0533	0.1755	0	0.0498	0	0	0	0	0	0
Case 3	0	0.0570	0.0538	0	0	0	0	0	0	0
Security	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Case 1	0	0.0507	0	0.0842	0	0	0.0440	0	0.0977	0.0628
Case 2	0	0.0520	0	0.0674	0.0545	0	0	0.0499	0.1029	0.1233
Case 3	0.0543	0	0	0.1042	0.0568	0.1113	0	0	0.1071	0.0606
Security	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
Case 1	0	0.1134	0.1147	0.0737	0	0.0617	0	0.0811	0	0.1026
Case 2	0	0	0.0592	0	0.0548	0	0	0.0660	0	0.0914
Case 3	0	0.0744	0.1030	0	0.0538	0	0	0.1005	0.0632	0

Table 9
Possibilistic efficient portfolios under different upper bounds ($\varepsilon_j = 0.03$).

Security	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Case 1	0	0.0794	0	0.0788	0.0788	0	0	0.0788	0	0
Case 2	0	0.0731	0	0.0731	0.1578	0	0	0.0731	0	0
Case 3	0	0.1045	0	0.0815	0.0815	0.0644	0	0.0817	0	0
Security	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Case 1	0.0792	0	0.0536	0.0822	0.0788	0	0	0	0	0.1203
Case 2	0.0731	0	0.0731	0.0733	0.0731	0	0	0	0	0.0751
Case 3	0.0814	0	0	0.0811	0.0814	0	0	0	0	0.0912
Security	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
Case 1	0	0	0.0788	0	0.0584	0	0	0.0848	0	0.0481
Case 2	0	0	0.0731	0	0.0571	0	0	0.0724	0	0.0526
Case 3	0	0	0.0814	0	0.0367	0	0	0.0816	0	0.0516

any discontinuities in this particular cardinality constrained efficient frontier. In general, for a specified risk value, with more constraints included, the expected returns become smaller. Moreover, the lower and upper bounds have to be chosen carefully so that feasible solutions exist.

Furthermore, to demonstrate the effects of the transaction costs on the portfolio selection, given the risk tolerance parameter $\lambda = 0.45$ and the desired number of assets in the portfolio $m = 8$, the histograms for the optimal portfolio strategies when transaction costs $k_j = 0$, $k_j = 0.003$, and $k_j = 0.006$, $j = 1, \ldots, 30$, are given in Fig. 2. We can see that, (a) the optimal portfolio includes securities 4, 11 and 16 for $k_j = 0$, which are not invested for $k_j = 0.003$ and $k_j = 0.006$;

Table 10
Robustness analysis for the MABC algorithm.

SN	MCN	Limit	$m = 12, \lambda = 0.45$		$m = 18, \lambda = 0.65$	
			Objective value	Relative error (%)	Objective value	Relative error (%)
80	2000	120	−0.01491	0	0.001779	0
70	3000	150	−0.01481	0.6707	0.001798	1.0680
60	2800	130	−0.01478	0.8719	0.001803	1.3491
50	2600	80	−0.01471	1.3414	0.001816	2.0798
40	2500	60	−0.01468	1.5426	0.001782	0.1686
30	2900	90	−0.01464	1.8109	0.001847	3.8224
20	2600	70	−0.01461	2.0121	0.001831	2.9230
20	2400	100	−0.01462	1.9450	0.001787	0.4497
10	2300	100	−0.01454	2.4816	0.001839	3.3737
10	2100	50	−0.01453	2.5486	0.001845	3.7099

Table 11
Performance comparisons with $\lambda = 0.6$.

		GA	SA	PSO	DE	ABC	MABC
$m = 8$	Min	2.22E−4	7.55E−4	−1.48E−5	8.26E−5	−1.17E−4	**−3.98E−4**
	Max	0.00120	0.00289	3.06E−4	0.00143	3.54E−4	**−3.23E04**
	Mean	8.36E−4	0.00195	1.15E−4	7.74E−4	1.30E−4	**−3.56E−4**
	Std	2.90E−4	5.25E−4	9.33E−5	3.52E−4	1.19E−4	**2.03E−5**
$m = 12$	Min	5.19E−4	0.0014	2.07E−4	6.26E−4	3.41E−4	**−2.89E−4**
	Max	0.0013	0.0025	5.98E04	0.0014	8.89E−4	**−1.89E−4**
	Mean	0.0010	0.0020	3.98E−4	0.0010	6.13E−4	**−2.25E−4**
	Std	2.17E−4	2.78E−4	9.40E−5	2.33E−4	1.59E−4	**2.42E−5**
$m = 20$	Min	0.0011	0.0012	9.26E−4	9.32E−4	8.69E−4	**6.81E−4**
	Max	0.0016	0.0019	0.0011	0.0015	0.0012	**8.42E−4**
	Mean	0.0014	0.0016	0.0010	0.0012	0.0011	**7.66E−4**
	Std	1.24E−4	1.82E−4	5.37E−5	1.42E−4	8.13E−5	**4.39E−5**

Table 12
Performance comparisons with $m = 12$.

		GA	SA	PSO	DE	ABC	MABC
$\lambda = 0.8$	Min	0.0060	0.0069	0.0052	0.0055	0.0051	**0.0042**
	Max	0.0076	0.0098	0.0057	0.0070	0.0063	**0.0045**
	Mean	0.0070	0.0085	0.0055	0.0065	0.0058	**0.0043**
	Std	4.65E−4	8.29E−4	**1.47E−4**	4.33E−4	3.14E−4	7.95E−4
$\lambda = 0.5$	Min	−0.0024	−0.0018	−0.0027	−0.0023	−0.0025	**−0.00274**
	Max	−0.0017	−0.0000	−0.0024	−0.0019	−0.0023	**−0.00265**
	Mean	−0.0020	−0.0014	−0.0025	−0.0021	−0.0024	**−0.0027**
	Std	1.47E−4	2.43E−4	**5.01E−5**	1.14E−4	7.26E−5	8.90E−4
$\lambda = 0.2$	Min	−0.0142	−0.0135	−0.0147	−0.0140	−0.01464	**−0.01742**
	Max	−0.0128	−0.0097	−0.0144	−0.0124	−0.01452	**−0.01735**
	Mean	−0.0135	−0.0118	−0.0146	−0.0135	−0.01451	**−0.01739**
	Std	3.71E−4	9.50E−4	6.46E−5	3.42E−4	2.19E−5	**2.14E−5**

(b) when $k_j = 0.006$, the optimal portfolio includes securities 5 and 13, and it does not include these securities for $k_j = 0$ and $k_j = 0.003$; (c) although securities 19, 23 and 28 are all invested in three different cases, the proportions of these securities are also different. All these show that the investor adopts different investment strategies with the transaction costs changing. Moreover, for $k_j = 0$, $k_j = 0.003$, and $k_j = 0.006$, the corresponding portfolio returns are 2.798%, 2.271%, and 1.999%, respectively. This implies that when transaction costs are considered, the portfolio returns will be decreased, which is consistent with the investment theory.

Finally, similar to the above analysis, Tables 8 and 9 are given to demonstrate the effects of quantity constraint on optimal investment strategy. Given the risk tolerance parameter $\lambda = 0.25$ and the desired number of assets in the portfolio $m = 13$, the optimal portfolios are given under different cases. Table 8 shows the optimal portfolio selection with different lower bound constraints, i.e., case 1: $\varepsilon_j = 0.01$, case 2: $\varepsilon_j = 0.02$, case 3: $\varepsilon_j = 0.03$. Similarly, the influence of upper bound on the optimal portfolio strategy can be seen in Table 9, in which three cases are also considered, i.e., case 1: $\delta_j = 0.25$, case 2: $\delta_j = 0.5$, case 3: $\delta_j = 0.75$. From Tables 8–9, we can find out that different values of lower and upper bounds lead to different investment strategies, which implies that quantity constraints also play an important role in portfolio decision-making.

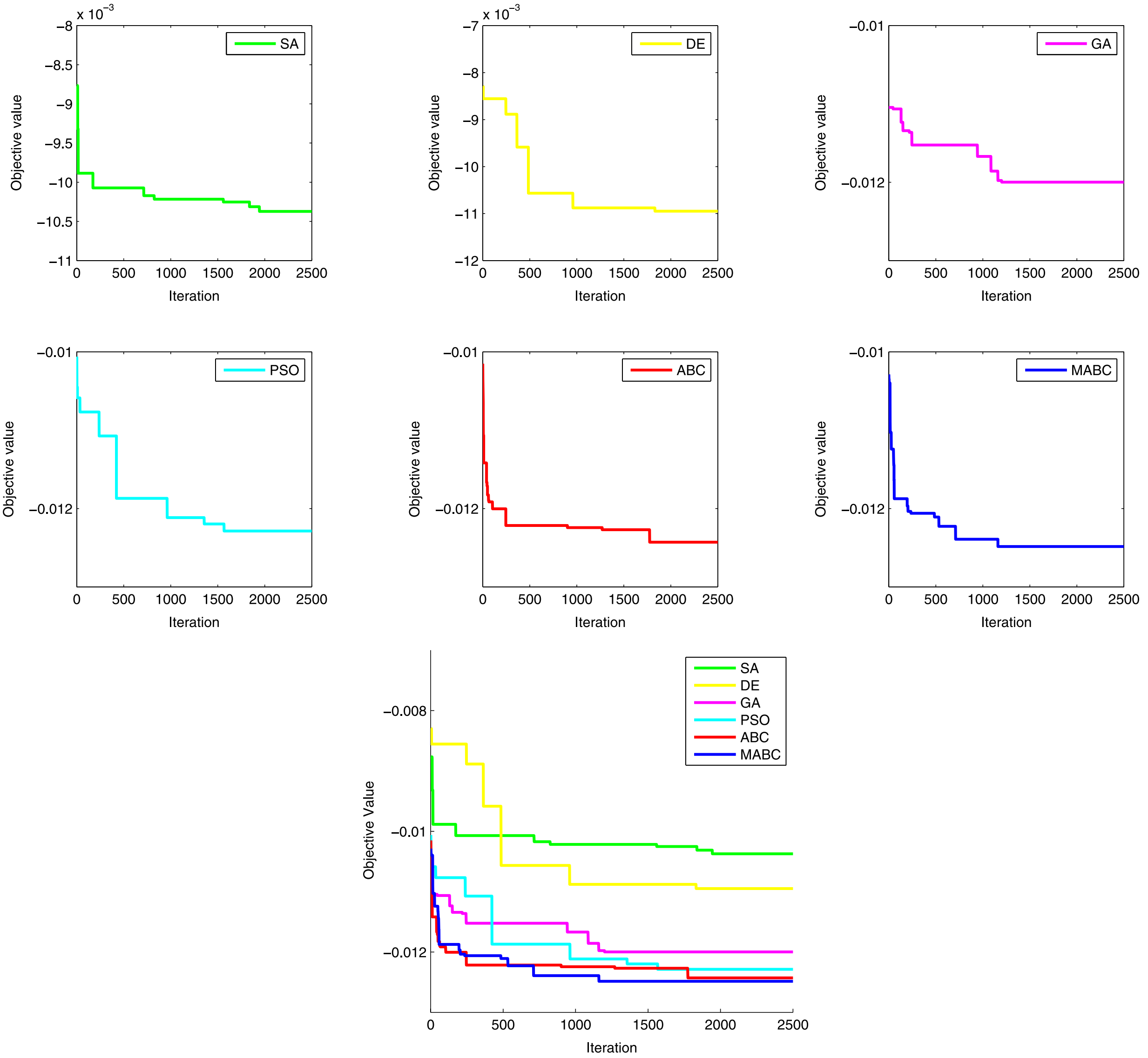

Fig. 3. Convergence characteristics of different algorithms ($\lambda = 0.3$, $m = 5$).

4.3. *Performances of the modified ABC*

To show the robustness of MABC algorithm, we use relative error as the index, i.e., (maximum−actual value)/maximum × 100%, where the maximum is the maximal value of all the objective values calculated. The detailed results are shown in Table 10. It can be seen that the relative errors do not exceed 4% by choosing different parameters in the proposed algorithm, which implies that the designed algorithm is effective and robust to set parameters.

Furthermore, performance indicator parameters such as minimum, maximum, mean and standard deviation values of the optimal objective by different algorithms after 20 independent runs are tabulated in Tables 11–12. In these two tables, the best results are bolded. From Tables 11 and 12, it can be easily observed that, in all cases, the minimum, maximum, and mean results obtained by the MABC algorithm are better than those of other listed algorithms. That is to say, the modified ABC can obtain a better solution for the proposed portfolio problem. Moreover, in most cases, the lower values of standard deviations in MABC algorithm indicate the fact that the algorithm not only leads to more quality solutions but also it is more stable and reliable in finding quality solutions.

Finally, the convergence characteristic of different algorithms is shown in Fig. 3. From Fig. 3, it can be seen that modified ABC can quickly converge to a better solution much faster than the other algorithms.

In summary, obtained results show that the MABC algorithm possesses superior performance in accuracy, convergence speed, stability and robustness, as compared to the other algorithms. Hence, the MABC algorithm may be a good alternative to deal with complex constrained portfolio optimization problems.

5. Conclusion

In the financial market, the expected rates of security returns cannot be well reflected by historical data because of the high volatility of market environment. In this paper, we propose a possibilistic semi-absolute deviation portfolio selection problem which includes transaction costs, cardinality and quantity constraints. The proposed model is a mixed integer nonlinear programming problem for which there exists no efficient algorithm. Thus, we develop a novel algorithm, MABC, to solve this model by introducing a chaotic initialization approach and a hybridization method of ABC and PSO. Finally, a numerical example is given to illustrate the effectiveness of the proposed model and the corresponding algorithm. The experimental results demonstrate that real-world constraints have a great impact on the optimal investment strategy, and MABC algorithm has a better performance than the standard ABC algorithm and other heuristic algorithms, such as GA, SA, PSO, DE.

Finally, for future researches three areas are proposed: first adding other constraints of real market such as minimum transaction lots, second changing the proposed model to a multi period one, third investigating the effects of three control parameters *SN*, *limit* and *MCN* on the performance of MABC algorithm based on some well-known benchmark functions.

Acknowledgments

The author would like to thank the Editor and the anonymous reviewers for their valuable comments and suggestions to improve presentation of the paper. This research was supported by the Specially Commissioned Project of the Capital University of Economics and Business, the Humanity and Social Science Youth foundation of Ministry of Education of China (No. 13YJC630012), and the Training Programme Foundation for the Beijing Municipal Excellent Talents (No. 2013D005019000007).

References

[1] H.M. Markowitz, Portfolio selection, J. Finance 7 (1952) 77–91.
[2] H. Grootveld, W. Hallerbach, Variance vs downside risk: is there really that much difference? European J. Oper. Res. 114 (1999) 304–319.
[3] D. Roy, Safety-first and the holding of assets, Econometrica 20 (1952) 431–449.
[4] H. Markowitz, Portfolio Selection: Efficient Diversification of Investments, Wiley, New York, 1959.
[5] H. Konno, H. Yamazaki, Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market, Manage. Sci. 37 (1991) 519–531.
[6] M.G. Speranza, Linear programming models for portfolio optimization, J. Finance 12 (1993) 107–123.
[7] O. Ogryczak, A. Ruszczynski, From stochastic dominance mean-risk model: semideviation as risk measure, European J. Oper. Res. 116 (1999) 33–50.
[8] M.R. Young, A minimax portfolio selection rule with linear programming solution, Manage. Sci. 44 (1998) 673–683.
[9] U.W. Spreitzer, V. Reznik, On the optimization of a CAPM portfolio using lower partial moments as measure of risk and using the possibility of safeguarding its loss, Physica A 378 (2007) 423–426.
[10] J.C. Li, D.C. Mei, The returns and risks of investment portfolio in a financial market, Physica A 406 (2014) 67–72.
[11] R. Mansini, W. Ogryczak, M.G. Speranza, Twenty years of linear programming based portfolio optimization, European J. Oper. Res. 234 (2014) 518–535.
[12] D.X. Shaw, S. Liu, L. Kopman, Lagrangian relaxation procedure for cardinality-constrained portfolio optimization, Optim. Methods Softw. 23 (2008) 411–420.
[13] T.J. Chang, N. Meade, J. Beasley, Y. Sharaiha, Heuristics for cardinality constrained portfolio optimization, Comput. Oper. Res. 27 (2000) 1271–1302.
[14] M. Ehrgott, K. Klamroth, C. Schwehm, An MCDM approach to portfolio optimization, European J. Oper. Res. 155 (2004) 752–770.
[15] H. Soleimani, H.R. Golmakani, M.H. Salimi, Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm, Expert Syst. Appl. 36 (2009) 5058–5063.
[16] K.P. Anagnostopoulos, G. Mamanis, A portfolio optimization model with three objectives and discrete variables, Comput. Oper. Res. 37 (2010) 1285–1297.
[17] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Ant colony optimization: a metaheuristic approach to multiobjective portfolio selection, Ann. Oper. Res. 131 (2004) 79–99.
[18] A. Fernández, S. Gómez, Portfolio selection using neural networks, Comput. Oper. Res. 34 (2007) 1177–1191.
[19] T. Cura, Particle swarm optimization approach to portfolio optimization, Nonlinear Anal. RWA 10 (2009) 2396–2406.
[20] H.R. Golmakani, M. Fazel, Constrained portfolio selection using particle swarm optimization, Expert Syst. Appl. 38 (2011) 8327–8335.
[21] T. Krink, S. Mittnik, S. Paterlini, Differential evolution and combinatorial search for constrained index-tracking, Ann. Oper. Res. 172 (2009) 153–176.
[22] T. Krink, S. Mittnik, Multiobjective optimization using differential evolution for real-world portfolio optimization, Comput. Manag. Sci. 8 (2011) 157–179.
[23] B. Niu, Y. Fan, H. Xiao, B. Xue, Bacterial foraging based approaches to portfolio optimization with liquidity risk, Neurocomputing 98 (2012) 90–100.
[24] S.K. Mishra, G. Panda, R. Majhi, Constrained portfolio asset selection using multiobjective bacteria foraging optimization, Oper. Res. 14 (2014) 113–145.
[25] H. Tanaka, P. Guo, Portfolio selection based on upper and lower exponential possibility distributions, European J. Oper. Res. 114 (1999) 115–126.
[26] C. Carlsson, R. Fullér, P. Majlender, A possibilistic approach to selecting portfolios with highest utility score, Fuzzy Sets and Systems 131 (2002) 13–21.
[27] E. Vercher, J.D. Bermúdez, J.V. Segura, Fuzzy portfolio optimization under downside risk measures, Fuzzy Sets and Systems 1 (2007) 361–377.
[28] K. Smimou, C.R. Bector, G. Jacoby, Portfolio selection subject to experts' judgments, Int. Rev. Financ. Anal. 17 (2008) 1036–1054.
[29] W. Chen, Y. Yang, H. Ma, Fuzzy portfolio selection problem with different borrowing and lending rates, Math. Probl. Eng. 2011 (2011) 15. Article ID 263240.
[30] S.J. Sadjadi, S.M. Seyedhosseini, Kh. Hassanlou, Fuzzy multi period portfolio selection with different rates for borrowing and lending, Appl. Soft Comput. 11 (2011) 3821–3826.
[31] B. Wang, S. Wang, J. Watada, Fuzzy portfolio selection models with Value-at-Risk, IEEE Trans. Fuzzy Syst. 19 (2011) 758–769.
[32] J.S. Kamdem, C.T. Deffo, L.A. Fono, Moments and semi-moments for fuzzy portfolio selection, Insurance Math. Econom. 51 (2012) 517–530.
[33] R.C. Tsaur, Fuzzy portfolio model with different investor risk attitudes, European J. Oper. Res. 227 (2013) 385–390.
[34] J. Li, J. Xu, A class of possibilistic portfolio selection model with interval coefficients and its application, Fuzzy Optim. Decis. Mak. 6 (2007) 123–137.
[35] C.C. Lin, Y.T. Liu, Genetic algorithms for portfolio selection problems with minimum transaction lots, European J. Oper. Res. 185 (2008) 393–404.
[36] W. Chen, W.G. Zhang, The admissible portfolio selection problem with transaction costs and an improved PSO algorithm, Physica A 389 (2010) 2070–2076.
[37] J.D. Bermúdez, J.V. Segura, E. Vercher, A multi-objective genetic algorithm for cardinality constrained fuzzy portfolio selection, Fuzzy Sets and Systems 188 (2012) 16–26.

[38] P. Gupta, G. Mittal, M.K. Mehlawat, Multiobjective expected value model for portfolio selection in fuzzy environment, Optim. Lett. 7 (2013) 1765–1791.
[39] W.G. Zhang, Y.J. Liu, W.J. Xu, A new fuzzy programming approach for multi-period portfolio optimization with return demand and risk control, Fuzzy Sets and Systems 246 (2014) 107–126.
[40] Y.J. Liu, W.G. Zhang, A multi-period fuzzy portfolio optimization model with minimum transaction lots, European J. Oper. Res. 242 (2015) 933–941.
[41] D. Karaboga, An idea based on honeybee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
[42] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[43] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (2008) 687–697.
[44] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[45] J.C. Bansal, H. Sharma, S.S. Jadon, Artificial bee colony algorithm: a survey, Int. J. Adv. Intell. Paradig. 5 (2013) 123–159.
[46] A.L. Bolaji, A.T. Khader, M.A. Al-Betar, M.A. Awadallah, Artificial bee colony algorithm, its variants and applications: a survey, J. Theor. Appl. Inf. Technol. 47 (2013) 434–459.
[47] D. Karaboga, B. Gorkem, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (2014) 21–57.
[48] Z. Wang, S. Liu, X. Kong, An improved artificial bee colony algorithm for portfolio optimization problem, Int. J. Adv. Comput. Technol. 3 (2011) 67–74.
[49] A.H.L. Chen, Y.C. Liang, C.C. Liu, An artificial bee colony algorithm for the cardinality-constrained portfolio optimization problems, in: Proceedings of the IEEE World Congress on Computational Intelligence, 2012, pp. 1–8.
[50] M. Tuba, N. Bacanin, Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean–variance portfolio selection problem, Appl. Math. Inf. Sci. 8 (2014) 2831–2844.
[51] C.M. Hsu, An integrated portfolio optimisation procedure based on data envelopment analysis, artificial bee colony algorithm and genetic programming, Int. J. Syst. Sci. 45 (2014) 2645–2664.
[52] W. Chen, Y. Yang, H. Ma, M. Sun, Application of artificial bee colony algorithm to portfolio adjustment problem with transaction costs, J. Appl. Math. 2014 (2014) 12. Article ID 192868.
[53] M. Lobo, M. Fazel, S. Boyd, Portfolio optimization with linear and fixed transaction costs, Ann. Oper. Res. 152 (2007) 341–365.
[54] B. Liu, Uncertainty Theory, second ed., Springer, 2007.
[55] C. Carlsson, R. Fullér, On possibilistic mean value and variance of fuzzy numbers, Fuzzy Sets and Systems 122 (2001) 325–326.
[56] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (2010) 5682–5687.
[57] C.H. Yang, S.W. Tsai, L.Y. Chuang, C.H. Yang, An improved particle swarm optimization with double-bottom chaotic maps for numerical optimization, Appl. Math. Comput. 219 (2012) 260–279.
[58] B. Alatas, E. Akin, B. Ozer, Chaos embedded particle swarm optimization algorithms, Chaos Solitons Fractals 40 (2009) 1715–1734.
[59] A.H. Gandomi, G.J. Yun, X.S. Yang, S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 327–340.
[60] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173.
[61] W. Xiang, M. An, An efficient and robust artificial bee colony algorithm for numerical optimization, Comput. Oper. Res. 40 (2013) 1256–1265.
[62] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inform. Sci. 192 (2012) 120–142.
[63] W. Gao, S. Liu, L. Huang, A global best artificial bee colony algorithm for global optimization, J. Comput. Appl. Math. 236 (2012) 2741–2753.
[64] A. Banharnsakun, B. Sirinaovakul, T. Achalakul, The best-so-far ABC with multiple patrilines for clustering problems, Neurocomputing 116 (2013) 355–366.
[65] D. Karaboga, B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, Appl. Soft Comput. 23 (2014) 227–238.
[66] C. Zhang, J. Zheng, Y. Zhou, Two modified artificial bee colony algorithms inspired by grenade explosion method, Neurocomputing 151 (2015) 1198–1207.
[67] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948.
[68] G. Chen, S. Chen, Y. Fang, S. Wang, A model for portfolio selection based on possibilistic return rates, Math. Econ. 23 (2006) 19–25 (in Chinese).