Selection of Alternatives of Investment in Information for Oilfield Development Using Evolutionary Real Options Approach

Version Date: April 28, 2001

By: **Marco Antonio Guimarães Dias (*)**

Abstract:

Consider an undeveloped oilfield with uncertainty about the size and quality of its reserves. There are some alternatives to invest in information to reduce the risk and to reveal some characteristics of the reserve. This paper presents an *evolutionary real options* model of optimization under uncertainty with genetic algorithms and Monte Carlo simulation, to select the best alternative of investment in information. There is a legal time to expiration of the option to start the investment for the oilfield development. The model considers both the technical uncertainties revealed by the information and the market uncertainty using two different stochastic processes for the oil prices, which are simulated. Monte Carlo simulations evaluate the decision rule curves generated in the evolutionary process. The process evolves toward a near optimum solution, giving the real option value and the optimal decision rule. The evolutionary programming under uncertainty was performed in C++ environment with good results and a description of the programming procedure is provided.

Keywords: real options, evolutionary real options, genetic algorithms, Monte Carlo simulation, investment in information, evolutionary programming, optimization under uncertainty, valuation of projects.

(*) Marco A.G. Dias is a doctoral researcher at PUC-Rio and also Technical Consultant by Petrobras. Comments are welcome and can be send by email: marcodias@petrobras.com.br

1 - Introduction

The investments to development of petroleum reserves are both large and face considerable uncertainty. There are at least two important sources of uncertainties, *market uncertainty* represented mainly by the oil prices, and *technical* (or private) *uncertainty* about the size and the quality of the reserve[1]. Information about oil in the ground is relatively expensive and reveals partial information on the size and the quality of the reserve. The selection of the best alternative for the investment in information (including not investing in information anymore) is one very important but complex practical challenge, even more in a dynamic framework considering the expiring time of the option to start the oilfield development, and the market uncertainty.

Real options approach has been used lately to consider both the uncertainties and the managerial flexibility (options) available into a dynamic framework with the aim to maximize the value of the investment opportunity. See for example Dixit & Pindyck (1994), Trigeorgis (1996), and Dias (1999). In an earlier paper (Dias, 2000), was pointed out that the practical problems of the *curse of dimensionality*[2] and the *curse of modeling*[3] have driven part of the recent real options research to the Monte Carlo simulation approach[4], due to its modeling flexibility. The major problem is the difficulty to perform optimization (backward) with simulation (forward), which in general is necessary for American-type[5] options.

This paper follows the precedent work (Dias, 2000) in the sense of investigating a more direct[6] alternative to optimize the real option value using evolutionary approach combined with Monte Carlo simulations of stochastic processes and technical uncertainties. Specifically as optimizer tool is used *genetic algorithms* (GA) with data structures, so the evolutionary programming in the sense of Michalewicz (1996). The optimization problem is performed through the evolution of solutions toward the optimum, using genetic operators like special crossover and mutation operators. This follows the Darwin's principles of evolution,

---

[1] Quality of the reserve is in a broad sense, not only properties of permeability/porosity of the reservoir-rock, the quality and properties of the oil and/or gas, and reservoir inflow mechanism. In cases as of deepwater offshore oilfields, uncertainty about the quality of the reserves includes the uncertainty about the technology cost and performance. The quality of the reserve later will be more precisely defined.

[2] Curse of dimensionality is the exponential computational time explosion with the problem dimension.

[3] Curse of modeling is a problem formulation with an explicit system. Changing some aspects of the model is necessary to change all the optimization procedure.

[4] See for example Cortazar & Schwartz (1998) and Ibáñez & Zapatero (1999).

[5] American options can be exercised before the expiration, so there is the problem to find out the optimal earlier exercise of the American option. The optimal exercise curve or the threshold curve is the decision rule along the time that maximizes the value of the option.

so there are analogies: the chromosomes with some genes (characteristics of the chromosomes) evolve by reproduction (crossover operators) and mutation. There is a population of chromosomes that evolve along generations. The best-adapted (high fitness) chromosomes have more chances to pass their genetic material (genes) for the following generations. Here fitness means the real options value of the alternative. The chromosome here is the set of characteristics of the decision rule of investment that maximizes the real option value of each alternative.

The literature of genetic algorithms and evolutionary programming is today well developed. See for example the classic texts of Holland (1975), Goldberg (1989), Davis (Eds.) (1991), and Koza (1992), and more recent contributions of Michalewicz (1996), and Michalewicz & Fogel (2000). For an overview see Pacheco (2000) and for applications in the petroleum industry see Mohaghegh (2000). For complex applications of investment under uncertainty, an *evolutionary real options* approach is birthing, and the aim of this paper is to contribute in this process.

The technical uncertainty is modeled with distributions of probabilities that are revealed by the investment in information. There are different alternatives of investment in information with different costs. The market uncertainty is divided into two cases and modeled with two stochastic processes. The first one is the more popular geometric Brownian motion. The second one is the more realistic mean-reversion model for the oil prices. The results are present for the two cases. Computational details are also discussed.

This paper is organized as follows. In the second section is described the petroleum field case and the alternatives to invest in information. In the third section is described the evolutionary approach modeling issues like the genetic operators and the chromosome, and is described the programming issues. The results are presented in the fourth section. The fifth section presents the concluding remarks.

2 – Description of the Petroleum Field Development Problem, the Uncertainties, and the Alternatives

2.1 – The Petroleum Field Development Problem and the Main Variables

Suppose there is an oilfield that was discovered, (partially) delineated, remaining undeveloped with T years before the expiration of the option in the block concession[7]. The reserve size (B, in million of

[6] The traditional optimization under uncertainty is the Bellman's backward procedure (dynamic programming).
[7] This means that if the option time expires without the oil company (owner of the track rights) presenting a plan to immediate start of investment, the National Agency get back the block and may to bid it again.

equivalent[8] barrels) is uncertain. The investment to develop[9] (D, in million US$) the oilfield is function of the reserve size.

The goal of the paper is to select the best alternative to invest in information, prior the development of the oilfield, and to compare with the alternative of not investing in information. Before describing the alternatives to invest in information, is necessary to set the base case for the oilfield development and the main equations to evaluate the decision to develop the oilfield.

In the base case, D is estimated in US$ 480 million (all monetary values are present values) and the expected reserve size is 120 million barrels. The investment is optional (is not obligatory) and this option expires in two years. The traditional value of the petroleum field is given by the static *net present value* (NPV). The NPV expression can be written:

$$\mathbf{NPV = q\ P\ B - D} \qquad \mathbf{(1)}$$

Where:

q = economic quality of the reserve[10], which is uncertain with expected value of 20%;

P = petroleum price, suppose the current value is US$ 20/bbl and the future value is uncertain;

B = reserve size, which is uncertain but expected to be 120 million barrels; and

D = development cost, function of B, is assumed to be US$ 480 million for B = 120 million barrels.

Note that for the base case value pointed above, the NPV is zero (= 0.2 x 20 x 120 – 480).

The investment cost D is at least a function of the reserve size (B). Hence, for larger reserves the investment must adjust the oilfield scale, and so on. The equation for the development investment is given by the simple equation (all values in millions), with the values adopted for the base case:

$$\mathbf{D = fixed\ investment + (variable\ cost \times B) = 240 + (2 \times B) \quad [in\ \$million]} \qquad \mathbf{(2)}$$

[8] All the oil reserves have associated gas. Equivalent reserves means that the gas reserve is added into the value of B, using the economic equivalence between x m3 of gas with one barrel of oil, where x depends mainly of the local gas market demand.
[9] In order to develop the oilfield is necessary to invest in development wells, production platform and its processes facilities, pipelines, etc.
[10] The reserve is more valuable as higher is q. The value of q depends of several factors: the permo-porosity properties of the reservoir-rock; the quality and properties of the oil and/or gas; reservoir inflow mechanism; operational cost; country taxes; discount rate; etc. For details about q see: http://www.puc-rio.br/marco.ind/quality.html

2.2 – The Uncertainty in the Oil Prices Modeled with Two Stochastic Processes

The model of the market uncertainty represented by the oscillations in the oil prices is performed with stochastic processes and is divided into two cases. The first one is the traditional geometric Brownian motion, which is used in most financial and real options models. The theoretical solution, by using a difference finites approach to solve a partial differential equation (PDE), is available so that is possible to compare this solution with the evolutionary approach solution. The second stochastic process is a more realistic one for the oil prices (and most cases of price of commodities), the mean-reversion stochastic process. There is no alternative theoretical solution for the specific mean-reversion process specified in this paper, although is possible the alternative PDE approach.

The introduction of a non-standard mean-reverting process illustrated an advantage of the evolutionary approach over the PDE method. While is easy to change the stochastic process for the oil prices in this evolutionary model, the PDE approach needs the rebuilding of the PDE and new software to solve the PDE using finite difference methods. For the evolutionary approach, is necessary only to change the function that simulates the oil prices, because the rest of the program remains the same.

2.2.1 – Geometric Brownian Motion

First, assume that the oil prices follow the popular Geometric Brownian Motion (GBM), in the format of a *risk-neutralized* stochastic process[11], that is, using a risk-neutral drift $(r - \delta)$ instead the real drift α, is:

$$\mathbf{dP = (r - \delta)\, P\, dt + \sigma\, P\, dz} \qquad \mathbf{(3)}$$

Where:

r = interest rate, assumed to be 8% p.a.;

δ = convenience yield of the oil, assumed to be 8% p.a., too;

σ = volatility of the oil prices, assumed to be 25% p.a.; and

dz = Wiener increment = $\boldsymbol{\varepsilon} \sqrt{\mathbf{dt}}$, where $\varepsilon \sim N(0, 1)$

The equation necessary to perform the Monte Carlo simulation of petroleum price sample paths is:

$$\mathbf{P_t = P_{t-1} \exp\{ (r - \delta - 0,5\, \sigma^2)\, \Delta t + \sigma\, N(0, 1) \sqrt{\Delta t} \}} \qquad \mathbf{(4)}$$

[11] For a discussion of this equation and its discretization in order to perform the Monte Carlo simulation, see for example Clewlow and Stickland (1998), and for general discussion of discretization of stochastic differential equations see the more advanced text of Kloeden & Platen (1992).

2.2.2 – The Mean-Reversion Process

In opposite to the geometric Brownian motion case, there is no consensus about the best format for a mean-reverting stochastic process. One of most famous mean-reversion model is due to Schwartz (1997). This paper will present another mean-reverting model, which he used before in practical project finance applications. Simulations of this model was compared with the Schwartz model, and although the Schwartz model is slight simpler, the adopted model performs better in the region near the long run equilibrium price level[12]. This case is developed because is a not a standard of the literature.

Initially consider the following *Arithmetic Ornstein-Uhlenbeck* process for a stochastic variable x(t):

$$\mathbf{dx = \eta\ (\bar{x} - x)\ dt + \sigma\ dz} \qquad \textbf{(5)}$$

This means that there is a reversion force over the variable x toward an equilibrium level $\bar{\mathbf{x}}$. This is like a spring force. The velocity of the reversion process is given by the parameter η. The solution of this equation can be view below including the stochastic integral (see for example Kloeden & Platen, 1992):

$$\mathbf{x(T) = x(0)\ e^{-\eta T} + \bar{x}\ (1 - e^{-\eta T})\ + \sigma\ e^{-\eta T} \int_0^T e^{-\varsigma t} dz(t)} \qquad \textbf{(6)}$$

The variable x(t) has Normal distribution with the following expressions for the mean and variance[13]:

$$\mathbf{E[x(T)] = x(0)\ e^{-\eta T} + \bar{x}\ (1 - e^{-\eta T})} \qquad \textbf{(7)}$$

$$\mathbf{Var[x(T)] = (1 - e^{-\eta T})\ \sigma^2/(2\,\eta)} \qquad \textbf{(8)}$$

The equation 7 can be view as a weighted average between the current level x(0) and the equilibrium level $\bar{\mathbf{x}}$. The weights sum one and depend of the time and the reversion speed. Equation 8 shows that in the long run, the variance reaches an asymptotic value of $\sigma^2/(2\,\eta)$.

Now the stochastic process for the oil price P(t) is chosen so that the oil prices are function of x(t) described by the equation 5. First let us to set $\bar{\mathbf{x}} = \ln(\bar{\mathbf{P}})$, where $\bar{\mathbf{P}}$ is the price of equilibrium in the long

[12] The mean-reversion force near of the equilibrium level is too weak compared with the adopted model. In addition, in this paper the adopted starting oil price (current price) was equal to the equilibrium level of US$ 20/bbl, so the adopted model permits to study better the reversion effect, contrasting with the geometric Brownian case.

[13] See for example Dixit & Pindyck (1994, chapter 3).

run. The idea is to set the prices as Log-Normal, with mean $E[P(T)] = e^{E[x(T)]}$. The process $P(t) = e^{x(t)}$ doesn't work because the exponential of a Normal distribution adds the half of the variance in the Log-Normal distribution. In order to reach the goal, the half of the variance is compensated using the equation:

$$\mathbf{P(t) = exp\{x(t) - 0.5\ Var[x(t)]\}} \qquad \mathbf{(9)}$$

Where Var[x(t)] is given by the equation 8. In the risk-neutral format[14], the process x(t) (equation 5) is simulated using the expression[15]:

$$\mathbf{x_t = x_{t-1}\, e^{-\eta\,\Delta t} + [\ln(\bar{P}) + ((r - \rho)/\eta)]\,(1 - e^{-\,\eta\,\Delta t})) + \sigma\,\sqrt{(1 - \exp(-2\,\eta\,\Delta t))/(2\,\eta)}\ N(0, 1)} \qquad \mathbf{(10)}$$

Where ρ is the risk-adjusted discount rate for the developed reserve. The simulation of the process is now simple. Calculate x(t) using the equation 10 and sampling the standard Normal distribution. Use the equation 9 combined with the equation 8 to get the price process P(t).

The adopted values for the mean-reversion variables are:

- Long run equilibrium price $\bar{\mathbf{P}}$ = US$ 20/bbl;
- Mean-reversion speed[16] η = 0.3466 year^{-1}; and
- Risk-adjusted discount rate ρ = 12% p.a.

2.3 – The Optimal Decision Rule: the Investment Threshold

The development threshold gives the *decision rule for the development*. With the simulated oil price P(t) is possible to estimate the value of a developed reserve V(t) = q B P(t). In order to ease the model, is better to work with normalized value of the reserve V/D. For V/D = 1, the NPV is zero. The use of normalized value of the developed reserve V/D, instead for example the oil price, for decision rule curve permits to combine the technical and market, that is values of P, q, B, D(B), in the same threshold curve. The real

[14] In the risk-neutral format, the drift α of the process is replaced by $r - \delta$, where δ is the dividend yield. In the mean-reversion case, the dividend yield is not constant, it is function of P: $\delta = \rho - \alpha = \rho - \eta\,(\bar{x} - x)$. The risk neutral format permits to use the risk free interest rate as the correct discount rate.

[15] The correct discrete format for the continuous-time process of equation 5 is a first-order autoregressive process, AR(1), see Dixit & Pindyck, 1994, p.76. The process is risk neutralized by changing the drift.

[16] There is the following relation between the mean-reversion speed η and the half-life of the process H: $H = \ln(2)/\eta$. Half-life is the time that is expected for the variable stochastic x to reach the half of way toward the equilibrium level. The value of η corresponds to H = 2 years, which is a reasonable value for the logarithm of the oil price process.

option value F is homogeneous of degree one[17] in V and D: F(V/D, D/D) = (1/D) F(V, D); and it permits to use V/D in the maximization problem. The threshold (V/D)* is homogeneous of degree zero in V and D, see the classical paper of McDonald & Siegel (1986, p.713). See Dixit & Pindyck (1994, pgs.207-211) for the homogeneity concept in the real option stochastic differential equation context.

The threshold is the critical (V/D)* level that makes optimal the immediate investment to develop the oilfield. It is the rule to exercise the option (exercise at or above the threshold). The threshold curve is the decision rule that maximizes the real options value, determining the optimal exercise of the real option, and it is function of the time. The threshold curve will be determined using the evolutionary approach. Before, is presented how work the threshold curve with the simulation of the uncertain V/D.

The figure below illustrates this point. Two sample paths of the normalized value of the reserve V/D are showed. One path reaches the threshold line at the point "A", so the decision rule is to exercise the option at this time (the option is "deep in the money"), earning F(t = 1) = NPV = V – D. This value is a future value (t = 1 year), so to calculate the present value just multiply by the discount factor exp(– r t). The other path pass all the option period without reach the threshold curve (see point "B" in the figure), in this case the option values zero.

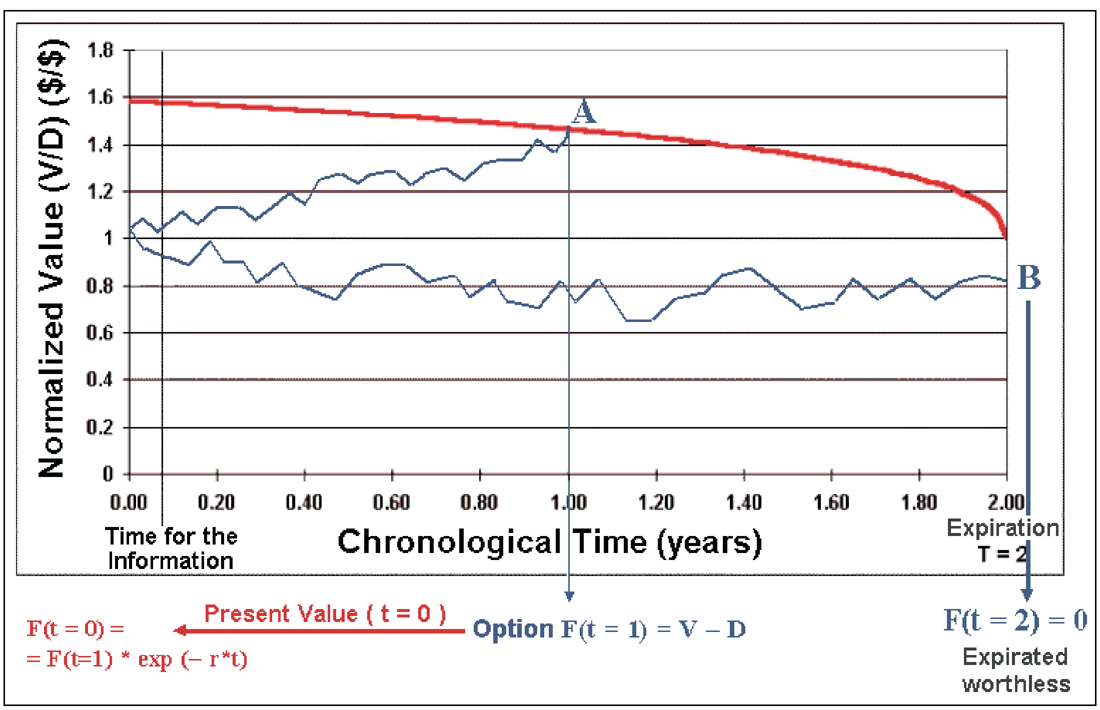

Figure 1 – The Threshold Curve and Exercise of the Real Option Simulated Value

[17] A function is homogeneous of degree n if F(t**x**) = t^n F(**x**) for all t > 0, n ∈ þ, and **x** is the vector of variables.

The picture above shows the time for the information to be available, too. This time depends on the alternative of investment in information adopted. The alternatives are presented below.

2.4 – The Alternatives to Invest in Information

Each alternative for the acquisition of information has a cost and reveals different levels of information. After the investment in information, there are several possible scenarios for B and q (but the expected values remains the same). These scenarios are assumed to follow *triangular* probability distributions that are defined by the parameters (minimum, probable, maximum). In addition, investing in information means to delay the possibility to exercise the development investment option, because *in most cases* there is no sense to invest in information in parallel with the development investment. The investment in information has the objective to design the development plan. The alternatives are modeled considering its cost (cost of information), the information revelation potential given by the probability distributions, and the time to delay the development option exercise possibility. The alternatives are:

Alternative 1: Long-Term Test

This alternative is simply to stay with a rig over a drilled well performing a long-term test in this well, that is, to produce the well for 25 days measuring flow rate, bottom-hole pressure, and wellhead pressure. The rig daily rate is assumed to be US$120,000 per day.

Cost = 25 days/rig x 120 M$/day = US$ 3 MM
Development Delay: 30 days
Revealed Information: B ~ Triang (B_0 x 0.8 ; B_0 ; B_0 x 1.2);
q ~ Triang (q_0 x 0.9 ; q_0 ; q_0 x 1.1)

Alternative 2: Additional Well (ADR, Slim)

This alternative comprises to drill a slim (cheaper) well known as "acquisition of data from the reservoir" (ADR) in a reservoir region with high degree of uncertainty.

Cost = US$ 4 MM
Development Delay: 30 days
Revealed Information: B ~ Triang (B_0 x 0.5 ; B_0 ; B_0 x 1.5);
q ~ Triang (q_0 x 0.9 ; q_0 ; q_0 x 1.1)

Alternative 3: Drilling Schedule without Test

The idea is to set the (development) drilling schedule in order to optimize the information rather than to optimize the drilling costs. In practice, the optimum for drilling cost is to drill all the wells in one cluster of wells, and only after this to mobilize the rig to another cluster. The optimum for the information typically will be to drill only one well in one cluster and to mobilize the rig to another cluster and again drilling only one well there. With this procedure, more parts of the reservoir are investigated by the development drilling. The costs are the additional rig time to mobilize/demobilize and the additional time on the development schedule. In case of the alternative 3, no tests are performed in the well after the drilling.

Cost = 4 mob/demob = 4 x 10 days x 120 M$/d = US$ 4.8 MM

Development Delay: 5 months

Revealed Information: B ~ Triang (B_0 x 0.4 ; B_0 ; B_0 x 1.6)

q ~ Triang (q_0 x 0.8 ; q_0 ; q_0 x 1.2)

Alternative 4: Drilling Schedule with Well-Testing

This alternative is similar to the alternative 3 except that production test is performed in each well drilled. Each test is assumed to take 10 days, so the cost is two times the alternative 3, but the information revealed is richer.

Cost = 4 mob/demob + tests = 4 x (10 + 10) days x 120 M$/d = US$ 9.6 MM

Development Delay: 6 months

Revealed Information: B ~ Triang (B_0 x 0.3 ; B_0 ; B_0 x 1.7)

q ~ Triang (q_0 x 0.6 ; q_0 ; q_0 x 1.4)

3 – Evolutionary Approach Modeling Issues

3.1 – The Genetic Algorithm Codification

The evolutionary approach job is to generate decision rules (the threshold lines) that maximize the value of the option. In order to do this, the threshold line needs to be coded by a set of genes with characteristics of this threshold line. This is performed by using two free points and one function. By changing the time variable, that is, using the time to expiration ($\tau = T - t$) instead the chronological time, is possible to use a known function, the logarithm function. So, the threshold curve is modeled using with two free points (to represent the threshold curve near of the expiration) and *one logarithm function*[18] (for most of the threshold curve). The picture below illustrates this point.

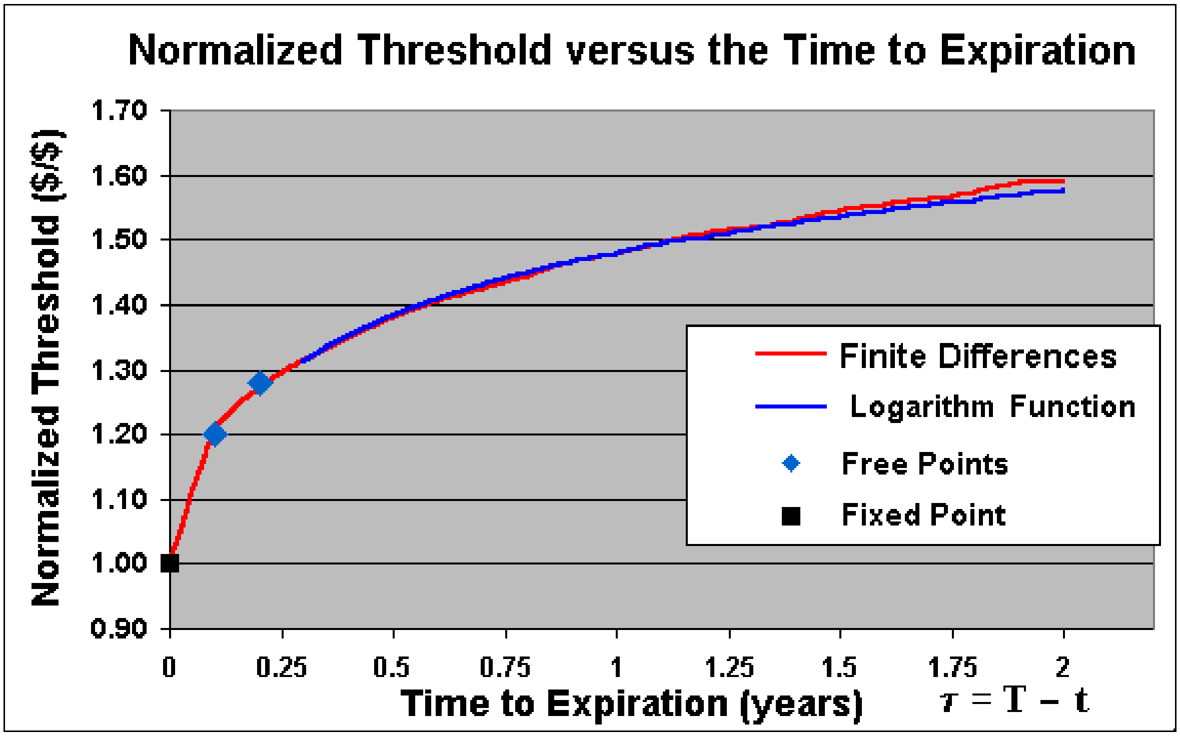

Figure 2 – The Threshold and the Points and Function Approximation

The function has the format $\mathbf{(V/D)_\tau = c + d\ ln(\tau)}$, where the "c" and "d" are coefficients of the function and τ (= T – t) is the time to expiration. The fixed point for $\tau = 0$, is the known heuristic that at the expiration (without option to delay) the NPV rule works, so invest if NPV = V – D > 0, which corresponds to invest if V/D > 1, in this normalized approach.

The experience shows that near of expiration the threshold has a more variable shape. The free points are sufficient flexible to capture these shape variations. This heuristic representation of the threshold curve with four variables is sufficiently simple and reaches the range of possible optimal solutions. In the genetic algorithm model, each chromosome has 4 genes representing the four parameters that characterize the threshold curve. The genes comprise the free values (a, b) and the coefficient values for the logarithm curve (c, d). The picture below illustrates this gene codification.

[18] For American *put*, Ju (1998) uses a *multipiece exponential function* to approximate the early exercise boundary (threshold curve). For American *call*, the logarithm approximation is the analogue idea. The symmetry put-call is nowadays well known in the literature of American options (see for example Carr & Chesney, 1996).

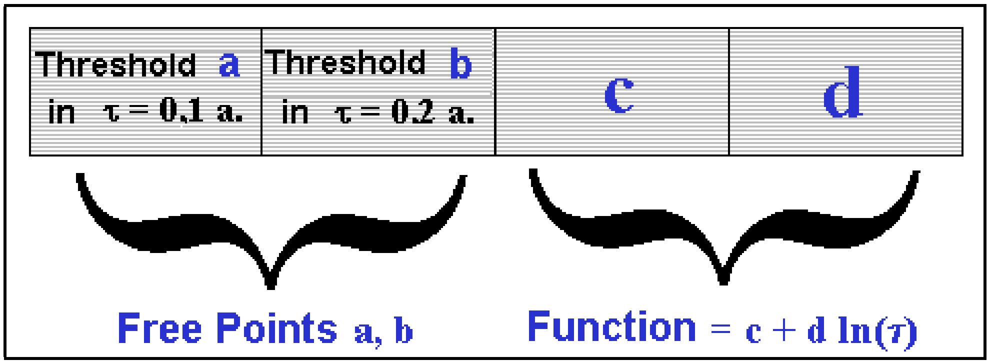

Figure 3 – The Genes of the Chromosome

The vector $(a, b, c, d)^T$ will be decoded to the threshold curve $(V/D)_t$, through the $(V/D)_t$ vector generation function. The free points "a" and "b" are chosen to be V/D respectively at the times $\tau = 0.1$ year and $\tau = 0.2$ year (near of the expiration, where the logarithm function doesn't work well). Between the fixed point ($V/D = 1$ at $\tau = 0$) and the free points, a linear interpolation is used. The logarithm curve "start" (by the expiration time point of view) at the point immediately after the point "b". For the model, which needs to work in discrete-time, the logarithm function "starts" at the instant $0.2 + \Delta t$, where Δt is the discrete time interval used in the model.

It is important also to set some *restrictions* for the genetic algorithm in order to accelerate the algorithm convergence. In order to do this, is used a heuristic that the threshold curve is monotonically decreasing (increasing) with the chronological (expiration) time. So there are the following restrictions in the GA model:

$$\mathbf{b > a} \qquad \mathbf{(11)}$$

$$\mathbf{c + d * \ln(0.2 + \Delta t) > b} \qquad \mathbf{(12)}$$

Note that both restrictions are linear restrictions (the logarithm argument is a number, not a gene). In addition to the restrictions, is necessary to specify the domain for the genes. The model will work with floating point numbers (real numbers) instead the binary representation common in genetic algorithms textbooks. These domains are:

$$\mathbf{a \in (1.0, 1.6)}$$

$$\mathbf{b \in (1.0, 1.8)}$$

$$\mathbf{c \in (1.0, 2.0)}$$

$d \in (0.0, 0.6)$

The genetic algorithm procedure is summarized in the picture below. The steps comprise:

- the initialization of the population (initial set of candidate solutions, which needs to be feasible with the restrictions and the genes domains);
- the valuation of the chromosomes (the more complex part, which will use the Monte Carlo simulation and the concepts of real options, see next topics);
- the ranking of the chromosomes for the parents selection;
- the parents selection for the reproduction;
- the reproduction with the help of the genetic operators of crossover and mutation (generating a new set of candidate solutions); and
- the process evolves through the "generations" (the above procedure is repeated for n generations), until an optimum (or more frequent, a near optimum) solution is reached. See the figure below.

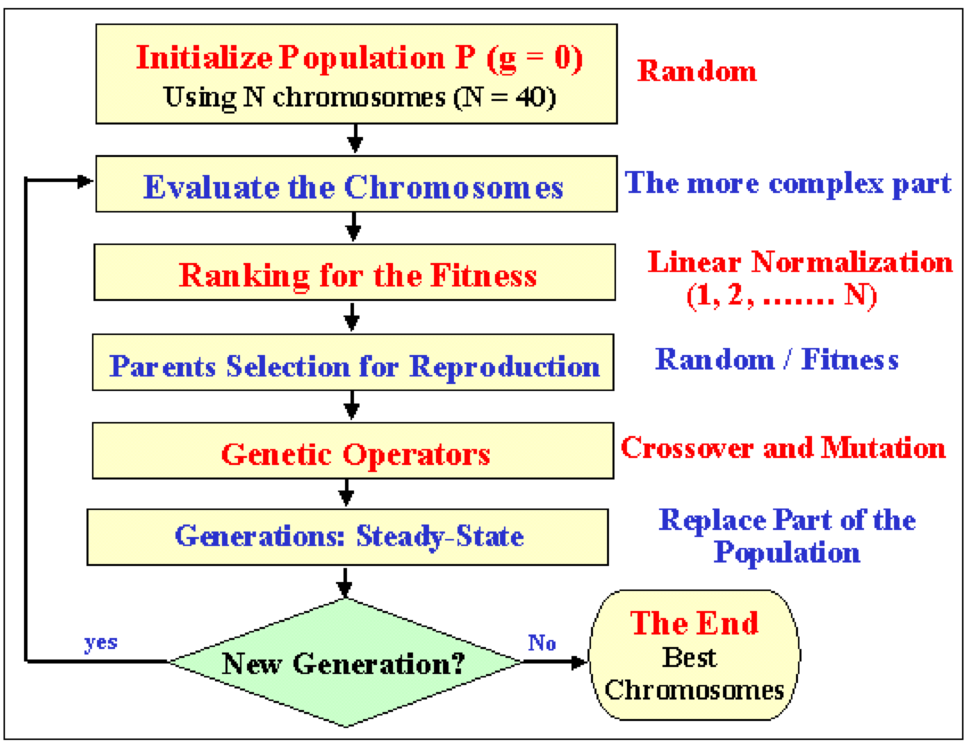

Figure 4 – Flowchart for the Genetic Algorithm

The option to work in a C++ environment calls the idea of looking for a C++ library that works with floating point genes, linear restrictions, and permits to regulate the selection pressure. See in Dias (2000) the critic of the RiskOptimizer because it uses very low selection pressure and doesn't permit the user to change this set. The C++ environment is also faster than Excel for the evaluation of the chromosomes that need the Monte Carlo simulation.

The only free C library available with these characteristics is the software ***Genocop***, developed by the Prof. Michalewicz and available to download in his website[19]. There are Genocop versions that handle linear restrictions (Genocop I) and more recent version that handles non-linear restrictions (Genocop III). In this paper was chosen the Genocop I (version 3.0) because there are only linear restrictions and this version is easier to adapt in the C++ program. The major adaptations in the Genocop code were: changes due moving from C to C++ language; integration with a complex evaluation module; and moving from the original Genocop "chromosome by chromosome" evaluation, to parallel evaluation of all population.

Genocop I works with until seven genetic operators. However the boundary mutation operator will not be used because is known that the domain boundaries are not optimal for the genes. Other operator (whole non-uniform mutation) is only a small variation and will be not used. Michalewicz (1996) describes the five operators used in this paper. The operators are:

- Uniform Mutation;
- Non Uniform Mutation;
- Arithmetical Crossover;
- Simple Crossover; and
- Heuristic Crossover.

3.2 – <u>Modeling the Optimization under Uncertainty</u>

The figure below presents the idea for the evolutionary real options approach. The genetic algorithm (GA) generate a population of decision rules, the threshold curves candidates to be the optimal decision rule. The Monte Carlo simulation generates the stochastic process for the oil prices and combine with the probability distribution revealed by the information. The threshold curve determines the level which the option is exercised. The rule is evaluated. The process is stopped only when the number of generations matches the total generations set as program input.

[19] See the Genocop website at http://www.coe.uncc.edu/~zbyszek/gchome.html

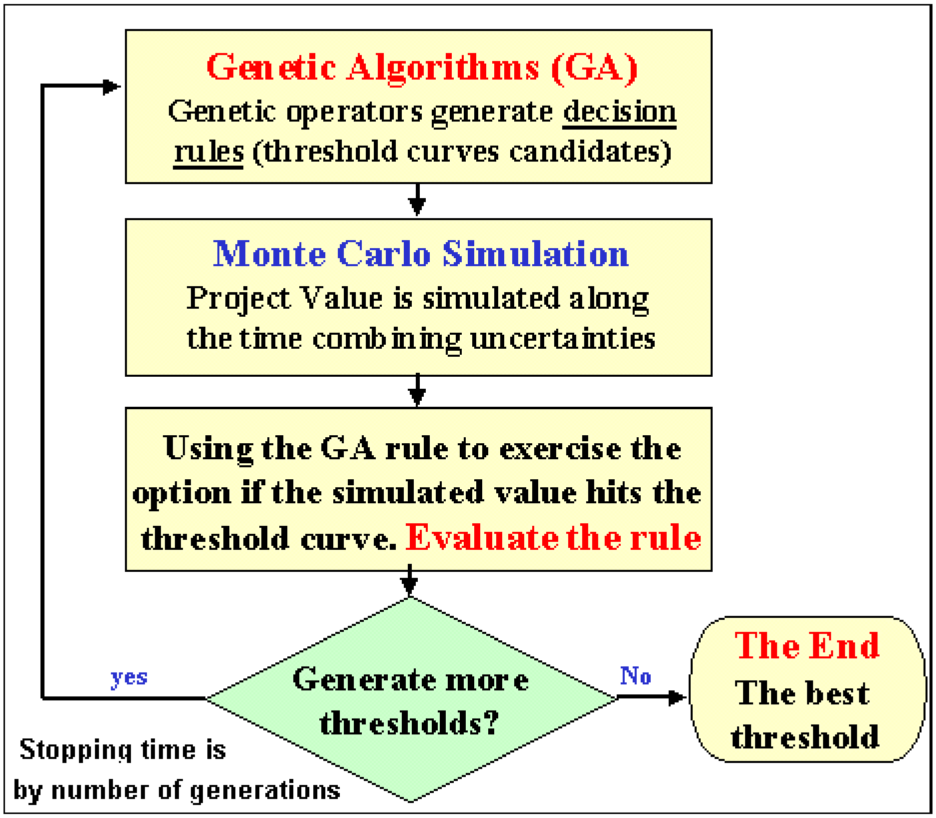

Figure 5 – Flowchart of the Evolutionary Real Options Model

The Monte Carlo simulation is divided into two parts. First the generation of technical scenarios (k) and second the generation of the petroleum prices iterations. NIT denotes the number of iterations for the technical scenarios and NI denotes the number of petroleum prices iterations. For each technical scenario are performed NI iterations for the oil prices. So, the total number of Monte Carlo simulations corresponds to NIT x NI. For the base case with NIT = 50 and NI = 1,000 the total of iterations is 50,000. By using a time step of 0.05 year, meaning 40 time intervals in each path for the two years to expiration base case. For each time step, a set of calculus is performed, so to evaluate each chromosome are performed 50,000 x 40 = 2 millions of sets of calculus and instructions! A number much higher than the used in the earlier paper (Dias, 2000) in the Excel environment.

The idea of Monte Carlo simulation to evaluate the chromosome (the threshold curve) is illustrated below for three technical scenarios revealed by the investment in information, an out-of-money option (bad technical news), an at-the-money option (neutral news), and an in-the-money option (good news was revealed), respectively.

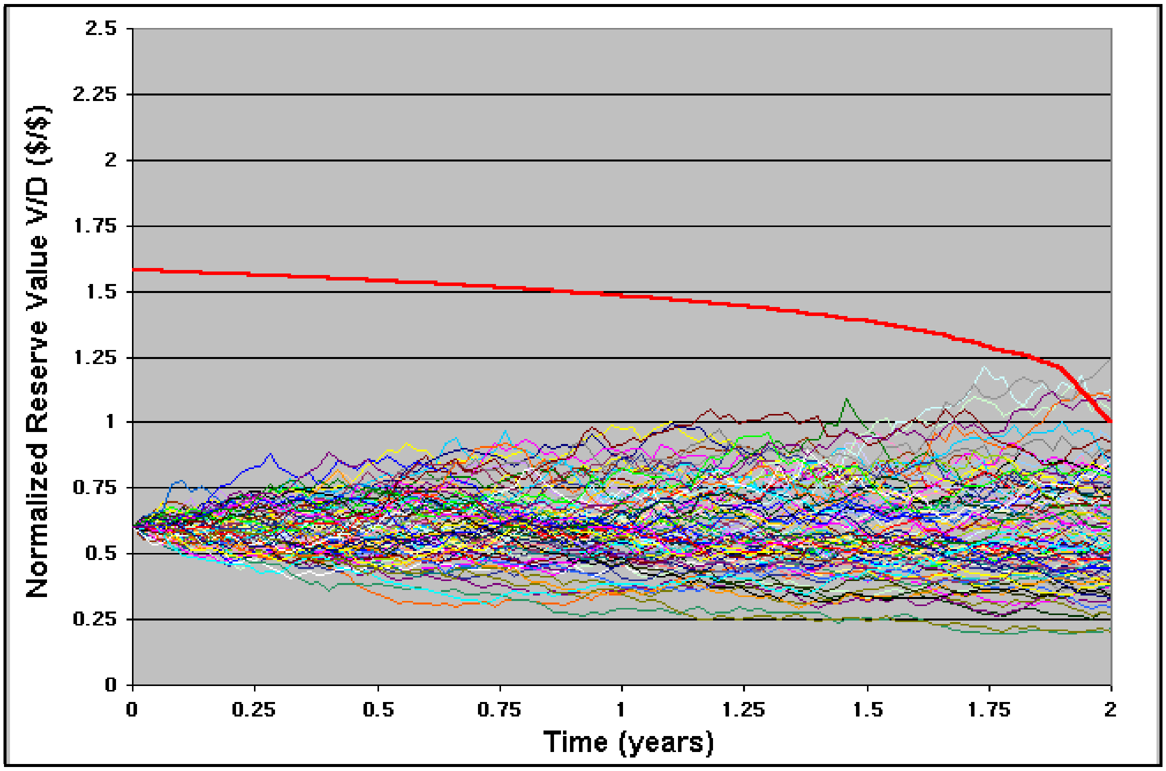

Figure 6 – The Monte Carlo Simulation Starting from a Bad Scenario

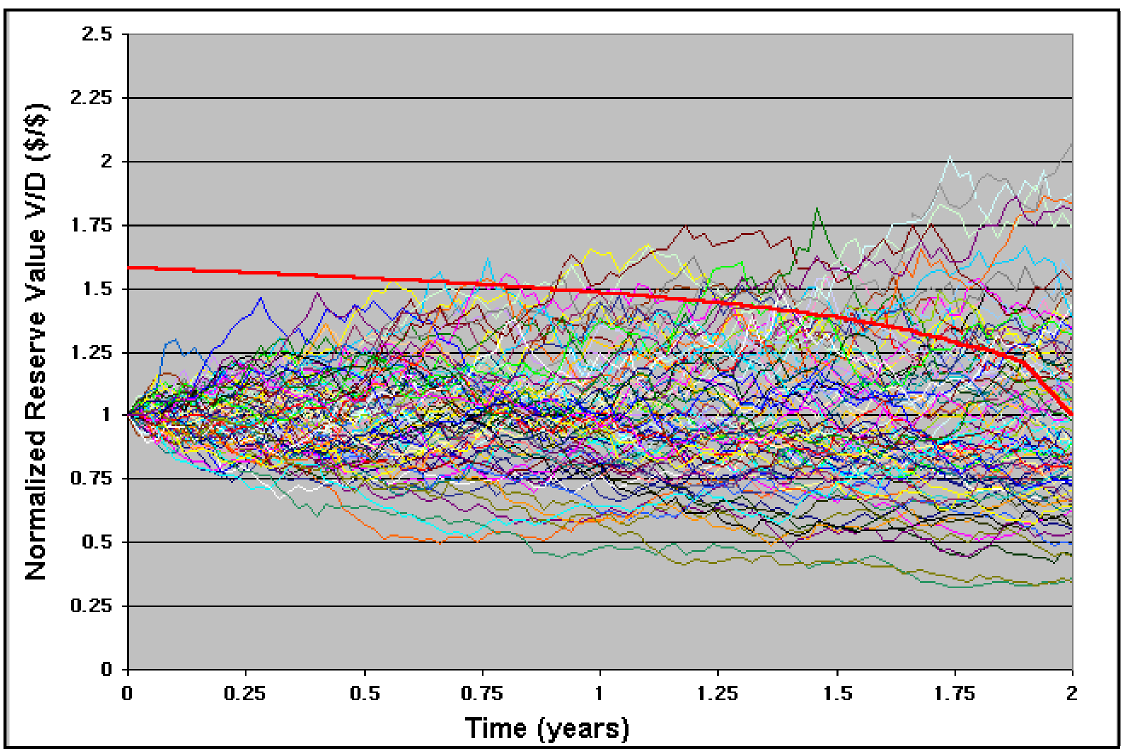

Figure 7 – Monte Carlo Simulation Starting from an Expected Scenario

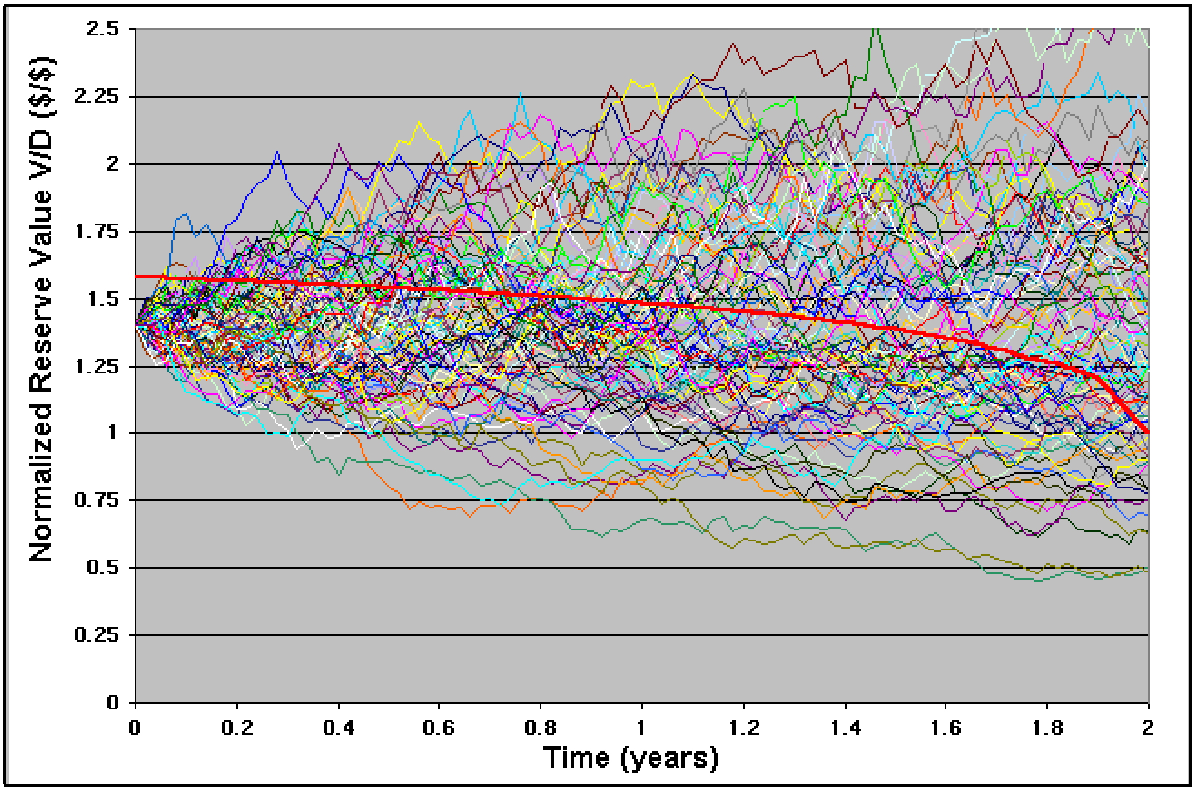

Figure 8 – Monte Carlo Simulation Starting from a Good Scenario

The most interesting practical idea in this programming was the use of text files with previously simulated technical scenarios and standard Normal distributions. The C++ program read two text files, one file with a table of technical scenarios and other file with a table of N(0, 1). See the picture below.

Scenarios File

k	$(q.B)_k$	D_k
1		
2		
3		
:		
:		
:		
:		
NIT		

Standard Normals File

i \ t	Δt	$2\Delta t$	…….	T
1	N(0, 1)	N(0, 1)		N(0, 1)
2	N(0, 1)	N(0, 1)		N(0, 1)
3	N(0, 1)	N(0, 1)		N(0, 1)
:	:	:		:
:	:	:		:
:	:	:		:
:	:	:		:
NI	N(0, 1)	N(0, 1)		N(0, 1)

Figure 9 – Offline Simulations Are Read from Two Files

The first file has a matrix with two columns (the product q*B and the correspondent D, which is function of the B used in the first column) and NIT rows (where NIT = number of technical scenarios). The other file has T/Δt columns and NI iterations (NI = number of paths for the petroleum price simulation.

The advantages of this procedure (using two text files) are:

(a) All the chromosomes of all generations are evaluated using the same random numbers set, preventing luckly valuation of bad chromosomes with dissemination of bad genetic material for the next generations. In addition, is possible to use commercial Monte Carlo softwares that have techniques of variance reduction[20];

(b) Permits to work offline searching for a near statiscally "perfect" set of technical scenarios distribution and set of N(0, 1) values. It is possible to run several simulations offline until to get a non-biased table of values (matching the desired mean and other probabilistic moments). The online simulation of scenarios and N(0, 1) doesn't permit this control;

(c) The N(0, 1) file is very flexible[21], because all practical stochastic processes for oil price will need N(0, 1). For other stochastic process, just change the function of oil price in the C++ program that is possible to use the same file of N(0, 1). For larger time to expiration, just add more columns of selected simulations of N(0, 1); and

(d) With two files the evaluation of alternatives is very flexible. For another alternative of investment in information, which will reveal a new set of scenarios, is necessary only to change the file with technical scenarios, because the file with standard Normal numbers is the same. The N(0, 1) file only change if you want more precision or if the time to expiration changes (in the last case, only add new columns with independent simulations for higher T, or reduce the number of columns for lower T).

[20] Example is the @Risk, which has embedded the efficient Latin Hypercubic sampling.

[21] In addition, note that the offline simulation of P(t) or even V, instead N(0, 1) is not practical, because every time you want to change the initial P, you need to perform other offline simulations and to generate other text file with a table. Using a text file of N(0, 1) is much more flexible than the direct generation of P(t) or V or V/D.

The picture below shows the flowchart of the program where the entire population is evaluated in parallel. So, this picture presents the evaluation of one generation with J chromosomes (population size = J). The cycle is repeated for G generations. The evaluation of each technical scenario is displayed separately for sake of space and understanding.

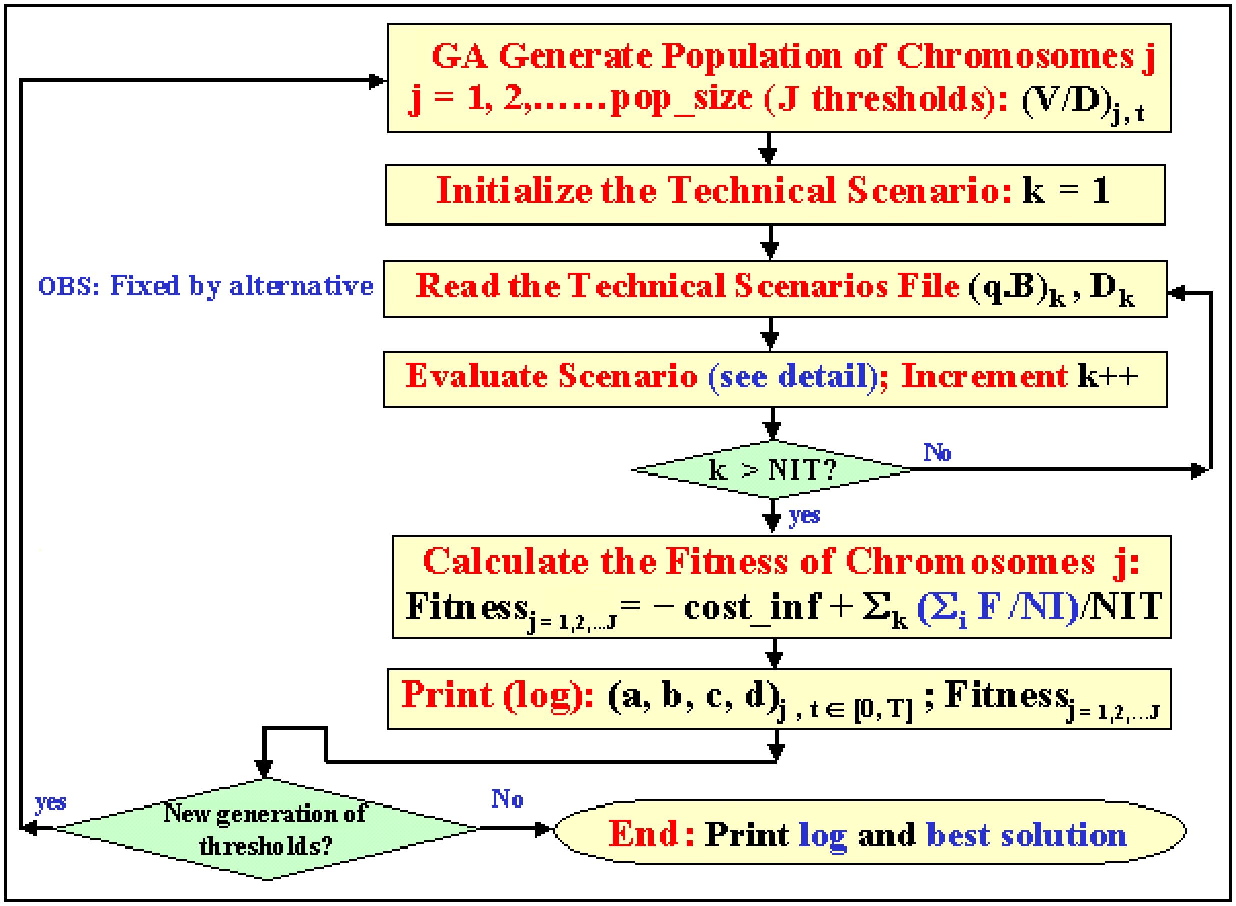

Figure 10 – Flowchart for the Evolutionary Programming

The technical scenario evaluation is performed according the following flowchart, which details the previous flowchart. For each chromosome j in each path i and in each instant t of this path is performed the comparison between the simulated $(V/D)_{i,\ t}$ with the chromosome threshold level $(V/D)_{j,\ t}$. If the simulated V/D is higher the option is exercised in that path and for that chromosome j. Before passing to another path, the program wait until all chromosomes to be evaluated in that path. The option value for each chromosome in a scenario k is the average option value for NI paths. The chromosomes evaluation vector (j = 1, 2, ….J) in scenario k is storaged.

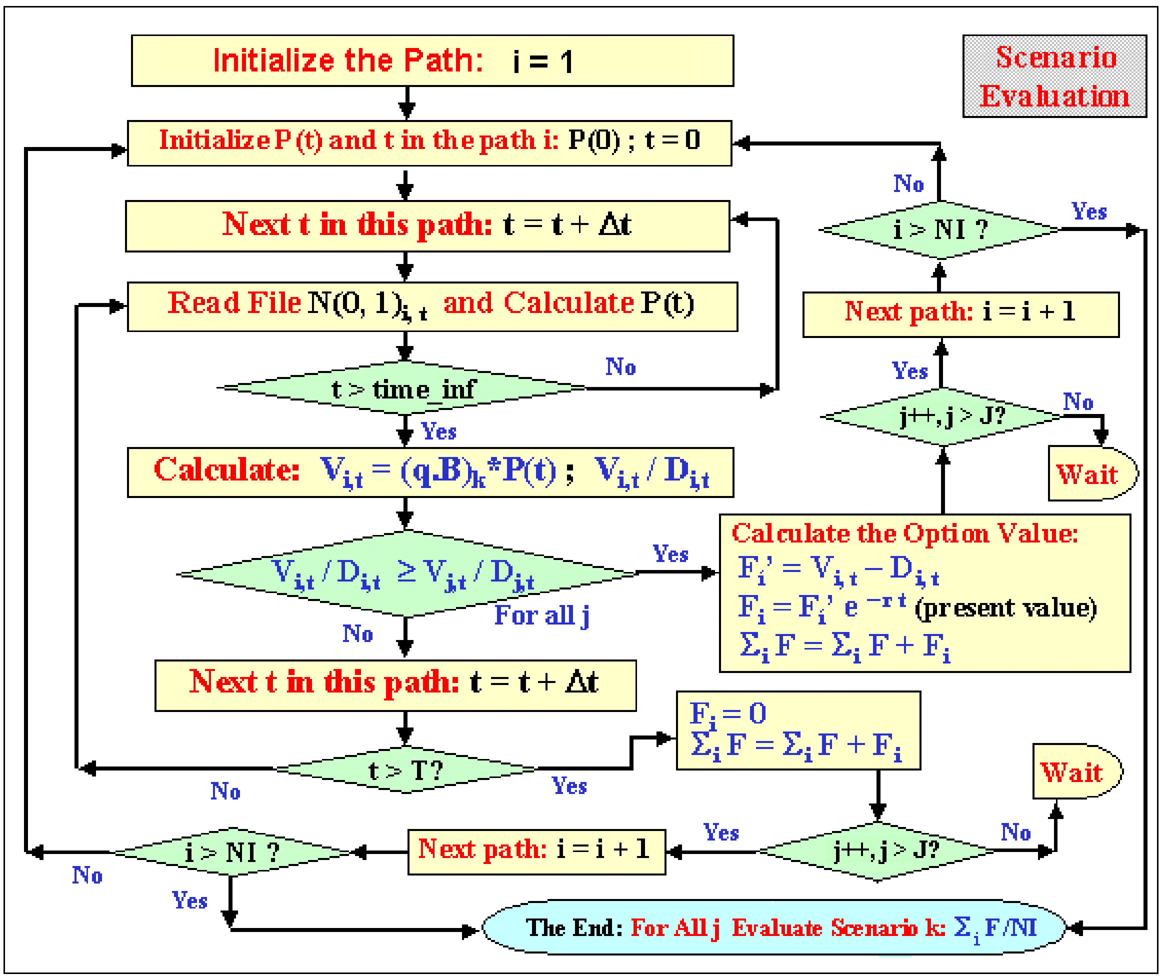

Figure 11 – The Flowchart of a Scenario Evaluation

Although the flowcharts present the evaluation of a single chromosome, as is performed in the original Genocop program, the final version of the program evaluate all the population in parallel, improving a lot the computational efficiency[22].

[22] Thanks for Prof. M.A.C. Pacheco for the idea.

4 – Results from the Model

Several experiments were performed for both stochastic processes and for all the alternatives, including the alternative zero, the alternative of not investing in information. In most cases more than one experiment were performed for the same case changing a small genetic algorithm parameter or even without changing anything, just to see the effect of the random initial population in the evolutionary process.

For all cases at least one experiment with population[23] of 40 chromosomes and 30 generations was performed. In reality 31 generations because the original Genocop names generation 1 for the two first generations (it is possible to think the first one as generation zero). The typical computational time for a Pentium III with 800 MHz is presented in the table below (the case presented is the GBM for alternative 3 using the parallel version).

Table 1 – Computational Time for Pentium III, 800 MHz

Total running time for 31 generations x population of 40	1,763 sec. (29.38 min.)
Time for each generation	56.87 seconds
Time for each chromosome	1.42 sec.

The average time for each chromosome depends mainly of the population size. Higher population means lower average time for each chromosome because the program print in the output file the chromosomes genes and its evaluation, in each change of generation (access to the hard disk reduces the computational efficiency). This computational time is excellent considering that one of main criticism of the genetic algorithm is the slowness to reach an acceptable solution.

[23] The optimal balance population x generations x number of experiments is a difficult empirical question. Sometimes even a small population is sufficient to perform a good job. This is the case of Vallée & Basar (1999, p.207), who used an initial population of only 4 chromosomes of 14 genes, and their algorithm got fast convergence (10 to 20 generations) to Stackelberg equilibrium for both action and cost in a fish war game problem.

Now let us see if the evolutionary approach reaches acceptable solutions or not. For this, the case of geometric Brownian motion and the case of mean-reversion are examined separately.

4.1 – Results for the Geometric Brownian Motion

This section is started with the theoretical results for the alternative zero. The theoretical results were calculated using two methods. First was used the software "Extendible" (see Dias & Rocha, 1998) to get directly the real option value for the alternative zero, which reached the value of US$ 60.165 millions. Second, the approximated chromosome was estimated to match the threshold curve generated by the genes (like a theoretical chromosome). Using this "theoretical" chromosome and the same procedure to evaluate a chromosome used by the C++ program, that is, the same file of technical scenarios and the same file with a fixed table of random numbers sampled from a standard Normal distribution[24], the value was US$ 61.174 millions. The difference is due the numerical errors of both the finite differences used in the Extendible and the sampled distribution used in the second case.

The evolutionary program result was very close to the theoretical solution. In reality the GA's real option value was even higher than the theoretical value: US$ 62.367 million! The difference can be attributed to the numerical errors mentioned before and to the distributions sampling imperfections. For the random numbers used for the price paths and scenarios, the GA solution is even better than the "theoretical" solution. Probably this occurred due the imperfections in the sampled numbers used. But given the table of random numbers used, the GA performed its job searching for an optimal (or near) solution.

The chart below illustrated the comparison between the theoretical threshold and the threshold that the evolutionary program reached. Note that only near of the expiration the GA threshold is clearly different of the theoretical solution. The vertical graph scale was amplified to highlight the differences.

[24] An Excel file with Visual Basic macro was developed to both debug the C++ program and to evaluate specific chromosomes. This spreadsheet perform exactly the same job than the C++, except the part of GA. The Excel of course is much slower than C++, but permits to examine specific chromosomes, including the "theoretical" chromosome.

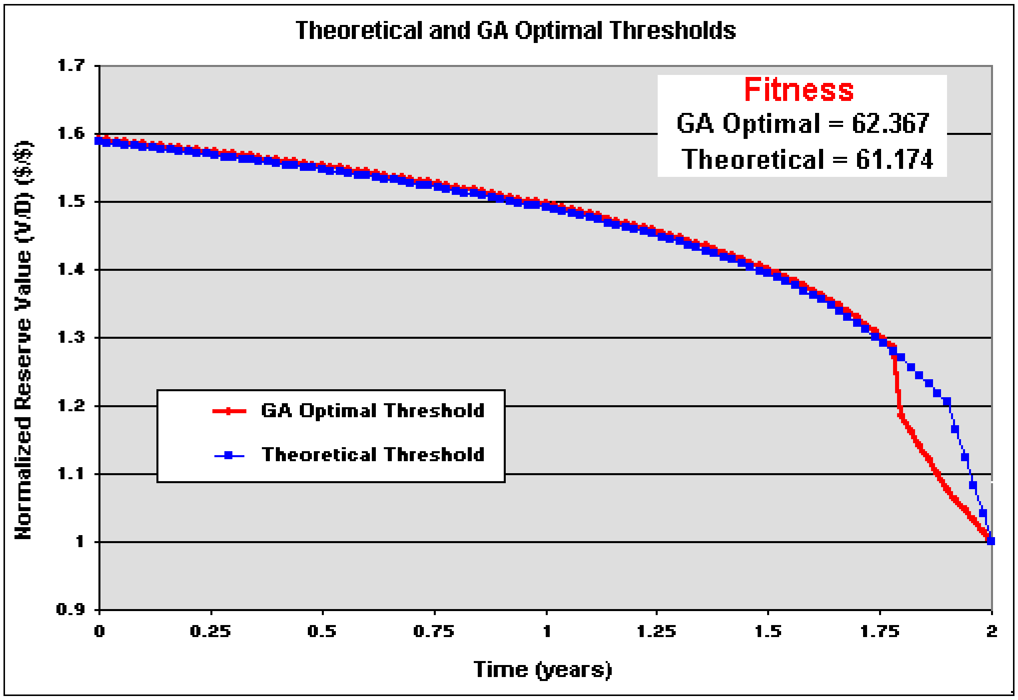

Figure 12 – Comparison of the Theoretical and the GA Solutions (Alternative 0)

The table below presents the best GA solutions for each alternative of investment in information. Sometimes a larger experiment does not improve the solution. The vector X is the chromosome and comprises 4 values (the genes), as described before.

Table 2 – The Results of the Alternatives: the Geometric Brownian Motion Case

Vector X	Alternative	Case gen/pop_size	Best Cromosome (Real Option) Value	Generation of the best
1.07775891 1.18433571 1.49595857 0.13903511	0	46/40	62.367344	20
1.07506621 1.25485146 1.41950107 0.11539662	1	30/40	58.8848	28
1.10754013 1.17158198 1.50048268 0.1630988	2	30/40	60.37175751	30
1.12806344 1.20405042 1.4649235 0.15491864	3	30/40	62.04074097	29
1.13385618 1.17532325 1.4527483 0.09308907	4	30/40	63.13252258	29

Note that **the best alternative is the alternative 4** (the more expensive but the alternative that reveals more information). The alternative 3 has approximately the same value of alternative of not investing in information. The alternatives 1 and 2 are worse than the alternative zero, so it is better not investing in information than invest in the alternatives 1 and 2.

4.2 – Results for the Mean Reversion

The mean reversion case is expected to present lower real options value: because the reversion force, the price path cannot go to far from the US$ 20/bbl level as occurred with the geometric Brownian motion. In addition, is expected that the threshold level to be a bit lower than the GBM case.

The following chart presents the GA threshold for the best experiment for the alternative 4 using mean-reversion. Note that the threshold level is inferior to the GBM case.

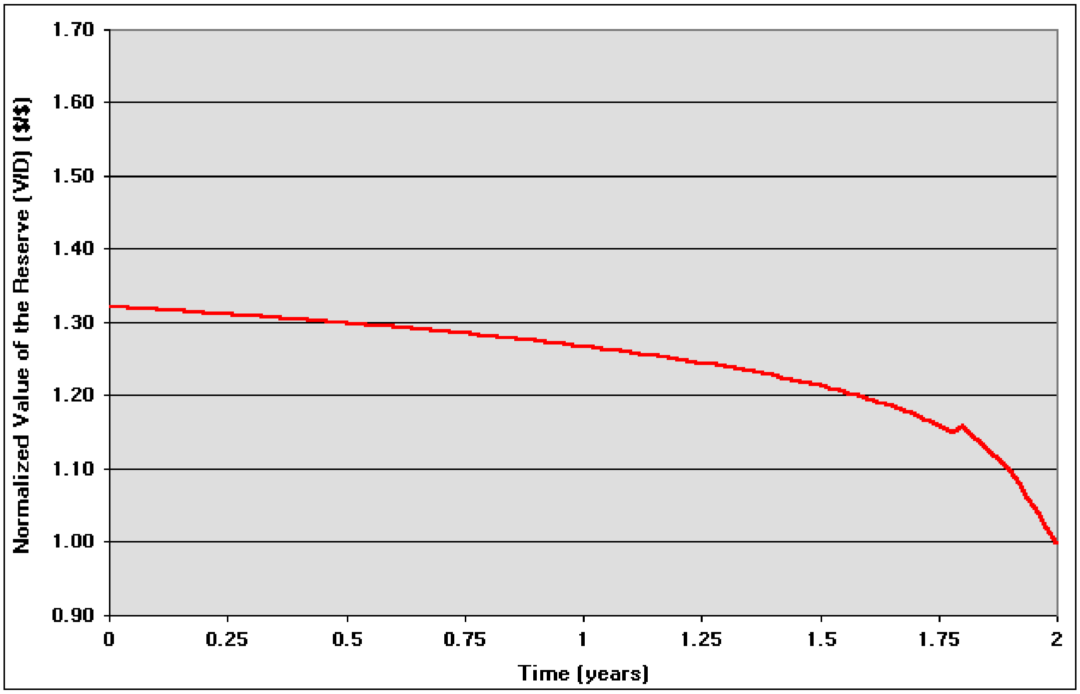

Figure 13 – The Mean-Reversion Threshold Case: Alternative 4

The following figure presents the GA threshold for the best experiment for the alternative 3 using mean-reversion. Note that the threshold solution is very close of the alternative 4.

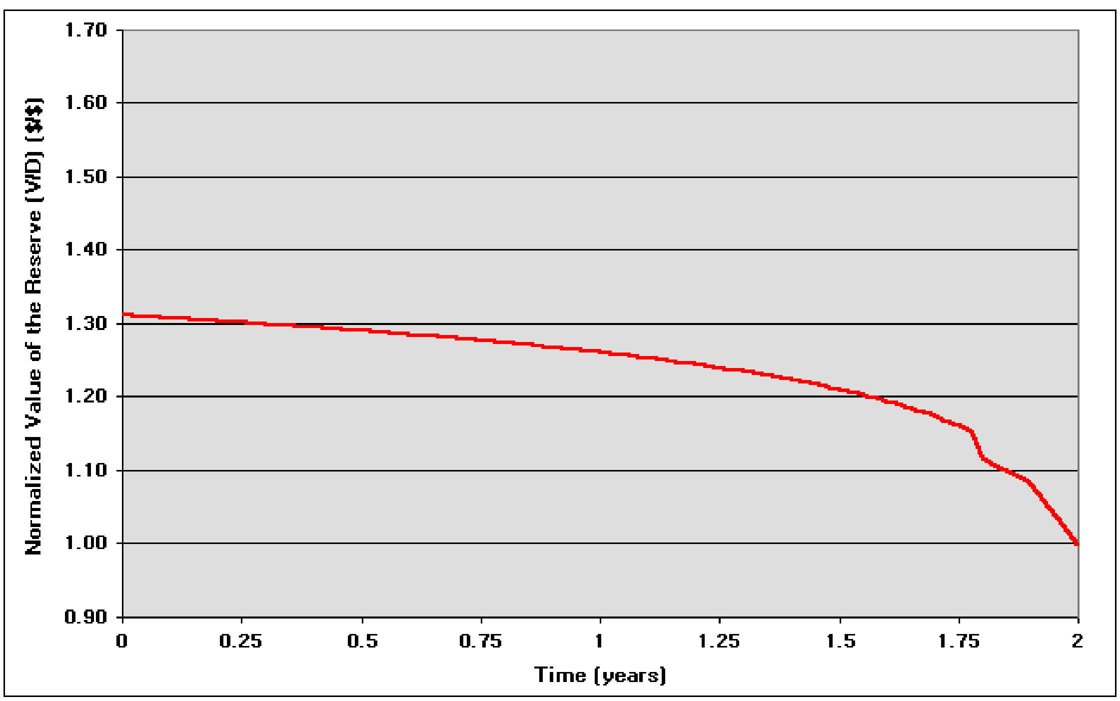

Figure 14 – The Mean-Reversion Threshold Case: Alternative 3

The following table presents all the best experiment results for the alternatives in the mean-reversion case. Again **the alternative 4 was the best one**. In this case, however, the alternative 3 is slightly better than in the GBM case. Again the alternatives 1 and 2 are worse than not investing in information (alternative 0), but this time the alternative 2 is very close of alternative 0.

Table 3 – The Results of the Alternatives: the Mean-Reversion Case

Vector X	Alternative	Case gen/pop_size	Best Cromosome (Real Option) Value	Generation of the best
1.14562201 1.15272117 1.25457656 0.07184473	0	60/40	49.04401398	40
1.03630149 1.04026902 1.22839284 0.0424125	1	25/20	46.12804794	25
1.13232374 1.16823387 1.22926068 0.04372992	2	30/40	48.77129364	30
1.08463931 1.1159513 1.26126921 0.07226396	3	30/40	50.0196991	30
1.10089219 1.15892744 1.2680186 0.07771666	4	30/40	51.71102524	30

In order to illustrate the performance of the evolutionary programming, the figure below presents the evolution of the best chromosome through the generations. Sometime a new best solution causes a jump in this performance curve.

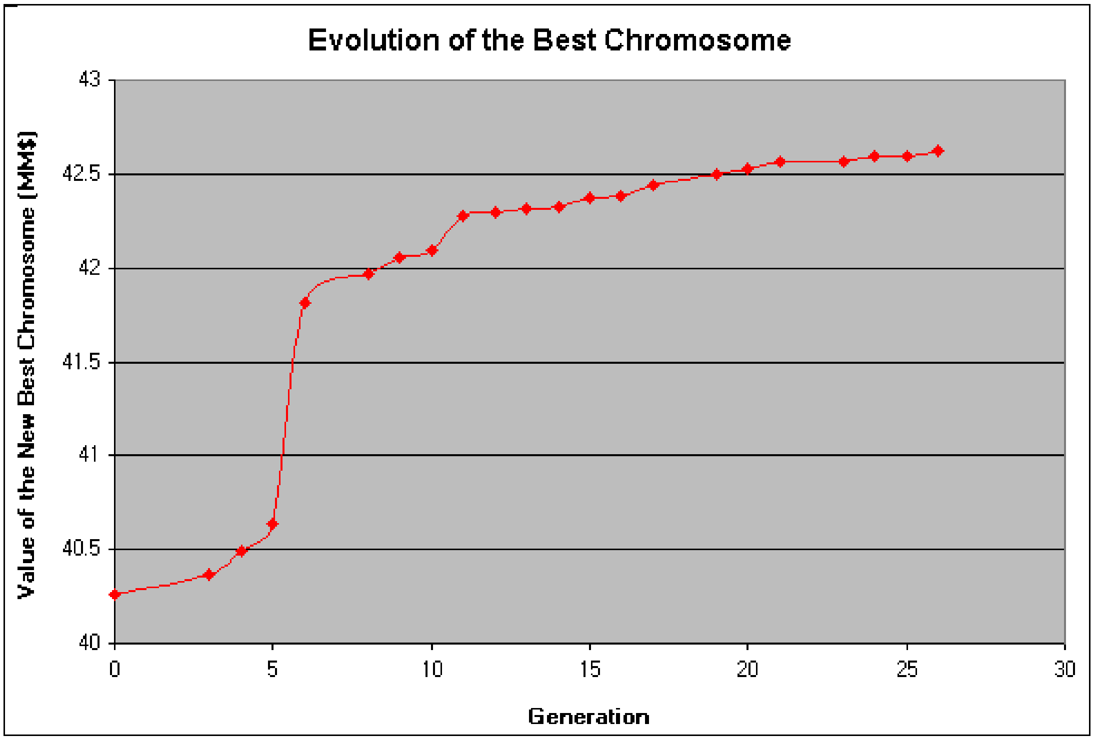

Figure 15 – The Best Solution Evolution through the Generations (mean-reverting case)

6 – <u>Concluding Remarks</u>

The evolutionary real options approach used to select the best alternative to invest in information revealed competitive with other more traditional methods like the PDE approach. In addition, the evolutionary approach looks much more flexible when choosing the stochastic processes and the way to combine technical and market uncertainties.

The best alternative in this paper was just the more expensive (alternative 4), but revealing more information than the cheaper alternatives. This result was robust with the stochastic processes used. Although the real options value is lower for mean-reverting case, the ranking of alternatives remained the same when passing from the GBM to mean-reverting case.

One important extension is to allow jumps (Poisson process) combined with the mean-reversion, as is done in Dias & Rocha (1998). This improvement will demand the program to read another file, the jump frequency and size probabilities distribution.

The faster C++ environment permitted more effective software for professional applications. Additional optimizations are possible, but the current computational time is competitive for practical applications.

Acknowledgements: Petrobras supported this research. The author wishes to express his gratitude to his research colleagues in the discipline of "Topics of Applied Computational Intelligence", Dept. of Electrical Engineering, PUC-Rio, with their comments that enriched the paper. Special acknowledgements are due to Prof. M.A.C. Pacheco, mainly for the suggestion that improves a lot the computational speed; to Profa. M.M.R. Vellasco, and to A.F.G. Dias, software consultant, which was decisive in the process of building the C++ software. Any error is mine.

References

Clewlow, L. & C. Strickland (1998): "Implementing Derivatives Models"
John Wiley & Sons Ltd., 1998, 309 pp.

Cortazar, G. & E.S. Schwartz (1998): "Monte Carlo Evaluation Model of an Undeveloped Oil Field"
Journal of Energy Finance & Development, vol.3, nº 1, pp.73-84; and also Working Paper UCLA #1-98, PUC-Chile and UCLA, January 1998, 20 pp.

Davis, L. (Eds.) (1991): "Handbook of Genetic Algorithms"
Van Nostrand Reinhold, New York, 1991, 385 pp.

Dias, M.A.G. (2000): "Real Options Evaluation: Optimization under Uncertainty with Genetic Algorithms and Monte Carlo Simulation"
Working Paper, Dept. of Electrical Engineering, PUC-Rio, July 2000, 23 pp.

Dias, M.A.G. (1999): "A Note on Bibliographical Evolution of Real Options"
in L. Trigeorgis, Eds., *Real Options and Business Strategy – Applications to Decision Making* – Risk Books, 1999, pp. 357-362

Dias, M.A.G. & K.M.C. Rocha (1998): "Petroleum Concessions with Extendible Options Using Mean Reversion with Jumps to Model Oil Prices"
Paper presented at the to the 3rd Annual International Conference on Real Options, June 1999, Netherlands; previous version of this paper were present in 1998 at Stavanger, Norway (May), and Vitória, ES, Brazil (December).

Dixit, A.K. & R.S. Pindyck (1994): "Investment under Uncertainty"
Princeton University Press, Princeton, N.J., 1994, 468 pp.

Goldberg, D.E. (1989): "Genetic Algorithms in Search, Optimization & Machine Learning"
Addison Wesley Longsman, Inc., 1989, 412 pp.

Holland, J.H. (1975): "Adaptation in Natural and Artificial Systems"
MIT Press Edition, 1992 (original 1975), 211 pp.

Ibáñez, A. & F. Zapatero (1999): "Monte Carlo Valuation of American Options Through Computation of the Optimal Exercise Frontier"
Working Paper, Instituto Tecnológico Autónomo de México & University of South California, August 1999, 30 pp.

Ju, N. (1998): "Pricing an American Option by Approximating Its Early Exercise Boundary as a Multipiece Exponential Function"
Review of Financial Studies, vol.11, nº 3, Fall 1998, pp.627-646

Kloeden, P.E. & E. Platen (1992): "Numerical Solution of Stochastic Differential Equations"
Springer-Verlag, 1992, 636 pp.

Koza, J.R. (1992): "Genetic Programming – On the Programming of Computers by Means of Natural Selection"
MIT Press, 1992, 819 pp.

McDonald, R. & D. Siegel (1986): "The Value of Waiting to Invest"
Quarterly Journal of Economics, November 1986, pp.707-727

Michalewicz, Z. (1996): "Genetic Algorithms + Data Structures = Evolution Programs"
Springer-Verlag, Third Edition, 1996, 387 pp.

Michalewicz, Z. & D.B. Fogel (2000): "How to Solve It: Modern Heuristics"
Springer-Verlag, corrected second printing, 2000, 467 pp.

Mohaghegh, S. (2000): "Virtual-Intelligence Applications in Petroleum Engineering: Part 2 – Evolutionary Computing"
Journal of Petroleum Technology, October 2000, pp.40-46

Pacheco, M.A.C. (2000): "Algoritmos Genéticos: Princípios e Aplicações" ("Genetic Algorithms: Principles and Applications")
Class Notes, ICA, PUC-Rio, 2000, 6 pp. (in Portuguese)

Schwartz, E.S. (1997): "The Stochastic Behavior of Commodity Prices: Implications for Valuation and Hedging"
Journal of Finance, vol.52, nº 3, July 1997, pp.923-973

Trigeorgis, L. (1996): "Real Options - Managerial Flexibility and Strategy in Resource Allocation"
MIT Press, Cambridge, MA, 1996, 427 pp.

Vallée, T. & T. Basar (1999): "Off-Line Computation of Stackelberg Solutions with Genetic Algorithm"
Computational Economics, vol.13, pgs.201-209

Appendix: Software Interface

There are two versions for the C++ software in function of the stochastic process used for the oil prices. One is the Geometric Brownian Motion C++ version and the other one is the Mean-Reversion C++ version. The figure below presents the latter version.