North American Actuarial Journal

Routledge
Taylor & Francis Group

ISSN: 1092-0277 (Print) 2325-0453 (Online) Journal homepage: https://www.tandfonline.com/loi/uaaj20

Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios

Ben Mingbin Feng, Zhenni Tan & Jiayi Zheng

To cite this article: Ben Mingbin Feng, Zhenni Tan & Jiayi Zheng (2020) Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios, North American Actuarial Journal, 24:2, 275-289, DOI: 10.1080/10920277.2019.1685394

To link to this article: https://doi.org/10.1080/10920277.2019.1685394

Published online: 31 Jan 2020.

North American Actuarial Journal, 24(2), 275–289, 2020
© 2020 Society of Actuaries
ISSN: 1092-0277 print / 2325-0453 online
DOI: 10.1080/10920277.2019.1685394

Routledge
Taylor & Francis Group

Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios

Ben Mingbin Feng, Zhenni Tan, and Jiayi Zheng
Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada

The valuation of large variable annuity portfolios is an important enterprise risk management task but is computationally challenging due to the need for simulation. Existing methods in the literature only use simple experimental designs with significant room for improvement. This article identifies three major components in an efficient valuation framework. In addition, we propose optimal experimental designs and provides analytical insights for each component. Our numerical results show that our proposal achieves significantly higher accuracy than state-of-the-art alternatives without requiring any additional computational resource.

1. INTRODUCTION

Variable annuities (VAs) are equity-linked annuity products that offer policyholders the opportunity to invest in predefined sub-accounts, set up by insurance companies, that are invested in equities and/or bonds. Typically the policyholder makes one lump-sum payment or a series of payments to the insurance company in exchange for benefit payments immediately or at some future date. On one hand, VAs enable policyholders to gain investment returns through various types of rides. On the other hand, VAs are exposed to both market risk and mortality risk, creating challenging risk management problems to insurers, which can only be solved through time-consuming computer simulation experiments. The goal of this article is to propose, analyze, and test an efficient simulation procedure to valuate large VA portfolios.

Over the last decade, VAs have become one of the most innovative and popular insurance products for retirees, with over $2 billion annual sales in the United States from 2010 to 2017. One reason for such popularity is the flexibility that VAs offer through different well-designed embedded guarantees to satisfy the needs of different policyholders. There are two main categories of embedded guarantees: guaranteed minimum death benefit (GMDB) and guaranteed minimum living benefit (GMLB). A GMDB provides a guaranteed minimum amount to beneficiaries upon the death of the policyholder. A GMLB often provides some form of guaranteed benefits to the beneficiaries upon survival of the policyholder at a certain time. See Hardy (2003) for details of different VA guarantees.

VA guarantees range from the apparently straightforward embedded put options of standard guaranteed minimum maturity benefits (GMMBs), through to complex combinations of path-dependent, exotic lookback, and tandem options, for example, in the guaranteed minimum income benefit (GMIB) studied by Marshall, Hardy, and Saunders (2010). Because of the complexity of the embedded guarantees, Monte Carlo simulation is often the only feasible method to estimate the fair market value of a VA contract. Depending on the specific simulation models used, such estimation could be computationally intensive. Such computational challenge is further aggravated due to the large size of VA portfolios in practice.

Efficient valuation of large VA portfolios has attracted significant research interest recently (see, for example, Gan 2013, 2015a; Gan and Lin 2015; Liu and Tan 2017; Xu et al. 2018). In this article, we consider a three-component valuation framework as shown in Figure 1. The proposed procedure requires commensurate computation similar to those proposed in Gan (2013) and Gan and Lin (2015) but has significantly higher accuracy. Figure 1 is a new representation of the valuation framework, which we believe is more illustrative and straightforward than the original representation in Gan and Huang (2017). We acknowledge that estimating sensitivities—that is, Greeks—is also an important task in practice. The proposed simulation procedure can be easily adapted to such a case by replacing the valuation simulator by a Greek simulator.

Address correspondence to Ben Mingbin Feng, Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. E-mail: ben.feng@uwaterloo.ca

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaaj.

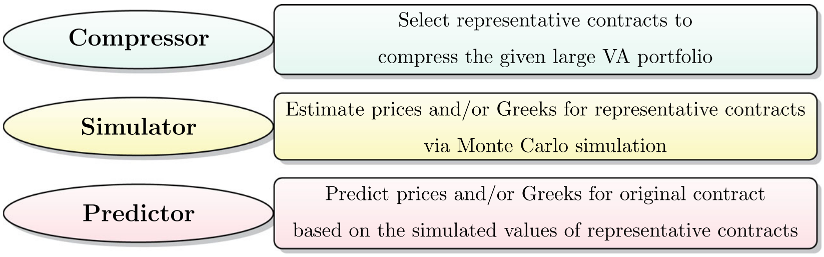
Compressor
Select representative contracts to compress the given large VA portfolio
Simulator
Estimate prices and/or Greeks for representative contracts via Monte Carlo simulation
Predictor
Predict prices and/or Greeks for original contract based on the simulated values of representative contracts

FIGURE 1. Three-Component Valuation Framework for Large VA Portfolios.

The main contributions of this article include the following:

- Innovative and efficient experimental designs in all three components of the valuation framework in Figure 1. Specifically, we consider a new compressor that has provable optimality, a two-stage simulator that aims to optimally allocate a given computation budget, and a simplistic predictor that serves as a worst-case benchmark for future development of new predictors within the framework.
- General design principles and analytical insights for efficient simulation procedures in the three-component valuation framework. For example, one should consider the synergy among the three components rather than designing each one of them in isolation.
- A comprehensive numerical study to illustrate the advantages of the three components in our proposed simulation procedures compared to other well-known alternatives. In particular, we demonstrate that the performance of each component as well as the overall simulation procedure is significantly more accurate, less computationally demanding, or both compared to well-known alternative procedures in the literature.

The remainder of this article is structured as follows. Relevant literatures on variable annuities, clustering algorithms, and simulation designs is reviewed in Section 2. The designs and analyses for the compressor, simulator, and predictor are provided in Sections 3, 4, and 5, respectively. Section 6 presents results of three numerical experiments to demonstrate the efficiency of our proposed method. Section 7 concludes with main findings in the current research and inspirations for future research.

2. LITERATURE REVIEW

Broadly speaking, the selection of representative VAs contracts from a given large VA portfolio can be categorized into three approaches: The first approach, as studied by Liu and Tan (2017) and Hejazi, Jackson, and Gan (2017), draws ideas from low-discrepancy sequence in quasi Monte Carlo (Niederreiter 1992) and Latin hypercube sampling (McKay, Beckman, and Conover 1979) to select representative contracts. Such selection of representative contracts has little reliance, if at all, on the characteristics of the given portfolio; this is referred to as the "portability property" by Liu and Tan (2017). The second approach matches some statistical properties—for example, moments—between the representative portfolio and the original portfolio (e. g., Vadiveloo 2012; Gan and Valdez 2018). The last approach uses clustering algorithms to select representative VA contracts. Gan (2013) proposed using the *k*-prototypes algorithm (Huang 1998) because it is suitable to cluster mixed-type data; Gan and Huang (2017) proposed a TFCM++ algorithm based on the well-known truncated fuzzy *c*-means (TFCM) algorithm (Dunn 1973; Bezdek 2013; Gan, Lan, and Sima 2016) and is initialized by *k*-means++ (Arthur and Vassilvitskii 2007). Our proposal falls into this last category and borrows ideas from clustering using representatives (Guha, Rastogi, and Shim 1998), which is a clustering algorithm designed specifically for large data sets.

Optimal simulation budget allocation (Chen et al. 2000, 2008; Lee et al. 2004) is an effective technique for improving the efficiency of simulation experiments. See Glasserman (2004) and Broadie, Du, and Moallemi (2011) for optimal allocations in financial and actuarial applications. In the same spirit, we show in Section 4 that, for accurate valuation of a VA portfolio, it is optimal to allocate more simulations to contracts with greater complexity.

3. COMPRESSOR: HARNESS THE POWER OF RANDOMNESS

Gan (2013) and Gan and Lin (2015) argued that *k*-prototypes (Huang 1998) is a suitable clustering algorithm for selecting representative contracts in a large VA portfolio; we agree with their arguments and use *k*-prototypes as a key component in the compressor. They also suggested two ad hoc heuristics to the *k*-means algorithm to address some practical considerations. In this section we identify shortcomings in their heuristics and propose an alternative practical remedy with similar computations and far better performance.

3.1. k-Prototypes and the Subset Clustering Heuristics

The following summary of the k-prototypes algorithm is based on Gan (2013) and Gan and Lin (2015). Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ be a portfolio of N VA contracts, where $\boldsymbol{x}_i$ encodes a vector of attributes (e.g., account value, gender, time to maturity, etc.) of the ith VA contract. Suppose that all VA contracts can be represented by d attributes: the first d_1 attributes are numeric and the other $d_2 = d - d_1$ are categorical. The dissimilarity or the distance between two contracts $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as (Huang 1998)

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{h=1}^{d_1} (x_{ih} - x_{jh})^2 + \lambda \sum_{h=d_1+1}^{d} \mathbf{1}\{x_{ih} = x_{jh}\}}, \tag{1}$$

where x_{ih} (x_{jh}) denotes the hth component of $\boldsymbol{x}_i$ ($\boldsymbol{x}_j$, respectively) and $\mathbf{1}\{\cdot\}$ is the indicator function. The weight λ is used to balance the two types of attribute; this is not the focus of this article.

Remark 1. In our numerical experiments, every numerical attribute is standardized by subtracting its minimum and dividing by its range. The standardized numerical attributes are between 0 and 1 and thus have the same range as the categorical attribute, so we set $\lambda = 1$.

The goal of our compressor is to partition $\boldsymbol{\mathcal{X}}$ into K homogeneous clusters $\boldsymbol{\mathcal{C}}_1, ..., \boldsymbol{\mathcal{C}}_K$ such that contracts within the same cluster are similar to each other and those in different clusters are distinct. Mathematically, the following within-cluster sum of squared (WCSS) is minimized:

$$WCSS(\boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}_1, ..., \boldsymbol{\mathcal{C}}_K) = \sum_{j=1}^{K} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{C}}_j} D^2(\boldsymbol{x}, \bar{\boldsymbol{x}}_j), \tag{2}$$

where $D(\cdot,\cdot)$ is as defined in (1) and $\bar{\boldsymbol{x}}_j$ is the center or prototype of cluster $\boldsymbol{\mathcal{C}}_j$. The WCSS is a measure of the "representativeness" of the selected VA contracts $\bar{\boldsymbol{\mathcal{X}}} = \{\bar{\boldsymbol{x}}_1, ..., \bar{\boldsymbol{x}}_K\}$ to the entire portfolio $\boldsymbol{\mathcal{X}}$; the smaller it is, the more representative $\bar{\boldsymbol{\mathcal{X}}}$ is. After a random initialization of K cluster prototypes, the k-prototypes algorithm repeats the following two steps until convergence:

1. **Update memberships**: Assign each VA contract to the closest prototype.
2. **Update prototypes**: For each cluster, update each numerical (categorical) attribute of the cluster prototype as the mean (mode, respectively) of that attribute among all members in that cluster.

Gan (2013) and Gan and Lin (2015) proposed a clustering algorithm to select representative contracts, which we will referred to as the subset clustering (SC) heuristic hereinafter. In essence, the SC heuristic adds the following two ad hoc modifications to the standard k-prototypes algorithm:

- **Divide and conquer**: In practice, the size of the VA portfolio is often so large that direct application of k-prototypes takes unacceptably long. So Gan (2013) and Gan and Lin (2015) proposed a heuristic that first divides the given portfolio of N contracts into s subsets and then applies k-prototypes in each subset; the number of contracts and prototypes in each subset is N/s and K/s respectively. The K/s prototypes across s subset are then collected for further processing.

  Figure 2 illustrates the deficiency of such divide and conquer strategy using a two-dimensional example, where eight clusters are desired from 400 uniform random points in the unit square $[-1, 1]^2$. Euclidean distance is used for illustration so the k-prototypes algorithm coincides with the k-means algorithm. The problem is divided into two subsets, each having 200 points and four clusters. The four distinct clusters in each subsets are labeled by different shapes and colors. As shown in Figure 2, the four cluster centers in the first subset are fairly close to the cluster centers in the second subset. Using similar logic, the representative VA contracts identified in each subset can be fairly similar. This is dissatisfying, both intuitively and mathematically.

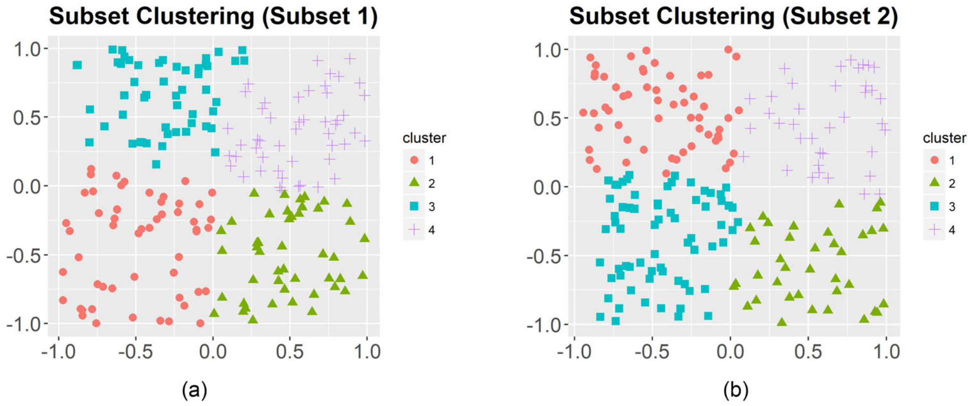

FIGURE 2. A 2D Illustration of the Deficiency of the SC Heuristic: (a) SC Heuristic in the First Subset and (b) SC Heuristic in the Second Subset. *Note:* Suppose that eight clusters are desired from 400 uniform random points in $[-1,1]^2$. The problem is divided into two subsets, each having 200 points and four clusters (with different shapes and colors).

- **Nearest neighbor mapping**: Though the prototypes identified by the *k*-prototypes algorithm encodes different VA contracts, they are often not contracts in the given portfolio. So Gan (2013) and Gan and Lin (2015) proposed a cleanup heuristic that maps each final prototype to the nearest neighbor, among all original VA contracts. The distinct nearest neighbors are then used as the representative contracts.

  This cleanup step has some undesirable consequences too. For instance, because the prototypes identified in different subsets may be very similar, some cluster centers are mapped to the same nearest neighbor contract in the original portfolio. This means that some prototypes are redundant and thus to be removed. This redundancy reduces the number of representative contracts and wastes some computations. Last but not least, to the best of our knowledge, there is no systematic way to determine the number of subsets. Section 3, Page 797 in Gan (2013) suggested that “cluster each subset into 3 or 4 clusters” without justifications.

  This cleanup step is also costly and ineffective. It requires $N \times K$ pairwise distance calculations, which is a considerable number of computations. Secondly, in our preliminary studies, such clean-up step in fact worsens the WCSS in the compressor and does not offer any improvement in the valuation of the VA portfolio.

In summary, these two modifications are not the best use of computations and in some cases they can be wasteful.

3.2. The Simple Random Sampling and Clustering Method

To address the practical considerations in compressing large VA portfolios and to avoid the shortcomings of the SC heuristic, we propose a new compressor design, called the simple random sampling and clustering (SRSC) method, that selects representative contracts in a given large VA portfolio in two steps:

1. Draw a random sample of size n^* from the given VA contracts.
2. Apply a chosen clustering algorithm to the selected random sample.

Remark 2. In our numerical experiments, the k-prototypes algorithm is used in the SRSC method mainly for illustration purposes. Other clustering algorithms, such as TFCM++ (Gan and Huang, 2017), can also be used. The key idea in the SRSC method is to apply the clustering algorithm to a random sample drawn from the data set rather than to dataset itself, e.g., the entire portfolio.

If the sample size is small relative to the original portfolio, then the execution time is significantly reduced. The cost of such reduction is the risk of missing some clusters of the original portfolio in the random sample. Consider cluster $\mathcal{C}_j$ in the original portfolio and let n^* be the sample size in the first step of the SRSC method. In contrast to the SC method, where the number of subsets s is chosen by experience or by some ad hoc heuristic, the random sample size n^* in the SRSC can be chosen with theoretical justification. Theorem 1 provides a theoretical requirement for the sample size n^* in the SRSC method. The sample size n^* should be sufficiently large such that, for some $0 \leq f \leq 1$, the probability of drawing less than $f|\mathcal{C}_j|$ contracts from $\mathcal{C}_j$ in the random sample is small. As argued in Guha, Rastogi, and Shim (1998), it is reasonable to assume that the probability of missing cluster $\mathcal{C}_j$ is low if the random sample contains at least $f|\mathcal{C}_j|$ contracts from $\mathcal{C}_j$, for some $0 \le f \le 1$. Let X_j be the number of contracts belonging to $\mathcal{C}_j$ in the random sample of SRSC method; then with sample size n^* one can show (see, for example, Guha, Rastogi, and Shim 1998) that $\mu_j = \mathbb{E}[X_j] = \frac{n^*}{N}|\mathcal{C}_j|$.

Theorem 1. (Paraphrase of Theorem 1 in Guha, Rastogi, and Shim 1998). *For a cluster $\mathcal{C}_j$ and a given probability threshold δ, if the sample size n^* satisfies*

$$n^* \ge n_j^* := fN + \frac{N}{|\mathcal{C}_j|}\ln\left(\frac{1}{\delta}\right) + \frac{N}{|\mathcal{C}_j|}\sqrt{\left(\ln\left(\frac{1}{\delta}\right)\right)^2 + 2f|\mathcal{C}_j|\ln\left(\frac{1}{\delta}\right)}, \tag{3}$$

then the probability of drawing a small number of contracts from cluster $\mathcal{C}_j$ is small. Specifically,

$$Pr(X_j{<}f|\mathcal{C}_j|){<}\delta. \tag{4}$$

Proof. See Guha, Rastogi, and Shim (1998). □

Equation (3) is intuitive in that we need to sample more than a fraction f of the total portfolio for the sample to contain at least $f|\mathcal{C}_j|$ contracts belonging to cluster $\mathcal{C}_j$ with high probability. Denoting the smallest cluster in the portfolio by $\mathcal{C}_{j_{\min}}$, one can show that $n^*_{j_{\min}} = \max_j\{n_j^*\}$ where n_j^* are as defined in (3). This means that, with a sample size of at least $n^*_{j_{\min}}$, the sample contains at least $f|\mathcal{C}_j|$ contracts from any cluster $\mathcal{C}_j$ with high probability $1-\delta$.

Corollary 1 provides a similar sample size requirement under a more practical situation where we are only interested in clusters larger than a certain size and require a fixed number of contracts from the smallest cluster.

Corollary 1. *Consider a cluster $\mathcal{C}_j$ whose size is bounded below by $|\mathcal{C}_j| \ge \frac{N}{K\rho}$ for some $\rho{>}1$. For a given probability threshold δ, if the sample size n^* satisfies*

$$n^* \ge n_j^* := \xi K\rho + K\rho\ln\left(\frac{1}{\delta}\right) + K\rho\sqrt{\left(\ln\left(\frac{1}{\delta}\right)\right)^2 + 2\xi\ln\left(\frac{1}{\delta}\right)}, \tag{5}$$

then the probability of drawing less than ξ contracts from cluster $\mathcal{C}_j$ is small. Specifically,

$$Pr(X_j{<}\xi){<}\delta. \tag{6}$$

Proof. See Guha, Rastogi, and Shim (1998). □

The significance of Equation (5) is its *independence of the original portfolio size N*. This means that, if we believe (a) that there are 100 clusters in the portfolio (i.e., $K = 100$), (b) we are interested only in clusters whose size is greater than half of the average cluster size (i.e., $\rho = 2$), and (c) with 99% probability (i.e., $\delta = 0.01$) at least 10 contracts from the smallest cluster (i.e., $\xi = 10$) are sampled, then a sample size of about 5,000 suffices, regardless of the number of contracts in the portfolio. Last but not least, because the prototypes identified by the SRSC method are not likely to be similar, it does not require the nearest neighbor mapping cleanup step.

3.3. A Computational Comparison between the SC Heuristic and SRSC Method

The common goal for both the SC heuristic and the SRSC method is to identify K clusters in a portfolio of N contracts via clustering algorithms. The SC method applies the k-prototypes algorithm ***s times***, once in each subset to group N/s contracts into K/s clusters. We paraphrase the SRSC method for ease of comparison. The random sampling step in the SRSC method can be viewed as first randomly dividing the portfolio into s subsets and then randomly selecting one of the subsets, such that $n^* = N/s$. Then the SRSC method applies that k-prototypes algorithm ***only once*** to group N/s contracts into K clusters. We consider the per iteration computations required in these two cases.

Roughly speaking, the per iteration computations in a k-prototypes algorithm are a product of the number of pairwise distance calculations between contracts and cluster prototypes. Figure 3 depicts the pairwise distance calculations in the two methods.

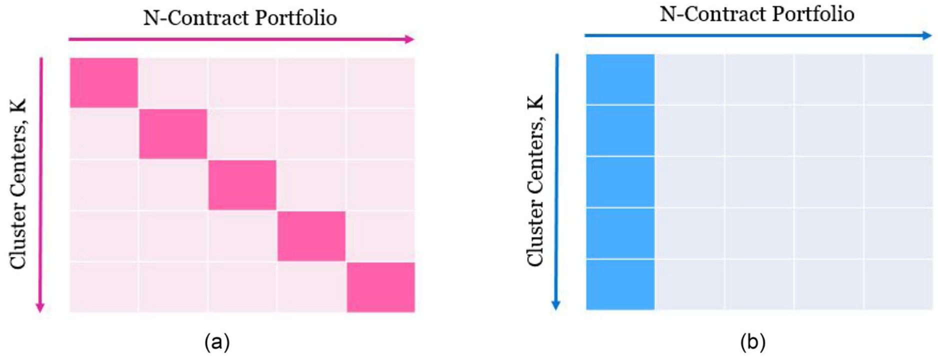

FIGURE 3. Per Iteration Pairwise Distance Calculations Required for the SC and SRSC Methods: (a) Computations of the SC Method and (b) Computations of the SRSC Method.

- For the SC heuristic, the *k*-prototypes algorithm is applied s times, as shown by the five diagonal blocks in Figure 3(a). In each iteration, the number of pairwise distance calculations equals $N/s \times K/s$. So the total number of distance calculations per iteration, summing over all s subsets, is NK/s.
- For the SRSC method, the *k*-prototypes algorithm is applied once, as shown in the first column in Figure 3(b). In each iteration, the number of pairwise distance calculations equals $N/s \times K = NK/s$.

Therefore, the two compressors require the roughly the same number of pairwise distance calculations per iteration (summing over s subsets for the SC heuristic).

In the above discussions, the number of iterations in the *k*-prototypes algorithms is omitted. Our numerical experiments in Section 6 show that our SRSC method has both shorter runtime and smaller WCSS compared to the SC heuristic under the same settings.

4. SIMULATOR: WITH GREAT COMPUTING POWER COMES GREAT CONTRACT COMPLEXITY

In this section we propose a simulator that optimally allocates a given computation budget among the representative VA contracts output by the compressor. For illustration we consider a simulation model that estimates the fair market value for each VA contract and for the entire portfolio; the main ideas behind this optimal allocation can also be applied to estimation of other quantities of interest such as Greeks.

We consider a fixed budget scenario where the available computing power is predetermined. In practice, this constraint may be expressed as a time limit or a fixed monetary amount that the user is willing to spend. Equal allocation of this simulation budget among all contracts is common in the literature (Gan 2013, 2015b; Gan and Lin 2015; Gan and Huang 2017; Hejazi, Jackson, and Gan 2017). However, this is often a suboptimal usage of the simulation budget; the variance of the resulting portfolio value could be high. Our proposal optimally utilizes the available simulation budget so that the variance of the estimated portfolio value is minimized.

Let $\bar{\boldsymbol{\mathcal{X}}} = \{\bar{\boldsymbol{x}}_1, ..., \bar{\boldsymbol{x}}_K\}$ be the set of representative VA contracts output by the compressor, as discussed in Section 3. Given a simulation model that generates independent and identically distributed (i.i.d.) replications of the fair market value (or other quantities of interest) of any VA contract, suppose that the value of the jth representative contract is estimated by the sample average of R_j i.i.d. replications; that is, $\widehat{Y}_j = \frac{1}{R_j}\sum_{i=1}^{R_j} V_j^{(i)}$. Then the value of the representative portfolio is given by $\widehat{Y}_p = \sum_{j=1}^{K} \widehat{Y}_j$. In addition, denote the variance for each i.i.d. replication of the jth contract by $\sigma_j := \mathbb{V}\mathrm{ar}[Y_j]$. Then the variance of the portfolio fair market value is given by

$$\sigma_p^2 := \mathbb{V}\mathrm{ar}[\widehat{Y}_p] = \mathbb{V}\mathrm{ar}\left[\sum_{j=1}^{K}\left(\frac{1}{R_j}\sum_{i=1}^{R_j} V_j^{(i)}\right)\right] = \sum_{j=1}^{K}\frac{\sigma_j^2}{R_j}. \tag{7}$$

Suppose that the given simulation budget allows a total of R i.i.d. replications across all contracts. The simulation budget allocation scheme $\{R_1, ..., R_K\}$ to minimizes the portfolio variance σ_p solves the following optimization problem:

$$\min_{R_1,\ldots,R_K}\left\{\sum_{i=1}^{K}\frac{\sigma_i^2}{R_i}\,\middle|\,\sum_{i=1}^{K}R_i=R\right\}. \tag{8}$$

Problem (8) is common in the design and analysis of computer experiments (Glasserman 2004). Proposition 1 presents the optimal budget allocation scheme in our setting.

Proposition 1. *If $R>0$ and $\sigma_j>0$ for some contract j, then the optimal simulation budget allocation $\{R_1,\ldots,R_K\}$ is given by*

$$R_j^*=\frac{\sigma_j}{\sum_{i=1}^{K}\sigma_i}R. \tag{9}$$

The minimized portfolio variance is

$$\sigma_{opt}^2=\frac{1}{R}\left(\sum_{j=1}^{K}\sigma_j\right)^2. \tag{10}$$

Proof. The Lagrangian function for (8) is $\mathcal{L}(R_1,\ldots,R_K,\lambda)=\sum_{j=1}^{K}\frac{\sigma_j^2}{R_j}+\lambda(\sum_{j=1}^{K}R_j-R)$. The first-order optimality condition $\frac{\partial\mathcal{L}}{\partial R_j}=-\frac{\sigma_j^2}{R_j^2}+\lambda=0$ indicates that $R_j^*=\sigma_j/\sqrt{\lambda}$ is proportional to σ_j. In addition, the first-order optimality condition $\partial\mathcal{L}/\partial\lambda=\sum_{j=1}^{K}R_j-R=0$ indicates that $\sum_{j=1}^{K}\sigma_j/\sqrt{\lambda}=R$ and so $\sqrt{\lambda}=\sum_{j=1}^{K}\sigma_j/R$. Therefore, the optimal simulation budget allocation is given by (9). Substituting (9) into (7) gives the minimized portfolio variance in (10). □

The optimal budget allocation (9) indicates that more computing power should be allocated to more complicated contracts. This is intuitive because the contracts with large variances have more adverse impacts on the accuracy of the portfolio value estimator. Therefore, shifting some computations from contracts with low variances to those with high variances can reduce the overall portfolio variance. We provide a more quantitative analysis of the optimal allocation scheme and the minimized portfolio variance below.

Consider a random variable σ_I that takes the value σ_j with probability $1/K$ for all $j=1,\ldots,K$. This is the standard deviation of a randomly selected contract in the representative portfolio. Its first and second moments are $\mathbb{E}[\sigma_I]=\sum_{j=1}^{K}\sigma_j/K$ and $\mathbb{E}[\sigma_I^2]=\sum_{j=1}^{K}\sigma_j^2/K$. To appreciate the minimized variance (10), we compare it to the portfolio variance resulting from the equal allocation of simulation budget; that is, each contract value is estimated with R/K i.i.d. replications. The portfolio variance under such equal allocation is $\sigma_{eq}^2=\sum_{i=1}^{K}\frac{\sigma_i^2}{R/K}$, which is larger than the minimized variance by

$$\sigma_{eq}^2-\sigma_{opt}^2=\frac{K^2}{R}\left[\frac{\sum_{i=1}^{K}\sigma_i^2}{K}-\frac{\sum_{i=1}^{K}\sigma_i}{K}\frac{\sum_{i=1}^{K}\sigma_i}{K}\right]=\frac{K^2}{R}\left[\mathbb{E}\left[\sigma_I^2\right]-\left(\mathbb{E}[\sigma_I]\right)^2\right]=\frac{K^2}{R}\mathbb{V}\mathrm{ar}[\sigma_I].$$

Because the representative contracts are the centers of distinct clusters in the compressor, they are likely to be significantly different from each other, resulting in vastly different variances. Consequently, the variance of σ_I, $\mathbb{V}\mathrm{ar}[\sigma_I]$, is likely to be large. As shown in our numerical experiments, the variance reduction achieved by optimally allocating the given simulation budget is significant.

In practice, the per sample standard deviations σ_j are unknown, so (9) could not be implemented. In such cases, we suggest a two-stage approach to allocate the given simulation budget:

- Stage I: Run a pilot simulation using a fraction (e.g., 10%) of the simulation budget to estimate the standard deviations $\widehat{\sigma}_j$. The pilot simulation budget is equally allocated among the contracts, say R_0 replications for each contract.
- Stage II: Let $\widehat{R}_j^*$ be the optimal allocation (9) with σ_j replaced by the estimated $\widehat{\sigma}_j$. Run additional simulations; that is, $\max\{0,\widehat{R}_j^*-R_0\}$ replications for the jth contract.

The additional simulations in Stage II aims to allocate the remaining simulation budget in a way that the overall allocation to the jth contract is $\max\{R_0,\widehat{R}_j^*\}$; the pilot budget should be small, so it is rare for R_0 to be the maximum. The specific simulation model for VA contracts in our numerical experiment is identical to that in Gan (2013) and thus restatement is omitted.

As a final remark, our proposed optimal simulation budget allocation scheme (8) minimizes the variance of the representative portfolio value so it could be suboptimal for estimating the original portfolio value. Nevertheless, our numerical experiments show that our simulator achieves far higher accuracy than those with equal allocations. We envision to developing further improvements to the simulator in the future.

5. PREDICTOR: ALL METAMODELS ARE WRONG, SOME ARE USEFUL

With the outputs of the compressor and the simulator—that is, the representative VA contracts $\bar{\boldsymbol{x}}_1, ..., \bar{\boldsymbol{x}}_K$ and their estimated values $\widehat{Y}_1, ..., \widehat{Y}_K$—the goal of the predictor is to use the resulting dataset $\mathcal{D} := \{(\bar{\boldsymbol{x}}_1, \widehat{Y}_1), ..., (\bar{\boldsymbol{x}}_K, \widehat{Y}_K)\}$ to predict the values of the original VA contracts and/or that of the original portfolio. Specifically, the predictor has two main tasks:

1. ***Calibration*** of a predictive model. Use the dataset $\mathcal{D}$ to calibrate the model

$$Y(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\beta}), \tag{11}$$

   where the weight vector $\boldsymbol{\beta}$ denotes the model parameters to be calibrated. Denote the calibrated parameters and predictive model by $\tilde{\boldsymbol{\beta}}$ and $\tilde{Y}(\boldsymbol{x})$, respectively.
2. ***Prediction*** using the calibrated model: Use the calibrated predictive model to predict market values $\tilde{Y}(\boldsymbol{x}_1), ..., \tilde{Y}(\boldsymbol{x}_N)$ for the original contracts and/or the original portfolio value $\tilde{Y}(\boldsymbol{\mathcal{X}}) = \sum_{i=1}^{N} \tilde{Y}(\boldsymbol{x}_i)$.

These two steps are known as metamodeling (see Barton and Meckesheimer [2006] for a review) in the design and analysis of computer experiments; metamodels are models of the simulation model of interest.

In this section we propose, but do not advocate, a simple benchmark metamodel, called the cluster size multiple (CSM) metamodel, which requires minimal computational requirements but has reasonable accuracy. The CSM metamodel is too simplistic for practical applications but can serve as a worst-case benchmark for the development of new predictors within the three-component valuation framework.

The main idea of the CSM metamodel is straightforward: Because the K representative contracts are the cluster prototypes, one should use the simulated value of each prototype to represent the values of all contracts in the associated cluster. Mathematically, the CSM prediction for any contract $\boldsymbol{x}$ is given by

$$\tilde{Y}^{CSM}(\boldsymbol{x}) = \widehat{Y}_{j^*}, \qquad \text{where } j^* = \arg\min_{j}\{d(\boldsymbol{x}, \boldsymbol{x}_j) : j = 1, ..., K\}. \tag{12}$$

Therefore, the CSM prediction contract value for any given contract $\boldsymbol{x}$ is the simulated value for the representative contract $\boldsymbol{x}_{j^*}$ that is closest to $\boldsymbol{x}$. Prediction for all original contracts and thus for the original portfolio requires about $N \times K$ pairwise distance calculations. This computational cost is considerably smaller than almost any other metamodeling approach[1], which often requires matrix inversions (e.g., regression, kriging), solving a linear system of equations (e.g., radial basis functions), or iterative calibration algorithms (e.g., neural networks, random forest).

Despite simplicity, the CSM metamodel (12) is fairly intuitive and takes advantage of a well-designed compressor. Specifically, with a well-designed compressor, all contracts within the same clusters have similar attributes and so are likely to have similar values. Such constant prediction within a cluster is admittedly simplistic and could result in large estimation error for individual contracts. Nevertheless, because every representative contract is a cluster prototype that lies in the "center" of the cluster, the estimation errors within a cluster are likely to have cancellation effect, resulting in a reasonable estimation for the portfolio value.

For illustration we compare the CSM model to kriging (Oliver and Webster 1990; Kleijnen 2009), which is a state-of-the-art metamodel in the relevant literature (Gan 2013; Gan and Lin 2015). Kriging models the simulated value of any given VA contract $\boldsymbol{x}$ as

$$Y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})\boldsymbol{\beta} + \mathtt{M}(\boldsymbol{x}), \tag{13}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is a vector of known functions of $\boldsymbol{x}$, $\boldsymbol{\beta}$ is a parameter vector of compatible dimension, and $\mathtt{M}(\boldsymbol{x})$ is a realization of a zero mean random field (Staum 2009). Specifically, $\mathtt{M}$ is regarded as a random sample from a space of functions that exhibit

[1] The inverse distance weighting (Hejazi, Jackson, and Gan 2017) method has similar computational costs. However, we found that its prediction accuracy is very low, which is consistent with findings in Hejazi, Jackson, and Gan (2017). Therefore, we do not consider this method any further.

TABLE 1
VA Contract Specification

Attributes	Values	Distribution
Guarantee type	{GMDB only, GMDB + GMWB}	50%, 50%
Gender	{Male, Female}	50%, 50%
Age	$\{20, 21, ..., 60\}$	Discrete uniform
Premium	$[10,000, 50,000]$	Continuous uniform
GMWB withdrawal rate	$\{4\%, 5\%, 6\%, 7\%, 8\%\}$	Discrete uniform
Maturity	$\{10, 11, ..., 25\}$	Discrete uniform

spatial correlation. Such spatial correlation is often model by a covariance function $\omega(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ for any two VA contracts $\boldsymbol{x}, \boldsymbol{x}'$; ω is known up to the parameter $\boldsymbol{\theta}$. The functional form of ω is selected by the user and in general both $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are to be calibrated. In general, the kriging prediction for VA contract $\boldsymbol{x}$ is

$$\tilde{Y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})\tilde{\boldsymbol{\beta}} + \boldsymbol{\omega}(\boldsymbol{x})\Sigma^{-1}(\widehat{\boldsymbol{Y}} - \boldsymbol{F}\boldsymbol{\beta}), \tag{14}$$

where $\boldsymbol{\omega}(\boldsymbol{x})$ is a row vector whose ith element is $\omega(\boldsymbol{x}, \bar{\boldsymbol{x}}_i), \Sigma_{ij} = \omega(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j), \widehat{\boldsymbol{Y}} = [\widehat{Y}_1, ..., \widehat{Y}_K]^\top$, and $\boldsymbol{F}$ is a matrix whose ith row is $\boldsymbol{F}_{i\cdot} = \boldsymbol{f}(\boldsymbol{x}_i)$. Gan (2013) and Gan and Lin (2015) used a particular spherical covariance function that has straightforward calibration, which is a special case of the general kriging methodology.

6. NUMERICAL EXPERIMENTS

In this section, we present some test results to demonstrate the efficiency of the proposed three-component valuation framework. We will illustrate the efficiency of the compressor, simulator, and predictor in isolation as well as in combination. The accuracy and speed of our proposals are compared to those in other state-of-the-art methodologies. All numerical experiments presented in this section were run using Intel Xeon CPU E5-2650 v2 @ 2.60GHz.

6.1. Synthetic VA Portfolio

We generate synthetic VA portfolios whose attributes are summarized in Table 1. This specification is identical to those in Gan (2013) and Gan and Lin (2015); this was the state-of-the-art literature at the time of our research. Depending on the specific purposes of illustration, portfolios of different sizes are generated in different experiments.

6.2. Experiment 1: Efficient Compressor

In this section we show that our SRSC compressor is a reliable method to select representative contracts in a given VA portfolio. In particular, the SRSC compressor takes a similar amount of CPU time as the SC heuristic but achieves accuracy similar to that of the k-prototypes algorithm being applied to the entire VA portfolio.

In this experiment we generate $N = 10,000$ synthetic VA contracts according to Table 1 and aim to identify $K = 50$ representative contracts. Both N and K are set smaller than their practical values so that the k-prototypes algorithm can be applied to the entire portfolio with reasonable computations. We will examine examples with larger VA portfolios later.

For the each synthetic VA portfolio, three compressors are examined:

- Apply the k-prototypes algorithm directly to the entire portfolio. Though this compressor is infeasible for large VA portfolios in practice, it merely serves as a benchmark in this small illustrative example.
- Apply our SRSC method with sample size $n^* = 1,000$. The sample size is chosen for ease of comparison: It coincides with the size of each subset in the SC heuristic. The computations for the SC and SRSC compressors should be similar in this case, as discussed in Section 3.3. Duplicates are removed, so only distinct representative VA contracts are selected.
- Apply the SC heuristic with $s = 10$ subsets. In this case, five prototypes are identified in each subset of size 1,000. The number of prototypes in each subset is similar to the value suggested by Gan (2013). The prototypes in all 10 subsets

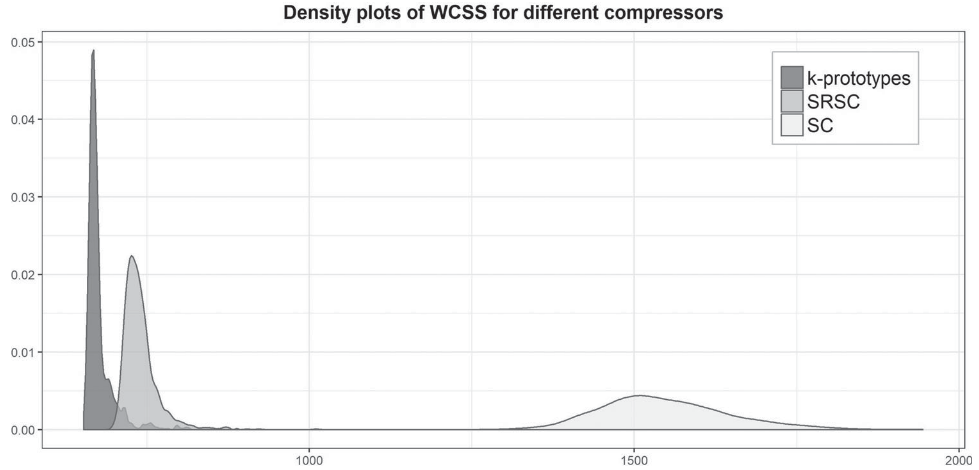

FIGURE 4. Density Plots of 1,000 Within-Cluster Sum of Squares of Different Compressors. *Note*: The goal is to identify 50 clusters in a VA portfolio of 10,000 contracts.

TABLE 2
Summary Statistics of Three Compressors in 1,000 Repeated Experiments

	k-Prototypes	SRSC	SC
Average runtime (in seconds)	7.39	0.79	0.81
Average WCSS	676	740	1,546
Average distinct representative contracts	50	50	35

are then mapped to the nearest neighbor in the original portfolio. Duplicates are removed, so only distinct representative VA contracts are selected.

To assess the performance of the above compressors, we repeated the above experiment 1,000 times, each of which used a synthetic VA portfolio that was independent of all other experiments.

Figure 4 depicts the density plots of the WCSS of different compressors in the 1,000 independent repeated experiments. Table 2 provides a more quantitative summary of the same experiments. We see from the leftmost density plot in Figure 4 that the *k*-prototypes compressor achieves the smallest WCSS; this is the ideal case benchmark. The density plot of WCSS of the SRSC compressor is close to that of the *k*-prototypes compressor; the density plot of WCSS of the SC compressor is far from the other two. More over, we see from the density plot that the value and the variability of WCSS for the SC compressor are both larger than those of the SRSC compressor. The first row of Table 2 shows that the SRSC and SC compressors have similar runtime, which is consistent with discussions in Section 3.3. The runtime of the SRSC and SC compressors is approximately one tenth of the *k*-prototypes compressor, as expected, because the original portfolio is divided into 10 subsets. We see from the second row of Table 2 that, compared to the WSCC of the *k*-prototype compressor, the SRSC compressor increases the WCSS only by less than 10%. In contrast, compared to the WSCC of the *k*-prototype compressor, the SC compressor's WCSS is more than twice as much. The last row in Table 2 illustrates the redundancy of the SC compressor as discussed at the end of Section 3.1. Though the SC compressor wants to identify 50 clusters, it only identifies 35 distinct prototypes on average. This redundancy reduces the number of distinct representative contracts and increases the WCSS.

6.3. Experiment 2: Efficient Simulator

In this section we demonstrate the ease and effectiveness of implementing the optimal allocation scheme (8). In particular, we continue with the small illustrative example: We aim to optimally allocate a fixed budget among 50 representative contracts such that the variance of the estimated representative portfolio value is minimized.

In this experiment, we assume that the total simulation budget is 50,000 replications. This means that each VA contract is simulated 1,000 replications in an equal allocation scheme. To optimally allocate the simulation budget, we first equally

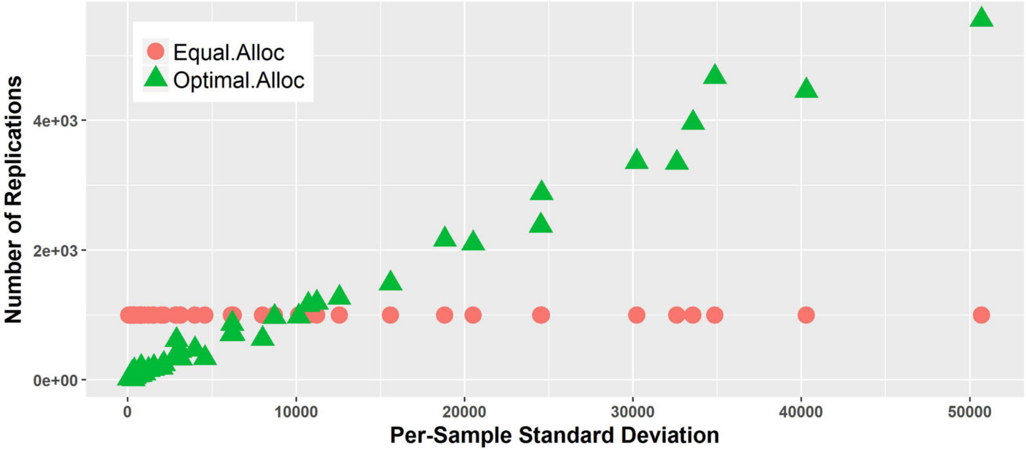

FIGURE 5. Two-Stage Allocation Scheme $\widehat{R}_j^*$ vs. per Sample Standard Deviation σj.

allocate 10% of the simulation budget—that is, $R_0 = 100$ replications per contract—to estimate the per sample standard deviations $\widehat{\sigma}_j$. Then the remaining 90% of the budget is allocated to the contracts. Specifically, the jth contract will run an additional $\max\{0, \widehat{R}_j^* - R_0\}$, replications, where $\widehat{R}_j^*$ is the optimal allocation scheme (8) with σ_j replaced by the estimated $\widehat{\sigma}_j$.

Figure 5 depicts the relationship between the two-stage allocation scheme $\widehat{R}_j^*$ and an accurately estimated per sample standard deviation σ_j. The number of replications $\widehat{R}_j^*$ on the y-axis is calculated based on estimated $\widehat{\sigma}_j$ using 10% of the simulation budget. The per sample standard deviation σ_j on the x-axis is estimated using the entire simulation budget. Each point in the figure represents one VA contract; different budget allocations schemes are labeled by different shapes and colors. The dots form a horizontal line if the budget is equally allocated among all contracts. According the Proposition 1, if the true standard deviations σ_j were available, then R_j^* would be proportional to σ_j. With some variability, the triangles form a straight line in Figure 5. This shows that the desired proportionality is approximately achieved using only 10% of the simulation budget to estimate $\widehat{\sigma}_j$. In our experiments, the two-stage allocation scheme achieved a 65% variance reduction compared to equal allocation with the same simulation.

6.4. Experiment 3: Putting It All Together

In this experiment we consider a more realistic example with a larger VA portfolio and more clusters. Specifically, we consider $J = 100$ independent synthetic VA portfolios, each having size $N = 100{,}000$, and then estimate their fair market values by different simulation procedures. For each synthetic VA portfolio, we first run a benchmark Monte Carlo simulation with 1, 000 independent replications to estimate the value of each contract; these estimates are then summed to produce the estimated portfolio value. The results of this benchmark experiment are then used to assess the performances of other simulation procedures in the same three-component valuation framework. In the compressor, we will consider two different numbers of representative contracts; for example, $K = 100$ and $K = 500$. In the simulator, the simulation budget is $R = 1,000 \times K$ replications. Therefore, in an equal allocation scheme each contract in the representative portfolio is estimated by averaging 1,000 independent sample paths. These settings are similar to those in Gan (2013) for ease of comparisons.

6.4.1. Assessing Accuracies

In the literature, it is customary to consider the percentage difference between the portfolio value estimated by the proposed procedure and that by a benchmark procedure, which is usually a time-consuming Monte Carlo simulation. For example, table 5 in Gan (2013) and table 3 in Hejazi, Jackson, and Gan (2017) depict such performance measures. In addition, to the best of our knowledge, such comparison is done for only one large synthetic VA portfolio (e.g., 200,000 contracts in Gan [2013] and 100,00 contracts in Hejazi, Jackson, and Gan [2017]).

There are two shortcomings when accuracy is assessed in this manner:

(i) Portfolio value estimate as a sum of the individual contract value estimates, especially in a large portfolio, could have a so-called netting effect. This means that the estimation error in contract value may be canceled out when aggregated into a portfolio value estimate. Though the main objective of this article and the aforementioned studies is to accurate

TABLE 3
Summary of Performance Measures of the Three Simulation Procedures in Experiment 3

		Gan2013	OptDesign & Kriging	OptDesign & CSM
	Compressor	SC	SRSC	SRSC
	Simulator	Equal allocation	TWo-stage allocation	Two-stage allocation
	Predictor	Kriging	Kriging	CSM
$K = 100$	CMRE (SE)	48% (0.16%)	25% (0.12%)	36% (0.12%)
	PMRE (SE)	6.4% (0.29%)	1.7% (0.11%)	2.7% (0.16%)
	Runtime (in seconds)	16/0.6/20	16/0.8/20	16/0.8/1.6
$K = 500$	CMRE (SE)	21% (0.08%)	13% (0.04%)	25% (0.03%)
	PMRE (SE)	1.8% (0.1%)	1.1% (0.07%)	0.8% (0.06%)
	Runtime (in seconds)	83/3/27	41/4/28	41/4/6

Note: Runtime for the benchmark Morte Carlo simulation is about 730 s.

estimate the value of large VA portfolios, it is an additional merit if the proposed simulation procedure can provide accurate estimates for the individual contracts too.

(ii) Numerical conclusions based on one large VA portfolio can have limited generality. In addition, the robustness of the proposed algorithm cannot be assessed unless the same procedure is applied to different independent portfolios.

We assess both the portfolio- and contract-level accuracies of different simulation procedures that are applied to multiple independent large VA portfolios. In particular, the portfolio-level accuracy is assessed by the portfolio mean relative error (PMRE):

$$PMRE = \frac{1}{J}\sum_{j=1}^{J}\frac{|\widehat{\mu}_p^{(j)} - \mu_p^{(j)}|}{\mu_p^{(j)}}, \tag{15}$$

where $\mu_p^{(j)}$ is the j-th synthetic portfolio value estimated by the benchmark Monte Carlo experiment and $\widehat{\mu}_p^{(j)}$ is the corresponding portfolio value estimated by one other simulation procedure of interest. Similarly, the contract-level accuracy is assessed by the contract mean relative error (CMRE):

$$CMRE = \frac{1}{J}\sum_{j=1}^{J}\left[\frac{\sum_{i=1}^{N}|\widehat{\mu}_i^{(j)} - \mu_i^{(j)}|}{\mu_p^{(j)}}\right], \tag{16}$$

where $\mu_i^{(j)}$ is the i-th contract value in the jth synthetic portfolio estimated by the benchmark Monte Carlo experiment and $\widehat{\mu}_i^{(j)}$ is the corresponding contract value estimated by one other simulation procedure of interest.

6.4.2. Main Results

To examine the performance of our new designs, we compare and contrast three simulation procedures:

- Gan (2013) uses the SC heuristic as the compressor, equal allocation in the simulator, and kriging as its predictor. This identical to the simulation procedure in Gan (2013); hence the name.
- `OptDesign & Kriging` uses the SRSC method as the compressor, two-stage allocation in the simulator, and kriging as its predictor. Comparing `OptDesign & Kriging` to `Gan2013` showcases the advantages of our new compressor and two-stage simulation budget allocation.
- `OptDesign & CSM` uses the same compressor and simulator as `OptDesign & Kriging` but CSM as the predictor. Comparing `OptDesign & CSM` to `OptDesign & Kriging` reveals some interesting future research questions.

Note that the kriging predictor can be implemented in two different ways: The portfolio value can be estimated efficiently by solving one $K \times K$ system of linear equations, without any knowledge of the individual contract value. Estimating individual contract values requires solving similar $K \times K$ systems of linear equations N times and thus is more time consuming. We ran both implementations to access both the portfolio- and contract-level accuracies. The runtime for the efficient portfolio value estimation is reported.

Table 3 summarizes the main results of this experiment. The average CMREs and PMREs over $J = 100$ independent synthetic VA portfolios are shown along with their standard errors. The runtimes for the three components in each simulation procedures are shown separately. We first make some general observations that are not specific to the simulation procedures. Comparing the accuracy measures between the bottom two panels—for example, $K = 100$ vs. $K = 500$—we see that the accuracies improve while runtime increases as the compressor identifies more clusters. This is because the original portfolio can be better represented by more contracts but more computations are required; this is consistent with the observations in Gan (2013) and Liu and Tan (2017). Comparing the contract mean relative errors (CMREs) with the corresponding portfolio mean relative errors (PMREs), we see that in general portfolio value estimations are more accurate than contract value estimations; this asserts the "netting" effect in portfolio estimation.

Comparing `OptDesign & Kriging` to `Gan2013` we see that our new compressor (SRSC) and simulator (two-stage allocation) significantly improves the accuracy of estimation. Specifically, the CMRE (PMRE) is reduced by 48% (38%, respectively) when $K = 100$ and by 73% (39%) when $K = 500$. Then runtime for `OptDesign & Kriging` is no longer, and sometimes much shorter, than that for Gan2013. This means that the error reductions are achieved by better designs, not by additional computations.

Comparing `OptDesign & CSM` to `OptDesign & Kriging` and `Gan2013` we see that, despite its simplicity, the `CSM` metamodel can be a reasonable predictor in some cases. When $K = 100$, `OptDesign & CSM` is not as accurate as `OptDesign & Kriging` but more accurate than `Gan2013`. The first comparison shows that interpolation methods can improve prediction quality, whereas the second comparison shows that a well-designed compressor and simulator is no less important than a good predictor. When $K = 500$, we see that `OptDesign & Kriging` has accurate estimations at both the contract level and the portfolio level. It is surprising to observe that `OptDesign & CSM` has the worst contract level accuracy but the best portfolio-level accuracy. This suggest a significant "netting" effect in this method.

As discussed in Section 5, we do not advocate the CSM metamodel in practical applications. Nevertheless, it should serve as a worst-case benchmark for new metamodel proposals: For a new predictor proposal to be valuable, it should at least be more accurate than the CSM metamodel. Examination of a suitable or even provably optimal metamodel for valuating large VA portfolios remains a subject for future research.

7. CONCLUDING REMARKS AND FUTURE RESEARCH

In this article we propose a new method for valuation of large VA portfolios. The three components of the general valuation framework are analyzed:

1. The SRSC compressor identifies a small set of representative contracts and has a provable performance guarantee as shown in Proposition 1.
2. The two-stage simulation budget allocation is based on the optimal allocation scheme (8) and is shown to have satisfactory performance.
3. The CSM metamodel may be too simplistic for practical applications but can serve as a worst-case benchmark for the development of new predictors within the three-component valuation framework.

With similar or fewer computations, our numerical results show that the proposed method has significant higher accuracy than state-of-the-art alternatives.

We identify two specific venues for further developments of the proposed method. Firstly, we will further explore the synergies among different components when designing efficient experiments. For example, the representative contracts are often distinct as a result of a clustering-based compressor and such a distinction results in great variance in the optimal allocation scheme (8). As another example, the CSM metamodel only has intuitive justifications because, after clustering, all VA contracts within the same cluster are expected to be similar. In the future, we hope to take advantage of such synergy when designing an efficient predictor. For example, the simulator may use the membership information from the compressor to enhance estimation. In addition, one can consider metamodels such as stochastic kriging (Ankenman, Nelson, and Staum 2010), which uses more information from the simulator than just the sample mean. Secondly, as discussed in Section 4, the optimal allocation in the simulator minimizes the variance for estimating the representative portfolio value, not the original portfolio value. In the future we will consider addressing such inconsistency by formulating and solving a new budget allocation problem.

There are several limitations and weaknesses to the proposed simulation procedure and the valuation framework in practical applications. Firstly, the three-component valuation framework aims to provide accurate portfolio value estimations but is not suitable for estimating individual contract values and sometimes is not even feasible for the latter purpose. Secondly, further searches for a suitable predictor are required. The CSM predictor should only be used a worst-case benchmark; one should search for other predictors with superior performances.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions.

FUNDING

We acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC)10.13039/501100000038, funding reference number 03755.

ORCID

Ben Mingbin Feng http://orcid.org/0000-0002-9748-6435

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2010. Stochastic kriging for simulation metamodeling. *Operations Research* 58: 371–82. doi:10.1287/opre.1090.0754

Arthur, D., and S. Vassilvitskii. 2007. *k*-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, ed. N. Bansal, K. R. Pruhs, and C. Stein, 1027–35. New Orleans, LA: Society for Industrial and Applied Mathematics.

Barton, R. R., and M. Meckesheimer. 2006. *Metamodel-based simulation optimization*. Volume 13 of *Handbooks in operations research and management science*. Amsterdam: Elsevier.

Bezdek, J. C. 2013. *Pattern recognition with fuzzy objective function algorithms*. New York, NY: Springer Science & Business Media.

Broadie, M., Y. Du, and C. C. Moallemi. 2011. Efficient risk estimation via nested sequential simulation. *Management Science* 57: 1172–94. doi:10.1287/mnsc.1110.1330

Chen, C. H., D. He, M. Fu, and L. H. Lee. 2008. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing* 20: 579–95. doi:10.1287/ijoc.1080.0268

Chen, C. H., J. Lin, E. Ycesan, and S. E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems* 10: 251–70.

Dunn, J. C. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3: 32–57. doi:10.1080/01969727308546046

Gan, G. 2013. Application of data clustering and machine learning in variable annuity valuation. *Insurance: Mathematics and Economics* 53: 795–801. 2013.09.021. doi:10.1016/j.insmatheco

Gan, G. 2015a. Application of metamodeling to the valuation of large variable annuity portfolios. In *Proceedings of the 2015 Winter Simulation Conference*, ed. L. Yilmaz, W. K. Chan, I.-C. Moon, T. M. K. Roeder, C. M. Macal, and M. D. Rossetti, 1103–14. Piscataway, NJ: IEEE Press.

Gan, G. 2015b. A multi-asset monte carlo simulation model for the valuation of variable annuities. In *Proceedings of the 2015 Winter Simulation Conference*, ed. L. Yilmaz, W. K. Chan, I.-C. Moon, T. M. K. Roeder, C. M. Macal, and M. D. Rossetti, 3162–63. Piscataway, NJ: IEEE Press.

Gan, G., and J. Huang. 2017. A data mining framework for valuing large portfolios of variable annuities. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ed. S. Matwin, S. Yu, and F. Farooq, 1467–75. New York, NY: ACM.

Gan, G., Q. Lan, and S. Sima. 2016. Scalable clustering by truncated fuzzy c-means. *Big Data and Information Analytics* 1: 247–59. doi:10.3934/bdia.2016007

Gan, G., and X.S. Lin. 2015. Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance: Mathematics and Economics* 62: 138–50. doi:10.1016/j.insmatheco.2015.02.007

Gan, G., and E.A. Valdez. 2018. Regression modeling for the valuation of large variable annuity portfolios. *North American Actuarial Journal* 22: 40–54. doi:10.1080/10920277.2017.1366863

Glasserman, P. 2004. *Monte Carlo methods in financial engineering*. New York, NY: Springer.

Guha, S., R. Rastogi, and K. Shim. 1998. Cure: An efficient clustering algorithm for large databases. *ACM Sigmod Record* 27: 73–84.

Hardy, M. 2003. *Investment guarantees: modeling and risk management for equity-linked life insurance*. Vol. 215. Hoboken, NJ: John Wiley & Sons.

Hejazi, S. A., K. R. Jackson, and G. Gan. 2017. A spatial interpolation framework for efficient valuation of large portfolios of variable annuities. *Quantitative Finance and Economics* 1: 125–44. doi:10.3934/QFE.2017.2.125

Huang, Z. 1998. Extensions to the *k*-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2: 283–304.

Kleijnen, J. P. C., 2009. Kriging metamodeling in simulation: A review. *European Journal of Operational Research* 192: 707–16. doi:10.1016/j.ejor.2007.10.013

Lee, L. H., E. P. Chew, S. Teng, and D. Goldsman. 2004. Optimal computing budget allocation for multi-objective simulation models. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 586–594.

Liu, K., and K. S. Tan. 2017. Real-time valuation of large variable annuity portfolios: A green mesh approach. Working paper.

Marshall, C., M. Hardy, and D. Saunders. 2010. Valuation of a guaranteed minimum income benefit. *North American Actuarial Journal* 14: 38–58. doi:10.1080/10920277.2010.10597576

McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21: 239–45. doi:10.2307/1268522

Niederreiter, H. 1992. *Random number generation and quasi-Monte Carlo methods*. Philadelphia, PA: SIAM.

Oliver, M. A., and R. Webster. 1990. Kriging: A method of interpolation for geographical information systems. *International Journal of Geographical Information Systems* 4: 313–32.

Staum, J. 2009. Better simulation metamodeling: The why, what, and how of stochastic kriging. In *Proceedings of the 2015 Winter Simulation Conference*, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 119–133. Piscataway, NJ: IEEE Press.

Vadiveloo, J. 2012. Replicated stratified sampling: A new financial modeling option. *Actuarial Research Clearing House* 1: 1–4.

Xu, W., Y. Chen, C. Coleman, and T. F. Coleman. 2018. Moment matching machine learning methods for risk management of large variable annuity portfolios. *Journal of Economic Dynamics and Control* 87: 1–20. doi:10.1016/j.jedc.2017.11.002

Discussions on this article can be submitted until January 1, 2021. The authors reserve the right to reply to any discussion. Please see the Instructions for Authors found online at http://www.tandfonline.com/uaaj for submission instructions.