Optimal Quantization : Evolutionary Algorithms vs Stochastic Gradient

S. Ben Hamida
CMAP, URA CNRS 756,
Ecole Polytechnique,France;
e-mail: *sana@cmapx.polytechnique.fr*

M. Mrad
Société Générale, France
and CERMSEM, Université Paris I, France
e-mail: *moez.mrad@sgcib.com*

Abstract- We propose a new method based on evolutionary optimization for obtaining an optimal L^p-quantizer of a multidimensional random variable. First, we remind briefly the main results about quantization. Then, we present the classical gradient-based approach (this approach is well detailed in [2] and [7] for p=2) used up to now to find a "local" optimal L^p-quantizer. Then, we give an algorithm that permits to deal with the problem in the evolutionary optimization framework and illustrate a numerical comparison between the proposed method and the stochastic gradient method. Finally, a numerical application to option pricing in finance is provided.

1 Introduction

Quantization is a *signal processing* technique that consists in approximating a random variable X with values in a continuous state space by an other random variable with values in a finite state space $\Gamma = \{x_1, ..., x_N\}$. A natural way to achieve this approximation is to project X on the grid Γ following the closest neighbour rule. This technique has been recently used to solve (via Monte carlo simulation) some high-dimensional problems arising in finance such as numerical integration [6], pricing of American options on a multi-dimensional underlying [2], pricing European options on a multi-dimensional underlying in the Uncertain Volatility framework, ... It has been shown in [2] that in order to obtain relevant results, one has to take a grid Γ that minimizes the L^p-mean error between the continuous variable and its quantized form. Up to now, this optimization problem was achieved using some gradient-based algorithms such as Lloyd's method I and Stochastic Gradient Algorithm (**SGA**) (for further details about these techniques one can see [7] for $p = 2$ and X gaussian). In high dimension ($d \geq 2$), the optimization problem has not a unique solution and an optimal quantizer provided by the algorithms mentioned above is in fact a local minimizer of the L^p-mean error.

In this paper, we suggest to use evolutionary optimization in order to obtain a "global" optimal L^p-quantizer of X . The evolutionary algorithms (**EA**s) are powerful stochastic zeroth order optimization algorithms based on a crude imitation of natural Darwinian evolution. Given an *objective function* to optimize over a search space E, they perform a random search in E in the hope to reach the global optimum. A detailed presentation of EAs is given in section 4.

This paper is organized as follows. Section 2 describes briefly the quantization of a continuous random variable. Section 3 presents the stochastic gradient method. In Section 4, we show how to deal with the optimization problem using (**EA**s). Sections 5 and 6 provide a numerical comparison between our method and a the stochastic gradient method.

2 Optimal L^p-quantization

We consider a μ-distributed random variable $X \in L^p_{\mathbb{R}}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The L^p-quantization ($p \geq 1$) consists in L^p-approximating X by a random vector $\widehat{X}$ taking its values in a finite grid $\Gamma = \{x_1, ..., x_N\}$ where $\forall i \in < 1, N >$, $x_i \in \mathbb{R}$. From signal processing we can prove that the best way for approximating or "quantizing" a random variable X using a grid Γ is to project X on Γ following the nearest neighbour rule. This leads to a Borel partition $\{C_j(\Gamma)\}_{1 \leq j \leq N}$ of $\mathbb{R}$ called Voronoï tesselation of X.

The Voronoi quantizer of X is then given by:

$$\widehat{X} = \sum_{i=1}^{N} x_i \mathbf{1}_{\mathbf{C_i(\Gamma)}}(\mathbf{X})$$

We say that $\Gamma = (x_1, ..., x_N)$ is an optimal L^P-quantizer of X, if Γ solves the problem of minimizing the L^P-mean

of the quantization error:

$$\|X-\widehat{X}\|_p = \left. \sum_{i=1}^{N}\mathbb{E}(\mathbf{1}_{\mathbf{C_i(\Gamma)}}(\mathbf{X})\|\mathbf{X}-\mathbf{x_i}\|^{\mathbf{p}})\right)^{\frac{1}{p}}$$

Instead of minimizing the L^p-mean error, we usually work for simplicity with the quantity $D_N^{X,p}(\Gamma) = \mathbb{E}(\min_{1\leq i\leq N}\|X - x_i\|^p)$, called the L^P-distorsion. The function $\Gamma \rightarrow D_N^{X,p}(\Gamma)$ is continuous and reaches a minimum denoted $\underline{D}_N^{X,p}$ at some N-tuple having N pairwise distinct components Furthermore, it is easy to establish that this minimum $\underline{D}_N^{X,p}$ decreases to 0 as the size N of the optimal quantizer goes to infinity (see e.g. [4, 6] for a proof of these properties).

3 Stochastic Gradient Algorithm

Stochastic gradient methods are based on the integral representation of the criterion to be optimized, which is the case with distortion : $\Gamma \rightarrow D_N^{X,p}(\Gamma)$. Let $(w^t)_{t\in\mathbb{N}^*}$ be a sequence of iid μ-distributed random variables and $(\gamma_t)_{t\in\mathbb{N}^*}$ a sequence of positive steps satisfying $\sum_t \gamma_t = +\infty$ and $\sum_t \gamma_t^2 < +\infty$. Then starting from an initial N-tuple Γ^0 with N pairwise distinct components, set:

$$\Gamma^{t+1} = \Gamma^t - (\gamma_{t+1}/p)H_p(\Gamma^t, w^{t+1}) \qquad (3.1)$$

Formula (3.1) can be developed as follows if one sets $\Gamma^t := \left(x^{1,t}, ..., x^{N,t}\right)$:
Let $i\,(t+1) \in \arg\min_i \left|x^{i,t} - w^{t+1}\right|$. Then, we have:

$$\begin{cases} x^{i(t+1),t+1} = x^{i(t+1),t} - \gamma_{t+1}\ x^{i(t+1),t} - w^{t+1\ p-1}\ \frac{x^{i(t+1),t}-w^{t+1}}{|x^{i(t+1),t}-w^{t+1}|} \\ x^{i,t+1} = x^{i,t}, i \neq i\,(t+1) \end{cases}$$

The choice of the descent step γ_t is crucial, for further details about this issue one can see [7, 4] for the optimal quadratic quantization of a gaussian law.

4 Optimal L^p-Quantization with the EA

As described in section 3, the SGA works with a single solution and gives a local optimum close to the starting point. The aim of using EAs is to maintain a set of solutions instead of one, that are manipulated competitively by some variation operators, in order to perform a parallel search over the search space E.

4.1 Evolutionary algorithms: a brief overview

Let $\Pi_t = (\Gamma_1^t, \Gamma_2^t, \cdots, \Gamma_M^t)$ denote the population at the generation t, where $M \in \mathbb{N}$ is the population size and Γ_i^t is a potential solution to the problem. The first population Π_0 is initialized randomly on the search space E. Then, the population evolves by cycles of mutation/recombination/selection which tends to decrease the fitness. The following procedure describes a simple structure of an evolutionary algorithm [8].

Structure of an Evolutionary Algorithm

$t \leftarrow 0$
initialize population $\Pi_0 = (\Gamma_1^0, \Gamma_2^0, \cdots, \Gamma_M^0)$
evaluate (Π_0)
while (**not** stopping-condition) **do**
begin
    $t \leftarrow t+1$
    select V_t from Π_{t-1}
    $W_t \leftarrow$ alter V_t (mutation+recombination)
    evaluate W_t
    select Π_t from W_t
end

4.2 Evolutionary Algorithms for optimal L^p-quantization

Recall that to obtain an optimal L^p-quantizer of a random variable $X \in L^p_{\mathbb{R}}$, we should search for a Voronoï tessellation of this variable in $(\mathbb{R})^{\mathbb{N}}$ that minimizes the L^p-distortion. The following sub-sections explain how to use an EA for optimal L^p-quantization.

4.2.1 Computation of the objectif function

Let $f(\Gamma_m^t)$, $m \in \langle 1, M\rangle$ denote fitness (i.e. objectif function value) of the individual Γ_m^t. For the optimal quantization, the fitness of a given individual $\Gamma_m^t = (x_m^{1,t}, x_m^{2,t}, \cdots, x_m^{N,t})$, $m \in \langle 1, M\rangle$ will be considered as equal to its distortion.

To estimate the distortion by Monte-Carlo simulation, we consider $(X^{(l)})_{l\in<1,L>}$ a set of $L \in \mathbb{N}$ (in practice $L >> N$) iid realizations of the random variable X in $(\mathbb{R})^{\mathbb{N}}$. From now on, This set is named the *Reference Sample*. The objectif function to optimize is then given by:

$$\widehat{f}(\Gamma_m^t) = \frac{1}{L}\sum_{l=1}^{L} \left. \sum_{i=1}^{N}\mathbf{1}_{\mathbf{C_i(\Gamma_m^t)}}(\mathbf{X^{(l)}})\|\mathbf{X^{(l)}}-\mathbf{x_m^{i,t}}\|^{\mathbf{p}}\right) \qquad (4.1)$$

4.2.2 The evolution scheme

A set of genetic operators define the dynamic evolution of the population. Two kinds of operators are used: selection operators and variation operators. Selection operators determine candidates for reproduction and replacement. Variation operators (mutation and crossover) are generally stochastic operators used to produce new individuals by combining and perturbing the information contained in the parents.

Selection

The selection step is performed twice in each generation for both crossover and new population construction. We start with the generation Π_{t-1}, we use first selection procedure to choose a set of individuals V_t from Π_{t-1} for reproduction. The selection is applied a second time to choose which individuals from W_t will be part of the new population Π_t. The second step is also called replacement.

The probability $ps(\Gamma_m^t)$ for an individual Γ_m^t, $m \in \langle 1, M\rangle$ to be selected is given by : $ps(\Gamma_m^t) = \frac{e^{-f(\Gamma_m^t)}}{\sum_{i=1}^{M} e^{-f(\Gamma_i^t)}}$ We Notice that this probability of selection is increasing with the fitness.

Crossover

Let Γ^k and Γ^s be two selected grids in V_t. A new offspring Γ^r is created, with a probability p_c by merging the contained information in the parents Γ^k and Γ^s [5] as follows : $\Gamma^r = \alpha\Gamma^k + (1-\alpha)\Gamma^s$ where α is an N-tuple of $[0,1]^d$-uniform random variables and the multiplication is considered component-wise.

Mutation Each element Γ_m in W_t has a probability p_m to be mutated and gives birth to a new individual $\Gamma^{m\prime}$, that will replace Γ^m in W_t: $\Gamma'_m = \Gamma_m + \varepsilon$, where ε is a N-tuple of $\mathbb{R}$ independent random Gaussian variables with a mean of zero and a variable standard deviation β_t. The parameter β_t itself is subject to mutation in order to scale the movement of the grid points on the search space along evolution.

5 Numerical tests for Normal distribution

In this section, we specialize the discussion to the optimal quadratic quantization of a d-dimensional Gaussian vector. We compare the grids obtained by the EA to those obtained by the SGA in terms of final distortion and geometric symmetry.

In order to obtain a "good" optimal grid with the SGA, we use the same sequence $(\gamma_t)_{t\in\mathbb{N}^*}$ as in [7, § 3.2.2] (This choice is inferred from a work done in this paper on quantization of $[0,1]^d$-Uniform law). We also mention the following important issue highlighted in [7]. Actually, the simulation of points with too large norms may cause dramatic effects on the procedure (3.1) when the step γ_t is not yet small enough. In order to avoid this, we will (first) simulate some spherically truncated Normal variables (w^t) (calibrating the threshold radius so as to keep at least 99% of their mass). This truncation has a stabilizing effect on the procedure. Then, to get a quantization of the original Normal distribution, instead of doing like in [7], i.e. completing the optimization by processing a Lloyd's method I with non truncated Normally distributed random numbers, we continue to use the procedure (3.1) with non truncated Normal variables (w^t) (since, for large values of t, γ_t becomes small). One verifies that, when the number of points is large, this only affects the location of the peripheral points. On the other hand, as expected, it slightly increases the distortion (but it produces more accurate results for numerical integration of course).

For stability reasons, we also do a similar work when using the EA. Actually we will start first by using a truncated Reference Sample, in order to estimate the fitness (i.e. the distortion) of a given individual (i.e. grid) (The truncated Reference Sample, is obtained from the (non-truncated) Reference Sample, by keeping only the points of in a hyper-sphere of which the radius is calibrated so as to keep at least 99% of the mass). Then, to get a quantization of the original Normal distribution, we can either use the procedure (3.1) with non truncated Normal variables (w^t) (starting with a small γ_0) or use an EA with a non-truncated Reference Sample (We use the initial Reference sample from which the truncated Reference Sample was extracted) and with an initial population equal to the final population of the previous EA.

Before going further, let's introduce on some notations:

SGA (Tr)	Using SGA with truncated Normal variables w^t .
SGA (n-Tr)	Using SGA with non-truncated Normal variables w^t .
EA (Tr)	Using EA with truncated Reference Sample.
EA (n-Tr)	Using EA with non-truncated Reference Sample.

We then, introduce two procedures that we will use to

obtain an optimal quantizer of the Normal distribution. The first procedure uses the gradient optimization method (SGA). It starts with a random grid and arrives at grid(1) by applying SGA with truncation (SGA(Tr)). Then, using Grid(1) as a starting point, the optimization procedure continue without truncation (SGA(n-Tr)) and arrives at Grid (1 bis).

Similarly, Procedure 2 involves using EA(Tr) (EA with truncation), starting with a random population and arriving at population (2) with the best element in Grid(2). Then, using population (2) as a starting population, the evolution continue without truncation (EA(n-Tr)) and arrive at population (2 bis) with the best element in Grid (2bis).

We give below different detailed results for dimensions 2 and 3. We also give some other results showing the behavior of the difference between the minimal distortions given by the two algorithms when the dimension d increases (from 1 to 7). The values of EA parameters described in section 4 are given in the following table. These parameters were found by preliminary numerical experimentations.

EA parameter settings	
Population size (M)	70
Crossover Probability (p_c)	0.6
Mutation Probability (p_m)	0.9
Initial standard deviation for mutation (β_0)	0.03
Maximum Number of Generations (t_{max})	800 for EA (Tr)
	1200 for EA (n-Tr)
SGA parameter settings	
Descent-step (γ_t)	As in [7]
Maximum Number of iterations (t_{max})	4 000 000 for SGA (Tr)
	6 000 000 for SGA (n-Tr)

For the EA (Tr) (or SGA (Tr)) step, we use a truncated Reference Sample containing 50 000 realizations of the d-dimensional random variable under study. of which the norms are smaller that 3 (we consider 3 as radius threshold).

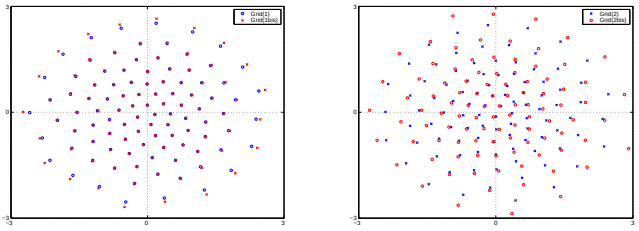
Dimension 2 and 3

We see in Figure 1 below, that the use of a non-truncated step (i.e EA (n-Tr) or SGA (n-Tr)) after a truncated one, affects essentially the peripheral points of the grid:

The distortions of the different obtained optimal grids are summarized in the table 1.

Distortion as a function of dimension

In this paragraph, we are interested in the impact of the dimension on the distortion level obtained with EA and SGA



Superposition Grid (1) vs Grid (1bis) Superposition Grid (2) vs Grid (2bis)

Figure 1: Superposition of the grids obtained with procedure 1 (left) and procedure 2 (right) with N=100 and d=2.

	Dimension 2	Dimension 3
Grid (1)	$D_{final}^{Tr} = 0.03297$	$D_{final}^{Tr} = 0.20439$
Grid(1 bis)	$D_{final}^{n-Tr} = 0.03987$	$D_{final}^{n-Tr} = 0.24488$
Grid (2)	$D_{final}^{Tr} = 0.03285$	$D_{final}^{Tr} = 0.19435$
Grid (2 bis)	$D_{final}^{n-Tr} = \mathbf{0.03855}$	$D_{final}^{n-Tr} = \mathbf{0.22987}$

Table 1: Distortions of the optimal grids with d=2 and d=3

approaches. Experiments done with N=14 show that the spread between the distortions obtained by the two types of algorithms increases with dimension. This spread becomes significant when the dimension exceeds 3. The distortion obtained by the EA is always smaller than the one obtained by the SGA. The figure 2 gives the relative spread between the distortions of Grid (1 bis) and Grid (2 bis) defined by : *(Distortion of Grid (1 bis) - Distortion of Grid (2 bis))/Distortion of Grid (2 bis)* as a function of dimension.

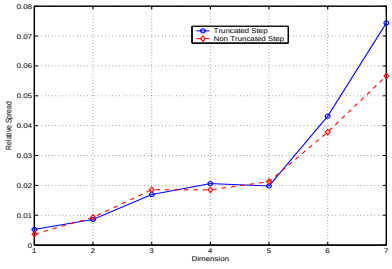


Figure 2: Relative spread as function of dimension for N=14.

So, when dealing with some high-dimensional numerical probabilistic problems, one can expect that using EAs instead of gradient based algorithms in order to compute the optimal quantizers will give more accurate results.

6 Application to the pricing of an American option

In this section, we focus on the problem of approximating the value of an optimal stopping problem by the *quantization tree method* introduced by Bally, Pagès and Printemps [1].

The quantization tree method consists in approximating a continuous-time process with values in a continuous state space by a discrete-time process with a finite state space. At each time step the finite state space is obtained by optimal quantization. The transition matrix of the approximating discrete-time process is usually obtained by classical Monte Carlo technique. The main objective of this section is to use an EA instead of a SGA for generating the optimal grids, and to analyze the performance of this method.

Numerical Results

Wee consider a bermudan option on the geometric mean of d assets : $f(s) := \left[K - \left(\prod_{i=1}^{d} s^i\right)^{\frac{1}{d}}\right]^+$. In the multi-dimensional Black and Scholes framework, the geometric mean of d non-dividend lognormal processes is equivalent to a particular lognormal process with a dividend yield.
The following are the parameters used for the simulations :

instantanious interest rate r	0.06
volatility	$\sigma_i = 0.3$ and $\rho_{ij} = 0.5$
maturity T	1
initial values	$S_0^i = 36$
strike K	40
time step	$1/5$

In order to price this bermudan option by quantization tree method, one need a set of grids $S = \{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}$, where each grid is associated to a time step $T_i = \frac{iT}{m}, (i = 1, \cdots, m)$. We consider 3 kinds sets of grid's set $S_j, j = 1, 2, 3$: The grids of the first (resp. second) set are obtained via Stochastic Gradient Algorithm (resp. Evolutionary Algorithm). Whereas, each grid of the third set S_j is a random grid sampled with respect to the underlying dynamics (no optimization).

We compare the prices obtained using quantization tree method with each one of these sets of grids to those obtained by the finite difference method.

We see that, the prices obtained with a set of optimal grids (i.e. computed with an EA) are closer to the true prices (Especially for dimensions 3 and 4) than the prices obtained with a set of local optimal grids (i.e. computed with an SGA) or random grids. We also remark that the difference between the EA-based prices and the other ones increases as the dimension rises.

Grids	Mean	Std. dev.	$\frac{\text{Std. dev.}}{\text{True price}}$
Dimension 1 (True price = 5.6571)			
Random	5.6540	0.0020	0.035%
SGA	5.6606	0.0017	0.003%
EA	5.6604	0.0017	0.003%
Dimension 2 (True price = 5.2642)			
Random	5.2550	0.0021	0.040%
SGA	5.2674	0.0020	0.038%
EA	5.2639	0.0026	0.049%
Dimension 3 (True price = 5.1192)			
Random	5.0901	0.0020	0.039%
SGA	5.0887	0.0019	0.037%
EA	5.1102	0.0026	0.051
Dimension 4 (True price = 5.0432)			
Random	5.0079	0.0039	0.078 %
SGA	5.0033	0.0028	0.056 %
EA	5.0407	0.0025	0.050 %

Table 2: Bermuda option prices computed with random and optimized grids

Bibliography

[1] V. Bally and G. Pagès (2005). A quantization algorithm for solving discrete time multi-dimensional optimal stopping problems, *Bernoulli*. 11(5), 893-932.

[2] V. Bally, G. Pagès and J. Printems (2005). A quantization method for pricing and hedging multi-dimensional American style options, Mathematical Finance. 15(1), 119-168.

[3] J. Bucklew and G. Wise (1997). Multidimensional Asymptotic Quantization Theory with r^{th} Power distorsion Measures, *IEEE on information Theory: Special issue on quantization* **28**, n^o 2, 239-277.

[4] S. Graf and H.Luschgy. Foundations of quantization for probability distributions, *Lecture Notes in Mathematics n^{o}1730*, Springer, 230.

[5] Z Michalewicz (1996). *Genetic Algorithms+Data Structures=Evolution Programs*, Springer Verlag.

[6] G. Pagès (1997). A space vector quantization method for numerical integration, *J. of Applied and Computational Mathematics* **89**, 1-38.

[7] G. Pagès and J. Printems (2003). Optimal quadratic quantization for numerics: the Gaussian case, Monte Carlo Methods & Applications. 9(2), 135-166.

[8] M. Schoenauer and Z. Michalewicz (1997). *Evolutionary Computation: An Introduction* , Control and Cybernetics, 287-299, Springe.