TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/10 pp 75–86
DOI: 10.26599/TST.2018.9010057
Volume 23, Number 1, February 2018

A New Method of Portfolio Optimization Under Cumulative Prospect Theory

Chao Gong*, Chunhui Xu, Masakazu Ando, and Xiangming Xi

Abstract: In this paper, the portfolio selection problem under Cumulative Prospect Theory (CPT) is investigated and a model of portfolio optimization is presented. This model is solved by coupling scenario generation techniques with a genetic algorithm. Moreover, an Adaptive Real-Coded Genetic Algorithm (ARCGA) is developed to find the optimal solution for the proposed model. Computational results show that the proposed method solves the portfolio selection model and that ARCGA is an effective and stable algorithm. We compare the portfolio choices of CPT investors based on various bootstrap techniques for scenario generation and empirically examine the effect of reference points on investment behavior.

Key words: portfolio choice; cumulative prospect theory; bootstrap method; adaptive real-coded genetic algorithm

1 Introduction

In finance, portfolio choice is the process of allocating a person's investable wealth to various financial assets according to criteria that determine the best possible tradeoff between risk and return. Modern portfolio theory, which was proposed by Markowitz[1], provides the theoretical background for the relationship between risk and return of a portfolio and the importance of diversification. The optimal portfolio choice lies on the efficient frontier that combines the return and its standard deviation. Expected Utility Theory (EUT) is a widely accepted criterion to determine portfolio choices under a certain degree of uncertainty. Under EUT, investors (who are generally assumed to be uniformly risk-averse) evaluate the final outcomes using objective probabilities.

Although these theories have been useful in modeling portfolio choices, substantial empirical and experimental evidence, such as the paradoxes outlined in the work of Refs. [2, 3], has revealed that they do not reflect reality. Owing to the assumption that people are rational, these theories merely demonstrate how people should behave instead of making their actual portfolio choices under risk. Prospect Theory (PT), which was proposed by Kahneman and Tversky[4], is widely considered as the best available description of people's actual behavior when evaluating risk in experimental settings, particularly when psychological insights are incorporated[5].

Inspired by Quiggin[6], who distorted the cumulative probabilities of ranked outcomes instead of individual probabilities, Tversky and Kahneman[7] developed Cumulative Prospect Theory (CPT) to overcome inconsistencies with first-order stochastic dominance. CPT can account for diminishing sensitivity, loss aversion, different risk attitudes, and financial phenomena such as the paradoxes of Allais[2] and Ellsberg[3]. Although CPT has received a significant amount of attention (for example, Benartzi and Thaler[8] explained the famous equity premium puzzle using CPT), to the best of our knowledge, few papers have applied it to portfolio choices. The reason may be that the CPT function is generally non-convex and non-concave, which means that traditional optimization

• Chao Gong, Chunhui Xu, and Masakazu Ando are with the Department of Risk Science in Finance and Management, Chiba Institute of Technology, Chiba 275-0016, Japan. E-mail: gongchao80@gmail.com.
• Xiangming Xi was with Department of Automation, Tsinghua University, Beijing 100084, China and joined Huawei Technologies Co. Ltd. since 2016.
∗ To whom correspondence should be addressed.
Manuscript received: 2016-11-13; accepted: 2016-12-23

methods such as Lagrange multipliers and convex duality are not effective and the CPT function may have numerous local maxima[9].

In this paper, we present a method that couples scenario techniques for simulating the scenario of the real stock market with a genetic algorithm to determine the optimal solution. The major challenge is to provide data on mathematical models in determining optimal solutions to address uncertainties in the field of financial investment. The effectiveness of the mathematical models hinges on the quality of the scenarios. Bradley and Crane[10] first introduced these techniques to the financial world. Several scenario generation methods have been used to support financial decision-making. We focus on three variants of the bootstrap method for scenario generation. The bootstrap method, which was introduced by Efron[11], is a form of resampling in statistics. The key idea is to provide a resampling simulation technique to estimate the complex characteristics of the underlying population. The bootstrap method does not generate random variates but instead repeatedly samples the original data[12]. This method is a highly effective tool in the absence of a parametric distribution for a set of data. The bootstrap method is used when the number of available samples is relatively small and a larger number of observations is required. The use of bootstrapping for scenario generation has been suggested by Kouwenberg and Zenios[13]. In the analysis of financial time series, the probability distribution of a data set is unknown; the bootstrap method is suitable for assessing the distribution properties of some statistic of such data.

Genetic Algorithms (GAs) are robust search and optimization techniques. Unlike gradient-based methods, GAs do not need further information on the objective function besides the result, which is easy to compute. GAs have been used successfully in various fields. In recent years, numerous studies have shown that GAs can efficiently solve optimal portfolio problems in finance. For example, Chang et al.[14] and Yang[15] used GAs to solve the mean-variance portfolio optimization problem and Tsao[16], Baixauli-Soler et al.[17], and Ranković et al.[18] solved mean-VAR problems using GAs. However, to the best of our knowledge, no study has applied GAs to CPT to solve the portfolio choice problem.

The contributions of our study are as follows. First, we develop a CPT model for optimal portfolio selection; second, we couple the bootstrap method for the evaluation of investment portfolio scenarios with a genetic algorithm to determine the optimal solution; third, we introduce an Adaptive Real-Coded Genetic Algorithm (ARCGA) to find the optimal solution for CPT investors who want to allocate investments among various financial assets; and finally, we present an empirical comparison of different choices under various reference points in the CPT model.

The rest of this paper is organized as follows. In Section 2, we briefly discuss CPT investors and present an objective model for portfolio selection under CPT. In Section 3, we introduce a method to solve the objective function by coupling scenario generation with a GA. Section 4 presents the empirical results. Finally, conclusions are provided in Section 5.

2 Objective Function for CPT Investors

First, we review the framework of CPT. The key elements of CPT are as follows:

- CPT investors appraise their investment according to its relative value with respect to a reference point, which separates the investment into gains and losses.
- CPT investors display different behaviors with respect to gains and losses. Thus, the value function is concave with respect to gains and convex with respect to losses.
- CPT investors are more sensitive to losses than to gains.
- CPT investors apply excessive weights to small probabilities and underestimate the weights of large probabilities using a weighing function, which is a nonlinear transformation of objective probability.

Assumption 1 CPT investors are more concerned with return than with the final wealth.

Assumption 2 No transaction costs exist in the financial market, and CPT investors do not borrow cash for investment.

Consider a set of investment assets $i = 1, 2, \dots, n$. At the end of a certain holding period these assets generate random returns $\boldsymbol{R} = (R_1, R_2, \dots, R_n)^{\mathrm{T}}$. The CPT investors want to apportion their budget to these assets by deciding on a specific allocation $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^{\mathrm{T}}$, $x_i \geqslant 0$ (no short sales permitted) and $\sum_{i=1}^{n} x_i = 1$ (basic budget constraint). Using the vector $\mathbf{1} = (1, 1, \dots, 1)^{\mathrm{T}}$, we may write the basic budget constraints in vector form as

$$X = \{\boldsymbol{x} : \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1, \boldsymbol{x} \geqslant 0\}.$$

In the next horizon period the uncertain return of the portfolio is denoted by $R_p = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{R} = \sum_{i=1}^{n} x_i R_i$. This equation indicates that the current selection may affect future investment returns.

Let r_f be the value of a (scalar) reference point that separates gains and losses. We define the deviation Y from the reference level by

$$Y = R_p - r_f \tag{1}$$

Obviously, Y is treated as a random variable. We suppose that $Y_1, \ldots, Y_m$ form a random sample with some distribution. Let Y_1 denote the smallest value in the random sample, Y_2 denote the next smallest value, and so on. In this way, Y_m denotes the largest value in the sample, and Y_{m-1} denotes the next largest value. Thus, the random variables $Y_1, \ldots, Y_m$ are the order statistics of the sample. Let $y_1, \ldots, y_i, y_0, y_{i+1}, \ldots, y_m$ denote the values of the order statistics for an arbitrarily sample with probability $p_1, \ldots, p_i, p_0, p_{i+1}, \ldots, p_m$, respectively. If all values of $y_1, \ldots, y_m$ are nonzero, then $y_0 = 0$ with probability $p_0 = 0$ is inserted; otherwise, $y_0 = 0$ exists with probability $p_0 \neq 0$. (In fact, the presence or absence of this zero value in the results has no effect on the CPT value, as shown later in this paper.) According to Tversky and Kahneman[7], the CPT investors evaluate the investment

$$(y_1, p_1; \ldots; y_i, p_i; y_0, p_0; y_{i+1}, p_{i+1}; \ldots; y_m, p_m) \tag{2}$$

As mentioned EUT assumes that the investors are risk-averse in the gains. However, CPT assumes that investors express the outcomes as deviations from some reference point and response, and are more sensitive to losses than to gains. The value function v is defined by Tversky and Kahneman[7] as

$$v(y) = \begin{cases} v^+(y), & y \geqslant 0; \\ -\lambda v^-(-y), & y < 0 \end{cases} \tag{3}$$

where $v^+(y) = y^\alpha$ with $0 < \alpha < 1$, $v^-(y) = y^\beta$ with $\alpha \leqslant \beta < 1$ and $v(0) = v^+(0) = v^-(0) = 0$. Obviously, the functions $v^+(\cdot)$ and $v^-(\cdot)$ are increasing, twice differentiable, invertible, and concave[19]. Tversky and Kahneman[7] suggested the value of $\alpha = \beta = 0.88$ and $\lambda = 2.25$. In fact, for $\alpha, \beta < 1$, the S-shaped value function exhibits risk aversion over gains and risk seeking over losses. The parameter λ captures loss aversion if we assume that investors consider losses to be more than twice as important as gains.

The parameters of value function define the degree of risk aversion with respect to gains, the degree of risk preference with respect to losses, and the degree of loss aversion. The parameter α represents risk aversion with respect to gains and the parameter β represents risk preference with respect to losses. Regarding gains, higher values of α signify that CPT investors are becoming increasingly risk-averse. A large value β implies that the degree of risk-seeking is higher with respect to loss. The parameter λ represents the loss aversion: the higher the value of λ, the more loss-averse the CPT investors. As shown in Fig. 1, the dash-dot curve corresponds to $\alpha = 0.2$, $\beta = 0.4$, $\lambda = 1$; the dotted curve corresponds to $\alpha = 0.6$, $\beta = 0.9$, $\lambda = 2$; the solid curve corresponds to $\alpha = 0.88$, $\beta = 0.88$, $\lambda = 2.25$; and the dashed line corresponds to $\alpha = 1$, $\beta = 1$, $\lambda = 2.25$.

CPT investors do not weigh the outcomes according to objective probabilities. Moreover, the weighting functions have different parameters over the domains of gains and losses, denoted by $w^+(\cdot)$ and $w^-(\cdot)$, respectively. Tversky and Kahneman[7] proposed the following functions:

$$w^+(P) = \frac{P^\gamma}{[P^\gamma + (1-P)^\gamma]^{1/\gamma}} \tag{4}$$

$$w^-(P) = \frac{P^\delta}{[P^\delta + (1-P)^\delta]^{1/\delta}} \tag{5}$$

where $w^+ : [0,1] \to [0,1]$ and $w^- : [0,1] \to [0,1]$ are non-decreasing and differentiable with $w^+(0) = w^-(0) = 0$ and $w^+(1) = w^-(1) = 1$. The parameters of weighting functions define the degree of distortion to the objective probabilities. The smaller the values of γ and δ, the greater the degree of distortion. As shown in

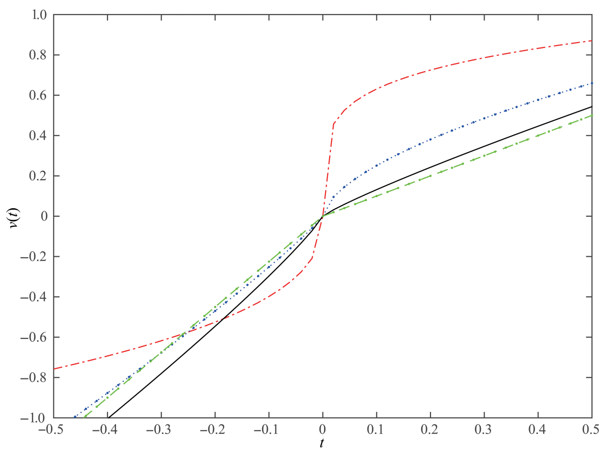


Fig. 1 Value functions.

Fig. 2, the dash-dot curve corresponds to $\gamma = 0.4$; the dotted curve corresponds to $\gamma = 0.61$; the solid curve corresponds to $\gamma = 0.69$; and the solid line corresponds to $\gamma = 1$.

Ingersoll[20] showed that $0.28 < \gamma, \delta < 1$ ensures that $w^+(\cdot)$ and $w^-(\cdot)$ are all increasing functions. Tversky and Kahneman[7] estimated that $\gamma = 0.61$, $\delta = 0.69$. For the case of $\gamma = \delta = 1$, the weighting functions have the following linear form:

$$w^+(P) = w^-(P) = P \tag{6}$$

The original version of the PT suffers from potential violations of first-order stochastic dominance. To avoid this violation, we apply CPT[7]. Although the probabilities are weighted in PT, the cumulative probabilities are weighted in CPT:

$$\pi_i = \begin{cases} \pi_i^+ = w^+(p_i + \cdots + p_m) - \\ \qquad w^+(p_{i+1} + \cdots + p_m), \\ \pi_i^- = w^-(p_1 + \cdots + p_i) - \\ \qquad w^-(p_1 + \cdots + p_{i-1}) \end{cases} \tag{7}$$

where p_i denotes outcome y_i $(i = 1, \ldots, m)$.

The CPT value of the investment for stocks is expressed as

$$V(x) = \sum_{i=1}^{m} \pi_i \cdot v(y_i(x)) \tag{8}$$

CPT investors make portfolio choices by maximizing their CPT value; that is, CPT investors determine their investments by maximizing the value of Eq. (8). Thus, we propose the following objective function:

$$\begin{aligned} &\max V(x), \\ &\text{s.t.} \sum_{i=1}^{n} x_i = 1, \\ &\qquad x_i \geqslant 0, i = 1, \ldots, n \end{aligned} \tag{9}$$

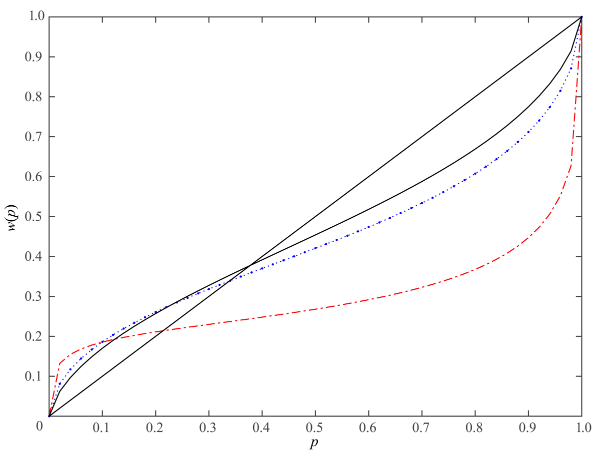


Fig. 2 Probability weighting functions.

3 Proposed Approach: Bootstrap Method + Genetic Algorithm

3.1 Bootstrap method

A critical problem in portfolio selection is the description of a random investment portfolio return. In general, the problem is addressed by a set of random returns or their expected return. A set of scenarios can be generated by various methods, such as a historical approach, bootstrap method, or Monte Carlo simulation. In this paper, we use previous observations of asset returns to generate the expected returns through the bootstrap method. That is, we combine historical data with the bootstrap technique to simulate the required number of sample data. However, determining the parameters of the returns is difficult. Thus, we use the Non-Parametric Bootstrap (NPB) method.

Consider a strictly stationary time series of the i-th investment asset held for T time periods. This time series is expressed by $\boldsymbol{R}_i = (R_{i,1}, \ldots, R_{i,T})$, which means that the joint probability distribution of $(R_{i,1}, \ldots, R_{i,T})$ does not change when shifted in time. As mentioned, finding the probability distribution of $\boldsymbol{R}_i$, which is denoted by F_i, is difficult. Let $\theta(F_i)$ be a parameter of interest such as the mean, median, or standard deviation of F_i. Let $\hat{\theta}(\boldsymbol{R}_i)$ be an estimator of $\theta(F_i)$ computed using observations $\boldsymbol{R}_i$. Here, we focus on the mean of the returns.

The bootstrap method does not require any parametric assumption on F_i and can use smaller sample sizes as a formalization of the resampling procedure for statistical inference. Let the observed data take the values $\boldsymbol{r}_i = (r_{i,1}, \ldots, r_{i,T})$. The mean return of one asset is $\bar{r}_i = \frac{1}{T}\sum_{j=1}^{T} r_{i,j}$ and the expected portfolio return at time index $T + 1$ is expressed as $r_p = \sum_{i=1}^{n} x_i \bar{r}_i$. Next, draw T sample data $\boldsymbol{r}_i^* = (r_{i,1}^*, \ldots, r_{i,T}^*)$ from $(r_{i,1}, \ldots, r_{i,T})$ by using bootstrap method. The mean $\bar{r}_i^* = \frac{1}{T}\sum_{i=1}^{T} r_{i,T}^*$ can be computed from $\boldsymbol{r}_i^*$. Note that the number of sampled data in the bootstrap method is equal to the number of observed data, and no permutation occurs because we have performed random sampling without replacement.

By repeating this process S times, we obtain the following scenario matrix:

$$\boldsymbol{R}_s = \begin{bmatrix} \bar{r}_1^{*1} \bar{r}_1^{*2} \cdots \bar{r}_1^{*S} \\ \bar{r}_2^{*1} \bar{r}_2^{*2} \cdots \bar{r}_2^{*S} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ \bar{r}_n^{*1} \bar{r}_n^{*2} \cdots \bar{r}_n^{*S} \end{bmatrix} \tag{10}$$

A vector $\boldsymbol{R}_p^S = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{R}_s = (r_p^1, \ldots, r_p^s)$ is obtained from the scenario matrix through multiplication by a set of decision-making vectors. If the elements in $\boldsymbol{R}_p^S$ are sorted in ascending order of value, we obtain a result similar to Formula (2). Finding the set of optimal decision-making vectors that maximize the objective function is discussed later in this paper.

Selecting the best bootstrap technique for estimating the mean depends on whether the observed data $\boldsymbol{R}_i$ are assumed to be independent or dependent, which can be difficult to identify. Scenario generation should be considered to encompass all future possibilities. We will refer to the bootstrap method for independent data as the Standard Bootstrap (SB) technique and that for dependent data as the Moving Block Bootstrap (MBB) technique and Non-overlapping Block Bootstrap (NBB) technique.

3.1.1 Standard bootstrap

The SB method is implemented by sampling the data randomly with replacement, i.e., observed data can be resampled with a constant probability $1/T$. We can derive $\boldsymbol{R}_i^* = (R_{i,1}^*, \ldots, R_{i,T}^*)$ from $(R_{i,1}, \ldots, R_{i,T})$. For a more comprehensive review of the SB technique, see Kouwenberg and Zenios[13].

3.1.2 Moving block bootstrap methods

Note that $\boldsymbol{R}_i = (R_{i,1}, \ldots, R_{i,T})$ is treated as a series of outcomes with probability $1/T$. However, this assumption is not always valid, especially for financial time series. Singh[21] showed that the SB technique[11] failed to produce valid approximations in the presence of dependent data. To overcome the limitations of the SB technique for dependent financial time series data, Hall[22] suggested resampling the data using blocks of observed data instead of individual data, and Kunsch[23] proposed resampling blocks of observations at a time (see also Ref. [24]). The dependence structure of the random variables at short lag distances is preserved by keeping the neighboring observations together within the blocks. As a result, resampling blocks allows one to carry this information over to the bootstrap variables. A similar method was called the "moving block bootstrap"[25].

We suppose that $\boldsymbol{R}_i = (R_{i,1}, \ldots, R_{i,T})$ is the observed financial time series of the i-th assets. Let ℓ be an integer satisfying $1 \leqslant \ell < T$. We define the overlapping blocks $\boldsymbol{B}_{i,1}, \ldots, \boldsymbol{B}_{i,M}$ of length ℓ as

$$\begin{aligned} \boldsymbol{B}_{i,1} &= (R_{i,1}, \ldots, R_{i,\ell}), \\ \boldsymbol{B}_{i,2} &= (R_{i,2}, \ldots, R_{i,\ell+1}), \\ &\ldots \\ \boldsymbol{B}_{i,M} &= (R_{i,T-\ell+1}, \ldots, R_{i,T}) \end{aligned} \tag{11}$$

where $M = T - \ell + 1$. To generate the MBB samples, we select $b = T/\ell$ blocks at random with replacement from $(\boldsymbol{B}_{i,1}, \boldsymbol{B}_{i,2}, \ldots, \boldsymbol{B}_{i,M})$. As each resampled block has ℓ elements, concatenating the elements of the b resampled blocks serially yields $T = b \cdot \ell$ bootstrap observations. Some typical choices of ℓ are $\ell = CT^{1/k}$, for $k = 3, 4$, where $C \in \mathbf{R}$ is a constant[26].

3.1.3 Non-overlapping block bootstrap

Another bootstrap technique involves resampling from non-overlapping blocks to generate the bootstrap observations[27]. Suppose that ℓ is an integer in $[1, T]$ (note that NBB is equivalent to SB when ℓ=1). Let $N = T/\ell$ and generate NBB samples by selecting N blocks at random with replacement from the collection $(\bar{\boldsymbol{B}}_{i,1}, \bar{\boldsymbol{B}}_{i,2}, \ldots, \bar{\boldsymbol{B}}_{i,N})$, where

$$\begin{aligned} \bar{\boldsymbol{B}}_{i,1} &= (R_{i,1}, \ldots, R_{i,\ell}), \\ \bar{\boldsymbol{B}}_{i,2} &= (R_{i,\ell+1}, \ldots, R_{i,2\ell}), \\ &\ldots \\ \bar{\boldsymbol{B}}_{i,N} &= (R_{i,(N-1)\ell+1}, \ldots, R_{i,T}) \end{aligned} \tag{12}$$

Examining the characteristics of the NBB estimators is easier than those of the MBB estimators of a population parameter because NBB uses non-overlapping blocks. However, the NBB estimators typically have higher MSEs for a given block size ℓ compared with their MBB counterparts[28].

3.2 Adaptive real-coded genetic algorithm technique

In CPT, the S-shaped value function is non-concave and non-smooth; moreover, the weighting functions are nonlinear functions. He and Zhou[9] proved that the optimal portfolio allocation in the CPT context is a non-convex and non-concave problem. Therefore, traditional optimization methods face significant difficulty in dealing with our proposed model. GAs offer a number of advantages over traditional optimization methods. The ability of GAs enables finding a solution for difficult problems with non-convex and non-concave solution spaces.

The concept of GAs, which can be described as an "intelligent" probabilistic search algorithm, was developed by Holland[29] in the 1960s and 1970s. The idea was inspired by the evolutionist theory explaining the origin of species. Being a population-based approach, GA is well suited to solve the optimization problem of CPT. The ability of GA to simultaneously search various regions of a solution space enables finding a diverse set of solutions for difficult problems with non-convex, discontinuous, and multimodal solution spaces. GA initially has a population consisting of a set of vector chromosome, which are generated randomly to explore the solution space of a problem.

Traditionally, the genes in a chromosome are represented by binary coded strings. However, RCGA has used real-valued genes to solve continuous optimization problems[30,31]. The use of real-valued genes allows for improved adaption to numerical optimization of continuous problems. RCGA is able to exploit the gradation of the functions with continuous variables, and to avoid the Hamming cliff effect suffered by a Binary Coded Genetic Algorithm (BCGA). The convergence speed of RCGA is good, and unlike BCGA, RCGA involves no coding and decoding processes. In RCGA, each chromosome represents one decision vector and every gene corresponds to the weight of one asset.

We developed an ARCGA, with adaptive mechanism that improves the efficiency of the operators through the evolutionary process. This adaptive method solves the problem of portfolio choice under CPT within the feasible operating region. Two important issues are involved in RCGA. One is selection pressure, without which the search process would be a random algorithm. The effective selection pressure ensures that chromosomes with higher fitness value have a higher chance of surviving under crossover and mutation. The second is population diversity, which produces genotype of the offspring that differ from those of their parents. A highly diverse population can increase the probability of exploring the global optimum and prevent the premature convergence to a local optimum[32,33]. RCGA involves a tradeoff between selection pressure and population diversity because the two factors act against one another. Therefore, they should be controlled to ensure optimal balance. Thus, in this paper, we propose a technique for increasing the selection pressure and an adaptive method that retains the balance between the selection pressure and population diversity processes. The pseudocode of ARCGA is shown in Algorithm 1, where $P(g)$ represents the parents, $M(g)$ represents the mating pool, $Q'(g)$ represents the offspring from $M(g)$ after the crossover operation, $Q(g)$ represents the offspring from $Q'(g)$ after the mutation operation, and g denotes the generation.

3.2.1 Generation of initial population

The genes of a chromosome are real numbers between 0 and 1 representing the weights invested in the assets under CPT. The most popularly used initialization method is the random generation in which every datum is generated uniformly in the range [0,1] in a random, independent manner. $w_i, i = 1, \ldots, n$, is represented as a number generated randomly in the initialization phase.

If the sum of these data is greater than 1, the constraint in Formula (9) will be violated. To overcome this problem, the portfolio weights are obtained by normalizing w_i as follows:

$$x_i = w_i / \sum_{i=1}^{n} w_i \tag{13}$$

where x_i represents the weight invested in asset i after normalization.

The described procedure is repeated η times, so that we can obtain η solutions to form the first population $\boldsymbol{x}^{g=1} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\eta\}$. We hope that $\boldsymbol{x}^g$ will evolve and gradually converge to $\boldsymbol{x}^*$ as the evolutionary process

Algorithm 1 Pseudocode of ARCGA

Begin
- Let $g := 1$;
- Initialize population $P(g)$;

while *not termination condition* **do**
- Evaluate the fitness value of each chromosome;
- Sort the chromosomes in the descending order of its fitness values;
- Calculate the adaptive parameters based on fitness value;
- Create a mating pool $M(g)$ from the population $P(g)$ by operator of selection;
- Create offspring-crossed $Q'(g)$ from $M(g)$ by operator of crossover;
- Create offspring $Q(g)$ from $Q'(g)$ by operator of mutation;
- Select the best chromosome from the population $P(g)$ according to fitness value and randomly replace one chromosome in $Q(g)$;

$P(g+1)$ = normalized $Q(g)$;
- $g := g + 1$;

end

end

continues.

3.2.2 Evaluation

The fitness value of each chromosome is measured by an objective function. We evaluate the fitness values of chromosomes in $P(g)$ with Formula (9) and order f_j^g, $j = 1, \ldots, \eta$, which are the fitness values in every generation.

3.2.3 Selection

Various methods are used to apply the selection operator in RCGA. Truncation selection is known as the most efficient form of directional selection[34]. This approach ranks all chromosomes according to their fitness values and selects the best $h\%$ as parents. Truncation selection has been used extensively in evolution strategies[35,36]. Other popular methods include $(\mu + \lambda)$ selection or (μ, λ) selection, where μ is the number of parents and λ is the number of offspring. The top μ individuals form the next generation, with the selection being from parents and children in the $(\mu + \lambda)$ case and from children only for (μ, λ). Typically, λ is one or two times μ[37]. Top-N selection is employed to select the N best chromosomes from the population by some scholars[38]. In addition, the "replace worst" strategy replaces the population if the new chromosome is better than the existing worst chromosome. Goldberg and Deb[39] showed that higher selection pressure exists in a population that deletes the worst chromosome even if others are selected at random.

Inspired by these ideas, we introduce a selection operator called Duplicated Top-N Selection (DTNS), in which the best κ chromosomes are copied twice to the mating pool. This approach ensures that the best κ chromosomes are retained and the κ worst chromosomes are replaced. The remaining chromosomes are placed in the mating pool unchanged. In this manner, the best chromosomes in the population have more opportunities to be chosen and the worst chromosomes are eliminated from the population. This condition leads to improved convergence in terms of the quality of chromosomes and computation time. In our work, the number of κ is obtained by rounding operation of the population size multiplied by $h\%$.

3.2.4 Crossover

We use the typical arithmetical crossover of each parent to produce two offspring in the crossover step. Michalewicz[31] suggested that the arithmetical crossover operator is the best option for RCGA. We assume that chromosomes $x = (x_1, \ldots, x_n)$ and $x' = (x'_1, \ldots, x'_n)$ have been selected for crossover. The offspring are given as follows:

$$\hat{x}_i = \xi x_i + (1-\xi)x'_i, \quad \hat{x}'_i = \xi x'_i + (1-\xi)x_i \tag{14}$$

where ξ is a uniform random number in $[-0.5, 1.5]$.

3.2.5 Mutation

Mutation is generally applied at the gene level and reintroduces genetic diversity to the population, which helps the search to escape from local optima. We apply a non-uniform mutation operator. We suppose that $x = (x_1, \ldots, x_i, \ldots, x_n)$ is a chromosome, and that $x_i \in [a_i, b_i]$, where a_i and b_i are respectively the lower and upper bounds of x_i, is the element to be mutated in generation g. The resulting chromosome is $x^{'} = (x_1, \ldots, x'_i, \ldots, x_n)$, where x'_i is obtained by

$$x'_i = \begin{cases} x_i + \Delta(g, b_i - x_i), & \text{if } \gamma = 0; \\ x_i - \Delta(g, x_i - a_i), & \text{if } \gamma = 1 \end{cases} \tag{15}$$

with γ being a random number that takes either value zero or one, and

$$\Delta(g, l) = l(1 - r^{(1-\frac{g}{G})^\tau}) \tag{16}$$

where r is a random number from the interval [0,1], G is the maximal generation number, and τ is a user-selected parameter that determines the degree of non-uniformity[31]. This function provides a value in the range $[0, l]$ such that the probability of returning a number close to zero increases as g increases. As a result, this operator performs a uniform search in the initial stages (when g is small) and a more local search in the final stages[40,41].

3.2.6 Elitist method

The main disadvantage of selection is that the best chromosome in each generation may not be preserved. The elitist method[42] can isolate the best chromosome and transfer it to the next generation. In this manner, the best chromosome obtained during the entire process of RCGA is guaranteed to survive. Rudolph[43] showed that convergence to the global optimum is not an inherent property of the Canonical Genetic Algorithm (CGA), but rather is a consequence of the algorithmic trick of keeping track of the best solution found over time. The major drawback of this approach is its tendency to get stuck in a local extremum[18]. Obviously, the combination of elitist and adaptive method can prevent the aforementioned limitations.

3.2.7 Stopping criterion

The algorithm terminates when the following stopping criteria is satisfied.

$$g = G \tag{17}$$

where g is the current number of generations and G denotes the maximum number of generations, which is a pre-fixed threshold.

3.2.8 Parameter selection

Although RCGA has many advantages over BCGA, it can often suffer from premature convergence due to lack of population diversity. Conversely, it can also suffer from slow convergence[40]. To solve these problems, RCGA has been hybridized with other optimization methods[44,45]. For example, some researchers have worked to improve the crossover operators of the RCGA[46,47]. Population diversity in RCGA is important throughout the search process, not only in the initial stages, because it determines how the set evolves with each generation to explore the search region. Although a number of rules have been suggested in the literature to improve the population diversity, they are generally tailored toward solving certain problems[46]. Subbaraj et al.[48,49] developed a self-adaptive real-coded genetic algorithm to solve the combined heat and power economic dispatch problem. However, p_c and p_m were assigned constant values in their paper.

In essence, RCGA uses the values of p_c and p_m to balance the capacity to converge to an optimum (local or global) after locating the region containing the optimum and the capacity to explore new regions of the solution space in search of the global optimum[50]. The probabilities of crossover and mutation are varied depending on the fitness values of the solutions. This condition encourages the exploration of the search space because of the accelerating gene disruption, and helps to prevent premature convergence. Thus, in our study, the adaptive probabilities of crossover and mutation are used to maintain diversity in the population and sustain the convergence capacity of RCGA. In general, GA use values p_c in the range of [0.5, 1] and p_m in the range of [0.001, 0.05].

Srinivas and Patnaik[50] used the $f_{\max}^g - \overline{f}$ to detect the convergence of GA, and varied p_c and p_m depending on the value of this metric. Scholars are increasingly using adapting crossover and mutation probabilities instead of fixed values[51–54]. Based on the work of Refs. [50, 51], the following expressions

$$p_c^g = k_1(f_{\max}^g - \overline{\overline{f}}^g)/(f_{\max}^g - \overline{f}^g),$$

$$p_c^g = k_2(f_{\max}^g - \overline{\overline{f}}^g)/(f_{\max}^g - \overline{f}^g) \tag{18}$$

are used.

We set $k_1 = 1.0$ and $k_2 = 0.5$. $f_{\max}^g$ and $\overline{f}^g$ represent the maximum fitness value and the average fitness value of the population at each generation, respectively. $\overline{\overline{f}}^g$ is the average of the fitness values that are greater than $\overline{f}^g$. We restricted p_c and p_m to the ranges recommended ranges. The adverse effect caused by poor chromosomes can be avoided by calculating the difference in fitness values. This condition is clarified in the degree of convergence between the chromosomes with larger fitness values in the population.

4 Numerical Computation Experiments

4.1 Parameters of the CPT investors and data

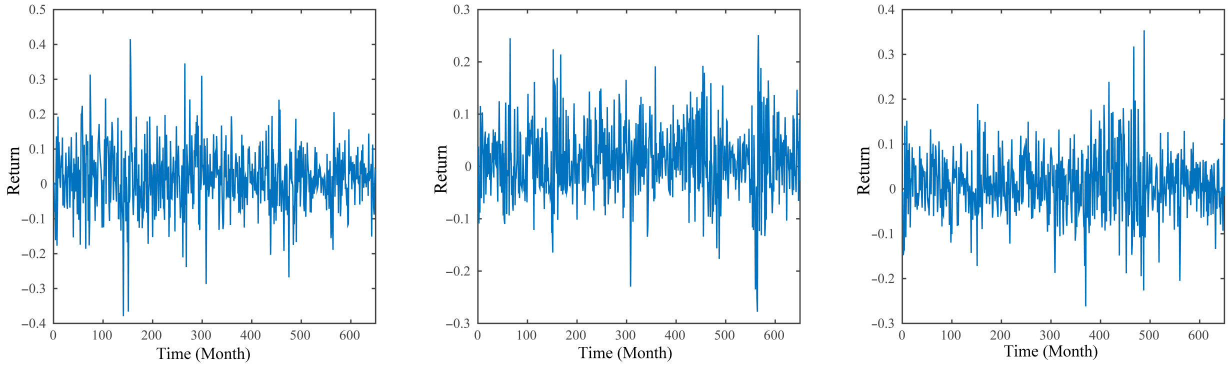
Five parameters ($\alpha, \beta, \gamma, \delta,$ and λ) describe the investors' objective function. We adopt the values provided by Ref. [7], as listed in Table 1.

Choosing the historical period is important for generating scenarios, but no rule exists for determining the length of the time period. Given the monthly return, we select a longer interval as soon as possible. In this paper, we consider portfolios composed of Walt Disney (DIS), General Electric (GE), and International Business Machines (IBM) stocks, and use the adjusted monthly closing price for the period January 2, 1962 to April 1, 2016 (dividends are not included). A total of 651 observations are reported (data obtained from Yahoo! Finance).

The difference between the log and arithmetic returns is negligible for a one-day horizon. However, some error typically occurs in the portfolio log-returns if we neglect the conversion between the log-return and arithmetic return over much longer horizons. As a result, the difference between the log and arithmetic returns is generally considered, and we use the arithmetic returns and obtain 650 monthly returns for each of the three stocks considered in this study, as shown in Fig. 3.

Table 1 Parameters of objective function.

Parameter	Value
α	0.88
β	0.88
γ	0.61
δ	0.69
λ	2.55



(A) DIS (b) GE (c) IBM

Fig. 3 Monthly returns for three stocks

The foundation of time series analysis is stationarity. First, we test for the unit root of stock returns, which tells us whether a time series variable is non-stationary and possesses a unit root. The results of ADF tests show that the returns on each of the three stocks reject the null hypothesis at the 1% significance level, i.e., no unit roots exist in the three sequences, and they can be considered as stationary sequences (see Table 2). We now analyze the descriptive statistics for the rate of return. Table 2 presents the basic statistics of the sample data. Each stock exhibits positive expected returns. Interestingly, the standard deviation of the GE returns is less than that of the IBM stocks, but GE's stock has a higher expected return than IBM's and is preferred by rational people, as discussed later in this paper. Furthermore, in the IBM stock, the skewness is small, indicating that the distributions are largely symmetric. Each stock exhibits larger kurtosis values and has a heavy-tailed distribution. The JB test shows that none of the stocks is normal at the 1% significance level. The normality can also be tested using a Q-Q plot. If the data are normally distributed, then the quantiles are on a straight line. As shown in Fig. 4, a significant deviation from the straight line occurs in the tails for each stock, especially in the lower tail, indicating that the distribution of standardized returns is more heavy-tailed than the normal distribution.

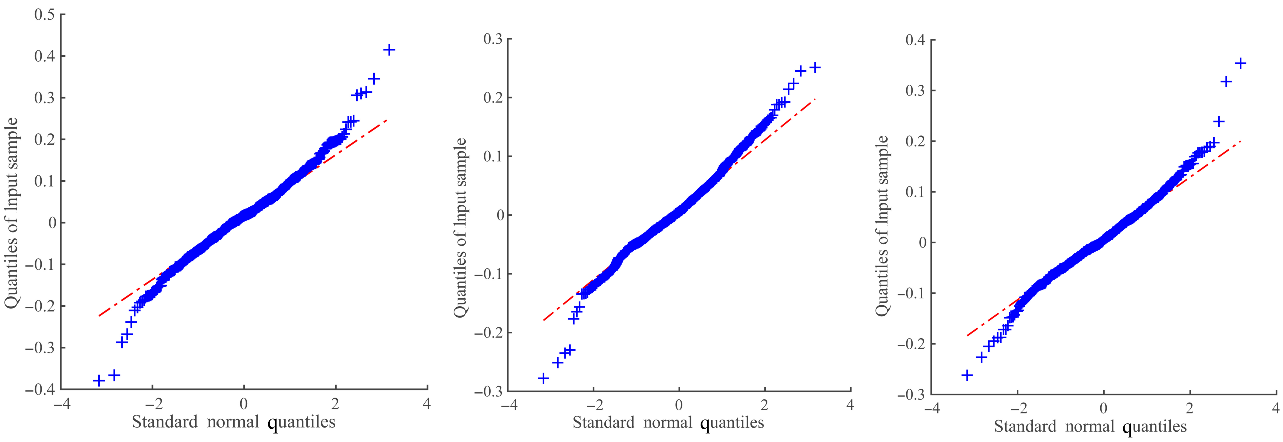
Table 2 Descriptive statistics and test results for sample data.

Company	Mean	Std.Dev	Skewness	Kurtosis	JB test	ADF test
DIS	0.0155	0.0895	0.0042	4.9735	1	0.001
GE	0.0106	0.0686	0.0540	4.2515	1	0.001
IBM	0.0089	0.0696	0.2270	4.8843	1	0.001

4.2 Computational experiments

All algorithms are programmed in MATLAB 2014b and run on a 2.6 GHz and 8 GB RAM Apple MacBook Pro computer.

First, we set the block length to 10 for MBB and NBB. Second, we generated 650 data at random with



(a) DIS (b) GE (c) IBM

Fig. 4 Q-Q plots for three stocks.

replacement using the SB technique and 65 blocks at random with replacement using the MBB and NBB techniques. Third, we repeated these procedures 10 000 times to produce three scenario matrices, i.e., SB, MBB, and NBB matrix, in which each row corresponds to one stock and each column represents every result of a single simulation. The relevant statistics for the simulation samples are presented in Table 3. No significant change occurs in the mean compared with the original data. However, the standard deviation, skewness, and kurtosis of the sample data for the three stocks have changed dramatically. All of the sample data passed the JB test for normality at the 1% significance level.

Reference points are an important concept in CPT, but limited research has been conducted on their impact on investment behavior. Several reference point scenarios have been discussed by Pirvu and Schulze[55], but they did not provide further details.

As mentioned, each column of the scenario matrices represents every result of a single simulation. We can obtain a CPT value by coupling a solution generated by ARCGA with one scenario matrix. By continuously generating solutions, we identify the optimal solution that maximizes the objective function. The ARCGA was used to produce different initial populations of sizes 50, 100, and 150, and the algorithm was executed 100 times. All 100 results were the same for each population size. The results are presented in Table 4.

The experimental results show that the proposed ARCGA is an effective and stable algorithm for the given objective function. For comparison, an exhaustive method was used to find the global optima. The results show that these solutions are exactly equal to those generated by the proposed ARCGA.

Generally, the various scenarios have a significant influence on investment behavior under CPT, and the MBB scenario has the most significant effect.

Table 3 Descriptive statistics and test results for simulation data.

Method	Company	Mean	Std.Dev.	Skewness	Kurtosis	JB test
SB	DIS	0.0154	0.0035	-0.0030	2.9830	0
	GE	0.0106	0.0027	0.0143	2.9401	0
	IBM	0.0089	0.0027	0.0317	2.9334	0
NBB	DIS	0.0155	0.0032	-0.0157	3.0165	0
	GE	0.0107	0.0026	-0.0120	2.9701	0
	IBM	0.0089	0.0025	-0.0011	2.9803	0
MBB	DIS	0.0160	0.0011	0.0207	2.9612	0
	GE	0.0107	0.0009	-0.0148	3.0711	0
	IBM	0.0093	0.0009	0.0517	2.9895	0

Table 4 CPT value, portfolio return and optimal solution with different reference point.

Method	r_f	CPT Value	Return	DIS*	GE*	IBM*
SB	0.003	0.0194	0.0154	1	0	0
	0.004	0.0165	0.0143	0.8047	0.0913	0.1040
	0.005	0.0135	0.0131	0.6050	0.1814	0.2136
	0.006	0.0080	0.0123	0.4442	0.3102	0.2455
NBB	0.003	0.0191	0.0151	0.9075	0.0925	0
	0.004	0.0169	0.0145	0.7914	0.2086	0
	0.005	0.0146	0.0139	0.6755	0.3245	0
	0.006	0.0114	0.0128	0.4708	0.4418	0.0874
MBB	0.003	0.0215	0.0160	1	0	0
	0.004	0.0200	0.0160	1	0	0
	0.005	0.0185	0.0160	1	0	0
	0.006	0.0169	0.0160	1	0	0

Note: r_f denotes the reference point value, Return denotes the portfolio return, and superscript $*$ denotes the optimal investment ratio for that stock

Regardless of how the reference point changes, CPT investors always put all of their money into DIS under the MBB scenario.

Our second observation is that CPT investors change their investment ratio significantly under SB and NBB as the value of r_f changes. Furthermore, as r_f increases, the CPT values decrease significantly and the returns diminish, thereby demonstrating numerically that high expectations lead to great disappointment.

5 Conclusion

In this study, we formulated an optimal portfolio selection model for a single period under CPT. Considering that the objective function is non-convex, non-concave, and non-smooth, we proposed an ARCGA to determine the optimal solution. We compared the portfolio choices of CPT investors based on various bootstrap techniques for scenario generation. Computational experiments show that the ARCGA can efficiently and stably solve the portfolio problem for CPT investors. In addition, this study is the first to consider the effect of different reference points on investment behavior under CPT.

Future research aims to extend the optimization models to incorporate risk constraints. Risk management is currently a major topic, so we will consider VaR and CVaR, which are popular tools for risk management in finance.

References

[1] H. Markowitz, Portfolio selection, *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[2] M. Allais, Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'école américaine. *Econometrica: Journal of the Econometric Society*, pp. 503–546, 1953.

[3] D. Ellsberg, Risk, ambiguity, and the savage axioms, *The Quarterly Journal of Economics*, vol. 75, no. 4, pp. 643–669, 1961.

[4] D. Kahneman and A. Tversky, Prospect theory: An analysis of decision under risk, *Econometrica: Journal of the Econometric Society*, vol. 47, no. 2, pp. 263–292, 1979.

[5] N. C. Barberis, Thirty years of prospect theory in economics: A review and assessment, *National Journal of Economic Perspectives*, vol. 27, no. 1, pp. 173–196, 2013.

[6] J. Quiggin, A theory of anticipated utility, *Journal of Economic Behavior & Organization*, vol. 3, no. 4, pp. 323–343, 1982.

[7] A. Tversky and D. Kahneman, Advances in prospect theory: Cumulative representation of uncertainty, *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.

[8] S. Benartzi and R. H. Thaler, Myopic loss aversion and the equity premium puzzle, *The Quarterly Journal of Ecnomics*, vol. 110, no. 1, pp. 73–92, 1995.

[9] X. D. He and X. Y. Zhou, Portfolio choice under cumulative prospect theory: An analytical treatment, *Management Science*, vol. 57, no. 2, pp. 315–331, 2011.

[10] S. P. Bradley and D. B. Crane, A dynamic model for bond portfolio management, *Management Science*, vol. 19, no. 2, pp. 139–151, 1972.

[11] B. Efron, Bootstrap methods: Another look at the jackknife, *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.

[12] B. T. Efron, An introduction to the bootstrap, *Monograps on Statistics and Applied Probability*, vol. 57, no. 57, 1993.

[13] R. Kouwenberg and S. A. Zenios, Stochastic programming models for asset liability management, in *Handbook of Asset and Liability Management*, S. A. Zenios and W. T. Ziemba, eds. vol. 1, 2006, pp. 253–303.

[14] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, Heuristics for cardinality constrained portfolio optimisation, *Computers & Operations Research*, vol. 27, no. 13, pp. 1271–1302, 2000.

[15] X. Yang, Improving portfolio efficiency: A genetic algorithm approach, *Computational Economics*, vol. 28, no. 1, pp. 1–14, 2006.

[16] C.-Y. Tsao, Portfolio selection based on the mean–var efficient frontier, *Quantitative Finance*, vol. 10, no. 8, pp. 931–945, 2010.

[17] J. S. Baixauli-Soler, E. Alfaro-Cid, and M. O. Fernandez-Blanco, Mean-var portfolio selection under real constraints, *Computational Economics*, vol. 37, no. 2, pp. 113–131, 2011.

[18] V. Ranković, M. Drenovak, B. Stojanović, Z. Kalinić, and Z. Arsovski, The mean-value at risk static portfolio optimization using genetic algorithm, *Computer Science and Information Systems*, vol. 11, no. 1, pp. 89–109, 2014.

[19] C. Bernard and M. Ghossoub, Static portfolio choice under cumulative prospect theory, *Mathematics and Financial Economics*, vol. 2, no. 4, pp. 277–306, 2010.

[20] J. Ingersoll, Non-monotonicity of the tversky-kahneman probability-weighting function: A cautionary note, *European Financial Management*, vol. 14, no. 3, pp. 385–390, 2008.

[21] K. Singh, On the asymptotic accuracy of efron's bootstrap, *The Annals of Statistics*, vol. 9, no. 6, pp. 1187–1195, 1981.

[22] P. Hall, Resampling a coverage pattern, *Stochastic Processes and Their Applications*, vol. 20, no. 2, pp. 231–246, 1985.

[23] H. R. Kunsch, The jackknife and the bootstrap for general stationary observations, *The Annals of Statistics*, vol. 17, no. 3, pp. 1217–1241, 1993.

[24] P. Bühlmann and H. R. Künsch, The blockwise bootstrap for general parameters of a stationary time series, *Scandinavian Journal of Statistics*, vol. 22, no. 22, pp. 35–54, 1993.

[25] R. Y. Liu and K. Singh, Moving blocks jackknife and bootstrap capture weak dependence, *Exploring the Limits of Bootstrap*, vol. 225, p. 248, 1992.

[26] J.-P. Kreiss and S. Lahiri, Bootstrap methods for time series, *Time Series Analysis: Methods and Applications*, vol. 30, no. 1, 2012.

[27] E. Carlstein, The use of subseries values for estimating the variance of a general statistic from a stationary sequence, *The Annals of Statistics*, vol. 14, no. 3, pp. 1171–1179, 1986.

[28] S. N. Lahiri, Theoretical comparisons of block bootstrap methods, *Annals of Statistics*, vol. 27, no. 1, pp. 386–404, 1999.

[29] J. H. Holland, Adaptation in natural an artificial systems, 1975.

[30] F. Herrera, M. Lozano, and J. L. Verdegay, Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis, *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, 1998.

[31] Z. Michalewicz, *Genetic Algorithms+ Data Structures= Evolution Programs*. Springer Science & Business Media, 2013.

[32] K. Deb and M. Goyal, A combined genetic adaptive search (geneas) for engineering design, *Computer Science and Informatics*, vol. 26, no. 1, pp. 30–45, 1996.

[33] K. Deb and H.-G. Beyer, Self-adaptation in real-parameter genetic algorithms with simulated binary crossover, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 1999, pp. 172–199.

[34] J. F. Crow and M. Kimura, Efficiency of truncation selection, *Proceedings of the National Academy of Sciences*, vol. 76, no. 1, pp. 396–399, 1979.

[35] D. Thierens and D. Goldberg, Convergence models of genetic algorithm selection schemes, in *Parallel Problem Solving from Nature—PPSN III*, 1994, pp. 119–129.

[36] T. BDack, F. Hoffmeister, and H.-P. Schwefel, A survey of evolution strategies, in *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991, pp. 2–9.

[37] F. Hoffmeister and T. Bäck, *Genetic Algorithms and Evolution Strategies: Similarities and Differences*. Springer, 1991.

[38] P. J. Hancock, An empirical comparison of selection methods in evolutionary algorithms, in *Evolutionary Computing*. Springer, 1994, pp. 80–94.

[39] D. E. Goldberg and K. Deb, A comparative analysis of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms*, vol. 1, pp. 69–93, 1991.

[40] P. Kaelo and M. Ali, Integrated crossover rules in real coded genetic algorithms, *European Journal of Operational Research*, vol. 176, no. 1, pp. 60–76, 2007.

[41] K. Deep and M. Thakur, A new mutation operator for real coded genetic algorithms, *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211–230, 2007.

[42] D. Thierens and D. Goldberg, Elitist recombination: An integrated selection recombination ga, in *Evolutionary Computation, 1994 IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on*, 1994, pp. 508–512.

[43] G. Rudolph, Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[44] R. Chelouah and P. Siarry, Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, *European Journal of Operational Research*, vol. 148, no. 2, pp. 335–348, 2003.

[45] Y. Yun, M. Gen, and S. Seo, Various hybrid methods based on genetic algorithm with fuzzy logic controller, *Journal of Intelligent Manufacturing*, vol. 14, nos. 3&4, pp. 401–419, 2003.

[46] S. Tsutsui and D. E. Goldberg, Search space boundary extension method in real-coded genetic algorithms, *Information Sciences*, vol. 133, no. 3, pp. 229–247, 2001.

[47] O. Hrstka and A. Kučerová, Improvements of real coded genetic algorithms based on differential operators preventing premature convergence, *Advances in Engineering Software*, vol. 35, no. 3, pp. 237–246, 2004.

[48] P. Subbaraj, R. Rengaraj, and S. Salivahanan, Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm, *Applied Energy*, vol. 86, no. 6, pp. 915–921, 2009.

[49] P. Subbaraj, R. Rengaraj, and S. Salivahanan, Enhancement of self-adaptive real-coded genetic algorithm using taguchi method for economic dispatch problem, *Applied Soft Computing*, vol. 11, no. 1, pp. 83–92, 2011.

[50] M. Srinivas and L. M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[51] W.-Y. Lin, W.-Y. Lee, and T.-P. Hong, Adapting crossover and mutation rates in genetic algorithms, *J. Inf. Sci. Eng.*, vol. 19, no. 5, pp. 889–903, 2003.

[52] S. Blum, R. Puisa, J. Riedel, and M. Wintermantel, Adaptive mutation strategies for evolutionary algorithms, in *The Annual Conference: EVEN at Weimarer Optimierungsund Stochastiktage*, 2001.

[53] J. Zhang, H. S.-H. Chung, and W.-L. Lo, Clustering-based adaptive crossover and mutation probabilities for genetic algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 326–335, 2007.

[54] H. Yuan, Adaptive crossover rate and mutation rate in genetic algorithms, *Journal of Capital Normal University (Natural Science Edition)*, vol. 3, p. 003, 2000.

[55] T. A. Pirvu and K. Schulze, Multi-stock portfolio optimization under prospect theory, *Mathematics and Financial Economics*, vol. 6, no. 4, pp. 337–362, 2012.

Chao Gong received the MS degree in industrial economics from Yunnan University of Finance and Economics, Kunming, China, in 2009. He is currently a PhD candidate in the Department of Risk Science in Finance and Management, Chiba Institute of Technology, Chiba, Japan. His research areas include financial risk management, machine learning, and intelligent computing.

Chunhui Xu received the PhD degree in industrial engineering and management from Tokyo Institute of Technology, Japan, in 1995. He is a professor in Chiba Institute of Technology, Japan. His current research interests include financial investment, decision science, and operations research.

Masakazu Ando received the PhD degree in division of administration from Nanzan University, Japan, in 2000. He is a professor in Chiba Institute of Technology, Japan. His current research interests include economic statistics and applied mathematics.

Xiangming Xi received the PhD degree from Tsinghua University in 2016. He is currently a software engineer in Huawei Technology Co. Ltd. His major research interests focus on continuous piecewise linear optimization modeling and continuous piecewise linear algorithms and the corresponding applications on software analysis.