

4

Metodologia Proposta para o Cálculo do Valor de Opções Reais por Simulação Monte Carlo com Aproximação por Números *Fuzzy* e Algoritmos Genéticos.

4.1.

Introdução

Neste capítulo descreve-se em duas partes a metodologia proposta para a aproximação do valor de opções reais e da regra de decisão ótima para várias opções de investimento em um projeto, considerando incertezas técnicas e de mercado.

Na primeira parte descreve-se a metodologia para aproximar o valor da opção real, baseada em números *fuzzy* para representar incertezas técnicas e processos estocásticos conhecidos para representar a incerteza de mercado (preço da *commodity*), unidos à simulação estocástica (simulação Monte Carlo), com o objetivo de reduzir o tempo computacional da simulação Monte Carlo.

Na segunda parte descreve-se a metodologia para aproximar uma regra de decisão ótima e determinar o valor da opção real para o caso de se ter várias alternativas (opções) de investimento em um projeto. Este método emprega um algoritmo genético e processos estocásticos conhecidos para representar a incerteza de mercado (preço da *commodity*), unidos à simulação estocástica (simulação Monte Carlo) e às técnicas de redução de variância.

4.2.

Descrição da Metodologia Proposta de Avaliação de Opções Reais por Aproximação com Números *Fuzzy*

Nesta seção é descrita a metodologia proposta que combina a simulação estocástica com os números *fuzzy* e processos estocásticos para calcular o valor de uma opção real. Nesta metodologia modela-se a incerteza de mercado por processos estocásticos conhecidos (movimento geométrico browniano, processo de reversão à média, processo de reversão à média com saltos, etc.) e as incertezas

técnicas por números *fuzzy* triangulares, ao invés da distribuição de probabilidade triangular como é usado na solução tradicional de avaliação de opções reais por simulação estocástica.

Na Figura 16 mostram-se os principais módulos da metodologia proposta e a seguir, descreve-se detalhadamente cada módulo.

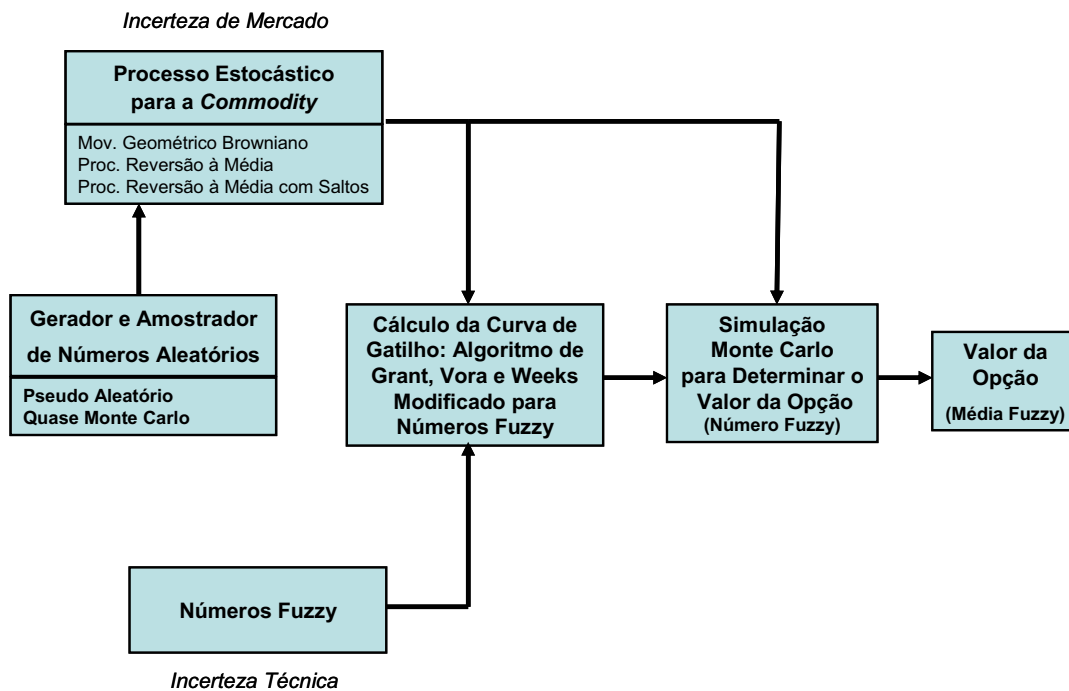


Figura 16 – Principais partes da metodologia proposta para determinar o valor da opção por aproximação com números *fuzzy*

4.2.1.

Módulo: Gerador e amostrador de números aleatórios

Este primeiro módulo corresponde ao gerador de números aleatórios [64] [65] [66] e ao amostrador de números aleatórios da distribuição normal (de média zero e variância um, $N(0,1)$). Estes números são usados no processo aleatório para o preço da *commodity*. Este módulo é importante porque apesar do processo de simulação necessitar de um grande número de amostras aleatórias (números aleatórios), na prática não é possível obter esta grande quantidade de números aleatórios. Para resolver este problema são utilizados os geradores.

Neste trabalho foram empregados dois tipos de geradores de números aleatórios: o primeiro denominado Pseudo-aleatório ou *Simple Random Sampling*

(SRS) [67] [68] e o segundo um gerador de seqüências de baixa discrepância ou quase Monte Carlo [68] [69] [70] [71] [72].

Os geradores são funções que retornam números aparentemente aleatórios (Pseudo-aleatórios) de uma distribuição uniforme de um determinado intervalo $[0, 1]$. Estes números são finitos e seguem uma seqüência; assim, quando o último número da seqüência é fornecido pelo gerador, a seqüência se repete. Por esta razão, é importante contar com um gerador de números aleatórios com uma seqüência o suficientemente grande para evitar que esta se repita várias vezes durante o processo de simulação em um determinado problema.

O gerador implementado neste trabalho consegue gerar 2^{319} números antes de repetir a seqüência. Este gerador garantiu que a seqüência não se repetiu nos experimentos realizados.

O gerador de números Quase Monte Carlo (Quase Aleatório) ou de seqüências de baixa discrepância [70] [71] implementado neste trabalho permite criar uma seqüência de 2^{32} números. Devido ao fato destes números estarem espalhados uniformemente no intervalo $[0, 1]$, nos problemas práticos de simulação são necessários menos números que quando se usa um gerador Pseudo-Aleatório (vide Apêndice B).

O amostrador de números aleatórios permite obter realizações de uma determinada distribuição de probabilidade [67] [68] [73] [74], isto é, transforma o número de uma distribuição uniforme fornecido pelo gerador de números aleatórios em um número de uma distribuição normal com média zero e variância um, $N(0, 1)$.

4.2.2.

Módulo: Processo Estocástico para a *Commodity*

Neste módulo encontram-se os diferentes processos estocásticos que podem seguir o preço da *commodity* (incerteza de mercado) [1] [6] [8]. Foram considerados os seguintes processos estocásticos (vide Apêndice C):

- Movimento Geométrico Browniano;
- Processo de Reversão à Média;
- Processo de Reversão à Média com Saltos.

Estes processos recebem em cada instante de tempo um número aleatório de uma distribuição normal, $N(0, 1)$, e retornam o valor do preço da *commodity* para esse instante.

4.2.3. Módulo: Números *Fuzzy*

Este módulo é responsável por representar cada incerteza técnica por um número *fuzzy* triangular.

O número *fuzzy* triangular usado para representar incerteza tem os mesmos parâmetros que as distribuições de probabilidade triangular usadas pela metodologia tradicional de solução. Em outras palavras, a forma do triângulo que representa o número *fuzzy* é a mesma que do triângulo que representa a distribuição de probabilidade.

Este tipo de representação é bastante usada na bibliografia de números *fuzzy* (parâmetros do número *fuzzy* igual aos parâmetros da distribuição de probabilidade). Entretanto, isso não significa que esta forma de definir os parâmetros do número *fuzzy* seja a que melhor representa a incerteza definida pela distribuição de probabilidade. Até agora, no entanto, não se tem bibliografia que descreva alguma forma de mapear a incerteza representada por uma distribuição de probabilidade triangular em um número *fuzzy* triangular.

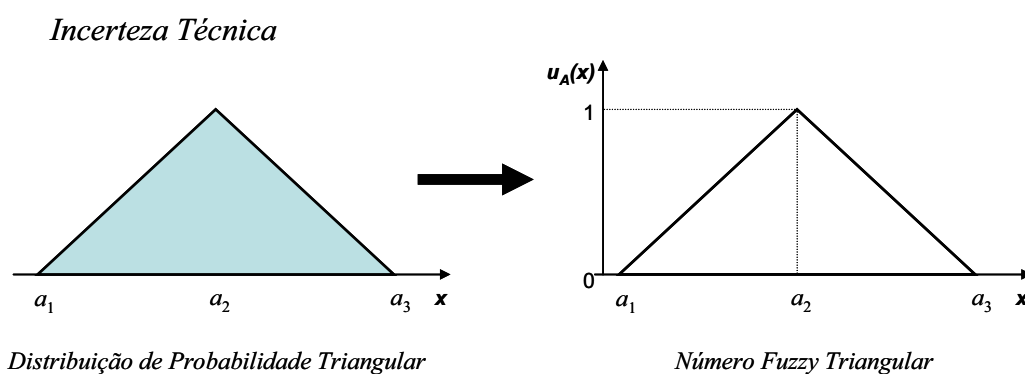


Figura 17 – Representação da incerteza técnica por um número *fuzzy* triangular

4.2.4. Módulo: Cálculo da Curva de Gatilho

No processo de avaliação de uma opção real é preciso determinar a curva de exercício ótimo da opção, ou curva de gatilho. Esta curva representa a fronteira entre o exercício da opção, na parte superior da curva, e o não exercício ou manter a opção até ter melhores condições, na parte inferior da curva.

Para determinar esta curva de gatilho da opção emprega-se o algoritmo de Grant, Vora e Weeks [4], o qual é um algoritmo recursivo que deve ser utilizado seguindo a programação dinâmica, começando na expiração e decrementando o tempo até o momento atual. Maiores detalhes são fornecidos no Apêndice D.

Neste trabalho, o algoritmo de Grant, Vora e Weeks foi modificado para trabalhar com números *fuzzy*. Deste modo, a equação que calcula o valor da opção é transformada em uma equação *fuzzy*, onde as variáveis que representam as incertezas técnicas são números *fuzzy*, a variável que representa a incerteza de mercado (preço da *commodity*) é considerada como singleton e as constantes continuam como constantes.

A seguir é descrito de forma esquemática o algoritmo de Grant, Vora e Weeks modificado para trabalhar com números *fuzzy*. Para isso define-se S como o valor do ativo objeto e X como o preço de exercício da opção. Como foi visto no capítulo 2, o valor de uma opção de compra é calculado por:

$$C_t(S_t, X) = \text{Max}(0, S_t - X) \quad (38)$$

onde C_t é o valor da opção em t , que agora é um número *fuzzy*.

Passo 1: Discretiza-se o tempo de vida da opção em um número determinado de intervalos e considera-se como condição de fronteira no vencimento da opção $S_T^* = X$, onde S_T^* é um ponto da curva de gatilho; este primeiro ponto de curva é um singleton. No instante T , no vencimento, é ótimo exercer a opção sempre que ela estiver “*in the money*” (Figura 18).

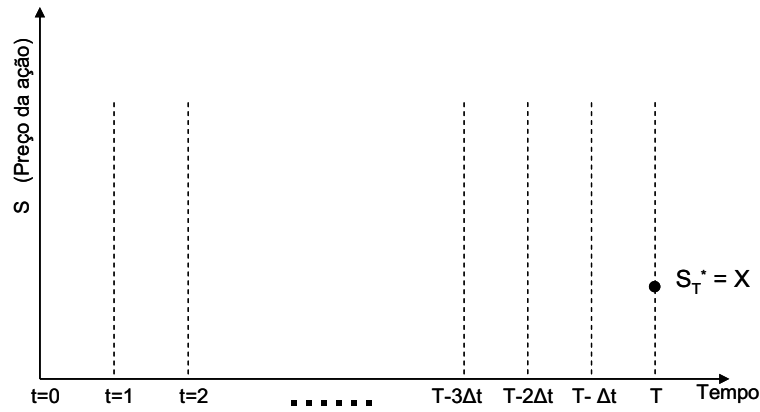


Figura 18 - Discretização do tempo no algoritmo de Grant, Vora e Weeks

Passo 2: No instante $T - \Delta t$ adota-se o preço inicial do ativo, $S_{T-\Delta t}$, igual ou próximo de S_T^* . Inicia-se a simulação Monte Carlo obtendo-se diversos valores para a opção no tempo T . Cada valor da opção calculado na simulação é um número *fuzzy*. O valor final da opção é calculado usando a equação (39) e descontado de e^{-tr} para levar o valor ao tempo $T - \Delta t$ (Figura 19).

$$C_{t_F}(S_t^*) = \frac{1}{N} \sum_{i=1}^N C_{t_F}^{(i)}(S_t^*) \quad (39)$$

onde N é o número de iterações da simulação Monte Carlo. Note-se que este valor final da opção é também um número *fuzzy*.

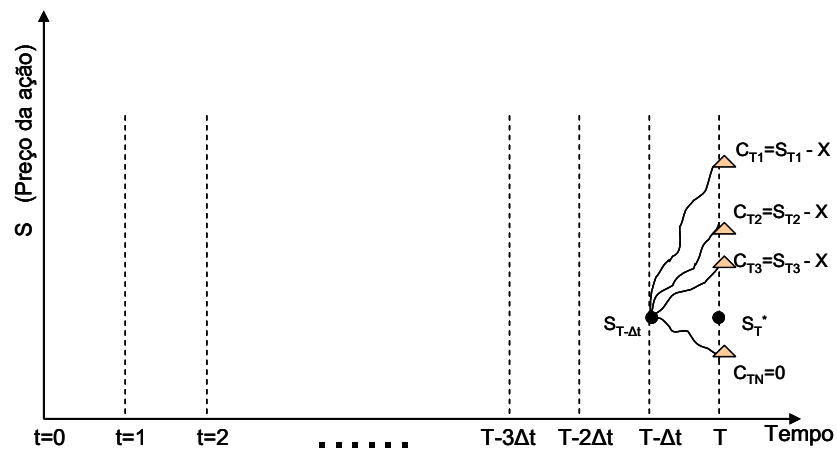


Figura 19 - Cálculo do valor da opção no tempo T por simulação Monte Carlo

Passo 3: Se o preço $S_{T-\Delta t}$ estimado pela equação (40) não for o preço crítico (preço para o exercício ótimo da opção), este é incrementado e repete-se a simulação Monte Carlo (Figura 20).

$$E[C_{T-\Delta t}] = S_{T-\Delta t}^* - X \quad (40)$$

onde $E[C_{T-\Delta t}]$ é média da simulação e é também um número *fuzzy*.

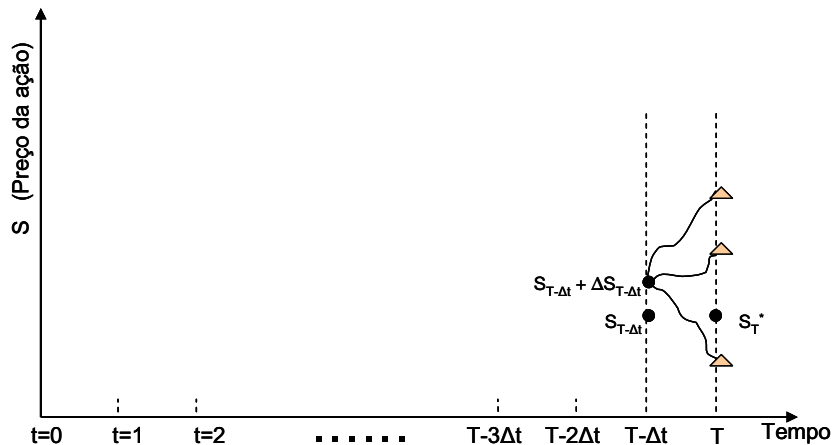


Figura 20 - Incremento de $S_{T-\Delta t}$ e simulação Monte Carlo

Passo 4: Calculado $S_{T-\Delta t}^*$, equação (39), repetem-se os passos 2 e 3, testando a opção para todos os momentos até o vencimento T . Continua-se recursivamente este processo até chegar ao tempo inicial, t_0 , construindo-se assim a curva de gatilho (Figura 21).

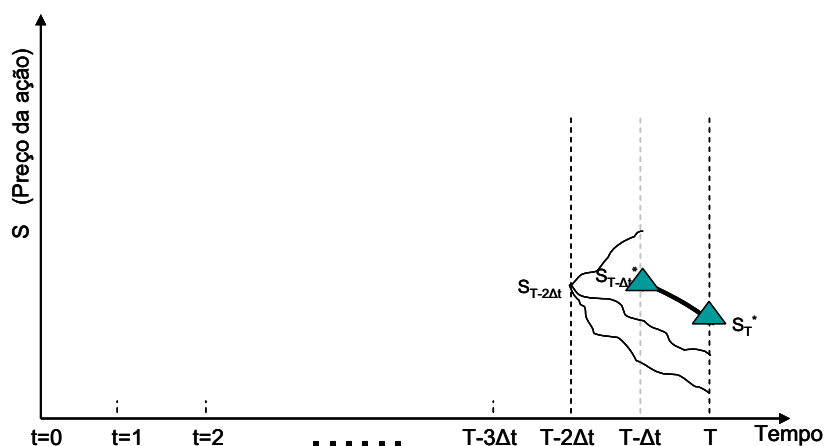


Figura 21 - Repetir recursivamente os passos 2 e 3

Passo 5: Terminada a curva de gatilho, realizam-se novas simulações Monte Carlo a partir do preço inicial S_0 dado pelo mercado. Calcula-se o valor final da opção, que é o valor médio de todas as opções trazidas ao valor presente (Figura 22).

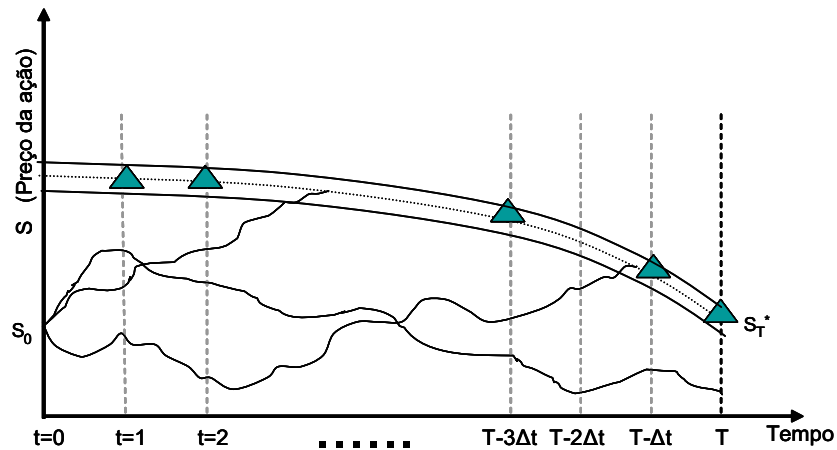


Figura 22 - Cálculo do valor final da opção

Desta forma é construída a curva de gatilho. Note-se que cada ponto da curva é um número *fuzzy*.

4.2.5. Módulo: Simulação Monte Carlo para Determinar o Valor da Opção

Neste módulo calcula-se o valor da opção real seguindo o procedimento descrito a seguir.

A curva de gatilho (*fuzzy*) obtida no módulo anterior define a região de exercício, acima da curva, e a região de espera (manter a opção) abaixo da curva. Com a curva de gatilho pronta, procede-se à simulação dos caminhos para o preço da *commodity* para cada instante t , desde t_0 até a expiração T . O número de iterações da simulação Monte Carlo corresponde ao número de caminhos para o preço da *commodity* que se vai simular.

Cada caminho do preço é comparado com o preço crítico da curva de gatilho em cada instante t . Se o preço da *commodity* supera a curva de gatilho (alcança a região de exercício), o valor da opção é calculado para esse preço em t , (segundo a equação do valor da opção para cada problema, de forma geral $C_t(S_t, X) = \text{Max}(0, S_t - X)$), em seguida, passa-se para a próxima iteração

(próximo caminho do preço da *commodity*). Quando o exercício é feito no instante $t > 0$, o valor da opção é atualizado pela taxa de desconto livre de risco, obtendo-se assim o valor da opção para essa iteração (V_{opF}). Se o caminho do preço é concluído sem ter sido alcançada a região de exercício, então o valor da opção para esse caminho é zero ($V_{opF} = 0$). O valor da opção real resultante da simulação (V_F) é a média dos valores da opção obtidos em cada simulação:

$$V_F = \frac{1}{N} \sum_{i=1}^N V_{opF}^{(i)} \quad (41)$$

onde N é o número de iterações da simulação Monte Carlo (número de caminho do preço da *commodity*).

4.2.6.

Módulo: Valor da Opção

Neste módulo transforma-se em um número real o valor da opção real resultante da simulação (V_F), que é um número *fuzzy*. Isto é necessário para se poder comparar o resultado com a metodologia tradicional (simulação Monte Carlo pura) e porque, na prática, a decisão gerencial é tomada considerando o valor da opção como um número real.

Assim, o valor da opção real é a média do número *fuzzy* resultante da simulação (V_F). O cálculo desta média é feito segundo mostrado em [75] [76] [77] (vide Apêndice E).

No Apêndice F são descritos, de forma esquemática, as metodologias de avaliação de opções reais, tanto a metodologia tradicional por simulação estocástica como a metodologia híbrida com números *fuzzy* proposta (ver figuras 46, 47, 48, 49 e 50).

4.3.

Descrição da Metodologia Proposta para a Obtenção de uma Regra de Decisão Ótima por Aproximação com Algoritmos Genéticos

Nesta seção é descrita a metodologia proposta que combina a simulação estocástica com um algoritmo genético e processos estocásticos para aproximar uma regra de decisão ótima (curva de gatilho) e determinar o valor da opção real no caso de se ter várias opções de investimento em um projeto.

Nesta metodologia modela-se a incerteza de mercado por processos estocásticos conhecidos (movimento geométrico browniano e processo de reversão à média) e considera-se que não existem incertezas técnicas.

Na Figura 23 mostram-se os principais módulos da metodologia proposta e a seguir descreve-se cada módulo.

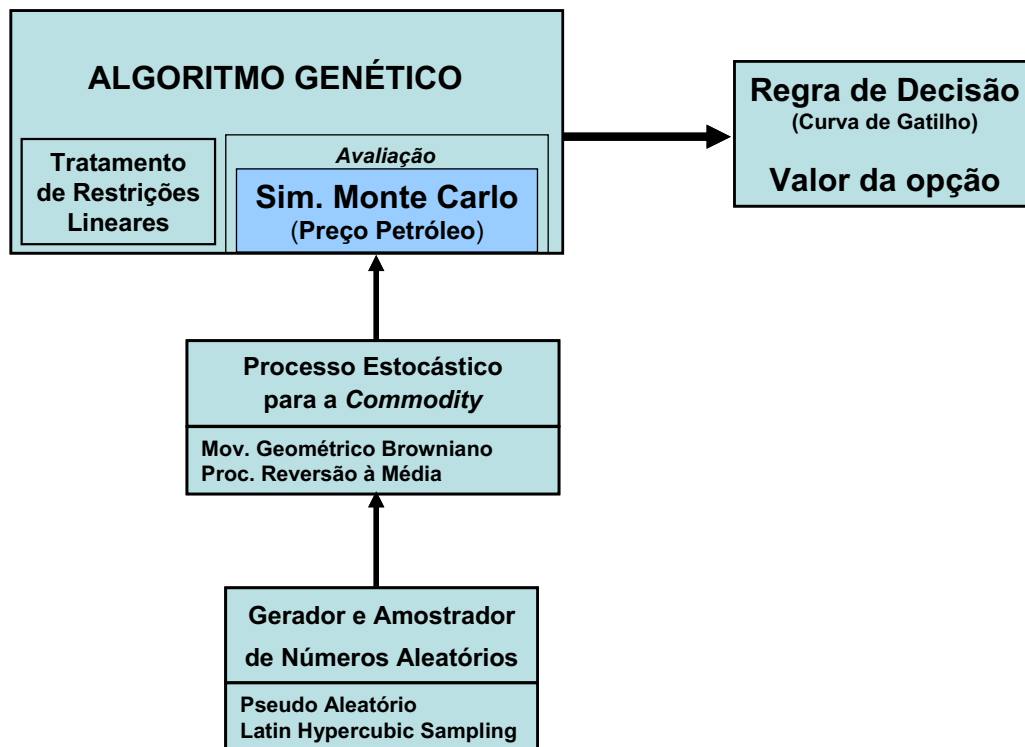


Figura 23 – Principais partes da metodologia proposta para obter uma regra de decisão ótima e determinar o valor da opção por aproximação com um algoritmo genético

4.3.1. Módulo: Gerador e amostrador de números aleatórios

Este módulo é similar ao descrito na seção 4.2.1, o qual é responsável por gerar números aleatórios [64] [65] [66] e amostrar números aleatórios da distribuição normal (de média zero e variância um, $N(0,1)$). Estes números são usados no processo aleatório para o preço da *commodity*, para construir os caminhos do preço.

Empregaram-se dois tipos de geradores de números aleatórios: o primeiro um Pseudo-aleatório ou *Simple Random Sampling* (SRS) [67] [68] e o segundo que faz uso dos números deste gerador e aplica a técnica de redução de variância

denominada *Latin Hypercube Sampling* (LHS) [9] [10] [78] para gerar números de uma distribuição normal $N(0,1)$ espalhados uniformemente (vide Apêndice B).

O amostrador de números aleatórios permite obter realizações de uma determinada distribuição de probabilidade [67] [68] [73] [74], isto é, transforma o número de uma distribuição uniforme fornecido pelo gerador de números Pseudo-aleatórios em um número de uma distribuição normal com média zero e variância um, $N(0, 1)$.

4.3.2.

Módulo: Processo Estocástico para a *Commodity*

Neste módulo encontram-se os processos estocásticos que podem seguir o preço da *commodity* (incerteza de mercado) [1] [6] [8]. Foram considerados os seguintes processos estocásticos (vide Apêndice C):

- Movimento Geométrico Browniano;
- Processo de Reversão à Média.

Estes processos recebem em cada instante de tempo um número aleatório de uma distribuição normal, $N(0, 1)$, e retornam o valor do preço da *commodity* para esse instante.

4.3.3.

Módulo: Algoritmo Genético

Este módulo contém o algoritmo genético, implementado como uma biblioteca em C++ Builder, em cooperação com técnicos do ICA. Esta biblioteca de algoritmos genéticos foi inspirada no sistema Genocop (Genetic Algorithm for Numerical Optimization for Constraint Problems) [14] e baseada em templates e programação orientada a objetos, o que permitiu uma construção modular. Um módulo associado ao algoritmo permite tratar problemas com restrições (de domínio, lineares e não lineares).

O algoritmo genético é usado para aproximar a curva de gatilho e determinar o valor da opção quando se tem várias opções de investimento em um projeto. Cada opção de investimento no projeto tem sua própria curva de gatilho. Logo, a regra de decisão é formada pela interseção das curvas de gatilho de cada alternativa de investimento (opção).

Cada curva de gatilho é aproximada por uma curva logarítmica do tipo $a + b \log(\tau)$ mais um ponto livre que é situado próximo da expiração da opção, o que ocorre em um tempo τ (tempo de expiração da opção). Considera-se também a possibilidade de existirem períodos de espera entre as regiões formadas pelas curvas de gatilho das alternativas. Estas regiões de espera são também aproximadas por funções logarítmicas e um ponto livre: $a_w - b_w \ln(\tau)$. O valor dos coeficientes das funções (a, b, a_w, b_w) e os pontos livres são determinados pelo algoritmo genético

O cromossomo é formado pelos parâmetros da função (a , b e o *ponto livre*) de cada curva e sujeito a um conjunto de restrições que delimitam o espaço de busca, as quais serão apresentadas no capítulo 7. O cromossomo tem uma estrutura matricial, onde cada gene do cromossomo representa uma alternativa e este gene está formado por três alelos que contem os três parâmetros da curva de gatilho da alternativa. Na Figura 24 apresenta-se a estrutura do cromossomo para N alternativas de investimento.

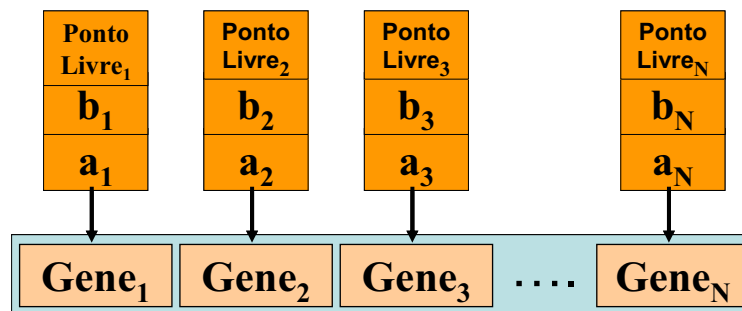


Figura 24 – Estrutura do cromossomo para N alternativas de investimento.

Para tratar as restrições faz-se uso de um módulo dentro da biblioteca de algoritmos genéticos que verifica no cromossomo o cumprimento das restrições para cada alternativa. Assim, se no cromossomo, algum dos parâmetros para a curva de gatilho de uma alternativa não cumpre as restrições, descarta-se só o alelo do gene correspondente a essa alternativa, sendo este substituído por outro. Este esquema apresenta uma vantagem sobre o tratamento tradicional em Algoritmos genéticos, na qual para o mesmo caso de algum dos parâmetros de uma alternativa não cumprir com as restrições, se descarta todo o cromossomo, perdendo-se muito tempo na inicialização e na própria evolução.

O objetivo do algoritmo genético é determinar os valores dos parâmetros das curvas de gatilho que maximizem o valor da opção.

Na avaliação do cromossomo emprega-se a simulação Monte Carlo. Através da simulação Monte Carlo simulam-se os caminhos para o preço da *commodity* para cada instante t , desde t_0 até a expiração T . O número de iterações da simulação Monte Carlo corresponde ao número de caminhos para o preço da *commodity* que se vai simular. O cromossomo é avaliado da seguinte forma: com os parâmetros das curvas de gatilhos (genes do cromossomo) constrói-se a regra de decisão (interseção das curvas de gatilho). Logo, cada caminho do preço é comparado com o preço crítico da regra de decisão (curva de gatilho) em cada instante t . Se o preço da *commodity* supera a curva de gatilho (alcança a região de exercício), o valor da opção é calculado para esse preço em t (de forma geral $C_t(S_t, X) = \text{Max}(0, S_t - X)$). Este valor da opção representa a avaliação do cromossomo para essa iteração (caminho do preço). Caso contrário, isto é, se o preço da *commodity* não supera a curva de gatilho, o valor da opção é zero. Na Figura 25 mostra-se um exemplo de dois caminhos diferentes para o preço da *commodity* até a expiração da opção. No primeiro, o preço da *commodity* intercepta a curva de gatilho no tempo t_1 ; nesse momento a regra de decisão determina que a opção de investimento representada pela curva de gatilho seja exercida. Deste modo, calcula-se o valor da opção nesse momento (t_1) e este valor se traz ao valor presente multiplicando pelo fator $\exp(-rt)$. O outro caminho do preço da *commodity* passa todo o tempo de vida da opção sem interceptar a curva de gatilho; neste caso o valor da opção é zero.

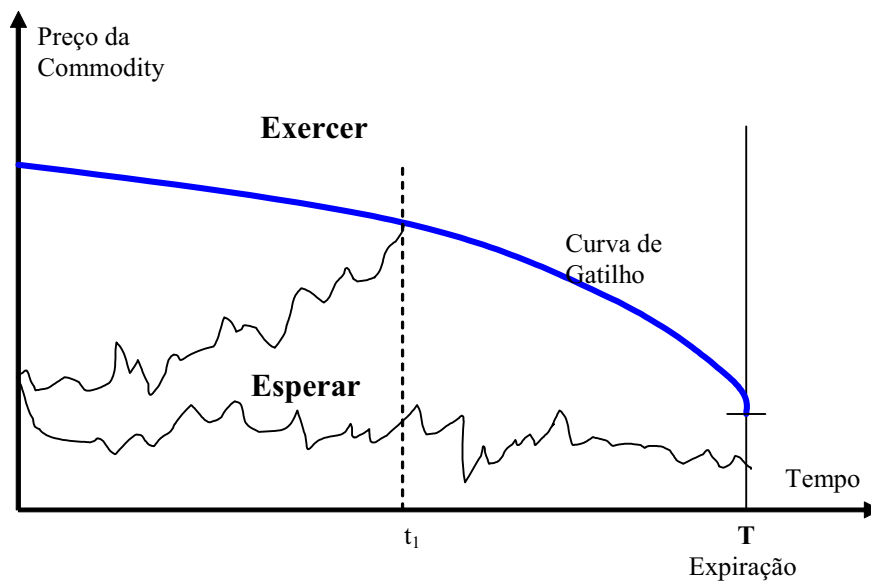


Figura 25 – Curva de gatilho e caminhos do preço da *commodity*.

Repete-se este processo para todos os caminhos. Logo, a avaliação deste cromossomo que o algoritmo genético busca maximizar é a média do valor da opção obtido para cada caminho do preço.

Maiores detalhes do processo de avaliação do cromossomo são apresentados no capítulo 7, na descrição do estudo de casos.

4.3.4. Módulo: Regra de Decisão e Valor da Opção

Este módulo recebe o resultado do algoritmo genético, isto é, o cromossomo com maior avaliação (melhor indivíduo). Este cromossomo contém os parâmetros das curvas de gatilho de todas as opções de investimento e em conjunto representa a regra de decisão ótima. Com estes parâmetros neste módulo é construída a regra de decisão ótima e o valor da opção é o valor da avaliação do cromossomo.