Global optimization for model points automatic selection in life insurance portfolios

Ana M. Ferreiro[a], Enrico Ferri[a], José A. García-Rodríguez[a], Carlos Vázquez[a]

[a] *Department of Mathematics and CITIC, University of A Coruña, Campus Elviña s/n, 15071–A Coruña, Spain*

Abstract

This work deals with the automatic selection of model points portfolios of life insurance policies that reproduce the original portfolio, in the sense that they retain the market risk properties of the initial portfolio. In order to achieve this goal, we first propose a risk functional that incorporates the uncertain evolution of forward LIBOR rates to the portfolios of life insurance policies.

Once we have chosen the proper risk functional, the problem of finding the model points of the replicating portfolio is formulated as a problem of minimizing the distance between the original and the target model points portfolios, under the measure given by the proposed risk functional. In this way, a high dimensional global optimization problem arises and a suitable hybrid global optimization algorithm is proposed for the efficient solution of this problem.

Several examples illustrate the performance of parallel computing tools for the evaluation of the risk functional, as well as the efficiency of the hybrid Basin Hopping optimization algorithm to obtain the model points portfolio.

Keywords: Model points portfolio; LIBOR market model; risk functional; hybrid optimization algorithms; Monte Carlo simulation

1. Introduction

A very relevant problem in life insurance companies is the so called Asset Liability Management (ALM), which consists of the joint management of assets and liabilities portfolios, to ensure the future wealth and profitability of the insurance company (for example, see [6, 10] and the references therein). For this purpose, it is important to compute the joint projection of the future cashflows of both portfolios, which can be done numerically by using Monte Carlo algorithms. Computing these projections with the original portfolios, with a high number of policies (hundreds of thousands) in the liabilities side, can lead to

Email addresses: aferreiro@udc.es (Ana M. Ferreiro), enrico.ferri@udc.es (Enrico Ferri), jagrodriguez@udc.es (José A. García-Rodríguez), carlosv@udc.es (Carlos Vázquez)

a highly demanding computational time task or even prohibitive in reasonable time schedules.

Having this in view, insurance companies are allowed to compute these projections by replacing any homogeneous group of policies with some suitable representative contracts, usually known as the related model points. This procedure is permitted under suitable conditions in such a way that the inherent risk structure of the original portfolio is not misrepresented. We refer to the document [5] for further details. For example, in [4], this models points representation is performed by controlling the impact on Tail-Var and related risk measures.

This new portfolio could be understood as a compressed version of the original one that should retain the same risk properties as the original portfolio, once we have chosen a proper function to measure the associated risk in this replacement. The problem can be framed in the domain of portfolio representation, where the risk functional gauges the averaged differences between the values of original and model points portfolio, the last one taken in a specific set of portfolios [9]. The risk measure is usually gauged in terms of the fluctuation of some underlying stochastic risk factors inside a time horizon. The interest rate term structure and the mortality trend of the population are two of the main factors when dealing with the valuation and the risk management of life insurance products [12]. In the present paper, as a first attempt to introduce the proposed mathematical and computational methodologies for the automatic selection of model points, we just consider the uncertain future fluctuations of interest rates by considering a LIBOR market model for forward LIBOR rates. Additional risk factors could be considered in future works, as well as the dependence among them when more than one risk factor is considered.

However, even after choosing the right risk measure functional, finding the structure of the model points portfolio regarding this risk measure can result into a very hard computational problem. Finding the new model points portfolio can be posed as an optimization problem, where we have to minimize the distance between both portfolios in terms of the chosen risk measure.

The main goal of this article is to develop an original technique for the automatic generation of the model points portfolio. This objective mainly involves two tasks. First, we have to choose/fix and build a correct risk measure attending to market models. In our case we will build this measure using the LIBOR market model for forward interest rates (see [1], for details). Secondly, once we have formulated the problem in terms of a distance minimization based on this measure, we need to numerically compute this objective function in a highly efficient way, as next we have to build a numerical optimization method for the minimization of this objective function. As we will see, the resulting optimization problem is a global optimization one, so that efficient global optimization algorithms have to be built. More precisely, the algorithms need to be fast and robust, so that they won't get stuck in local minima.

The structure of the paper is as follows. In Section 2, we recall the LIBOR market model that will be one of the building blocks of our risk functional. In Section 3, we describe the model points portfolio selection problem. In Section

4, we present the definition of the LIBOR based risk measure. In Section 5, we show the numerical discretization of the problem. In Section 6, we show some numerical examples: first, we show the parallel performance of the cost function implementation; secondly, we show some examples with known solution to validate the convergence of the optimization algorithm to the analytical solutions; and finally we show an application to a real portfolio without analytical solution.

2. Model setup. LIBOR Market Model

In this section, following [1] we describe the risk-free dynamics of the discounted bond price, when considering the LIBOR Market Model governing the time evolution of the forward rates.

Libor rates. Let N be a positive integer and hence define the finite set $\mathcal{T} = \{T_0, T_1, ..., T_N\}$ to be a fixed tenor structure, with $T_0 = 1$ and $T_0 < T_1 < ... < T_N$, so that T_n corresponds to a specific maturity time. We write $\tau_n \triangleq T_n - T_{n-1}$, for $n = 1, ..., N$, to denote the corresponding accruals. Moreover, we set $\mathcal{I} \triangleq (0, 1)$ to be the unit interval on the real line, which corresponds to the period of one year.

Here and in the sequel, we shall write $B_n(t)$ to denote the risk-neutral discounted price at time $t \in \mathcal{I}$ of a (zero-coupon) bond expiring at the tenor date T_n, for any $n = 0, ..., N$. Moreover, we shall denote by $F_n(t)$ the value at time $t \in \mathcal{I}$ of the (simply-compounded) LIBOR forward rate associated to the accrual period $(T_{n-1}, T_n]$, for $n = 1, ..., N$. In this respect, we recall that $F_n(t)$ is defined in such a way that the following condition is met:

$$B_n(t)(1 + F_n(t)\tau_n) = B_{n-1}(t).$$

Hence, for any $n = 1, ..., N$ we can write

$$B_n(t) = B_0(t) \prod_{k=1}^{n} \frac{1}{1 + F_k(t)\tau_k}, \qquad \text{for any } t \in \mathcal{I}.$$

It is worth to be highlighted that since $t < T_n$, for any $t \in \mathcal{I}$ and $n = 0, ..., N$, the price $B_n(t)$ is always well defined.

Stochastic setting. Hereafter we shall always consider the price $B_0(t)$ of the bond expiring at the tenor date $T_0 = 1$, for $t \in \mathcal{I}$, as the reference numeraire process. Hence, we denote by $\mathbb{Q}$ the forward measure related to T_0, i.e. the martingale measure associated to the numeraire process $B_0(t)$, for $t \in \mathcal{I}$.

Throughout this article, we shall fix a N-dimensional Wiener process $W(t) = (W_1(t), ..., W_N(t))$, for $t \in \mathcal{I}$, defined on the suitable complete probability space $(\Omega, \mathscr{F}, \mathbb{Q})$, and we write $\varrho = (\varrho_{nk})_{nk}$ to denote the corresponding (positive defined) correlation matrix, i.e.

$$dW_n(t)dW_k(t) = \varrho_{nk}dt.$$

In particular, we shall assume constant correlation coefficients given by the usual parameterization:

$$\varrho_{nk} = \exp(-\beta \mid T_n - T_k \mid),$$

with $\beta = 0.01$ in the numerical examples. Note that these coefficients will correspond to the correlation between LIBOR forward rates.

Moreover, for any given $h = (h_1, ..., h_N) \in \mathbb{R}$, we define the following norm:

$$\|h\|_W = \left\{ \sum_{n,k=1}^{N} \varrho_{nk} h_n h_k \right\}^{1/2}.$$

Later on, any stochastic process shall be defined on the probability space $(\Omega, \mathscr{F}, \mathbb{Q})$. Further, since $\mathbb{Q}$ is understood as the reference market measure, we shall refer to any process defined on such a space as a risk-free process.

LIBOR Market Model. For each $n = 1, ..., N$, let $\sigma_n(t)$, for $t \in \mathcal{I}$, be a given deterministic function. According to the LIBOR Market Model, given any fixed $n = 1, ..., N$, the risk-free dynamics associated to the process $F_n(t)$, for $t \in \mathcal{I}$, is defined as

$$dF_n(t) = \mu_n(t)dt + \sigma_n(t)F_n(t)dW_n(t), \tag{1}$$

jointly with some given initial condition $F_n(0)$, where, at any time $t \in \mathcal{I}$, the drift component $\mu_n(t)$ is completely determined by following identity:

$$\mu_n(t) = \sigma_n(t)F_n(t) \sum_{k=1}^{n} \frac{\varrho_{nk}\tau_k\sigma_k(t)F_k(t)}{1 + F_k(t)\tau_k}.$$

Concerning the modelling of volatilities, in this article we choose the widely used parameterization:

$$\sigma_n(t) = [a + b(T_n - t)] \exp[(T_n - t)] + d$$

Furthermore, in the numerical examples we have chosen the constant parameters: $a = 0.07$, $b = 0.2$, $c = 0.6$ and $d = 0.075$.

For any fixed $t \in \mathcal{I}$, set $\mu(t) = (\mu_1(t), ..., \mu_N(t))$ and thus define $\Sigma(t)$ to be the matrix whose components are given by

$$\Sigma_{nk}(t) = \sigma_n(t)F_n(t)\delta_{nk}, \qquad \text{for any } n, k = 1, ..., N, \tag{2}$$

where δ_{nk} denotes the Kronecker delta. Moreover, we shall write $\Sigma_n(t)$ to denote the nth row of the matrix $\Sigma(t)$, for any $n = 1, ..., N$. Then, when setting $F(t) = (F_1(t), ..., F_N(t))$, we may regard (1) as a N-dimensional dynamics by means of the following compact form notation:

$$dF(t) = \mu(t)dt + \Sigma(t)dW(t), \tag{3}$$

jointly with the initial condition $F(0) = (F_1(0), ..., F_N(0))$.

Moreover, for any $n = 1, ..., N$ we shall write

$$\tilde{B}_n(t) = \frac{B_n(t)}{B_0(t)}, \qquad \text{for any } t \in \mathcal{I},$$

to denote the discounted price processes associated to the bond expiring at the tenor date T_n.

The following result provides the risk-free dynamics for the discounted price of any bond expiring at some tenor date in $\mathcal{T}$.

Lemma 1. *For any $n = 1, ..., N$, the discounted bond price process $\tilde{B}_n(t)$ admits the dynamics*

$$d\tilde{B}_n(t) = -\varepsilon_n(t)\tilde{B}_n(t)dW(t), \tag{4}$$

where we set

$$\varepsilon_n(t) \triangleq \sum_{k=1}^{n} \frac{\tau_k}{1 + F_k(t)\tau_k}\Sigma_k(t). \tag{5}$$

Proof. Fix $n = 1, ..., N$ and notice that according to the identity (1) the discounted price $\tilde{p}_n(t)$ at time $t \in \mathcal{I}$, that is given by (4), satisfies the following condition:

$$\ln \tilde{B}_n(t) = -\sum_{k=1}^{n} \ln(1 + F_k(t)\tau_k).$$

For any diffusion process $X(t)$, for $t \in \mathcal{I}$, driven by the Wiener process W, let $\mathrm{DC}X(t)$ denote its vector diffusion coefficient. In particular, the representation (3) implies that

$$\mathrm{DC}F_n(t) = \Sigma_n(t), \tag{6}$$

for any $t \in \mathcal{I}$. Thus, we obtain

$$\begin{aligned} \mathrm{DC} \ln \tilde{B}_n(t) &= -\sum_{k=1}^{n} \mathrm{DC} \ln(1 + \tau_k F_k(t)) \\ &= -\sum_{k=1}^{n} \frac{\tau_k}{1 + \tau_k F_k(t)} \mathrm{DC}F_k(t) \\ &= -\sum_{k=1}^{n} \frac{\tau_k}{1 + \tau_k F_k(t)} \Sigma_k(t). \end{aligned} \tag{7}$$

In order to complete the proof, notice that

$$\mathrm{DC}\tilde{B}_n(t) = B_n(t)\mathrm{DC} \ln \tilde{B}_n(t), \qquad \text{for any } t \in \mathcal{I}.$$

□

3. Insurance Policies Portfolio

In this section, we discuss the problem of the model points selection when dealing with a portfolio of term insurance policies, i.e. those contracts that pay a lump sum benefit on the death of the policy owner, provided that it occurs until a specific term that is defined in the contract. For the sake of simplicity, we assume that the benefit related to each policy to be always represented by a unit amount of a certain currency.

We assume to deal with policies that are unaffected by credit risk, i.e. the insurance company always guarantees the entire benefit that is provided for in the contract. On the other hand, we do not analyse the revenues received by the insurance company and thus we do not take into account the premiums stream of the contract nor any further expenses that are responsibility of the client.

Term insurance portfolios. We assume the generic term insurance policy within a given portfolio to be labelled by both the age of the policy owner at time $t = 0$ and the term date of the contract. In this respect, let us fix I, $J \in \mathbb{N}$ and let $\mathcal{X} = \{x_1, ..., x_I\}$ and $\mathcal{Y} = \{y_1, ..., y_J\}$ be two finite sets of real values such that $x_i \geq 0$ and $y_j \geq 1$, for any $i = 1, ..., I$ and $j = 1, ..., J$. Here and in the sequel, any couple (x_i, y_j) uniquely defines the family of policies related to the class of individuals that are aged $x_i \in \mathcal{X}$ at time $t = 0$ and to the term of the contract $y_j \in \mathcal{Y}$.

Next, for any $i = 1, ..., I$, we shall write $\mu(s, x_i + s)$ to denote the force of mortality at time $s \geq 0$ related to the class of individuals labelled by $x_i \in \mathcal{X}$. In this respect, we suppose that $\mu(s, x_i + s)$ is a deterministic observable function, for any $x_i \in \mathcal{X}$ and $s \geq 0$. Moreover, we shall make the convenient assumption that

$$\mu(s, x_i + s) = 0, \qquad \text{for any } i = 1, ..., I \text{ and any } s \in \mathcal{I}. \tag{8}$$

Remark 1. It is worth to be highlighted that the hypothesis (8) is convenient since it guarantees that any portfolio of term insurance policies does not change within the time interval $\mathcal{I}$ due to the death of the policy owners. Besides, such an assumption is acceptable since the events occurring within the first year only cause a minimal impact on the performance of the overall portfolio.

According to this setup, we define the survival index as

$$S(x_i, T_n) = \exp\left\{ - \int_1^{T_n} \mu(s, x_i + s) ds \right\}, \qquad \text{for any } x_i \in \mathcal{X} \text{ and } n = 1, ..., N, \tag{9}$$

which is understood as the proportion of those individuals labelled by $x_i \in \mathcal{X}$ that survive to the age $x_i + T_n$.

In particular, we consider a Gompertz-type law modelling the force of mortality [11], by setting

$$\mu(s, x_i + s) = a(s) \exp\{(x_i + s) b(s)\}, \qquad \text{for any } s \geq 1 \text{ and } i = 1, ..., I,$$

where $a(s)$ and $b(s)$ are deterministic functions for $s \geq 1$, which are considered to be observables. Assuming that $a(s) = 0$ for $s \in \mathcal{I}$ we guarantee that the previous convenient assumption (8) holds. Throughout, we write S_T to denote the derivative of S in its second variable, which is given by

$$S_T(x_i, T_n) = -S(x_i, T_n)\mu(T_n, x_i + T_n).$$

The following definition introduces the notation for the term insurance policies we shall consider later on.

Definition 1. For any $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$, the discounted risk-free value at time $t \in \mathcal{I}$ of a term insurance policy owned by an individual with age $x_i \in \mathcal{X}$ and with term $y_j \in \mathcal{Y}$ is given by

$$z_{ij}(t) = -\sum_{n=1}^{N} S_T(x_i, T_n)\tilde{B}_n(t)\mathbb{1}_{\{T_n \leq y_j\}}. \tag{10}$$

Any linear combination of the processes (10) gives the risk-free discounted value of a term insurance portfolio. This is stated by the following definition.

Definition 2. We call (term insurance) policy portfolio any matrix $v = (v_{ij})_{ij}$, for $i = 1, ..., I$ and $j = 1, ..., J$. Moreover, for any policy portfolio v, its risk-free discounted value $v(t)$ at time $t \in \mathcal{I}$ is defined as

$$v(t) = \sum_{ij} z_{ij}(t)v_{ij}. \tag{11}$$

Given any policy portfolio v, we understand any of its components v_{ij} as the amount of policies owned by the class of individuals labelled by the age $x_i \in \mathcal{X}$ and the term $y_j \in \mathcal{Y}$. In what follows, for any couple of policy portfolios v_1 and v_2 we shall write $(v_1 - v_2)(t) \triangleq v_1(t) - v_2(t)$, for any $t \in \mathcal{I}$.

4. Model Points Risk Estimation

In this section we apply the theory presented in [9] to the previously introduced LIBOR rates setting. More precisely, we take into account LIBOR forward rates dynamics as the stochastic underlying stochastic factor. For this purpose, we shall always assume a policy portfolio v relative to $\mathcal{X}$ and $\mathcal{Y}$ to be *a priori* given. Moreover, we fix a set of term insurance policy portfolios $\mathcal{W}$ and we refer to any $w \in \mathcal{W}$ as a model points portfolio. In this setting we introduce the following definition.

Definition 3. We refer to the functional $R(\cdot|v) : \mathcal{W} \to \mathbb{R}$ defined by setting

$$R(w|v) = \int_{\mathcal{I}} \mathbb{E}|(v - w)(t) - \mathbb{E}(v - w)(t)|^2 dt, \qquad \text{for any } w \in \mathcal{W},$$

as *the model points risk functional* induced by v over $\mathcal{W}$.

We understand the model points risk functional $R(w|v)$ as the error that occurs when the policy portfolio v is replaced by some model points portfolio $w \in \mathcal{W}$. Such an error is assessed as the average changes of the difference of the two portfolios in terms of the stochastic fluctuation of the interest rate risk term structure.

Definition 4. Let $R(\cdot|v)$ be the model points risk functional induced by v over $\mathcal{W}$. A model points portfolio $w^* \in \mathcal{W}$ is said to be *optimal relative to* v if the following inequality holds true

$$R(w^*|v) \leq R(w|v), \qquad \text{for any } w \in \mathcal{W}. \tag{12}$$

Notice that, in the special case when $v \in \mathcal{W}$ one has that $R(v|v) = 0$. In this respect, we regard any model points portfolio $w^* \in \mathcal{W}$ satisfying the inequality (12) as an optimal representation of the policy portfolio v.

The following result provides an alternative representation of the model points risk functional induced by v over $\mathcal{W}$.

Proposition 1. *The model points risk functional induced by a portfolio v over $\mathcal{W}$ admits the form*

$$R(w|v) = \mathbb{E}\Big\{ \int_{\mathcal{I}} \Big\| \sum_n \Big\{ \sum_{ij} (v_{ij} - w_{ij}) S_T(x_i, T_n) \mathbb{1}_{\{T_n \leq y_j\}} \Big\} \varepsilon_n(t) \tilde{B}_n(t) \Big\|_W^2 (1-t) dt \Big\},$$
$$\textit{for any } w \in \mathcal{W}. \tag{13}$$

Proof. Let $E = \mathbb{R}^N$ and for any $t \in \mathcal{I}$, set $\tilde{B}(t) \triangleq (\tilde{B}_1(t), ..., \tilde{B}_N(t))$. Thus, let $U = (\mathcal{X} \times \mathcal{Y})^{\mathbb{R}}$ be the class of real matrices $r = \{r(x_i, y_j) : x_i \in \mathcal{X}, \text{ and } y_j \in \mathcal{Y}\}$.

Consider the functional $Z \in \mathcal{L}(E, U)$ which is given for any $q \in E$ by

$$Z(q)(x_i, y_j) \triangleq - \sum_n q_n S_T(x_i, T_n) \mathbb{1}_{\{T_n \leq y_j\}}, \qquad \text{for } x_i \in \mathcal{X} \text{ and } y_j \in \mathcal{Y},$$

and note that the process $z(t) = \{z(t) : t \in \mathcal{I}\}$ defined by the identity (10) is recovered when setting

$$z(t) = Z(\tilde{B}(t)), \qquad \text{for any } t \in \mathcal{I}.$$

Since Z is linear, for any $q \in E$, the Frechét derivative $\nabla Z(q) \in \mathcal{L}(E, U)$ satisfies for any $q' \in E$ the following identity:

$$(\nabla Z(q) q')(x_i, x_j) = - \sum_n q'_n S_T(x_i, T_n) \mathbb{1}_{\{T_n \leq y_j\}}, \qquad \text{for } x_i \in \mathcal{X} \text{ and } y_j \in \mathcal{Y}.$$

On the other hand, for any $t \in \mathcal{I}$, when letting $\varepsilon(t)$ be the matrix with the nth row given by $\tilde{B}_n(t)\varepsilon_n(t)$, where $\varepsilon_n(t)$ is defined in (5), we have that for any $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$, the following identity holds

$$(\nabla Z(\cdot)\varepsilon(t))(x_i, x_j) = - \sum_n \tilde{B}_n(t) \varepsilon_n(t) S_T(x_i, T_n) \mathbb{1}_{\{T_n \leq y_j\}}. \tag{14}$$

Consider now the function $\zeta : I \times E \to U$ defined as

$$\zeta(t,q) \triangleq Z(q), \qquad \text{for any } t \in \mathcal{I} \text{ and } q \in E. \tag{15}$$

In this respect, it is worth to be highlighted that the identification (15) is allowed since the survival index (9) does not depend on $t \in \mathcal{I}$, which is the case when imposing the condition (8).

Then, since (4) may be rewritten as the following E-valued dynamics,

$$d\tilde{B}(t) = -\varepsilon(t)dW(t), \tag{16}$$

then Proposition 5 in [9] can be applied to obtain

$$R(w|v) = \mathbb{E}\Big\{ \int_{\mathcal{I}} \Big\| \sum_{ij} (v_{ij} - w_{ij})(\nabla Z(\tilde{B}(t))\epsilon(t))(x_i, x_j) \Big\|_H^2 (1-t)dt \Big\}, \quad \text{for any } w \in \mathcal{W}. \tag{17}$$

which, jointly with the identity (14), gives the representation (13). □

Finally, with the previous notations we summarize that the selection of model points policy portfolio w^* for an original portfolio v is posed as the global optimization problem

$$w^*(v) = \operatorname{argmin}_{w \in \mathcal{W}} R(w|v). \tag{18}$$

In next section we describe the numerical techniques we propose for solving the global optimization problem. As they are highly computational demanding we also propose to take advantage of parallel computing techniques, which are specially well-suited for the kind of numerical algorithms we handle.

5. Numerical methods

In view of the nature of the global optimization problem (18), for a given original portfolio v, we need to propose efficient numerical methods to minimize the model points risk functional $R(\cdot \mid v)$, which is finally given by expression (13).

Note that the model points risk functional is identified as the cost function in the optimization literature.

The global optimization numerical methods we propose will require a very large number of evaluations of the model points risk functional (13), which in turn requires the simulation of the LIBOR rates dynamics. We will refer to each evaluation as the discretization of the model points risk functional or cost function. Each evaluation has to be performed as fast as can be allowed by the available software and hardware computational tools. As we will consider Monte Carlo techniques for the simulation of the forward LIBOR rates, the opportunity of using parallel computing techniques for the evaluation of the cost function comes into place.

Therefore, in this section we describe the two main involved tasks in the numerical methods. Thus, in Section 5.1 we show the numerical discretization of the model points risk functional (13) and its parallel implementation. Next, in Section 5.2, we discuss the proposed numerical methods for the global optimization problem (18), which require very efficient and fast evaluations of the functional discretization.

5.1. Discretization of the model points risk functional

Next, we describe the different issues related to the discretization of the expression (13) of the model points risk functional, which is evaluated a large number of times in the global optimization numerical method.

First, note that the involved expectation is computed by Monte Carlo simulation, so that each simulation requires the computation of the evolution of forward LIBOR rates according to the previously described LIBOR Market model. Following the ideas in [1], taking logarithmic rates and using Ito lemma in (1), the forward rate dynamics satisfies the following equation with a deterministic diffusion coefficient

$$d \ln F_n(t) = \sigma_n(t) \sum_{k=1}^{n} \frac{\varrho_{nk} \tau_k \sigma_k(t) F_k(t)}{1 + F_k(t)\tau_k} dt + \sigma_n(t) dW_n(t). \tag{19}$$

Next, applying the classical Euler-Maruyama scheme for the time discretization of (19), we get

$$\begin{aligned} \ln \hat{F}_n(t+\Delta t) = \ln \hat{F}_n(t) + \sigma_n(t) \sum_{k=1}^{n} \frac{\rho_{nk} \delta_k \sigma_k(t) \hat{F}_k(t)}{1 + \delta_k \hat{F}_k(t)} \Delta t \\ - \frac{\sigma_n(t)^2}{2} \Delta t + \sigma_n(t)(\hat{W}_n(t+\Delta t) - \hat{W}_n(t)), \end{aligned} \tag{20}$$

for $n = 1, \ldots, N$, where $\hat{F}_n(t)$ is the approximation to $F_n(t)$ and $\hat{W}_n(t+\Delta t) - \hat{W}_n(t) \equiv \sqrt{\Delta t}\mathcal{N}(0,1)$ simulates the increment of the multidimensional Wiener process $dW_n(t)$ at time t. The discretization is performed over a uniform mesh defined by the mesh nodes $t_q = q\Delta t$, for $q = 0, \ldots, N_t$ where Δt denotes the constant time step in the Euler scheme, which exhibits strong convergence of order one in Δt.

Next, we describe the discretization of expression (13) for the functional $R(w|v)$. More precisely, if we denote by $\hat{R}(w|v)$ its approximation, then

$$\begin{aligned} \hat{R}(w|v) &= \frac{1}{N_p} \sum_{p=1}^{N_p} \left(\frac{1}{N_t} \sum_{q=1}^{N_t} ||R_{pq}(w|v)||^2 (1 - t_q) \right) \\ &= \frac{1}{N_p} \sum_{p=1}^{N_p} \left(\frac{1}{N_t} \sum_{q=1}^{N_t} R_{pq}(w|v) \cdot C \cdot R^t_{pq}(w|v)(1 - t_q) \right), \end{aligned}$$

where C denotes the correlation matrix, N_t is number of time steps, N_p the number of simulations and R_{pq} is defined by:

$$R_{pq}(w|v) = \sum_{n=1}^{N} (R^*_{pq}(v) - R^{**}_{pq}(w))\nu_{pq}(T_n),$$

with the vector $\nu_{pq}(T_n)$ given by

$$\nu_{pq}(T_n) = \epsilon^p_n(t_q)\tilde{B}^p_n(t_q) = -\tilde{B}^p_n(t_q)\sum_{k=1}^{n} \frac{\tau_k}{1+\tau_k F_n(t_q)}\Sigma_k(t_q),$$

where index p is associated to a particular simulation of forward LIBOR rates and discounted bond price, index q is related to time t_q and index n is related to maturity T_n in the tenor structure. Moreover, we have used the notation

$$R^*_{pq}(v) = \sum_{i,j} v_{ij} S_T(x_i, T_n)(T_n - y_j),$$

$$R^{**}_{pq}(w) = \sum_{i,j} w_{ij} S_T(x_i, T_n)(T_n - y_j),$$

where v_{ij} denotes the number of contracts with age x_i and maturity y_j with nominal one in the original portfolio and w_{ij} denotes the analogous in the model points portfolio.

Concerning the force of mortality, we consider the Gompertz type law modelling with constant parameters, i.e. $\mu(x) = a\exp(bx)$, where we will take $a = 0.0003$ and $b = 0.06$.

The computational cost associated to the evaluation of the discretized risk functional is actually high: note that we use a Monte Carlo numerical scheme with each path involving the simulation of forward LIBOR rates and additionally we have a computationally intensive loop in the policies corresponding to the original portfolio. Also note that the optimization algorithm essentially requires the repeated evaluation of the risk funcional (cost function), which therefore needs to be calculated in a very efficient way.

For this purpose, we have carried out the parallel implementation of the cost function, by using a multi CPU setting and the OpenMP API. Moreover, a parallel random number generation algorithm for multi CPU architectures has been used for parallelizing Monte Carlo simulation. More precisely, in our case we have used the well known Tina's Random Number Generator library (TRNG).

We would like to emphasize that the whole code has been implemented from scratch in C++, including the LIBOR rates simulator and the Monte Carlo technique for computing the risk functional. The psedocode for the risk functional discretization is shown in Algorithm 1.

Algorithm 1: Cost function. Pseudocode.

```
Data: Δt, N_t, Euler time step size and number of time steps
Data: t_i, i = 1, . . . , N_t, Times for Euler Squeme
Data: N_p, Number of Monte Carlo paths
Data: M, Number of tenors
Data: T_k, k = 0, . . . , M, Tenor structure
Data: F_k(0), k = 0, . . . , M, Forward initial rates
Data: Initial volatilities, σ_k(0)
Data: Policies portfolio:
Data: numP, Number of polices in the portfolio
Data: N_p(i), A_p(i), E_p(i), i = 1, . . . , numP, number of contracts, age and expiration of each police of
      the portfolio
Data: Structure of model points portfolio:
Data: numMP, Number of model points
Data: N_mp(i), A_mp(i), E_mp(i), i = 1, . . . , numMP, number of contracts, age and expiration of each
      police of the model points portfolio
Data: a, b, data for survival model
Data: β, data for correlations
S(age, T, t): Force of mortatity function L, build correlation matrix
L = CC^T, Perform Cholesky decomposition
/* Parallel loop */
for j = 1 to N_Paths do
    for i = 1 to N_t − 1 do
        /* Compute vector for the Brownian motions */
        for k = 1 to M do
            Generate the Brownian step ΔB_k(t_i)
        ΔW(t_i) = CΔB(t_i)
        /* Compute vector of Forward rates */
        for k = 1 to M do
            Simulate [F_k(t_i) using log-Euler scheme (20)
            ln F_k(t_i) = ln F_k(t_{i−1}) + μ_k(t_i)(t_i − t_{i−1}) + σ_k(t_i)F_k(t_i)ΔW_k(t_i)
            where μ_k(t_i)
            μ_k(t_i, ω_j) = σ_k(t_i)F_k(t_i) Σ_{h=1}^{k} ϱ_kh τ_h σ_h(t)F_h(t_i) / (1+F_h(t_i)τ_h) .
    for i = 1 to N_t − 1 do
        p(t_i, k) = 1
        for k = 1 to M do
            Compute the discounted bond prices
            B(t_i, k) = B(t_i, k) · 1 / (1 + F_k(t_i)τ_k)
            Compute the diffusive component of the discounted bond price
            ν_ij(T_k) = τ_k / (1 + τ_k F_k(t_i)) σ_k(t_i)F_k(t_i)
        Rj = 0
        for k = 1 to M do
            sum_p = 0
            for l = 1 to NumP do
                sum_p+ = N_P l[l] ∗ S(A_P[l], T_k, t_i) ∗ (T_k <= E_p[l])
            sum_mp = 0
            for l = 1 to N_MP do
                sum_mp+ = N_MP[l] ∗ S(A_MP[l], T_k, t_i)(T[k] <= E_MP[l])
            Rj = Rj + (sum_p − sum_mp) ∗ (−B(t_i, k) ∗ ν_ij(T_k))
        R+ = Rj ∗ L ∗ Rj ∗ (1 − t_i)
R = (1/NPaths) ∗ (1/N_t) ∗ R
```

5.2. Global optimization algorithms

Obtaining the model points portfolio results to be a very difficult problem, as it involves solving a global optimization problem in high dimension. More precisely, the dimension of the searching space is given by the number of policies in the model points portfolio, which is equal to $I \times J$ in our case.

For solving global optimization problems, stochastic algorithms are usually required. They have the advantage that they can deal with complex problems, discarding local optima and avoiding getting stuck in this local solutions. However, their main disadvantage is associated to their slow convergence, due to their stochastic nature. One example of these kind of algorithms is Simulated Annealing (SA), see [8] and references therein for details. On the other hand, deterministic local optimization algorithms are faster, their disadvantage being that they can't scape from local minima. Some examples of these local algorithms are Pattern Search or gradient based methods like NCG, BFGS, L-BFGS [13] or L-BFGS-B [2].

One possible technique to obtain faster global optimization algorithms comes from mixing both kinds of algorithms, thus obtaining the so called hybrid algorithms, which can benefit from the global convergence properties of the stochastic ones and from the speed of convergence of the local optimization algorithms (see Figure 1, for a sketch of the behavior of hybrid algorithms). One example of hybrid algorithms is the Basin Hopping (BH) algorithm, see [14] and [7], as well as references therein. Basically, in BH algorithm a Simulated Annealing is used for sampling the searching space by randomly generating neighbors, and local gradient algorithms are applied to capture the minima starting from the generated points by the stochastic sampler.

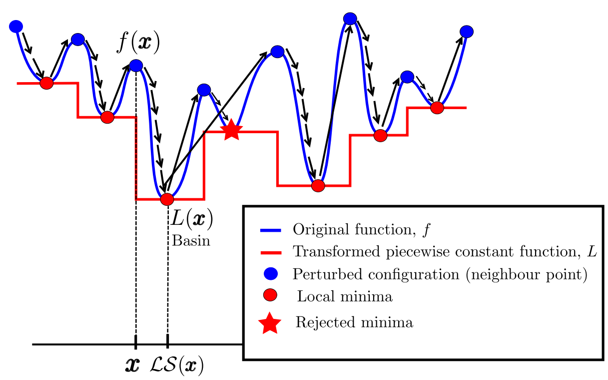

Figure 1: Sketch of Basin Hopping algorithm.

The pseudocode of the Basin Hopping algorithm is shown in Algorithm 2. The whole optimization routines have been implemented from scratch in C++, following the previous works [8, 7].

6. Numerical examples

In this section we present two tests to validate the proposed model points selection technique, and an application to a real world scenario, with unknown solution.

First, we recall that the cost function is stochastic and its computation requires a Monte Carlo simulation technique. In this respect, all the tests in this section have been performed by using 1000 paths for the Monte Carlo simulation in the computation of the cost function.

In this section we also show the comparison between the global SA method and the hybrid BH with L-BFGS-B as local optimizer, as the alternatives to solve

Algorithm 2: Basin Hopping method, pseudocode.

$\boldsymbol{y}$ = sampled from a uniform random distribution in D;
Number of successive rejections: $j = 0$;
Iteration number: $k = 0$;
Initial position: $\boldsymbol{x}_0 = \boldsymbol{x}^* = \mathcal{LS}(\boldsymbol{y})$;
while *($j < J$) **or** ($T < T_{min}$)* **do**
    for i=0:N **do**
        $\boldsymbol{y}_k$ = random uniform in $B(\boldsymbol{x}_k, \boldsymbol{r}_k)$;
        $\boldsymbol{u}$ = random uniform in $[0, 1]$;
        $\Delta = L(\boldsymbol{y}_k) - L(\boldsymbol{x}^*)$;
        if $\boldsymbol{u} < \exp(-\Delta/k_B T)$ **then**
            $\boldsymbol{x}^* = \boldsymbol{x}_{k+1} = \mathcal{LS}(\boldsymbol{y}_k)$;
            $j = 0$;
        else
            $j = j + 1$;
        $k = k + 1$;
    Update radius $\boldsymbol{r}_k$;
    $T = \rho \cdot T$;

the problem. More precisely, we will present some graphs with the evolution of the value of the cost function with respect to the the number of evaluations. Note that the number of evaluation is closely related to the computational cost.

Concerning the tenor structure of the LIBOR model, we consider 100 tenors, with maturities ranging from 1 to 100 and initial rates are $F_1(0) = 0.01$, $F_2(0) = 0.02$, $F_3(0) = 0.03$, $F_4(0) = 0.04$ and $F_n(0) = 0.05$, for $n \geq 5$.

All tests have been performed in a server with 16 GB of RAM, and 16 CPU cores (two Intel Xeon E5-2620 v4 at 2.10GHz).

6.1. Performance of the parallel implementation of risk functional evaluation

As we mentioned before, the computational cost of the proposed optimization algorithms is closely related to the cost of evaluation of the risk functional, which is performed a large number of times during the optimization procedure. Therefore, as a previous step to the presentation of numerical examples, we show the performance of the multi-CPU implementation of the model points risk functional calculation. For this purpose, we consider from 100 to 10000 policies in the original portfolio and 45 policies in the model points portfolio. Moreover, we consider 1000 paths for the Monte Carlo simulation.

Figure 2 and Table 1 show the obtained speed up for the parallel implementation of the risk functional with different numbers of policies in the original portfolio and different number of cores being used. As it is illustrated by both, it occurs that the speed-up gets closer to the number of used cores when the number of policies increases. Therefore, parallelization results more interesting and efficient for large enough portfolios.

N. policies	N. cores	Time (seconds)	Speedup
10000	1	943.25	-
	2	517.50	1.82
	4	249.21	3.79
	8	130.70	7.21
	16	688.5	13.70
5000	1	518.81	-
	2	270.87	1.91
	4	137.97	3.76
	8	70.22	7.38
	16	417.32	13.43
1000	1	102.65	-
	2	53.28	1.92
	4	27.73	3.70
	8	14.99	6.85
	16	8.36	12.28
500	1	54.56	-
	2	28.63	1.90
	4	15.06	3.62
	8	8.63	6.32
	16	5.15	10.58
100	1	17.351	-
	2	9.21	1.88
	4	4.94	3.51
	8	2.67	6.48
	16	1.88	9.21

Table 1: Parallel implementation of the risk functional evaluation. Time and speedups.

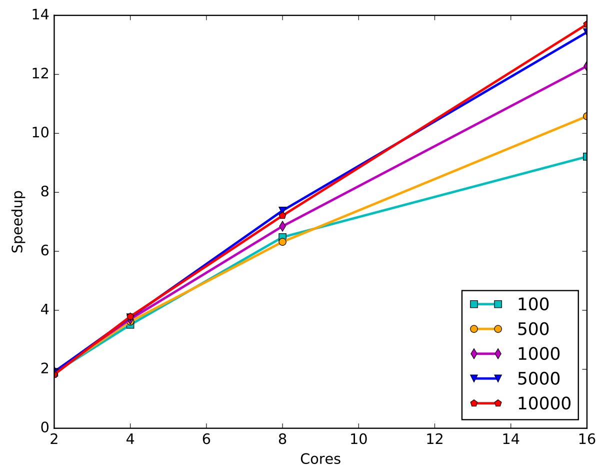

Figure 2: Speedups of parallel implementation vs. number of cores for different numbers of policies in the original portfolio.

6.2. Example 1: Selection of one model point

In this example we aim to represent an original portfolio with a model points portfolio with one single model point.

More precisely, the original portfolio contains 10 policies, the main data of which are shown in Table 2. The model points portfolio consists in just one single model point with age equal to 42 and maturity equal to 28 years. Therefore, the objective is to determine the number of contracts in this model point that minimizes the corresponding risk functional.

Therefore, in this case the cost function (risk funtional) is given by a one dimensional function. If we evaluate the cost function for different number of contracts, we obtain the expected parabolic convex function, so that the global minimum can be graphically checked (see Figure 3). Note that in this example we do not apply an optimization method, but just evaluate the risk functional for the different number of contracts and identify the particular number of contracts (≈ 1315000) that minimizes the risk funcional with $(x_1, y_1) = (42, 28)$.

6.3. Example 2: Repeated policies classification

In this second example we validate the proposed technique for a problem with known solution. Thus, we compute the model points portfolio by using the two previouosly discussed global optimization algorithms: the pure global SA algorithm and the hybrid BH algorithm, so the we can compare the performance of both algorithms for the hard problem we pose.

In this problem, for building the original portfolio we start from the small portfolio of 10 policies in Example 1 (see Table 2), and we build up the large original portfolio by repeating each policy 1000 times, so that we end up with

Age	Maturity	Number of contracts
20.0	50.0	50000
25.0	45.0	100000.0
30.0	40.0	150000.0
35.0	35.0	200000.0
40.0	30.0	250000.0
45.0	25.0	300000.0
50.0	20.0	350000.0
55.0	15.0	400000.0
60.0	10.0	450000.0
65.0	5.0	500000.0

Table 2: Original portfolio for Example 1.

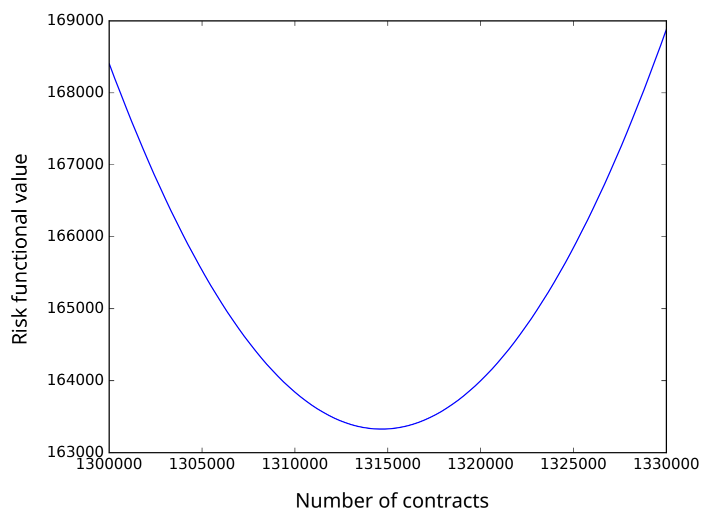

Figure 3: Graph of the one dimensional risk functional in Example 1.

an original portfolio containing 10000 policies. Although they are repeated (actually, there are only 10 types of policies), each policy is understood as a different individual one.

Concerning the model points portfolio, we try to represent the previously described original portfolio with a 10 model points portfolio with the same structure (ages and maturities) as the one in Table 2, so that the problem consists of finding the number of contracts in the model points for that model points portfolio that better represents the original portfolio with respect to the risk functional.

Clearly, the analytical solution consists of multiplying the number of contracts in Table 2 by 1000. Moreover, by choosing the model points portfolio in this way the corresponding value of cost function is equal to zero.

Age	Maturity	Number of contracts
20.0	50.0	50000000
25.0	45.0	100000000
30.0	40.0	150000000
35.0	35.0	200000000
40.0	30.0	250000000
45.0	25.0	300000000
50.0	20.0	350000000
55.0	15.0	400000000
60.0	10.0	450000000
65.0	5.0	500000000

Table 3: Obtained solution in Example 2 with BH.

As we can see in Figure 4 for the BH method with L-BFGS, the method reaches the very small value 1.75×10^{-08} of the cost function, the exact solution being equal to zero. Moreover, in Table 3 the obtained values for the number of contracts of the model points portfolio are shown. We note that these value are rounded to the eighth decimal digit, thus matching those ones corresponding to the exact solution. These roundings explain the small difference between 1.75×10^{-08} and zero in the cost function. On the other hand, the computational time was 87.35 hours (about 3.63 days) using L-BFGS, when using 16 cores (and 32 threads).

Next, in Figure 5 we show the convergence of the SA algorithm for solving the optimization problem. In this case, the computational time was 839.35 hours (about 35 days). The obtained solution is the same as with BH algorithm.

By using this example with analytical solution, we have checked that the proposed technique is able to classify repeated policies in their corresponding buckets, which is a desirable property of the risk function.

Moreover, we would like to emphasize that in the optimization algorithms we have imposed very hard stopping criteria in the involved numerical methods to guarantee a very small error with respect to the analytical solution (high accuracy). This leads to high computational times, even though we use some

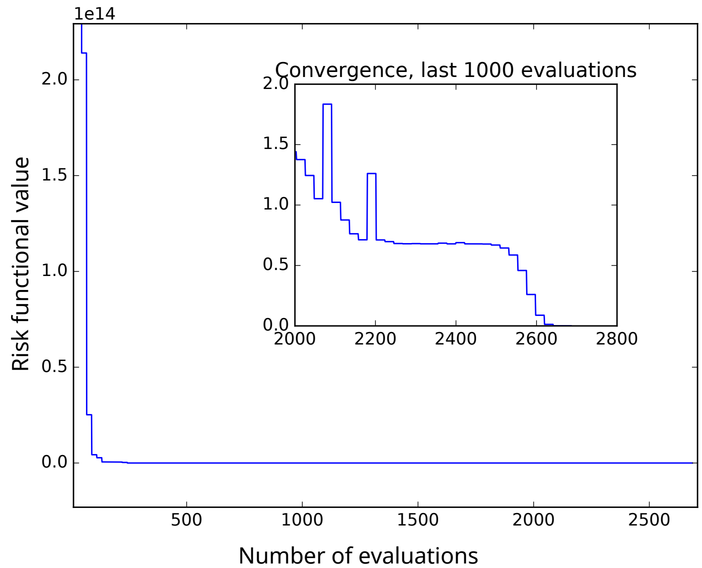

Figure 4: Convergence of the BH algorithm for Example 2.

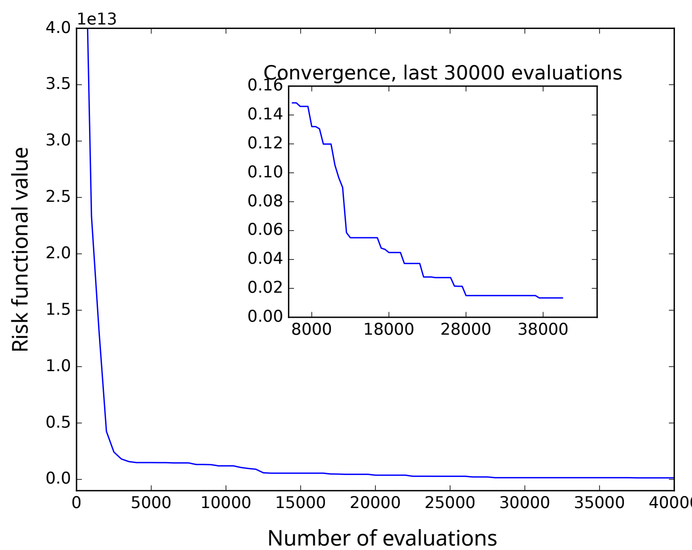

Figure 5: Convergence of the SA algorithm for Example 2.

parallel computing tools. Also, in view of computational times we note that BH results to be more efficient than SA.

6.4. Example 3: Real scenario

In this example we present a realistic synthetic example. More precisely, in this case the original portfolio consists of 10000 different policies, and we want to represent it with a portfolio of 45 model points, which is given by a grid of 9 ages and 5 maturities. Ages will vary from 30 to 70 with step 5 years and maturities range from 5 to 25 with step 5 years, thus accounting for a total of 45 model points.

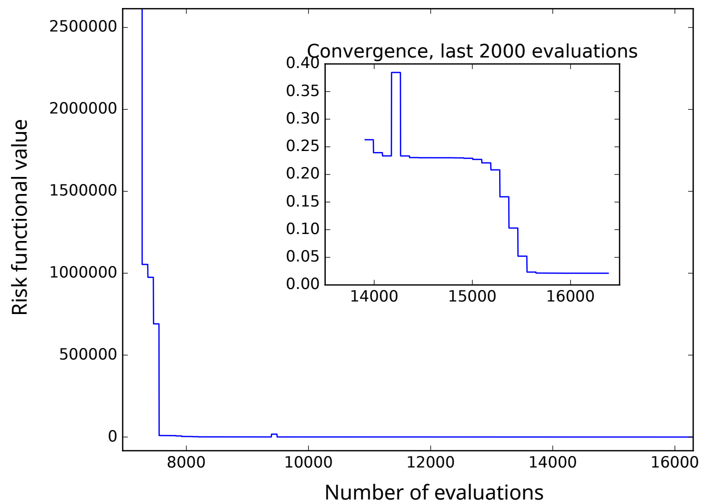

Figure 6: Convergence of the BH algorithm for Example 3.

This is a much more difficult test than the analytical one in Example 2. The required total computing time by the BH hybrid optimization algorithm, with L-BFGS-B as local minimizer, was 136.43 hours when using 16 processors for the parallel evaluation of the risk functional. The obtained final value of the model points risk function is 0.175927. Moreover, the computed number of contracts in the model points portfolio are shown in Table 4 and Figure 7.

The global property of the hybrid algorithm is of great importance for this test, as several iterations of the global optimization algorithm were needed to reach the minimum, thus avoiding to get stuck to a local minimum as it may happen with a pure local optimization method. Also the convergence speed and accuracy of the L-BFGS-B local optimizer results to be a key point, as the SA pure global algorithm was not able to converge for this realistic example.

7. Conclusions

In this paper, we mainly propose a methodology to address the problem of selection of a model points portfolio for a given initial portfolio of life insurance policies. The methodology requires to overcome both mathematical and computational challenges.

First, the selection is based on the definition of an appropriate market risk funtional to evaluate how the model points portfolio represents the initial one. In this respect, a realistic appropriate market risk functional has been proposed,

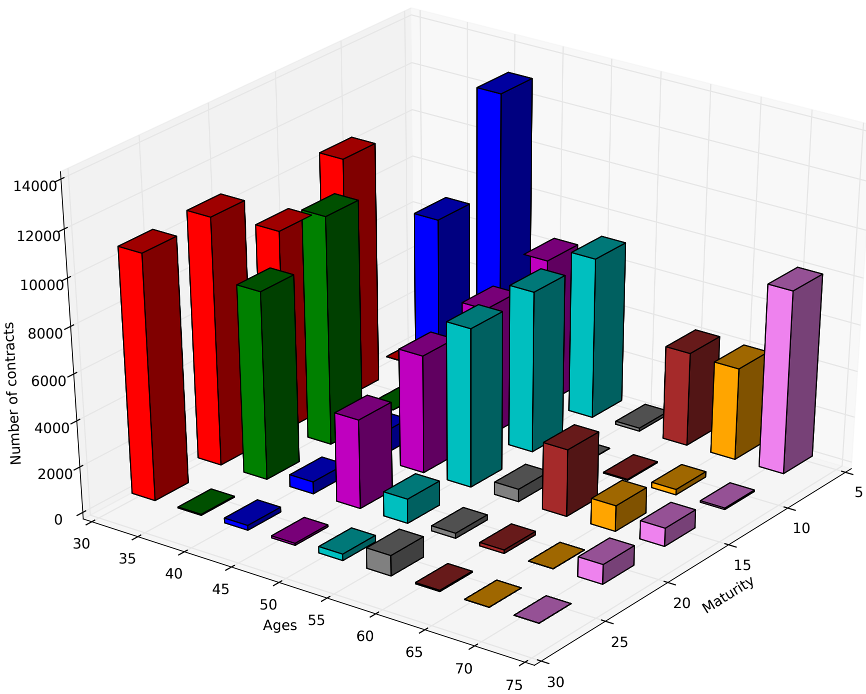

Figure 7: Solution for Example 3.

which is mainly based on the LIBOR market model for the evolution of forward rates.

Secondly, once the risk functional has been chosen, obtaining the model points portfolio involves the solution of a high dimensional global optimization problem, so that the numerical optimization methods require a very large number of risk functional evaluations. The risk function has been discretized by using a Monte Carlo. In order to speed up the high computational cost associated to each evaluation, it has been parallelized in a multi CPUs setting using OpenMP. Concerning the choice of the optimization methods, the hybrid Basin Hopping optimization algorithm, which combines the advantages of mixing stochastic global optimization algorithms with gradient local optimizers, has been proposed and compared with an alternative purely global optimization technique. The accuracy and performance of the algorithm have been analyzed with several numerical examples, both in cases with known solution and also in a real example without analytical solutions. The results show that the hybrid global optimization algorithm more suitable for selecting the model portfolio.

Although the proposed methodology seems successful from the mathematical and computational perspectives, future work to improve both aspects can be addressed and it is under consideration by the authors.

From the computational side, in order to reduce the computational time, the implementation in (multi-)GPUs of the functional evaluation and/or the

		Maturities				
		5	10	15	20	25
Ages	30	0.0490344	10363.8	8678.28	10654	10560.2
	35	0.363568	0.386776	9859.99	8118.85	42.143
	40	12843.9	8892.82	984.418	523.38	208.743
	45	6318.73	5653.26	5108.62	3867.53	84.3039
	50	6970.11	6976.25	6874.87	1062.52	280.726
	55	132.469	10.4804	543.098	219.955	889.387
	60	4057.54	38.8013	2917.22	151.168	51.1771
	65	3991.46	234.856	1120.47	0.996919	0.32282
	70	7923.19	63.4409	797.372	859.543	0.203422

Table 4: Obtained solution with BH in Example 3. The number of contracts is divided by 10^4

global optimization algorithm is under consideration by the authors. Note that this issue requires the migration of the code to CUDA and a very efficient implementation to get advantage of the GPUs architectures. In this respect, some authors of this article have already developed implementations in GPUs for ALM in insurance problems in [6].

From the financial and mathematical modelling side, the problem of the optimal model points selection for life insurance portfolios has been addressed by considering the changes in interest rates over time as the main source risk. This means that no stochastic fluctuation affecting the time evolution of the mortality rate has been taken into account, and a deterministic survival trend over time for any cohorts is considered. However, evidences of the stochastic behaviour in the trend of the life expectancy can be found in [3] and the references therein. In this respect, one may refer either to mortality or longevity risk to account for the possibility faced by the insurer to suffer losses due to unexpected changes in the long-terms life trend displayed by the policy owners. As a direct consequence, a more sophisticated approach for the model points selection should be obtained by considering the stochastic time evolution of both the interest rate term structure and the mortality rate. In this context, the dependence between mortality risk and interest rate risk cover a central issue. A first attempt to model this dependence is carried out in [12].

Acknowledgements

This work has been partially funded by EU H2020-MSCA-ITN-2014 (WAKE-UPCALL Grant Agreement 643045), Spanish MINECO (Grant MTM2016-76497-R) and by Xunta de Galicia (Grant CN2014/044)

References

[1] Brigo, D., and Mercurio, F. *Interest Rate Models - Theory and Practice With Smile, Inflation and Credit.* Springer, 2006.

[2] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. on Scientific Computing 16* (1995), 1190–1208.

[3] Cairns, A., Blake, D., and Dowd, K. Pricing death: Frameworks for the valuation and securitization of mortality risk. *ASTIN Bulletin: The Journal of the IAA 36* (2006), 79–120.

[4] Denuit, M., and Trufin, J. Model points and tail-var in life insurance. *Insurance: Mathematics and Economics 64* (2015), 268–272.

[5] EIOPA. QIS5 Technical Specification. *European Commission.* (2010).

[6] Fernández, J. L., Ferreiro, A. M., García-Rodríguez, J. A., and Vázquez., C. GPU parallel implementation for asset-liability management in insurance companies. *Journal of Computational Science 24* (2018), 232–254.

[7] Ferreiro, A. M., García-Rodríguez, J. A., Vázquez, C., e Silva, E. C., and Correia, A. GPU parallelization of two-phase optimization algorithms. *Mathematics and Computers in Simulation 156* (2019), 67–90.

[8] Ferreiro, A. M., Rodríguez, J. A. G., López-Salas, J., and Vázquez, C. An efficient implementation of parallel simulated annealing algorithm in GPUs. *Journal of Global Optimization 57* (2013), 863–890.

[9] Ferri, E. Optimal model points portfolio in life insurance. `https://arxiv.org/abs/1808.00866`.

[10] Gerstner, T., Griebel, M., Holtz, M., Goschnick, R., and Haep, M. A general asset-liability management model for the efficient simulation of portfolios of life insurance policies. *Insurance: Mathematics and Economics 42 (2)* (2008), 704–716.

[11] Gompertz, B. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. in a letter to Francis Baily, Esq. FRS & c. *Philosophical Transactions of the Royal Society of London 115* (1825), 513–583.

[12] Jalen, L., and Mamon, R. Valuation of contingent claims with mortality and interest rate risks. *Mathematical and Computer Modelling 49* (2009), 1893–1904.

[13] Liu, D. C., and Nocedal, J. On the limited memory method for large scale optimization. *Mathematical Programming B 45* (1989), 503–528.

[14] Wales, D. J., and Doye, J. P. K. Global optimization by Basin-Hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 Atoms. *Journal Physical Chemistry A 101* (1997), 5111–5116.