Ann Oper Res (2015) 229:409–427
DOI 10.1007/s10479-014-1768-2

Two-stage portfolio optimization with higher-order conditional measures of risk

Sıtkı Gülten · Andrzej Ruszczyński

Published online: 13 March 2015
© Springer Science+Business Media New York 2015

Abstract We describe a study of application of novel risk modeling and optimization techniques to daily portfolio management. In the first part of the study, we develop and compare specialized methods for scenario generation and scenario tree construction. In the second part, we construct a two-stage stochastic programming problem with conditional measures of risk, which is used to re-balance the portfolio on a rolling horizon basis, with transaction costs included in the model. In the third part, we present an extensive simulation study on real-world data of several versions of the methodology. We show that two-stage models outperform single-stage models in terms of long-term performance. We also show that using high-order risk measures is superior to first-order measures.

Keywords Stochastic programming · Scenario tree generation · Coherent measures of risk · Portfolio optimization · Risk

1 Introduction

The main objective of this paper is to evaluate the usefulness of several risk modeling and optimization techniques for daily stock portfolio optimization. In the portfolio optimization problem in its simplest form, the return rates of n assets are represented by an n-dimensional random vector R, with R_j denoting the return rate of asset $j = 1, \dots, n$. The n-dimensional vector z represents the distribution of the capital among assets: z_j is equal to the fraction of the capital invested in asset j. The total return rate of the portfolio at the end of the investment period is

S. Gülten
Department of Management Science and Information Systems, Rutgers University,
1 Washington Park, Newark, NJ 07102, USA
e-mail: sgulten@business.rutgers.edu

A. Ruszczyński (✉)
Department of Management Science and Information Systems, Rutgers University,
94 Rockefeller Road, Piscataway, NJ 08854, USA
e-mail: rusz@business.rutgers.edu

$$R^\top z = \sum_{j=1}^{n} R_j z_j.$$

The portfolio problem is to find an "optimal" way to distribute the initial capital among the n assets, under the condition that $z \in Z$, where $Z \subset \Re^n$ is a convex and compact set of feasible asset allocations. As the return rates of the assets are random, the portfolio return rate is a random variable, and thus the meaning of "optimal" depends of the modeling approach.

In a pioneering study, Markowitz (1952) argued that portfolio performance can be measured by using two scalar characteristics: the mean of the portfolio return, $E\big[R^\top z\big]$, and the variance of the return, $\mathrm{Var}\big[R^\top z\big]$, which characterizes its riskiness. We can then minimize the variance for a fixed value of the mean, or maximize the mean, while keeping the variance bounded. Since then, numerous theoretical and practical studies evaluated the usefulness of the mean–variance approach in portfolio optimization.

Further improvement was made by considering more general mean–risk models, with different measures of variability (Kijima and Ohnishi 1993). By considering consistency with stochastic dominance, the papers (Ogryczak and Ruszczyński 1999, 2001, 2002), introduced a family of mean–semideviation models, which are particularly useful for portfolio models (see, e.g., Mansini et al. 2003; Ruszczyński and Vanderbei 2003).

In the last decade, axiomatic models of risk have been studied extensively, in particular, *coherent risk measures*, introduced by Artzner et al. (1999). In the following definition, the uncertain outcomes X and Y represent losses, and $\mathbb{1}$ denotes the sure loss of 1. Coherent risk measures are functionals $\rho : \mathcal{X} \to \Re$ defined on a suitable vector space $\mathcal{X}$ of random outcomes, which satisfy the following axioms:

Convexity: $\rho(\alpha X + (1-\alpha)Y) \le \alpha\rho(X) + (1-\alpha)\rho(Y), \quad \forall X, Y \in \mathcal{X}, \alpha \in [0, 1]$;
Monotonicity: If $X, Y \in \mathcal{X}$, and $X \le Y$, then $\rho(X) \le \rho(Y)$;
Translation Property: If $a \in \Re$ and $X \in \mathcal{X}$,then $\rho(X + a\mathbb{1}) = \rho(X) + a$;
Positive Homogeneity: If $\beta \ge 0$ and $X \in \mathcal{X}$, then $\rho(\beta X) = \beta\rho(X)$.

The inequality in the monotonicity axiom is understood in the almost sure sense.

Important examples of coherent risk measures are models of the form

$$\rho(x) = E[X] + \gamma r[X], \tag{1}$$

where $r(\cdot)$ is the upper semideviation of order $p \ge 1$, given in Eq. (2), or a weighted mean-deviation from quantile in Eq. (3):

$$r(X) = E[(X - E[X])_+^p]^{1/p}, \tag{2}$$

$$r_\alpha(X) = \min_\eta E\left[\max\left(\frac{1-\alpha}{\alpha}(X-\eta), \eta - X\right)\right], \quad \alpha \in (0, 1). \tag{3}$$

For these both cases, when $\gamma \in [0, 1]$, the mean-risk model is coherent (Ruszczyński and Shapiro 2006b). It is also worth mentioning that the deviation from quantile (3) is related to the Average (Conditional) Value at Risk (see, Rockafellar and Uryasev 2000, 2002) by the formula [cf. Shapiro et al. 2009, sec. 6.2.4]

$$E[X] + r_\alpha[X] = \mathrm{AVaR}_\alpha[X] = \frac{1}{\alpha}\int_{1-\alpha}^{1} F_X^{-1}(\beta)\, d\beta = \min_\eta \left\{\eta + \frac{1}{\alpha} E\big[(X-\eta)_+\big]\right\}.$$

Here $F_X^{-1}(\cdot)$ is the quantile function of X.

We can formulate the general one-stage portfolio problem with a risk measure $\rho(\cdot)$ as an objective function as follows:

$$\min_{z\in Z} \rho[-R^\top z]. \tag{4}$$

We adapt the convention that the argument of the risk measure $\rho[\cdot]$ represents cost (losses) and that is why we use the minus sign in front of the return rate. A fundamental modeling issue is to choose the risk function $r(\cdot)$ used in this model. For the measures of risk (2) with $p = 1$ and (3), the resulting optimization problem (4) is a linear programming problem, which can be efficiently solved by specialized techniques (Mansini et al. 2003). Parametric methods of Ruszczyński and Vanderbei (2003) allow for generating a family of solutions, corresponding to a range of values of the parameter γ in (1).

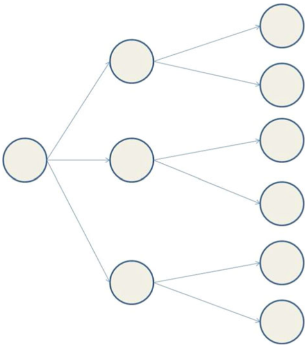
However, if a portfolio optimization model is used in a rolling horizon fashion, as in Matmoura and Penev (2013), with re-balancing in regular time intervals, it makes sense to include the re-balancing action and associated transaction costs into the model. To address this issue in the simplest possible way, a *two-stage model* can be formulated. In this model, an option to re-balance the portfolio at the end of the first period is available. Let us denote by R^t_j the return rate of asset $j = 1, \dots, n$ in stage $t \in \{1, 2\}$. Asset allocations are denoted by n-dimensional vectors z and y, where z_j represents the amount of capital invested in asset j at stage 1, and y_j the amount invested at stage 2. The vector y may depend on the observations gathered in stage 1. The end portfolio value in stage 1 is given by $(\xi^1)^\top z$ and the end value at stage 2 is $(\xi^2)^\top y$, where

$$\xi^t = \mathbb{1} + R^t,$$

with $\mathbb{1}$ denoting the sure outcome of 1. The random vectors ξ^1 and ξ^2 are, in general, dependent. If they have have finite numbers of realizations, the most transparent way is to represent them in a form of a *scenario tree*. An example of such a tree is depicted in Fig. 1. The nodes at levels one and two represent realizations of ξ^1 and ξ^2, respectively. The node at level zero is known as the root node and represents the beginning of the process. Each node at level one represents a different realization of ξ^1. It is connected to a set of children nodes at level two, which represent possible outcomes of ξ^2, following the first stage outcome. With each arc of the tree, a probability is associated. Probabilities of arcs leading to nodes at level one are the probabilities of realizations of ξ^1. The probabilities of arcs leading to nodes at level 2, are conditional probabilities of realizations of ξ^2.

The two-stage portfolio problem allows us to model the re-allocation option within an optimization problem. In stage 1, asset allocations z are to be determined. Then a realization

Fig. 1 Scenario tree



of ξ^1 is observed, and the allocations can be changed to y. In the scenario tree setting, there can be a different value of y for each node at level 1. Finally, the realization of ξ^2 is observed. As a result, the final portfolio value can be calculated. Such an approach, with the use of dynamic measures of risk, has been first developed in Miller (2008) and Miller and Ruszczyński (2011). We formally define the two-stage model in Sects. 2 and 3.

Our intention is to advance these ideas, by exploring new scenario tree generation methods and using higher order measures of risk.

In Sect. 2, several scenario tree generation methods are described. First, a multivariate GARCH model is used to generate an adequate number of scenarios to model the random returns. Then, three different tree construction algorithms are developed: K-means and Two-Step cluster algorithms in forward and backward forms, and a backward multi-facility location algorithm. In order to evaluate the quality of the scenario trees, Monge–Kantorovich transportation model is formulated to compare the probability distributions of the “original” probability distribution (empirical distribution supported on the scenarios generated) with the probability distributions supported on the constructed scenario trees.

In Sect. 3, a two-stage portfolio problem with an option to re-balance is modeled by using higher-order conditional risk measures. A risk-averse multicut method is proposed to solve this model.

In Sect. 4, computational results will be presented to compare scenario generation techniques and performance of the constructed portfolios on portfolios constructed from the components of the Dow Jones Index.

Finally, at the conclusion of this study, an overall summary of our contribution and a list of some possible future research directions will be presented.

2 Scenario tree generation

2.1 Background

A substantial body of literature exists about generating scenario trees for stochastic optimization models. Heitsch and Romisch (2009) proposed a theory-based heuristics for generating scenario trees from an initial set of scenarios, and applied these heuristics in electric power management. Their proposed heuristics have a recursive scenario reduction algorithm and also bundling steps based on forward or backward scenario tree generation methods. They used the stability result in multi-stage stochastic programs from the study in Heitsch et al. (2006) to compare the closeness of the original probability distribution to its scenario tree approximation. The conditions on the initial approximation in applications is constructed from a discrete probability distribution by using a sampling method or from a general probability distribution by using discretization schemes. However, the algorithm can be used as a heuristics for scenario tree generation in other applications.

Hochreiter and Pflug (2007) showed that the problem of obtaining accurate and valuable scenario tree approximations can be viewed as the problem of optimally approximating a given distribution by using a distance function. In that paper, it is found that the best approach is to use the Wasserstein distance in tree approximation. The resulting optimization problem can be formulated as a multi-dimensional facility location problem, and then well-known heuristic algorithms for multi-facility location problems can be applied. They also showed that a scenario tree is constructed as a nested facility location problem to use in multi-stage stochastic programs. A multi-stage stochastic mean-risk financial programming problem is used to test the model. They concluded that if the objective of the approximation is to achieve a controlled matching of certain moments and a controllable coverage of heavy tails, scenario tree generation based on multidimensional facility location will be the best fit.

Our approach builds on these contributions, with the intention to be able to handle huge trees arising in financial applications.

2.2 Scenario generation

Financial data are insufficient for construction optimization models based on empirical distributions alone. It is imperative to generate scenarios that were not observed in practice, but are possible according to statistics. For this purpose, we need a model for scenario generation.

The first step in scenario generation is to obtain data to construct scenarios. In this study, daily stock price data of the 30 companies included in the Dow Jones index are obtained for a period of three years, from September 2, 2008, to November 30, 2011.

Next, in order to model the probabilistic information in the data, an adequate number of scenarios must be generated. As we are interested in an adequate modeling of the tail behavior, multivariate GARCH models appear to be particularly useful. In multivariate GARCH models, the most important issue is the parametrization of the covariance matrix, at a minimum loss of generality. In this study, we will use multivariate GO-GARCH(1,1) (Van Der Weide 2002) model to generate scenarios. Its structure can be summarized as follows.

We assume that the observed vector-valued time series $\{x_t\}$ (of dimension $m = 30$) is a linear combination of unobserved m-dimensional normal vectors $\{y_t\}$ having uncorrelated components, that is,

$$x_t = Zy_t, \quad t = 0, 1, 2, \ldots$$

The square matrix Z is assumed to be constant over time and invertible.

Unobserved components have a diagonal covariance matrix $H_t = \mathrm{diag}\{h_{i,t},\ i = 1, \ldots, m\}$, and thus the covariance matrix of x_t is $V = E\big[x_t x_t^T\big] = ZH_tZ^T$. The crucial component of the model is the evolution of the diagonal elements $h_{i,t}$ of the covariance matrix H_t:

$$h_{i,t+1} = (1 - \alpha_i - \beta_i) + \alpha_i y_{i,t}^2 + \beta_i h_{i,t}, \quad i = 1, \ldots, m, \quad t = 0, 1, 2, \ldots,$$

where the initial matrix $H_0 = I$.

The historical data are used to calculate the least-squares estimates of the matrix Z and the coefficients (α_i, β_i), $i = 1, \ldots, m$.

Once the model is constructed, it can be used to generate an arbitrary number of scenarios. Assuming that the data were collected for the period $t = 0, 1, \ldots, T$, we generate scenarios for times $T + 1, T + 2, \ldots$. In our case, we use only two next steps, that is, $T + 1$ and $T + 2$, to prepare scenarios for ξ^1 and ξ^2.

2.3 Tree generation

Raw scenarios are not suitable for two-stage optimization models, because after stage 1, while deciding about allocations y for the second stage, we would know not only the past return realizations ξ^1, but also the future realizations ξ^2 (see the left part of Fig. 2). Constructing a scenario tree eliminates this problem as it can be seen in the right part of Fig. 2.

There are two different ways to design scenario tree generation methods, *forward* and *backward*. In a forward tree construction, starting at the first stage, one merges selected nodes into clusters and moves forward until the last stage. Forward tree construction is illustrated in Figs. 2 and 3. In a backward tree construction, one starts at the last stage, merges selected

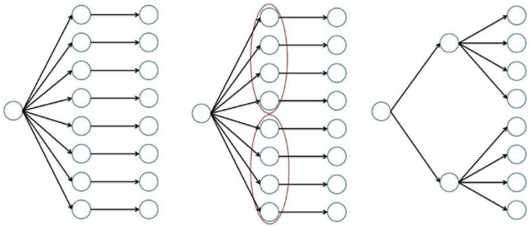


Fig. 2 First-stage clustering in a forward tree construction

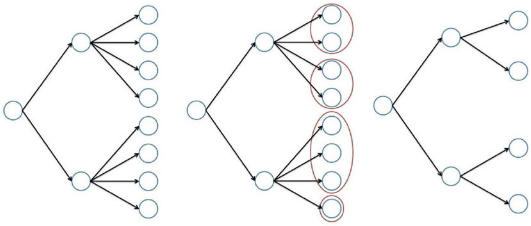


Fig. 3 Second-stage clustering in a forward tree construction

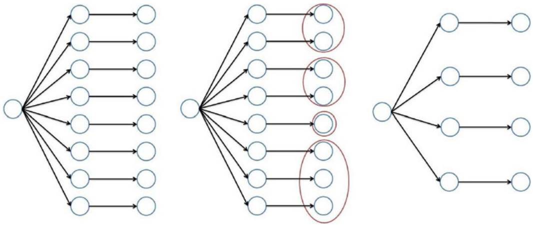


Fig. 4 Second-stage clustering in a backward tree construction

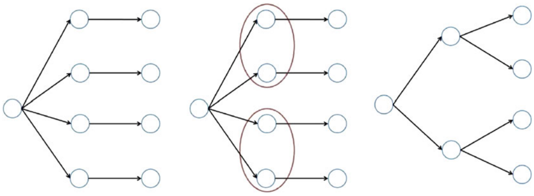


Fig. 5 First-stage clustering in a backward tree construction

nodes into clusters and this will join all their predecessors as well. In this way, we move backward until the first stage. Backward tree construction is illustrated in Figs. 4 and 5. In scenario tree generation, it is important to maintain probability information while reducing the number of scenarios, that is, to assign to a scenario representing a group of scenarios the sum of their probabilities.

Since the quality of multi-stage stochastic optimization models depends heavily on the quality of the underlying scenario model, this study will focus on constructing a scenario tree for two-stage portfolio optimization problem, so that portfolio optimization problem with rebalancing can be solved in a more time-efficient way.

Five different scenario generation methods are used in this study: *K*-means and two-step clustering methods in forward and backward versions, and a backward scenario generation method based on the idea of a multi-facility location problem. Then, these five models will be compared by using a mass-transportation model.

2.4 Multi-facility-location-based scenario tree generation method

We will use a hybrid method which is composed of two different algorithms within a backward scenario tree generation method. In the first part, a nearest centroid based heuristic is used to form a given number of clusters in the second stage.

Choosing initial centroids is very important in a clustering algorithm since bad initialization can lead to poor results. The fastest and easiest way to choose initial centroids is pure random. However, it will very likely lead to poor results since it can choose these initial centroids close to each other. In the initialization part of our algorithm, we want to choose the initial centroids that are spread out from each other. We will use a very similar initialization approach as in Arthur and Vassilvitskii (2007). Nearest centroid based heuristic is explained in Algorithm 1.

Algorithm 1 Nearest centroid heuristic

```
Initialization:
The first cluster mean is chosen randomly from the data points.
Next cluster mean is chosen from the remaining data points with a probability given to each point based on
squared distance from its closest existing cluster mean.
Iteration:
Find the center for each cluster.
For each node, calculate the distance to the each center constructed before. If there is a closer center for that
node, reassign the selected node to its closest center.
Continue until no new reassignments.
```

The distance measure used in this algorithm is as follows:

$$d_{ij} = \sqrt{\sum_{n=1}^{30} \left[(r^1(i,n) - r^1(j,n))^2 + (r^2(i,n) - r^2(j,n))^2 \right]}, \tag{5}$$

where r^1 represents the return rate at the first stage, r^2 represents the return rate at the second stage, i and j represent the scenarios, and n represents the security.

In the second part of this hybrid method, in order to aggregate first-stage nodes, a multi–facility location problem is formulated.

We denote by J the total number of scenarios (after the initial greedy aggregation) and by $I < J$ the desired number of first-stage nodes. For $i, j = 1, \ldots, J$ we use d_{ij} to denote the distance calculated according to formula (5). The multi-facility location problem can be formulated as follows:

$$\begin{aligned}
\min & \sum_{i=1}^{J} \sum_{j=1}^{J} d_{ij} x_{ij} \\
\text{s.t.} & \sum_{i=1}^{J} x_{ij} = 1, \quad \forall j = 1, \ldots, J, \\
& \sum_{i=1}^{J} v_i = I, \\
& x_{ij} \le v_i, \quad \forall i, j = 1, \ldots, J, \\
& \text{all } x \text{ and } v \text{ variables are binary.}
\end{aligned}$$

In the problem above, v_i is the decision to use scenario i as a first-stage node, and x_{ij} represents the assignment of scenario j to node i. It is a large-scale problem, and we solve it by a greedy method using the linear programming relaxation, in which the binary variables x and v are allowed to take any values in [0,1]. After a relaxed problem is solved, all $v_i = 1$ are permanently fixed; if none is equal to 1, we choose the one that is closest to 1, and fix it at 1. After that, a reduced problem with a smaller number of variables is solved, etc. Our experience indicates that this procedure does not lead to significant errors and allows for processing large data sets.

2.5 *K*-means scenario tree generation method

K-means is a clustering method in which a set of n observations $S = \{x_1, x_2, \ldots, x_n\}$ are partitioned into k clusters ($k \le n$). Each scenario is a d-dimensional vector, where $d = 30$ in this study (the number of Dow Jones stocks). Let $S = S_1 \cup S_2 \cup \cdots \cup S_k$ be the partition of the set; the objective of the *K*-means method is to minimize the sum of squares within clusters:

$$\begin{aligned} \min_{S_1,\ldots,S_k} \quad & \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - m_i(S_i) \right\|^2 \\ \text{subject to} \quad & S_1 \cup S_2 \cup \cdots \cup S_k = S, \\ & S_i \cap S_j = \emptyset \text{ if } i \neq j. \end{aligned}$$

In the problem above, $m_i(S_i)$ is the mean of the points in S_i.

The first k initial means are randomly selected from the scenario set. Then, every scenario is associated with the nearest mean. Next, the centroid of each cluster becomes the new mean for that cluster. Finally, when no new centroids are created, the method stops.

The *K*-means algorithm is used to construct scenario trees in two-stage stochastic portfolio problem is given in Algorithm 2.

Algorithm 2 *K*-means

```
First-Stage
Initialization: Given an initial set of $k$ means $m_1^{(1)}, m_2^{(1)}, \ldots, m_k^{(1)}$
Assignment Step: Assign each observation to the cluster with the closest mean.
    $S_i^{(t)} = \left\{ x_j : \left\| x_j - m_i^{(\tau)} \right\| \le \left\| x_j - m_{i*}^{(\tau)} \right\|, \ \forall i^* = 1, \ldots, k \right\}$
Update Step: Calculate the new means to be the centroid of the observations in the cluster.
    $m_i^{(\tau+1)} = \frac{1}{\left| S_i^{(\tau)} \right|} \sum_{x_j \in S_i^{(\tau)}} x_j$
Stop when the assignments do not change.
Second-Stage
Let $e_i$ be the number of children nodes for each cluster in the first-stage, $i = 1, \ldots, k$
for $i = 1$ to $k$ do
    Initialization: From $e_i$ observations select an initial set of $l_i$ means $m_1^{(1)}, m_2^{(1)}, \ldots, m_{l_i}^{(1)}$.
    Assignment Step: Assign each observation to the cluster with the closest mean.
        $S_i^{(\tau)} = \left\{ x_j : \left\| x_j - m_i^{(\tau)} \right\| \le \left\| x_j - m_{i*}^{(\tau)} \right\|, \ \forall i^* = 1, \ldots, l_i \right\}$
    Update Step: Calculate the new means to be the centroid of the observations in the cluster.
        $m_i^{(\tau+1)} = \frac{1}{\left| S_i^{(\tau)} \right|} \sum_{x_j \in S_i^{(\tau)}} x_j$
Calculate the probabilities and conditional of each first-stage and second-stage scenarios, respectively.
```

In the K-means model, Euclidean distance is used as a metric, and the number of first-stage clusters k, and second stage clusters l, are input parameters. Therefore, good results from this method depend on the appropriate choice of k and l.

2.6 Two-step clustering

IBM's (SPSS Inc 2004) two-step clustering scenario tree generation method is designed for very large data sets. The method requires only one pass of the data, and has two major steps. In the first step, scenarios are grouped into many small preclusters. Then, these preclusters are clustered into a desired number of clusters. The method is explained in Algorithm 3.

Algorithm 3 Two-Step Clustering Algorithm

```
First-Stage
Step 1: Formation of Preclusters
Form preclusters by making each scenario a cluster in itself.
For each scenario, find if it should be merged with a previously formed precluster or there should be a new precluster based on a selected distance measure considering the weights (agglomerative hierarchical clustering).
When preclustering is complete, all scenarios in one precluster become a single entity.
Step 2: Clustering of Preclusters
Take the preclusters obtained in Step 1 as an input, and group them into desired number of clusters by using agglomerative hierarchical clustering considering the weights.
Second-Stage
Let $e_i$ be the number of children nodes for each cluster in the first-stage, $i = 1, \dots, k$
for $i = 1$ to $k$ do
  Apply Step 1 & 2 explained above to each cluster $i$ with $e_i$ observations.
Calculate the probabilities and conditional probabilities of the first-stage and second-stage scenarios.
```

As it is in K-means model, Euclidean distance is used as a metric, and the number of clusters at the first stage k, and at the second stage l, are input parameters.

2.7 Quality of scenario trees

No universally good scenario tree generation method exists. Therefore, we need a measure to evaluate the quality of the trees generated. One possibility is use error values of methods or convergence of the objective function values can be used. Pflug defined an approximation error in Pflug (2001) as follows. Let the original problem be

$$\min_{x \in X} \quad F(x; \tilde{\xi}),$$

and the tree-based problem be

$$\min_{x \in X} \quad F(x; \tilde{\eta}),$$

with the solution x^*. Then, the approximation error can be calculated as follows:

$$e_f(\tilde{\eta}, \tilde{\xi}) = F(x^*; \tilde{\xi}) - \min_x F(x; \tilde{\xi}) \geq 0.$$

However, to calculate this error function is not easy because we need to find the true objective value for a given solution x, and the true optimal solution, which is practically impossible. Therefore, in this study a *mass transportation model* will be formulated to compare the original distribution with the distribution in the scenario tree. The model evaluates the *Monge–Kantorovich metric*.

In the model below, we use the following data:

p_i—the probability of path i in the original distribution

q_j—the probability of path j in the tree

d_{ij}—euclidean distance between path i in the original distribution and path j in the tree based on Formula (5)

We look for the cheapest way to move the probability mass from the "original" distribution (the empirical distribution supported on the generated scenarios) and the distribution on the tree. The variables f_{ij} represent the probability mass to be moved from path i in the original distribution to path j in the tree:

$$\begin{aligned} \min \quad & \sum_i \sum_j d_{ij} f_{ij} \\ \text{subject to} \quad & \sum_j f_{ij} = p_i, \quad \forall\, i, \\ & \sum_i f_{ij} = q_j, \quad \forall\, j, \\ & f_{ij} \geq 0, \quad \forall i, j. \end{aligned} \tag{6}$$

The optimal value of this problem will be used as a measure of the quality of the tree.

3 Two-stage portfolio optimization problem with risk measures

3.1 General formulation

The problem (4), when the risk function is a semideviation of order 1, or weighted mean-deviation from quantile, can be formulated and solved as a linear programming problem (see Mansini et al. 2003; Ruszczyński and Vanderbei 2003 and also Miller and Ruszczyński 2011 for additional insights). We shall focus on the two-stage version, as outlined in Sect. 1. In what follows, we assume that the sequence of returns (R^1, R^2) has a distribution supported on a scenario tree.

Let p^1 be the first-stage probability vector, with p_i^1 denoting the probability of outcome R_i^1 at the first stage, and let p_i^2 be the conditional probability vector for each node i in the first level, with p_{il}^2 denoting the probability of moving to node l in the second-stage from node i. The corresponding realizations R_{il}^2 with probabilities p_{il}^2 represent the conditional distribution of R^2, given that the realization R_i^1 occurred.

The two-stage portfolio optimization problem can be formulated as follows

$$\min_{z \in Z} \rho_1\big[V(z)\big], \tag{7}$$

where $V(z)$ is a random variable representing the *conditional risk measure* at stage 2. Its realizations, $V_i(z)$, are the optimal values of the second stage problem:

$$V_i(z) = \min_{y_i \in Y_i(z)} \rho_{2i}\big[-(\xi_i^2)^\top y_i\big]. \tag{8}$$

In Eq. (8), the argument of the conditional measure of risk $\rho_{2i}[\cdot]$ is the negative of the total wealth after stage 2. The feasible set $Y_i(z)$ depends on the initial investments z and the realization of the returns in the first stage, which is represented by the branch index i. In the absence of transaction costs, it would have the form resulting from the simple cash balance equation:

$$Y_i = \left\{ y \in Y : \left(\xi^1\right)^\top z = \mathbb{1}^\top y \right\}.$$

However, the presence of transaction costs makes this formulation insufficient. In the next section, we describe a detailed model for the case of a mean–semideviation measure of risk.

The use of composition risk measures in (7)–(8) follows form the general theory of time-consistent dynamic measures or risk (see, Ruszczyński 2010 and the references therein).

For the main concepts and results of the theory of dynamic measures of risk, see Artzner et al. (2007), Cheridito et al. (2006), Föllmer and Penner (2006), Fritelli and Scandolo (2006), Pflug and Römisch. (2007), Riedel (2004) and Ruszczyński and Shapiro (2006a) and the references therein.

3.2 The mean-semideviation model

In this section, risk averse two-stage portfolio problem will be formulated with the mean-semideviation risk function of order r.

Indices:

$i \in \{1, \ldots, I\}$—First-stage scenarios,
$l \in \{1, \ldots, L(i)\}$—Second-stage scenarios for each first-stage scenario,
$j \in \{1, \ldots, n\}$—Securities.

Parameters:

p_i—probability of a first-stage scenario i,
p_{il}—conditional probability of second-stage scenario l after the first-stage scenario i,
R_{ji}—return rate of security j in first-stage scenario i,
R_{jil}—return rate of security j in second-stage scenario l after the first-stage scenario i,
ε_j—relative transaction cost of security j,
γ—risk aversion constant,

First-Stage Variables:

z_j—amount invested in security j in the first-stage,
ξ—auxiliary variables representing shortfalls at the first-stage,
u—expectation at the first-stage,

Second-Stage Variables:

y_{ji}—new position in security j in scenario i after the first-stage,
b_{ji}—amount spent to buy security j in scenario i after the first-stage,
s_{ji}—value of security j sold in scenario i after the first-stage,
σ—auxiliary variables representing shortfalls at the second-stage,
m—conditional expectation at the second-stage.

We assume that the initial capital is 1, and thus (in the simplest version)

$$z \in Z = \left\{ z \in \mathbb{R}^n : \sum_{j=1}^{n} z_j = 1,\ z_j \geq 0,\ j = 1, \ldots, n \right\};$$

more complex restrictions on the initial investments are possible as well, as long as they define a polyhedral set Z. In order to estimate the (relative) transaction costs, the following bid-ask spread formula is used:

$$\varepsilon_j = \frac{(\text{AskPrice}_j - \text{BidPrice}_j)/2}{\text{AskPrice}_j}, \quad j = 1, \ldots, n.$$

This formula assumes that a "fair price" is half-way between the bid and the ask prices, and ignores transaction costs due to the price impact of large trades. The link between the first-stage variables z and the second-stage variables y is provided by the cash balance equation:

$$y_{ji} = (1 + R_{ji})z_j + (1 - \varepsilon_j)b_{ji} - (1 + \varepsilon_j) \cdot s_{ji}, \tag{9}$$

in which we symmetrically assign transaction costs to the sales and purchases. The first-stage problem (7) with mean-semideviation risk functions of order $r \geq 1$ can be now formulated as follows:

$$\min_{z \in Z} \quad \sum_{i=1}^{I} p_i V_i(z) + \gamma \left(\sum_{i=1}^{I} p_i \left(\max(V_i(z) - p^\top V(z), 0) \right)^r \right)^{1/r},$$

where $V_i(z)$ is the optimal value of ith second-stage problem (8).

The first-stage problem can be rewritten as follows:

$$\begin{aligned} \min \quad & u + \gamma \Big(\sum_{i=1}^{I} p_i \xi_i^r \Big)^{1/r} \\ \text{s.t.} \quad & u = \sum_{i=1}^{I} p_i V_i(z), \\ & u \leq V_i(z) + \xi_i, \quad i = 1, \ldots, I, \\ & \xi \geq 0, \\ & z \in Z. \end{aligned} \tag{10}$$

The second-stage problem (8) with mean-semideviation risk function of order $r \geq 1$ is formulated in scenario i as follows:

$$\begin{aligned} \min \quad & -m_i + \gamma \Big(\sum_{l=1}^{L(i)} p_{il} \sigma_{il}^r \Big)^{1/r} \\ \text{s.t.} \quad & y_{ji} = (1 + R_{ji})z_j + (1 - \varepsilon_j)b_{ji} - (1 + \varepsilon)s_{ji}, \quad j = 1, \ldots, n, \\ & \sum_{j=1}^{n} \big(b_{ji} - s_{ji} \big) = 0, \\ & W_{il} = \sum_{j=1}^{n} (1 + R_{jil}) y_{ji}, \quad l = 1, \ldots, L(i), \\ & m_i = \sum_{l=1}^{L(i)} p_{il} W_{il}, \\ & m_i \leq W_{il} + \sigma_{il}, \quad l = 1, \ldots, L(i), \\ & b \geq 0, \quad s \geq 0, \quad \sigma \geq 0. \end{aligned} \tag{11}$$

The optimal value of this problem is denoted by $V_i(z)$. In a more general formulation, we may use different risk-aversion parameters γ in the first-stage problem (10) and in the second stage problems (11), making them dependent on the scenario i. We can also add to problem (11) additional restrictions on the allocations y_{ji}, $j = 1, \ldots, n$, as long as they define a nonempty polyhedral set. In particular, keeping the first-stage investments unchanged, that is, setting $y_{ji} = (1 + R_{ji})z_j$, $j = 1, \dots, n$, should be feasible for problem (11).

3.3 Risk-averse multicut method for higher-order conditional measures of risk

If $r = 1$, the problems (10) and (11) (for $i = 1, \dots, I$) can be put together into a large-scale linear programming problem, with V_i, $i = 1, \dots, I$, treated as variables [given by the formula in the first row of (11)]. The dimension of this problem, however, is of order $I \times (L + n)$ variables and constraints, which becomes unmanageable for realistic sizes of scenario trees. If $r > 1$, an additional complication arises from the fact that we have to deal with a large-scale *nonlinear* optimization problem.

We shall, therefore, develop a *decomposition method* for solving problem (10)–(11), based based on Benders decomposition. In order to describe this method, we have to recall the *dual representation* of measures of risk. For a coherent measure of risk $\rho : \mathscr{X} \to \Re$, where $\mathscr{X}$ is the vector space of random variables on a finite probability space having I elementary events, a closed convex set $\mathscr{A}$ of probability measures on this space exists, such that

$$\rho(V) = \max_{\mu \in \mathscr{A}} \sum_{i=1}^{I} \mu_i V_i, \quad V \in \mathscr{X}. \tag{12}$$

This representation, first proved in Artzner et al. (1999), is valid in a much more general setting as well (see, Ruszczyński and Shapiro 2006b; Shapiro et al. 2009 and the references therein). The set $\mathscr{A}$ is the subdifferential of $\rho(\cdot)$ at zero. Analytical expressions for the sets $\mathscr{A}$ for popular measures of risk (including the mean-semideviation measure) are available (see Shapiro et al. 2009).

Owing to (12), the first-stage problem (7) becomes

$$\min_{z \in Z} \max_{\mu \in \mathscr{A}} \sum_{i=1}^{I} \mu_i V_i(z), \tag{13}$$

Two main issues arise from this formulation. First, solving the "max" problem above is hard by using all of the elements in $\mathscr{A}$, especially, when $r > 1$. Second, there is no easy expression for $V_i(z)$, $i = 1, \dots, I$, which are optimal values of problems (11).

In order to handle the first issue, rather than using $\mathscr{A}$, it approximation from within, $\text{conv}\big(\{\mu^0, \mu^1, \dots, \mu^{k-1}\}\big)$ will be used. Here $\text{conv}(C)$ denotes the convex hull of a set C, and $\mu^1, \dots, \mu^{k-1}$ are elements of $\mathscr{A}$ collected in iterations $1, \dots, k-1$ of the method. For μ^0 we substitute the nominal probability distribution p, which is an element of $\mathscr{A}$ for all practically relevant measures of risk, including the mean–semideviation measure.

We construct an approximation of problem (13) as

$$\min_{z \in Z} \max_{\kappa = 0,1,\dots,k-1} \sum_{i=1}^{I} \mu_i^{\kappa} V_i(z).$$

It is an approximation from below, because the maximum is evaluated over a subset of $\mathscr{A}$ rather than over $\mathscr{A}$. Equivalently, the problem above can be written as *linear* programming problem:

$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & \alpha \geq \sum_{i=1}^{I} \mu_i^{\kappa} V_i(z), \quad \kappa = 0, 1, \dots, k-1. \end{aligned}$$

It is essential to stress that when new points μ^κ are added, then these approximations become more and more accurate, leading eventually to the solution of the original problem, as in standard cutting-plane methods of convex optimization.

To deal with the second issue, the unknown functions $V_i(z)$ will be replaced with piecewise linear convex functions constructed from *cuts*, derived from the solutions of subproblems (11) at earlier iterations. This is a standard way of dealing with parameter dependent subproblems, similar to expected-value two-stage problems (see, Ruszczyński 2003 and the references therein). In general, each cut is an inequality

$$V_i(z) \geq \hat{v}_i^\kappa + \left(g_i^\kappa\right)^\top \left(z - z^\kappa\right),$$

where $\hat{v}_i^\kappa$ is an optimal value of problem (11) for scenario i in iteration κ of the method, with $z = z^\kappa$. The subgradient $g_{i\kappa}$ can be calculated from the Lagrange multipliers π_i^κ associated with the constraints of problem (11) involving the parameters z:

$$g_{ij}^\kappa = (1 + R_{ji})\pi_{ji}^\kappa. \tag{14}$$

The reader may consult (Miller and Ruszczyński 2011) for details of the cut construction in two-stage risk-averse linear programming, which is identical in our case.

The algorithm for risk-averse multicut method is presented below.

Risk-averse multicut method

Step 0: Set $k = 1$.
Step 1: Solve the master problem,

$$\begin{aligned} \min_{z,v,\alpha} \quad & \alpha \\ \text{s.t.} \quad & \alpha \geq \sum_{i=1}^{I} \mu_i^\kappa v_i, \quad \kappa = 0, 1, \ldots, k-1, \\ & v_i \geq \bar{v}_{i\kappa} + g_\kappa^\top (z - z^\kappa), \quad \kappa = 1, \ldots, k-1, \quad i = 1, \ldots, I, \\ & z \in Z, \quad v \geq v_{\min}. \end{aligned}$$

Denote the solution by z^k, α^k, v^k.
Step 2: For each $i = 1, \ldots, I$:
Solve the second-stage problem (11) and let $\hat{v}_i^k$ be its optimal value and π_{ji}^k the Lagrange multiplier associated with the re-balancing constraint for security j in constraint (9). Then, calculate g_κ by using Eq. (14).
Step 3: Calculate $\rho_1^k = \rho_1\left[\hat{v}^k\right]$ and $\mu^k \in \partial\rho_1\left[\hat{v}^k\right]$.
Step 4: If $\rho_1^k = \alpha^k$, then stop; otherwise, increase k by 1 and go to Step 1.

In the risk-averse multicut method we assume that the set Z is compact, and that we know lower bounds $\hat{v}_{\min}$ for the optimal values of each second stage problem.

From general convergence results in Ruszczyński (2006), convergence of this method is finite with the mean-semideviation risk function of order 1. If $r \geq 2$, convergence still can be proved, by the general properties of cutting plane methods for convex programming (see, e.g., Ruszczyński 2006, Thm 7.7).

The calculation of μ in Step 3 depends on the risk measure applied. For mean-semideviation risk function of order r, the subgradient can be calculated as follows. We calculate the shortfall values

$$\sigma_i = \max\Big(0,\, \hat{v}_i^k - \sum_{m=1}^{I} p_m \hat{v}_i^k\Big), \quad i = 1, \ldots, I,$$

and the semideviation of order r,

$$\varphi = \Big(\sum_{i=1}^{I} p_i (\sigma_i)^r\Big)^{1/r}.$$

Then, by applying the rules of subdifferential calculus (see, e.g., Ruszczyński 2006, sec. 2.5), we can obtain $\mu^k \in \partial\rho_1\big[\hat{v}^k\big]$ from the following formula:

$$\mu_i^k = p_i\left[1 + \frac{\gamma}{(\varphi)^{r-1}}\Big((\sigma_i)^{r-1} - \sum_{m=1}^{I}(\sigma_m)^{r-1} p_k\Big)\right], \quad i = 1, \ldots, I. \tag{15}$$

4 Computational results

First, daily returns of Dow Jones companies' from September 2, 2008 to November 30, 2011 were used to calibrate a multivariate GO-GARCH(1,1) model of the returns. The model was used to generate a large number of scenarios for the next two days. The generated scenarios were used to construct two-stage scenario trees by employing all five scenario tree generation methods discussed in Sect. 2. The methods were compared in terms of their time and solution quality. The results in Table 1 show that two-step clustering methods are slightly faster in terms of CPU times compared to other methods.

Because of computational complexity, for the purpose of comparing tree generation methods, the sizes of the scenario trees were restricted to 10,000 and 20,000 scenarios. Table 2 contains the comparison of the quality of scenario trees obtained by different methods: the optimal value of the Monge–Kantorovich metric given by (6). The results show that tree quality is stable for each scenario tree generation technique. However, the multi-facility location clustering method is better with scenario trees constructed from both 10,000 and 20,000

Table 1 CPU Time (in secs) of scenario tree generation methods

Original scenario size	Tree scenario size	*K*-means (forward)	*K*-means (backward)	Two-step (forward)	two-step (backward)	Multi-facility Location
80,000	5,000	675.2	682.1	628.2	621.4	850.7
80,000	10,000	1,210.8	1,272.6	1,152.4	1,145.3	1,675.4

Table 2 Accuracy of scenario tree generation methods

Original scenario size	Tree scenario size	*K*-means (forward)	*K*-means (backward)	Two-step (forward)	Two-step (backward)	Multi-facility Location
10,000	500	0.00719	0.00735	0.00859	0.00875	0.00705
20,000	500	0.00718	0.00733	0.00855	0.00869	0.00699

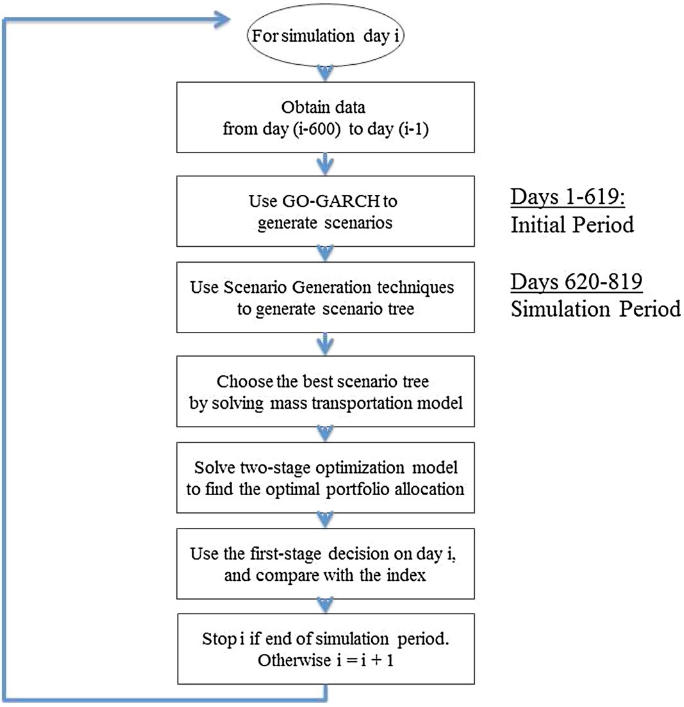

For simulation day i
Obtain data from day (i-600) to day (i-1)
Use GO-GARCH to generate scenarios
Use Scenario Generation techniques to generate scenario tree
Choose the best scenario tree by solving mass transportation model
Solve two-stage optimization model to find the optimal portfolio allocation
Use the first-stage decision on day i, and compare with the index
Stop i if end of simulation period. Otherwise i = i + 1
Days 1-619: Initial Period
Days 620-819 Simulation Period

Fig. 6 Simulation analysis

scenarios. Therefore, in the next part of the study, we used the multi-facility location-based method to generate scenario trees in a rolling horizon fashion.

In the next part of the study, a simulation analysis was carried out. Each day, the preceding 619 days of data were used to calibrate a multivariate GO-GARCH(1,1) model. The model was then used to generate 80,000 scenarios for the following two days. The multi-facility location clustering method was used to construct a scenario tree for the next two days. Then the tree model with conditional measures of risk was solved, the investments were re-balanced, and the method continued. On the next day, new return data were available, new scenario trees were generated, new models solved, etc. The steps of the simulation study are depicted in Fig. 6.

The simulation study had two objectives. First, we compared the two-stage portfolio model with the static model where $b_{ji} = s_{ji} = 0$. Based on the cumulative wealth graphs in Figs. 7 and 8, we can say that two-stage portfolio model performs better than the static model for both mean-semideviation risk functions of order 1 and 2. This was partly due to the reduced volume of trades, which resulted in significantly lower transaction costs, but also to a better portfolio composition.

Secondly, we compared two-stage portfolio models with the mean-semideviation risk measures of first-order and higher orders (2 and 3) with static minimum variance model where $b_{ji} = s_{ji} = 0$. In each case we used fixed $\gamma = 0.9$ and bid-ask spread transaction costs. The two-stage portfolio optimization problem was solved with the risk-averse multi-cut method, implemented in MATLAB with the CPLEX solver. We observed that there is no significant difference between two-stage portfolio problem with mean-semideviation risk measure of second order and third order. As we can see from Figs. 8 and 9, using the second-

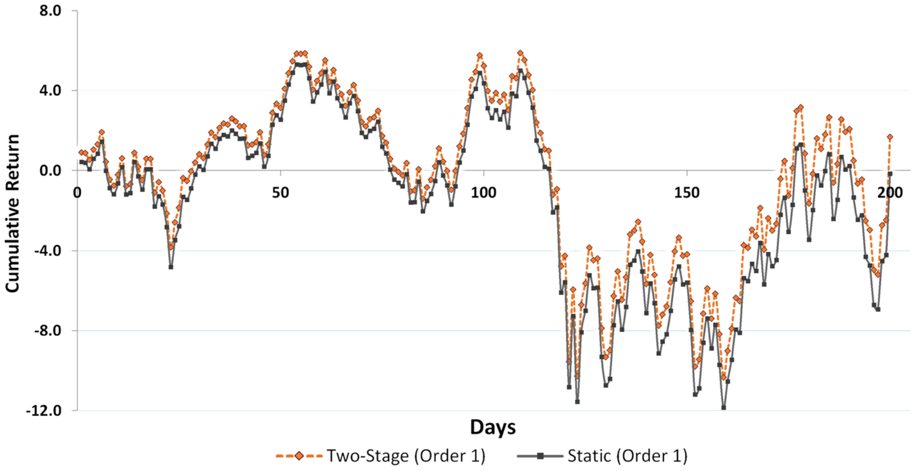


Fig. 7 Performance of the static and two-stage portfolios with mean-semideviation (Order 1)

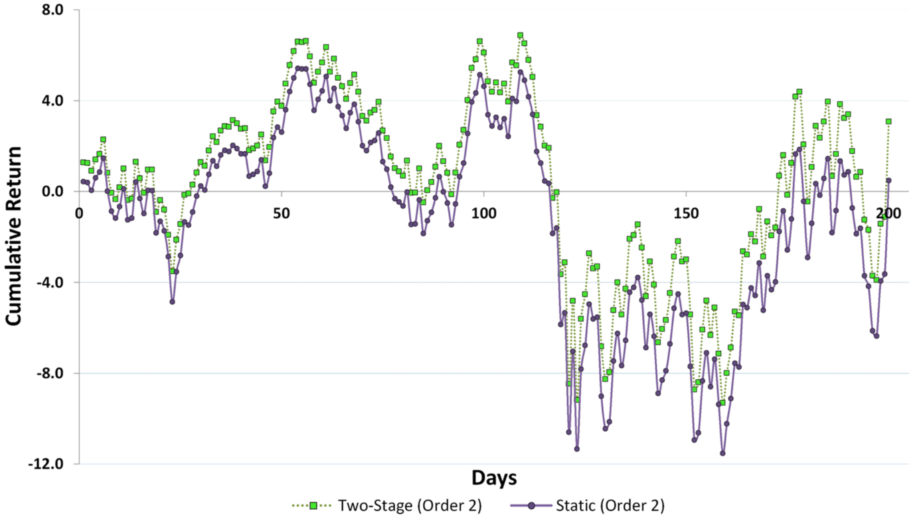


Fig. 8 Performance of the static and two-stage portfolios with mean-semideviation (Order 2)

order methods leads to significant improvements in cumulative wealth trajectories compared to first order and Markowitz's minimum variance models. This is consistent with the findings of Matmoura and Penev (2013), where other higher-order risk measures were employed (with a static model and no transaction costs).

Finally, we can compare the performance of the portfolios with mean-semideviation of order 1 and 2 with Dow Jones Index. Based on the following graph, the simulation analysis shows that portfolio with mean-semideviation of order 2 performs better than other portfolios.

Next, we compare β values of the portfolios with mean-semideviation of order 2 when risk aversion constant is either $\gamma = 0.9$ or $\gamma = 0.5$. β value of portfolio (0.92) with $\gamma = 0.5$ is slightly more than β value of portfolio (0.85) with $\gamma = 0.9$. This means both portfolios generally move in the same direction as the benchmark (Dow Jones). However, movement of

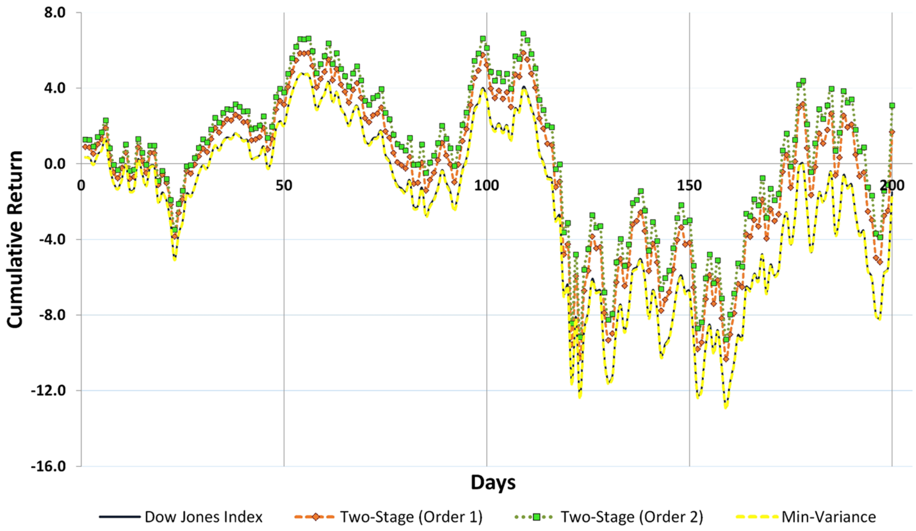


Fig. 9 Performance of the two-stage portfolios and Dow Jones Index

the mean-semideviation portfolio of order 2 with $\gamma = 0.9$ is generally less than the movement of the mean-semideviation portfolio of order 2 with $\gamma = 0.5$.

5 Conclusion

In this study, two-stage portfolio models with higher-order conditional risk measures are studied. First, adequate number of scenarios are generated to model the probabilistic information on random data. Next, scenario trees are constructed by different methods, and the best one is chosen based on the distance between the probability distribution on the scenario tree and the empirical distribution on the raw scenarios. It is found that the two-step forward clustering method is most efficient in terms of the CPU time, because it passes over the data just once. However, that multi-facility location clustering method is the most accurate, in terms of the Monge-Kantorovich metric.

Next, conditional mean-semideviation risk functions of first-order and higher-orders (Order 2 and 3) are used to formulate the risk-averse two-stage portfolio problem on the trees generated from the multi-facility location clustering method. The problems are solved by a generic risk averse multicut algorithm for any higher-order risk function. The results show that the portfolio allocations for mean-semideviation models of first-order and higher-orders are similar. However, portfolios with the mean-semideviation of second-order and third-order perform better compared to mean-semideviation of first-order, minimum variance model, and the Dow Jones index. All two-stage models outperform the static model.

References

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. SIAM*, pp. 1027–1035.

Artzner, P., Delbaen, F., Eber, J. M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, *9*, 203–228.

Artzner, P., Delbaen, F., Eber, J. M., Heath, D., & Ku, H. (2007). Coherent multiperiod risk adjusted values and Bellmans principle. *Annals of Operations Research*, *152*, 5–22.

Cheridito, P., Delbaen, F., & Kupper, M. (2006). Dynamic monetary risk measures for bounded discrete-time processes. *Electronic Journal of Probability*, *11*, 57–106.

Föllmer, H., & Penner, I. (2006). Convex risk measures and the dynamics of their penalty functions. *Statistics Decisions*, *24*(1), 61–96.

Fritelli, M. M., & Scandolo, G. (2006). Risk measures and capital requirements for processes. *Mathematical Finance*, *16*, 589–612.

Heitsch, H., & Romisch, W. (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, *118*(2), 371–406.

Heitsch, H., Romisch, W., & Strugarek, C. (2006). Stability of multistage stochastic programs. *SIAM Journal on Optimization*, *17*, 511–525.

Hochreiter, R., & Pflug, G. Ch. (2007). Financial scenario generation for stochastic multi-stage decision processes as facility location problems. *Annals of Operations Research*, *152*, 257–272.

Kijima, M., & Ohnishi, M. (1993). Mean-risk analysis of risk aversion and wealth effects on optimal portfolios with multiple investment opportunities. *Annals of Operations Research*, *45*, 147–163.

Mansini, R., Ogryczak, W., & Speranza, M. G. (2003). LP solvable models for portfolio optimization: A classification and computational comparison. *IMA Journal of Management Mathematics*, *14*(3), 187–220.

Markowitz, H. M. (1952). Portfolio selection. *Journal of Finance*, *7*, 77–91.

Matmoura, Y., & Penev, S. (2013). Multistage optimization of option portfolio using higher order coherent risk measures. *European Journal of Operational Research*, *227*(1), 190–198.

Miller, N. L. (2008). Mean-risk portfolio optimization problems with risk-adjusted measures. PhD thesis, Rutgers University.

Miller, N., & Ruszczyński, A. (2011). Risk-averse two-stage stochastic linear programming: Modeling and decomposition. *Operations Research*, *59*, 125–132.

Ogryczak, W., & Ruszczyński, A. (1999). From stochastic dominance to mean-risk models: Semideviations as risk measures. *European Journal of Operational Research*, *116*(1), 33–50.

Ogryczak, W., & Ruszczyński, A. (2001). On consistency of stochastic dominance and mean-semideviation models. *Mathematical Programming*, *89*, 217–232.

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

Pflug, G. Ch. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, *89*, 251–271.

Pflug, G. Ch., & Römisch, W. (2007). *Modeling, measuring and managing risk*. Singapore: World Scientific.

Riedel, F. (2004). Dynamic coherent risk measures. *Stochastic Processes and Their Applications*, *112*, 185–200.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value at risk. *The Journal of Risk*, *2*, 21–41.

Rockafellar, R. T., & Uryasev, S. (2002). Conditional value at risk for general loss distributions. *Journal of Banking and Finance*, *26*, 1443–1471.

Ruszczyński, A. (2003). Decomposition methods. In A. Ruszczyński & A. Shapiro (Eds.), *Stochastic Programming* (pp. 141–211). Amsterdam: Elsevier.

Ruszczyński, A. (2006). *Nonlinear Optimization*. Princeton: Princeton University Press.

Ruszczyński, A. (2010). Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, *125*(2), 235–261.

Ruszczyński, A., & Vanderbei, R. (2003). Frontiers of stochastically nondominated portfolios. *Econometrica*, *71*, 1287–1297.

Ruszczyński, A., & Shapiro, A. (2006a). Conditional risk mappings. *Mathematical Operations Research*, *31*, 544–561.

Ruszczyński, A., & Shapiro, A. (2006b). Optimization of convex risk functions. *Mathematical Operations Research*, *31*, 433–452.

Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). *Lectures on stochastic programming: Modeling and theory. SIAM*.

SPSS Inc. (2004).TwoStep Cluster Analysis. Technical Report.

Van Der Weide, R. (2002). GO-GARCH: A multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, *17*(5), 549–564.