data

MDPI

Article

Nested Stochastic Valuation of Large Variable Annuity Portfolios: Monte Carlo Simulation and Synthetic Datasets

Guojun Gan * and Emiliano A. Valdez

Department of Mathematics, University of Connecticut, 341 Mansfield Road, Storrs, CT 06269-1009, USA; emiliano.valdez@uconn.edu

* Correspondence: guojun.gan@uconn.edu; Tel.: +1-860-486-3919

Received: 11 July 2018; Accepted: 30 August 2018; Published: 1 September 2018

Abstract: Dynamic hedging has been adopted by many insurance companies to mitigate the financial risks associated with variable annuity guarantees. To simulate the performance of dynamic hedging for variable annuity products, insurance companies rely on nested stochastic projections, which is highly computationally intensive and often prohibitive for large variable annuity portfolios. Metamodeling techniques have recently been proposed to address the computational issues. However, it is difficult for researchers to obtain real datasets from insurance companies to test metamodeling techniques and publish the results in academic journals. In this paper, we create synthetic datasets that can be used for the purpose of addressing the computational issues associated with the nested stochastic valuation of large variable annuity portfolios. The runtime used to create these synthetic datasets would be about three years if a single CPU were used. These datasets are readily available to researchers and practitioners so that they can focus on testing metamodeling techniques.

Keywords: Monte Carlo; regime-switching multivariate Black–Scholes; metamodeling; variable annuity; portfolio valuation

1. Introduction

A variable annuity (VA) is a popular life insurance product created by insurance companies to address many people's concerns about outliving their assets [1,2]. Under a VA policy, the policyholder agrees to make a lump-sum or a series of purchase payments to the insurer and in return the insurer agrees to make benefit payments to the policyholder, beginning either immediately or on a future date. Policyholders choose to invest their money in one or more investment funds provided by the insurance company. A main feature of VAs is that they come with guarantees or riders, which are designed to protect the policyholder's capital against market downturns.

There are two types of guaranteed benefits embedded in VA policies: death benefits and living benefits. A guaranteed minimum death benefit (GMDB) guarantees a specified amount to the beneficiary upon the death of the policyholder regardless of the performance of the investment portfolio. Examples of living benefits include the guaranteed minimum withdrawal benefit (GMWB), the guaranteed minimum income benefit (GMIB), the guaranteed minimum maturity benefit (GMMB), and the guaranteed minimum accumulation benefit (GMAB). A GMWB guarantees that the policyholder can take systematic annual withdrawals of a specified amount from the policy over a period of time, even though the investment portfolio might be depleted. A GMIB guarantees that the policyholder can convert the VA policy to an annuity according to a specified rate. A GMMB guarantees that the policyholder can receive a specific amount at the maturity of the policy. A GMAB guarantees that the policyholder can renew the contract during a specified window after a specified waiting period.

Due to these attractive guarantees, many VA policies were sold in the past two decades. Figure 1 shows the annual VA sales in the United States during the period from 2008 to 2017. In the figure, we see that, except for 2017, the annual sales in all these years were above $100 billion. The guarantees embedded in VA policies are financial guarantees and cannot be adequately addressed by traditional actuarial methods [3]. To mitigate the financial risks associated with the VA guarantees, many insurance companies with a VA business have adopted dynamic hedging [4,5].

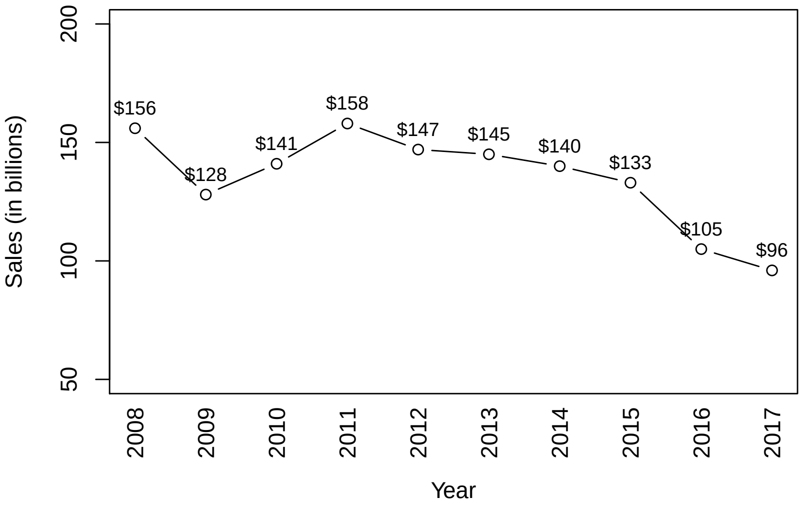

Figure 1. Variable annuity sales in the United States from 2008 to 2017. The numbers are obtained from LIMRA Secure Retirement Institute.

To simulate the performance of dynamic hedging for VA products, insurance companies rely on nested stochastic projections [5]. Nested stochastic projections are also referred to as "stochastic on stochastic" projections. Figure 2 conceptualizes the structure of a typical nested stochastic projection, which involves two layers of stochastic projections. At each node of an outer stochastic path, a set of inner stochastic paths is embedded. Usually the outer stochastic paths are real-world scenarios, which reflect a realistic pattern of underlying market prices that are used to generate realistic distributions of outcomes. In contrast, the inner stochastic paths are risk-neutral scenarios, which use unrealistic assumptions about risk premiums for purposes of calculating derivative prices under the no-arbitrage assumption.

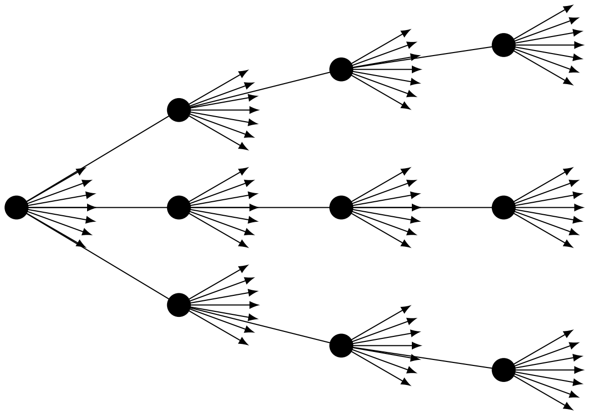

Figure 2. Nested stochastic projections.

The computation of nested stochastic projections for a large VA portfolio is highly computationally intensive and often prohibitive because every policy in the portfolio needs to be projected over many paths for a long time horizon [6]. For example, if we use 1000 real-world scenarios in the outer layer and 1000 risk-neutral paths in the inner layer, and project the cash flows at yearly steps for 30 years, then the total number of projections for each policy is

$$1000 \times 1000 \times 30 \times 31/2 = 4.65 \times 10^8,$$

which is already a big number. For a portfolio of 100,000 contracts, the number of projections would be 4.65×10^{13}. Suppose that a single CPU can process 200,000 cash flow projections in a second. Then, it will take this CPU

$$\frac{4.65 \times 10^{13}}{200,000 \times 3600 \times 24 \times 365} = 7.37$$

years to process all the cash flow projections for the portfolio. The amount of time shown in the above equation is just the runtime used to project the cash flows once. To calculate the Greeks, we need to project the cash flows multiple times at different shocks of the market. This will increase the runtime multifold.

Recently, metamodeling approaches have been proposed to address the computational issues associated with the valuation of large VA portfolios (see, for example, [7–16]). The main idea of these metamodeling approaches is to build a predictive model based on a set of representative VA policies and their fair market values (or other quantities of interest). The predictive model is then used to estimate the fair market values for all the policies in the portfolio. This can reduce the number of policies that are valued by Monte Carlo simulation. Since predictive models are usually much faster than Monte Carlo simulation, the gain in valuation runtime is significant.

However, it is difficult for academic researchers to obtain real datasets from insurance companies to assess the performance of metamodeling techniques. In this paper, we create synthetic datasets that can be used by researchers and practitioners to test metamodeling methods for the efficient valuation of large VA portfolios under nested stochastic simulation. In particular, we implement a nested stochastic valuation engine that is used to calculate the Greeks for VA policies along outer layer paths. The purpose of this work is to relieve researchers from spending time on creating such datasets, which can be extremely time-consuming to create. This paper differs from a previous paper [10] in that this paper focuses on developing comprehensive synthetic datasets while the previous paper focuses on metamodeling. This paper also differs from the paper [17] because this paper is about creating synthetic datasets under the stochastic-on-stochastic valuation framework while the paper [17] is about creating synthetic datasets for valuation only at time zero.

The remaining part of this paper is structured as follows. Section 2 presents a nested stochastic simulation engine for valuing the guarantees embedded in variable annuities. In Sections 3 and 4, we present synthetic datasets that can be used to test the performance of metamodeling techniques. In Section 5, we conclude the paper with some remarks. The software that implements the nested Monte Carlo simulation engine is described in Appendix A.

2. Nested Stochastic Valuation

In this section, we describe the nested stochastic valuation engine. In particular, we introduce the risk-neutral scenario generator, the real-world scenario generator, and the cash flow projections.

2.1. Risk-Neutral Scenario Generator

Risk-neutral scenarios are used in the inner loop to calculate the dollar Deltas. We use a multivariate Black–Scholes model introduced by Carmona and Durrelman [18] to generate risk-neutral scenarios. This model is also described in [17].

Let $S^{(1)}, S^{(2)}, \ldots, S^{(k)}$ be k indices in the financial market. Under the multivariate Black–Scholes model, the risk-neutral dynamics of the k indices are given by [18]:

$$\frac{\mathrm{d}\, S_t^{(h)}}{S_t^{(h)}} = r_t \,\mathrm{d}\, t + \sum_{l=1}^{k} \sigma_{hl} \,\mathrm{d}\, B_t^{(l)}, \quad S_0^{(h)} = 1,\ h = 1, 2, \ldots, k \tag{1}$$

or

$$S_t^{(h)} = \exp\left[\left(\int_0^t r_s \,\mathrm{d}\, s - \frac{t}{2}\sum_{l=1}^{k}\sigma_{hl}^2\right) + \sum_{l=1}^{k}\sigma_{hl} B_t^{(l)}\right], \quad h = 1, 2, \ldots, k, \tag{2}$$

where $B_t^{(1)}, B_t^{(2)}, \ldots, B_t^{(k)}$ are independent standard Brownian motions, r_t is the short rate of interest, and the matrix (σ_{hl}) is used to capture the correlation among the indices.

Let $t_0 = 0, t_1 = \Delta, \ldots, t_m = m\Delta$ be time steps with equal space Δ and suppose that the continuous forward rate is constant within each period. For $j = 1, 2, \ldots, m$, the accumulation factor of the hth index for the period (t_{j-1}, t_j) can be calculated as:

$$A_j^{(h)} = \frac{S_{j\Delta}^{(h)}}{S_{(j-1)\Delta}^{(h)}} = \exp\left[\left(f_j - \frac{1}{2}\sum_{l=1}^{k}\sigma_{hl}^2\right)\Delta + \sum_{l=1}^{k}\sigma_{hl}\sqrt{\Delta} Z_j^{(l)}\right], \tag{3}$$

where f_j is the annualized continuous forward rate for period (t_{j-1}, t_j) and

$$Z_j^{(l)} = \frac{B_{j\Delta}^{(l)} - B_{(j-1)\Delta}^{(l)}}{\sqrt{\Delta}}.$$

By the property of Brownian motion, we know that $Z_1^{(l)}, Z_2^{(l)}, \ldots, Z_m^{(l)}$ are independent random variables with a standard normal distribution.

The continuous return for the period (t_{j-1}, t_j) is calculated as:

$$R_j^{(h)} = \ln A_j^{(h)} = \left(f_j - \frac{1}{2}\sum_{l=1}^{k}\sigma_{hl}^2\right)\Delta + \sum_{l=1}^{k}\sigma_{hl}\sqrt{\Delta} Z_j^{(l)}. \tag{4}$$

The matrix

$$\sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{pmatrix}$$

can be obtained from the following Cholesky decomposition of the covariance matrix Σ:

$$\sigma \cdot \sigma' = \Sigma = \mathrm{diag}(\boldsymbol{\nu})\mathbf{R}\,\mathrm{diag}(\boldsymbol{\nu}), \tag{5}$$

where $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_k)$ is a vector of index volatilities, $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix with $\boldsymbol{\nu}$ as diagonal elements, and $\mathbf{R}$ is the correlation matrix. In matrix form, Equation (4) can be expressed as

$$\begin{pmatrix} R_j^{(1)} \\ R_j^{(2)} \\ \vdots \\ R_j^{(k)} \end{pmatrix} = \begin{pmatrix} f_j - \frac{1}{2}\nu_1^2 \\ f_j - \frac{1}{2}\nu_2^2 \\ \vdots \\ f_j - \frac{1}{2}\nu_k^2 \end{pmatrix}\Delta + \sigma \begin{pmatrix} Z_j^{(1)} \\ Z_j^{(2)} \\ \vdots \\ Z_j^{(k)} \end{pmatrix}\sqrt{\Delta}. \tag{6}$$

Algorithm 1 shows the pseudo-code of the risk-neutral scenario generator. Once we have index scenarios simulated from Equation (3), we can obtain the investment fund scenarios by blending these index scenarios as follows:

$$F_{ij}^{(h)} = \sum_{l=1}^{k} w_{hl} A_{ij}^{(l)}, \quad h = 1, 2, \ldots, g,$$

where g is the number of investment funds and

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{21} & w_{22} & \cdots & w_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{g1} & w_{g2} & \cdots & w_{gk} \end{pmatrix}$$

is the fund mapping that maps the k indices to the g investment funds.

Algorithm 1: Pseudo-Code of the Risk-Neutral Scenario Generator.

```
Input: Forward rates $\mathbf{f}$, volatilities $\boldsymbol{\nu}$, correlation matrix $\mathbf{R}$, seed $s$, $n$, $m$, $\Delta$
Output: Scenario matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(k)}$
Set the seed of the random number generator to be $s$;
Calculate covariance matrix $\Sigma$;
Calculate the Cholesky decomposition $\sigma$;
for $i = 1$ to $n$ do
    for $j = 1$ to $m$ do
        Generate a vector of normal random number $(z_1, z_2, \ldots, z_k)$;
        Get a vector of index returns $(R_1, R_2, \ldots, R_k)$ by Equation (6);
        Let $A_{ij}^{(h)} \leftarrow \exp(R_h)$, where $A_{ij}^{(h)}$ is the $(i, j)$-th entry of $\mathbf{A}^{(h)}$;
    end
end
Save the scenario matrices into files.
```

2.2. Real-World Scenario Generator

Real-world scenarios are used in the outer loop to simulate the movements of the market. Risk-neutral scenarios are prospective and parameters of the risk-neutral scenario generator are calibrated to market data. Real-world scenarios are retrospective and the parameters of a real-world scenario generator are calibrated to historical data.

In practice, the regime-switching model [19] is typically used to generate real-world scenarios. Here, we introduce a multivariate two-regime regime-switching model for generating correlated real-world scenarios for multiple indices. Within a regime and a time period, the evolution of the indices follows the multivariate log-normal model.

Let ρ_t denote the regime at time t and M denote the transition matrix, i.e.,

$$M = \begin{pmatrix} p_{1,1} & p_{1,2} \\ p_{2,1} & p_{2,2} \end{pmatrix},$$

where

$$p_{s,r} = P(\rho_{t+1} = r | \rho_t = s), \quad , s, r = 1, 2.$$

Let $\pi = (\pi_1, \pi_2)$ be the unconditional probability distribution of the regime-switching process. Then, we have

$$\pi M = \pi,$$

which gives [19]:

$$\pi_1 = \frac{p_{2,1}}{p_{1,2} + p_{2,1}}, \quad \pi_2 = \frac{p_{1,2}}{p_{1,2} + p_{2,1}}. \tag{7}$$

Let $S^{(1)}$, $S^{(2)}$, ..., $S^{(k)}$ be k indices in the financial market. Under the multivariate two-regime regime-switching log-normal model, the risk-world dynamics of the k indices are given by:

$$S_t^{(h)}\Big|_{\rho_t} = \exp\left[\mu_h^{(\rho_t)} t + \sum_{l=1}^{k} \sigma_{hl}^{(\rho_t)} B_t^{(l)}\right], \quad S_0^{(h)} = 1, \quad h = 1, 2, \ldots, k, \tag{8}$$

where $B_t^{(1)}$, $B_t^{(2)}$, ..., $B_t^{(k)}$ are independent standard Brownian motions, $\mu_h^{(\rho_t)}$ is the geometric mean of the hth index in the regime ρ_t, the matrix $(\sigma_{hl}^{(\rho_t)})$ is used to capture the correlation among the indices in the regime ρ_t, and $\rho_t \in \{1, 2\}$ is the regime number.

Let $t_0 = 0$, $t_1 = \Delta$, ..., $t_m = m\Delta$ be time steps, where Δ is the time step. Then, for $j = 1, 2, \ldots, m$, the accumulation factor of the hth index for the period (t_{j-1}, t_j) in the regime ρ_j can be calculated as

$$A_j^{(h)}\Big|_{\rho_j} = \frac{S_{j\Delta}^{(h)}\Big|_{\rho_j}}{S_{(j-1)\Delta}^{(h)}\Big|_{\rho_j}} = \exp\left[\mu_h^{(\rho_t)} \Delta + \sum_{l=1}^{k} \sigma_{hl}^{(\rho_t)} \sqrt{\Delta} Z_j^{(l)}\right], \tag{9}$$

where

$$Z_j^{(l)} = \frac{B_{j\Delta}^{(l)} - B_{(j-1)\Delta}^{(l)}}{\sqrt{\Delta}}.$$

In matrix form, the returns can be expressed as

$$\left.\begin{pmatrix} R_j^{(1)} \\ R_j^{(2)} \\ \vdots \\ R_j^{(k)} \end{pmatrix}\right|_{\rho_j} = \left.\begin{pmatrix} \ln A_j^{(1)} \\ \ln A_j^{(2)} \\ \vdots \\ \ln A_j^{(k)} \end{pmatrix}\right|_{\rho_j} = \begin{pmatrix} \mu_1^{(\rho_j)} \\ \mu_2^{(\rho_j)} \\ \vdots \\ \mu_k^{(\rho_j)} \end{pmatrix} \Delta + \sigma^{(\rho_j)} \begin{pmatrix} Z_j^{(1)} \\ Z_j^{(2)} \\ \vdots \\ Z_j^{(k)} \end{pmatrix} \sqrt{\Delta}, \tag{10}$$

where the matrix $\sigma^{(\rho_j)}$ can be obtained from the following Cholesky decomposition of the covariance matrix $\Sigma^{(\rho_j)}$:

$$\sigma^{(\rho_j)} \cdot \sigma^{(\rho_j)'} = \Sigma^{(\rho_j)} = \operatorname{diag}(\boldsymbol{\nu}^{(\rho_j)}) \mathbf{R}^{(\rho_j)} \operatorname{diag}(\boldsymbol{\nu}^{(\rho_j)}), \tag{11}$$

where $\boldsymbol{\nu}^{(\rho_j)}$ is a vector of index volatilities for regime ρ_j, $\operatorname{diag}(\mathbf{v}^{(\rho_j)})$ is a diagonal matrix with $\boldsymbol{\nu}^{(\rho_j)}$ as diagonal elements, and $\mathbf{R}^{(\rho_j)}$ is the correlation matrix for regime ρ_j.

Let ρ_0 be the initial regime. Then, for $j = 1, 2, \ldots, m$, the regime for period j can be determined by generating a uniform random number u as follows:

$$\rho_j = \begin{cases} 2, & \text{if } \rho_{j-1} = 1 \text{ and } u \le p_{1,2}; \\ 1, & \text{if } \rho_{j-1} = 1 \text{ and } u > p_{1,2}; \\ 1, & \text{if } \rho_{j-1} = 2 \text{ and } u \le p_{2,1}; \\ 2, & \text{if } \rho_{j-1} = 2 \text{ and } u > p_{2,1}. \end{cases} \tag{12}$$

The continuous return of the hth index for the period (t_{j-1}, t_j) in the regime ρ_j is calculated

$$R_j^{(h)}\Big|_{\rho_j} = \ln A_j^{(h)}\Big|_{\rho_j} = \mu_h^{(\rho_t)} \Delta + \sum_{l=1}^{k} \sigma_{hl}^{(\rho_t)} \sqrt{\Delta} Z_j^{(l)}. \tag{13}$$

Algorithm 2 shows the pseudo-code of the two-regime regime-switching real-world scenario generator.

Algorithm 2: Pseudo-code of the two-regime regime-switching real-world scenario generator.

```
Input: $p_{1,2}$, $p_{2,1}$, $n$, $m$, $\Delta$, seed $s$, regime-1 volatilities $\mathbf{v}^{(1)}$, regime-2 volatilities $\mathbf{v}^{(2)}$, regime-1
       correlation $\mathbf{R}^{(1)}$, regime-2 correlation $\mathbf{R}^{(2)}$, regime-1 geometric means $\boldsymbol{\mu}^{(1)}$, regime-2
       geometrix means $\boldsymbol{\mu}^{(2)}$
Output: Scenario matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(k)}$
Set the seed of the random number generator to be $s$;
Calculate $\pi_1$ and $\pi_2$ by Equation (7);
Calculate covariance matrices $\Sigma^{(1)}$ and $\Sigma^{(2)}$;
Calculate the Cholesky decompositions $\sigma^{(1)}$ and $\sigma^{(2)}$;
for $i = 1$ to $n$ do
    Generate a uniform random number $u$;
    Let $\rho_0 = 1$ if $u \le \pi_1$, otherwise $\rho_0 = 2$;
    for $j = 1$ to $m$ do
        Generate a uniform random number $u$;
        Determine the regime $\rho_j$ according to Equation (12);
        Generate a vector of normal random number $(z_1, z_2, \ldots, z_k)$;
        Get a vector of index returns $(R_1, R_2, \ldots, R_k)$ by Equation (10);
        Let $A_{ij}^{(h)} \leftarrow \exp(R_h)$, where $A_{ij}^{(h)}$ is the $(i, j)$-th entry of $\mathbf{A}^{(h)}$;
    end
end
Save the scenario matrices into files.
```

From the above equation, we can derive the expectations and covariances of the conditioned returns as follows:

$$E\left[R_j^{(h)}\Big|_{\rho_j} \right] = \mu_h^{(\rho_t)}\Delta$$

and

$$\mathrm{Cov}\left(R_j^{(h)}\Big|_{\rho_j}, R_j^{(m)}\Big|_{\rho_j} \right) = \sum_{l=1}^{k} \sigma_{hl}^{(\rho_t)} \sigma_{ml}^{(\rho_t)} \Delta.$$

The return of the hth index for the period (t_{j-1}, t_j) can be expressed as

$$R_j^{(h)} = R_j^{(h)}\Big|_{\rho_j=1} I_{\{\rho_j=1\}} + R_j^{(h)}\Big|_{\rho_j=2} I_{\{\rho_j=2\}}, \tag{14}$$

where I is an indicator function.

The expected return for the period (t_{j-1}, t_j) can be calculated as p177 in [20]:

$$\begin{aligned} E\left[R_j^{(h)} \right] &= E\left[R_j^{(h)}\Big|_{\rho_j=1} \right] P(\rho_j = 1) + E\left[R_j^{(h)}\Big|_{\rho_j=2} \right] P(\rho_j = 2) \\ &= \mu_h^{(1)} \Delta \pi_1 + \mu_h^{(2)} \Delta \pi_2. \end{aligned} \tag{15}$$

From Equation (14), we have

$$R_j^{(h)} R_j^{(m)} = R_j^{(h)}\Big|_{\rho_j=1} R_j^{(m)}\Big|_{\rho_j=1} I_{\{\rho_j=1\}} + R_j^{(h)}\Big|_{\rho_j=2} R_j^{(m)}\Big|_{\rho_j=2} I_{\{\rho_j=2\}},$$

which gives

$$\begin{aligned} E\left[R_j^{(h)} R_j^{(m)}\right] &= \pi_1 \left(\mu_h^{(1)} \mu_m^{(1)} \Delta^2 + \sum_{l=1}^{k} \sigma_{hl}^{(1)} \sigma_{ml}^{(1)} \Delta\right) + \\ &\quad \pi_2 \left(\mu_h^{(2)} \mu_m^{(2)} \Delta^2 + \sum_{l=1}^{k} \sigma_{hl}^{(2)} \sigma_{ml}^{(2)} \Delta\right). \end{aligned}$$

Therefore, we have

$$\begin{aligned} \operatorname{Cov}\left(R_j^{(h)}, R_j^{(m)}\right) &= E\left[R_j^{(h)} R_j^{(m)}\right] - E\left[R_j^{(h)}\right] E\left[R_j^{(m)}\right] \\ &= \pi_1 \pi_2 \Delta^2 \left(\mu_h^{(1)} - \mu_h^{(2)}\right) \left(\mu_m^{(1)} - \mu_m^{(2)}\right) + \\ &\quad \pi_1 \sum_{l=1}^{k} \sigma_{hl}^{(1)} \sigma_{ml}^{(1)} \Delta + \pi_2 \sum_{l=1}^{k} \sigma_{hl}^{(2)} \sigma_{ml}^{(2)} \Delta. \end{aligned} \tag{16}$$

Letting $m = h$ in the above equation, we get the variance of the return as follows:

$$\begin{aligned} &\operatorname{Var}\left(R_j^{(h)}\right) \\ = \; & E\left[\left(R_j^{(h)}\right)^2\right] - \left(E\left[R_j^{(h)}\right]\right)^2 \\ = \; & \pi_1 \pi_2 \Delta^2 \left(\mu_h^{(1)} - \mu_h^{(2)}\right)^2 + \pi_1 \sum_{l=1}^{k} \left(\sigma_{hl}^{(1)}\right)^2 \Delta + \pi_2 \sum_{l=1}^{k} \left(\sigma_{hl}^{(2)}\right)^2 \Delta \\ = \; & \pi_1 \pi_2 \Delta^2 \left(\mu_h^{(1)} - \mu_h^{(2)}\right)^2 + \pi_1 \Delta \left(\nu_h^{(1)}\right)^2 + \pi_2 \Delta \left(\nu_h^{(2)}\right)^2, \end{aligned} \tag{17}$$

where $\nu_h^{(\rho_j)}$ is the volatility of the hth index in the regime ρ_j, i.e.,

$$\nu_h^{(\rho_j)} = \sqrt{\sum_{l=1}^{k} \left(\sigma_{hl}^{(\rho_j)}\right)^2}.$$

Equations (15) and (17) can be used to validate the real-world scenarios. These equations can also be used to specify parameters for the two-regime regime-switching model if we want to control the overall mean returns and the overall volatilities of the indices.

2.3. Nested Stochastic Valuation

To describe how nested stochastic projections are done, we let N_1 be the number of outer loop paths and let T_1 be the number of time nodes in the outer loop. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a portfolio of n VA policies. Algorithm 3 shows a high-level sketch of the nested stochastic valuation engine. At each node along each outer loop path, we calculate the fair market values of each policy using the risk-neutral scenarios. For details about how policies are aged along a real-world path and how the cash flows are projected along a risk-neutral path, readers are referred to [17].

Algorithm 3: A High-Level Sketch of the Nested Stochastic Valuation Engine.

```
Input: A portfolio $X$ of VA policies, risk-neutral scenarios, real-world scenarios, fund mapping,
       mortality tables
Output: Matrices of partial dollar deltas
for $p = 1$ to $N_1$ do
    for $j = 1$ to $T_1$ do
        for $i$ to $n$ do
            Age $\mathbf{x}_i$ to time $t_j$ along the $p$th real-world path;
            Calculate the fair market value of the aged policy with all indices shocked up 1%;
            Calculate the fair market value of the aged policy with all indices shocked down 1%;
            Calculate the total dollar delta of the aged policy;
            Let $M_{i,j}^{(p,h)}$ be the partial dollar delta of $\mathbf{x}_i$ on the $h$th index at time $t_j$ along the $p$th
              real-world path;
        end
    end
end
Return the matrices of partial dollar deltas $M^{(p,j)}$.
```

To assess the performance of dynamic hedging, partial dollar deltas are required as hedging is done by individual tradable indices. The partial dollar delta on the hth index is normally calculated as follows:

$$\frac{FMV(\ldots, AV_{h-1}, 1.01AV_h, AV_{h+1}, \ldots) - FMV(\ldots, AV_{h-1}, 0.99AV_h, AV_{h+1}, \ldots)}{0.02},$$

where AV_h denotes the account value invested in the hth index. However, calculating partial dollar deltas using the above equation requires projecting cash flows at many index shocks. This is prohibitive under the nested stochastic valuation framework.

To reduce the runtime, we only calculate total dollar delta at each node along an outer loop path as follows:

$$\Delta = \frac{FMV(1.01AV_1, \ldots, 1.01AV_k) - FMV(0.99AV_1, \ldots, 0.99AV_k)}{0.02},$$

where k is the number of indices. Then, we approximate the partial dollar deltas as follows:

$$\Delta^{(h)} = \frac{AV_h}{AV_1 + \cdots + AV_k}\Delta. \tag{18}$$

The relation given in Equation (18) can be derived as follows. Suppose that the fair market value of the guarantees embedded in a VA policy is a function of the total account value, i.e.,

$$FMV = f(AV_1 + AV_2 + \cdots + AV_k).$$

Then, the partial dollar delta on the hth index is calculated as

$$\Delta^{(h)} = \frac{\partial f}{\partial AV_h} AV_h = \frac{\partial f}{\partial TA} \cdot \frac{\partial TA}{\partial AV_h} \cdot AV_h = \frac{\partial f}{\partial TA} \cdot TA \cdot \frac{AV_h}{TA} = \Delta \cdot \frac{AV_h}{TA},$$

where $TA = AV_1 + \cdots AV_k$ is the total account value.

3. Synthetic Portfolio and Payoffs

In this section, we describe the synthetic portfolio and the payoffs of the guarantees embedded in the VA policies.

3.1. Synthetic Portfolio

We adopted a subset of the synthetic VA portfolio created in [17]. That synthetic portfolio contains 19 types of products, each of which has 10,000 policies. We selected 2000 policies from each product type. The subset contains 38,000 policies. Readers are referred to [17] for a description of the features or variables of the VA policies.

Table 1 shows the number of policies in each product type by gender. About 40% of the policies in each product are female. Table 2 shows the summary statistics of some numerical fields. In the table, we see that all funds have many zeros. This is because many policies generally do not invest in all the funds. The age is the number of years between the birth date and the current date. The time to maturity is calculated from the current date and the maturity date.

Table 1. Distribution of gender by product type.

Gender	ABRP	ABRU	ABSU	DBAB	DBIB	DBMB	DBRP
F	779	768	809	787	759	833	805
M	1221	1232	1191	1213	1241	1167	1195
Gender	**DBRU**	**DBSU**	**DBWB**	**IBRP**	**IBRU**	**IBSU**	**MBRP**
F	785	782	789	782	826	824	775
M	1215	1218	1211	1218	1174	1176	1225
Gender	**MBRU**	**MBSU**	**WBRP**	**WBRU**	**WBSU**		
F	798	798	813	812	781		
M	1202	1202	1187	1188	1219		

Table 2. Summary statistics of some fields. Note that `age` and `ttm` are calculated from the birth date, valuation date, and maturity date.

	Min	1st Q	Mean	3rd Q	Max
`gbAmt`	0.00	187,601.23	327,213.71	446,403.84	1,060,311.72
`gmwbBalance`	0.00	0.00	35,501.02	0.00	499,708.73
`withdrawal`	0.00	0.00	22,605.36	0.00	499,585.73
`FundValue1`	0.00	0.00	33,632.30	50,083.85	798,936.37
`FundValue2`	0.00	0.00	38,673.18	57,221.55	1,026,213.34
`FundValue3`	0.00	0.00	26,778.14	39,154.69	752,945.34
`FundValue4`	0.00	0.00	26,231.25	39,331.61	566,338.64
`FundValue5`	0.00	0.00	22,768.91	34,841.42	481,399.12
`FundValue6`	0.00	0.00	35,386.64	52,585.38	1,042,335.65
`FundValue7`	0.00	0.00	29,898.18	44,511.19	806,540.12
`FundValue8`	0.00	0.00	30,303.87	45,505.06	704,720.85
`FundValue9`	0.00	0.00	29,983.68	44,034.63	851,307.63
`FundValue10`	0.00	0.00	30,092.13	45,276.64	691,822.70
`age`	34.52	42.11	49.56	56.96	64.46
`ttm`	0.59	10.26	14.49	18.68	28.52

3.2. Guarantee Payoffs

We calculated the payoffs of the guarantees for the portfolio along each of the 1000 real-world path. The payoff is calculated as the sum of the death benefit and the living benefit. Figure 3 shows the guarantee payoffs of the portfolio at each month along the 1000 real-world path. In the figure, we see that there are some relatively large guarantee payoffs after the 300th month at some real-world paths. The large payoffs are caused the GMAB products, which allow policyholders to renew. A histogram of the present values of the guarantee payoffs along the real-world paths is shown in Figure 4. From the histogram, we see that the distribution of these present values is positively skewed.

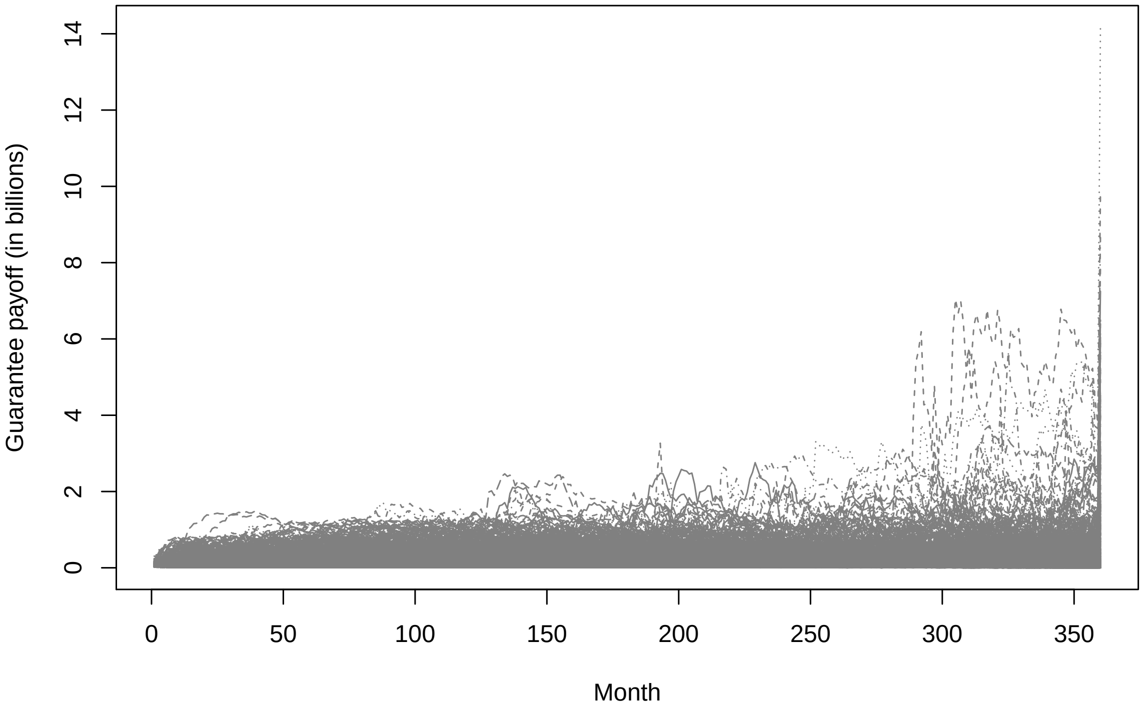

Figure 3. Guarantee payoffs along the 1000 real-world paths.

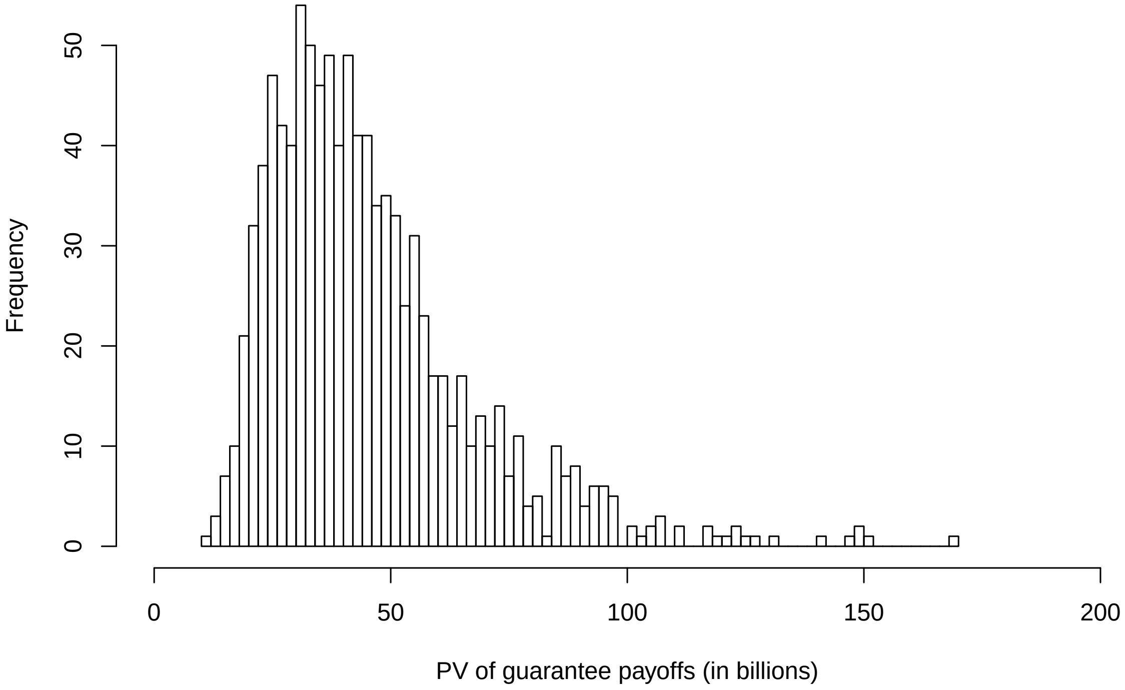

Figure 4. Present values of the guarantee payoffs.

We also calculated the present values of the guarantee payoffs along each of the 1000 real-world paths. Figure 5 shows a histogram of these present values. The histogram shows that the distribution of the present values is positively skewed. At some bad real-world paths, the guarantee payoffs are much larger than those at other real-world paths. Table 3a shows some summary statistics of these present values. At the best real-world path, the guarantee payoff of this portfolio is 11,421 millions. If the worst real-world path occurs, the guarantee payoff of this portfolio is 168,574 millions.

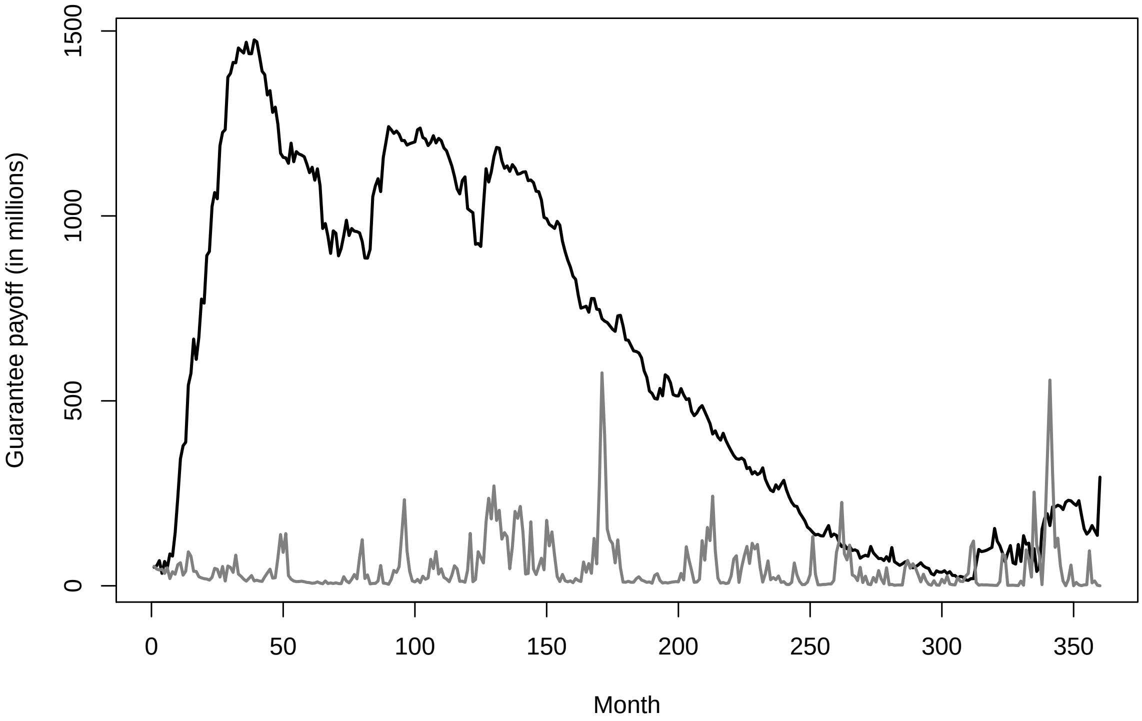

Figure 5. Guarantee payoffs along the worst (the dark line) and the best (the gray line) real-world paths.

Table 3. Summary statistics and conditional tail expectations of the present values of the guarantee payoffs. The numbers are in millions.

(a)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
11,421	30,393	40,556	45,535	54,793	168,574

(b)

CTE50	CTE75	CTE95
61,332.68	75,594.85	108,113.29

Table 3b shows the conditional tail expectations (CTEs) of the present values of the guarantee payoffs at three different levels. The CTE75 is calculated as the mean of present values from the worst $0.25 \times 1000 = 250$ real-world paths. In the table, we see that the CTE75 is around 75,595 millions.

Figure 6 shows the 1000 real-world paths of the five indices, which are the large cap equity, the small cap equity, the international equity, the fixed income, and the money market. The dark thick line in each subfigure corresponds to the worst real-world path, which produces the largest present value of the guarantee payoffs. The gray thick line in each subfigure corresponds to the best real-world path, which produces the lowest present value of the guarantee payoffs. In the figure, we see that the best path is above the worst path.

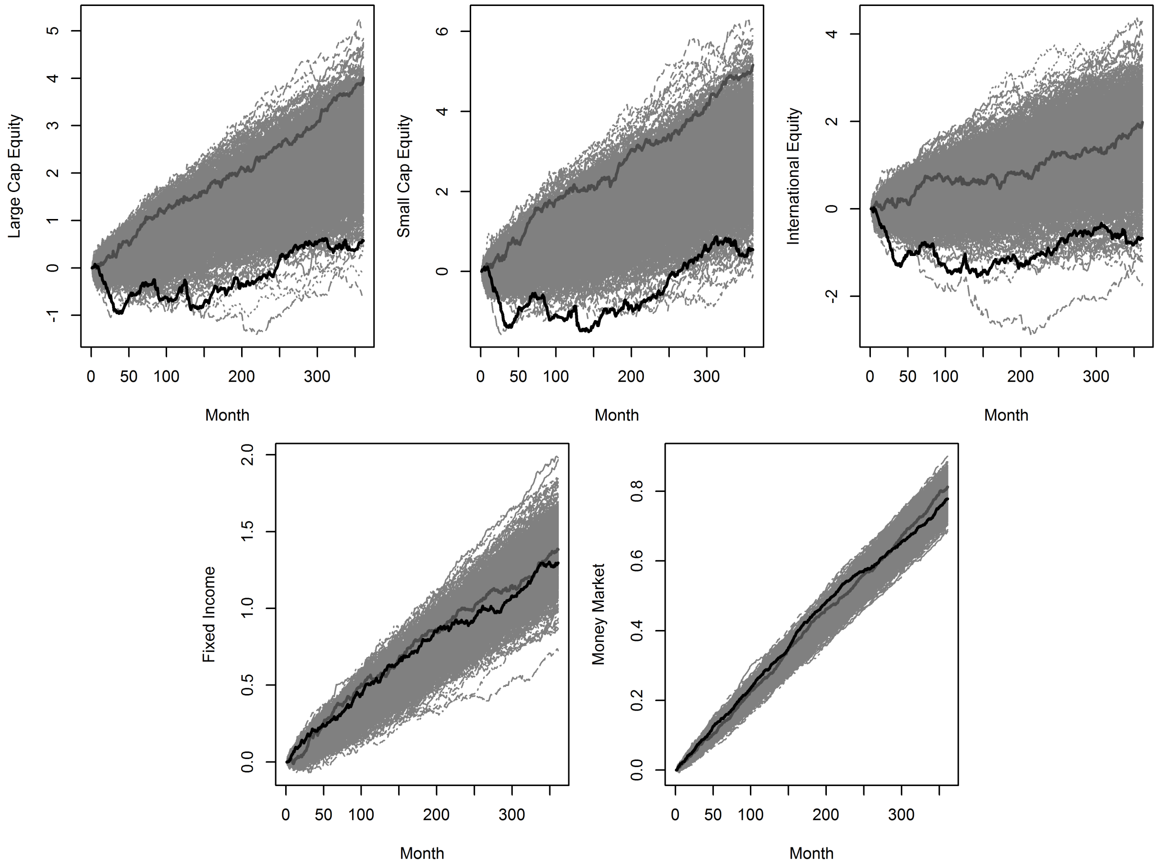

Figure 6. Real-world paths of the indices. The dark thick line is the worst real-world path. The gray thick line is the best real-world path.

Note that the best real-world path is not the one at the very top and the worst real-world path is not the one at the very bottom. This is because the payoffs of GMAB products in bull market are large. In other words, if the real-world path at the very top occurs, the GMAB products will incur large payoffs because the policyholders can renew by reseting the benefit to the higher of the account value and the existing benefit base.

Figure 5 shows the guarantee payoffs at monthly steps along the best and the worst real-world paths. In the figure, we see that in general the payoffs along the best real-world path are higher than those along the worst real-world path. At a few months near the end of the projection horizon, the payoffs at the best path are higher than those at the worst path. This is caused by the GMAB products, which have higher payoffs at better markets due to the renew feature.

4. Partial Dollar Deltas

In this section, we present the partial dollar deltas calculated by the nested Monte Carlo simulation method described in Section 2.

As discussed in Section 2, the nested stochastic valuation program produces many matrices of the partial dollar deltas. In fact, the program produces $N_1 \times H$ matrices of partial dollar deltas, where N_1 is the number of real-world paths and H is the number of indices. For $p = 1, 2, \ldots, N_1$ and $h = 1, 2, \ldots, H$, let $M^{(p,h)}$ be the matrix of the partial dollar deltas on the hth index:

$$M^{(p,h)} = \begin{pmatrix} M_{1,1}^{(p,h)} & M_{1,2}^{(p,h)} & \cdots & M_{1,T}^{(p,h)} \\ M_{2,1}^{(p,h)} & M_{2,2}^{(p,h)} & \cdots & M_{2,T}^{(p,h)} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1}^{(p,h)} & M_{n,2}^{(p,h)} & \cdots & M_{n,T}^{(p,h)} \end{pmatrix}, \tag{19}$$

where n is the number of policies in the portfolio, T is the number of time points where partial dollar deltas are calculated, and $M_{i,j}^{(p,h)}$ denotes the partial dollar delta of the ith policy on the hth index at jth evaluation time point along the pth real-world path. Since we used $N_1 = 1000$ real-world paths and $T = 31$ evaluation time points and the number of indices is $H = 5$, the number of matrices we produced is 5000. Each matrix has a size of $38,000 \times 31$. We saved all the matrices to CSV files with only six decimal places. If zip all the CSV files, the size of the zip file is around 20 GB.

4.1. Aggregate Results

The aggregate partial dollar deltas along a real-world path are calculated as follows:

$$M_j^{(p,h)} = \sum_{i=1}^{n} M_{i,j}^{(p,h)}. \tag{20}$$

In other words, the aggregate partial dollar deltas are the partial dollar deltas of the whole portfolio. The aggregate total dollar deltas are calculated as

$$M_j^{(p)} = \sum_{h=1}^{H} \sum_{i=1}^{n} M_{i,j}^{(p,h)}. \tag{21}$$

Figure 7 shows the aggregate partial dollar deltas and aggregate total dollar deltas along the 1000 real-world paths. In the figure, we have the following observations:

- The aggregate partial dollar deltas do not approach zero at the end of the projection horizon. This is caused by the GMAB products, which behave similar to call options.
- The guarantees are more sensitive to indices with higher volatilities. For example, the magnitudes of the aggregate partial dollar deltas on the small cap equity are larger than those on other indices.
- The aggregate total dollar deltas along the best and the worst real-world paths have similar magnitudes. This is because the dollar deltas of the GMAB product offset those of other products.
- For equity indices, which have high volatilities, the aggregate partial dollar deltas have similar magnitudes along the best and the worst real-world paths. For non-equity indices, which have low volatilities, the aggregate partial dollar deltas along the worst real-world path have higher magnitudes than those along the best real-world path.

Figures 8 and 9 show the histograms of the aggregate partial dollar deltas at the Year 1 and the Year 30, respectively. In the figures, we see that the distributions of the aggregate partial dollar deltas at the Year 30 is more skewed that those at the Year 1.

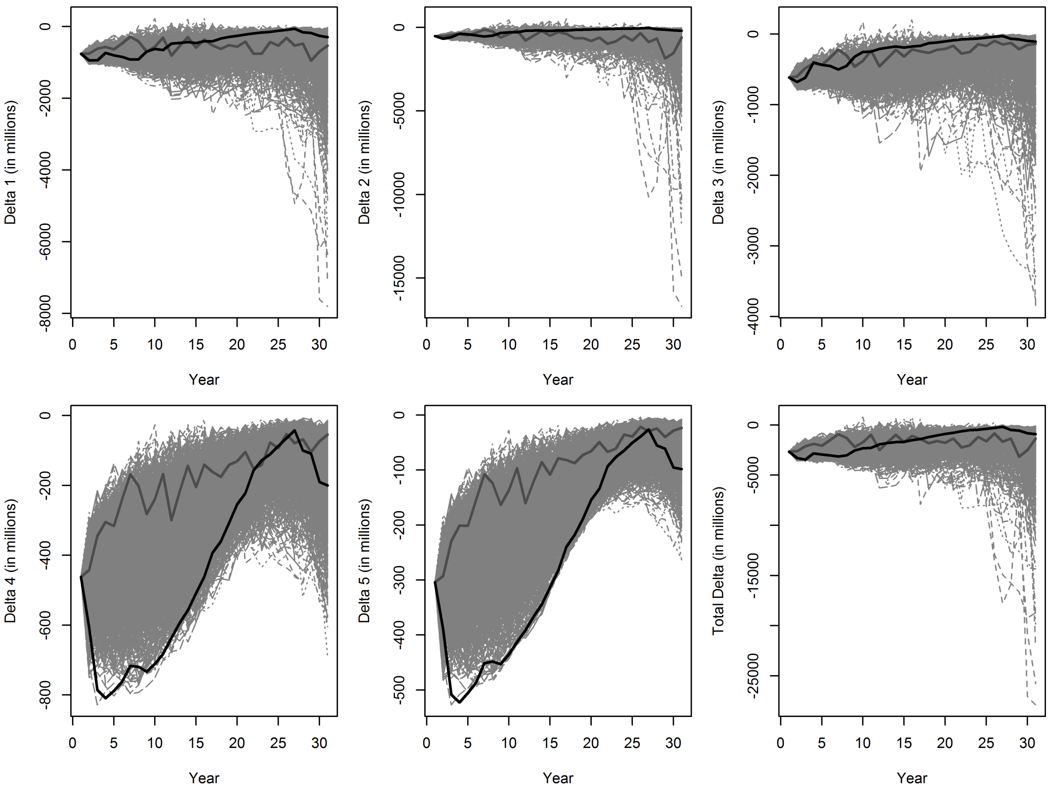

Figure 7. Aggregate partial dollar deltas and aggregate total dollar delta along the 1000 real-world paths. The dark thick and the gray thick lines correspond to the worst and the best real-world paths, respectively.

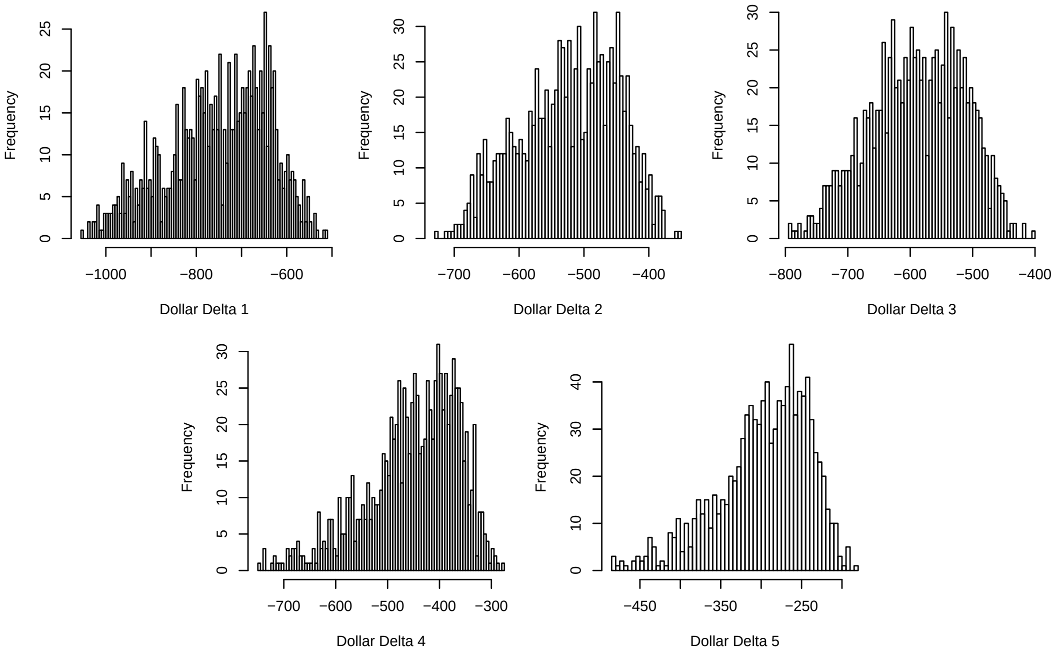

Figure 8. A histogram of the aggregate partial dollar deltas along the 1000 real-world paths at the end of Year 1. The numbers are in millions.

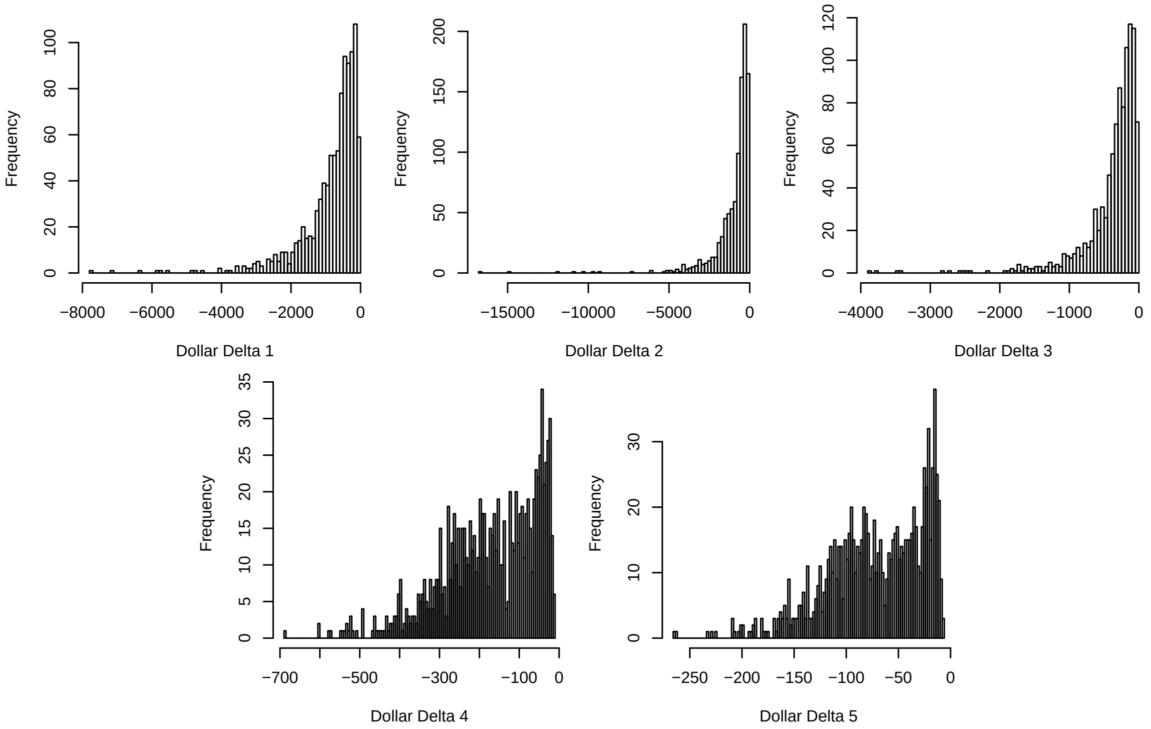

Figure 9. A histogram of the aggregate partial dollar deltas along the 1000 real-world paths at the end of Year 30. The numbers are in millions.

4.2. Seriatim Results

There are many seriatim results, making it difficult to show all the results in detail. In this section, we only show the seriatim results from the best and the worst real-world paths identified before. Figure 10 shows a histogram of the seriatim partial dollar deltas at the end of Year 1 if the best real-world path occurs. Figure 11 shows a similar histogram if the worst real-world path occurs. Both figures show that the distributions of the seriatim partial dollar deltas are highly skewed. In addition, some policies have positive dollar deltas if the best real-world path occurs. This is caused by the GMAB products as a bull market can trigger the renew option embedded in such products.

Figures 12 and 13 show the box plots of seriatim partial dollar deltas by product type at the end of Year 1 along the best and the worst real-world paths, respectively. In these figures, we see that the GMAB, GMIB, and GMMB products are more sensitive than the GMDB and GMWB products in terms of the magnitudes of the deltas. In addition, more policies have positive deltas when the best real-world path occurs than the case when the worst real-world path occurs.

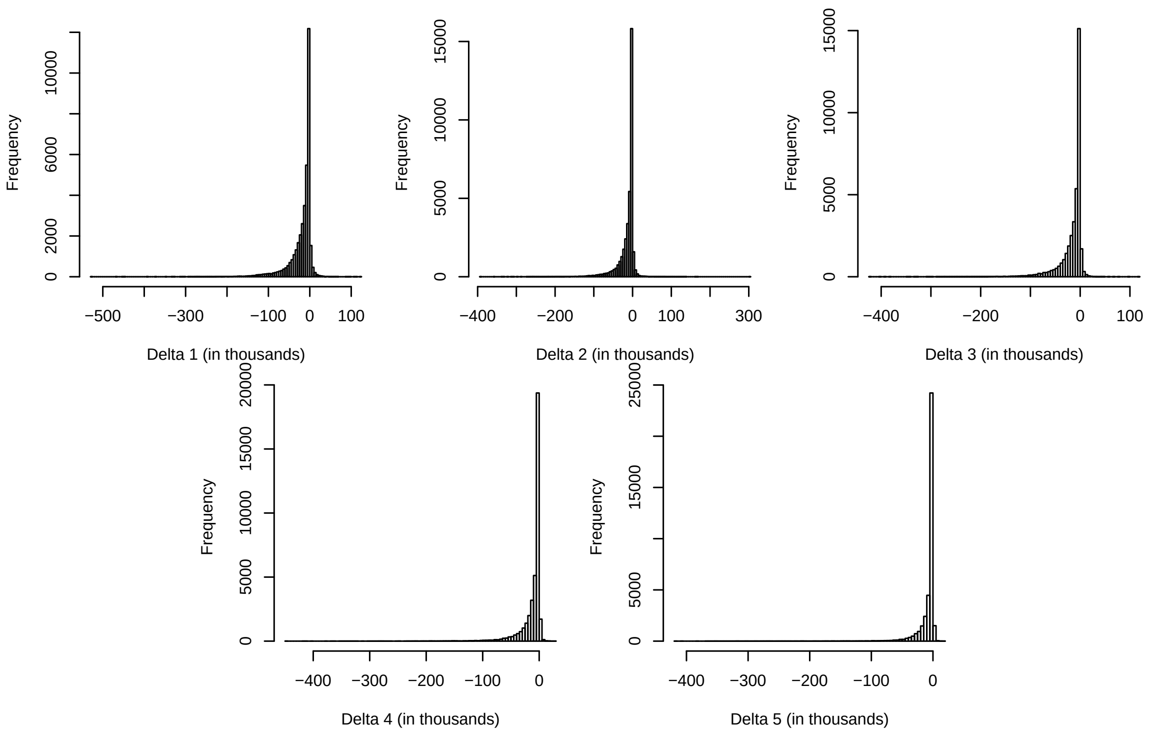

Figure 10. Histograms of the seriatim partial dollar deltas at the end of Year 1 of the best real-world path.

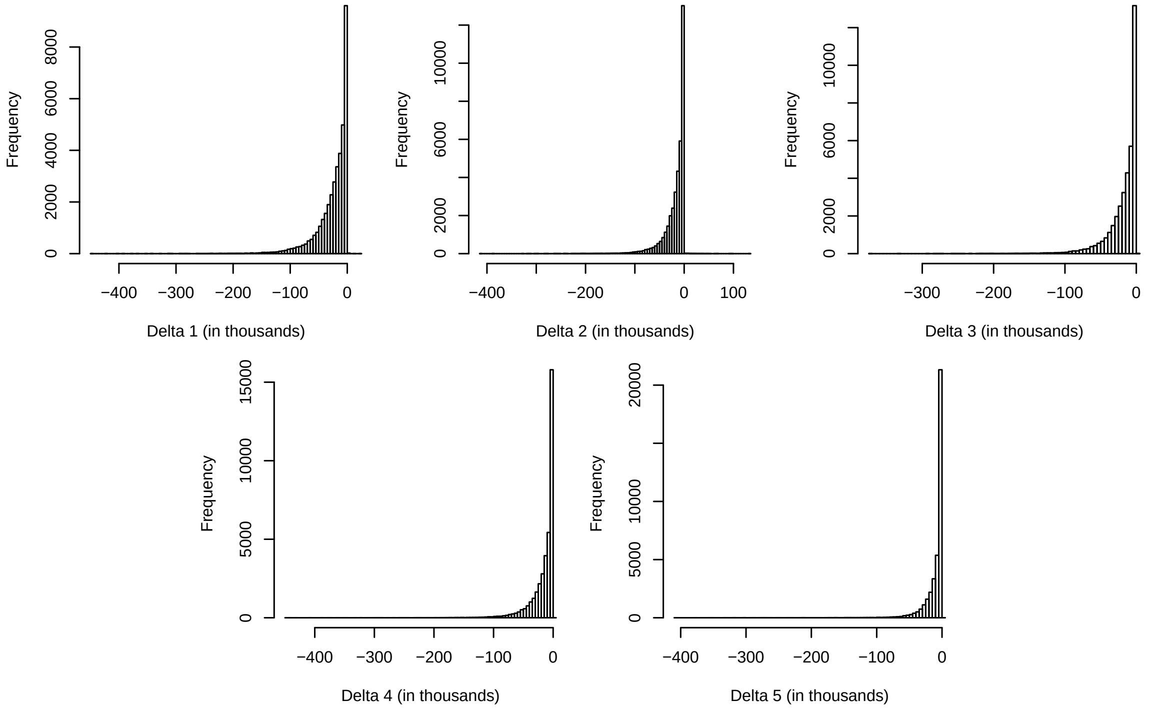

Figure 11. Histograms of the seriatim partial dollar deltas at the end of Year 1 of the worst real-world path.

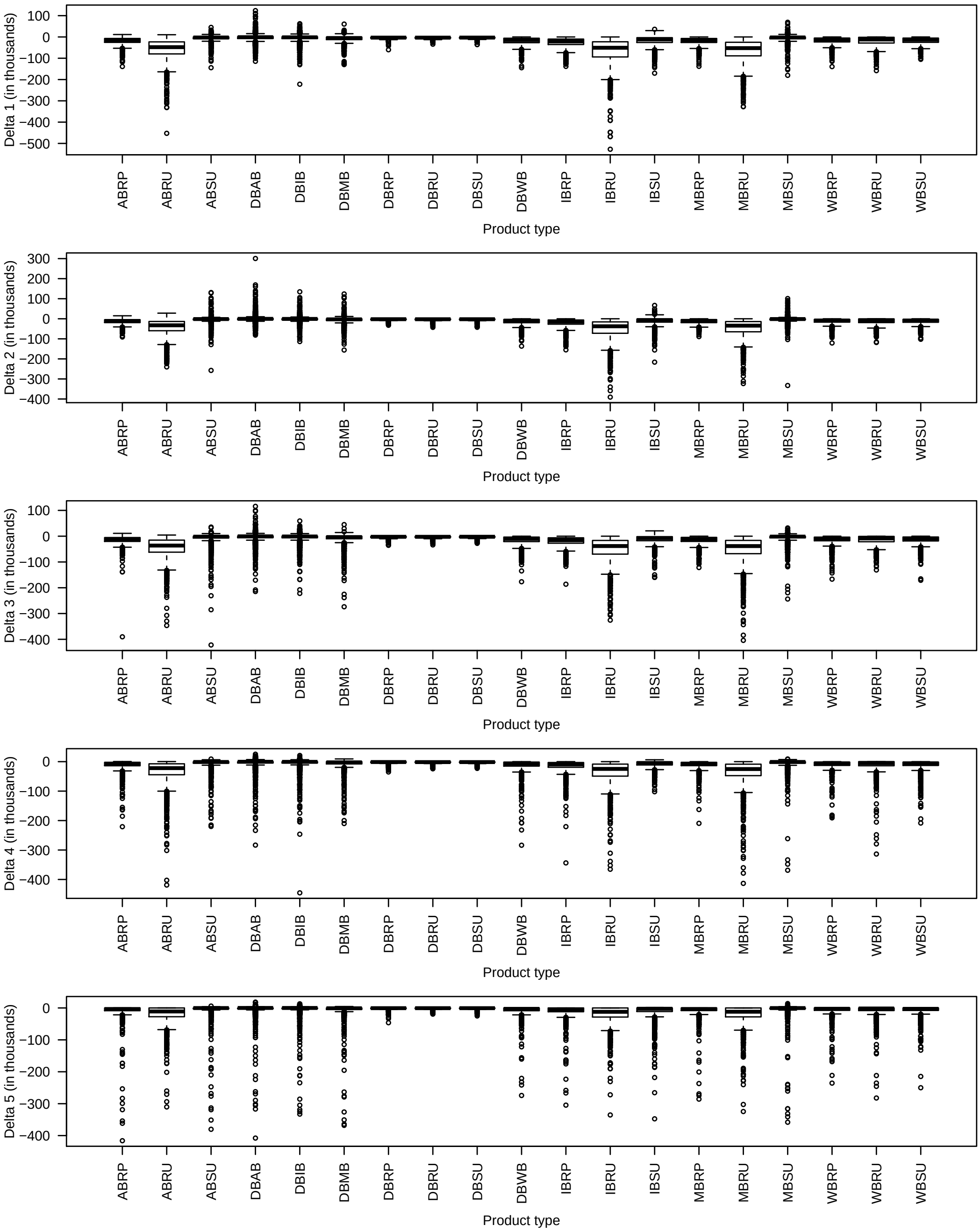

Figure 12. Box plots of the partial dollar deltas by product type at the end of Year 1 of the best real-world path.

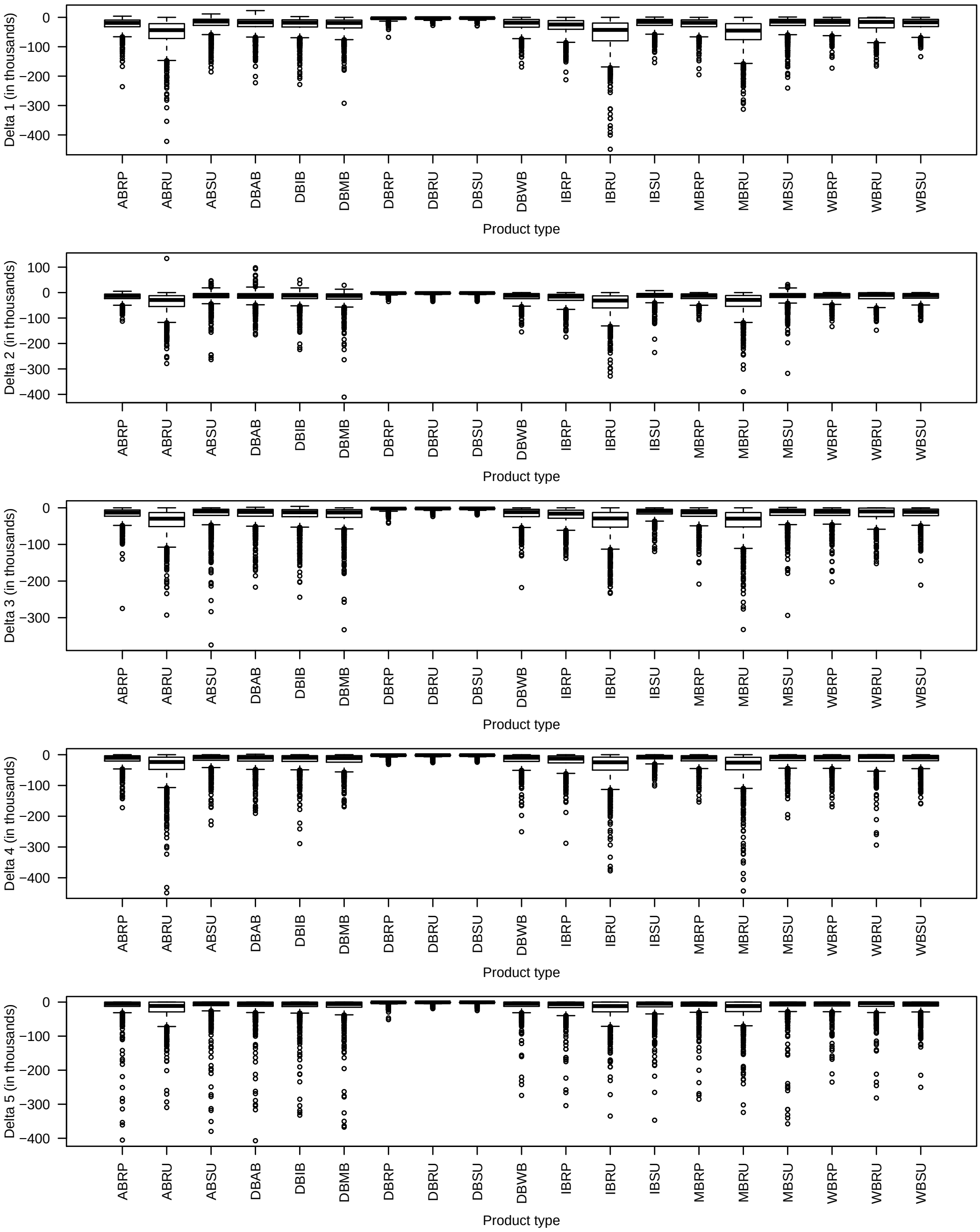

Figure 13. Box plots of the partial dollar deltas by product type at the end of Year 1 of the worst real-world path.

4.3. Runtime

We implemented the nested stochastic valuation engine as a distributed multi-threading program in Java. We used the HPC (High Performance Computing) cluster (https://hpc.uconn.edu/) at the University of Connecticut to run the program. In particular, we used eight instances of the program with 20 cores for each instance to calculate the partial dollar deltas for the portfolio. Each instance of the program handles one outer loop path at a time. The coordination between different instances is done via the mechanism of file locking. Even with 160 cores, it took about two weeks to get all the calculations done.

For the convenience of comparison, we accumulate the runtime used by all threads to get the runtime that would be used by a single core. Figure 14 shows a histogram of the runtime used to calculate the partial dollar deltas for an outer loop path. In the figure, we see that, if a single core is used, it would take the core about 20–32 h to finish the calculation for a single outer loop path. If we aggregate the runtime used to process all 1000 outer loop paths, the runtime is 93,722,002.966 s or 2.97 years. In other words, if we used a single CPU to calculate the partial dollar deltas for the portfolio of 38,000 VA policies with 1000 real-world path and 1000 risk-neutral paths, it would take this CPU about 2.97 years to finish the calculation. Note that we only calculated the partial dollar deltas at 30 time points along the outer loop paths. If we want to calculate the deltas at 360 time points along the outer loop paths, it would take a single core about 36 years.

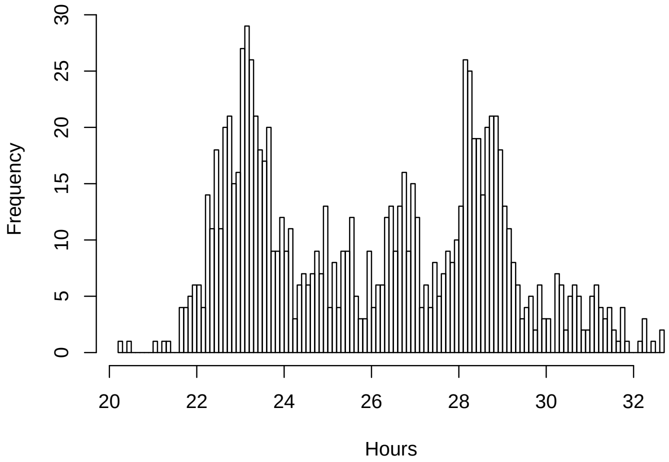

Figure 14. Distribution of the runtime for the 1000 real-world paths.

5. Concluding Remarks

Metamodeling techniques have been proposed to address the computational issues associated with the nested stochastic valuation of large VA portfolios. However, it is difficult for researchers to obtain real datasets from insurance companies to test the metamodeling techniques and publish the results in academic journals. It is the primary purpose of this paper to create synthetic datasets to address computational issues. These synthetic datasets can be used by researchers and practitioners to test techniques, especially metamodeling techniques, to speed up the nested stochastic valuation of large VA portfolios.

These synthetic datasets have some limitations. First, the synthetic VA policies are simpler than VA policies sold in the real-world. Second, the Monte Carlo simulation is also simpler than the one used in practice. For example, we did not consider the policyholder behavior in the cash flow projections. Although the synthetic datasets have limitations, we can still use them to test metamodeling techniques. If a metamodeling technique does not work for the synthetic datasets, then it is unlikely to work for real datasets.

Author Contributions: Both G.G. and E.A.V. conceived the primary idea of the methodology used. G.G. wrote the Java code to implement the methodology and create the synthetic data. In consultation with E.A.V., G.G. analyzed the data and produced the output. Both authors had periodic meetings to discuss the results and the writing of the manuscript.

Funding: This work is supported by a CAE (Centers of Actuarial Excellence) grant (http://actscidm.math.uconn.edu) from the Society of Actuaries.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Software and Datasets

We implemented the nested stochastic valuation engine as a distributed multi-threading program in Java. The datasets and the software code can be downloaded from http://www.math.uconn.edu/~gan/software.html.

References

1. Ledlie, M.C.; Corry, D.P.; Finkelstein, G.S.; Ritchie, A.J.; Su, K.; Wilson, D.C.E. Variable Annuities. *Br. Actuarial J.* **2008**, *14*, 327–389.
2. The Geneva Association Report. Variable annuities - an analysis of financial stability. Available online: https://www.genevaassociation.org/sites/default/files/research-topics-document-type/pdf_public/ga2013-variable_annuities_0.pdf. Accessed on August 30, 2018.
3. Boyle, P.; Hardy, M. Reserving for maturity guarantees: Two approaches. *Insurance Math. Econ.* **1997**, *21*, 113–127.
4. Chopra, D.; Erzan, O.; de Gantes, G.; Grepin, L.; Slawner, C. Responding to the Variable Annuity Crisis. McKinsey Working Papers on Risk. 2009. Available online: https://www.mckinsey.com/business-functions/risk/our-insights/responding-to-the-variable-annuity-crisis (Accessed on August 30, 2018.)
5. International Actuarial Association. *Stochastic Modeling: Theory and Reality from an Actuarial Perspective*; International Actuarial Association: Ontario, Canada, 2010.
6. Dardis, T. Model Efficiency in the U.S. Life Insurance Industry. *Model. Platform* **2016**, *3*, 9–16.
7. Gan, G. Application of data clustering and machine learning in variable annuity valuation. *Insurance Math. Econ.* **2013**, *53*, 795–801.
8. Gan, G.; Lin, X.S. Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance Math. Econ.* **2015**, *62*, 138 – 150.
9. Gan, G. Application of Metamodeling to the Valuation of Large Variable Annuity Portfolios. In Proceedings of the Winter Simulation Conference, Huntington Beach, CA, USA, 6–9 December 2015; pp. 1103–1114.
10. Gan, G.; Lin, X.S. Efficient Greek Calculation of Variable Annuity Portfolios for Dynamic Hedging: A Two-Level Metamodeling Approach. *N. Am. Actuarial J.* **2017**, *21*, 161–177.
11. Gan, G.; Valdez, E.A. An Empirical Comparison of Some Experimental Designs for the Valuation of Large Variable Annuity Portfolios. *Depend. Model.* **2016**, *4*, 382–400.
12. Hejazi, S.A.; Jackson, K.R. A neural network approach to efficient valuation of large portfolios of variable annuities. *Insurance Math. Econ.* **2016**, *70*, 169 – 181.
13. Gan, G.; Huang, J. A Data Mining Framework for Valuing Large Portfolios of Variable Annuities. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1467–1475.
14. Hejazi, S.A.; Jackson, K.R.; Gan, G. A Spatial Interpolation Framework for Efficient Valuation of Large Portfolios of Variable Annuities. *Quant. Financ. Econ.* **2017**, *1*, 125–144.
15. Gan, G.; Valdez, E.A. Regression Modeling for the Valuation of Large Variable Annuity Portfolios. *N. Am. Actuarial J.* **2018**, *22*, 40–54.
16. Xu, W.; Chen, Y.; Coleman, C.; Coleman, T.F. Moment matching machine learning methods for risk management of large variable annuity portfolios. *J. Econ. Dyn. Control* **2018**, *87*, 1 – 20.
17. Gan, G.; Valdez, E.A. Valuation of Large Variable Annuity Portfolios: Monte Carlo Simulation and Synthetic Datasets. *Depend. Model.* **2017**, *5*, 354–374.
18. Carmona, R.; Durrelman, V. Generalizing the Black-Scholes Formula to Multivariate Contingent Claims. *J. Comput. Financ.* **2006**, *9*, 43–67.
19. Hardy, M. A Regime-Switching Model of Long-Term Stock Returns. *N. Am. Actuarial J.* **2001**, *5*, 41–53.
20. Gan, G.; Ma, C.; Xie, H. *Measure, Probability, and Mathematical Finance: A Problem-Oriented Approach*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2014.