EFFICIENT DYNAMIC HEDGING FOR LARGE VARIABLE ANNUITY PORTFOLIOS WITH MULTIPLE UNDERLYING ASSETS

BY

X. SHELDON LIN AND SHUAI YANG

ABSTRACT

A variable annuity (VA) is an equity-linked annuity that provides investment guarantees to its policyholder and its contributions are normally invested in multiple underlying assets (e.g., mutual funds), which exposes VA liability to significant market risks. Hedging the market risks is therefore crucial in risk managing a VA portfolio as the VA guarantees are long-dated liabilities that may span decades. In order to hedge the VA liability, the issuing insurance company would need to construct a hedging portfolio consisting of the underlying assets whose positions are often determined by the liability Greeks such as partial dollar Deltas. Usually, these quantities are calculated via nested simulation approach. For insurance companies that manage large VA portfolios (e.g., 100k+ policies), calculating those quantities is extremely time-consuming or even prohibitive due to the complexity of the guarantee payoffs and the stochastic-on-stochastic nature of the nested simulation algorithm. In this paper, we extend the surrogate model-assisted nest simulation approach in Lin and Yang [(2020) *Insurance: Mathematics and Economics*, **91**, 85–103] to efficiently calculate the total VA liability and the partial dollar Deltas for large VA portfolios with multiple underlying assets. In our proposed algorithm, the nested simulation is run using small sets of selected representative policies and representative outer loops. As a result, the computing time is substantially reduced. The computational advantage of the proposed algorithm and the importance of dynamic hedging are further illustrated through a profit and loss (P&L) analysis for a large synthetic VA portfolio. Moreover, the robustness of the performance of the proposed algorithm is tested with multiple simulation runs. Numerical results show that the proposed algorithm is able to accurately approximate different quantities of interest and the performance is robust with respect to different sets of parameter inputs. Finally, we show how our approach could be extended to potentially incorporate stochastic interest rates and estimate other Greeks such as Rho.

Astin Bulletin 50(3), 913-957. doi:10.1017/asb.2020.26

KEYWORDS

Variable annuity portfolio, nested simulation, balanced sampling, spline regression, dynamic hedging.

1. INTRODUCTION

A variable annuity (VA) is an equity-linked annuity that provides investment guarantees to its policyholder. It has become one of the major products in the insurance market. According to Coleman *et al.* (2019), the total VA sales in the US in 2018 were around $100 billion. In addition, according to LIMRA Secure Retirement Institute (LIMRA SRI), the top 10 VA sellers in 2018 together accounted for 78% of the market share. Implied by these numbers are large VA portfolios managed by a few major insurance companies.

A main reason for the popularity of VA is their embedded guarantees which provide downside protection to the policyholders. The two broad categories of the guarantees are the guaranteed minimum death benefits (GMDBs) and the guaranteed minimum living benefits (GMLBs). A GMDB guarantees a minimum death benefit to the policyholder and it is typically embedded in all VA polices. GMLBs, on the other hand, can be elected as riders according to policyholders' choice. Two most offered GMLBs are the guaranteed minimum accumulation benefit (GMAB) and the guaranteed minimum withdrawal benefit (GMWB) (see Coleman *et al.*, 2019). A GMAB rider guarantees a minimum accumulation value of the policyholder's VA account at maturity, while a GMWB rider guarantees a minimum level of periodic withdrawals for the policyholder throughout the policy term. A detailed description of the guarantees can be found in, for example, the Insured Retirement Institute 2019 Fact Book. The popularity of the VA has stimulated a vast amount of research in academia on its modeling and risk management. For example, here are some of highly cited papers: Milevsky and Posner (2001), Boyle and Hardy (2003), Lin and Tan (2003), Milevsky and Salisbury (2006), Bauer *et al.* (2008), Dai *et al.* (2008), Lin *et al.* (2009), Bacinello *et al.* (2011), and Bernard *et al.* (2014).

Since the 2008 financial crisis, regulators have put more emphasis on the solvency of the insurance companies. The Solvency II regulatory framework requires insurance companies to calculate the *solvency capital requirement* (SCR) for their insurance portfolios, which is the minimum amount of capital to hold to remain solvent in a year with a probability of 99.5%. To calculate the capital requirements for a VA portfolio, one must first calculate the predictive total liability distribution at some future time point. As the guarantees of VAs may be viewed as multi-dated, path-dependent put options, the most widely used approach to calculating their predictive liability is *nested simulation*. Nested simulation is a stochastic-on-stochastic (SoS) algorithm that contains two parts: an outer-loop simulation and an inner-loop simulation. In the outer-loop simulation, the dynamics of the underlying assets are projected using multiple real-world economic scenarios (outer loops) from the

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

current time to a future time point of interest. The account value and guarantee base are then calculated for each policy along outer loops. In the inner-loop simulation, the VA liability at the future time point of each policy is calculated by simulating a large number of risk-neutral economic scenarios (inner loops). Lastly, the total VA liabilities among the simulated outer loops give the predictive total VA liability distribution. The relevant risk metrics, such as the value-at-risk (VaR) and the conditional value-at-risk (CVaR), of that distribution can then be easily determined. Bauer *et al.* (2012) provided a detailed introduction of the capital requirements calculation under the nested simulation approach.

Since most of the VA contributions are invested in the equity market, usually in multiple risky assets such as mutual funds, the VA liability are therefore exposed to the market risks which cannot be diversified. Moreover, the exposures can become large as the VA guarantees are long-dated liabilities that may span decades. Insurance companies hence would need to set aside a large amount of capital as a buffer to protect them from insolvency, limiting the amount of their available capitals for business expansion. The recent revisions in AG43/VM-21 and C3 Phase II regulations reward companies that have clearly defined hedging strategies (CDHS) by allowing them to hold a smaller amount of capital (CDHS credit). As a result, insurance companies are incentivized to perform in-house dynamic hedging for their VA portfolios. See, for example, Meyricke and Sherris (2014) and Varnell *et al.* (2019) for more discussion. According to a survey conducted by Willis Towers Watson[1] in 2013, which summarized the risk management programs by the top VA sellers in the US, most of surveyed insurance companies perform in-house hedging on a regular basis to manage their liability risks. In order to dynamically hedge the total VA liability, an insurance company would need to create a hedging portfolio consisting of the underlying assets whose positions depend on Greeks and partial dollar Deltas in particular. In practice, these partial dollar Deltas are normally calculated using the 'bump and revalue' approach, which has two steps. In the first step, an asset returns among the simulated outer loops are bumped upward/downward by a small amount (usually 1%); and in the second step, the total VA liability at those shocked scenarios are calculated by rerunning the nested simulation. The partial dollar Deltas are then estimated by the differences between the total VA liabilities under different shocked scenarios.

Due to the complex structure of the nested simulation algorithm, running the full nested simulation with a large VA portfolio can be extremely time-consuming or even prohibitive when the numbers of risk assets, policies, inner loops and outer loops are large. For example, if one calculates the predictive total liability distribution at a future time point for a VA portfolio containing 100,000 policies by running a nested simulation algorithm with 1000 outer loops and 10,000 inner loops with a computing system that can perform 5×10^6 projections per second, then the total runtime will take more than 2 days. If the quantities of interest are partial dollar Deltas of different assets, then the total runtime will be increased by multiple times due to the 'bump and

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

revalue' mechanism, preventing the relevant quantities to be timely calculated. As a result, reducing the simulation runtime has become a critical issue to the insurance companies when managing large VA portfolios.

According to the Willis Towers Watson survey, most of the practitioners utilized advanced IT infrastructure to accelerate the computing speed of a nested simulation program. Parallel computing is the most commonly adopted approach. The number of CPU cores used by the surveyed participants ranges from 100 to 3000. In addition, some leading software vendors such as Aon PathWise® have incorporated high performance GPU to reduce the runtime even further. Even with the advanced hardware, however, a full nested simulation still cannot be executed in many cases due to limitations such as computer memory. These challenges all together lead to an increasingly amount of research on this topic. See Gan and Lin (2015), Hejazi and Jackson (2016), Gan and Valdez (2018), Lin and Yang (2020) and references therein.

Most of these papers focused on the portfolio liability calculation. Gan and Lin (2017) studied the calculation of the partial dollar Delta for VA portfolios. They proposed a method such that a metamodel of the partial dollar Delta is fitted using a set of possible future market levels. When the market is open, the VA manager may use the fitted model to calculate the partial Dollar Deltas in real time. The methodology proposed in Gan and Lin (2017) is able to reduce the simulation runtime from days to hours. However, there are two main issues in their study. Firstly, the number of inner loops generated in their simulation is 1000, which is too low to produce accurate estimates of the VA liabilities/partial dollar Deltas. If the number of inner loops is increased to 10,000 to reduce the estimation error, then the runtime for training the Level 1 metamodel is expected to increase 10 times. Secondly, the number of predetermined future market levels is 50, which is too low to well represent the possible future scenarios, especially when multiple underlying assets are considered. Hence the Greeks at other market levels, which are estimated from the Level 2 metamodel, may be inaccurate.

Recently in Lin and Yang (2020), we proposed a surrogate model-assisted simulation algorithm in which different statistical models are incorporated into the simulation program. The proposed algorithm is used to estimate the predictive total liability distribution for a large VA portfolio, when assuming a single underlying asset. The purpose of this paper is to extend the work in Lin and Yang (2020) in several directions. Firstly, we consider a situation where the policyholders' accounts are invested in multiple underlying assets, and we study the selection of the set of representative outer loops in this situation. Secondly, we design an algorithm to estimate not only the total VA liability distribution but also other portfolio quantities such as partial dollar Deltas. Lastly, we extend our algorithm to a multi-period setting and perform a profit and loss (P&L) projection of the dynamic Delta hedging strategy. We demonstrate the importance of the dynamic hedging in the context of VA portfolio and illustrate how our proposed method can be used to efficiently perform the P&L projection.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

The rest of the paper is organized as follows. In Section 2, we introduce an efficient nested simulation algorithm for calculating the total VA liability with multiple underlying assets. In Section 3, we first modify the proposed algorithm to calculate the portfolio partial dollar Deltas, and then discuss how to implement a dynamic hedging program to hedge total VA liabilities in a multi-period setting using the proposed algorithm. Numerical results from several simulation studies are presented in Section 4. In Section 5, we illustrate how the proposed approach can be extended to handle stochastic interest rates and to estimate other Greeks such as Rho using a multidimensional regression method named the thin plate spline regression. We conclude the paper with some remarks in Section 6.

2. Efficient calculation of total VA liability with multiple underlying assets

The algorithm proposed by Lin and Yang (2020) incorporates statistical models that act as surrogate models into the nested simulation algorithm to approximate the relationship between the simulation inputs and outputs. Two types of models are used in the proposed algorithm: the linear model and the spline regression model. The linear model is used to approximate the relationship between the policies' attributes and the VA liabilities along each outer loop. Assisted with the linear model together with population sampling theory, the number of policies to run the nested simulation is reduced. The spline regression model is policy specific, and they are used to approximate the relationship between the policy's account values and liabilities of different outer loops. Assisted with the spline regression model together with the clustering algorithm, the numbers of inner loops and outer loops are reduced.

In Lin and Yang (2020), a single underlying asset is assumed and the goal of that paper is to calculate the total liability of a large VA portfolio. In this section, we extend the algorithm to estimate the predictive total liability distribution of large VA portfolios whose policyholders' accounts are invested in multiple underlying assets. The first subsection covers the selection of representative policies using a model-assisted population sampling approach. The second subsection focuses on scenario clustering when multiple assets are considered. The last subsection introduces the spline regression model which are used to estimate the VA liabilities of the representative policies.

2.1. Selection of representative policies through population sampling

We adopt a model-assisted population sampling framework to select a set of representative policies, which are used to estimate the total VA liability. For each outer loop (real-world scenario) a linear model is employed to approximate the relationship between a policyholder's attributes at the current time

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

t_0 and the predictive VA liability at a future time point of interest t_1, (e.g., a week). Consider a portfolio with N policies whose predictive liability distribution is calculated from a nested simulation algorithm with M outer loops. Let $L_p(s)$ be the predictive VA liability of policy p at t_1 of outer loop s, where $p = 1, ..., N$ and $s = 1, ..., M$. Let $\boldsymbol{x}_{p,0}$ be the attribute vector (e.g., account value, age, gender, guarantee type, and asset allocation) of policy p at t_0. The linear model is expressed as

$$L_p(s) = \boldsymbol{b}'(s)\boldsymbol{x}_{p,0} + e_p(s), \tag{2.1}$$

where $e_p(s)$ are assumed to be i.i.d. with mean 0 and variance $\sigma_p^2(s)$. Let $\pi_p(s)$ denote the first-order inclusion probability of policy p at outer loop s, which represents the likelihood of policy p being selected as one of the representative policies. Results from Nedyalkova and Tillé (2008) state that an optimal estimation strategy is to use $\pi_p(s) = n\sigma_p(s)/\sum_{p=1}^{N}\sigma_p(s)$, $p = 1, ..., N$, to select a balanced sample in which $\sum_{p=1}^{n}\boldsymbol{x}_{p,0}^*/\pi_p^*(s) = \sum_{p=1}^{N}\boldsymbol{x}_{p,0}$, where $\boldsymbol{x}_{p,0}^*$ and $\pi_p^*(s)$, respectively, represent the attributes and inclusion probability of the selected policies. Further, one way to select a balanced sample is through the Cube algorithm (Deville and Tillé, 2004), which will be introduced in Section 2.2. Once the representative policies are selected, the total liability $\sum_{p=1}^{N}L_p(s)$ is then estimated by the linear estimator $\sum_{p=1}^{n}L_p^*(s)/\pi_i^*(s)$, where n is the sample size with $n \ll N$. $L_p^*(s)$ and $\pi_p^*(s)$, $p = 1, ..., n$, respectively, are the VA liability and the inclusion probability of the representative policies.

Since the variances $\sigma_p^2(s)$, $p = 1, ..., N$, $s = 1, ..., M$, are not given as *a priori* information, we proposed a two-stage procedure to select the set of representative policies. In order to make the set of representative policies outer-loop independent, we assumed $\sigma_p(s)$ takes a multiplicative form $\sigma_p(s) = \gamma(s)h(\boldsymbol{x}_{p,0})$. As a result, the inclusion probability of policy p is $\pi_p(s) = nh(\boldsymbol{x}_{p,0})/\sum_{p=1}^{N}h(\boldsymbol{x}_{p,0})$. In the first stage, we assume $h(\boldsymbol{x}_{p,0})$ is a constant and independent of $\boldsymbol{x}_{p,0}$ so that $\pi_p(s) = n/N$ and select a set of policies using the Cube algorithm. In the second stage, $h(\boldsymbol{x}_{p,0})$ is identified through residual diagnostic using the liabilities estimated in the first stage. A new set of policies is selected again using the Cube algorithm with the updated inclusion probabilities, $\pi_p = nh(\boldsymbol{x}_{p,0})/\sum_{p=1}^{N}h(\boldsymbol{x}_{p,0})$, $p = 1, ..., N$. The resulting policies from this selection are the representative policies whose liabilities will be used to estimate the total VA liability. We remark here that the second stage is an improvement stage: the sets of policies selected from the first stage can already approximate the population quantities with high accuracy in many cases.

2.2. The Cube algorithm

The Cube algorithm, which is proposed by Deville and Tillé (2004), selects a nearly balanced random sample for a given first-order inclusion probability

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

through an iterative procedure. The algorithm contains two phases: a flight phase and a landing phase.

Let N be the population size and each population unit is r-dimensional. In our context, r is the dimension of the attribute vector. The flight phase iteratively translates the inclusion probabilities to a vector of at least $(N-r)$ zeros or ones. The balanced condition $\sum_{k\in S} \boldsymbol{x}_k/\pi_k = \sum_{k\in U} \boldsymbol{x}_k$ can be written in the matrix form:

$$A\boldsymbol{S} = A\boldsymbol{\pi}, \tag{2.2}$$

where $A = (\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N)$ a $r \times N$ matrix and $\boldsymbol{a}_k = \boldsymbol{x}_k/\pi_k$ for $k = 1, ..., N$, and $\boldsymbol{S} = (\mathbb{1}_1, \mathbb{1}_2, ..., \mathbb{1}_N)^t$ a random column vector, where $\mathbb{1}_k = 1$ or 0 indicating the inclusion of the kth unit. Equation (2.2) implies that all balanced samples form a subspace of $\mathbb{R}^N$ with dimension $N-r$. Hence $\boldsymbol{S}$ can be written as $\boldsymbol{\pi} + \boldsymbol{u}$, where $\boldsymbol{u}$ is in the kernel of matrix A, that is, $A\boldsymbol{u} = \boldsymbol{0}$. From this, in each iteration of the flight phase the inclusion probability vector is positioned randomly inside the kernel space of A until it reaches to a point that is close to a vertex of the N-dimensional hypercube.

There are three steps in each iteration in the flight phase. For a given inclusion probability vector $\boldsymbol{\pi}$, set $\boldsymbol{\pi}(1) = \boldsymbol{\pi}$, at iteration $i = 1, 2, ..., I$:

- Step 1: Randomly generate a vector $\boldsymbol{u}(i)$ in the kernel of matrix $A = (\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N)$. Set $\boldsymbol{u}_k(i) = 0$ if $\boldsymbol{\pi}_k(i) = 0$ or 1.
- Step 2: Compute $\lambda_1^*(i)$ and $\lambda_2^*(i)$, the largest values among $\lambda_1(i)$ and $\lambda_2(i)$ such that

$$0 \le \boldsymbol{\pi}(i) + \lambda_1(i)\boldsymbol{u}(i) \le 1,$$
$$0 \le \boldsymbol{\pi}(i) - \lambda_2(i)\boldsymbol{u}(i) \le 1.$$

- Step 3: Compute $\boldsymbol{\pi}(i+1)$ as follows:

$$\boldsymbol{\pi}(i+1) = \boldsymbol{\pi}(i) + \lambda_1^*(i)\boldsymbol{u}(i) \text{ with probability } \frac{\lambda_2^*(i)}{\lambda_1^*(i) + \lambda_2^*(i)},$$
$$\boldsymbol{\pi}(i+1) = \boldsymbol{\pi}(i) - \lambda_2^*(i)\boldsymbol{u}(i) \text{ with probability } \frac{\lambda_1^*(i)}{\lambda_1^*(i) + \lambda_2^*(i)}.$$

The above three steps iterate until $\boldsymbol{\pi}(i)$ stops changing. In the landing phase, each non-integer element resulting from the flight phase is adjusted to either zero or one by linear programming. The resulting vector with only zeros and ones gives a nearly balanced sample.

2.3. Selection of representative outer loops through scenario clustering

To further reduce the simulation runtime, Lin and Yang (2020) proposed a clustering-based method to select a subset of outer loops which are referred to as representative outer loops. By doing this, the nested simulation is run with

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

not only a fewer number of policies but also a fewer number of outer loops. The VA liability of a selected policy at the non-selected outer loops is then estimated through a spline regression model which will be introduced in Section 2.4.

As mentioned in the Introduction section, Lin and Yang (2020) considered a single underlying asset. In this paper, we extend the method for selecting representative outer loops to a more realistic setting where the VA accounts are invested into multiple underlying assets. In the following, we introduce scenario clustering under the multi-asset setting.

2.3.1. *The k-means clustering*

Let $\boldsymbol{\chi} = (\boldsymbol{\chi}_1, ..., \boldsymbol{\chi}_n)$ denote a data set, where $\boldsymbol{\chi}_i \in \mathbb{R}^d$, $i = 1, ..., n$, and let $\mathcal{C}_k = \{\mathcal{C}_1, ..., \mathcal{C}_k\} \subset \mathbb{R}^d$ be a partition of $\boldsymbol{\chi}$. For a given number of clusters k, the within-cluster sum of squares (WCSS) of $\boldsymbol{\chi}$ is defined as

$$\Phi(\boldsymbol{\chi}, \mathcal{C}_k) := \sum_{j=1}^{k} \sum_{i \in C_j} ||\boldsymbol{\chi}_i - \boldsymbol{\mu}_j||^2,$$

where $\boldsymbol{\mu}_j$ is the average of $\boldsymbol{\chi}_i$ for $i \in \mathcal{C}_j$ and $||\cdot||$ is the Euclidean norm. The solution to the k-means clustering is the partition $\mathcal{C}_k^* = \{\mathcal{C}_1^*, ..., \mathcal{C}_k^*\}$ which minimizes the WCSS, that is,

$$\mathcal{C}_k^* = \text{argmin}_{\mathcal{C}_k = \{\mathcal{C}_1, ..., \mathcal{C}_k\}} \Phi(\boldsymbol{\chi}, \mathcal{C}_k). \tag{2.3}$$

In theory finding the optimal solution $\mathcal{C}_k^*$ is NP-hard even for $k = 2$ (see Aloise *et al.*, 2009). Because of this, the solution to the k-means clustering is usually obtained through a greedy iterative algorithm called the *Lloyd algorithm* (Lloyd, 1982) or more widely known as the *k-means algorithm*. The algorithm starts with a set of randomly initialized cluster centers: $\boldsymbol{\mu}^{(0)} = (\boldsymbol{\mu}_1^{(0)}, ..., \boldsymbol{\mu}_k^{(0)})$, and repeats the following two steps at iterations $1, 2, ...$ until there is no further change to $\Phi(\boldsymbol{\chi}, \mathcal{C}_k)$:

- Assignment step: Assign each data point to a cluster whose center is the closest in terms of the squared Euclidean norm, that is,

$$\mathcal{C}_j^{(t)} = \{\boldsymbol{\chi}_i : ||\boldsymbol{\chi}_i - \boldsymbol{\mu}_j^{(t-1)}||^2 \le ||\boldsymbol{\chi}_i - \boldsymbol{\mu}_l^{(t-1)}||^2, \quad l = 1, ..., k\}.$$

- Update step: Recalculate the cluster centers by

$$\boldsymbol{\mu}_j^{(t)} = \frac{\sum_{i \in \mathcal{C}_j^{(t)}} \boldsymbol{\chi}_i}{\textbf{card}(\mathcal{C}_j^{(t)})}.$$

The iterations will converge in a finite number of steps (see Arthur and Vassilvitskii, 2006). However, due to the random initialization, the clustering

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

resulted from the k-means algorithm is a local optimum. Because of this, a common practice is to run the k-means algorithm multiple times and choose the clustering that gives the lowest WCSS.

We remark a property of the WCSS function which will be used later on: $\Phi(\boldsymbol{\chi}, \mathcal{C}_k)$ is a decreasing function in k for $k = 1, ..., n$. Intuitively as the number of cluster increases, the degree of homogeneity among the data in each cluster increases and the within-cluster variation becomes smaller. The proof of this property can be found in many references, for example, Rebagliati (2013). In the next subsection, we apply the k-means algorithm to scenario clustering.

2.3.2. *Multidimensional scenario clustering using k-means*

When multiple assets are considered, each outer loop is represented by a vector containing the simulated returns of the underlying assets. Let $\boldsymbol{R}(s) = (R_1(s), ..., R_d(s))^t$ be a return vector with $R_k(s)$ representing the kth asset return from t_0 to t_1 at the sth outer loop, $k = 1, ..., d$ and $s = 1, ..., M$, and $\boldsymbol{\omega}_p = (\omega_{p,1}, ..., \omega_{p,d})^t$ be a vector containing the asset allocation of policy p. We assume that the policyholders are not allowed to short sell any underlying asset, that is, $\boldsymbol{\omega}_p \in [0, 1]^d$, as it is always the case. The total return of policy p's account at all M outer loops is given by the vector $\boldsymbol{\omega}_p^t \boldsymbol{R} = (\boldsymbol{\omega}_p^t \boldsymbol{R}(1), ..., \boldsymbol{\omega}_p^t \boldsymbol{R}(M))$. Let $\boldsymbol{\mathcal{P}}_m = \{\mathcal{P}_1, ..., \mathcal{P}_m\} \subset \mathbb{R}$ be a partition of the set $\boldsymbol{\omega}_p^t \boldsymbol{R}$. In order to reduce the number of outer loops, one needs to find a partition $\boldsymbol{\mathcal{P}}_m^* = \{\mathcal{P}_1^*, ..., \mathcal{P}_m^*\}$ that minimizes WCSS:

$$\Phi(\boldsymbol{\omega}_p^t \boldsymbol{R}, \boldsymbol{\mathcal{P}}_m) := \sum_{j=1}^{m} \sum_{s=1}^{M_j} (\boldsymbol{\omega}_p^t \boldsymbol{R}(s) - \boldsymbol{\omega}_p^t \boldsymbol{\mu}(j))^2, \tag{2.4}$$

where $M_j = \mathbf{card}(\mathcal{P}_j)$ and $\boldsymbol{\mu}(j)$ represents the mean vector of $\boldsymbol{R}(s) \in \mathcal{P}_j$, $j = 1, ..., m$. Notice that the objective function (2.4) depends on the asset allocation vector $\boldsymbol{\omega}_p$ implying that clustering the total return in the multiple asset case is policy specific. To overcome this shortcoming, we modify the clustering method such that the clustering is policy independent. Define $\boldsymbol{\mathcal{Q}}_m = \{\mathcal{Q}_1, ..., \mathcal{Q}_m\}$ to be a partition of $(\boldsymbol{R}(1), ..., \boldsymbol{R}(M))$. The optimal partition, in terms of k-means clustering, is a partition $\boldsymbol{\mathcal{Q}}_m^* = \{\mathcal{Q}_1^*, ..., \mathcal{Q}_m^*\}$ which minimizes the following WCSS:

$$\Phi(\boldsymbol{R}, \boldsymbol{\mathcal{Q}}_m) := \sum_{j=1}^{m} \sum_{s=1}^{N_j} ||\boldsymbol{R}(s) - \boldsymbol{\mu}(j)||^2, \tag{2.5}$$

where $N_j = \mathbf{card}(\mathcal{Q}_j)$ and $\boldsymbol{\mu}(j)$ represents the mean vector of $\boldsymbol{R}(s) \in \mathcal{Q}_j$.

Proposition 2.1. *Let* $\boldsymbol{\mathcal{Q}}_m = \{\mathcal{Q}_1, ..., \mathcal{Q}_m\} \subset \mathbb{R}^d$ *be a partition of* $(\boldsymbol{R}(1), ..., \boldsymbol{R}(M))$. *For any* $\boldsymbol{\omega}_p \in [0, 1]^d$ *and* $\sum_{k=1}^{d} \omega_{p,k} = 1$, $\Phi(\boldsymbol{\omega}_p^t \boldsymbol{R}, \boldsymbol{\mathcal{Q}}_m) \leq ||\boldsymbol{\omega}_p||^2 \Phi(\boldsymbol{R}, \boldsymbol{\mathcal{Q}}_m) \leq \Phi(\boldsymbol{R}, \boldsymbol{\mathcal{Q}}_m)$.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

Proof. Let $N_j = \mathbf{card}(\mathcal{Q}_j)$, and $\boldsymbol{\mu}(j)$ be the average of $\boldsymbol{R}(s) \in \mathcal{Q}_j$ for $j = 1, ..., m$. Applying the Cauchy–Schwarz inequality to (2.4), we find that

$$\sum_{j=1}^{m}\sum_{s=1}^{N_j}(\boldsymbol{\omega}_p^t\boldsymbol{R}(s) - \boldsymbol{\omega}_p^t\boldsymbol{\mu}(j))^2 \le ||\boldsymbol{\omega}_p||^2 \sum_{j=1}^{m}\sum_{s=1}^{N_j}||\boldsymbol{R}(s) - \boldsymbol{\mu}(j)||^2$$
$$\le \sum_{j=1}^{m}\sum_{s=1}^{N_j}||\boldsymbol{R}(s) - \boldsymbol{\mu}(j)||^2. \tag{2.6}$$

The second inequality is followed from the fact that $||\boldsymbol{\omega}_p||^2 \le 1$ when $\boldsymbol{\omega}_p \in [0, 1]^d$ and $\sum_{k=1}^{d}\omega_{p,k} = 1$. □

Proposition 2.1 implies that if we run the k-means algorithm with $(\boldsymbol{R}(1), ..., \boldsymbol{R}(M))$ and use the resulting partition $\mathcal{Q}_m^*$ to cluster $(\boldsymbol{\omega}_p^t\boldsymbol{R}(1), ..., \boldsymbol{\omega}_p^t\boldsymbol{R}(M))$, then the WCSS $\Phi(\boldsymbol{\omega}_p^t\boldsymbol{R}, \mathcal{Q}_m^*)$ will be smaller than $\Phi(\boldsymbol{R}, \mathcal{Q}_m^*)$. In addition, the property mentioned at the end of Section 2.3.1 indicates that the difference between these two WCSS is decreasing in m, the number of clusters. Hence, when m is relatively large the optimal partition of $(\boldsymbol{R}(1), ..., \boldsymbol{R}(M))$ would also provide a good partition for $(\boldsymbol{\omega}_p^t\boldsymbol{R}(1), ..., \boldsymbol{\omega}_p^t\boldsymbol{R}(M))$. This justifies the policy-independent selection of representative outer loops, in which the policy-specific asset allocations are not considered and the k-means algorithm is run only with the simulated assets returns.

2.4. A spline regression approach to calculating VA liabilities

After running the reduced nested simulation, we obtain m pairs of predictive account values and VA liabilities for each selected policy. In order to estimate the VA liabilities of the selected policies at all outer loops, we used the spline regression as a surrogate model to approximate the relationship between the VA liabilities and the predictive account values. The main advantage of the spline regression is its ability in capturing a wide range of nonlinear relationships. The spline regression and similar approaches such as the least-squares Monte Carlo (LSMC) method have been applied in various applications to reduce their computational burden. For example, Hong *et al.* (2017) applied a kernel smoothing approach for managing portfolio risks; Bauer and Ha (2015) and Krah *et al.* (2018) applied the LSMC method to estimate the SCR under the Solvency II framework; and Duong (2019) more recently applied the spline regression model to estimate the SCR for life insurance companies. Although our approach has some similarities to some of the existing approaches, we focus on the integrated use of the spline regression model for the selection of representative outer loops, which is justified by the statistical properties of the penalized spline estimator.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

Denote $L_p(s)$ and $A_p(s)$, respectively, as representative policy p's VA liability and account value at the future time point of interest at a generic outer loop s. We assume the following spine regression model:

$$L_p(s) = \sum_{g=1}^{G} \beta_{p,g} \psi_g(A_p(s)) + \epsilon_p(s), \tag{2.7}$$

where $\epsilon_p(s)$'s are i.i.d. with mean 0 and variance ν_p^2, and $\psi_g(\cdot)$, $g = 1, ..., G$ are a set of B-splines (see De Boor, 1978). For each representative policy, the spline model (2.7) is fitted through the penalized least-square approach using the m pairs of predictive account values and VA liabilities. The estimated parameters are

$$\boldsymbol{\beta}_p^* = \text{argmin}_{\boldsymbol{\beta}_p=(\beta_{p,1},...,\beta_{p,G})} \sum_{j \in \boldsymbol{s}^*} \left(L_p(j) - \sum_{g=1}^{G} \beta_{p,g} \psi_g(A_p(j)) \right)^2 + \lambda_p \int_{\mathcal{R}} \left(\left(\sum_{g=1}^{G} \beta_{p,g} \psi_g(x) \right)^{(q)} \right)^2 dx,$$

where $\boldsymbol{s}^*$ denotes the set of the representative outer loops, and λ_p is a smoothing/tuning parameter which is normally determined by cross-validation. As a result, the approximated predictive liability of representative policy p at an outer loop s is $\hat{L}_p(s) = \sum_{g=1}^{G} \beta_{p,g}^* \psi_g(A_p(s))$.

3. Efficient Calculation of Partial Dollar Deltas and Dynamic Hedging

In order to hedge the total VA liability, the insurance company needs to construct a hedging portfolio which consists of the underlying assets. The allocation of the underlying assets are determined by their Greeks and in particular partial dollar Deltas of the total VA liability. If the hedging is performed dynamically, then the insurance company would need to calculate the partial dollar Deltas on a regular basis (e.g., daily or weekly) to rebalance the hedging portfolio. In this section, we state how the proposed algorithm in Section 2 can be used to efficiently estimate the partial dollar Deltas for large VA portfolios.

We divide this section into three subsections. In the first subsection, we use the spline regression model to estimate partial dollar Deltas for individual VAs; in the second subsection, the population sampling framework is used to estimate partial dollar Deltas for large VA portfolios; and in the last subsection, we integrate the proposed algorithm to estimate the total liability and the partial dollar Deltas of the portfolio and to perform a P&L analysis over multiple periods of time.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

3.1. Calculating partial dollar Deltas for individual VAs using regression spline

Again, we denote t_1 the future time point of interest, say, a week. We assume a policyholder whose VA account is invested in d underlying assets. The values of those underlying assets at t_1 and outer loop s are denoted by $I_1(s)$, ..., $I_d(s)$. We use $L_p(s)$ to denote the policy p's VA liability at time t_1 and outer loop s, and $\Delta_{p,i}(s)$ to denote the partial dollar Delta of policy p's liability with respect to asset i for $i = 1, ..., d$ at outer loop s. Thus, we have

$$\Delta_{p,i}(s) = I_i(s) \times \frac{\partial L_p(s)}{\partial I_i(s)}.$$

As mentioned in the Introduction section, in practice the partial dollar Delta is normally calculated by the 'bump and revalue' method. Suppose that we want to find the partial dollar Delta with respect to asset i. First, we 'bump' up and down the asset price $I_i(s)$ by a small amount, for example, 1% of $I_i(s)$, and keep the other asset prices unchanged. Next, the policy's VA liability will be recalculated by running multiple inner loops at the 'bumped' asset prices. By denoting the recalculated VA liabilities as $L_p^{i+}(s)$ and $L_p^{i-}(s)$, respectively, the partial dollar Delta with respect to asset i is estimated as

$$\Delta_{p,i}(s) \approx \frac{L_p^{i+}(s) - L_p^{i-}(s)}{2\%}.$$

Since the liabilities are recalculated through simulation, the total runtime of finding a partial dollar Delta is roughly two times longer than that of calculating the predictive total liability distribution. If the partial dollar Delta is calculated for multiple assets, then the runtime will be significantly longer. Moreover, the accuracy of the partial dollar Delta estimates is extremely sensitive to the accuracy of the liability estimates. In some cases, the partial dollar Delta estimate may be totally off even though the liabilities are fairly accurately estimated. This is due to the fact that the difference between the liabilities is usually small with the 1% change in the underlying asset price, and it is easily dominated by the estimation error resulted from the simulation.

Here, we propose an alternative method based on the regression spline model to calculate the partial dollar Deltas. Recall that $A_p(s)$ denotes the policy p's predictive account value at t_1. Let $q_{p,i}$ denote the number of units that policyholder p invests in asset i, $i = 1, ..., d$. Then $A_p(s) = q_{p,1}I_1(s) + \cdots + q_{p,d}I_d(s)$. In Section 2.4, a spline regression model is fitted to approximate the VA liability in which $\hat{L}_p(s) = \sum_{g=1}^{G} \beta_{p,g}^* \psi_g(A_p(s))$. With this formula, the partial dollar Delta can be estimated by

$$\hat{\Delta}_{p,i}(s) = I_i(s) \times \frac{d\sum_{g=1}^{G} \beta_{p,g}^* \psi_g(A_p(s))}{dA_p(s)} \times \frac{\partial A_p(s)}{\partial I_i(s)} = q_{p,i} I_i(s) \times \sum_{g=1}^{G} \beta_{p,g}^* \frac{d\psi_g(A_p(s))}{dA_p(s)}.$$

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

Since each of the basis functions $\psi_g(\cdot)$, $g = 1, ..., G$, is of an analytical form, in theory the derivatives $\frac{d\psi_g(A_p(s))}{dA_p(s)}$, $g = 1, ..., G$, can be, respectively, calculated either analytically or using finite difference, once the spline model is fitted. The latter method is often preferred in nonparametric modeling (see https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/mgcv-FAQ.html for the discussion by the author of the `mgcv` `R` package). In our case, the derivative is calculated as $\frac{d\psi_g(A_p(s))}{dA_p(s)} \approx \left(\frac{\psi_g(A_p(s)+\epsilon)-\psi_g(A_p(s)-\epsilon)}{2\epsilon}\right)$, $g = 1, ..., G$, with $\epsilon = 10^{-5}$. With this method, the partial dollar Deltas are calculated based on function derivatives and hence they are less sensitive to the estimation errors. Moreover, due to the use of the regression spline rerunning, the nested simulation at shocked scenarios for the entire VA portfolio is completely avoided. Lastly, the derivative estimator will asymptotically converge to the true derivative as the number of training points increases (see Theorem 2.2 of Lin and Yang, 2020).

We remark that there are other widely used Monte Carlo approaches for calculating Greeks such as the pathwise (PW) estimation approach and the likelihood ratio (LR) estimation approach (see Glasserman, 2013). In the context of VA, Cathcart *et al.* (2015) studied the performance of those approaches in estimating the Greeks for VA with a GMWB rider. Even though the estimation errors of the PW and the LR approaches are shown to be smaller than those of the 'bump and revalue' method, significant efforts are needed to derive the expressions for different Greek estimators. In most cases, those expressions are complicated due to the path-dependency nature of the guarantees. In addition, a large number of simulation is needed in addition to the liability calculation in order to achieve a satisfactory convergence rate. Hence, to the best of our knowledge, the PW and LR methods or their hybrid forms are not readily applicable to very large nonhomogeneous VA portfolios from a computational perspective.

3.2. Partial dollar Deltas of VA portfolios

Section 3.1 shows how the spline regression can be used to estimate the partial dollar Deltas for a single VA policy. When the quantities of interest are the partial dollar Deltas of a large VA portfolio, it becomes necessary to also reduce the number of policies run for the nested simulation. Let $\Delta_i(s)$ be the partial dollar Delta of the VA portfolio with respect to asset i, $i = 1, ..., d$, at t_1 and outer loop s. Due to linearity, the portfolio partial dollar Deltas can be written as the summation of individual partial dollar Deltas:

$$\Delta_i(s) = I_i(s) \times \frac{\partial \sum_{p=1}^{N} L_p(s)}{\partial I_i(s)}.$$

Hence if the total VA liability estimated by a subset of policies are fairly accurate for any outer loop s, then the partial dollar Deltas estimated by the same set of policies are also expected to be accurate. This implies that the same

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

set of representative policies can be used to estimate both the total liability and the partial dollar Deltas of a large VA portfolio.

After the partial dollar Delta in asset i at t_1 and outer loop s are estimated for each of the representative policies, the partial dollar Delta of the VA portfolio can be estimated by the Horvitz–Thompson estimator:

$$\hat{\Delta}_i(s) = \sum_{p=1}^{n} \frac{\hat{\Delta}_{p,i}(s)}{\pi_p^*},$$

where π_p^*, $p = 1, ..., n$, are the inclusion probabilities of the representative policies. We remark that under this method the portfolio partial dollar Deltas can be estimated in real time for any observed market condition using the fitted spline models of the representative policies.

3.3. P&L analysis over multiple periods

The algorithms proposed in previous sections for estimating the total liability and the partial dollar Deltas reduce the simulation time significantly, which will be shown in Section 4. In this section, we first discuss how the proposed algorithms can be integrated to efficiently estimate the total VA liability and partial dollar Deltas over multiple periods of time. When the total VA liability is dynamically hedged, the degree of uncertainty of the terminal predictive distribution of the total VA liability is reduced. As a result, the insurance company's required capital may be significantly reduced and in turn its profitability increases. The amount of required capital when dynamically hedging is implemented may be obtained through a P&L analysis over multiple future scenarios and multiple periods. In the next, we provide a general framework of the P&L analysis in the context of dynamically hedging the VA liability.

3.3.1. *Efficient nested simulation over multiple periods*

Figure 1 illustrates the flow of the full nested simulation algorithm over multiple periods of time. Again, we denote t_0 the initial time and s_0 the initial market information. Starting from t_0, a large number of outer loops are generated from time t_0 to t_T. At each time step along each outer loop, another large number of inner loops are simulated to calculate the total VA liability at that time step. If the partial dollar Deltas are calculated using the 'bump and revalue' approach, then the total VA liability would need to be calculated at the shocked scenarios by simulating another large numbers of inner loops again. Due to the structure of the nested simulation algorithm calculating all the relevant quantities is therefore extremely time-consuming. However, by using the proposed algorithm it is possible to run the nested simulation at a very reasonable computing cost.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

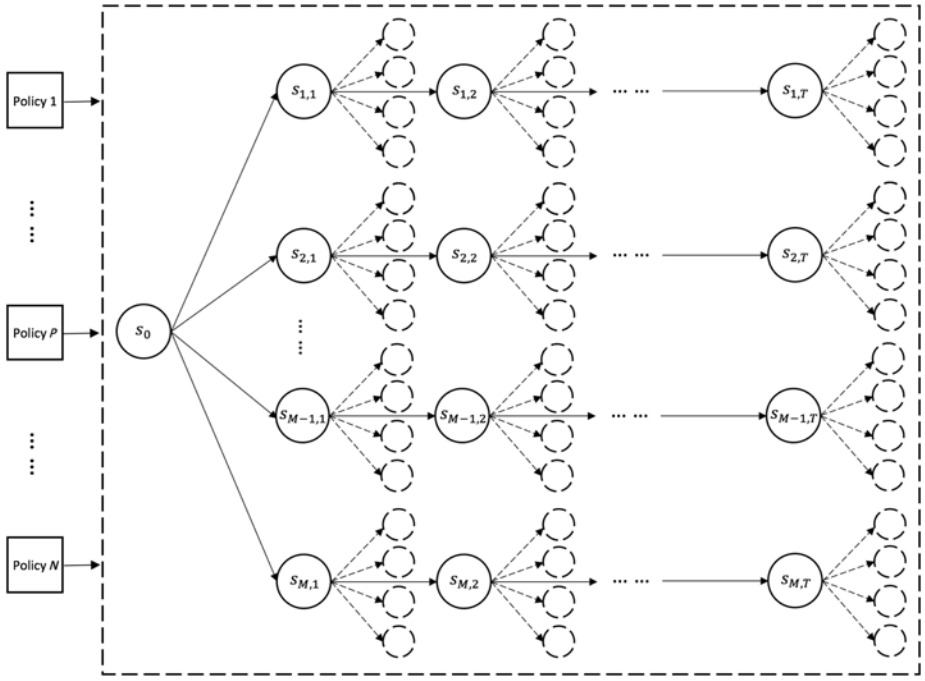

FIGURE 1: Nested simulation over multiple periods.

In order to reduce the simulation runtime to make the calculation feasible, we break the entire simulation algorithm into T pieces, where each piece is a single-period nested simulation running from time t_0 to t_y, $y = 1, ..., T$. The proposed algorithm with reduced number of policies and outer loops can then be applied to each individual one-time period to calculate quantities of interest at each future time point. The structure of the proposed algorithm is illustrated in Figure 2, and the procedure for calculating the total VA liability and partial dollar Deltas is given in the following:

- Step 1: Select a set of representative policies $\boldsymbol{z}^* = (\boldsymbol{z}_1^*, ..., \boldsymbol{z}_n^*)$ using the Cube algorithm.
- Step 2: Generate M outer loops from t_0 to t_T in a multi-period manner using a real-world economic scenario generator (ESG).
- Step 3: For each time step t_y, $y = 1, ..., T$,
  - Select a set of representative outer loops $\boldsymbol{R}_y^* = (\boldsymbol{R}_y^*(1), ..., \boldsymbol{R}_y^*(m))$ by running the k-means algorithm with the return vectors.
  - Run the reduced nested simulation on the selected sets of representative policies and representative outer loops.
  - For each selected policy, fit a spline regression model to estimate its VA liability, $\hat{L}_{p,y}(s)$, and partial dollar Deltas, $\hat{\Delta}_{p,i,y}(s)$, of all outer loops $s = 1, ..., M$.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

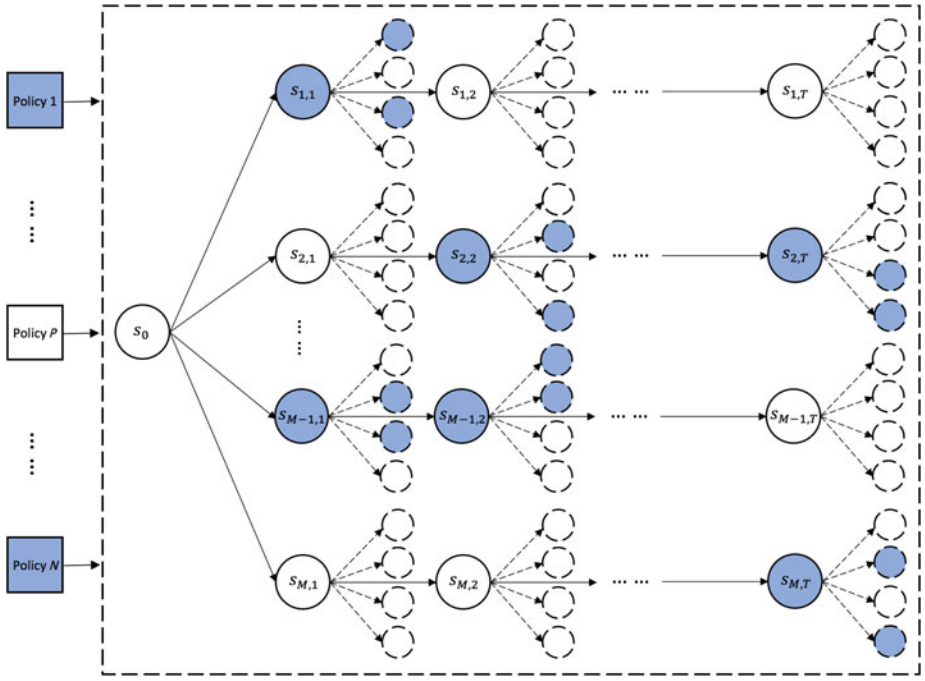

FIGURE 2: Proposed nested simulation over multiple periods.

– Estimate the total VA liability and portfolio partial dollar Deltas at time t_y of all outer loops, respectively, by

$$\frac{N}{n}\sum_{p=1}^{n}\hat{L}_{p,y}(s),\quad \frac{N}{n}\sum_{p=1}^{n}\hat{\Delta}_{p,i,y}(s).$$

We remark that the same set of representative policies is used to estimate all the quantities at all time points, but the set of representative outer loops are time dependent. To see this, consider two scenarios that are far away from each other at time t_1; the significant difference between these two scenarios will split them into two clusters. Suppose that the two scenarios become close to each other after time t_1. In this case, they will be assigned to the same cluster at t_2. Hence the clustering of the economic scenarios will be different across time, which implies the set of representative outer loops is time dependent.

3.3.2. *P&L analysis*

Let $\Delta_{i,y}(s)$ and $R_{i,y}(s)$, respectively, be the partial dollar Delta of the total VA liability with respect to asset i at time t_y and the total return of asset i from t_{y-1} to t_y along outer loop s, for $i = 1, ..., d$, $s = 1, ..., M$ and $y = 1, ..., T$. In addition, denote $B_y(s)$ the amount in the risk-free asset, and $L_y(s)$ the total VA liability at time t_y at outer loop s. At t_0, the hedging portfolio is such that the portfolio

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

value is the same as the initial total VA liability, that is, $L_0 = \sum_i \Delta_{i,0} + B_0$. For simplicity, the scenario indicator s is dropped in all of the notations at t_0.

From t_0 to t_1 (e.g., a week), the total VA liability changes from L_0 to $L_1(s)$; the amount invested in asset i changes from $\Delta_{i,0}$ to $\Delta_{i,0}R_{i,1}(s)$, $s = 1, ..., M$. Due to the change in the total liability, the amount invested in asset i needs to be rebalanced to $\Delta_{i,1}(s)$. Since the portfolio is self-financing, the rebalancing together with the interest earnings brings the value in risk-free asset to $B_1(s) = B_0 e^{r\delta_t} + \sum_i \Delta_{i,0}R_{i,1}(s) - \sum_i \Delta_{i,1}(s)$ at t_1, where r is the annualized risk-free rate compounded continuously and δ_t is the length of the time interval measured in year (e.g., $\delta_t = 1/52$ representing a week).

The above procedure is repeated for all periods, and the value of the hedging portfolio at time t_y along outer loop s is given by $\sum_i \Delta_{i,y}(s) + B_y(s)$. The terminal P&L at the end of the entire period is the difference between the value of the hedging portfolio (before rebalancing) and the total VA liability at t_T. More precisely, the distribution of the terminal P&L random variable is

$$\sum_i \Delta_{i,T-1}(s)R_{i,T}(s) + B_{T-1}(s)e^{r\delta_t} - L_T(s), s = 1, ..., M.$$

The corresponding estimated P&L distribution can be obtained by replacing $\Delta_{i,T-1}$, $B_{T-1}(s)$ and $L_T(s)$ by their estimated values calculated using the algorithm in Sections 2 and 3. The relevant risk metrics such as the VaR or CVaR at significant level α of the loss distribution can then be obtained from the estimated terminal P&L distribution.

4. NUMERICAL STUDIES

Several numerical studies on the proposed algorithm are now conducted in this section. In the first subsection, we introduce a multivariate regime-switching lognormal (RSLN) model to be used as the real-world and the risk-neutral ESGs in the nested simulation algorithm. For illustration purposes, we create a synthetic VA portfolio containing 100,000 policies whose attributes are given in the second subsection. In the third subsection, the predictive total liability distribution and the partial dollar Deltas of the synthetic VA portfolio are calculated using both the full nested simulation and the proposed algorithm. Results are compared in terms of the approximation accuracy and the algorithm runtime. In the last subsection, we implement a dynamic hedging program with the proposed algorithm to hedge the total liability for the synthetic VA portfolio over multiple time periods and conduct a P&L analysis.

4.1. Multivariate RSLN model

We use the multivariate RSLN model as the ESG for both inner-loop and outer-loop simulations. This model has been used to describe the joint

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

dynamics of a set of correlated assets in many papers such as Ng and Li (2013) and Chen and Yang (2011). We remark that our proposed algorithm does not assume any specific model for the ESG and it works for a wide range of models. In the following, we first specify the model and then demonstrate the scenario clustering in the multiple asset setting using the k-means algorithm.

4.1.1. *Model specification*

We consider a case where the policyholders' account values are invested in three assets: two risky assets and the money market which earns a risk-free rate. We assume the joint dynamics of the two risky assets, $S_{1,t}$ and $S_{2,t}$, follow a bivariate RSLN model under the real-world measure:

$$\begin{cases} dS_{1,t} = \mu^{\mathcal{G}}_{1,t} S_{1,t} dt + \sigma^{\mathcal{G}}_{1,t} S_{1,t} dW^{\mathbb{P}}_{1,t}, \\ dS_{2,t} = \mu^{\mathcal{G}}_{2,t} S_{2,t} dt + \sigma^{\mathcal{G}}_{2,t} S_{2,t} dW^{\mathbb{P}}_{2,t}. \end{cases} \tag{4.1}$$

Parameters that have superscript $\mathcal{G}$ are regime dependent. Since the market is incomplete in the regime-switching framework, the risk-neutral measure is not unique. Ng and Li (2013) used the Esscher transform to find an equivalent risk-neutral measure under which all risky assets earn the risk-free rate, r, on average. The assets volatilities and the regime-specified parameters such as the transition probabilities remain unchanged under the risk-neutral measure. This method for finding a risk-neutral measure has also been used in other studies, for example, Bollen (1998). Hence, the joint dynamics under the risk-neutral measure are

$$\begin{cases} dS_{1,t} = rS_{1,t} dt + \sigma^{\mathcal{G}}_{1,t} S_{1,t} dW^{\mathbb{Q}}_{1,t}, \\ dS_{2,t} = rS_{2,t} dt + \sigma^{\mathcal{G}}_{2,t} S_{2,t} dW^{\mathbb{Q}}_{2,t}. \end{cases} \tag{4.2}$$

For the numerical implementation, we adopt the parameters given in Ng and Li (2013) which are fitted using the weekly S&P500 and S&P600 indices data. The S&P500 index is calculated based on the returns of 500 large-cap US companies; and the S&P600 index, on the other hand, is based on the returns of 600 small-cap US companies. These two indices together give investors a relatively high exposure to the entire US equity market. We denote $R_{1,t}$, $R_{2,t}$ the weekly total returns of the S&P500 index and S&P600 index from $t - \frac{1}{52}$ to t for $t = 1/52, 2/52, \ldots$. The weekly joint dynamics of the two indices under the real-world measure $\mathbb{P}$ in the first regime is

$$\begin{cases} \ln R_{1,t} = 0.003710 + a^{\mathbb{P}}_{1,t}, \\ \ln R_{2,t} = 0.002915 + a^{\mathbb{P}}_{2,t}, \end{cases} \tag{4.3}$$

where the innovation terms $(a^{\mathbb{P}}_{1,t}, a^{\mathbb{P}}_{2,t})'$, $t = 1/52, 2/52, \ldots$, follows a bivariate normal distribution under $\mathbb{P}$ with zero means, standard deviations of 0.009145

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

and 0.006098, and a correlation of 0.8115. The dynamics under the second regime is

$$\begin{cases} \ln R_{1,t} = 0.001010 + b^{\mathbb{P}}_{1,t}, \\ \ln R_{2,t} = 0.000340 + b^{\mathbb{P}}_{2,t}, \end{cases} \tag{4.4}$$

where $(b^{\mathbb{P}}_{1,t}, b^{\mathbb{P}}_{2,t})'$ follows a bivariate normal distribution under $\mathbb{P}$ with zero means, standard deviations of 0.01697 and 0.01411, and a correlation of 0.8115. The innovation terms are assumed independent between regimes. The transition probabilities are $p_{12} = 0.035248$ and $p_{21} = 0.029042$. In addition, we assume a risk-free rate of 2% per year which is equivalent to 0.0385 % per week.

4.1.2. *Scenario clustering*

In Section 2.3, we introduced a policy-independent method for selecting the representative outer loops, in which the k-means algorithm is run with the simulated return vectors. We stated in Proposition 2.1 that if the number of representative outer loops is relatively large (e.g., around 100), then this approach would produce a good partition for the policy's total return regardless of the asset allocation. In the following, we will demonstrate this through several examples.

We generate 1000 outer loops from $t_0 = 0$ to $t_1 = 1/52$ (a week) using the multivariate regime-switching model (4.3) and (4.4). Let $\boldsymbol{R} = (\boldsymbol{R}(1), ..., \boldsymbol{R}(1000))$, where $\boldsymbol{R}(s) = (R_1(s), R_2(s), R_f)^t$ denotes the simulated asset returns at outer loop s whose elements are the returns of the S&P500 index, the S&P600 index and the risk-free asset. Let $\boldsymbol{\omega}_p = (\omega_{p,1}, \omega_{p,2}, \omega_{p,f})^t$ be the asset allocation of policy p, such that the total returns of this policy's account value from t_0 to $t_1 = 1/52$ is $\boldsymbol{\omega}_p^t \boldsymbol{R}$.

There are four subfigures in Figure 3: each corresponds to a generic asset allocation which is given in the title. To demonstrate there are three curves in each subfigure: the black curve corresponds to the function $||\boldsymbol{\omega}_p||^2 \Phi(\boldsymbol{R}, \mathcal{Q}^*_m)$ against different m, where $m = 5, 10, ..., 250$ and $\Phi(\boldsymbol{R}, \mathcal{Q}^*_m)$ are the WCSS obtained by running the k-means algorithm with $\boldsymbol{R}$ and $\mathcal{Q}^*_m$ are the resulted partitions; the red curve corresponds to the function $\Phi(\boldsymbol{\omega}_p^t \boldsymbol{R}, \mathcal{Q}^*_m)$ which are the WCSS of $\boldsymbol{\omega}_p^t \boldsymbol{R}$ with partition $\mathcal{Q}^*_m$; and the blue curve corresponds to the function $\Phi(\boldsymbol{\omega}_p^t \boldsymbol{R}, \mathcal{P}^*_m)$ which are the WCSS obtained by running the k-means algorithm with $\boldsymbol{\omega}_p^t \boldsymbol{R}$ and $\mathcal{P}^*_m$ are the resulted partitions.

The results in Proposition 2.1 can be clearly observed from Figure 3: as the number of cluster increases, the difference between using $\mathcal{Q}^*_m$ and using $\mathcal{P}^*_m$ to partition the total returns $\boldsymbol{\omega}_p^t \boldsymbol{R}$ becomes smaller. Furthermore, it is worth to note that the WCSS curves are not monotonically decreased in m. This is due to the fact that clusterings obtained from the k-means algorithm are local optima.

In order to quantitatively measure the performance of using $\mathcal{Q}^*_m$ to partition $\boldsymbol{\omega}_p^t \boldsymbol{R}$ for different $\boldsymbol{\omega}_p$'s, we calculate the *fraction of the explained variance*,

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

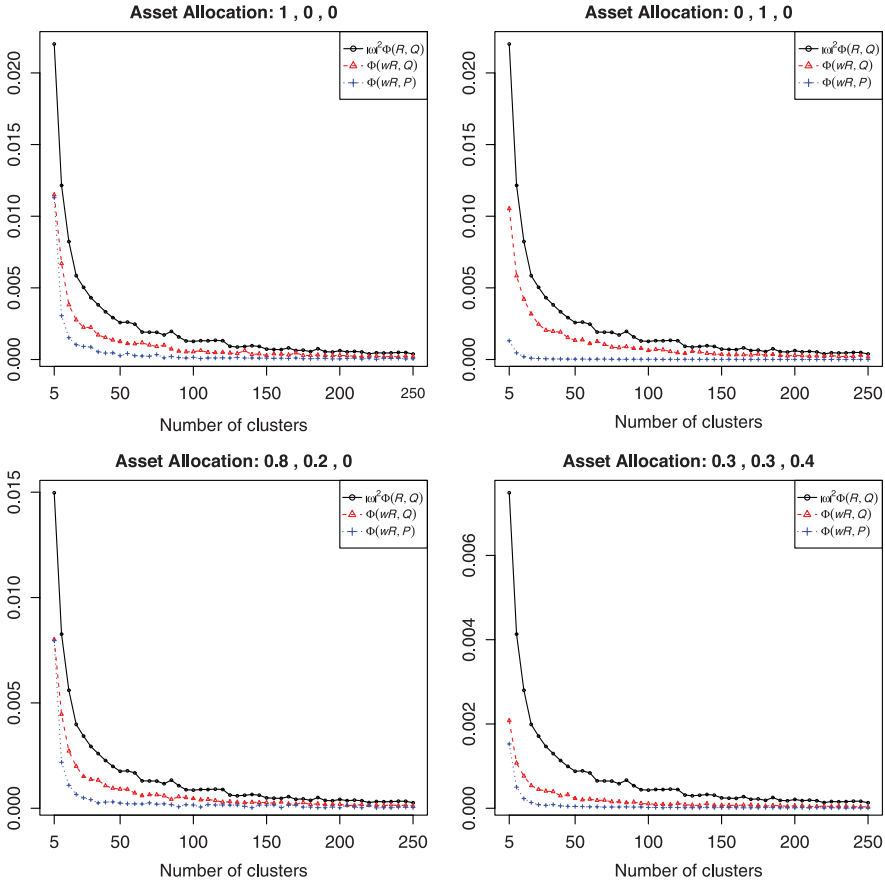

FIGURE 3: Scenario clustering for different asset allocations.

which is defined as the ratio of the between-cluster variation to the total variation of the data. In the VA context, this ratio is given by

$$1 - \frac{\Phi(\boldsymbol{\omega}_p^t \boldsymbol{R}, \boldsymbol{\mathcal{Q}}_m^*)}{\sum_{s=1}^{1000} (\boldsymbol{\omega}_p^t \boldsymbol{R}(s) - \boldsymbol{\omega}_p^t \boldsymbol{\mu})^2}, \tag{4.5}$$

where $\boldsymbol{\mu}$ is the mean vector of all simulated returns $\boldsymbol{R}(s)$, $s = 1, ..., 1000$. The larger this ratio is, the higher proportion of the total variance is explained by clustering and the higher similarity among data within each cluster. In the extreme case where m equals to the total number of data points, this ratio becomes one, as each cluster contains a single datum so there is no within-cluster variations. The idea of using this measurement to attest the effectiveness of clustering can be dated back to Thorndike (1953). Ketchen and Shook (1996) discussed using this ratio to determine the number of clusters in the context of the k-means clustering.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

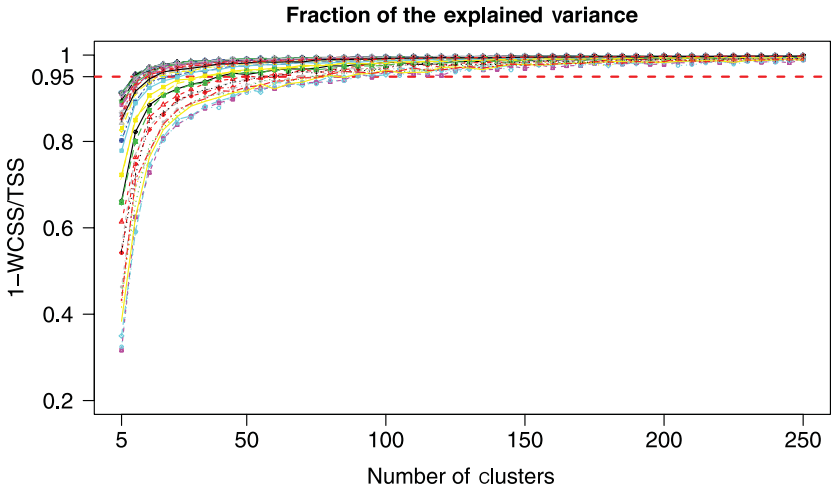

FIGURE 4: Explained variance of different asset allocations.

In Figure 4, we plot several explained variance curves against the number of clusters. Each curve corresponds to (4.5) with a randomly generated asset allocation. The red dashed line is the 95% explained ratio. It can be seen in our setting that 100 clusters (i.e., 100 representative outer loops) would produce a variance explained ratio of at least 95% for almost all asset allocations. Hence, in practice this measurement and corresponding analysis may be used to determine the number of representative outer loops.

4.2. A synthetic VA portfolio

We now create a synthetic VA portfolio containing 100,000 policies whose attributes and the corresponding distributions are given in Table 1. The attribute variables and their distributions are similar to the one we used in Lin and Yang (2020), which are designed according to the Society of Actuaries (SOA) and LIMRA 2015 Study on Variable Annuities.[2] In addition to the demographic information, each policy is assigned a random asset allocation. The possible values of the asset weights and their distributions are given in the bottom three rows of Table 1.

4.3. Efficient VA liability calculation for individual policies

In this subsection, we use two generic but typical VA policies as examples to validate the use of the spline regression approach introduced in Section 2.4 and to demonstrate its efficiency. The attributes of the two policies are given in the following:

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 1

PORTFOLIO ATTRIBUTES AND DISTRIBUTIONS.

Attribute	Value	Distribution
Gender	Male, Female	Uniform
Policyholder's age	45–85	Uniform
Maturity	10–25 years	
Guarantee type	GMDB	GMWB: 15% among 45–60
	GMDB+GMAB	30% among 61–70
	GMDB+GMWB	30% among 71–80
		20% among 81–85
		GMAB: 50% among 45–60
		30% among 61–70
		15% among 71–80
		5% among 81–85
Annual withdrawal rate	1/Maturity	–
Account value	\$10,000, 20,000, ... , 500,000	40% between 10,000 and 50,000
		50% between 50,000 and 250,000
		10% above 250,000
Withdrawal benefit base	Initial account value	–
Death benefit base	Initial account value	–
Death benefit guarantee design	Ratcheting or roll-up	Uniform
	Roll-up rate 1–5%	
Accumulation benefit	Initial account value	–
Accumulation benefit guarantee design	Ratcheting or roll-up	
	Roll-up rate 1–5%	Uniform
Mortality table	1996 IAM	–
Asset allocation:		
– Risk-free asset	40%, 45%, ..., 60%	Uniform
– S&P 500	0, 5%, ..., 60%	Uniform
– S&P 600	The remaining weight	–

- VA1: gender = female, age = 58, initial account value = \$370,000, term of maturity = 20 years, GMDB (roll up 2% per year) + GMMB (roll up 2% per year), asset allocation = (0.25, 0.3, 0.45).
- VA2: gender = male, age = 68, initial account value = \$ 90,000, term of maturity = 24 years, GMDB (ratchet) + GMWB (withdraw rate = 1/24 per year), asset allocation = (0.15, 0.25, 0.6).

The two subfigures at the top and the bottom of Figure 5 correspond to VA1 and VA2, respectively. The two subfigures on the left are scatterplots of the predicted account values and the VA liabilities in a year, which are calculated from 1000 inner loops, at the 100 representative outer loops. From these

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

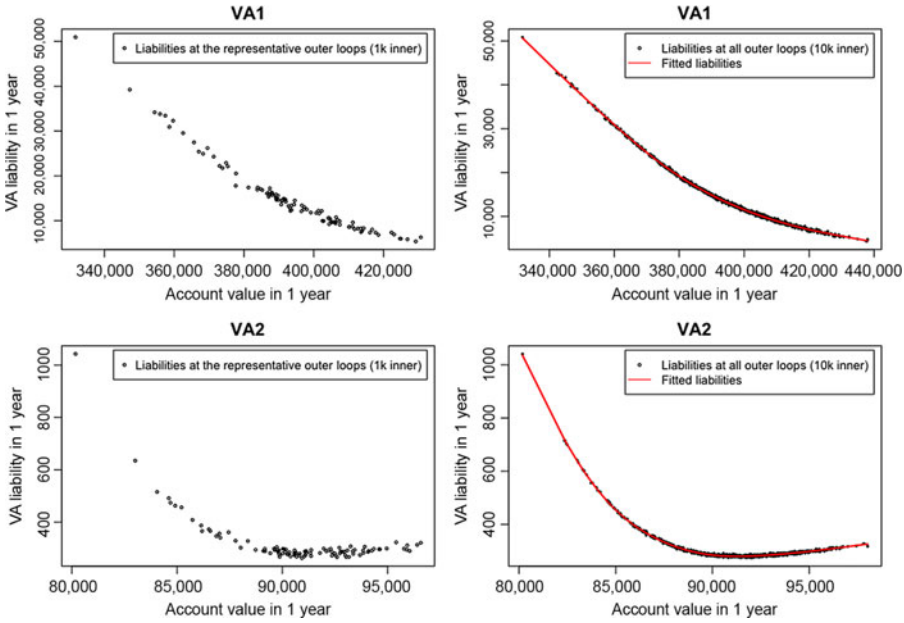

FIGURE 5: Spline regression for VA policies with three underlying assets.

scatterplots, the relationship between the predicted account values and liabilities can be well captured by smooth curves. The two subfigures on the right of Figure 5 show the predicted VA liabilities which are calculated from 10,000 inner loops of the two generic policies at all outer loops, and the VA liabilities that are approximated by the spline regression model. It can be seen that the spline model is able to provide good approximations to the VA liabilities for different cases.

4.4. Predictive total VA liability distribution and partial dollar Deltas

In this section, the performance of the proposed algorithm is illustrated, in which the predictive distribution and the partial dollar Deltas of the total VA liability, for the periods of 12 weeks and 24 weeks, respectively, are calculated using both the full nested simulation and the proposed algorithm. The proposed algorithm will be run with two settings: the numbers of inner loops and outer loops are 1000 and 100 in both settings but the number of representative policies are 2000 and 4000, respectively.

Suppose that $\theta, \hat{\theta}$ are one of the statistics of the predictive total VA liability distribution in Table 2, obtained from the full simulation and the proposed algorithm, respectively. The statistics are compared in terms of their absolute percentage errors $\left|\frac{\hat{\theta}-\theta}{\theta}\right|$. The results in Table 2 show that the proposed algorithm is able to accurately approximate the predictive total VA liability

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 2

ABSOLUTE PERCENTAGE ERRORS OF THE TOTAL VA LIABILITY ESTIMATES.

	n_p = 2000		n_p = 4000	
	26 weeks (%)	52 weeks (%)	26 weeks (%)	52 weeks (%)
Mean	1.38	0.50	0.55	0.03
VaR(90)	1.53	0.77	0.80	0.04
CVaR(90)	1.37	0.63	0.50	0.26
VaR(95)	1.83	0.65	0.73	0.14
CVaR(95)	1.31	1.01	0.64	0.59
VaR(99)	0.94	0.89	0.72	0.42
CVaR(99)	0.95	1.88	0.61	1.85

TABLE 3

COMPARISON OF RUNTIME.

	Full nested simulation	Proposed algorithm	
Number of policies	100,000	2000	4000
Number of outer loops	1000	100	100
Number of inner loops	10,000	1000	1000
Scenario selection	–	0.01(s)	0.01(s)
Policy selection	–	0.5(s)	2(s)
Reduced nested simulation	–	300(s)	500(s)
Spline fitting	–	30(s)	50(s)
Full simulation	2(d)	–	–
Total running time	2(d)	5.5(m)	9(m)

distribution. Comparing the errors between the two settings, the one with 4000 representative policies gives smaller approximation errors in general.

The simulation inputs and runtimes are provided in Table 3. Both of the full nested simulation algorithm and the proposed algorithm are implemented in R by parallel computing with 30 CPU cores (Intel®Xeon®CPU E7-8891 v2 @3.20GHz). The selection of the representative outer loops and the representative policies take almost no time to executive. The majority of the runtime of the proposed algorithm comes from running the reduced nested simulation. The fitting of the spline regression takes certain amount of runtime; however, it is less than 1 min even with 4000 policies.

Next, the proposed algorithm is used to calculate the partial dollar Deltas of the total VA liability with respect to the S&P500 and S&P600 indices. Again, the proposed algorithm is run with two settings with 2000 and 4000 representative polices. For consistency and illustration purposes, we calculate and compare the partial dollar Deltas of the VA portfolio in 26 and 52 weeks, respectively. In Table 4, two error measures of the estimated partial dollar Deltas for the two indices are reported: the average percentage error (APE)

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 4

ESTIMATION ERRORS OF THE PARTIAL DOLLAR DELTAS.

	$n_p = 2000$				$n_p = 4000$			
	26 weeks		52 weeks		26 weeks		52 weeks	
	S&P500 (%)	S&P600 (%)	S&P500 (%)	S&P600 (%)	S&P500 (%)	S&P600 (%)	S&P500 (%)	S&P600 (%)
APE	−0.36	0.32	−0.42	0.30	−0.88	0.54	−0.44	0.24
AAPE	4.87	3.43	5.38	3.63	3.88	1.95	5.14	2.26

and the average absolute percentage error (AAPE) whose definitions are given in the following:

$$\text{APE}_i = \frac{1}{M}\sum_{s=1}^{M}\frac{\hat{\Delta}_i(s)-\Delta_i(s)}{\Delta_i(s)},$$

$$\text{AAPE}_i = \frac{1}{M}\sum_{s=1}^{M}\left|\frac{\hat{\Delta}_i(s)-\Delta_i(s)}{\Delta_i(s)}\right|,$$

where Δ_i and $\hat{\Delta}_i$ are the partial dollar Deltas with respect to asset i obtained from the full nested simulation and the proposed algorithm. In our study, $i = 1, 2$ and $M = 1000$.

The aforementioned errors of the partial dollar Deltas estimates are reported in Table 4. From the numerical results, the proposed algorithm produces close estimates to the partial dollar Deltas. Again, the algorithm with 4000 representative policies in general produce smaller estimation errors comparing to the setting with 2000 representative policies. The runtime of using the full nested simulation to calculate the partial dollar Deltas takes days due to the 'bump and revalue' mechanism, while the time of using the proposed algorithm to calculate the partial dollar Deltas are the same as those reported in Table 3 since rerunning the nested simulation at the 'bumped' scenarios are avoided.

The results in Table 4 are estimated using policies that are selected with $\pi_p = n/N$, $p = 1, ..., N$. As mentioned in Section 2.1, a second-stage selection with unequal inclusion probabilities could be used to reduce the standard error of the partial dollar Delta estimates. In the following, we will apply this two-stage procedure to the estimation of partial dollar Deltas.

Similar to model (2.1), we assume the following linear model between the partial dollar Deltas and the policy attributes at time 0:

$$\Delta_{p,i}(s) = \boldsymbol{\varphi}'(s)\boldsymbol{x}_{p,0} + \varepsilon_p(s), \tag{4.6}$$

where $E(\varepsilon_p(s)) = 0$, $\text{var}(\varepsilon_p(s)) = \varsigma_p^2(s)$, and $i = 1, 2$ correspond the partial dollar Delta of S&P500 index and S&P600 index, respectively. The goal is to identify a form for the residual standard deviation $\varsigma_p(s)$. As proposed in Lin and Yang (2020), we assume a separable form for $\varsigma_p(s)$, where $\varsigma_p(s) = \varrho(s)l(\boldsymbol{x}_{p,0})$. With this assumption, an optimal strategy is to select a balanced sample with $\pi_p = nl(\boldsymbol{x}_{p,0})/\sum_{p=1}^{N} l(\boldsymbol{x}_{p,0})$, $p = 1, ..., N$. We refer the interest readers to Lin and Yang (2020) for the detail of the two-stage selection procedure.

According to the residual diagnostics, we find the residual variations are associated to the initial account value the most and they become significantly more uniform after the residuals are standardized by a power function of the initial account value. This finding can be justified intuitively as the VA liability of policies with larger account values tend to be more sensitive to the underlying assets' movements, and the deviation from the fitted value to the true

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 5

COMPARISON OF THE PARTIAL DOLLAR DELTA ESTIMATES.

	S&P500		S&P600	
Scenario 1	w/o second-stage	w/ second-stage	w/o second-stage	w/ second-stage
Mean	−721, 467	−728, 577	−725, 312	−727, 776
Standard deviation	24,728	22,495	30,124	20,879
CV	3.42%	3.09%	4.15%	2.87%
Scenario 2	w/o second-stage	w/ second-stage	w/o second-stage	w/ second-stage
Mean	−714, 450	−720, 567	−719, 709	−721, 244
Standard deviation	23,447	21,060	29,203	20,172
CV	3.28%	2.92%	4.06%	2.80%
Scenario 3	w/o second-stage	w/ second-stage	w/o second-stage	w/ second-stage
Mean	−690, 761	−694, 744	−698, 237	−698, 210
Standard deviation	20,827	18,416	27,454	18,897
CV	3.02%	2.65%	3.93%	2.71%

value is likely to be large as well. In particular, we identify $l(\boldsymbol{x}_{p,0}) = A_{p,0}^{1/5}$ and use $\pi_p = nA_{p,0}^{1/5} / \sum_{p=1}^{N} A_{p,0}^{1/5}, p = 1, ..., N$, to select the second-stage representative policies.

In order to compare the standard errors of the partial dollar Delta estimates with and without the second-stage selection, we randomly select 50 balanced samples where each contains 4000 policies to estimate the partial dollar Delta of both indices in 26 weeks. The mean, standard errors and the coefficient of variation (cv) of the estimates are reported in Table 5 for three randomly selected scenarios. Both of the mean and standard deviation numbers are in thousands.

From the reported numbers, the mean estimates are very close between the two estimation methods. This is expected as the Horvitz–Thompson estimator is design-unbiased regardless of the inclusion probabilities. However, the standard deviations of the partial dollar Deltas estimated using the second-stage representative policies are indeed smaller. With the second-stage representative policies, the standard deviations of the estimates are reduced by around 10% and 30% for the S&P500 and S&P600 indices, respectively.

Finally, we remark that the use of the second-stage selection may be context-specific. If the goal is to estimate portfolio-level quantities of interest in one period, then incorporating the second stage may be beneficial to increase the confidence level in the estimates. However, when using the proposed algorithms for more complicated valuations such as the P&L analysis (Section 4.5), performing residual diagnostics may not be economical for all future time points and quantities of interest. In such cases, one may use the

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

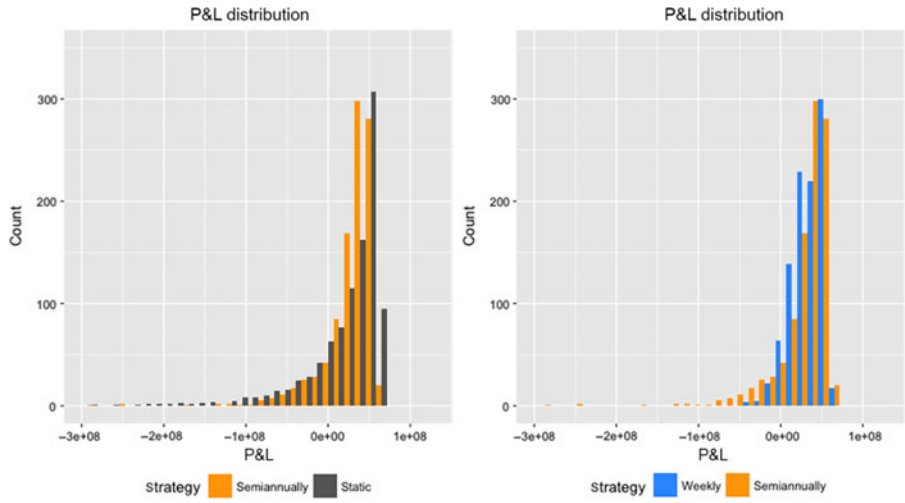

FIGURE 6: P&L distributions with different rebalance frequencies.

same inclusion probability and run the proposed algorithm multiple times in order to reduce the estimates' standard errors.

4.5. P&L analysis

In this subsection, we implement a dynamic hedging program with the proposed algorithm to hedge the total VA liability of the synthetic portfolio presented in Section 4.2 over a year (52 weeks). The proposed algorithm is run with 100 selected representative outer loops, 1000 inner loops and 2000 selected representative policies. Three hedging strategies are applied and compared: static hedging which means no rebalancing occurs since the beginning, semiannually dynamic hedging which has one rebalancing occurs at the 26th week and weekly dynamic hedging.

Following the proposed method in Section 3.3, we break the entire period into several time intervals depending on the rebalancing frequency to calculate the partial dollar Deltas. We conduct the P&L analysis, which are essential for calculating the capital requirements, for the three aforementioned hedging strategies. In Figure 6, the P&L distributions of the three hedging strategies are shown. The horizontal axis represents the P&L of the total portfolio in a year, with positive values representing profits and negative values representing losses. The left (right) subfigure compares the P&L distribution of the static hedging strategy to that of the semiannually (weekly) hedging strategy. As expected, the P&L distributions of the dynamic hedging strategies are more concentrated around zero and have a lighter left tail comparing to that of the static hedging strategy. This implies that the insurance company will have a lower probability of suffering a significant loss from their VA block.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 6

SUMMARY STATISTICS OF THE P&L DISTRIBUTIONS.

	Static(S) (000's)	Semiannually(SA) (000's)	S/SA	Weekly(W) (000's)	SA/W
VaR(95)	75,936	33,269	2.28	1759	18.91
CVaR(95)	152,208	75,204	2.02	11,998	6.27
VaR(97.5)	121,170	56,080	2.16	8261	6.79
CVaR(97.5)	207,699	106,958	1.94	18,838	5.68
VaR(99)	197,928	82,695	2.39	18,231	4.54
CVaR(99)	273,183	163,514	1.67	28,380	5.76

In order to quantify the tail risk, we report several summary statistics of the P&L distributions in Table 6. In the fourth and the fifth columns, we provide the ratios of the risk measure between two different hedging strategies. On average the semiannually dynamic hedging strategy halves the risk measures and the weekly dynamic hedging strategy reduces those measures even further. As a result, the insurance company will have a significant capital requirement reduction if the dynamic hedging strategy is implemented.

4.6. Robustness of the proposed algorithm

The results in the previous subsections are obtained from a single run of the proposed algorithm. Due to the randomness in the sampling procedure, the sets of representative policies and representative outer loops will be different in different runs. This creates a sampling risk to the proposed algorithm. Therefore, it is important to examine the robustness of the performance of the proposed algorithm with respect to different sets of selected representative policies and representative outer loops.

Due to the prohibitive computing time, it is impossible to implement the full nested simulation to test the accuracy and the robustness of the algorithm. Instead we illustrate the robustness of the proposed algorithm by running it 50 times over 52 weeks, an approach similar to bootstrapping. It will be tested with the two settings given in Section 4.4: the number of representative outer loops is 100 under both settings and the number of representative policies are 2000 and 4000, respectively. Each time the algorithm is run, the quantities of interest will be estimated using different sets of representative policies and representative outer loops. The idea here is similar to that of the cross-validation where a model is tested multiple times with different sets of training and testing data sets. If the proposed algorithm is robust with respect to different sets of selected inputs, then one would expect similar estimated quantities over the entire period from different runs.

The results are presented in Figures 7 and 8, where each figure corresponds to a generic real-world economic scenario (outer loop) over the 24-week period.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

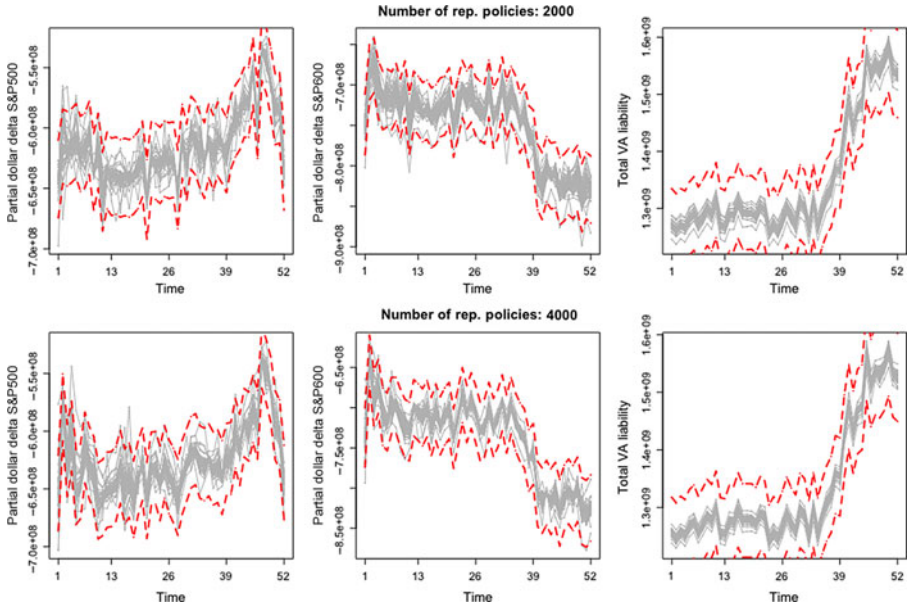

FIGURE 7: Estimates of different quantities from different runs (Scenario 1).

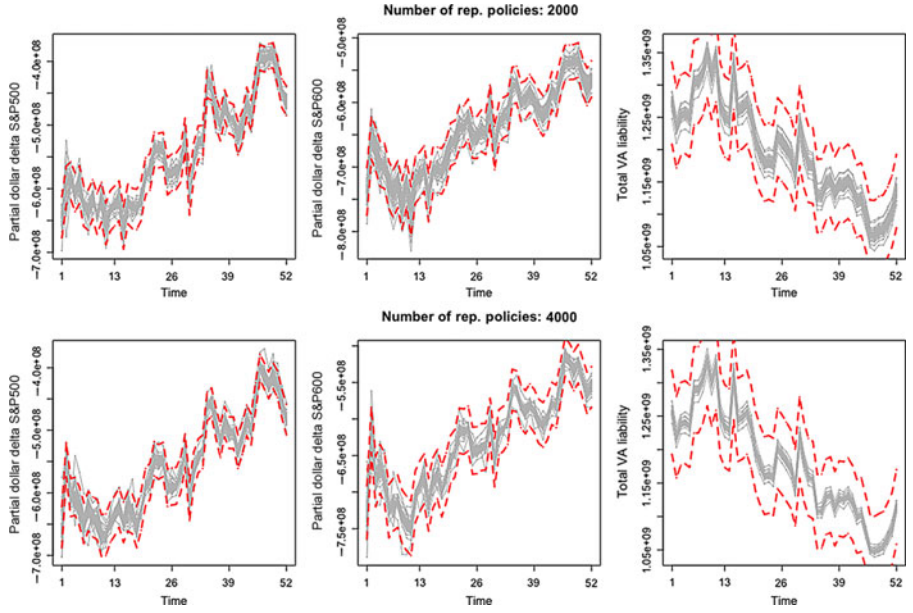

FIGURE 8: Estimates of different quantities from different runs (Scenario 2).

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

The top and bottom three subfigures in each figure correspond to the settings of 2000 and 4000 representative policies, respectively. The subfigures in each row, from the left to the right, display 50 curves of the estimated partial dollar Delta with respect to the S&P 500 index, the S&P 600 index and the estimated total VA liability. The red dashed curves are the mean trajectories of the 50 estimated curves shifted upward and downward by 5%.

Looking at the subfigures, the estimated total liability curves from different simulation runs are almost parallel to each other. In addition, most of the estimated liability curves fall inside the 5% band (5% up and 5% down) of the mean curve, implying the algorithm is robust in estimating the total liability under both settings.

On the other hand, the estimated partial dollar Deltas curves are relatively more volatile. This may be due to the fact that the partial dollar Deltas are second-order quantities whose estimation errors are more sensitive than those of the liability estimates. Nonetheless, the estimated partial dollar Deltas curves of each scenario show the same overall trend throughout the time, and they in general go against the total liability movements. Comparing the estimated partial dollar Deltas curves between the two settings, the ones that are estimated from 4000 representative policies clearly fall in a narrower band and the majority of those estimated curves are away from the mean curves by less than 5%.

We remark that the runtime of a single simulation over 52 weeks using the proposed algorithm with 2000 representative policies is around 2.2 h, and that with 4000 representative policies takes about 3.6 h. With this scale of runtime, the insurance company can not only perform frequent rebalance of the hedging portfolio, but can also perform long-term valuation to assess existing hedging strategy on a regular basis. If more advanced computing system is used, then the total runtime will be further reduced. Under this situation, the insurance company may perform the reduced simulation algorithm multiple times and use the averages from the multiple runs as the estimates for the quantities of interest, which in theory contain less estimation errors. In Appendix A.1, we demonstrate the efficiency of our method using another example with five underlying assets.

5. Further extension

In this section, we show how our proposed approach may be used to incorporate stochastic interest rates and to calculate other Greeks. For illustration, We now assume that the stochastic short rate follows the Vasicek model and the Greek to calculate is Rho. The extension involves a more general form of the penalized spline regression model: the thin plate spline regression model.

5.1. A RSLN–Vasicek model

As in Section 4.1.1, we consider that policyholders' accounts are invested into three assets: two risky assets and the money market, and the two risky assets

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

follow the system of stochastic differential equations (4.1). Now, instead of assuming a constant interest rate, we assume that the short interest rate r_t follows a Vasicek model under the real-world measure:

$$dr_t = \kappa^{\mathbb{P}}(\mu_r^{\mathbb{P}} - r_t)dt + \sigma_r dW_{r,t}^{\mathbb{P}}. \tag{5.1}$$

In order to transit the interest rate dynamics from the real-world measure $\mathbb{P}$ to the risk-neutral measure $\mathbb{Q}$, we apply an affine transformation such that $dW_{r,t}^{\mathbb{Q}} = dW_{r,t}^{\mathbb{P}} + (\lambda_0 + \lambda_1 r_t)dt$. This approach of transition between measures has been used in most finance literature. See Dai and Singleton (2003) for example. As a result, the joint dynamics of the three assets under the risk-neutral measure are

$$\begin{cases} dS_{1,t} = r_t S_{1,t}dt + \sigma_{1,t}^{\mathcal{G}} S_{1,t} dW_{1,t}^{\mathbb{Q}}, \\ dS_{2,t} = r_t S_{2,t}dt + \sigma_{2,t}^{\mathcal{G}} S_{2,t} dW_{2,t}^{\mathbb{Q}}, \\ dr_t = \kappa^{\mathbb{Q}}(\mu_r^{\mathbb{Q}} - r_t)dt + \sigma_r dW_{r,t}^{\mathbb{Q}}, \end{cases} \tag{5.2}$$

where $\kappa^{\mathbb{Q}} = \kappa^{\mathbb{P}} + \sigma_r\lambda_1$ and $\mu_r^{\mathbb{Q}} = \frac{\kappa^{\mathbb{P}}\mu_r^{\mathbb{P}} - \sigma_r\lambda_0}{\kappa^{\mathbb{P}} + \sigma_r\lambda_1}$.

For simplicity, we assume that $dW_{r,t}^{\mathbb{P}}$ ($dW_{r,t}^{\mathbb{Q}}$) is independent of $dW_{1,t}^{\mathbb{P}}$ ($dW_{1,t}^{\mathbb{Q}}$) and $dW_{2,t}^{\mathbb{P}}$ ($dW_{2,t}^{\mathbb{Q}}$), and the parameters in the Vasicek model are the same under both regimes in each risk measure. Realistically speaking, these innovation terms should be correlated and the parameterization should be different in different regimes. However, adding these complications should not affect the performance of our surrogate modeling method.

For the numerical implementation, again we let the two risky assets be the S&P500 and the S&P600 indices. The model parameters are provided in Section 4.1.1. For the Vasicek model, we set $\kappa^{\mathbb{P}} = 0.8$, $\mu^{\mathbb{P}} = 0.02$, $\sigma_r = 0.1$ and $\lambda_0 = 0.01$, $\lambda_1 = 0.5$, which lead to the risk-neutral parameters being $\kappa^{\mathbb{Q}} = 0.85$ and $\mu^{\mathbb{Q}} = 0.01764$.

5.2. Thin plate spline regression

The thin plate spline regression is a tool to approximate multivariate functions. It is the extension of the penalized spline regression model in the multidimensional case. Similar to the penalized regression spline model, the thin plate spline regression is very flexible in capturing different nonlinear relationships between the predictor variables and the response variable. In the following, we adapt the notation in Section 5.5 of Wood (2017).

Consider a multivariate model

$$y_i = g(\boldsymbol{x}_i) + \epsilon_i,$$

where $\boldsymbol{x}_i$ is of d dimensions, and ϵ_i, for $i = 1, ..., n$, are independent error terms with mean zero. The thin plate smoothing spline, denoted by $\hat{f}$, is an estimator

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

of g that minimizes the following objective function:

$$||\boldsymbol{y}-\boldsymbol{f}||^2+\lambda J_{md}(f), \tag{5.3}$$

where $\boldsymbol{y}=(y_1,...,y_n)'$ denotes the observed response variables, $\boldsymbol{f}=(f(\boldsymbol{x}_1),...,f(\boldsymbol{x}_n))'$ are the function values evaluated at the observed predictors, λ is the smoothing parameter which controls the smoothness of the fitted surface and $J_{md}(f)$ is the penalty term which is given by

$$J_{md}(f)=\int\sum_{v_1+...+v_d=m}\frac{m!}{v_1!...v_d!}\left(\frac{\partial^m f}{\partial x_1^{v_1}...\partial x_d^{v_d}}\right)^2 dx_1...dx_d. \tag{5.4}$$

It can be seen that the penalized spline regression model corresponds to the case where $d=1$ and $m=2$. The solution to (5.3) is the *thin plate smoothing spline estimator* which takes the following form:

$$\hat{f}(\boldsymbol{x})=\sum_{i=1}^{n}\delta_i\eta_{md}(||\boldsymbol{x}-\boldsymbol{x}_i||)+\sum_{j=1}^{M}\alpha_j\phi_j(\boldsymbol{x}), \tag{5.5}$$

where δ_i, $i=1,...,n$, and α_j, $j=1,...,M$, are the estimated coefficients; the parameter $\boldsymbol{\delta}$ satisfies the constraint $\boldsymbol{\phi}'\boldsymbol{\delta}=\boldsymbol{0}$ with the (i,j) entry of matrix $\boldsymbol{\phi}$ being $\boldsymbol{\phi}_{ij}=\phi_j(\boldsymbol{x}_i)$; and the $M=\binom{m+d-1}{d}$ basis functions ϕ_i are linear independent polynomials of degree less than m, and they span the space of functions whose J_{md} values equal zero. For instance, if $m=d=2$, then the three polynomial basis functions are $\phi_1(\boldsymbol{x})=1$, $\phi_2(\boldsymbol{x})=x_1$ and $\phi_3(\boldsymbol{x})=x_2$. Further, functions η_{md} are defined as

$$\eta_{md}(r)=\begin{cases}\dfrac{(-1)^{m+1+d/2}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!}r^{2m-d}\log(r), & \text{for even } d,\\ \dfrac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!}r^{2m-d}, & \text{for odd } d.\end{cases}$$

It can be seen that the thin plate smoothing spline estimator contains two parts: the first part $\sum_{i=1}^{n}\delta_i\eta_{md}(||\boldsymbol{x}-\boldsymbol{x}_i||)$ can be thought as 'wiggliness' part which tries to interpolate the data and the second part $\sum_{j=1}^{M}\alpha_j\phi_j(\boldsymbol{x})$ can be thought as the smoothing part which controls the smoothness of the fitted function since its J_{md} value is zero.

As mentioned in Wood (2017), the thin plate smoothing spline estimator has many desired properties except the high computational cost. The latter is due to the expression (5.5) in which the number of parameters equals to the number of data points (there are M linear constraints in $\boldsymbol{\phi}'\boldsymbol{\delta}=\boldsymbol{0}$). This motivates the invention of the *thin plate spline regression* (Wood, 2017).

The idea of the thin plate spline regression is to keep the smoothing term $\sum_{j=1}^{M}\alpha_j\phi_j(\boldsymbol{x})$ in the estimator while it truncates the first part $\sum_{i=1}^{n}\delta_i\eta_{md}(||\boldsymbol{x}-\boldsymbol{x}_i||)$ by extracting the principal components of matrix $\boldsymbol{E}$ where $\boldsymbol{E}_{ij}=$

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

$\eta_{md}(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$. This procedure can be implemented efficiently using the Lanczos iteration. For more technical details, please refer to Wood (2017).

5.3. Estimating Rho from the thin plate spline regression model

Since the account value and the interest rate are measured in different units, we scale both variables to their z-scores prior to the fitting. In particular, given a data set $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)$, the z-score of element θ_j, $j = 1, ..., n$, is defined as $z_j = (\theta_j - \bar{\boldsymbol{\theta}})/\sigma(\boldsymbol{\theta})$, where $\bar{\boldsymbol{\theta}}$ and $\sigma(\boldsymbol{\theta})$, respectively, denote the mean and the standard deviation of $\boldsymbol{\theta}$. This standardization approach has commonly been used to avoid numerical instability issue, for example, Gan and Lin (2015). In fact, this step is essential because the thin plate spline regression is an *isotropic* smoothing approach, which assumes a unit change in one variable is equivalent to a unit change in another variable.

For illustration purpose, we consider a case where the quantities of interest are in a year ($t = 1$). Let $L_p(s), A_p(s)$ again denote the VA liability and account value of policy p at $t = 1$ at outer loop s. Furthermore, we denote $x_{A_p}(s)$ and $x_i(s)$ as the standardized account value and interest rate, respectively. We assume the following model between $L_p(s)$ and $\boldsymbol{x}_p(s) = (x_{A_p}(s), x_i(s))$:

$$L_p(s) = g(\boldsymbol{x}_p(s)) + \epsilon_s, \tag{5.6}$$

where g is fitted by the thin plate spline regression. Next we illustrate the performance of the thin plate regression model in fitting the VA liabilities using some generic and typical VA policies:

- VA5: gender = female, age = 59, initial account value = \$30,000, term of maturity = 13 years, GMDB (roll up 3% per year) + GMWB (withdraw rate = 1/13 per year), asset allocation = (0.35, 0.1, 0.55). That is, the weights of S&P500, S&P600 and the money market are 0.35, 0.1 and 0.55, respectively.
- VA6: gender = female, age = 57, initial account value = \$20,000, term of maturity = 22 years, GMDB (roll up 3% per year) + GMMB (roll up 2% per year), asset allocation = (0.55, 0, 0.45).
- VA7: gender = male, age = 41, initial account value = \$40,000, term of maturity = 25 years, GMDB (roll up 1% per year) + GMMB (ratcheting), asset allocation = (0.35, 0.25, 0.4).
- VA8: gender = male, age = 40, initial account value = \$50,000, term of maturity = 23 years, GMDB (ratcheting) + GMMB (roll up 1% per year), asset allocation = (0.25, 0.35, 0.4).

The following are the simulation configurations for calculating the Rhos under the two methods: full simulation and thin plate regression approach:

- Full simulation: 2000 outer loops, 10,000 inner loops, three runs at +10/-10bps on interest rates and a base case scenario in order to calculate Rhos and the VA liabilities.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

- Thin plate spline regression: 200 selected outer loops (selected by applying scenario clustering on $\boldsymbol{R}(s) = (R_1(s), R_2(s), i(s))$, where $R_1(s)$, $R_2(s)$ and $i(s)$ are the cumulative returns of the S&P 500, S&P 600 indices and the interest rate at $t = 1$ at outer loop s), 10,000 inner loops and one-time run for the base case scenario at the selected outer loops to obtain the VA liabilities to fit the model. The Rhos are calculated using the fitted thin plate spline regression model by plugging in the +10/–10bps adjusted interest rates. For the thin plate spline regression, we set $m = 5$ to prevent the fitted surface from oscillating too much in the domain.

As described above, the inner loops for the thin plate regression approach are run for only a 10th of the entire outer loops. In addition, the 'bump and revalue' method for calculating Rhos is completely avoided. The simulation time is therefore shortened for around 30 times for each individual VA policies.

In Figure 9, we show the performance of the thin plate regression model in approximating the VA liabilities. Each row of these figures corresponds to an example VA listed above. The first subfigure from the left shows the selected training data points which are used to fit the thin plate regression model (black) and the estimated VA liabilities from the fitted model at all outer loops (red). The second subfigure compares the simulated VA liabilities (black) and the estimated ones (red) at all outer loops. It can be seen that the simulated VA liabilities almost lie on a surface with 10,000 inner loops simulated at each outer loop. This finding justifies the use of the thin plate regression approach. For better visualization, the third subfigure is the second one looking along the y-axis ('standardized interest rate'). It is clear that the fitted VA liabilities (red) form a smooth surface and the simulated ones (black) are distributed around the fitted surface. Lastly, the fourth subfigure is the QQ-plots between the simulated liabilities and the fitted ones, which clearly indicate a good performance of the model.

On the other hand, Figure 10 shows four QQ-plots between the simulated Rhos, which are calculated using the 'bump and revalue' method, and the estimated Rhos, which are calculated using the thin plate spline regression model. Again, the data points are distributed around the reference line. However, the estimation errors for the Rhos are in general larger than those of the liability estimates. This is expected since the Rhos are estimated directly from the derivatives of the fitted surface which in theory contain more estimation errors because they are second-order quantities.

Several estimation errors are reported in Table 7. The definition of the APE and the AAPE are given in Section 4.4. From the numerical results, the relative errors of the liability estimates are very small while those of the Rho estimates are comparably larger, which is expected due to the aforementioned reason. In addition, we found that the estimation errors tend to increase as the guarantee becomes more complicated. Finally, we remark that there is a trade-off between the number of training points (or simulation time) and the estimation errors. We experimented the method using 300, 400 and 500 representative outer loops

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

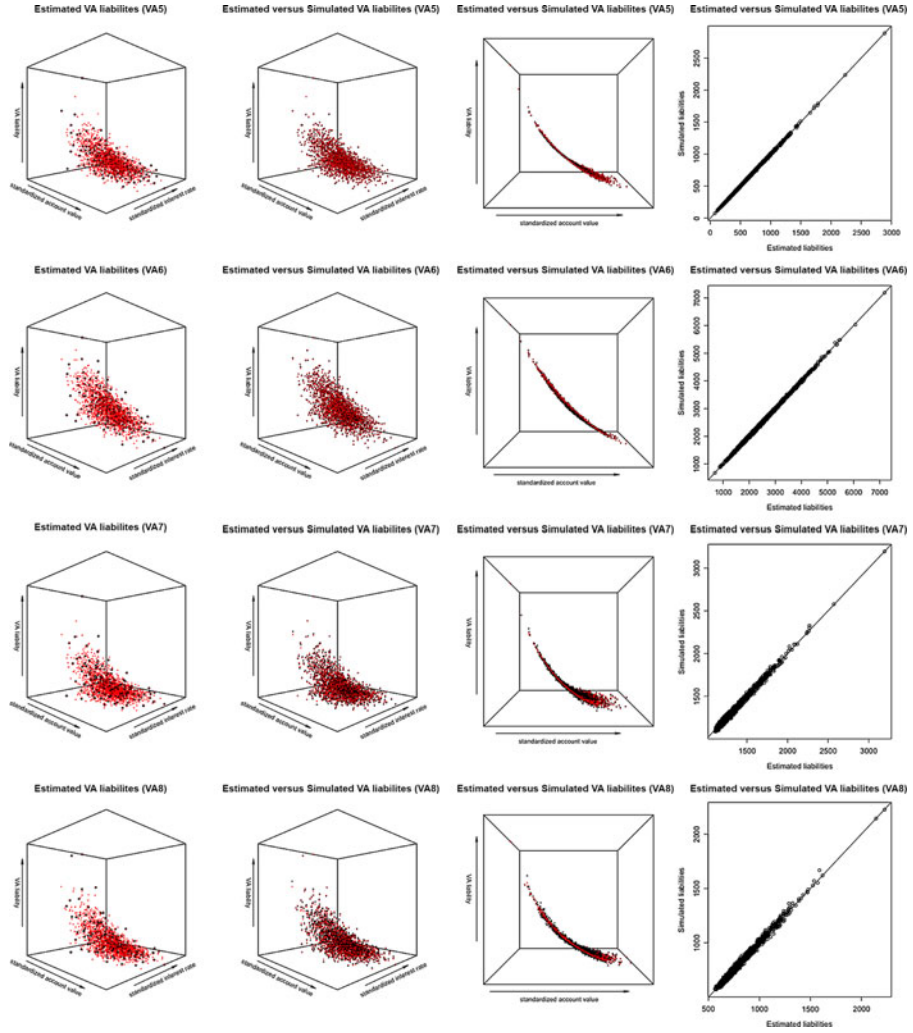

FIGURE 9: Thin plate spline regression fitting results.

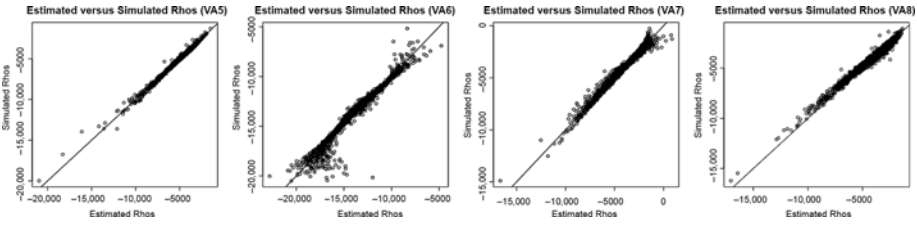

FIGURE 10: Estimated Rho versus simulated Rho.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE 7

ESTIMATION ERRORS OF THE VA LIABILITIES AND RHOS.

	VA5 (%)	VA6 (%)	VA7 (%)	VA8 (%)
Liab APE	0.0222	−0.0582	−0.0471	0.0570
Liab AAPE	0.8782	0.8800	1.0892	1.5530
Rho APE	0.6147	1.1112	−1.9851	3.0870
Rho AAPE	2.2539	2.0516	4.6393	5.8700

out of 2000, and we observed that the estimation errors in general decrease, although not significantly, in the number of representative outer loops among all cases.

Due to the inclusion of the Vasicek model, the inner loops need to be regenerated at each outer loop with a new set of initial values (asset levels, interest rate, etc). Hence, the simulation time for each individual policy is increased significantly. From our experiment, for each VA policy the runtime for a nested simulation with 2000 outer loops and 10,000 inner loops takes approximately 3 h.[3] This implies that the calculation of Rho for the entire outer loops will take around 6 h using the 'bump and revalue' method for each individual VA policy. Because of this, high performing computing hardware such as GPU is required in order to perform the full simulation and assess the performance of the proposed approach. However, the multiple of speed increasing from the proposed approach is invariant across different computing systems (e.g., with 2% representative policies and the previous mentioned simulation parameters, the runtime reduction would be $50 \times 30 = 1500$ times).

6. CONCLUDING REMARKS

For insurance companies that are managing large VA portfolios, hedging against the market risk is critical to ensure solvency. The complexity of the guarantee payoffs and the SoS nature of the nested simulation algorithm make the dynamic hedging of large VA portfolios almost impossible in reality, especially when there are multiple underlying assets. In this paper, we apply a surrogate model-assisted nested simulation framework to efficiently calculate the total VA liability and the partial dollar Deltas for large VA portfolios with multiple underlying assets over multiple time periods. The proposed algorithm is implemented in order to perform a P&L analysis for a large synthetic VA portfolio over a 1-year period, from which the importance and effectiveness of the dynamic hedging strategy are demonstrated. From the numerical results, a weekly dynamic hedging strategy can reduce various risk measures of the predictive total VA liability distribution by half from those of the static hedging strategy. In addition to illustrate the efficiency of the proposed algorithm, we demonstrate its robustness by running the algorithm multiple times to estimate

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

various quantities. Results show that the majority of the estimated values fall into the range of 5% around the means; the total liability estimates are more robust than the partial dollar Delta estimates and the robustness generally increases with the number of representative policies.

We also extend the spline regression to thin plate spline regression to estimate Rho for individual VA policies under a stochastic interest rate setting. The results show the efficiency of our approach and they may lead to various future research directions. One of the future research topics could be incorporating more risk factors such as stochastic volatility into the model. However, as the number of risk factors increases, the model fitting may be limited due to the *curse of dimensionality*. In this case, some dimension reduction tools could be used to reduce the number of predictors in the surrogate models. For example, Cheng *et al*. (2019) applied the transfer learning to extract the key features for each policy in order to perform clustering. Similar method may be applied to risk factors to identify the principal risk factors. On the other hand, the proposed method may be applied to calculate other quantities of interest such as portfolio VaR and CVaR. In these cases, the spline/thin plate spline regression may be applied with different response variables.

NOTES

1. Infographic: Variable Annuity Hedging Survey (2013). Retrieved from https://www.towerswatson.com/en/Insights/Newsletters/Americas/americas-insights/2013/Insights-Variable-Annuity-Hedging-Survey.

2. Variable Annuity Guaranteed Benefits Utilization (2015). Retrieved from https://www.soa.org/globalassets/assets/files/resources/research-report/2018/variable-annuity-guaranteed-utilization.pdf.

3. Run in parallel using the `doSNOW` package in `R` with 30 CPU cores (Intel®Xeon®CPU E7-8891 v2 @3.20GHz).

REFERENCES

ALOISE, D., DESHPANDE, A., HANSEN, P. and POPAT, P. (2009) NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, **75**(2), 245–248.

ARTHUR, D. and VASSILVITSKII, S. (2006) How slow is the k-means method? *Symposium on Computational Geometry*, Vol. 6, pp. 1–10.

BACINELLO, A.R., MILLOSSOVICH, P., OLIVIERI, A. and PITACCO, E. (2011) Variable annuities: A unifying valuation approach. *Insurance: Mathematics and Economics*, **49**(3), 285–297.

BAUER, D. and HA, H. (2015) A least-squares Monte Carlo approach to the calculation of capital requirements. *World Risk and Insurance Economics Congress, Munich, Germany*, pp. 2–6.

BAUER, D., KLING, A. and RUSS, J. (2008) A universal pricing framework for guaranteed minimum benefits in variable annuities. *ASTIN Bulletin: The Journal of the IAA*, **38**(2), 621–651.

BAUER, D., REUSS, A. and SINGER, D. (2012) On the calculation of the solvency capital requirement based on nested simulations. *ASTIN Bulletin: The Journal of the IAA*, **42**(2), 453–499.

BERNARD, C., HARDY, M. and MACKAY, A. (2014) State-dependent fees for variable annuity guarantees. *ASTIN Bulletin: The Journal of the IAA*, **44**(3), 559–585.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

BOLLEN, N.P. (1998) Valuing options in regime-switching models. *Journal of Derivatives*, **6**, 38–50.
BOYLE, P. and HARDY, M. (2003) Guaranteed annuity options. *ASTIN Bulletin: The Journal of the IAA*, **33**(2), 125–152.
CATHCART, M.J., LOK, H.Y., MCNEIL, A.J. and MORRISON, S. (2015) Calculating variable annuity liability 'Greeks' using Monte Carlo simulation. *ASTIN Bulletin: The Journal of the IAA*, **45**(2), 239–266.
CHEN, P. and YANG, H. (2011) Markowitz's mean-variance asset–liability management with regime switching: A multi-period model. *Applied Mathematical Finance*, **18**(1), 29–50.
CHENG, X., LUO, W., GAN, G. and LI, G. (2019) Fast valuation of large portfolios of variable annuities via transfer learning. *Pacific Rim International Conference on Artificial Intelligence*, pp. 716–728.
COLEMAN, M., HAYES, R., LOMBARDO, K. and RUIZ, A. (2019) Variable annuities: Market pressures push the case for model sophistication. Retrieved from https://www.willistowerswatson.com/en/insights.
DAI, M., KUEN KWOK, Y. and ZONG, J. (2008) Guaranteed minimum withdrawal benefit in variable annuities. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, **18**(4), 595–611.
DAI, Q. and SINGLETON, K. (2003) Term structure dynamics in theory and reality. *The Review of Financial Studies*, **16**(3), 631–678.
DE BOOR, C. (1978) *A Practical Guide to Splines.* New York: Springer.
DEVILLE, J.-C. and TILLÉ, Y. (2004) Efficient balanced sampling: the cube method. *Biometrika*, **91**(4), 893–912.
DUONG, Q.D. (2019) Application of Bayesian penalized spline regression for internal modeling in life insurance. *European Actuarial Journal*, **9**(1), 67–107.
GAN, G. and LIN, X.S. (2015) Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance: Mathematics and Economics*, **62**, 138–150.
GAN, G. and LIN, X.S. (2017) Efficient Greek calculation of variable annuity portfolios for dynamic hedging: A two-level metamodeling approach. *North American Actuarial Journal*, **21**(2), 161–177.
GAN, G. and VALDEZ, E.A. (2017) Valuation of large variable annuity portfolios: Monte Carlo simulation and synthetic datasets. *Dependence Modeling*, **5**(1), 354–374.
GAN, G. and VALDEZ, E.A. (2018) Regression modeling for the valuation of large variable annuity portfolios. *North American Actuarial Journal*, **22**(1), 40–54.
GLASSERMAN, P. (2013) *Monte Carlo Methods in Financial Engineering*, Vol. 53. New York: Springer Science & Business Media.
HEJAZI, S.A. and JACKSON, K.R. (2016) A neural network approach to efficient valuation of large portfolios of variable annuities. *Insurance: Mathematics and Economics*, **70**, 169–181.
HONG, L.J., JUNEJA, S. and LIU, G. (2017) Kernel smoothing for nested estimation with application to portfolio risk measurement. *Operations Research*, **65**(3), 657–673.
KETCHEN, D.J. and SHOOK, C.L. (1996) The application of cluster analysis in strategic management research: An analysis and critique. *Strategic Management Journal*, **17**(6), 441–458.
KRAH, A.-S., NIKOLIĆ, Z. and KORN, R. (2018) A least-squares Monte Carlo framework in proxy modeling of life insurance companies. *Risks*, **6**(2), 62.
LIN, X.S. and TAN, K.S. (2003) Valuation of equity-indexed annuities under stochastic interest rates. *North American Actuarial Journal*, **7**(4), 72–91.
LIN, X.S., TAN, K.S. and YANG, H. (2009) Pricing annuity guarantees under a regime-switching model. *North American Actuarial Journal*, **13**(3), 316–332.
LIN, X.S. and YANG, S. (2020) Fast and efficient nested simulation for large variable annuity portfolios: A surrogate modeling approach. *Insurance: Mathematics and Economics*, **91**, 85–103.
LLOYD, S. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory*, **28**(2), 129–137.
MEYRICKE, R. and SHERRIS, M. (2014) Longevity risk, cost of capital and hedging for life insurers under solvency ii. *Insurance: Mathematics and Economics*, **55**, 147–155.

MILEVSKY, M.A. and POSNER, S.E. (2001) The titanic option: valuation of the guaranteed minimum death benefit in variable annuities and mutual funds. *Journal of Risk and Insurance*, **68**(1), 93–128.
MILEVSKY, M.A. and SALISBURY, T.S. (2006) Financial valuation of guaranteed minimum withdrawal benefits. *Insurance: Mathematics and Economics*, **38**(1), 21–38.
NEDYALKOVA, D. and TILLÉ, Y. (2008) Optimal sampling and estimation strategies under the linear model. *Biometrika*, **95**(3), 521–537.
NG, A.C.-Y. and LI, J.S.-H. (2013) Pricing and hedging variable annuity guarantees with multiasset stochastic investment models. *North American Actuarial Journal*, **17**(1), 41–62.
REBAGLIATI, N. (2013) Strict monotonicity of sum of squares error and normalized cut in the lattice of clusterings. *International Conference on Machine Learning*, pp. 163–171.
THORNDIKE, R.L. (1953) Who belongs in the family? *Psychometrika*, **18**(4), 267–276.
VARNELL, E., KENT, J., WARD, R., OSMAN, R. and GILCHRIST, A. (2019) Insurers face challenges on manage- ment actions. Retrieved from http://ch.milliman.com/uploadedfiles/insight/life-published/pdfs/solvency-ii-presents-challenges.pdf.
WOOD, S.N. (2017) *Generalized Additive Models: An Introduction with R.* London: Chapman and Hall/CRC.

X. SHELDON LIN
Department of Statistical Sciences
University of Toronto
100 St George Street
Toronto, ON M5S 3G3, Canada
E-Mail: sheldon@utstat.toronto.edu

SHUAI YANG (Corresponding author)
Department of Statistical Sciences
University of Toronto
100 St George Street
Toronto, ON M5S 3G3, Canada
and
PathWise Solutions Group LLC
Aon
Suite 2300, 20 Bay Street
Toronto, ON M5J 2N9, Canada
E-Mails: shuai.yang@mail.utoronto.ca, alex.yang2@aon.com

APPENDIX A

A.1. A case study with five underlying assets

In this Appendix, we apply our proposed algorithm to another simulation study in which five underlying assets are invested as VA policyholders typically choose more than two mutual funds. In order to do this, we assign each policy in the synthetic portfolio (Section 4.2) a random asset allocation through fund mapping (Gan and Valdez, 2017). We conduct the P&L analysis for the VA portfolio under different hedging strategies over 26 weeks, in which the portfolio total liabilities and partial dollar Deltas are estimated through the efficient nested simulation algorithm. Regarding the configuration of the efficient algorithm, we use 4000 representative policies and 200 (out of 1000) representative outer loops.

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE A1

FUND MAPPING OF 10 INVESTMENT FUNDS.

Fund	US large	US small	Intl equity	Fixed income	Money market
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	0.6	0.4	0	0	0
7	0.5	0	0.5	0	0
8	0.5	0	0	0.5	0
9	0	0.3	0.7	0	0
10	0.2	0.2	0.2	0.2	0.2

A.2. Fund mapping

In practice, the policyholders may be able to choose which assets to invest from a large pool of investable funds. This makes the hedging of the VA portfolio extremely difficult. In order to implement the hedging program, the insurance company normally maps the underlying investable funds to a smaller set of funds (see Gan and Valdez, 2017). This can be done by regressing the returns of an underlying fund to those of the small set of funds.

We adapt the fund mapping strategy used in Gan and Valdez (2017) to generate the asset allocation for all policies in the synthetic portfolio. We assume there are 10 investable funds can be chosen by the policyholders and these 10 funds are mapped to 5 index funds: US large, US small, International equity, Fixed income and Money market. The fund mapping is given in Table A1.

To generate an asset allocation, we first generate a random number r from 1 to 10, then r funds are randomly picked from the 10 funds and the asset allocation is obtained by averaging the mapping coefficients of the selected funds. For example, suppose we have $r = 3$ and the three chosen funds are fund 1, 2 and 10. Then the asset allocation of the five funds is (0.4, 0.4, 0.2/3, 0.2/3, 0.2/3). In theory, there are up to 2^{10} possible asset allocations based on the 10 mapped funds given in Table A1.

A.3. Economic scenario generator

The joint dynamic of the five index funds are assumed to follow a multivariate RSLN model (see Section 4.1.1). The parameters of the model are given in Table A2.

A.4. Spline modeling

In this subsection, we justify the use of the spline regression model for the five asset case. Similar to Section 2.4, we use two generic VA policies whose attributes including asset allocations are provided in the following:

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE A2

PARAMETRIC ASSUMPTION OF THE ESG.

	US large	US small	Intl equity	Fixed income	Money market
Weekly drift (R1)	0.2134%	0.2563%	0.1604%	0.0879%	0.0508%
Weekly drift (R2)	−0.0565%	−0.0977%	−0.1481%	0.0879%	0.0508%
Weekly vol (R1)	1.5254%	2.0039%	1.7451%	0.4341%	0.0901%
Weekly vol (R2)	3.0578%	4.0091%	3.4918%	0.4341%	0.0901%
Correlation					
US large	1				
US small	0.8068	1			
Intl equity	0.7906	0.7025	1		
Fixed income	−0.1028	−0.1887	−0.1027	1	
Money market	0.0226	−0.0215	−0.0007	0.1559	1
Transition prob.	$p_{R1\to R2}$= 0.05	$p_{R2\to R1}$= 0.05			

- VA3: gender = female, age = 55, initial account value = $120,000, term of maturity = 23 years, GMDB (roll up 2% per year) + GMWB (withdraw rate = 1/23 per year), asset allocation = (0.275, 0.1, 0.125, 0.25, 0.25)
- VA4: gender = male, age = 65, initial account value = $150,000, term of maturity = 20 years, GMDB (ratchet), asset allocation = (0.30, 0.26, 0.34, 0.1, 0)

Figure A1 demonstrates the effectiveness of the spline regression model in the five assets case. Again, the two subfigures on the left support the use of smooth curves to approximate the relationship between the predicted account values and liabilities. The two subfigures on the right show that the fitted spline models can approximate the VA liabilities with high accuracy.

A.5. P&L analysis

We compare the P&L distribution of three strategies: static hedging, quarterly hedging (rebalance at the 13th week) and weekly hedging (see Figure A2). The overall observation is consistent with that in Figure 6. As the hedging becomes more frequent, the P&L distribution becomes more spiked. The risk metrics and reduction ratios for the five asset case are reported in Table A3. Due to the modeling assumption, the notional amounts and reduction ratios are different from those in Table 6. However, similar to what is observed from Table 6, all of the risk metrics have been reduced significantly when moving from static hedging to dynamic hedging.

A.6. Robustness test

Similar to the robustness study presented in Section 4.6, here we run the proposed nested simulation to estimate the total portfolio liabilities and partial dollar Deltas 20 times over

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

TABLE A3

SUMMARY STATISTICS OF THE P&L DISTRIBUTIONS.

	Static(S) (000's)	Quarterly(Q) (000's)	S/Q	Weekly(W) (000's)	Q/W
VaR(95)	4502	1564	2.88	564	2.77
CVaR(95)	5513	2507	2.20	960	2.61
VaR(97.5)	5140	2049	2.51	904	2.27
CVaR(97.5)	6234	3215	1.93	1238	2.60
VaR(99)	6092	3221	1.89	1209	2.66
CVaR(99)	7270	4340	1.68	1545	2.81

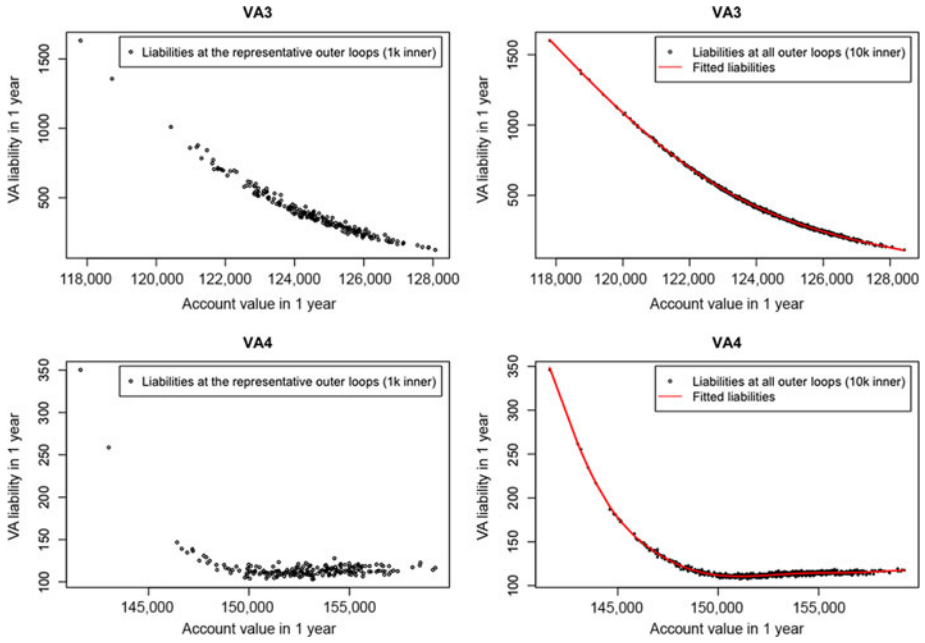

FIGURE A1: Spline regression for VA policies with five underlying assets.

26-week period to demonstrate the robustness of the algorithm for the five assets case. Two generic real-world scenarios are used for illustration purpose. The overall observation is consistent with that in Sectoin 4.6; the liability trajectories are almost parallel to each other across different simulation runs while the partial dollar Delta paths are more wiggly. Most of the curves, however, fall in the 5% range around the mean curves (see Figures A3 and A4).

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

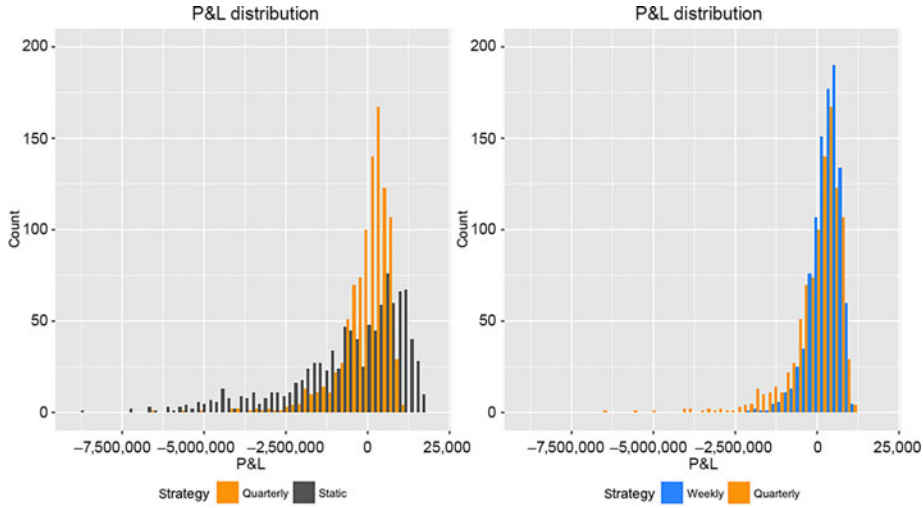

FIGURE A2: P&L distributions with different rebalance frequencies.

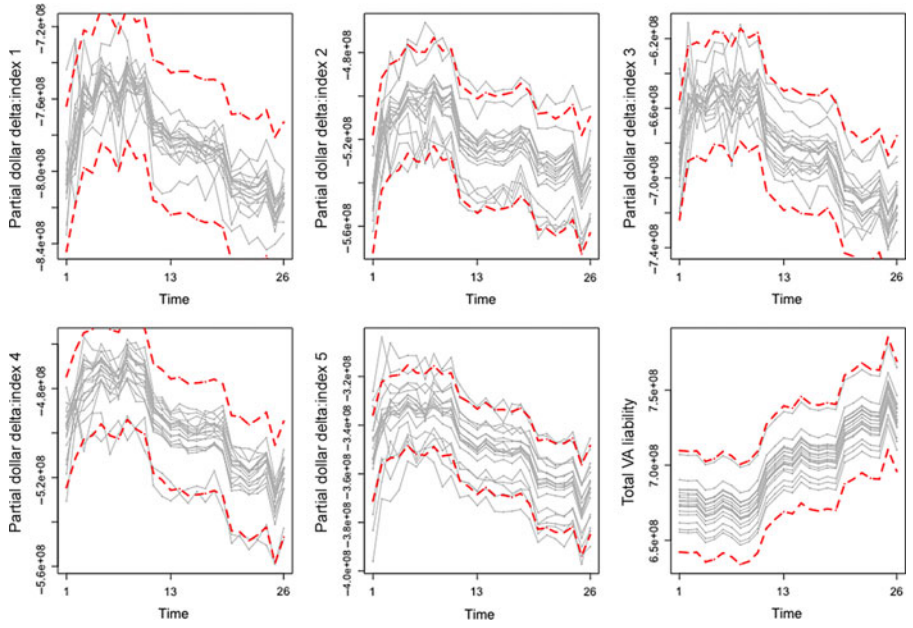

FIGURE A3: Estimates of different quantities from different runs (Scenario 1).

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press

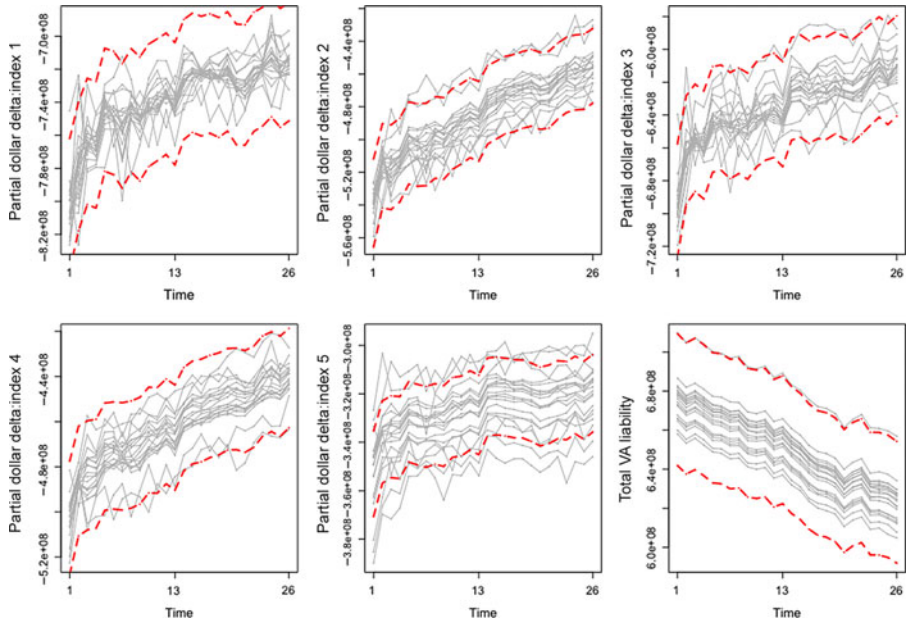

FIGURE A4: Estimates of different quantities from different runs (Scenario 2).

https://doi.org/10.1017/asb.2020.26 Published online by Cambridge University Press