ECOLE POLYTECHNIQUE

CENTRE DE MATHÉMATIQUES APPLIQUÉES
UMR CNRS 7641

91128 PALAISEAU CEDEX (FRANCE). Tél: 01 69 33 41 50. Fax: 01 69 33 30 11

http://www.cmap.polytechnique.fr/

Optimal Quantization using Evolutionary Algorithms

Sana Ben Hamida, Moez Mrad

R.I. N^0 564 *January 2005*

Optimal Quantization using Evolutionary Algorithms

Sana BEN HAMIDA
Centre de Mathématiques Appliquées (France)
Sana@cmapx.polytechnique.fr
Ecole Supérieure de Technologies
et Informatique (Tunisie)
Sana.BenHamida@esti.rnu.tn

Moez MRAD
CERMSEM, Université Paris I
Paris, France
Société Générale
IR & Credit Quantitative Research
moez.mrad@sgcib.com

Abstract

We propose a new method based on evolutionary optimization for obtaining a "global" 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 [4] and [15] 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 . Finally, we provide a numerical comparison between our method and the stochatic gradient method.

Key words: Optimal quantization, Evolutionary optimization, Stochastic Gradient Algorithms.

2000 MSC classification: primary: 94A29 ; secondary: 62L20, 65D30, 91B28.

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 [14], pricing of American options on a multi-dimensional underlying [4], pricing European options on a multi-dimensional underlying in the Uncertain Volatility framework [12], ... It has been shown in [4] that in order to obtain relevant results (good estimations), 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 [15] for $p = 2$ and X gaussian). In high dimension ($d \geq 2$), the optimization problem has not a unique solution [6] 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 perfom a random search in E in the hope to reach the global optimum [8, 1]. It is based on some artificial evolution of a population of *individuals* (P-uple of points over E). At each *generation* (step), individuals undergo some variation, giving birth to a new population. The set of new individuals is created by the process of selecting individuals according to their *fitness* (value of the objective function in the problem domain) and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment (*fitness*) than the individuals that they were created from, just as in natural adaptation. A more detailed presentation of EAs is given in section 4.

As mentioned above Pagès and Printemps gave in [15] some numerical procedures in order to get a local optimal *quadratic* ($p = 2$) quantizer of gaussian law. By comparison to this work the approach presented in this paper has a wider scope since it permits to obtain a global optimal L^p-quantizer for any $p \geq 1$ and any random variable $X \in L^p_{\mathbb{R}^d}$ provided that we are able to simulate it.

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 an evolutionary algorithm. 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 $X \in L^p_{\mathbb{R}^d}$ 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 $\forall i \in < 1, N >$, $x_i \in \mathbb{R}^d$. 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 $\mathbb{R}^d$ called Voronoï tesselation of X satisfying:

$$C_j(\Gamma) \;\subset\; \left\{ y \in \mathbb{R}^d \;:\; |x_j - y| = \min_{1\leq k\leq N_i} |x_k - y| \right\}$$

where $|.|$ represents the Euclidean norm on $\mathbb{R}^d$.
The Voronoi quantizer of X is then given by:

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

The grid or N-tuple $\Gamma \in \left(\mathbb{R}^d\right)^N$ is also called the quantizer of 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:

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

Instead of minimizing the L^p-mean error, we usually work for simplicity with the quantity $D_N^{X,p}(\Gamma) = \mathbb{E}(\min_{1\leq j\leq N}\|X - x_j\|^p)$, called the L^P-distorsion. The function $\Gamma \to D_N^{X,p}(\Gamma)$ is continuous and reaches a minimum denoted $\underline{D}_N^{X,p}$ at some N-tuple having N pairwise distinct components and lying in the convex hull of the support of the distribution of μ. 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. [7, 14] for a proof of these properties). The rate of convergence toward 0 of the distortion found in several steps by Zador [18], Bucklew & Wise [5] and finally Graf & Luschgy [7], is given by the following theorem:

Theorem .1 *Assume that $X \in L_{\mathbb{R}^d}^{p+\eta}(\Omega, \mathcal{F}, \mathbb{P})$ for some $\eta > 0$. Then*

$$\lim_N (N^{\frac{p}{d}} \underline{D}_N^{X,p}) = J_{p,d}\, \|\varphi\|_{\frac{d}{d+p}}$$

where $\mu(du) = \varphi(u).\lambda_d(u) + \overline{\mu}$, $\overline{\mu} \perp \lambda_d$ (λ_d is the Lebesgue measure on $\mathbb{R}^d$) and $J_{p,d}$ is a constant depending on p and d.

From [4, 14], we can see that working with optimal grids (or at least grids of which the distorsions are as low as possible) reduces the approximation error while estimating the results via Monte Carlo simulation. In the next two sections, we focus on how to obtain these optimal grids. The first one, will present the SGA which is one of the most used algorithms in practice to deal with this problem (for a further details, one can see [4] and [15] for p=2). Whereas the second one, will be dedicated to the resolution of this problem with an EA.

3 Stochastic Gradient Algorithm

Stochastic gradient methods are based on the integral representation of the criterion to be optimized. This is the case with distorsion, actually the function $\Gamma \to D_N^{X,p}(\Gamma)$ is continuously differentiable for any $p > 1$ (see e.g., [7] when p=2 or [14]) and the gradient can be obtained by formal differentiation:

$$\nabla D_N^{X,p}(\Gamma) = \int H_p(\Gamma, u)\mu(du)$$

where $H_p(\Gamma, u) = (H_p(x_i, u))_{1 \le i \le N}$ and

$$H_p(x_i, u) = p\,|x_i - u|^{p-1} \frac{x_i - u}{|x_i - u|} \mathbf{1}_{\{|x_i - u| < \min_{j \ne i} |x_j - u|\}} \text{ for all } 1 \le i \le N$$

When the dimension d is bigger than 2, the computation of $\nabla D_N^{\mu,p}(\Gamma)$ is time-consuming and the use of a deterministic gradient descent is not feasable. Instead we use a stochastic gradient descent defined as follows: Let $\left(w^t\right)_{t \in \mathbb{N}^*}$ 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:

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

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:

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

The choice of the descent step γ_t is crucial, for further details about this issue one can see [15] for the optimal quadratic quantization of a gaussian law. Furthermore, it is established in [7] that, if $X \in L_{\mathbb{R}^d}^{p+\eta}(\Omega, \mathcal{F}, \mathbb{P})$ for some $\eta > 0$, then the sequences $\left(\delta^t\right)_{t \ge 1}$ and $\left(\alpha^{i,t}\right)_{t \ge 1}$ of random variables recursively defined by

$$\delta^{t+1} := \delta^t\,(1 - \gamma_{t+1}) + \gamma_{t+1} \frac{x^{i(t+1),t} - w^{t+1}}{\left|x^{i(t+1),t} - w^{t+1}\right|} \left|x^{i(t+1),t} - w^{t+1}\right|^{p-1}, \quad \delta^0 := 0 \tag{3.2}$$

$$\alpha^{i,t+1} := \alpha^{i,t}\,(1 - \gamma_{t+1}) + \gamma_{t+1} \mathbf{1}_{\{i = i(t+1)\}}, \quad \alpha^{i,0} := 1/N, \quad 1 \le i \le N \tag{3.3}$$

satisfy on the event $\left\{\Gamma^t \to \Gamma^*\right\}$

$$\delta^t \to D_N^{X,p}\left(\Gamma^*\right)/p \quad \text{and} \quad \alpha^{i,t} \to \mu\left(C_i\left(\Gamma^*\right)\right), \; 1 \le i \le N \;\; \text{as } t \to \infty$$

These costless procedures yield the parameters (weights of the Voronoi cells, L^p-quantization error of Γ^*) necessary to exploit the N-tuple Γ^* for numerical purpose.

Concerning practical implementations of the algorithm, it is to be noticed that, when $p = 2$ at each, step the N-tuple Γ^{t+1} lives in the convex hull of Γ^t and w^{t+1} which has a stabilizing effect on the procedure.

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. This characteristics are very helpful in exploring multiple search directions and thus increasing the chance to reach the global optimum.

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 [16].

Structure of an Evolutionary Algorithm

```
t ← 0
initialize population Π_0 = (Γ_1^0, Γ_2^0, ··· , Γ_M^0)
evaluate (Π_0)
while (not stopping-condition) do
begin
    t ← t + 1
    select V_t from Π_{t−1}
    W_t ← alter V_t (mutation+recombination)
    evaluate W_t
    select Π_t from W_t
end
```

From mathematical point of view, this algorithm corresponds to an (inhomogeneous) Markov chain $(\Pi_t)_{t\in\mathbb{N}}$ in E^M:

$$\Pi_{t-1} \underset{selection}{\longrightarrow} V_t \underset{mutation+recombination}{\longrightarrow} W_t \underset{selection}{\longrightarrow} \Pi_t,$$

where the initial population is an iid sample on E, and individuals evolve following different transition kernels for mutation and recombination.

4.2 Evolutionary Algorithms for optimal L^p-quantization

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

4.2.1 Generating the initial population

The first step in the evolutionary algorithm is to generate a family of random grids (individuals) on the search space $E = (\mathbb{R}^d)^N$ at generation $t = 0$. The initialization process is similar to the one used for the SGA, with the difference that we start with set (or a population) of M initial grids (instead of one): $\Pi_0 = (\Gamma_1^0, \Gamma_2^0, \cdots, \Gamma_M^0)$ where each $\Gamma_m^0 \in (\mathbb{R}^d)^N$, $m \in \langle 1, M\rangle$ is generated randomly[1] on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

[1]Each initial grid can be for example composed of N iid samples of X. In such case we speak about self-qantization.

4.2.2 Computation of the objective function

Let $f(\Gamma_m^t),\ m \in \langle 1, M\rangle$ denote fitness of the individual Γ_m^t. The fitness measure f of an individual is usually directly related to its objective function value. 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 distorsion (The distorsion of a grid was introduced in Section 2):

$$\begin{aligned} f(\Gamma_m^t) &= D_N^{X,p}(\Gamma_m^t) &= \mathbb{E}\left(\min_{1\le j\le N} \left\|X - x_m^{j,t}\right\|^p\right) \\ & &= \left(\sum_{i=1}^N \mathbb{E}(\mathbf{1}_{C_i(\Gamma_m^t)}(X)\|X - x_m^{i,t}\|^p\right) \end{aligned}$$

We want to estimate the distorsion 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 $(\mathbb{R}^d)^N$. From now on, This set is named the *Reference Sample*. The objective function to optimize is then given by:

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

4.2.3 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.

<u>Selection</u>

The selection step is performed twice in each generation. We start with the generation Π_{t-1}, we use first selection procedure to choose a set of individuals V_t from for reproduction. By applying variation operators (that we will see in the two paragraphs below) to the V_t we obtain a second set W_t. Then, we use the selection procedure for the second time to choose which individuals from W_t will be part of the new population Π_t [2]. 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[2] is given by :

$$ps(\Gamma_m^t) = \frac{e^{-f(\Gamma_m^t)}}{\sum_{i=1}^M e^{-f(\Gamma_i^t)}} \tag{4.2}$$

[2]Either for reproduction, i.e. form Π_{t-1} to V_t or for replacement, i.e. from W_t to Π_t

We recall here that $f(\Gamma_m^t) = D_N^{X,p}(\Gamma_m^t)$. We Notice that this probability of selection is increasing with the fitness. We also remark that after replacement, the frequencies of better individuals have increased at the expense of worse individuals.

<u>Crossover</u>

The crossover operator is a function from E^k (in most cases $k = 2$) into E. It allows parents to exchange some *genetic material* to build up new solutions. The motivation being that we can use good solutions to develop even better ones. It is considered as an exploitation operator.

In the case of $k = 2$, a "couple" of elements from V_t (selected individuals for reproduction) has a probability p_c to be crossed. Let Γ^k and Γ^s be two selected grids[3] in V_t. A new offspring Γ^r is created by merging the contained information in the parents Γ^k and Γ^s. There are many ways to merge information [10]. We mention here the case of the linear recombination that creates a new individual Γ^r by :

$$\Gamma^r = \alpha\Gamma^k + (1-\alpha)\Gamma^s$$

where α is an N-tuple of $[0,1]^d$-uniform random variables and the multiplication is considered component-wise. For L^p-Quantization, the use of this crossover operator means that each point of the produced grid Γ^r is a result of a linear recombination of the corresponding points in the parent grids. Γ^r is then added to W_t to undergo mutation.

<u>Mutation</u>

Mutation is a stochastic operator from E into E, of which the idea is to perturb a given grid in order to make a local step on the search space. Its is seen as an exploration operator [17]. 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. The mutation is achieved by adding a normally distributed random noise to Γ^m:

$$\Gamma'_m = \Gamma_m + \varepsilon$$

where ε is a N-tuple of $\mathbb{R}^d$ 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. This mutation is based on the strategy of self-adaptation of standards deviations [1]

[3]Note there is not reference to the time here, since we are performing intermediate operations to go from generation $t-1$ to generation t. The crossover operation (it also the same issue with the mutation operator) is completely defined as an operation on a given group (here a pair) of elements of a given set of grids

Stopping criteria

To stop the evolution process, we use a simple stopping criterion based on the generation counter t. We define a maximum number of generations (t_{max})for the whole evolution and a maximum number of generations without improvement of the best individual. Both criteria are checked each iteration.

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 distorsion and geometric symmetry.

In order to obtain a "good" optimal grid[4] with the SGA, we use the same sequence $(\gamma_t)_{t\in\mathbb{N}^*}$ as in [15, § 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 [15]. 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 $\left(w^t\right)$ (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 [15], i.e. completing the optimization by processing a Lloyd's method I with nontruncated Normally distributed random numbers, we continue to use the procedure (3.1) with non truncated Normal variables $\left(w^t\right)$ (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 in order to obtain an optimal grid. Actually we will start first by using a truncated Refrence Sample[5], in order to estimate the fitness (i.e. the distorsion) of a given individual (i.e. grid) (The truncated Refrence Sample, is obtained from the (non-truncated) Refrence 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 $\left(w^t\right)$ (starting with a small γ_0) or use an EA with a non-truncated Reference Sample (We use the initial Reference sample from

[4] i.e uniformly symmetric in relation to the point 0.

[5] The Reference Sample was introduced in Sub-section 4.2.2

which the truncated Reference Sample was extracted) and with an intial population equal to the final population of the previous EA.

Before going further, let's indroduce on some notations that will be helpful in the interpretation of the obtained results. We start by the four notations in the table below:

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 four procedures that we will use to obtain an optimal quantizer of the Normal distribusion under study:

	— Chronological order ⟶			
	EA (Tr)	EA (n-Tr)	SGA (Tr)	SGA (n-Tr)
Procedure 1	No	No	Yes Start: Random grid End: Grid (1)	Yes Start: Grid (1) End: Grid (1 bis)
Procedure 2	Yes Start: Random population End: Population (2). Its best element is denoted Grid (2).	Yes Start: Population (2) End: Population (2 bis). Its best element is denoted Grid (2 bis).	No	No
Procedure 3	Yes Start: Random population End: Population (2). Its best element is denoted Grid (2).	No	No	Yes Start: Grid (2) End: Grid (3)
Procedure 4	Yes Start: Random population End: Population (2). Its best element is denoted Grid (2).	Yes Start: Population (2) End: Population (2 bis). Its best element is denoted Grid (2 bis).	No	Yes Start: Grid (2 bis) End: Grid (3 bis)

Table 1: The four used optimization procedures

We applied these four procedures in order to obtain an optimal quantizer of a d-dimensional standard Normal distribution. 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 distorsions given by the two algorithms when the dimension d increases (from 1 to 6). The values of EA parameters described in section 4 are giving in table 2:

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)

Table 2: EA Parameter settings

For the EA (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). This set is built iteratively. Each time we generate a random vector. If its norm is smaller than the threashold radius, then we add it to the Reference Sample else we reject it and put it an other set that we call Tail Sample. This operation is repeated till the size of the truncated Reference Sample reaches 50 000. For the EA (n-Tr) step, the used non-truncated Reference Sample will be the union of the truncated Reference Sample and the Tail Sample.

The parameters of the SGA are given in table 3:

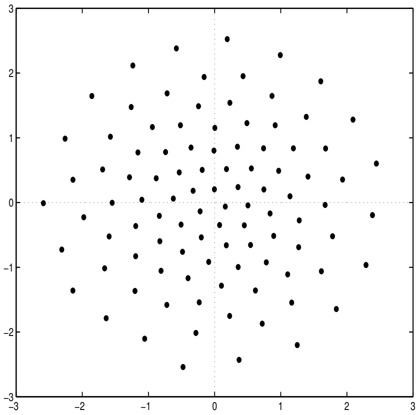
Descent-step (γ_t)	As in [15]
Maximum Number of iterations (t_{max})	4 000 000 for SGA (Tr)
	6 000 000 for SGA (n-Tr)

Table 3: SGA Parameter settings

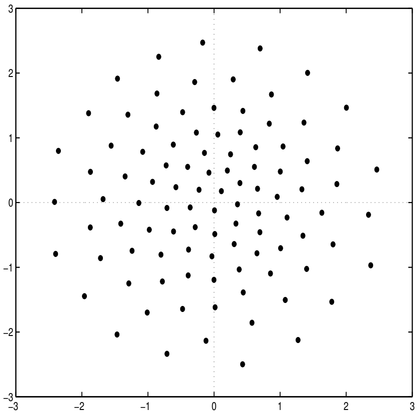
Remark .1 *In order to compare the two optimization approches, the distorsions of the optimal grids are calculated empirically using the truncated Reference Sample for the EA (Tr) and the SGA (Tr) steps (theses distorsions will be denoted D^{Tr}_{final})and the non-truncated Reference Sample for the EA (n-Tr) and SGA (n-Tr) steps (These distorsions will be denoted D^{n-Tr}_{final}). Note that as in [15], it is also possible for the optimal grids obtained by an SGA-based procedure to obtain an estimation of the distorsion with* (3.3). *Nevertheless, we numerically noticed that this "companion" distorsion is in fact a biased estimator of the distorsion an can be different from the distorsion calculated empirically used a Reference Sample.*

Dimension 2

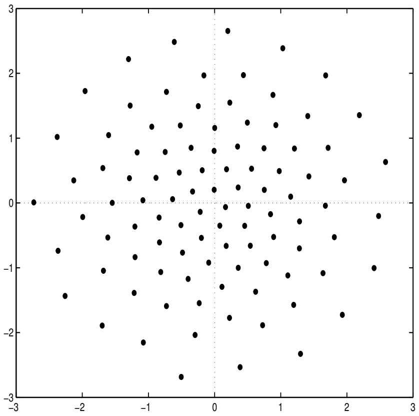
We give here the 6 grids introduced in Table 1 and their final distorsions:



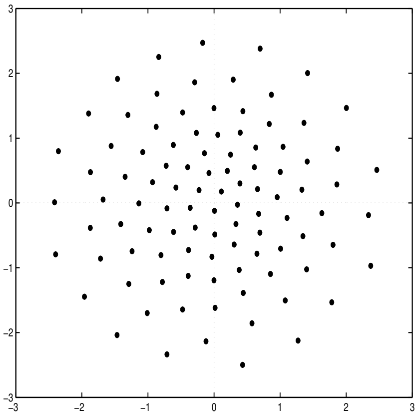
Grid (1), $D^{Tr}_{final} = 0.03297$



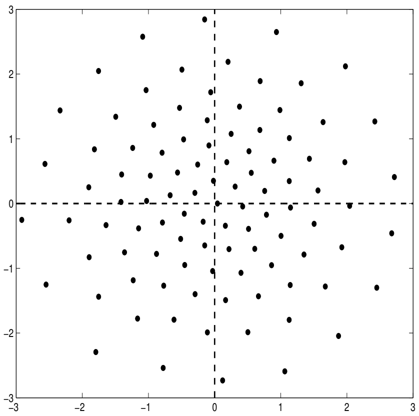
Grid (2), $D^{Tr}_{final} = 0.03285$



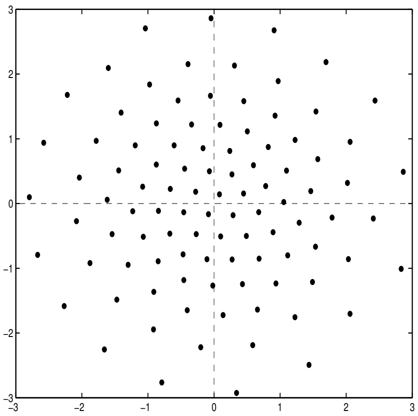
Grid (1 bis), $D^{n-Tr}_{final} = 0.03987$



Grid (2 bis), $D^{n-Tr}_{final} = 0.03855$



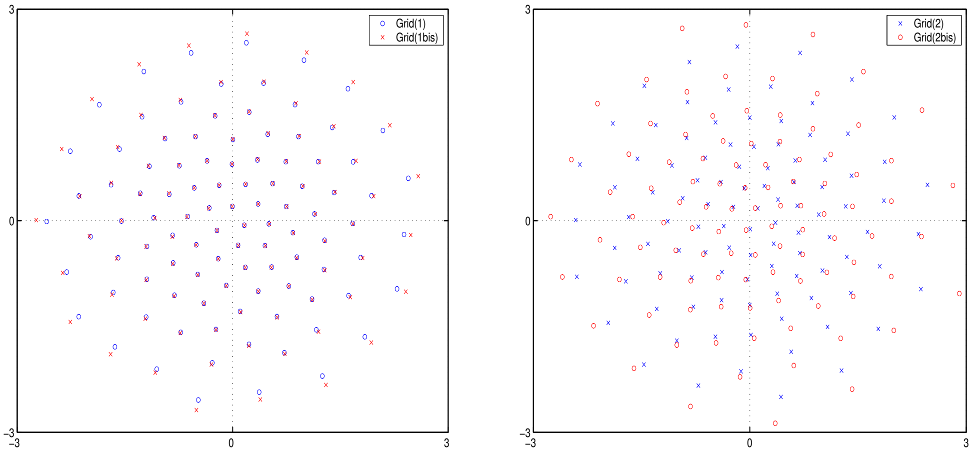
Grid (3), $D^{n-Tr}_{final} = 0.039605$



Grid (3 bis), $D^{n-Tr}_{final} = 0.03847$

Figure 1: Optimal grids for $N = 100$

We see in Figure 2 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:

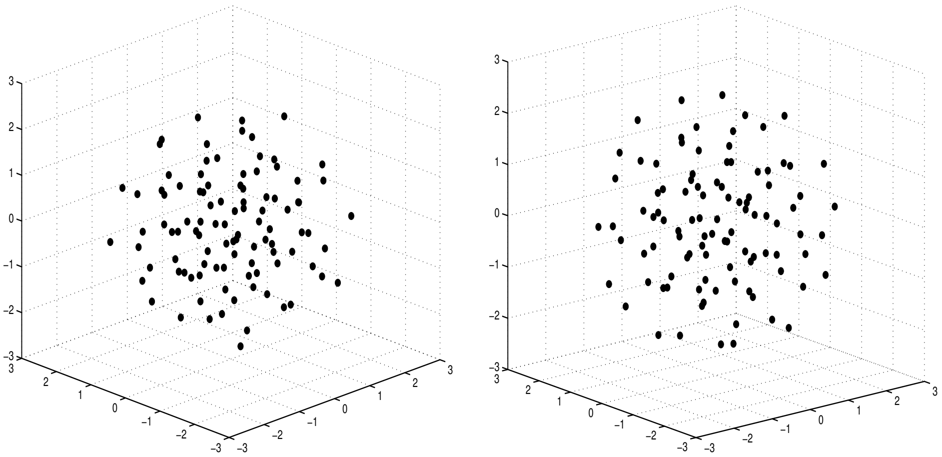


Superposition of Grid (1) and Grid (1bis) Superposition of Grid (2) and Grid (2bis)

Figure 2: Superposition of the grids

Dimension 3

Here also we concentrate on the same 6 grids as in dimension 2. For clarity reasons, we just give 2 of these 6 grids:



Grid (1 bis), $D^{n-Tr}_{final} = 0.24488$ Grid (3 bis), $D^{n-Tr}_{final} = 0.2303$

Figure 3: Optimal Grids with $N = 100$

The distorsions of the different obtained optimal grids are summarized in the table below :

Grid (1)	$D^{Tr}_{final} = 0.20439$
Grid(1 bis)	$D^{n-Tr}_{final} = 0.24488$
Grid (2)	$D^{Tr}_{final} = 0.19435$
Grid (2 bis)	$D^{n-Tr}_{final} = 0.22987$
Grid (3)	$D^{n-Tr}_{final} = 0.2311$
Grid (3 bis)	$D^{n-Tr}_{final} = 0.23030$

Figure 4: The distorsions of the optimal grids in dimension 3

Distorsion as a function of dimension

In this paragraph, we are interested in the distorsions of 4 of the 6 optimal grids introduced in Table 1 when the dimension increases. We concentrate on the computation of 14-points quantizer. The obtained results are summarized in the following table.

Dimension	Grid (2) (i.e. EA (Tr) Step)	Grid (2 bis) (i.e. Procedure 2)	Grid (1) (i.e. SGA(Tr) Step)	Grid (1 bis) (i.e. Procedure 1)
1	0.010226	0.012163	0.01028	0.012208
2	0.220686	0.2457	0.226	0.2483
3	0.66015	0.7518	0.67152	0.766
4	1.23349	1.45398	1.25944	1.4814
5	1.84278	2.26552	1.88	2.3148
6	2.4229	3.11256	2.5322	3.235
7	2.9653	4.0244	3.2036	4.2659

Table 4: Distorsions as a function of dimension

Table 4 shows that the spread between the distorsions obtained by the two types of algorithms increases with dimension. This spread becomes significant when the dimension exceeds 3. The distorsion obtained by the EA is always smaller than the one obtained by the SGA. The figure below gives the relative spread between the distorsions of Grid (1 bis) and Grid (2 bis) defined by :

(Distorsion of Grid (1 bis) - Disortion Grid (2 bis))/Distorsion of Grid (2 bis)

as a function of dimension:

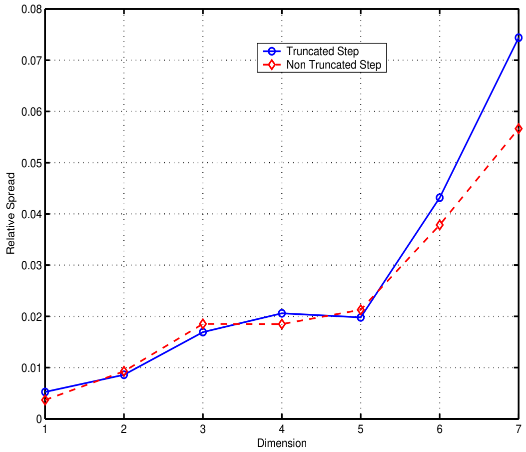


Figure 5: Relative spread as function of dimension.

So, when dealing with some high-dimensional numerical probablistic 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 [3].

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.

6.1 The Optimal stopping problem

Let $T > 0$ be a finite horizon, and $(\Omega, \mathcal{F}, \mathbb{P})$ a complete probability endowed with a filtration $\mathbb{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$ satisfying the usual conditions. The filtration $\mathcal{F}_t$ represents the information available up to time t. We also assume that $\mathcal{F}_0$ is trivial and that $\mathcal{F}_T = \mathcal{F}$. Let Y be an $\mathbb{F}-$adapted process valued in $\mathbb{R}_+$. For each $t \in [0, T]$ and $\omega \in \Omega$, $Y_t(\omega)$ represents the reward for stopping at time t in the state ω. Our main concern in this section is the following optimal stopping problem

$$U_0 \;:=\; \sup_{\tau\in\mathcal{T}_0} \mathbb{E}[Y_\tau]\ , \tag{6.1}$$

where $\mathcal{T}_0$ is the collection of all $\mathbb{F}-$stopping times valued in $[0,T]$.

6.2 Discrete-time approximation

Let $0=t_0<t_1<\ldots t_n=T$ be a partition of the interval $[0,T]$ with mesh $\max_{1\leq i\leq n}|t_i-t_{i-1}|$ $\longrightarrow 0$ as $n\to\infty$. One may choose for instance a constant mesh $t_i=iT/n$. A natural approximation of the optimal stopping problem (6.1) is obtained by restricting the maximization to those stopping times with values in $\{t_0,\ldots,t_n\}$. We shall call $\mathcal{T}_0^n$ the collection of all such stopping times, and

$$U_0^n \;:=\; \sup_{\tau\in\mathcal{T}_0^n} \mathbb{E}[Y_\tau]\ . \tag{6.2}$$

The convergence of U_0^n towards U_0 holds under mild conditions. Also, it is well-known that the value of the discrete-time stopping problem (6.2) can be computed by the following backward induction :

$$U_n^n \;:=\; Y_{t_n} \quad\text{and}\quad U_i^n \;=\; \max\left\{Y_{t_i},\mathbb{E}[U_{i+1}^n|\mathcal{F}_{t_i}]\right\}$$

for $i=0,\ldots,n-1$. Assume in addition that the process Y is Markov. Then we can rewrite the above backward algorithm in :

$$U_n^n \;:=\; Y_{t_n} \quad\text{and}\quad U_i^n \;=\; \max\left\{Y_{t_i},\mathbb{E}[U_{i+1}^n|Y_{t_i}]\right\}.$$

6.3 Approximation of the discrete-time optimal stopping problem by quantization

We next approximate, at each time t_i, the random variable Y_{t_i} by some random variable $\hat{Y}_{t_i}$ which can only take a finite number N_i of values $\Gamma_i := (y_{i,j})_{1\leq j\leq N_i}$. $\hat{Y}_{t_i}$ is the given by :

$$\hat{Y}_{t_i} \;:=\; \sum_{j=1}^{N_i} y_{i,j}\mathbf{1}_{C_j(y_i)}(Y_{t_i})\ .$$

At each time step t_i, the grid Γ_i is generated by optimal L^p-quantization.

We next turn to the problem of approximation of the value of the discrete-time optimal stopping problem. Set

$$\pi_k^{i,j} \;:=\; \mathbb{P}\left[\hat{Y}_{t_{k+1}}=y_{k+1,j}|\hat{Y}_{t_k}=y_{k,i}\right]\ ,$$

and define the approximation

$$\hat{U}_n^n \ := \ \hat{Y}_{t_n} \quad \text{and} \quad \hat{U}_k^n \ = \ \max\left\{\hat{Y}_{t_k}, \pi_k \hat{U}_{k+1}^n\right\},$$

where $U_k^n \in R^{N_k}$, and π_k is a matrix with entries $\pi_k^{i,j}$, and the maximum is taken component-wise.

The transition probabilities $\pi_k^{i,j}$ are approximated by Monte Carlo techniques. Let $\left(Y^{(m)}\right)_{1\leq m\leq M}$ be M copies of the process Y, we use the following classical Monte Carlo estimator[6] :

$$\tilde{\pi}_k^{i,j} \ = \ \frac{\sum_{m=1}^M \mathbf{1}_{Y_{t_k}^{(m)} \in C_i(y_k)} \mathbf{1}_{Y_{t_{k+1}}^{(m)} \in C_j(y_{k+1})}}{\sum_{m=1}^M \mathbf{1}_{Y_{t_k}^{(m)} \in C_i(y_k)}}$$

6.4 Numerical Results

6.4.1 The model

Let S be a process defined by the multi-dimensional Black-Scholes model :

$$\frac{dS_t^i}{S_t^i} = r dt + \sum_{j=1}^{d} \sigma_{ij} dW_t^j, i = 1, ..., d$$

Where $W = (W^1, \ldots, W^d)$ is a d-dimensional standard Brownian motion and r is a strictly positive constant and the matrix $[\sigma_{ij}]_{(i,j)\in <1,d>^2}$ is invertible. The reward process Y is defined by

$$Y_t \ := \ e^{-rt} f(S_t) \ ,$$

where f is a non-negative mapping from $\mathbb{R}_+^d$ into $\mathbb{R}_+$.

The most important feature of the Black-Scholes model is that for any time $t \in [0, T]$, the asset price S_t is a function of t and the Brownian motion W_t :

$$S_t^i \ = \ S_0^i \exp\left(\left(r - \sum_{j=1}^{d} \frac{\sigma_{ij}^2}{2}\right) t + \sum_{j=1}^{d} \sigma_{ij} W_t^j\right) \ .$$

Therefore, instead of quantizing S_t, one can quantize W_t. This remark, will be very helpful in practice.

[6]As proven in [13] we can also use a Malliavin-based estimator with less variance than the classical one.

6.4.2 The numerical scheme

Applying the quantization algorithm to the optimal stoping problem with reward process Y, we are reduced to the following backward scheme :

$$\begin{cases} \widehat{u}_n^i = e^{-rt_n} f(s_i^n), & i = 1, ..., N_n \\ \widehat{u}_k^i = \max(e^{-rt_k} h\left(s_i^k\right), e^{-r(t_{k+1}-t_k)} \sum\limits_{j=1}^{N_{k+1}} \widehat{\pi}_k^{i,j} \widehat{u}_{k+1}^j), & k = 0, ..., n-1, i = 1, ..., N_k \end{cases}$$

Where $s_i^k \in \mathbb{R}^d$ is given by:

$$\left(s_i^k\right)_l = \left(s_1^0\right)_l \exp\left(\left(r - \sum_{j=1}^{d} \frac{\sigma_{lj}^2}{2}\right) t_k + \sum_{j=1}^{d} \sigma_{lj} \left(x_i^k\right)_l\right)$$

and $x_i^k \in \mathbb{R}^d$ the i-th scenario $(i = 1, ..., N_k)$ of the multi-dimensional Brownian motion $W_{t_k}, k = 0, ..., n$.

6.4.3 Numerical Results

Wee consider a bermudan option on the geometric mean of d assets :

$$f(s) \quad := \quad \left[K - \left(\prod_{i=1}^{d} s^i\right)^{\frac{1}{d}}\right]^+ .$$

In the multidimensional Black and Scholes framework, the geometric mean of d non-dividend lognormal processes is equivalent to a particular lognormal process with a dividend yield. We give below the features of this process:

Initial value X_0	$\left(\prod\limits_{i=1}^{d} S_0^i\right)^{\frac{1}{d}}$
Equivalent volatility σ_{eq}	$\frac{1}{d^2} \sum\limits_{j=1}^{d} \left(\sum\limits_{j=1}^{d} \sigma_{ij}\right)^2$
Dividend yield q	$\frac{1}{2}\left(\frac{1}{d} \sum\limits_{i,j=1}^{d} \sigma_{ij}^2 - \sigma_{eq}^2\right)$

This allows us to use 1-dimensional finite difference scheme for the pricing of the bermudan option.

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

We consider different 3 sets of grids: The first set was obtained by with a Stochatic Gradient Algorithm, the second one is obtained is obtained with an Evolutionnary Algorithm and the third one is composed of random grids having the same law as the underlying process. 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. Each one of the used grids contains 100 Voronoi cells and the time step in the finite difference method is equal to 1/2000:

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 %

We see that, the prices obtained with a set of global 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.

References

[1] T. Bäck (1995). *Evolutionary Algorithms in theory and practice*, Oxford University Press.

[2] J. E. Baker (1985). Adaptive selection methods for genetic algorithms. *International Conference on Genetic Algorithms*, 101-111. Grefenstette editor.

[3] V. Bally and G. Pagès (2002). A quantization algorithm for solving discrete time multi-dimensional optimal stopping problems, *Bernoulli*, to appear.

[4] V. Bally, G. Pagès and J. Printems (2002). A quantization method for pricing and hedging multi-dimensional American style options, pré-pub. LPMA n^{o}753.

[5] 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.

[6] A. Gersho, J. Vaisey and K. Zeger (1992). Globally Optimal Vector Quantizer Disign by Stochastic Relaxation, *IEEE Transactions on signal processing* **40**, n^o 2, 310-322.

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

[8] J.H. Holland (1975). *Adaptation in natural and artificial systems*, Ann Arbor: University of Michigan Press.

[9] F.A. Longstaff and R.S. Schwartz (2001). Valuing American Options By Simulation : A simple Least-Square Approach, *Review of Financial Studies* **14**, 113-147.

[10] Z. Michalewicz (1996). *Genetic Algorithms+Data Structures=Evolution Programs*, (3rd Ed.) Springer Verlag, New-York.

[11] M. Mrad (2002). Robustness of Intrest Rate diffusion models, *Journal of Bond Trading and Management*, to appear.

[12] M. Mrad (2003). Pricing and Hedging of European multiasset derivatives with bounded stochastic volatility using quantization. Working paper.

[13] M. Mrad, N. Touzi and A. Zeghal (2003). Monte Carlo estimation of a joint density using Malliavin Calculus. Working paper.

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

[15] G. Pagès and J. Printems (2003). Optimal quadratic quantization for numerics: the Gaussian case, pré-pub. LPMA n^{o}809.

[16] M. Schoenauer and Z. Michalewicz (1997). *Evolutionary Computation: An Introduction* , Control and Cybernetics, 287-299, LNCS 1363, Springer Verlag.

[17] H.P. Schwefel (1981). *Numerical Optimization of Computer Models*, John Wiley & Sons.

[18] P.L. Zador (1982). Asymptotics quantization error of continuous signals and quantization dimension, *IEEE Trans. Inform. Theory* **28**, 139-148.