

ESG Calibration: The difficulty of the calibration process



Kevin Poulard
Consultant, ADDACTIS® Software
kevin.poulard@addactis.com



+32 (0)2 526 13 10



+33 (0)4 81 92 13 00

Argentina - Brazil - Belgium - Chile - Colombia - Costa Rica - Ecuador - France - Germany - Guatemala - Italy - Luxembourg - Mexico
Morocco - The Netherlands - Panama - Peru - Poland - Senegal - Singapore - Spain - Switzerland



Contents

01
04
08

ESG: A Goal that leads to
a process

Optimization process

Conclusion

In the recent years, Solvency II requires (re)insurers the ability to estimate financial and insurance risks. New quantitative measurements have been introduced in order to assess those risks. Moreover, stress tests and adverse scenarios should be taken into account to measure the risks' volatility.

For those reasons, Economic Scenario Generators (ESG) have become more and more inevitable especially for Life insurers. An Economic Scenario Generator is traditionally a set of mathematical models that are used to project and simulate economic and financial quantities. This field started with Wilkie^[1] and its investment model published in 1986. He used stochastic time series to model inflation, interest rates and equity's dividend. Then, this field had been the subject of extensive study and the majority of recent ESG are based on stochastic differential equations.

However, in order to use those models, we first need to find the model's parameters that will be used to simulate the scenarios. This is called the Calibration process and it is usually considered as the most important and complex step. In fact, we try to minimize the difference between instruments values from the market or historical data and the underlying model. Indeed, we need flexible optimization algorithms in order to calibrate several models with different specificities. Moreover, local optimum should be avoided at all costs. We will show how heuristic algorithms, and more specifically genetic algorithms, could help to find global optimum. Finally, analytic algorithms could be of great help to speed up convergence when constraints are added to the optimization process.

ESG: A GOAL THAT LEADS TO A PROCESS

1- ESG under Solvency II

As we explained, Solvency II requires (re)insurers a market consistent valuation of assets and insurance liabilities. In this framework, they have to determine the capital requirement using a fair value approach. In order to determine the fair value of liabilities, the Best Estimate (BE) is used to compute the technical provisions. This Best Estimate represents the discounted expectancy of future liability cash flows.

In order to express the need of an ESG, we will take a Life Profit Sharing product as an example. The policyholder's premium is invested in a pool of assets. Then, he will receive the maximum between a share of the profit and a Minimum Guaranteed Rate. Those products are linked to the underlying assets' performance and the policyholder's behavior should be taken into account because he can surrender from the contract at any time.

¹Wilkie, A.D. (1986). A stochastic investment model for Actuarial Use, Transactions of the Faculty of Actuaries. 39: 341-403

Because the Cash Flows are dependent of the underlying risk, the BE can be expressed as follows:

$$BE_{Stochastic} = \mathbb{E} \left[\sum_{t \geq 1} \delta(t) CF(t) \right] = \mathbb{E} \left[\sum_{t \geq 1} \delta^{(k)}(t) \mathbb{E}[CF(t) | R^{(k)}] \right]$$

Where

- $\delta^{(k)}(t)$: Discount rate for the k^{th} scenario at time t ;
- $CF(t)$: Cash Flow at time t ;
- $R^{(k)}$: Here, the Cash Flows depend on the financial risks $R^{(k)}$ (Assets valuation and Interest rates).

Then, we use the different scenarios produced by an ESG to compute the BE with Monte-Carlo simulations:

$$BE_{Stochastic} = \frac{1}{N} \sum_{k=1}^N \sum_{t \geq 1} \delta^{(k)}(t) \mathbb{E}[CF(t) | R^{(k)}]$$

If we consider that the underlying products assets are:

- Bonds: We will need Interest rates scenarios;
- Equities: We will need Equities returns scenarios.

Here, we see the usefulness of using several scenarios. In fact, it will trigger specific dynamic behavior of the policyholder in the case the asset or interest rate fall down.

Moreover, in a stochastic framework, we can compute the Financial Options and Guarantees (FOGs):

$$FOGs = BE_{Stochastic} - BE_{Determinist}$$

2- ESG scenarios creation

In the previous example, we saw where ESG scenarios could be useful. They are used to produce scenarios according several financial and economic quantities among which:

- Interest Rates;
- Equities returns;
- Inflation rate;
- Properties returns with the Capital and Rent;
- Credit spreads.

Where

- S_t : Equity's price;
- r_t : Interest rate on the period;
- σ : Equity's volatility that would be calibrated;
- W_t : Brownian movement; this is the source of randomness;
- Δ_t : Discretization step;
- $(W_{t+\Delta_t} - W_t) \sim N(0, \Delta_t)$: Brownian increments.

Of course, the random variables W_t could be correlated among the different economic quantities in order to represent the dependency among those quantities.

Those quantities are simulated using mathematical models. For example, if the user wants to create Equity returns scenarios using a Black-Scholes model, the following stochastic equation will be discretized:

$$\frac{dS_t}{S_t} = r_t dt + \sigma dW_t$$

$$S_{t+\Delta_t} = S_t \exp \left(\left(r_t - \frac{\sigma^2}{2} \right) \Delta_t + \sigma (W_{t+\Delta_t} - W_t) \right)$$

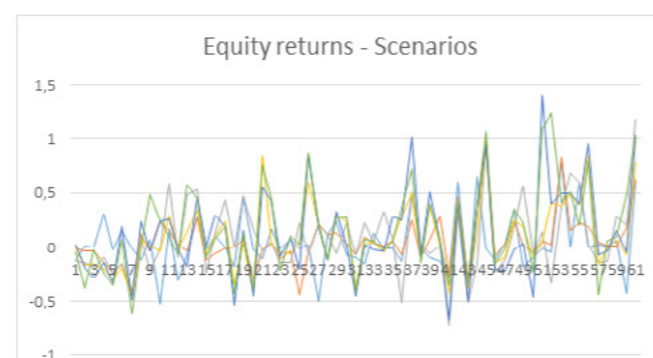


Figure 1: A given Equity's return scenario for different indexes

3- ESG process

When using an ESG, the following steps and questions should be asked:

- Projection context: Does the projection need to be Risk-Neutral or Real-World?
 - The Real-World approach will try to fit the historical and statistical behaviors of the desired economic quantities. This could be used in an ORSA model to project Business plan assumptions or to optimize the strategic asset allocation;
 - The Risk-Neutral approach is used to simulate scenarios that are consistent with market prices at the calibration date. Those are used in the Solvency II BE calculations.

In the following part, we will take the Risk-Neutral approach as our general example. However, the optimization algorithms framework described later could be applied for the Real-World calibration process.

- Model choice:
 - It will depend on the financial quantity projected and the projection context;
 - The model complexity should be taken into account in order to understand the parameters behind the model projection.
- Calibration: Crucial step that could lead to misleading scenarios
 - Calibration date and market data extraction;
 - Optimization process and algorithm.
- Simulation: The stochastic differential equation is discretized in order to project the financial or economic quantity.

Moreover, quantitative and qualitative validations should be done all along the process.

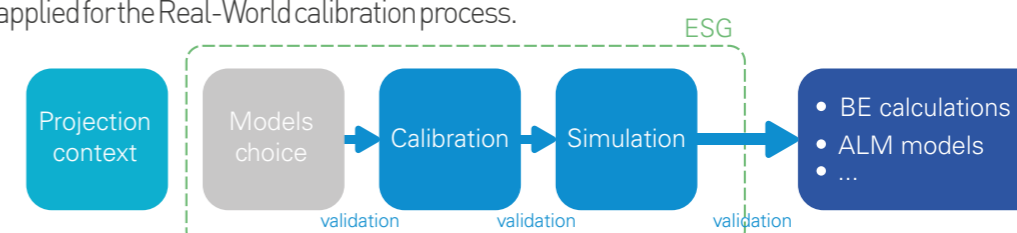


Figure 2: ESG Process

4- Risk-Neutral calibration: Data and optimization problem

The calibration process is considered as crucial because it will return the model's parameters used in the diffusion process. A mislead in the data preparation or optimization algorithm could lead to unusable scenarios even if the model is adapted.

Step 1: Data preparation

We have to select relevant data at the calibration step. The selected data depends on the three following questions:

- Which financial quantity we want to simulate?
 - Interest rate, equity's return, ...
- Which projection context are we in?
 - Risk-Neutral vs Real World.
- Which financial instruments do we select?
 - Caplets or Swaptions for Interest rates;
 - Call for Equities;
 - Bloomberg is mainly used to quote the instrument's price / volatility.

Step 2: Optimization problem

In the Risk-Neutral approach, we try to replicate the quoted market's instruments. In order to do so, we minimize the square error between the market's instrument value and the model's instrument value. The model's instrument value is calculated using a closed-form formula and depends on the model's parameters that we try to optimize.

Also, it is possible to calibrate on the price or on the volatility. Insurers prefer to calibrate on the volatility because it represents directly the risk underlying the instrument.

The optimization program will be as follows:

$$\theta^* = \underset{\theta}{\text{Argmin}} \sum_{i=1}^N m_i (\text{MarketIV}_i - \text{ModelIV}_i(\theta))^2$$

Where

- θ : Model's parameters;
- MarketIV_i : i^{th} instrument market volatility / price;
- ModelIV_i : i^{th} instrument model volatility / price that depends on the model's parameters θ ;
- m_i : i^{th} instrument weight:
 - $m_i=1$: More weight will be given to high instrument value;
 - $m_i=1/(\text{MarketIV}_i)^2$: Same weight will be applied (Relative difference);
 - $m_i \neq m_j, \forall i \neq j$: Custom weight driven by specific constraint (instrument's liquidity, ...).
- N : Number of instruments.

Indeed, numerical methods will be used in order to solve this optimization problem and calibrate the model's parameters.

OPTIMIZATION PROCESS

The optimization problem could be summarized as follows:

$$\min_{\theta} f(\theta)$$

$$s.t. \begin{cases} \theta \in [LB^d; UB^d] \\ h(\theta) \leq 0 \end{cases}$$

Where:

- f : Previous objective function; square errors between market price / volatility and closed-formula model price / volatility
- θ : Model's parameters
- d : Number of parameters
- LB and UB are respectively the parameters' Upper and Lower bounds

Numerical and heuristic methods are used over analytical methods for several reasons:

- The optimization function f is non-linear;
- Gradient function ∇f should be considered as unknown;
- The parameters θ are box-constrained;
- Non-linear constraints h could be added. This is the case in the Heston model calibration;
- The willingness to find a global optimum and avoid local ones;
- The difficulty to find analytical algorithms that behave well for all basic ESG models.

1- Metaheuristic and genetic algorithms

Metaheuristic is a sub-class of heuristic algorithms that are meant to be more efficient because they are driven by mathematical concepts. Metaheuristics are mainly composed of stochastic iterative algorithms used to find a global optimum. They try to learn the characteristics of the objective function to find the best solution.

The genetic algorithm is an evolutionary algorithm that belongs to those Metaheuristics. They are closely related to the biological field and use the same wording to describe the process. They have gained popularity in the late 80s with the artificial intelligence researches. From this starting point, many papers have been published on this subject in order to improve their behavior on different optimization programs but also to prove theoretical results on the convergence of those algorithms^[2].

We have chosen to implement the NSGA-II algorithm (Non-Dominated Sorting Genetic Algorithm)^{[3][4]}. It has the advantage to address multi-objective optimization problems as well as mono-objective optimization problems.

The general genetic algorithm process is as follows:

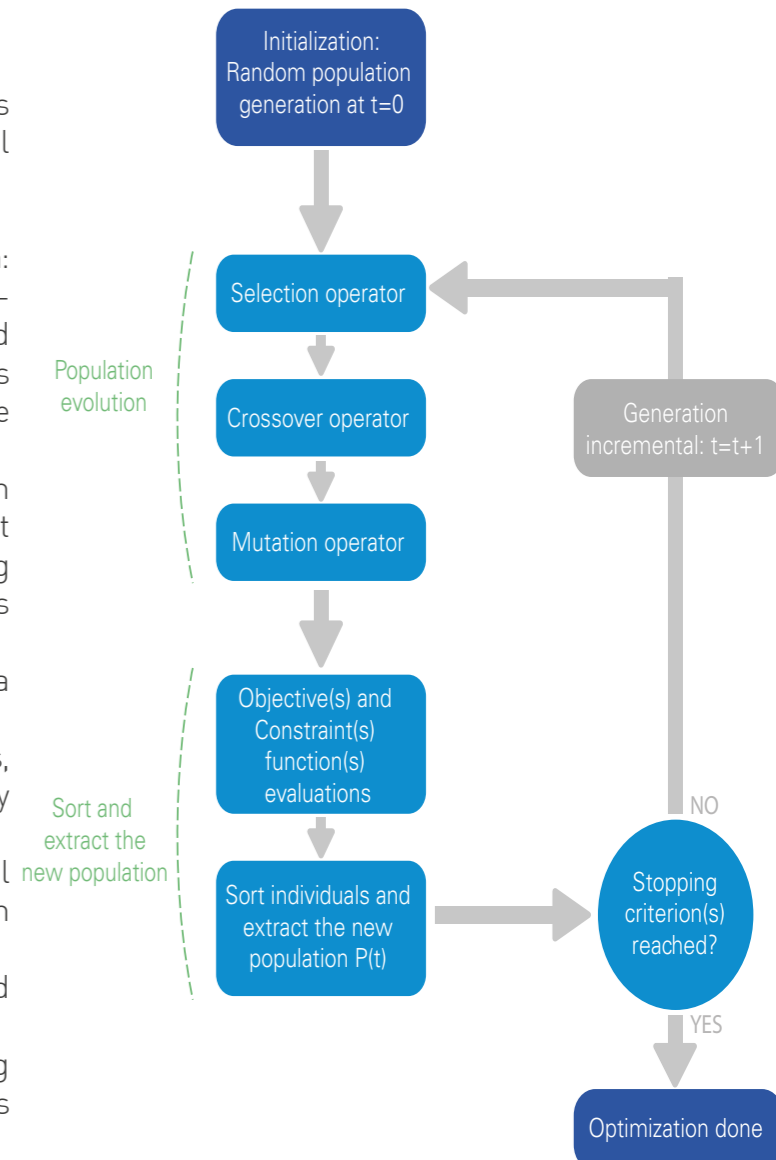
- A first generation of individuals is created. That means that we simulate n individuals where each individual represents a feasible solution of m variables.

Then, an iterative process begins:

- First, we select individuals from the previous generation:
 - A tournament selection could be used for mono-objective optimization program: this method consists of taking a random number of individuals and selects one using a geometric law (we suppose that the individuals are previously sorted);
 - The NSGA-II algorithm is taking randomly from the previous generation 2 individuals. Then, it compares and returns the best solution using fitness and diversity measures among individuals as criterions.
- We apply genetic operators in order to create a new population:
 - For the crossover operator, from two individuals, we will mix the corresponding variables randomly in order to obtain 2 new individuals;
 - For the mutation operator, we take an individual and apply (or not) a gaussian perturbation on each variable.
- For each individual, all objective function(s) and constraint(s) are evaluated.
- Then, we can sort the different individuals using their fitness and the constraints values. Indeed, this

process is much more complex than what it seems:

- For mono-objective genetic algorithm, the individual's fitness (value of the objective function) will be used. However, a balance should be found between the objective value of the individual and, if so, its constraint(s) value(s).
- The NSGA-II algorithm uses the pareto frontier definition in order to sort individuals in a multi-objective context. For that, a non-dominated sorting algorithm was implemented and individuals are classified inside different fronts. If there is a unique front, that means that all solutions are non-dominated.
- Finally, we test if the stopping criterion(s) are met to know if a new iteration / generation is needed. Different criterions could be used:
 - A maximum number of generations;
 - Stability among individuals;
 - Stability of the individuals among 2 generations / iterations;
 - Mix of those criterions.



²David E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, 1989.

³Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on evolutionary computation, Vol. 6, NO. 2, April 2002.

⁴N.Srinivas and Kalyanmoy Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation, 2(3):221 – 248, 1994

2- How to speed up convergence and deal with constraints?

After applying the genetic algorithm to calibrate the main ESG models, it appears that the implemented algorithm was very successful. However, we have seen two drawbacks:

- The convergence speed:
 - That could come from high computational cost to estimate the objective function. In fact, genetic algorithm is working with a population of individuals that need to be evaluated for each generation. Therefore, the convergence speed of the algorithm will be directly impacted by the objective function evaluation.
- For non-linear constrained problem, high difficulties to find global optimum solution.
 - Constraints are handled by the NSGA-II algorithm but a solution that violates a constraint; even really close than an optimal feasible solution, will always be classified as a worse non-violated solution really far from the optimum. Therefore, optimum that corresponds to a solution where the constraint is binding will have little chance to be returned by the classic NSGA-II algorithm.

In fact, those two 2 remarks are closely related and have been met for the Heston calibration model:

- The objective function requires the computation of M complex integrals on $[0; \infty[$. The integral is the real part of a complex number and it cannot be computed analytically. Therefore, it is computed numerically using quadrature and the computation cost will be non-negligible.
- The global optimum corresponds to the one where the constraint is binding.

It is possible to refine the genetic algorithm in order to improve its convergence rate and make it more efficient when the optimum is found where the constraint is binding. Different refinements exist and among the most interesting, we can quote:

- Penalty based strategy: We add a penalty on the solution that violates the constraint(s). This penalty could evolve during the iterations in order to keep diversity among individuals at the beginning of the process. However, the calibration of the penalty is complex.
- Repair algorithms: The genetic operators are built in such a way that the new individuals will not violate the constraints. In practice, those operators are difficult to build and are highly problem dependent.
- Dominance-based tournament selection^[5]: This refinement could be used for multi-objective problem. We use the same selection process than the NSGA-II algorithm but a perturbation is added randomly (with low frequency) in order to take a random individual (that potentially violates the constraint(s)). Again, the calibration of the perturbation probability will be complex.

However, the primary goal of using a genetic algorithm was to keep it the more general possible without having to tune the algorithm. Instead of refining the genetic algorithm, we will choose to perform random analytical optimization during

iterations on a random chosen individual:

- The probability to perform an analytical optimization on a given iteration will be p_{Opti} :

$$u \leq p_{Opti} \text{ where } u \sim U[0,1]$$
- To model which individual will be chosen as the starting point of the local optimization, we first need to sort individuals according to their fitness. Then, a truncated geometric law is used with probability of success p . It corresponds to the probability of performing local optimization on the sorted individuals. If we suppose that we have n individuals, the probability of choosing the i^{th} individual will be:

$$\forall i \in [1, n], \mathbb{P}(X = i) = p * \frac{(1-p)^{i-1}}{\sum_{i=1}^n p * (1-p)^{i-1}} = \frac{(1-p)^{i-1}}{1 - (1-p)^n}$$

Finally, we need to find effective analytical algorithm to solve the non-linear optimization program. We suppose that we will approximate the objective function gradient by finite difference method.

Here are the general local optimization algorithms that we can use when the Hessian of the objective function is unknown:

- For non-constraint problem:
 - Quasi-Newton method: L-BFGS-B algorithm^[6]
- For constraint problem:
 - Modified augmented Lagrangian with a Sequential Quadratic Programming inner algorithm^[7].

3- Calibration and scenarios validation

Finally, we should not forget to validate our Calibration process. we could note two important validation steps in a Risk-Neutral projection context:

- Market consistency: This could be done after the Calibration process. We check that the price / volatility of the financial instruments quoted for the calibration are replicated by the model's price / volatility.
- Martingality: After the Simulation process, we can check if the quantities (Zero-Coupon or Equity prices, ...) are following a martingale.



Figure 3: Calibration validation: Comparison between Market and Model price and volatility

⁶R.H.Byrd, P. Lu and J. Nocedal. A Limited Memory Algorithm for Bound Constrained Optimization, (1995), SIAM Journal on Scientific and Statistical Computing, 16, 5, pp. 1190-1208.

⁷Y.Ye, Interior algorithms for linear, quadratic, and linearly constrained non linear programming, PhD Thesis, Department of EES Stanford University, Stanford CA.

⁵A.Coello Coello, Carlos, Mezura-Montes Efrén, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics, July 2002

CONCLUSION

ESG could be of great need in the Life Insurance business and the Solvency II framework. Indeed, it allows to compute Best Estimate and Financial Options and Guarantees even for complex products.

However, it should not hide the complexity of such a tool. The optimization process could have a dramatic impact on the scenarios outputted and lead to interpretation and computational errors. Misleading scenarios are due to several reasons; wrong choice of model or data, optimization algorithm...

The optimization process itself is complex because of the non-linear characteristic of the objective function. To solve this problem, the metaheuristic and more specifically the genetic algorithm could be a great choice. It could be applied to the vast majority of ESG models which makes it flexible. Finally, we have explained how to speed up convergence specifically with non-linear constraints problems by performing random analytical optimization during the evolution process of the genetic algorithm.



www.addactis.com
contact@addactis.com
+32 (0)2 526 13 10