Author(s): Mäkinen, Raino A. E.; Toivanen, Jari

Title: Monte Carlo Expected Wealth and Risk Measure Trade-Off Portfolio Optimization

Year: 2024

Version: Published version

SIAM J. FINANCIAL MATH.
Vol. 15, No. 2, pp. SC41–SC53

Short Communication: Monte Carlo Expected Wealth and Risk Measure Trade-Off Portfolio Optimization*

Raino A. E. Mäkinen† and Jari Toivanen†

Abstract. A multiperiod portfolio optimization is described with Monte Carlo sampled risky asset paths under realistic constraints on the investment policies. The proposed approach can be used with various asset and risk models. It is flexible as it does not require dynamic programming or any transformations. As examples, the variance and semivariance risks are considered leading to mean-variance and mean-semivariance formulations, respectively. A quasi-Newton method with an adjoint gradient computation can solve the resulting optimization problems efficiently. Numerical examples show efficient frontiers together with optimal asset allocations computed for mean-variance and mean-semivariance portfolios with two and five assets.

Key words. dynamic portfolio management, mean-variance optimization, mean-semivariance optimization, constrained optimization, Monte Carlo simulation

MSC codes. 65C05, 90C31, 90C55, 91G10, 91G60

DOI. 10.1137/23M1624439

1. Introduction.

The single-period mean-variance optimization introduced by Markowitz [24] is the classical way to select investment portfolios. Dynamic and multiperiod generalization of this approach offers a more realistic model for portfolios as they can incorporate more constraints for investments as well as time- and wealth-dependent asset allocations. These generalizations offer robust asset allocations which are insensitive to model misspecification as was shown by van Staden, Dang, and Forsyth [29]. Continuous dynamic asset allocation problems have known analytical solutions with certain constraints. For example, Bielecki et al. [3] derive a solution when bankruptcy is not allowed, and when shorting selling is not allowed, Li, Zhou, and Lim [23] give an explicit solution. When discrete rebalancing is performed and realistic constraints are imposed on the portfolios, an analytical solution is not available in general and portfolio strategies need to be found numerically. This paper considers this case.

Brandt et al. [4] consider Monte Carlo simulation-based discrete-time portfolio allocation problems. While their approach is fairly flexible, it assumes the asset allocation to be independent of current wealth. This is restrictive and leads to suboptimal investment strategies. Instead, it is preferable to consider time- and wealth-dependent asset allocations to maximize the final wealth under a given level of risk aversion. These are called precommitment strategies by Basak and Chabakauri [2] and are typically not time-consistent; see [28], for ex-

*Received by the editors December 14, 2023; accepted for publication (in revised form) April 17, 2024; published electronically June 3, 2024.
https://doi.org/10.1137/23M1624439
Funding: This work was funded by the Academy of Finland, project 295897.
†Faculty of Information Technology, University of Jyväskylä, FI-40014 Jyväskylä, Finland (raino.a.e.makinen@jyu.fi, jari.a.toivanen@jyu.fi).

ample. In the case of the mean-variance optimization, there is an induced objective function for which the solution is time-consistent [12], [8], [31]. Cong and Oosterlee [6], [7] construct these precommitment strategies based on Monte Carlo simulated risky asset paths. Their approach drives a suboptimal multistage strategy to an optimal one using backward recursive programming. They perform the common transformation of the mean-variance problem with nonlinear conditional variance to a linear-quadratic (LQ) problem by an embedding technique by Li and Ng [22]. Another approach to constructing precommitment strategies is to formulate a Hamilton–Jacobi–Bellman (HJB) partial differential equation (PDE) for the strategy. This is an elegant approach that avoids sampling risky asset paths, but its numerical implementation is fairly cumbersome and, unlike Monte Carlo–based methods, it does not scale well for multiple risky assets. This HJB PDE approach has been considered by Wang and Forsyth [32], Dang and Forsyth [9], and Forsyth and Vetzal [13], for example.

Recently, several studies, including [5], [16], [26], [25], [30], have proposed neural networks (NNs) for financial optimization problems without relying on dynamic programming. These studies describe the control by an NN and the loss function is given by the financial objective. The resulting NN training problem is solved using the usual stochastic gradient methods in this context. Here we propose a similar, non–dynamic programming–based approach describing the control by a more traditional polynomial interpolation similar to many dynamic programming–based financial optimization studies including [6], [7], [9], [13], [32]. We solve the resulting optimization problems by a quasi-Newton optimization method. The proposed approach has two benefits: polynomial interpolations have well-established approximation convergence properties, and the quasi-Newton methods have fast convergence leading to shorter computation times.

This paper describes an optimization approach based on Monte Carlo simulated risky asset paths. The optimization is performed directly to the objective function given by the desired combination of the expected final wealth and the risk measure without any transformation. This leads to a nonlinear optimization problem for time- and wealth-dependent asset allocations for which it is easy to impose constraints. This proposed approach is flexible and can be easily generalized for many cases.

Quasi-Newton methods offer an efficient way to solve the resulting optimization problems. Particularly methods based on Broyden–Fletcher–Goldfarb–Shanno (BFGS) approximation [11] of the Hessian matrix have been shown to be efficient and are very popular. These methods require the gradient of the objective function with respect to the optimization variables, that is, the time- and wealth-dependent asset allocations. The adjoint technique gives an efficient way to compute this gradient. With Monte Carlo simulations Giles [14], [15] describes this technique. Kaebe, Maruhn, and Sachs [18] employ it to calibrate a market model. To the best of our knowledge, in the scientific literature, these techniques have not been used before to construct asset allocation strategies. Instead of applying automatic differentiation to compute the gradient, we derive an analytical expression for the gradient, which can be used for the efficient implementation of the method. The efficient frontier of possible portfolios is obtained by optimizing the portfolios with varying levels of investor risk aversion. We present numerical examples for the case of one and four risky assets.

The outline of this paper is the following: Section 2 describes the mean wealth-risk measure optimization problem. Section 3 gives the details of Monte Carlo simulation of the wealth as well as computation of the variance and semivariance of the final wealth which are the risk measures studied in this paper. Section 4 proposes a numerical solution method for the resulting optimization problem. Section 5 presents numerical examples of portfolio optimization. Section 6 gives the conclusions.

2. Mean wealth-risk measure portfolio optimization. Let there be $I+1$ investment assets. Let the ith asset S^i follow the stochastic differential equation (SDE)

$$dS^i = \mu^i S^i dt + \sigma^i S^i dZ^i, \tag{2.1}$$

where μ^i is the growth rate, σ^i is the volatility, and Z^i is the Wiener process. The correlation between these processes is specified by the correlation matrix. When the volatility is zero the asset is riskless.

The accumulated total wealth $W : [0,T] \to \mathbb{R}$ follows the SDE

$$\begin{cases} dW = W \left[\sum_{i=1}^{I+1} p^i \left(\mu^i dt + \sigma^i dZ^i \right) \right] + \pi dt, \\ W(0) = w_0, \end{cases} \tag{2.2}$$

where π is a contribution rate, $p^i = p^i(t,W)$ is the proportion of the wealth invested in the ith asset S^i_t at time t, and w_0 is the initial wealth. The last asset S^{I+1} is assumed to be riskless, that is, $\sigma^{I+1} = 0$ and $r := \mu^{I+1}$ is the riskless interest rate. Thus, the number of risky assets is I. The proportion of the wealth invested in the riskless asset is $p^{I+1} = 1 - \sum_{i=1}^{I} p^i$. Eliminating the proportion of the riskless asset from (2.2) we obtain the equivalent SDE that is better suited for computations:

$$\begin{cases} dW = W \left[r\,dt + \sum_{i=1}^{I} p^i \left((\mu^i - r)\,dt + \sigma^i dZ^i \right) \right] + \pi\,dt, \\ W(0) = w_0. \end{cases} \tag{2.3}$$

Let $P = (p^1 \cdots p^I)^T$ contain the proportions p^i, $i = 1, \dots, I$. Furthermore, let $\mathrm{E}[W_P(T)]$ and $\mathrm{RM}[W_P(T)]$ denote the expected value and risk measure for the final wealth $W(T)$ when following an investment strategy P. Typical risk measures are the variance $\mathrm{Var}[\cdot]$ and the semivariance $\mathrm{Semivar}[\cdot]$. The semivariance is a special case of downside risk models [10]. Under a discrete-time investment strategy, it leads to a well-posed problem [17]. The aim is to find a strategy $P^*_\lambda \in \mathcal{P}_{ad}$ such that

$$P^*_\lambda = \arg \max_{P \in \mathcal{P}_{ad}} \left(\mathrm{E}[W_P(T)] - \lambda \mathrm{RM}[W_P(T)] \right), \tag{2.4}$$

where $\lambda > 0$ describes the investor's risk aversion which grows with λ. Varying λ gives the Pareto optimal portfolios. The set $\mathcal{P}_{ad}$ defines allowed strategies. Forbidding short selling leads the lower bound $p_{\min}$ for the proportions p^i, $i = 1, \dots, I$, to be zero. Let $p_{\max}$ be the allowed amount of leverage. For example, allowing a 2:1 leverage ratio corresponds to $p_{\max} = 2$, while no leverage allowed corresponds to $p_{\max} = 1$. The proportions have to satisfy $p^i \le p_{\max}$. Furthermore, the sum of the proportions has to be at most $p_{\max}$, that is, $\sum_{i=1}^{I} p^i \le p_{\max}$.

3. Monte Carlo simulation of wealth.

For the moment, let the investment policy P be given and fixed. We approximate the solution of the SDE (2.3) by using the classical Euler–Maruyama scheme. Let $\Delta t = T/N$ be the time step, and let $\boldsymbol{Z} \in \mathbb{R}^{K\times I\times N}$ be an array of normally distributed pseudorandom numbers. In this paper, K is the number of Brownian paths.

Let W_k^n denote the kth random approximation of $W(n\Delta t)$. These values at the nth time step are collected to the vector $\boldsymbol{W}^n = (W_1^n \cdots W_K^n) \in \mathbb{R}^K$. One step of the numerical scheme reads

$$\boldsymbol{W}^{n+1} = \left[1 + \Delta t r + \sum_{i=1}^{I} \boldsymbol{P}_i^n(\boldsymbol{W}^n) \odot \left(\Delta t \left(\mu^i - r\right) + \sqrt{\Delta t}\sigma^i \boldsymbol{Z}_i^n\right)\right] \odot \boldsymbol{W}^n + \Delta t \pi, \tag{3.1}$$

where the vectors $\boldsymbol{P}_i^n(\boldsymbol{W}^n)$ contain the proportions evaluated at (t_n, W_k^n) for $k = 1, \ldots, K$, i.e., $\boldsymbol{P}_i^n(\boldsymbol{W}^n) = \left(p^i(t_n, W_1^n) \cdots p^i(t_n, W_K^n)\right)$.

Moreover, the vector $\boldsymbol{Z}_i^n \in \mathbb{R}^K$ contains the K random numbers for the ith asset at the time step n, and $\odot$ is the elementwise vector product operator.[1] This can be expressed in a more compact form

$$\boldsymbol{W}^{n+1} = \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \boldsymbol{W}^n + \Delta t \pi, \tag{3.2}$$

where

$$\boldsymbol{S}^n(\boldsymbol{P}^n) = 1 + \Delta t r + \sum_{i=1}^{I} \boldsymbol{P}_i^n(\boldsymbol{W}^n) \odot \left(\Delta t \left(\mu^i - r\right) + \sqrt{\Delta t}\sigma^i \boldsymbol{Z}_i^n\right). \tag{3.3}$$

The expected final wealth is given by

$$\mathrm{E}\left(\boldsymbol{W}^N\right) = \tfrac{1}{K} \sum_{k=1}^{K} W_k^N. \tag{3.4}$$

Its variance and semivariance are given by

$$\begin{aligned} \mathrm{Var}\left(\boldsymbol{W}^N\right) &= \tfrac{1}{K} \sum_{k=1}^{K} \left(W_k^N - \mathrm{E}\left(\boldsymbol{W}^N\right)\right)^2 \qquad \text{and} \\ \mathrm{Semivar}\left(\boldsymbol{W}^N\right) &= \tfrac{1}{K} \sum_{k=1}^{K} \left(\min\left\{W_k^N - \mathrm{E}\left(\boldsymbol{W}^N\right), 0\right\}\right)^2, \end{aligned} \tag{3.5}$$

respectively. Note that unlike here sometimes the semivariance is defined with the inverse of the number of samples below the expected value instead of the inverse of the number of all samples.

[1] Here we adopt the MATLAB style notation: If $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n, \beta \in \mathbb{R}$, then $\boldsymbol{z} = \boldsymbol{x} \odot \boldsymbol{y} + \beta$ means $z_i = x_i y_i + \beta$, $i{=}1, \ldots, n$.

4. Fully discrete optimization problem. Until now the policy P has been a continuous vector-valued function of time and wealth. Next, we introduce a parametrized strategy $P_h = (p_h^1 \cdots p_h^I)$, where each proportion depends only on a finite number of parameters.

Consider the $M \times N$ grid $G := \{0 = \mathcal{W}_1 < \mathcal{W}_2 < \cdots < \mathcal{W}_M = W_{\max}\} \times \{0 = t_0 < t_1 < \cdots < t_{N-1}\}$, where $t_n = n\Delta t$ and $W_{\max}$ is large enough such that $W_k(t) \in [0, W_{\max}]$ for all paths. Let $\{\psi_{m,n}(t,W)\}$ be the set of piecewise bilinear C^0-continuous basis functions associated with G, where each $\psi_{m,n}$ has the value one at (t_m, W_n) and zero elsewhere. Moreover, let us use the following notation for a set of $I{\cdot}M{\cdot}N$ parameters:

$$\mathbb{P} = (\boldsymbol{p}^1 \dots \boldsymbol{p}^I) \in \mathbb{R}^{I\cdot M\cdot N}, \quad \boldsymbol{p}^i = (p_{m,n}) \in \mathbb{R}^{M\cdot N}, \quad i = 1, \dots, I.$$

Now, we can define the following discretized proportions of the strategy P_h:

$$p_h^i(\boldsymbol{p}^i; t, W) = \sum_{m=1}^{M} \sum_{n=0}^{N-1} p_{m,n}^i \psi_{m,n}(t, W).$$

The parametrized and discretized optimization problem then reads

$$\mathbb{P}_\lambda^* = \arg\max_{\mathbb{P}\in\mathcal{U}} \mathcal{J}_\lambda(\mathbb{P}), \tag{4.1}$$

where the discrete objective function is defined by

$$\mathcal{J}_\lambda(\mathbb{P}) := J_\lambda\left(\boldsymbol{W}^N(\mathbb{P})\right) = \mathrm{E}\left(\boldsymbol{W}^N\right) - \lambda\,\mathrm{RM}\left(\boldsymbol{W}^N\right). \tag{4.2}$$

The set of admissible parameters in (4.1) is defined by

$$\mathcal{U} = \{\mathbb{P} \in \mathbb{R}^{I\cdot M\cdot N} \mid \mathbb{P}_{\min} \le \mathbb{P} \le \mathbb{P}_{\max} \quad \text{and} \quad \boldsymbol{A}\mathbb{P} \le \boldsymbol{b}\}, \tag{4.3}$$

where the lower bound vector $\mathbb{P}_{\min}$ and the upper bound vector $\mathbb{P}_{\max}$ result from the lower and upper bounds for the proportions p^i, $i = 1, \dots, I$, and the linear constraint $\boldsymbol{A}\mathbb{P} \le \boldsymbol{b}$ results from the limit for the leverage. The definitions of these vectors and the matrix $\boldsymbol{A}$ are given by $\mathbb{P}_{\min} = p_{\min}\boldsymbol{e}_{I\cdot M\cdot N}$, $\mathbb{P}_{\max} = p_{\max}\boldsymbol{e}_{I\cdot M\cdot N}$, $\boldsymbol{A} = \boldsymbol{I}_{M\cdot N} \otimes \boldsymbol{e}_I^T$, and $\boldsymbol{b} = p_{\max}\boldsymbol{e}_{M\cdot N}$, where $\boldsymbol{e}_n = (1 \cdots 1)^T \in \mathbb{R}^n$, I_n is the $n \times n$ identity matrix, and $\otimes$ is the Kronecker product operator.

To efficiently utilize gradient-type methods for the numerical solution of (4.1), it is essential to have the exact gradient $\nabla_{\mathbb{P}}\mathcal{J}_\lambda(\boldsymbol{W}^N(\mathbb{P}))$ rather than relying on its finite difference approximation. Exact gradient computations can be performed manually or with the assistance of automatic differentiation tools readily accessible in popular software libraries for machine learning and artificial intelligence, such as TensorFlow [1]. In what follows, we derive a concise expression for the gradient using the classical adjoint approach. Following that, we provide a brief overview of the advantages and challenges associated with the application of automatic differentiation tools.

The partial derivatives of the objective function with respect to the parameters defining the discrete investment strategy can be computed using the adjoint formulation [14] holding fixed the randomly generated Brownian path increments for every particular path calculation. In what follows, we assume a general parametrization of $p_h^i = p_h^i(\boldsymbol{p}^i; t, W)$, $\boldsymbol{p}^i \in \mathbb{R}^{M\cdot N}$, that is, the calculations are not restricted to any particular parametrization. We assume that the mapping $\mathbb{P} \mapsto \boldsymbol{W}^N$ is smooth and derive formally the explicit formula for $\frac{\partial \mathcal{J}_\lambda}{\partial p_j}$, where p_j is a component of $\mathbb{P}$.

Using the notation of section 3 we can express the Monte Carlo simulation of the wealth as the state problem

$$\begin{cases} \boldsymbol{W}^0 = w_0, \\ \boldsymbol{W}^{n+1} = \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \boldsymbol{W}^n + \Delta t\, \pi, \quad n = 0, \dots, N-1. \end{cases} \tag{4.4}$$

Define the Lagrangian with a set of Lagrange multipliers $\mathbb{Y} := \{\boldsymbol{Y}^0, \dots, \boldsymbol{Y}^N\}$:

$$\mathcal{L}_\lambda = \mathcal{J}_\lambda(\boldsymbol{W}^N) - \sum_{n=0}^{N-1} (\boldsymbol{Y}^{n+1})^{\mathrm{T}} (\boldsymbol{W}^{n+1} - \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \boldsymbol{W}^n - \Delta t\, \pi) - (\boldsymbol{Y}^0)^{\mathrm{T}} (\boldsymbol{W}^0 - w_0). \tag{4.5}$$

As $\mathbb{W} := \{\boldsymbol{W}^0, \dots, \boldsymbol{W}^N\}$ satisfies (4.4) for all $\mathbb{P}$, we may choose $\mathbb{Y}$ freely. We have $\mathcal{J}_\lambda(\boldsymbol{W}^N) = \mathcal{L}_\lambda(\mathbb{P}, \mathbb{W}, \mathbb{Y})$ and

$$\begin{aligned} \frac{\partial \mathcal{J}_\lambda}{\partial p_j} = \frac{\partial \mathcal{L}_\lambda}{\partial p_j} = (\nabla_{\boldsymbol{W}^N} \mathcal{J}_\lambda)^{\mathrm{T}} \frac{\partial \boldsymbol{W}^N}{\partial p_j} - \sum_{n=0}^{N-1} (\boldsymbol{Y}^{n+1})^{\mathrm{T}} \Big(\frac{\partial \boldsymbol{W}^{n+1}}{\partial p_j} - \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial p_j} \odot \boldsymbol{W}^n \\ - \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial W} \odot \frac{\partial \boldsymbol{W}^n}{\partial p_j} \odot \boldsymbol{W}^n - \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \frac{\partial \boldsymbol{W}^n}{\partial p_j} \Big) - (\boldsymbol{Y}^0)^{\mathrm{T}} \frac{\partial \boldsymbol{W}^0}{\partial p_j}. \end{aligned} \tag{4.6}$$

Above we have denoted

$$\begin{aligned} \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial W} &:= \sum_{i=1}^{I} (\Delta t(\mu^i - \mu^{i+1}) {p_h^i}' + \sqrt{\Delta t} \sigma^i {p_h^i}' \odot \boldsymbol{Z}_i^n), \\ \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial p_j} &:= \sum_{i=1}^{I} (\Delta t(\mu^i - \mu^{i+1}) \dot{p}_h^i + \sqrt{\Delta t} \sigma^i \dot{p}_h^i \odot \boldsymbol{Z}_i^n) \end{aligned}$$

with

$${p_h^i}' := \frac{\partial p_h(\boldsymbol{p}^i, t_n, \boldsymbol{W}^n)}{\partial W} \quad \text{and} \quad \dot{p}_h^i := \frac{\partial p_h(\boldsymbol{p}^i, t_n, \boldsymbol{W}^n)}{\partial p_j}. \tag{4.7}$$

Rearranging terms in (4.6) gives

$$\begin{aligned} \frac{\partial \mathcal{L}_\lambda}{\partial p_j} = \sum_{n=0}^{N-1} (\boldsymbol{Y}^{n+1})^{\mathrm{T}} \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial p_j} \odot \boldsymbol{W}^n + \left(\frac{\partial \boldsymbol{W}^N}{\partial p_j} \right)^{\mathrm{T}} \left(\boldsymbol{Y}^N - \nabla_{\boldsymbol{W}^N} \mathcal{J}_\lambda \right) \\ - \sum_{n=0}^{N-1} \left(\frac{\partial \boldsymbol{W}^n}{\partial p_j} \right)^{\mathrm{T}} \left(\boldsymbol{Y}^n - \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial W} \odot \boldsymbol{W}^n \odot \boldsymbol{Y}^{n+1} - \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \boldsymbol{Y}^{n+1} \right). \end{aligned} \tag{4.8}$$

If we choose $\mathbb{Y}$ to be the solution of the adjoint model

$$\begin{cases} \boldsymbol{Y}^N = \nabla_{\boldsymbol{W}^N} \mathcal{J}_\lambda(\boldsymbol{W}^N), \\ \boldsymbol{Y}^n = \dfrac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial W} \odot \boldsymbol{W}^n \odot \boldsymbol{Y}^{n+1} + \boldsymbol{S}^n(\boldsymbol{P}^n) \odot \boldsymbol{Y}^{n+1}, \quad n = N-1, N-2, \dots, 0, \end{cases} \tag{4.9}$$

then only the first term in (4.8) is nonzero and we avoid calculating $\frac{\partial \boldsymbol{W}^n}{\partial p_j}$. Thus, we finally have

$$\frac{\partial \mathcal{J}_\lambda}{\partial p_j} = \sum_{n=0}^{N-1} (\boldsymbol{Y}^{n+1})^{\mathrm{T}} \frac{\partial \boldsymbol{S}^n(\boldsymbol{P}^n)}{\partial p_j} \odot \boldsymbol{W}^n. \tag{4.10}$$

Remark 4.1. In the derivation of the formulas (4.9), (4.10), the part depending on the specific parametrization is contained in the derivatives appearing in (4.7). The continuous piecewise linear parametrization of the investment policy, p_h, is not continuously differentiable with respect to W. Thus, it may happen that (4.10) only gives a directional derivative. However, it is well known that standard quasi-Newton methods are relatively robust and efficient even in the nonsmooth case. For more discussion on that topic, see [20], [21], for example.

Remark 4.2. When it comes to using graph-based automatic differentiation tools like TensorFlow for solving large-scale stochastic optimal control problems, researchers noted, as seen in [19], that these tools, while optimized for training neural networks, face significant challenges due to high memory and initialization requirements. For this reason, it is preferable to use a manually computed gradient like the one in (4.10) when the number of parameters is large.

A "common subexpression elimination technique" introduced in [19] selectively uses the automatic differentiation leading to a comparable efficiency with a manually computed gradient. Applying this technique to our case would involve using the automatic differentiation to calculate derivatives $\nabla_{\boldsymbol{W}} \mathcal{J}_\lambda$, $\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{W}}$, and $\frac{\partial \boldsymbol{S}}{\partial \mathbb{P}}$ in (4.9)–(4.10) instead of applying it directly to the black box $\mathbb{P} \mapsto J_\lambda(\mathbb{P})$.

5. Numerical portfolio optimization examples.

In this section, we present two examples dealing with one risky asset and one example dealing with four risky and correlating assets. The computations have been performed using MATLAB [27] with the gradient computations implemented using (4.10). The final time and time step for Monte Carlo simulations are $T = 20$ and $\Delta t = 0.25$, respectively, leading to $N = 80$ time steps. In this section, we have used $K = 1000000$ paths. For the strategy p there are $M = 31$ grid points in the W direction with the last grid point at $W_{\max} = 30$, and the grid is refined for small W values. In practical computations, we employ the constant approximation $p(t_n, x) = p(t_n, W_{\max})$, $x > W_{\max}$, to ensure the use of a reasonably small constant $W_{\max}$.

In optimization, the quasi-Newton method with the BFGS approximation of the Hessian matrix was used. To guarantee a robust convergence of the optimizer for all the sampled λ values, we used 50 iterations for optimizations with one risky asset and 250 iterations for optimizations with four risky assets. The likely reason for a larger number of required iterations in the case of the higher-dimensional problem is not the larger number of parameters, but the multiple correlated investment assets.

5.1. Portfolio with one risky asset and no leverage.

We start by considering a pension plan example with one risky asset, that is, $I = 1$ with the following parameters: the interest rate $r = 0.03$, the volatility of the risky asset $\sigma^1 = 0.15$, the growth rate of the risky asset $\mu^1 = 0.0795$, the contribution rate $\pi = 0.1$, and the initial wealth $w_0 = 1$. Short selling is forbidden leading to $p_{\min} = 0$. Borrowing is not allowed leading the maximum proportion of the wealth invested in the risky asset to be $p_{\max} = 1$.

We compute the efficient frontiers using the variance and the semivariance as the risk measure. Furthermore, we compute also the efficient frontier given by the constant proportions p when increasing this constant from zero to one. These efficient frontiers are formed by performing the optimization for 11 values for the risk aversion parameter λ.

The mean-variance and mean-semivariance frontier plots for all three investment strategies are shown in Figure 1. The final wealth probability distributions for the three strategies when $\mathrm{E}[W(T)] = 8$ are depicted in Figure 2. The corresponding mean-variance and mean-semivariance optimized controls p are depicted in Figure 3.

We studied the convergence with respect to M (the number of discretization points in the W-direction) and with respect to K (the number of paths). The other parameters were the

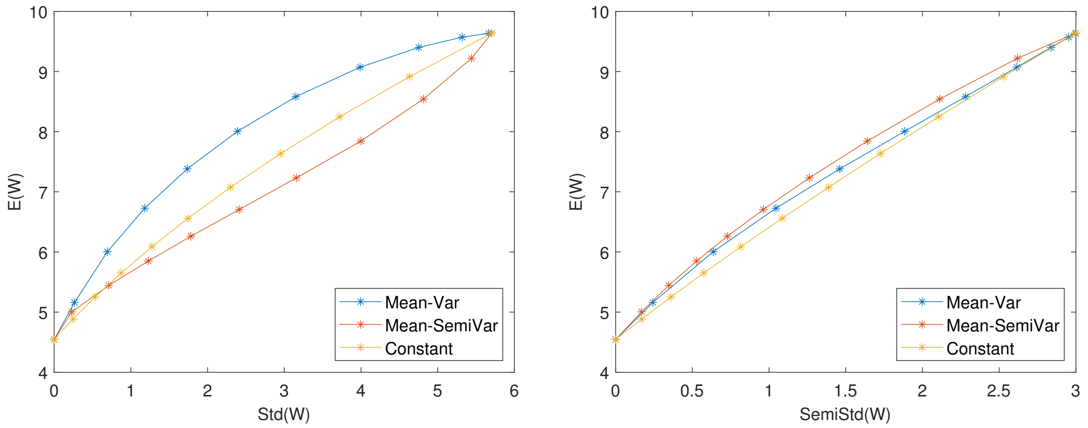

Figure 1. *The mean-variance and mean-semivariance frontiers for the mean-variance optimized portfolios, the mean-semivariance optimized portfolios, and the constant proportion portfolios.*

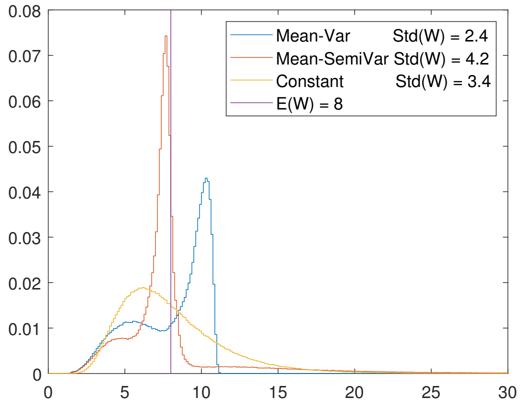

Figure 2. *The probability distributions for the final wealth when* $\mathrm{E}[W(T)] = 8$ *for the three investment strategies.*

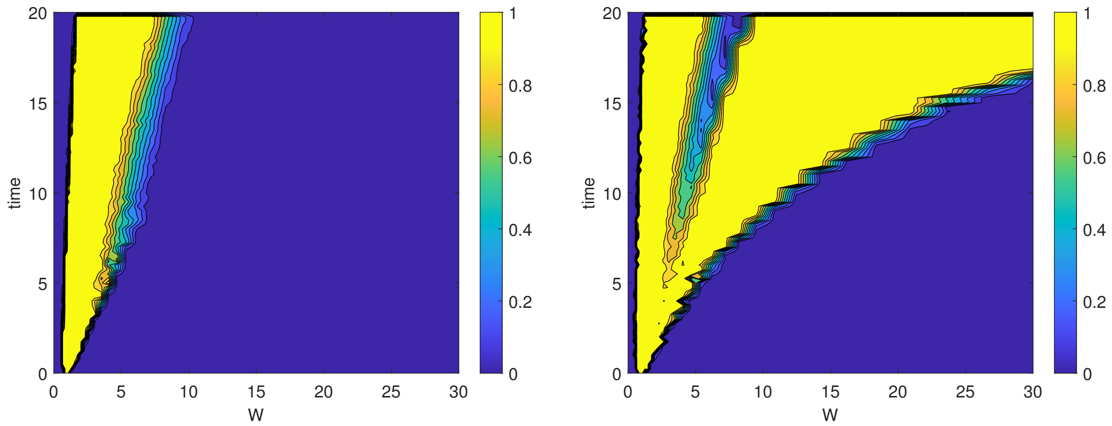

Figure 3. *Mean-variance and mean-semivariance optimized proportions p for $E[W] = 8$.*

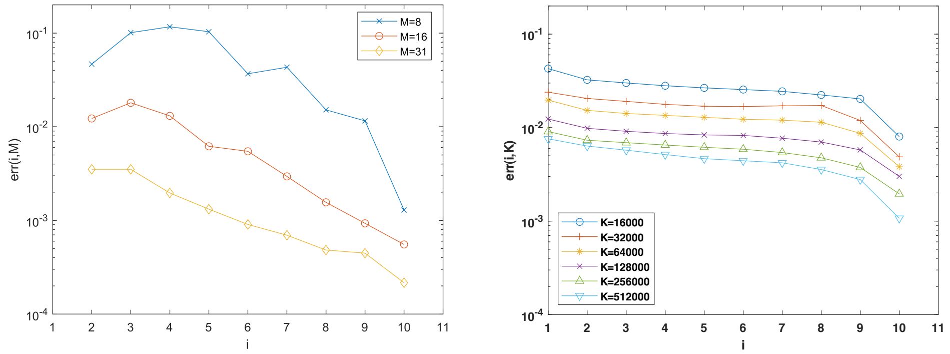

Figure 4. *Left: Difference $err(i, M) := |\mathcal{J}_{\lambda_i,M,10^6} - J_i^\dagger|$. Right: Difference $err(i, K) := |\mathcal{J}_{\lambda_i,31,K} - J_i^\sharp|$. Due to logarithmic y-axes, values with $err = 0$ are not plotted.*

same as above. Let $\mathcal{J}_{i,M,K} := \mathcal{J}_{\lambda_i}(\mathbb{P}^*)$, where $\mathbb{P}^*$ is the optimal control computed with M grid points in the W-direction and using K paths. Let $J_i^\dagger = \mathcal{J}_{i,61,10^6}$ and $J_i^\sharp = \mathcal{J}_{i,31,4\times 10^6}$. We consider the values $J_i^\dagger, J_i^\sharp$ good approximations to the exact optimal objective function values and we compare them with the values obtained using smaller M or K. In the latter case, we take the average of 100 cost evaluations using a different sets of paths. The results of the tests are depicted in Figure 4. From these tests, we can conclude that the empirical error in $\mathcal{J}_\lambda$ is roughly $\sim 1/M^2$ and $\sim 1/\sqrt{K}$. These results are consistent with the theoretical convergence properties of piecewise linear interpolation and the Monte Carlo method.

5.2. Portfolio with one risky asset and leverage. We keep the parameters the same as in section 5.1 except now the maximum leverage is given by $p_{\max} = 1.5$. This is the example considered by Wang and Forsyth [32] and Cong and Oosterlee [6]. The mean-variance and mean-semivariance efficient frontiers are formed by performing the optimization for 11 values for the risk aversion parameter λ. The mean-variance and mean-semivariance frontiers are shown in Figure 5. The mean-variance frontier agrees well with the efficient frontiers presented in [32, 6].

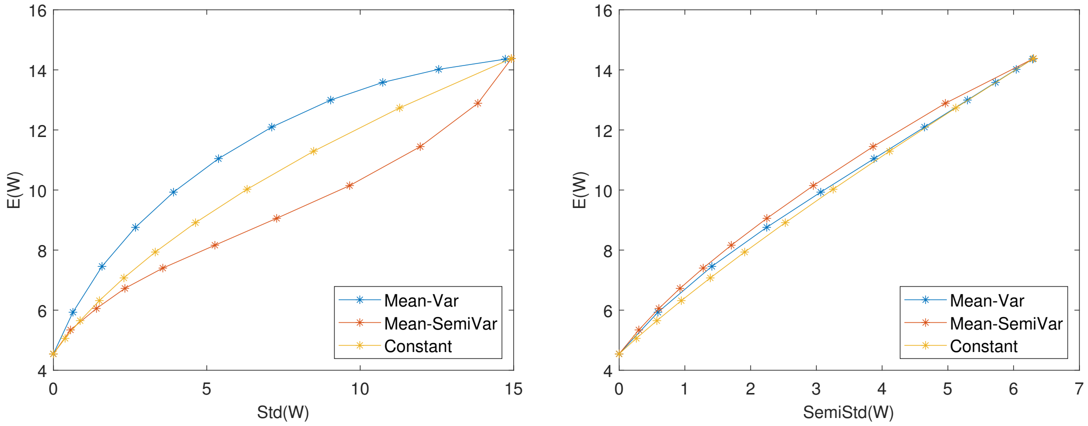

Figure 5. *The mean-variance and mean-semivariance frontiers for the mean-variance and mean-semivariance optimized portfolios and the constant proportion portfolios when the maximum leverage is* $p_{\max} = 1.5$.

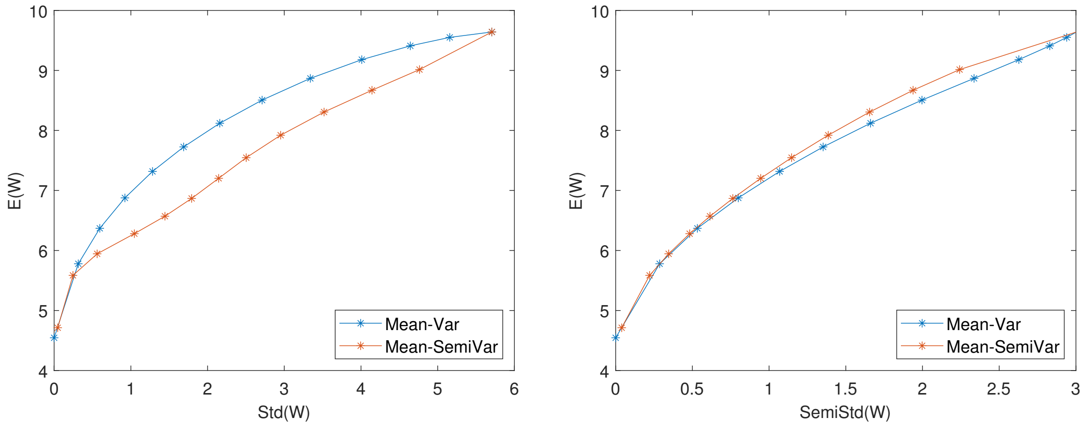

Figure 6. *The mean-variance and semi-semivariance frontiers for the mean-variance and mean-semivariance optimized portfolios for four risky assets.*

5.3. Portfolio with four risky assets. This is a generalization of the previous examples which adds three more risky assets. The volatilities of the four risky assets are given by the vector $\boldsymbol{\sigma} = (0.15\ 0.12\ 0.09\ 0.06)^T$, and their growth rates are given by the vector $\boldsymbol{\mu} = (0.0795\, 0.07\, 0.06\, 0.05)^T$. The correlation matrix between the risky assets is

$$\boldsymbol{C} = \begin{pmatrix} 1 & 0.6 & 0.2 & 0.1 \\ 0.6 & 1 & 0.4 & 0.2 \\ 0.2 & 0.4 & 1 & 0.4 \\ 0.1 & 0.2 & 0.4 & 1 \end{pmatrix}.$$

As before the interest rate is $r = 0.03$, the final time is $T = 20$, and the initial wealth is $w_0 = 1$. The short selling is not allowed and there is no leverage leading to $p_{\min} = 0$ and $p_{\max} = 1$. The mean-variance and mean-semivariance efficient frontiers are formed by performing the optimization for 13 values for the risk aversion parameter λ. The mean-variance and mean-semivariance frontier plots for the two optimized investment strategies are shown in Figure 6.

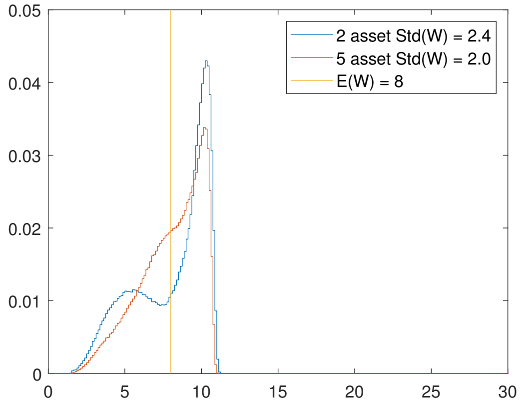

Figure 7. *The probability distributions for the final wealth when* $\mathrm{E}[W(T)] = 8$ *for one and four risky assets.*

The final wealth probability distributions for the mean-variance optimized portfolios with $\mathrm{E}[W(T)] = 8$ for two (one risky) and five (four risky) asset cases are depicted in Figure 7.

6. Conclusions.

We presented a very generic Monte Carlo-based approach for portfolio optimization. The models for the asset and the risk can be easily changed. The approach does not require dynamic programming or any transformations. Restrictions on the investment policies can be easily incorporated. In this paper, we used the variance and the semivariance as the risk measure. The numerical examples considered cases with two and five assets.

Acknowledgment. We thank the anonymous referees whose constructive comments improved the paper.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, et al., *TensorFlow: A system for large-scale machine learning*, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), 2016, pp. 265–283.

[2] S. Basak and G. Chabakauri, *Dynamic mean-variance asset allocation*, Rev. Financ. Stud., 23 (2010), pp. 2970–3016, https://doi.org/10.1093/rfs/hhq028.

[3] T. R. Bielecki, H. Jin, S. R. Pliska, and X. Y. Zhou, *Continuous-time mean-variance portfolio selection with bankruptcy prohibition*, Math. Finance, 15 (2005), pp. 213–244, https://doi.org/10.1111/j.0960-1627.2005.00218.x.

[4] M. W. Brandt, A. Goyal, P. Santa-Clara, and J. R. Stroud, *A simulation approach to dynamic portfolio choice with an application to learning about return predictability*, Rev. Financ. Stud., 18 (2005), pp. 831–873.

[5] H. Buehler, L. Gonon, J. Teichmann, and B. Wood, *Deep hedging*, Quant. Finance, 19 (2019), pp. 1271–1291.

[6] F. Cong and C. W. Oosterlee, *Multi-period mean-variance portfolio optimization based on Monte-Carlo simulation*, J. Econom. Dynam. Control, 64 (2016), pp. 23–38, https://doi.org/10.1016/j.jedc.2016.01.001.

[7] F. Cong and C. W. Oosterlee, *On pre-commitment aspects of a time-consistent strategy for a mean-variance investor*, J. Econom. Dynam. Control, 70 (2016), pp. 178–193, https://doi.org/10.1016/j.jedc.2016.07.010.

[8] X. Cui, D. Li, X. Qiao, and M. S. Strub, *Risk and Potential: An Asset Allocation Framework with Applications to Robo-Advising*, SSRN 3302111, 2019.

[9] D.-M. Dang and P. A. Forsyth, *Continuous time mean-variance optimal portfolio allocation under jump diffusion: An numerical impulse control approach*, Numer. Methods Partial Differential Equations, 30 (2014), pp. 664–698, https://doi.org/10.1002/num.21836.

[10] P. C. Fishburn, *Mean-risk analysis with risk associated with below-target returns*, Am. Econ. Rev., 67 (1977), pp. 116–126.

[11] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., A Wiley-Interscience Publication, John Wiley & Sons, Chichester, 1987.

[12] P. A. Forsyth, *Multiperiod mean conditional value at risk asset allocation: Is it advantageous to be time consistent?*, SIAM J. Financial Math., 11 (2020), pp. 358–384, https://doi.org/10.1137/19M124650X.

[13] P. A. Forsyth and K. R. Vetzal, *Dynamic mean variance asset allocation: Tests for robustness*, Int. J. Financ. Eng., 4 (2017), 1750021, https://doi.org/10.1142/S2424786317500219.

[14] M. B. Giles, *Monte Carlo Evaluation of Sensitivities in Computational Finance*, Technical Report 12, Oxford-Man Institute, University of Oxford, Oxford, UK, 2007.

[15] M. B. Giles, *Vibrato Monte Carlo sensitivities*, in Monte Carlo and Quasi-Monte Carlo Methods 2008, Springer, Berlin, 2009, pp. 369–382, https://doi.org/10.1007/978-3-642-04107-5_23.

[16] J. Han and W. E, *Deep learning approximation for stochastic control problems*, in NIPS Deep Reinforcement Learning Workshop, 2016.

[17] H. Jin, J.-A. Yan, and X. Y. Zhou, *Continuous-time mean–risk portfolio selection*, Ann. Inst. H. Poincaré Probab. Statist., 41 (2005), pp. 559–580, https://doi.org/10.1016/j.anihpb.2004.09.009.

[18] C. Kaebe, J. H. Maruhn, and E. W. Sachs, *Adjoint-based Monte Carlo calibration of financial market models*, Finance Stoch., 13 (2009), pp. 351–379, https://doi.org/10.1007/s00780-009-0097-9.

[19] P. Lambrianides, Q. Gong, and D. Venturi, *A new scalable algorithm for computational optimal control under uncertainty*, J. Comput. Phys., 420 (2020), 109710, https://doi.org/10.1016/j.jcp.2020.109710.

[20] C. Lemaréchal, *Numerical experiments in nonsmooth optimization*, in Progress in Nondifferentiable Optimization, E. A. Nurminski, ed., International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria, 1982, pp. 61–84.

[21] A. S. Lewis and M. L. Overton, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163, https://doi.org/10.1007/s10107-012-0514-2.

[22] D. Li and W.-L. Ng, *Optimal dynamic portfolio selection: Multiperiod mean-variance formulation*, Math. Finance, 10 (2000), pp. 387–406, https://doi.org/10.1111/1467-9965.00100.

[23] X. Li, X. Y. Zhou, and A. E. B. Lim, *Dynamic mean-variance portfolio selection with no-shorting constraints*, SIAM J. Control Optim., 40 (2002), pp. 1540–1555, https://doi.org/10.1137/S0363012900378504.

[24] H. M. Markowitz, *Portfolio selection*, J. Finance, 7 (1952), pp. 77–91.

[25] A. M. Reppen and H. Mete Soner, *Deep empirical risk minimization in finance: Looking into the future*, Math. Finance, 33 (2023), pp. 116–145.

[26] A. M. Reppen, H. Mete Soner, and V. Tissot-Daguette, *Deep stochastic optimization in finance*, Digit Finance, 5 (2023), pp. 91–111, https://doi.org/10.1007/s42521-022-00074-6.

[27] The MathWorks Inc., *MATLAB R2022b,* Natick, MA, 2022, https://www.mathworks.com.

[28] P. M. van Staden, D.-M. Dang, and P. A. Forsyth, *Mean-quadratic variation portfolio optimization: A desirable alternative to time-consistent mean-variance optimization?*, SIAM J. Financial Math., 10 (2019), pp. 815–856, https://doi.org/10.1137/18M1222570.

[29] P. M. van Staden, D.-M. Dang, and P. A. Forsyth, *The surprising robustness of dynamic Mean-Variance portfolio optimization to model misspecification errors*, European J. Oper. Res., 289 (2021), pp. 774–792, https://doi.org/10.1016/j.ejor.2020.07.021.

[30] P. M. Van Staden, P. A. Forsyth, and Y. Li, *Beating a benchmark: Dynamic programming may not be the right numerical approach*, SIAM J. Financial Math., 14 (2023), pp. 407–451, https://doi.org/10.1137/22M1530070.

[31] E. Vigna, *On time consistency for mean-variance portfolio selection*, Int. J. Theor. Appl. Finance, 23 (2020), 2050042.

[32] J. Wang and P. A. Forsyth, *Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation*, J. Econom. Dynam. Control, 34 (2010), pp. 207–230, https://doi.org/10.1016/j.jedc.2009.09.002.