Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 192 (2009) 500–511
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Stochastics and Statistics

On the effectiveness of scenario generation techniques in single-period portfolio optimization

Gianfranco Guastaroba [a,*], Renata Mansini [b], M. Grazia Speranza [a]

[a] University of Brescia, Department of Quantitative Methods, C.da S.Chiara 48/B, 25122 Brescia, Italy
[b] University of Brescia, Department of Electronics for Automation, via Branze 38, 25123 Brescia, Italy

Received 16 August 2006; accepted 27 September 2007
Available online 13 October 2007

Abstract

In single-period portfolio selection problems the expected value of both the risk measure and the portfolio return have to be estimated. Historical data realizations, used as equally probable scenarios, are frequently used to this aim. Several other parametric and non-parametric methods can be applied. When dealing with scenario generation techniques practitioners are mainly concerned on how reliable and effective such methods are when embedded into portfolio selection models. In this paper we survey different techniques to generate scenarios for the rates of return. We also compare the techniques by providing in-sample and out-of-sample analysis of the portfolios obtained by using these techniques to generate the rates of return. Evidence on the computational burden required by the different techniques is also provided. As reference model we use the Worst Conditional Expectation model with transaction costs. Extensive computational results based on different historical data sets from London Stock Exchange Market (FTSE) are presented and some interesting financial conclusions are drawn.

Keywords: Risk management; Conditional value at risk; Portfolio optimization; Scenario generation; Mixed integer linear programming

1. Introduction

The main challenging issue in portfolio selection problems is to provide evidence on how effective the mathematical models are as decision tools in determining reliable solutions to face uncertain future events. The effectiveness of the models strongly depends on the data input required by the models and on the methods used to generate it. In portfolio selection problems the investor aims at selecting a set of securities to optimize a performance measure under specific constraints. Many mathematical models for portfolio selection require the availability of a set of *scenarios*, where a scenario is a realization of a multivariate random variable representing the rates of return of all the securities. The accuracy and effectiveness of the models rely on the quality of the generated scenarios. Different methods can be used to generate scenarios. They range from the simple historical approach, based on the assumption that past realizations are representative of future outcomes, to more complex methods based on randomly re-sampling from historical data (Bootstrapping methods) or on randomly sampling from a chosen distribution function of the multivariate random variable (Monte Carlo simulation) or, again, forecasting methods. Each method used to generate scenarios is called a *scenario generation technique*.

Several academic researchers and practitioners have used scenario generation techniques as tools for supporting financial decision making. The applicability of these techniques for financial purposes has been first recognized by Bradley and Crane in [5] and, more recently, by Mulvey and Vladimirou [21] for asset allocation. Scenario generation techniques have also been used by Nielsen and Zenios [22] for fixed income portfolio management, by Carino

* Corresponding author. Tel.: +3930 298 8599; fax: +3930 240 0925.
E-mail addresses: guastaro@eco.unibs.it (G. Guastaroba), rmansini@ing.unibs.it (R. Mansini), speranza@eco.unibs.it (M.G. Speranza).

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.09.042

et al. [7] and by Consiglio et al. [10] for insurance companies.

This paper aims at comparing different techniques for the generation of scenarios of the rates of return in a portfolio selection problem and at evaluating their effectiveness for single-period portfolio management. We have analyzed five different scenario generation techniques, three of which are non-parametric while the remaining two are parametric. We have compared the in-sample and the out-of-sample characteristics of the optimal portfolios obtained by solving the same portfolio selection model when such techniques are used to generate the input data (the rates of return scenarios). The portfolio optimization model used as a basis for the comparison of the scenario generation techniques is based on the Worst Conditional Expectation (WCE) as performance measure (a detailed analysis of such measure and of its properties can be found in Mansini et al. [20]). The WCE is closely related to the measure called conditional value-at-risk (CVaR_{β}) [24]. More precisely, the two measures are equivalent in the case of continuous distributions of returns, while they can take different values for discrete distributions (see [23]). For the sake of simplicity and following other authors, we will refer to the model using the WCE simply as CVaR model. This is consistent with the recent use of LP formula for the WCE as a computational approximation to CVaR for continuous distributions (see [24]). Finally, notice that, according to the aims of this paper, any other single period portfolio selection model based on a different risk measure could have been alternatively used. In the following we provide a short description of the selected portfolio model.

We consider a situation where an investor intends to optimally select a portfolio of securities and to hold it until the end of a defined investment horizon. In order to correctly simulate the trading conditions encountered by the investor, we assume that, for each selected security, a fixed and a proportional cost are applied and that securities can be bought in fractions of stock units. A set of n securities (risky assets) is available for the investment. For each security $j = 1, 2, \ldots, n$, the rate of return is represented by a random variable R_j with a given mean $r_j = \mathbb{E}\{R_j\}$. We assume that for each random variable R_j its realization r_{jt} under scenario t, $t = 1, \ldots, T$, is known and that each scenario has a probability p_t to occur. Thus, the expected return r_j for security j is computed as mean rate of return over the T scenarios, i.e. $r_j = \mathbb{E}\{R_j\} = \sum_{t=1}^{T} p_t r_{jt}$. Let $\mathbf{x} = (x_j)_{j=1,2,\ldots,n}$ denote the vector of decision variables where x_j, $j = 1, 2, \ldots, n$, represents the fractional value of stock units invested in security j. We assume that no short sales are allowed, i.e. $x_j \geqslant 0$ for $j = 1, \ldots, n$. Each portfolio $\mathbf{x}$ defines a random variable $R_{\mathbf{x}} = \sum_{j=1}^{n} R_j x_j$ representing the portfolio return with expected value given by $\mu(\mathbf{x}) = \mathbb{E}\{R_{\mathbf{x}}\}$. Let $\mu_t = \sum_{j=1}^{n} r_{jt} x_j$ be the realization of the portfolio return $R_{\mathbf{x}}$ under scenario t. Then, $\mathbb{E}\{R_{\mathbf{x}}\} = \sum_{t=1}^{T} p_t \mu_t = \sum_{j=1}^{n} r_j x_j$.

Let q_j be the quotation of security j, $j = 1, 2, \ldots, n$, at the date of portfolio selection and $q_j x_j$ be the amount invested in security j at the same date. We define as f_j the fixed transaction cost incurred by the investor when selecting security j and as c_j the corresponding proportional transaction cost. Moreover, C is the maximum available capital for the investment whereas u_j, $j = 1, 2, \ldots, n$, represents the upper limit on the fractional number of units of security j that the investor can purchase. Finally, we denote by η an independent free variable representing (at optimum) the β-quantile, i.e. the maximum loss associated with the preferred level of β. For any $0 < \beta \leqslant 1$, the CVaR(β) model with transaction costs, by simplicity referred to as CVaR(β) model, can be defined as follows:

CVaR(β) Model

$$\max \quad \eta - \frac{1}{\beta} \sum_{t=1}^{T} p_t d_t \tag{1}$$

$$\eta - \sum_{j=1}^{n} (r_{jt} - c_j) q_j x_j + \sum_{j=1}^{n} f_j z_j \leqslant d_t, \quad t = 1, \ldots, T, \tag{2}$$

$$\sum_{j=1}^{n} (r_j - c_j) q_j x_j - \sum_{j=1}^{n} f_j z_j \geqslant \mu_0 \sum_{j=1}^{n} q_j x_j, \tag{3}$$

$$\sum_{j=1}^{n} q_j x_j = C, \tag{4}$$

$$x_j \leqslant u_j z_j, \quad j = 1, \ldots, n, \tag{5}$$

$$d_t \geqslant 0, \quad t = 1, \ldots, T, \tag{6}$$

$$x_j \geqslant 0, \quad j = 1, \ldots, n, \tag{7}$$

$$z_j \in \{0, 1\}, \quad j = 1, \ldots, n. \tag{8}$$

The objective function (1) maximizes the safety measure represented by the worst conditional expectation. Constraints (2) along with constraints (6) define the non-negative variables d_t as $\max\{0, \eta - y_t\}$, where $y_t = \sum_{j=1}^{n} (r_{jt} - c_j) q_j x_j - \sum_{j=1}^{n} f_j z_j$ is the net portfolio return under scenario t. Thus, each variable d_t measures the deviation of the portfolio net return y_t from the β-quantile η when $y_t < \eta$, whereas it is equal to zero in all the other cases. Constraint (3) establishes that the portfolio net mean return, expressed as difference between the portfolio mean return and the total transaction costs (proportional and fixed), must be at least equal to the portfolio required return $\mu_0 C$. Constraint (4) imposes that the total investment in the portfolio must be equal to C. Constraints (5) define the upper bound u_j on the investment in each security j. Since the fixed cost f_j, $j = 1, 2, \ldots, n$, is paid only if security j is selected, we introduce a binary variable z_j, $j = 1, 2, \ldots, n$, which is forced by constraints (5) to take value 1 if $x_j > 0$. Notice that if $x_j = 0$ then z_j is free to take any value. However, since the fixed costs influence the risk function, at optimum z_j will take value 0 as the most convenient of the two. Finally, constraints (7) avoid short sales on securities whereas constraints (8) define the binary conditions on variables z_j. In Guastaroba et al. [14] the CVaR(β) model with transaction costs has been studied and tested in a re-balancing framework.

The remainder of the paper is organized as follows. Section 2 introduces the scenario generation techniques analyzed and tested in the present paper. Section 3 deals with the experimental analysis and the comparison of the characteristics of the optimal portfolios obtained by using the different scenario generation techniques in the CVaR(β) model. Several historical data sets are used to span different market trends. The main objective of the work is to provide evidence on the model effectiveness in guiding financial decisions under different market conditions and to find out whether there exists a technique that outperforms all the others under different market trends. Some concluding remarks are presented in Section 4, while in the Appendix a more precise description for one of the analyzed parametric scenario generation techniques is provided.

2. Scenario generation techniques

The Linear Programming (LP) based portfolio selection model implemented assumes that returns are discrete random variables. In general, an index t is associated to each scenario, with $t = 1, \ldots, T$, where T is the total number of scenarios. Given n securities, a *scenario* consists of n return realizations, one for each security. We will refer to the tth realization of the rate of return of security j as its realization under scenario t. To estimate portfolio expected return and risk, T mutually exclusive scenarios, each of which occurring with probability p_t, $t = 1, \ldots, T$, are required. Herein, we shortly present different scenario generation techniques. Additional information can be found in the Appendix, while interested readers are referred to Kouwenberg and Zenios [17] and references therein. Among the techniques proposed in the literature, we have analyzed and tested the following ones:

- *Historical data technique* (Hist. Data). This method is one of the most frequently used for its simplicity. It is based upon the assumption that historical data are possible future scenarios. Usually scenarios are treated as equally probable. Such approach does not require any assumption on the distribution function for the rates of return and no correlations between securities have to be computed since they are implicitly considered in the data observed on the market. This procedure preserves the historical mean and variance of the returns. The potential drawback of this approach, and of all approaches making use of historical data, is that future price movements may be substantially different in nature from those observed in the past. Moreover, the number of scenarios that can be produced is limited by the amount of historical data available. In the remaining of the paper we will refer to such scenario generation method as *Hist. Data*. One of the most significant application of this technique can be found in Carino et al. [7,8].
- *Bootstrapping technique* (Boot(T)). The method combines the use of historical data with a bootstrapping technique. As for the historical data technique, a *scenario* corresponds to the joint realizations of the rates of return for all securities as observed in a given time period. Each scenario from the original historical data set can be re-sampled with a constant probability given by one over the cardinality of the set: to select the scenario t to be re-sampled an integer number t, uniformly distributed between 1 and the cardinality of the historical data set, is generated. The procedure is repeated until the new sample reaches the desired size. The approach preserves the correlations between securities. In computational experiments we will refer to such method as *Boot*(T), where T represents the sample size after the re-sampling. The bootstrapping technique has been developed with inferential purposes in Efron and Tibshirani [12], where the expanded sample is treated as a virtual population from which samples are drawn to verify the estimators variability. Frequently, the bootstrapping procedure is used when the size of the available sample is relatively small and one needs a larger number of observations (see [9]). The use of the bootstrapping technique for scenario generation purposes has been suggested by Kouwenberg and Zenios [17].
- *Block bootstrapping technique* (Block-Boot(T)). The block bootstrapping technique is a variant of the previous one which allows to retain original data correlations between periods through the use of the Bootstrapping on blocks of scenarios (see [6]). A block is a set of consecutive scenarios. Given the original time series and once the block length has been chosen the method, instead of single scenarios, re-samples blocks of the same length from the original set. The idea is to choose a block length large enough to guarantee that observations for an interval larger than such value will be nearly independent. For a more complete analysis on rules to correctly compute the block length we refer to Hall et al. [16]. In the remaining of the paper we will refer to such scenario generation method as *Block-Boot*(T), where T is the cardinality of the scenarios after the re-sampling. Both the Boot(T) and the Block-Boot(T) techniques are immune of systematic errors of selection. The potential drawback of these methods is due to the use of historical data, since future rates of return may be substantially different in nature from the past observed data.
- *Monte Carlo simulation techniques* (M-Norm(T), M-tStud(T)). Monte Carlo simulation is a parametric approach that consists in generating scenarios according to a specified distribution function. When using these techniques with a portfolio selection problem the first critical issue is the choice of the multivariate distribution function which better fits the historical rates of return. In practice, the most frequently used distribution function is the multivariate standard Normal distribution, $N(\mathbf{0}, \mathbf{I}_n)$, with zero mean and unit variance–covariance matrix $\mathbf{I}_n$, but other distribution functions can also be used. Since multivariate Normal distribution does not

consider the "fat-tails" or "heavy-tails" effect (see [19]) which frequently characterizes rates of return, we have decided to also consider the multivariate *t*-Student distribution as a possible alternative to the Normal distribution. We will refer to the former approach as to *M-Norm*(T) and to the latter as to *M-tStud*(T), where T is the number of scenarios sampled from the chosen distribution. To generate scenarios from a multivariate Normal distribution with known mean and covariance matrix we have used the method proposed in Levy [18, pp. 234–243], while to estimate the number of degrees of freedom v that controls the heaviness of the tails in a multivariate *t*-Student we have used empirical evidence suggesting that such parameter should range between 3 and 7 for most of the markets (see [13, pp. 510–511]). More precisely, we have decided to consider n different parameters v_j, one for each security $j = 1, \ldots, n$, to better take into account the different heaviness of the return tails in each of the available securities (kurtosis). This procedure is different from the traditional one but allows a more precise analysis of the extreme values in the return distribution of each security by better preserving similarities with historical data.

- *Multivariate generalized ARCH process technique* (M-GARCH(1,1)(T)). Volatility cluster effect is a typical feature characterizing financial time series. Such effect is related to the fact that volatility for financial time series is usually not constant over time (homoskedasticity) but presents heteroskedasticity so that large rates of return tend to be followed by other large values, of both signs, whereas small rates of return are usually followed by small values (see [3]). The most used stochastic process taking into account such effect is the GARCH(q,p) (Generalized ARCH process, see [1]), where q and p are the number of the required past residuals and of the past conditional volatility values, respectively. Many studies (see [4] and, more recently, [11]) extend the original univariate GARCH(q,p) stochastic process to the multivariate case. Usually, the number of parameters to be estimated in a conventional multivariate GARCH(q,p) is too large so that various approximations have been proposed. Due to its computational simplicity, the constant conditional correlation GARCH (CC-MGARCH) proposed by Bollerslev [2] is widely used to estimate the multivariate GARCH stochastic process starting from univariate GARCH processes. We will refer to this approach as *M-GARCH(1,1)*(T), since we have set parameters p and q to 1, where T is the number of generated scenarios. The M-GARCH(1,1)(T) stochastic process can be used as a scenario generation technique through the construction of an *event tree*. Each *node* in the tree is associated with a joint outcome of the rates of return of all securities. An *edge* between two consecutive nodes means the child node has been derived from the parent node according to the implemented stochastic process. Each parent node can generate more than one child node. Nodes with the same distance from the root node are said to belong to the same *stage*. A *scenario* is associated to each leaf (pending node) of the tree. When generating the event tree a crucial issue is related to the number of nodes and of stages to take into account. The objective is to find the right trade-off between a number of scenarios large enough to avoid approximation errors and small enough to be computationally tractable. We have decided to construct a binomial event tree where from each parent node two child nodes are generated and 2^l nodes belong to stage l. Thus, for instance, if the number of stages is 12, then 4096 scenarios will be generated. Details about the use of the CC-MGARCH stochastic process to generate a binomial event tree are provided in the Appendix.

The first three described scenario generation techniques are non-parametric while the last two are parametric. The basic difference between a non-parametric and a parametric approach is that in the former case scenarios generated are represented by real data, while in the latter scenarios can be totally hypothetical. The non-parametric approaches have the main advantage to be simple to compute and easy to understand. On the contrary, a parametric approach depends on the chosen distribution function, and the choice of such distribution, along with its parameters, may be difficult. Moreover, there is no evidence that the distribution that better fits the data will remain constant over time. Nevertheless, there are many reasons to support the use of a parametric approach with respect to a non-parametric one. First of all, non-parametric techniques are criticized to be strongly dependent upon a defined sample so that they can be hardly generalized to other samples. Furthermore, it has been claimed that if the observations in a sample are biased, then non-parametric approaches will be biased too. One of the main objectives of the present paper is to evaluate if the additional computational burden usually implied by parametric approaches is really compensated by better portfolio performances.

3. Experimental analysis

In this section we compare the portfolios selected by solving the CVaR(β) model with scenarios generated by using the five discussed techniques. Computational experiments have been conducted on a PC with a 3000 MHz Intel Pentium III processor and 1 GB of RAM. The model has been implemented in C++ by means of Concert Technology 2.0 and solved with CPLEX 9.0. We first present the testing environment, then the results of the in-sample and finally those of the out-of-sample analysis.

3.1. Testing environment

Historical data are represented by weekly rates of return, computed by using closing stock prices of the 100 securities composing the FTSE100 Index at the date of

Fig. 1. The four different market periods.

September 25th, 2005. No dividends have been considered. The FTSE100 Index, representing approximately about 80% of the entire London Stock Exchange market (FTSE), is recognized as one of the most important European benchmarks.

We have tested the model under different market behaviors. We have constructed four data sets corresponding to different in-sample and out-of-sample time periods. Each data set temporal positioning is shown in Fig. 1. The first data set is characterized by an increasing market trend in the in-sample period as well as in the out-of-sample period (*up–up trend*), the second data set by an increasing trend in the in-sample period and by a decreasing one in the out-of-sample period (*up–down trend*), the third data set by a decreasing trend in the in-sample period and by an increasing one in the out-of-sample period (*down–up trend*) and, finally, the last set by a decreasing trend in both the in-sample and the out-of-sample periods (*down–down trend*). Each of these data sets consists of 2 years of in-sample weekly observations (104 realizations) and 1 year of out-of-sample ones (52 realizations).

For each data set we have solved the CVaR(β) model by considering $C = 100{,}000$ Euros, three different levels of the minimum required rate of return μ_0 (0, 0.05 and 0.10 on yearly basis) and three different values of the quantile parameter β (0.01, 0.05 and 0.10). Moreover, we have assumed a fixed cost equal to 12 Euros and a proportional cost equal to 0.195% for all the securities. These are real-case conditions applied by Italian brokers for operations carried out on the FTSE market. Finally, upper bounds u_j are assumed to be computed as $u_j = C/q_j$, $j = 1, \dots, n$.

Table 1
Number of scenarios (sample size) generated with each technique

Hist. Data	Boot(T)	Block-Boot(T)	M-Norm(T)	M-tStud(T)	M-GARCH(1,1)(T)
104	1000	1000	1000	1000	1024
	5000	5000	5000	5000	4096
	10,000	10,000	10,000	10,000	8192

Table 1 shows the number of scenarios generated by the different techniques for each set of data. For all the techniques but for the historical data and for the M-GARCH(1,1)(T) the same number of scenarios has been set. In the case of historical data sampling (Hist. Data), there is only one sample represented by the 104 historical scenarios. For the M-GARCH(1,1)(T) approach the number of scenarios has to be a power of 2. Given a data set, for each value of β, each required rate of return μ_0, each technique and each scenario sample, an instance has been generated. This means 144 instances for each data set and 576 all together.

Table 2 shows, for each scenario generation technique, the average out of the four data sets computational time required to build an instance as a function of the number of scenarios taken into account. For the M-GARCH(1,1)(T) model, the precise size of each sample is indicated in parentheses. Notice that, in the worst case, the non-parametric approaches build an instance in less than 3 minutes. Generating scenarios with both Monte Carlo simulation techniques is more time consuming (always more than 5 minutes). Larger times required to create an instance by using a multivariate t-Student distribution with respect to a Normal one are due to the additional computational burden implied by the estimation of the number of degrees of freedom v_j for each security. The most time consuming approach is the M-GARCH(1,1)(T) model. To create a binomial event tree even for the smallest instances, has required a time larger than 1 hour.

Table 2
Average computational times (in minutes) to generate instances according to sample size

Technique	Sample size			
	104	1000	5000	10,000
Hist. Data	0.17	–	–	–
Boot(T)	–	0.42	1.35	2.50
Block-Boot(T)	–	0.50	1.43	2.67
M-Norm(T)	–	5	14	25
M-tStud(T)	–	20	29	40
M-GARCH(1, 1)(T)	–	60 (2^{10})	900 (2^{12})	3480 (2^{13})

3.2. In-sample analysis

In this section we present the characteristics of the optimal portfolios selected by the CVaR(β) model when using the described techniques to generate the scenarios. For the sake of clarity, we have decided to only report the results obtained by setting μ_0 equal to 5% on yearly basis and $\beta = 0.05$, respectively. The complete set of results is available in Guastaroba et al. [15].

As a starting point of the analysis, we have compared the portfolios obtained by using the same scenario generation technique when changing the size of the sample. The goal has been to verify if an ideal sample size can be worked out for each technique. At this aim we have analyzed the best ex-post performances over all the four data sets for each technique. The results indicate that the best sample size is equal to 1000 for the Boot(T) and Block-Boot(T) techniques, equal to 10,000 for M-Norm(T) and M-tStud(T) techniques and to 4096 scenarios for M-GARCH(1, 1)(T). A first conclusion which we feel we can draw is that with techniques like Bootstrapping and Block Bootstrapping it is not necessary to generate large samples. On the contrary, when using a parametric approach like Monte Carlo simulation it is evident how the larger the number of scenarios the better the selected portfolios. Usually, Monte Carlo simulation requires a massive number of scenarios. We have set the maximum sample size equal to 10,000 since, in extensive preliminary analysis, we have evaluated that the slightly better performances which could be obtained with a sample size larger than 10,000 do not justify the longer time required to solve the optimization model. For M-GARCH(1, 1)(T) technique the sample size equal to 4096 is selected since providing the best trade-off between average return and downside risk.

Tables 3 and 4 show the characteristics of the optimal portfolios selected by the model when using different scenario generation techniques over the four data sets. In particular, they report the results obtained by each technique when using its best sample size according to the previous analysis. Each table is divided into two parts providing the results for the two data sets characterized by the same trend in the in-sample period. Each part consists of four columns that show the number of securities selected (div), the minimum (min) and the maximum (max) portfolio shares and the computational time (in minutes) needed to optimally solve the instance. Summarizing the main figures about optimal in-sample portfolios, a consistent difference can be noticed between the cases with increasing trend in the in-sample period (i.e. *up–up* and *up–down* data sets) with respect to those with in-sample decreasing trend (i.e. *down–up* and *down–down* data sets). The number of securities selected when the market is decreasing in the in-sample period is considerably lower than the number when the market trend is increasing.

As far as the computational time is concerned, the most time consuming techniques are the two Monte Carlo approaches and, in particular, the M-Norm(10,000). This is mainly due to the large number of scenarios required by this technique. Moreover, when comparing computational times required to solve instances of the same size, we have noticed that both Monte Carlo techniques usually require almost twice the computational time required by the Bootstrapping or the Block Bootstrapping techniques (see [15]). When all the tested instances are taken into

Table 3
Up–up and up–down data sets: optimal portfolio characteristics with $\mu_0 = 5\%$ and $\beta = 0.05$

Instances	Up–up data set				Up–down data set			
	div	Shares		Time (in minutes)	div	Shares		Time (in minutes)
		min	max			min	max	
Hist. Data	14	0.026	0.244	6.10	11	0.020	0.247	0.02
Boot(1000)	14	0.027	0.244	65.65	11	0.027	0.316	0.45
Block-Boot(1000)	13	0.028	0.261	59.83	12	0.023	0.255	0.65
M-Norm(10,000)	8	0.037	0.292	318.27	8	0.043	0.577	56.32
M-tStud(10,000)	9	0.040	0.214	222.44	8	0.032	0.545	75.75
M-GARCH(1, 1)(4096)	7	0.028	0.351	0.58	6	0.048	0.312	1.38

Table 4
Down–up and down–down data sets: optimal portfolio characteristics with $\mu_0 = 5\%$ and $\beta = 0.05$

Instances	Down–up data set				Down–down data set			
	div	Shares		Time (in minutes)	div	Shares		Time (in minutes)
		min	max			min	max	
Hist. Data	4	0.078	0.679	0.01	5	0.066	0.588	0.01
Boot(1000)	5	0.061	0.663	0.14	5	0.083	0.544	0.43
Block-Boot(1000)	6	0.040	0.372	0.21	6	0.063	0.560	0.15
M-Norm(10,000)	4	0.062	0.621	5.57	7	0.052	0.478	35.99
M-tStud(10,000)	4	0.070	0.615	9.46	6	0.058	0.543	21.74
M-GARCH(1,1)(4096)	3	0.101	0.793	2.21	5	0.061	0.537	1.49

account the M-GARCH(1,1)(4096) is, on average, as time consuming as the Boot(1000) or the Block-Boot(1000) techniques and the Hist. Data technique is, on average, the most efficient one.

Finally, considering the computational times required to generate an instance (see Table 2), the overall most time consuming technique becomes the M-GARCH(1,1)(T) while the Hist. Data technique remains the most time saving technique.

3.3. Out-of-sample analysis

In the out-of-sample analysis, we examine the behavior of the portfolios selected by the CVaR(β) model in the 52 weeks following the portfolio selection date. As for the in-sample analysis, we only describe the results obtained with μ_0 equal to 5% on yearly basis and β equal to 0.05. Moreover, we describe the characteristics of the optimal portfolios selected by the model when each technique has

Table 5
Out-of-sample statistics for $\mu_0 = 5\%$ and $\beta = 0.05$

Model	#	r av	r med	std	s-std	Sortino index
Up–up data set						
Hist. Data	32	19.29	175.58	0.0128	0.0072	0.3135
Boot(1000)	32	18.09	181.32	0.0125	0.0072	0.2876
Block-Boot(1000)	31	19.71	115.69	0.0115	0.0064	0.3758
M-Norm(10,000)	32	25.28	20.48	0.0100	0.0048	0.7149
M-tStud(10,000)	34	25.39	4.13	0.0116	0.0063	0.5324
M-GARCH(1,1)(4096)	33	39.55	−2.46	0.0177	0.0099	0.5193
FTSE100	32	45.92	53.36	0.0257	0.0153	0.3411
Up–down data set						
Hist. Data	21	7.24	606.41	0.0358	0.0249	0.0087
Boot(1000)	24	3.79	460.22	0.0326	0.0229	−0.0054
Block-Boot(1000)	21	3.61	430.07	0.0331	0.0228	−0.0063
M-Norm(10,000)	25	−8.74	54.38	0.0163	0.0137	−0.1342
M-tStud(10,000)	25	−5.91	41.03	0.0149	0.0123	−0.1227
M-GARCH(1,1)(4096)	30	17.62	596.58	0.0355	0.0267	0.0430
FTSE100	29	−8.54	18.76	0.0267	0.0204	−0.0729
Down–up data set						
Hist. Data	30	26.82	−25.71	0.0132	0.0062	0.5803
Boot(1000)	29	24.65	−30.93	0.0116	0.0052	0.6305
Block-Boot(1000)	34	70.19	33.67	0.0233	0.0080	1.2750
M-Norm(10,000)	30	24.77	−31.40	0.0123	0.0052	0.6456
M-tStud(10,000)	26	23.11	−32.11	0.0121	0.0052	0.5804
M-GARCH(1,1)(4096)	29	16.45	−36.70	0.0120	0.0073	0.2499
FTSE100	30	34.19	27.39	0.0228	0.0101	0.4236
Down–down data set						
Hist. Data	22	−7.00	180.76	0.0121	0.0095	−0.1884
Boot(1000)	25	−4.70	165.91	0.0133	0.0106	−0.1321
Block-Boot(1000)	22	−5.01	−9.29	0.0143	0.0107	−0.1353
M-Norm(10,000)	22	−7.28	98.39	0.0138	0.0111	−0.1577
M-tStud(10,000)	23	−5.41	113.89	0.0128	0.0099	−0.1548
M-GARCH(1,1)(4096)	22	−9.82	142.82	0.0119	0.0102	−0.2146
FTSE100	22	−22.93	−14.30	0.0342	0.0283	−0.0852

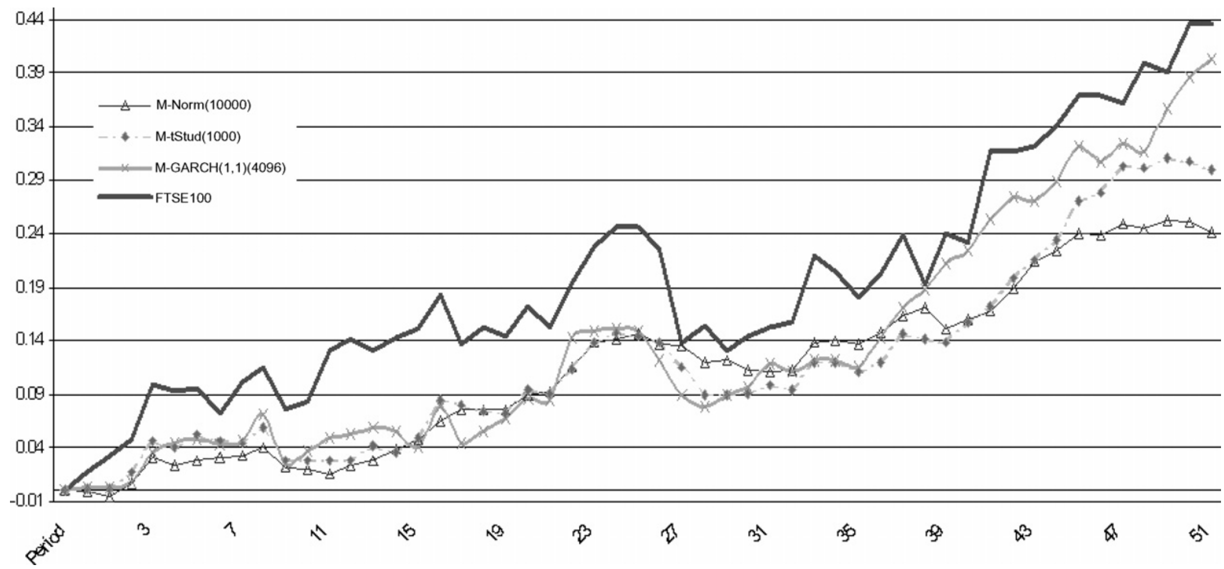

Fig. 2. Cumulative returns (up–up data set): comparison between portfolio optimization models ($\mu_0 = 5\%$ and $\beta = 0.05$) and the market index.

been applied with its best sample size. Details on the remaining results can be found in Guastaroba et al. [15].

To compare the out-of-sample optimal portfolio performances we have computed cumulative returns and the following six ex-post parameters: the number of times out of 52 the portfolio rate of return outperforms the corresponding required one (#), the average (r_{av}) and the median (r_{med}) of the rate of returns on yearly basis, the standard deviation (std), the semi-standard deviation (s-std), the *Sortino* index.

The latter provides a measure of the over-performance of the portfolio mean rate of return with respect to the required one per unit of downside risk (here measured by the semi-standard deviation). All the dispersion measures (std, s-std) and the *Sortino* index have been computed with respect to the minimum required rate of return to make them directly comparable in the different instances.

In Table 5 we show all the resulting values for the performance measures and the dispersion parameters we have computed. For each data set the market behavior of the FTSE100 index is also shown. Figs. 2–5 compare, for each data set, the ex-post cumulative returns of the market index FTSE100 with respect to the cumulative returns of the three best ex-post optimal portfolios. In some cases (see Figs. 3 and 5) we have plotted four instead of three portfolios selected cumulative returns, since in that data set the behavior of two techniques (namely, the Hist. Data and the Boot(T) techniques) are very similar.

As far as the ex-post parameters are concerned, the portfolios selected by using the Hist. Data technique, despite

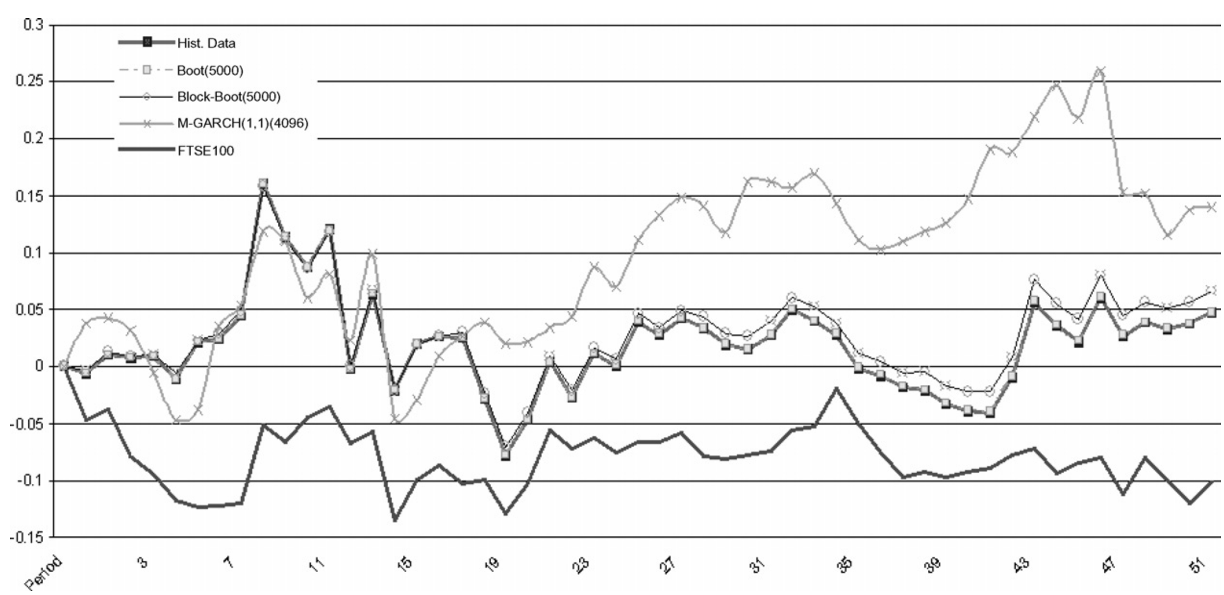

Fig. 3. Cumulative returns (up–down data set): comparison between portfolio optimization models ($\mu_0 = 5\%$ and $\beta = 0.05$) and the market index.

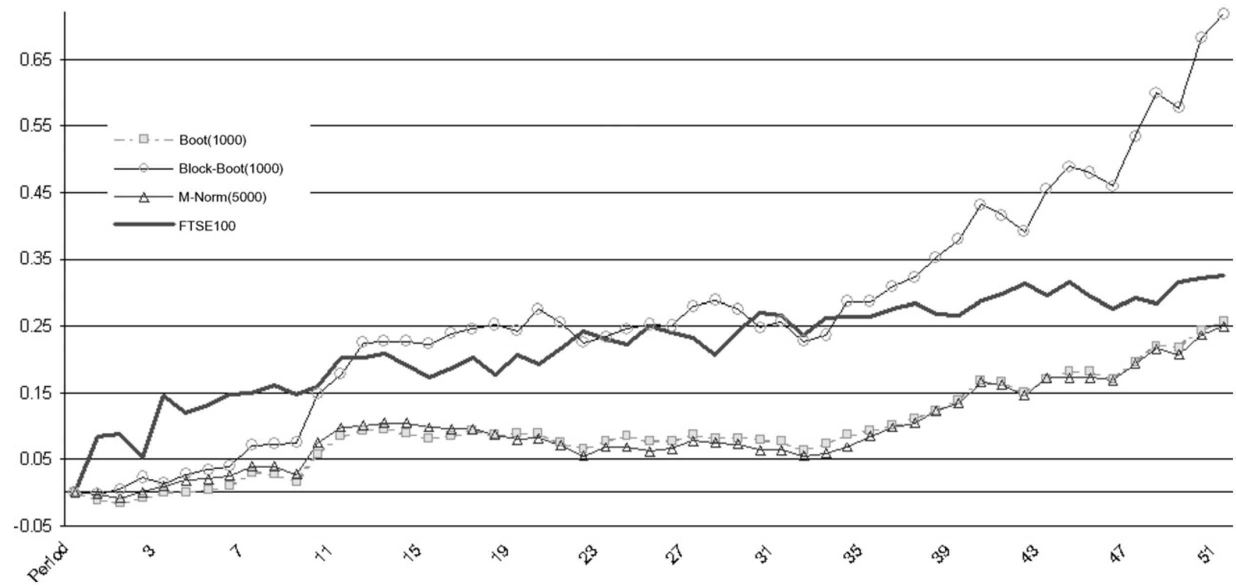

Fig. 4. Cumulative returns (down–up data set): comparison between portfolio optimization models ($\mu_0 = 5\%$ and $\beta = 0.05$) and the market index.

the simplicity of the method, yield rates of return very close to those of all the other portfolios (see Table 5) and have a value of the downside risk measure very similar to that of the portfolios selected by the other non-parametric techniques.

As expected, the Block-Boot(1000) dominates the Boot(1000) technique. In particular, the average rate of return of the portfolios selected by using the Block-Boot(1000) is altogether greater than that obtained by using the Boot(1000) technique, with a similar downside risk. In the *down–down* data set (see Fig. 5) after 4 weeks from the portfolio selection date, the Block-Boot(1000) shows the best ex-post cumulative return until the 33rd week. In the *up–down* data set the portfolio selected by using the Block-Boot(1000) technique have a behavior similar to that of the other non-parametric approaches. Finally, in the *down–up* data set (see Fig. 4) the Block-Boot(1000) is the technique that consistently outperforms all the other scenario generation techniques. To conclude, the Block-Boot(1000) seems to be the technique that in all the four data sets provides portfolios whose ex-post performance is never the worst and in several cases is the best. Moreover, such performances are obtained guaranteeing a downside risk that is not significantly different from that of the other non-parametric approaches and lower than the GARCH(1,1)(T) (see Table 5, especially for the s-std figures). Block-Boot(1000) turns out to be the best choice for an investor who might accept more risk only if compensated by a more than proportional increase in the expected return.

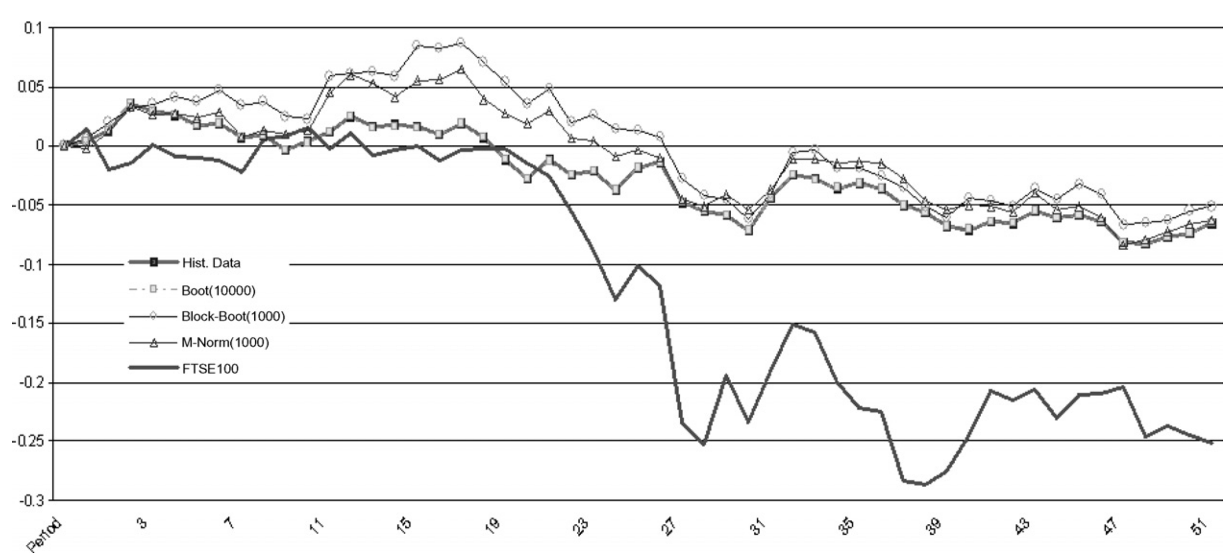

Fig. 5. Cumulative returns (down–down data set): comparison between portfolio optimization models ($\mu_0 = 5\%$ and $\beta = 0.05$) and the market index.

By analyzing the figures related to the parametric techniques, the portfolios selected by using the M-Norm(10,000) and the M-tStud(10,000) techniques are almost always those with the lowest downside risk, independently from the data set under analysis. In some data sets (see Table 5 for the *down–up* figures), the differences with respect to other approaches are not too large, while in others (see Table 5 for the *up–down* figures) the downside risk incurred by using both the Monte Carlo generation techniques is significantly lower than that of all the other approaches. Nevertheless, such values are usually associated to an average rate of return very similar to that of portfolios obtained by applying other techniques and in at least one data set (see Table 5 for the *up–down* figures) even worse. In order to choose between the use of a multivariate Normal distribution with respect to a multivariate *t*-Student distribution in a Monte Carlo approach, one may notice that the portfolios selected by using the M-Norm(10,000) technique often dominate those selected by using the M-tStud(10,000) technique when the market is increasing in the out-of-sample period (see Table 5 for the *up–up* and the *down–up* figures). The reverse is true when the market is decreasing in the ex-post period (see Table 5 for the *up–down* and the *down–down* figures). This can be explained by considering that the heaviness of the tails, and thus the degrees of freedom of the *t*-Student distribution, is not constant over time. In particular, when the market trend is decreasing the tails are heavier than when the market is increasing, and the empirical probability distribution differentiates from the normal one.

When all the tested data sets are considered, the best ex-post average rate of return is provided by the portfolios selected by using M-GARCH(1, 1)(4096) as scenario generator. More precisely, in the *up–up* and in the *up–down* data sets the average rate of return associated to portfolios obtained with the M-GARCH(1, 1)(4096) technique is always significantly greater than that of portfolios obtained by applying all the other techniques. The average rate of return in the *up–down* data set (see Table 5) is more than twice the average rate of return obtained by using the Hist. Data technique and it is almost 20% larger than the average of all the other techniques in the *up–up* data set. Unfortunately, the behavior is not confirmed in the remaining two data sets. In the *down–up* and in the *down–down* data set the portfolios selected by applying the M-GARCH(1, 1)(4096) technique are dominated by all the other portfolios. Nevertheless, since the difference in terms of average rates of return in these two data sets between M-GARCH(1, 1)(4096) and the other techniques is about 5% on yearly basis, and that the difference in the *up–up* and in the *up–down* data sets is significantly greater, a risk neutral investor may decide to choose the M-GARCH(1, 1)(4096) technique in order to maximize the expected portfolio rate of return. Through a deep analysis of all the out-of-sample portfolio performances for all the values of μ_0 and β (see [15]) one may notice that M-GARCH(1, 1)(T) is the technique that quite often provides the most unstable portfolios. This can be observed by looking at the std and the s-std values shown in Table 5 for the *up–up* and for the *up–down* data sets, and the s-std values computed in the *down–up* data set.

In almost every data set there is at least a portfolio selected by the CVaR(β) model that, in terms of average rate of return, outperforms the market index. This is particularly true when the market index is decreasing in the ex-post data set. In all such cases the difference between the average rate of return associated to a portfolio selected by the optimization model and the average rate of return of the market index is consistent, confirming that portfolio optimization models represent a valuable tool for financial decisions. In the *down–down* data set all the portfolios selected by the optimization model yield a mean rate of return higher than that provided by the market index, which is at least 13% on yearly basis worse than that of the optimized portfolios. In the *down–up* data set (see Fig. 4), the market index outperforms almost all the portfolios generated by the model by about 10% on yearly basis, but it is significantly outperformed by the portfolio obtained by using Block-Boot(1000) (the difference is about 36% on yearly basis, see Table 5). Only in the *up–up* data set the market obtains an average rate of return which is always greater than that reached by all the portfolios selected by the optimization model. Nevertheless, the difference using the M-GARCH(1, 1)(4096) is on average lower than 10% and in some cases (see Table 5) it reduces to about 6% on yearly basis. Similar conclusions can be drawn by analyzing Fig. 2. The market index has shown, on average, a cumulative return extremely unstable (see Figs. 2–5), in particular if compared with the optimized portfolios. The higher volatility of the FTSE100 index ex-post returns is confirmed by the dispersion measures shown in Table 5. The s-std value is twice the values associated to almost all the portfolios selected by the model in the *up–up* and in the *down–up* data sets and to all the model portfolios in the *down–down* data set. Only in the *up–down* data set the market index shows a s-std value better than some portfolios selected by the CVaR(β) model. However, the lower downside risk is frequently associated to a worse value of the mean rate of return. To conclude, both portfolios and market index have, on average, similar fluctuations but the size of these fluctuations, and in particular of the decreasing ones, is significantly lower for the portfolios selected by the optimization model.

By summarizing all the above results, we can draw the following conclusions:

1. independently of the scenario generation technique applied, the use of a portfolio optimization model represents a valuable tool for financial decisions. This is true when the market trend is increasing and becomes of crucial importance when the market trend is decreasing;
2. when comparing scenario generation techniques over different market trends, it is possible to conclude that the Block-Boot(1000) is the only technique that has

shown good performance in terms of both mean return and downside risk, while the M-GARCH(1, 1)(4096) technique is the best possible choice in order to maximize the expected portfolio rate of return;
3. the Hist. Data technique, despite its simplicity, generates portfolios with rates of return very close to those of the other optimized portfolios and with a downside risk incurred very similar to that of the portfolios selected by the other non-parametric approaches;
4. each of the analyzed techniques may require a different number of generated scenarios to be effectively used in an optimization model. In particular, computational results have shown that, on average, a well-defined size of the sample can be worked out for each scenario generation technique. Such size is equal to around 1000 scenarios for bootstrapping techniques, about 10,000 scenarios for Monte Carlo simulation and no more than 5000 for M-GARCH technique.

4. Conclusions

In this paper we have studied an optimal portfolio selection problem where scenarios for the rates of return are generated by means of three non-parametric and two parametric techniques. Specifically, we have solved a single period portfolio optimization model with transaction costs and the worst conditional expectation as performance measure. Extensive computational results on real-life data from London Stock Exchange have allowed us to draw some interesting conclusions. The first aim of the paper was to prove that the portfolio optimization model is a reliable tool to support financial decisions. The second aim was to find out whether there exists a technique to generate scenarios that outperforms all the others under different market trends. With respect to the first aim, we can conclude that portfolio optimization models represent a valuable tool for financial decisions providing good results when the market is increasing and becoming of crucial importance when the market is decreasing determining a strong reduction of the instability of the portfolio return. The effectiveness of portfolio optimization models is confirmed by observing that, in almost all the possible market behaviors, there is at least a portfolio selected by means of the portfolio optimization model that significantly outperforms the market index. With respect to the second aim, if the investor is risk averse and would like to hedge his investment from a possible dramatic fall in market quotation during the ex-post period, the Block-Boot(1000) is the best technique to use. If the investor is, on the contrary, more risk seeking in order to maximize his portfolio expected return, the use of M-GARCH(1, 1)(4096) technique is the suggested choice. Finally, the Hist. Data approach, the most widely used technique in practice and in academic research, considering its simplicity and the quality of the portfolios selected, can be a sensible choice.

Appendix. CC-MGARCH technique

Let $\mathbf{x}_t$ be the time series vector of the rates of return of n securities at time t. The CC-MGARCH technique is based on the following relations:

$$\mathbf{x}_t = \mathbb{E}\{x_t|\psi_{t-1}\} + \epsilon_t \tag{9}$$

and

$$\mathrm{Var}(\boldsymbol{\epsilon}_\mathbf{t}|\psi_{t-1}) = \mathbf{H}_\mathbf{t},$$

where ψ_{t-1} represents the σ-field generated by all the information available until time $t-1$ and $\boldsymbol{\epsilon}_\mathbf{t}$ is the vector of residuals, see [2]. $\mathbf{H}_\mathbf{t}$ is the time varying conditional covariance matrix that Bollerslev [2] suggests to compute as follows:

$$\mathbf{H}_\mathbf{t} = \mathbf{D}_\mathbf{t}\mathbf{R}\mathbf{D}_\mathbf{t}, \tag{10}$$

where $\mathbf{R}$ is a time invariant correlation matrix containing the unconditional correlations, $\mathbf{D}_\mathbf{t} = \mathrm{diag}\{\sqrt{h_{jt}}\}$ and h_{jt} is the conditional variance of security j at time t, $j = 1, \ldots, n$. In details, the diagonal matrix $\mathbf{D}_\mathbf{t}$ containing the conditional standard deviation can be generated estimating n univariate GARCH models. The correlation matrix constant over time $\mathbf{R}$ can be simply estimated as the unconditional correlation matrix of the standardized residuals, whereas the vector of residuals $\epsilon_\mathbf{t}$ is computed as $\epsilon_\mathbf{t} = \boldsymbol{\xi}_\mathbf{t}\mathbf{C}_\mathbf{t}$, where $\mathbf{C}_\mathbf{t}\mathbf{C}_\mathbf{t}^\mathrm{T} = \mathbf{H}_\mathbf{t}$ and $\xi_\mathbf{t}$ is distributed as a white noise, i.e. $\xi_\mathbf{t} \sim \mathbf{WN}(\mathbf{0}, \mathbf{1})$.

To forecast the following period, Eq. (10) can be rewritten as follows:

$$\mathbf{H}_{\mathbf{t+1}} = \mathbf{D}_{\mathbf{t+1}}\mathbf{R}\mathbf{D}_{\mathbf{t+1}}, \tag{11}$$

where

$$h_{jt+1} = \alpha_{0j} + \alpha_{1j}\epsilon_{jt+1}^2 + \beta_{1j}h_{jt+1}. \tag{12}$$

To use CC-MGARCH model for scenario generation one has to forecast the time varying conditional covariance matrix after one period by using (11) and (12). Then, after performing the Cholesky's factorization, s scenarios using (9) one period after are generated, where in $\boldsymbol{\epsilon}_{t+1}$ the matrix $\xi_{t+1,k}$, $k = 1, \ldots, s$, is constructed by randomly sampling n values s times from the historical standardized residuals. In this way, one may generate s time series vectors $x_{t+1,k}$, $k = 1, \ldots, s$, representing s forecasted outcomes of the multivariate random variable x_t. These s values represent the nodes at stage $t+1$ in the event tree. The nodes at stage $t+2$ can be computed recursively by forecasting one period after the time varying conditional variance by considering the realizations occurred in the parent nodes.

References

[1] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Economics 31 (1986) 307–327.
[2] T. Bollerslev, Modelling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model, Review of Economics and Statistics 72 (1990) 498–505.
[3] T. Bollerslev, R.Y. Chou, K.F. Kroner, ARCH modelling in finance: A review of the theory and empirical evidence, Journal of Econometrics 52 (1992) 5–59.

[4] T. Bollerslev, R. Engle, J.M. Wooldridge, A capital asset pricing model with time varying covariances, Journal of Political Economy 96 (1988) 116–131.

[5] S.P. Bradley, D.B. Crane, A dynamic model for bond portfolio management, Management Science 19 (1972) 139–151.

[6] P. Bühlmann, Bootstrap for time series, Statistical Science 17 (2002) 52–72.

[7] D.R. Carino, T. Kent, D.H. Myers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, W.T. Ziemba, The Russel–Yasuda Kasai model: An asset/liability model for a Japanese Insurance Company using multistage stochastic programming, Interfaces 24 (1994) 29–49.

[8] D.R. Carino, D. Myers, W.T. Ziemba, Concepts, technical issue, and uses of the Russel–Yasuda Kasai financial planning model, Operations Research 46 (1998) 450–462.

[9] Y. Chong Ho, Resampling methods: Concepts, applications, and justification, Practical Assessment, Research & Evaluation 8 (2003).

[10] A. Consiglio, F. Cocco, S.A. Zenios, The value of integrative risk management for insurance products with guarantees, Journal of Risk Finance 4 (2001) 1–11.

[11] Z. Ding, R. Engle, Large scale conditional covariance matrix modeling, estimation and testing, Academia Economic Papers 29 (2001) 157–184.

[12] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.

[13] P. Glasserman, Montecarlo Methods in Financial Engineering, Springer-Verlag, New-York, 2003.

[14] G. Guastaroba, R. Mansini, M.G. Speranza, On the use of CVaR model in a rebalancing portfolio strategy, Technical Report 249, Department of Quantitative Methods, University of Brescia, Italy, 2005, submitted for publication.

[15] G. Guastaroba, R. Mansini, M.G. Speranza, On the effectiveness of scenario generation techniques in single-period portfolio optimization, Technical Report 268, Department of Quantitative Methods, University of Brescia, Italy, 2006.

[16] P. Hall, J.L. Horowitz, B. Jing, On blocking rules for the bootstrap with dependent data, Biometrika 82 (1995) 561–574.

[17] R. Kouwenberg, S.A. Zenios, Stochastic Programming Models, Handbook of Asset and Liability Management, vol.1, Handbooks in Finance, Elsevier, 2006.

[18] G. Levy, Computational Finance: Numerical Methods for Pricing Financial Instruments, Elsevier, 2004.

[19] B. Mandelbrot, The variation of certain speculative prices, The Journal of Business of the University of Chicago 26 (1993) 394–419.

[20] R. Mansini, W. Ogryczak, M.G. Speranza, LP solvable models for portfolio optimization: A classification and computational comparison, IMA Journal of Management Mathematics 14 (2003) 187–220.

[21] J.M. Mulvey, H. Vladimirou, Stochastic network programming for financial planning problems, Management Science 38 (1992) 1643–1664.

[22] S.S. Nielsen, S.A. Zenios, A stochastic programming model for funding single premium deferred annuities, Mathematical Programming 75 (1996) 177–200.

[23] W. Ogryczak, A. Ruszczyński, Dual stochastic dominance and related mean–risk models, SIAM Journal on Optimization 13 (2002) 60–78.

[24] R.T. Rockafellar, S. Uryasev, Optimization of conditional value-at-risk, Journal of Risk 2 (2000) 21–41.