

## **4**

# **Sistema Inteligente de Otimização**

### **4.1.**

#### **Introdução**

Neste capítulo é detalhado o sistema inteligente proposto de otimização de alternativas de desenvolvimento de campos petrolíferos, descrevendo-se a modelagem global do sistema e as partes mais relevantes desta modelagem, a saber: o algoritmo de otimização que usa algoritmos genéticos; o aproveitamento de informações dos mapas de reservatório; a inferência da curva de produção; o modelo econômico para cálculo do valor presente líquido. Em seguida, são descritos a implementação do sistema e o uso de ambientes paralelos para diminuir o tempo computacional.

### **4.2.**

#### **Modelagem do Sistema**

Um sistema de otimização iterativo, como é o caso de algoritmos genéticos, pode ser dividido em dois blocos: o algoritmo de otimização e a função objetivo. No sistema proposto, o algoritmo de otimização baseia-se em algoritmos genéticos e o bloco de função objetivo é fundamentado no cálculo do valor presente líquido da alternativa.

Dado o problema tratado nesta tese, o bloco de função objetivo recebe a descrição de uma alternativa de desenvolvimento (arranjo de poços para o campo petrolífero), fornecida pelo algoritmo genético, e retorna o valor presente para esta alternativa. Neste processo de avaliação existem dois estágios sucessivos: obtenção as curvas de produção para o arranjo dos poços dado, e o cálculo do valor presente líquido a partir das curvas de produção obtidas e o cenário econômico existente.

A representação completa mais elementar do sistema inteligente de otimização proposto é um diagrama com 3 blocos, como se mostra na Figura 20 a seguir.

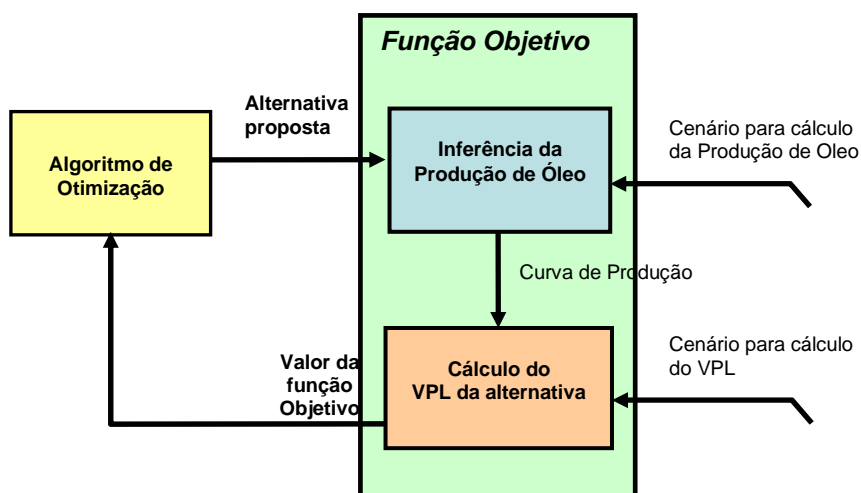


Figura 20. Principais blocos do sistema otimizador proposto

Nota-se na Figura 20 a seqüência de tarefas a serem realizadas durante as muitas iterações efetuadas pelo algoritmo genético:

- O algoritmo de otimização gera um cromossoma que contém a alternativa proposta para ser avaliada;
- Esta alternativa é enviada ao bloco de inferência de produção onde, a partir de informações sobre o campo petrolífero e sobre o cenário de produção, determina-se o perfil de produção da alternativa;
- A partir da distribuição de poços no campo petrolífero, do perfil de produção gerado e do cenário econômico existente (preços, custos, taxas, alíquotas...) é obtida a previsão dos fluxos de caixa de entradas e saídas a serem executados durante o tempo de produção da alternativa. Projetando os fluxos de caixa ao valor presente (vide seção 2.3.2.3) e descontando o custo de investimento, obtém-se o valor presente líquido (VPL) que vem a ser o valor da função objetivo do sistema;
- Por fim, o algoritmo de otimização, implementado por um algoritmo genético, estendido para o algoritmo genético distribuído global da seção (3.4.7.1.1) que, mediante os processos de seleção, reprodução e avaliação, busca a alternativa que maximize o valor presente líquido, isto é, otimiza as alternativas de desenvolvimento de campos petrolíferos.

Nas seções seguintes são detalhados os três blocos principais do sistema proposto.

### 4.3. Algoritmo de Otimização

Nesta seção detalha-se a modelagem feita com algoritmos genéticos como método de otimização da alternativa de produção, onde são descritos: a representação das soluções no cromossoma, os operadores genéticos e as restrições definidas na geração de soluções

#### 4.3.1. Representação das soluções no Cromossoma

Um Algoritmo Genético, como visto na seção (3.4.1), possui uma população de cromossomas, os quais representam soluções do problema em questão. No problema de otimização de alternativas de desenvolvimento de campos petrolíferos, uma solução é a definição do número e localização dos poços produtores e injetores. Portanto, o cromossoma deve representar um conjunto de poços com as suas respectivas características. A representação escolhida é mostrada na Figura 21.

Como não se conhece a priori o número de poços que uma certa alternativa deve conter, utiliza-se neste problema um cromossoma de tamanho variável, onde somente é especificado o número máximo de poços possível. A modelagem utilizada é baseada na metodologia de máscara de ativação proposta por (Zebulum, 1999). A máscara de ativação aparece como um elemento que permite utilizar o poço especificado no cromossoma ou, simplesmente, desconsiderá-lo.

O cromossoma utilizado segue o modelo desenvolvido por Almeida (2003) onde cada cromossoma armazena uma lista de poços contendo localização, tipo de poço (produtor, injetor) e geometria do poço (vertical, horizontal, direção da trajetória e comprimento desta para poço horizontal).

Cada gene do cromossoma contém a posição  $(i, j, k)$  da cabeça do poço<sup>1</sup> e o comprimento e direção  $(c, d)$  da trajetória do poço horizontal. No caso de poços verticais, apenas a parte  $(i, j)$  é manipulada geneticamente, pois não faz sentido ter parâmetros de trajetória. Para o eixo  $k$ , no caso de poços verticais, é considerado sempre o valor máximo  $k$  do *grid*. Isto serve para indicar ao

---

<sup>1</sup> No caso de poços verticais é o próprio poço; para poços horizontais, refere-se à extremidade que possui a conexão com o sistema de produção.

simulador que o poço vertical foi completado<sup>2</sup> para todas as camadas  $(1,2,\dots,k)$  do *grid* de reservatório.

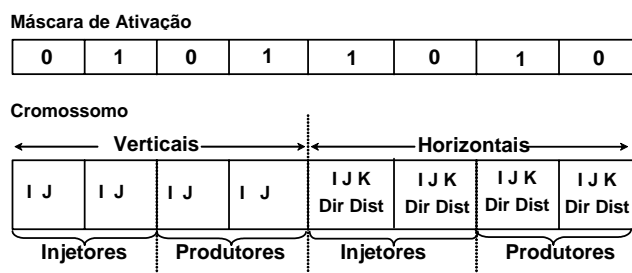


Figura 21. Representação do Cromossoma

Conforme ilustrado na Figura 21, o tipo e a geometria do poço estão determinados pela posição dentro do cromossoma: a primeira metade do cromossoma corresponde aos poços verticais, e a segunda metade aos poços horizontais. Dentro de cada uma destas metades tem-se uma segunda subdivisão onde a primeira refere-se a poços injetores e a segunda a poços produtores. Nesta figura ilustra-se um exemplo de cromossoma contendo 8 genes. Assim, a alternativa representada por este cromossoma pode conter como máximo 8 poços. Pela máscara de ativação mostrada na figura, apenas 4 destes poços são considerados no momento da avaliação, ou seja o cromossoma efetivamente contém 4 poços. Cabe ressaltar que para o trabalho foram empregados cromossomas com mais genes do que o exemplo da Figura 21.

#### 4.3.2. Operadores Genéticos

Na modelagem do Algoritmo Genético foram empregados vários operadores, alguns convencionais e outros criados para o problema em questão. Os operadores genéticos utilizados foram os seguintes:

Mutações: Uniforme, *LocalMove*, *AddWell*, *FlipWell*;

Cruzamentos: Aritmético, Simples.

A seguir é descrito o funcionamento de cada operador de mutação e cruzamento citado acima.

Na Mutaç o Uniforme o valor do gene  $g(i)$  do cromossomo escolhido   alterado da seguinte forma:

<sup>2</sup> Para um poço perfurado, a completaç o refere-se ao fato de realizar a conectividade real entre o poço e as camadas de  leo,  gua e g s existentes na reserva.

$$g(i) = \inf + (\sup - \inf) \cdot U(i) \quad (19)$$

onde  $\inf$  e  $\sup$  são os valores ínfimo e supremo do domínio do gene, e  $U(i)$  é uma amostra de uma distribuição uniforme (0,1]. Desta forma, garante-se que o valor do gene será substituído por um valor válido do seu domínio, com distribuição de probabilidade uniforme; daí o nome de mutação uniforme.

Na Mutação *LocalMove* o valor do gene do cromossomo escolhido sofre um deslocamento de  $\pm \Delta$  como na equação (20), onde  $\Delta$  é uma realização de uma variável aleatória com distribuição uniforme  $[-n, n]$ , onde o valor  $n$  é o máximo deslocamento possível definido de acordo com o problema em questão.

$$g(i)' = g(i) \pm \Delta \quad (20)$$

Essa mutação é adequada para o problema pois permite realizar pequenos deslocamentos dos poços existentes nos eixos  $i$  ou  $j$ , explorando a vizinhança de cada poço, como um passeio aleatório. Neste caso específico, o valor  $\Delta$  pode tomar valores do conjunto  $\{-1, 0, 1\}$ . Para evitar possíveis violações de domínio é feita uma verificação após ter sido feita esta mutação; no caso de sair do domínio, a operação é revertida.

Os dois seguintes operadores de mutação atuam na máscara de ativação, como se explica a seguir:

A Mutação *AddWell* atua na máscara de ativação colocando um valor "1" (gene ativado) na posição escolhida. Isto resulta em um aumento no número de genes ativos, seguindo a estratégia de varredura de cromossoma crescente (Zebulum 1999). Pela realização deste operador, poderão ser criados novos poços nas alternativas durante a evolução.

A Mutação *FlipWell* também atua na máscara do cromossomo, porém no sentido de ativar ou desativar genes, isto é, implementa a estratégia de varredura de cromossomo oscilante (Zebulum, 1999). Neste caso, podem acrescentar novos poços ou retirar poços existentes.

No Cruzamento Aritmético, o par de cromossomos escolhidos têm todos os seus genes alterados da seguinte maneira:

$$\begin{aligned} x'_1 &= U(i)x_1 + (1 - U(i))x_2 \\ x'_2 &= U(i)x_2 + (1 - U(i))x_1 \end{aligned} \quad (21)$$

onde  $U(i)$  é uma variável aleatória com distribuição uniforme [0,1). Este operador garante que os novos valores  $x'_1$  e  $x'_2$  continuem pertencendo ao domínio. Isto é válido sempre que o espaço de busca seja convexo.

O Cruzamento Simples funciona da mesma forma que o Cruzamento Aritmético, com a diferença que apenas uma parcela do cromossomo sofrerá alteração. Logo, no Cruzamento Simples é escolhido um ponto de corte e, a partir desse ponto até o final do cromossomo, os genes são alterados conforme a equação (21).

### **4.3.3. Restrições na geração de soluções**

As restrições consideradas no algoritmo genético estão mais relacionadas a informações dadas pelo especialista, tais como:

- Considerar uma distância mínima entre poços, em caso de haver poços com menor distância da permitida, preferir os poços horizontais e produtores. Esta restrição contorna a possibilidade de haver superposição de poços;
- Considerar um valor de comprimento máximo da trajetória do poço horizontal;
- Poços horizontais não podem extrapolar o campo petrolífero;

### **4.4. Aproveitamento de Informações dos Mapas de Reservatório**

Os mapas de qualidade (Cruz et al., 1999; Cruz, 2000) fornecem informações importantes do reservatório, como o potencial de produção de óleo existente em cada setor da reserva petrolífera. Em locais com alto potencial de óleo, é mais adequado colocar poços produtores e, em locais com potencial baixo ou mesmo nulo, não é recomendado colocar poços produtores.

Aproveitando a metodologia utilizada para obter os mapas de qualidade, podem ser obtidos outros mapas, como mapas de aquíferos ou mapas de produção de gás. Ao realizar as simulações com um único poço, o simulador fornece informações de óleo, gás e água, dentre as quais, apenas a informação de óleo é considerada. Se as informações de água forem mapeadas obter-se-ia um mapa do “potencial de água”, denominado neste trabalho mapa de aquífero.

Um mapa de aquífero pode fornecer uma idéia de onde poderiam ser colocados poços injetores se usados sistemas de injeção de água. Neste trabalho é aproveitado tanto o mapa de qualidade quanto o mapa de aquífero durante a inicialização da primeira população do algoritmo genético, da seguinte forma:

1. A partir dos mapas de qualidade e aquífero gerados, são obtidas versões normalizadas  $[0,1]$  destes mapas, dividindo o valor de qualidade de cada posição 2D pelo valor de qualidade máxima da reserva e fazendo o mesmo para o mapa de aquífero;
2. Durante a inicialização da primeira população é feito um sorteio para cada cromossoma com probabilidade 0,1. Os cromossomas que forem sorteados serão inicializados empregando o critério dos mapas, os não sorteados, serão inicializados aleatoriamente;
3. Os cromossomas sorteados serão inicializados da seguinte maneira:
  - a. O conteúdo dos genes (posições  $i, j, k$ , trajetória e direção) é preenchido de forma aleatória;
  - b. Para o poço resultante encontra-se o valor normalizado de qualidade se for produtor. Se for poço injetor, encontra-se o valor normalizado de mapa de aquífero;
  - c. Aplica-se o valor encontrado como taxa de sorteio para decidir se o poço estará ativo ou não. Desta forma, poços produtores que venham recair em setores de alta qualidade, terão muita chance de serem ativados, ocorrendo o contrário se recaírem em setores de pouca qualidade. Ocorrendo o mesmo para os poços injetores.

A Figura 22, ilustra um mapa de qualidade normalizado em vista 3D e a Figura 23, ilustra um mapas de aquífero, também normalizado em vista 3D indicando as áreas com maior possibilidade de ativar poços produtores e injetores respectivamente.

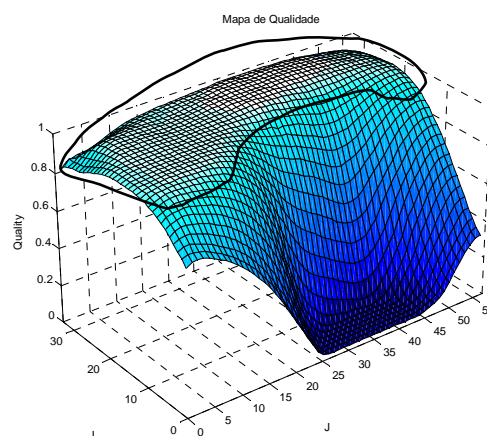


Figura 22. Mapa de qualidade normalizado e áreas de ativação de produtores

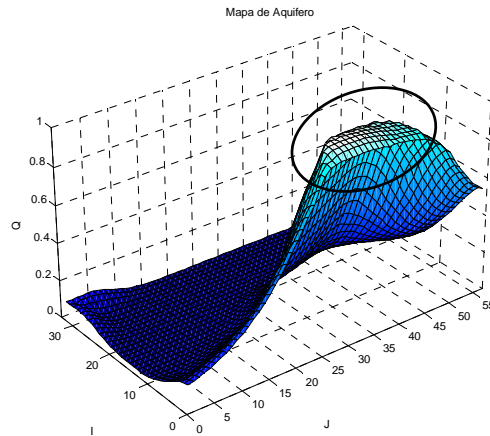


Figura 23. Mapa de aquífero normalizado e áreas de ativação de injetores

#### 4.5. Inferência de Curva de Produção

Este é um dos blocos da função de avaliação, que envolve o uso do simulador de reservatório e o uso de aproximadores de função de produção. O objetivo deste módulo é determinar as curvas de produção de óleo, gás e água para a configuração de poços dada pelo cromossoma do algoritmo genético. Os elementos principais são: o simulador de reservatórios e os dois modelos de aproximação de curva de produção desenvolvidos neste trabalho, detalhados em seção posterior. A Figura 24 a seguir, mostra um diagrama contendo o sistema de inferência.

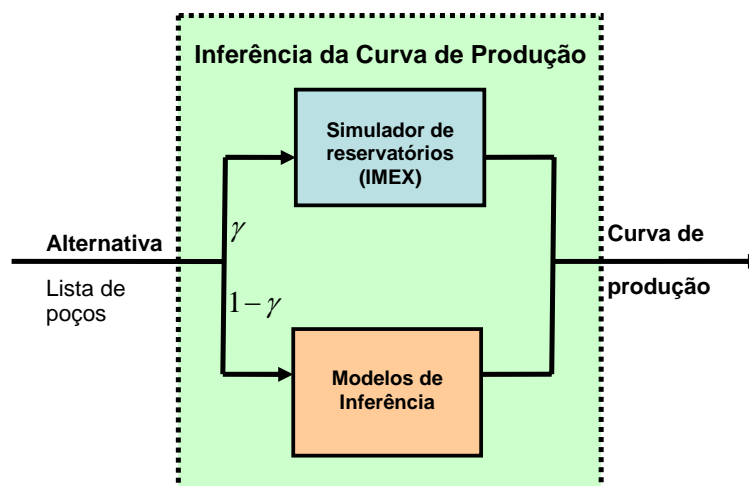


Figura 24. Bloco de inferência de curva de produção



No bloco de inferência estão contidos tanto o simulador de reservatórios e o bloco com os modelos de aproximação de curvas de produção.

Neste trabalho, o uso destes dois módulos é determinado através de uma **taxa de uso de simulador**  $\gamma$ . A decisão dentre usar o simulador de reservatórios ou usar os modelos de inferência é tomada após um sorteio que consiste em gerar um número aleatório uniforme  $u_i \in [0,1 >$  e verificar a seguinte expressão:

$$\begin{aligned} \text{se } u_i \leq \gamma &\rightarrow \text{usa o simulador} \\ \text{se } u_i > \gamma &\rightarrow \text{usa a inferência} \end{aligned} \quad (22)$$

A finalidade da utilização desses dois módulos complementares é a redução do tempo computacional necessário para a obtenção da avaliação de uma certa população.

Nas seções seguintes são detalhados os dois sub-módulos.

#### **4.5.1. Simulador de reservatórios**

Para se obter o valor de avaliação (VPL) de um cromossoma dado, primeiramente é necessário ter as curvas de produção de óleo, gás e água da configuração de poços do cromossoma. Como visto na seção (2.2.1), o simulador de reservatórios é o processo computacional que permite obter essas informações. No trabalho proposto é utilizado o Simulador IMEX da CMG, sobre o qual, podem ser vistos detalhes no apêndice A.

#### **4.5.2. Modelos de Aproximação.**

Uma desvantagem do uso do simulador é o tempo requerido para obter respostas sobre a produção de óleo, gás e água. Este é o motivo principal de se propor o uso de aproximações da curva de produção ao invés do uso intensivo do simulador.

Para isto, é necessário um procedimento de aprendizado a partir das informações de produção fornecidas pelo próprio simulador. Como foi visto, o simulador fornece uma curva de óleo acumulado gerada a partir da configuração da alternativa. Esta curva é composta através de uma série de pontos  $(t_i, Cummm(t_i))$ , onde  $t_i$  é o tempo de simulação e  $Cumm(t_i)$  é o valor de óleo

acumulado para o tempo  $t_i$ . A curva dada pelo simulador pode variar muito no número total de pontos, podendo ir desde 25 pontos até mais de 200 pontos.

#### 4.5.2.1. Pontos da curva de óleo acumulado

Neste modelo, o sistema de inferência de produção de óleo deve fornecer alguns pontos fixos da curva de óleo acumulado; isto é, os valores  $(t_i, Cumm(t_i))$  para alguns valores de  $t_i$ . Dado que a produção simulada para este trabalho é de normalmente 20 anos ou 7200 dias, os valores  $t_i$  escolhidos são: 90, 180, 360, 720, 1800, 3600, 5400 e 7200, como se mostra na Figura 25 a seguir.

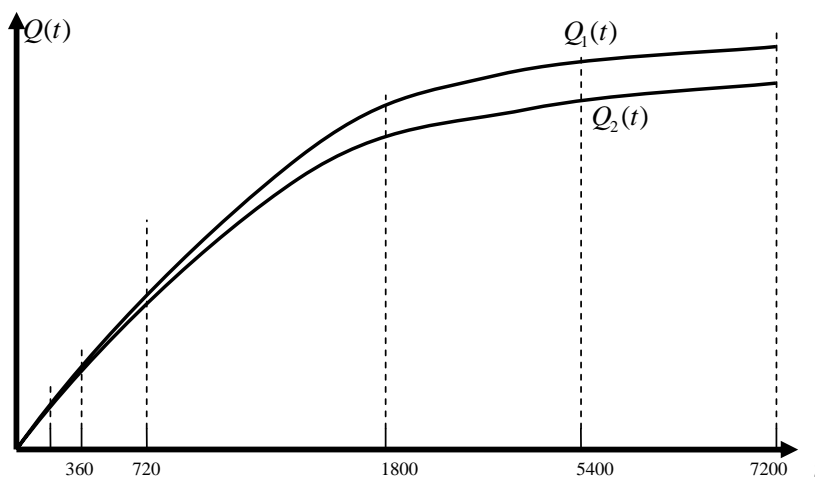


Figura 25. Curvas de produção acumulada e pontos escolhidos

Estes pontos foram escolhidos dessa forma devido à característica exponencial e monotônica das curvas de produção acumulada de óleo, onde normalmente existem grandes diferenças nos tempos iniciais e, à medida que se aproximam dos 20 anos, os valores variam muito menos. Este tipo de amostragem permite mapear curvas de produção de óleo sem cronograma de poços.

#### 4.5.2.2. Modelo de curva segmentada

Para realizar as análises de fluxo de caixa no módulo de cálculo do VPL, é mais útil ter-se o valor de produção de óleo diária ou anual, que equivale à obtenção da primeira derivada com respeito ao tempo da curva de óleo

acumulado. O programa de geração de relatórios do IMEX, *results reports*, pode retornar a curva *oil ratio*, que representa a produção de óleo diária, sempre que solicitada nos *scripts* que o *results reports* emprega.

A partir do modelo de curva de produção a partir de segmentos lineares e exponenciais de Ekern (1985), empregou-se uma parametrização que foi estendida para configurações com cronograma de poços, como mostrado na Figura 26, a seguir:

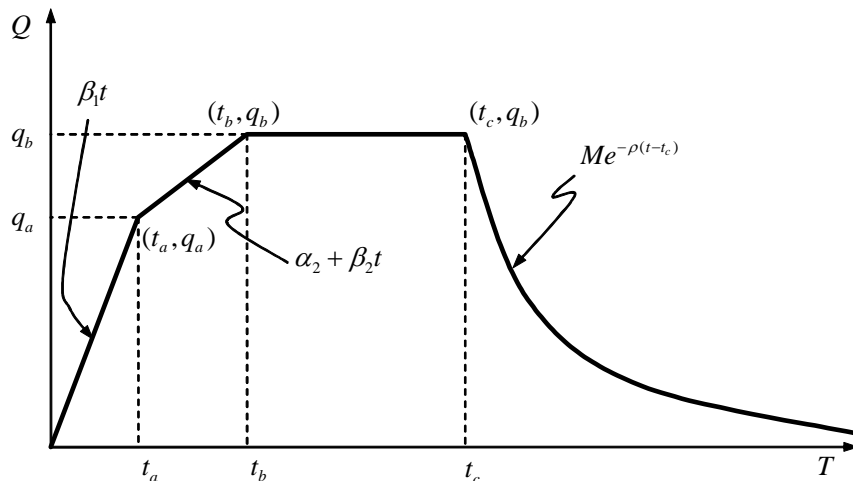


Figura 26. Modelo de curva segmentada da curva de produção.

O eixo das abscissas refere-se ao tempo na exploração (diário, mensal ou anual) e o eixo das ordenadas contém os valores de produção para cada tempo (*oil rate*). Esta curva é dividida em 4 segmentos, cada um com as suas próprias características:

*1º segmento*: linear para o início da produção → entrada paulatina dos poços de produção;

*2º segmento*: parte linear com menor inclinação → estabelecimento da produção;

*3º segmento*: parte linear de inclinação zero (platô) → os poços produtores extraem óleo ao máximo da sua vazão permitida;

*4º segmento*: apresenta um amortecimento exponencial a partir da máxima produção → originado pelo esgotamento do óleo existente na reserva ou perda de pressão no reservatório.

A função de produção que esta curva representa pode ser resumida pela função composta da equação (23) a seguir:

$$q(t) = \begin{cases} 0 < t \leq t_a \rightarrow f(t) = \alpha_1 + \beta_1 t \\ t_a < t \leq t_b \rightarrow f(t) = \alpha_2 + \beta_2 t \\ t_b < t \leq t_c \rightarrow f(t) = q_b \\ t_c < t \leq T \rightarrow f(t) = q_b e^{-\rho(t-t_c)} \end{cases} \quad (23)$$

As variações desta curva dependem dos parâmetros descritos na Tabela 1, a seguir;

Parâmetro	Descrição
$\alpha_1$	Elevação da 1ª reta
$\beta_1$	Inclinação da 1ª reta
$\alpha_2$	Elevação da 2ª reta
$\beta_2$	Inclinação da 2ª reta
$q_M$	Valor de máxima vazão
$\rho$	Taxa de amortecimento exponencial
$t_a$	Ponto de início da 2ª reta
$t_b$	Ponto de início do platô
$t_c$	Ponto de início do amortecimento exponencial

Tabela 1. Descrição dos parâmetros do modelo de curva segmentada

De acordo com o formato da curva, pode-se inferir a existência de relações de dependência entre alguns parâmetros, expressas no conjunto de equações lineares que se mostra a seguir.

$$\begin{aligned} \alpha_1 &= 0 \\ \alpha_2 &= (\beta_1 - \beta_2)t_a \\ \alpha_2 &= \beta_1 - \left( \frac{q_M - \beta_2}{t_b} \right) t_a \end{aligned} \quad (24)$$

Uma forma mais fácil de se representar a curva é usar diretamente os pontos  $(t_a, q_a)$ ,  $(t_b, q_b)$ ,  $(t_c, q_c)$ , mais o valor  $\rho$  de amortecimento exponencial, como parâmetros a serem estimados. Como  $q_b = q_c = q_M$ , dos 9 parâmetros mostrados na Tabela 1, apenas 6 são necessários para otimizar a curva. Estes 6 parâmetros possuem restrições lineares nos domínios, estabelecidas pelas seguintes desigualdades:

$$t_a < t_b < t_c \quad (25)$$

$$q_a \leq q_b \leq q_M \quad (26)$$

$$\rho > 0 \quad (27)$$

Finalmente, a curva segmentada pode ser representada da forma da equação (28) como uma função dos 6 parâmetros:

$$q(t) = F(t_a, t_b, t_c, q_a, q_b, \rho) \quad (28)$$

Para encontrar os valores que melhor aproximem uma dada curva, propõem-se duas abordagens:

1. Utilizar algoritmos genéticos para encontrar o conjunto de parâmetros que minimizem o erro quadrático médio entre o modelo e a curva dada pelo simulador;
2. Encontrar uma expressão de estimação destes parâmetros a partir de estimadores de mínimos quadrados.

#### 4.5.3. Aproximação com Algoritmos Genéticos

Para realizar esta aproximação é utilizado um algoritmo genético tradicional (seleção, cruzamento, mutação) com a seguinte configuração:

##### Representação do cromossoma:

O cromossoma contém os 6 parâmetros a serem otimizados:  $t_a$ ,  $t_b$ ,  $t_c$ ,  $q_a$ ,  $q_b$ , e  $\rho$  da função dada pela equação (28). Na Figura 27 apresenta-se a estrutura do cromossoma.

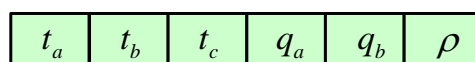


Figura 27. Cromossoma para a aproximação por AG.

##### Operadores genéticos:

Os seguintes operadores foram utilizados nesta modelagem:

- Cruzamentos aritmético, simples e geométrico, (Michalewicz, 1996);
- Mutações uniforme, não uniforme, de fronteira e gaussiana (Michalewicz, 1996).

##### Avaliação:

A avaliação é feita determinando-se o erro MSE (*Mean Squared Error*) existente entre os pontos  $q_i$  da curva modelada e os pontos  $q_i^{IMEX}$  fornecidos

pelo simulador. Os valores de tempo  $t_i$  consideram-se os mesmos que o simulador estabeleceu durante o processo de simulação e colocados na curva original; logo, os pontos  $q_i$  são calculados aplicando-se a equação (23) para os  $t_i$  do simulador. A equação (29) mostra como é calculado o erro MSE ajustado

$$score = \frac{1}{n} \sum_{i=1}^n (q_i - q_i^{IMEX})^2 e^{-\lambda t_i} \quad (29)$$

O amortecimento exponencial  $e^{-\lambda t_i}$  que aparece na expressão foi colocado com o intuito de dar maior ênfase aos pontos iniciais da curva. Isto se deve ao fato de que ao se calcular o VPL, também é dada maior importância aos pontos iniciais de produção de óleo (pontos menos deslocados no tempo).

Na avaliação é usado o erro MSE dada a característica deste de dar maior penalização a respostas com valores espúrios fazendo com que o algoritmo genético tente evitá-los na resposta otimizada.

### Erro MAPE

A aproximação feita pelo algoritmo genético emprega como métrica de otimização o erro MSE ponderado pela distância temporal da equação (29) como visto no parágrafo anterior, contudo durante a aproximação feita com modelos neurais e modelos *neuro-fuzzy* hierárquicos, o erro empregado é o MAPE (*mean absolute percentage error*), cuja expressão é mostrada na equação (30) a seguir.

$$MAPE = \frac{\sum_{i=1}^N \left| \frac{q_i - \tilde{q}_i}{q_i} \right|}{N} \quad (30)$$

onde

$q_i$  valor do ponto no tempo  $t_i$

$\tilde{q}_i$  valor aproximado do ponto no tempo  $t_i$

#### 4.5.4.

### Aprendizado com os modelos Neurais e *Neuro-Fuzzy* Hierárquico

Nesta seção mostra-se como foram desenvolvidos os modelos de aproximação das curvas de produção baseados em redes neurais e no modelo híbrido *neuro-fuzzy* hierárquico.

Cabe ressaltar que a localização e número de poços é um problema árduo, dado o número de combinações possíveis de poços na malha do campo. Supondo uma malha 30x30x1 que é relativamente simples, existem 900

possíveis posições de poços verticais; o número de combinações para 10 poços é:

$$C_n^r = C_{900}^{10} = \frac{900!}{10!(900-10)!} = 9.13841 \times 10^{22} \tag{31}$$

Isso sem considerar a possibilidade que estes poços podem ser injetores ou produtores, que aumentaria ainda mais o número de combinações possíveis.

Algumas restrições técnicas, como a distância mínima entre poços perfurados e o fato de não ter mais poços injetores do que produtores, podem ser consideradas, permitindo diminuir o número de combinações válidas.

Foram avaliados os dois modelos de aproximação de curva desenvolvidos neste trabalho: o primeiro para aproximar uma curva de óleo acumulado com 8 pontos (seção 4.5.2.1) e o segundo para uma curva segmentada de razão de produção visto na seção (4.5.2.2).

Para qualquer um dos modelos, primeiramente é necessário obter um conjunto de padrões de entrada e saída para efetuar o respectivo treinamento supervisionado, como é descrito a seguir.

**4.5.4.1. Padrões para curva de óleo acumulado**

No modelo de padrões de entrada e saída utilizado para o aprendizado de pontos de curva de óleo acumulado, cada poço é definido por um par de dados (i, j). Este modelo de aprendizado não considera os valores k de posição de cada poço, portanto está restrito para padrões que contenham apenas poços verticais. Como resultado desta modelagem, os modelos de aprendizado a serem usados devem conter 2n<sub>p</sub> entradas e 8 saídas, onde n<sub>p</sub> é o número de poços sendo considerado na aproximação. Este modelo de padrões é apresentado na Figura 28, a seguir.

Poços Produtores										Poços Injetores										Saídas							
i	j	i	j	i	j	i	j	i	j	i	j	i	j	i	j	i	j	i	j	SAIDAS							
1	3	6	15	3	1	1	15	15	1	15	15	1	30	1	1	30	1	27									
5	4	1	27	12	30	25	12	8	17	29	9	3	10	20	14	11	22	16	5								
1	1	5	6	23	7	3	4	2	9	7	12	4	15	13	8	21	3	23	6								

Figura 28. Formato dos padrões de entrada e saída: pontos de curva.

#### 4.5.4.2. Padrões para a curva segmentada

Para o modelo de curva segmentada (seção 4.5.2.2), que permite modelar alternativas com cronograma de poços, os padrões são um pouco diferentes, como mostra a figura a seguir.

Poços										Saídas					
i1	j1	i2	j2	i3	j3	i4	j4	i5	j5	ta	tb	tc	qa	qb	$\rho$
15	15	30	30	1	1	30	1	1	30	2	400	1300	12300	13000	0.12
30	1	30	30	1	30	15	15	1	1	12	200	1540	11000	14000	0.32

Figura 29. Formato dos padrões de entrada e saída: curva segmentada.

As entradas são, similares que o caso anterior, as locações dos poços  $(i, j)$ , sendo que nesta codificação, o sinal dos valores  $(i, j)$  determina se o poço é produtor (no caso positivo) ou injetor (no caso negativo). Já as saídas representam os pontos considerados no modelo de curva segmentada. Neste caso, o poço  $(i_1, j_1)$  é o primeiro a abrir no cronograma de poços, o poço  $(i_2, j_2)$  o segundo, e assim sucessivamente. Cabe ressaltar que a diferença de tempo de abertura é considerada como 60 dias.

Foram empregados três diferentes modelos de redes neurais e um modelo híbrido *Neuro-Fuzzy*, descritos a seguir:

- Redes neurais *feed-forward* com aprendizado *back-propagation*, taxa de aprendizado adaptativa, e termo de momento;
- Redes recorrentes de Elman;
- Redes *Radial Basis Functions*;
- Modelo *Neuro-Fuzzy* Hierárquico com particionamento Binário (NFHB).

Os resultados obtidos a partir destes modelos são descritos no capítulo 5.

#### 4.6. Cálculo do Valor Presente Líquido (VPL)

Neste módulo do sistema encontra-se o modelo econômico que permite calcular o valor presente líquido da alternativa.



#### 4.6.1. Modelagem Econômica

O Valor Presente Líquido (VPL) é a diferença entre o valor presente dos ganhos (VP) menos o valor presente das despesas (VD), como na equação (2) da seção (2.3.2.3) que é apresentada novamente a seguir.

$$VPL = VP - VD \quad (32)$$

No caso de desenvolvimento de campos petrolíferos, os ganhos VP provêm da receita dada pela produção de óleo e gás menos os custos de operação, custos fixos e variáveis; já as despesas VD são obtidas levando ao valor presente os custos de desenvolvimento  $D$ , isto é seguindo o modelo clássico utilizado na área de finanças corporativas (Brealey, 1981), onde o VPL vem a ser o valor esperado dos fluxos de caixa dados durante o tempo de produção.

A seguir são mostrados os detalhes do modelo de fluxo de caixa utilizado neste trabalho para cálculo do valor presente em alternativas de exploração petrolífera.

Para o custo de desenvolvimento  $D$ , são considerados todos os investimentos feitos para que o reservatório possa começar a produzir óleo, os quais incluem os poços a serem perfurados, as linhas de condução e a plataforma, conforme a expressão seguinte:

$$D = (a * f_{ph} + r)n_p + b + \sum_{j=1}^{n_p} |p_j - P_{PL}|_c \quad (33)$$

onde os parâmetros, conhecidos como CAPEX (*Capital Expenditure*), estão descritos na Tabela 2, e o valor do fator de custo por poços horizontais  $f_{ph}$  é calculado com a seguinte expressão

$$f_{ph} = \left( 1 + \frac{n_{ph}}{n_p} f_h \right) \quad (34)$$

onde  $n_{ph}$  é o número de poços horizontais da alternativa e o fator  $f_h$  é o fator de custo adicional para poços horizontais (Bittencourt & Horne, 1997) mostrado na Tabela 2 a seguir.

Parâmetro	Descrição
$a$	Custo médio de perfuração para cada poço mais o custo da Árvore de Natal <sup>3</sup>
$r$	Custo do riser para uma linha d'água de 1 quilômetro
$b$	Custo de transferência mais custo da plataforma e planta.
$p_j$	Posição do poço $j$ .
$P_{PL}$	Posição da plataforma.
$c$	Custo de linha / quilômetro.
$n_p$	Número de poços da alternativa.
$f_h$	Fator de custo de poços horizontais

Tabela 2. Descrição dos parâmetros CAPEX.

Usualmente, para uma dada alternativa, os custos de investimento da equação (33) são conhecidos no momento zero. Cabe dizer então que o valor presente do desenvolvimento  $VD$  neste caso é o próprio valor do desenvolvimento  $D$  sem amortecimento exponencial pois ocorre no tempo  $t_0$ .

Para o problema de reservatório, o valor presente dos ganhos é composto pela diferença entre as receitas da produção e os custos de operação que acontecerão durante o período de produção especificado. Assim, o valor presente dos ganhos é o valor presente da receita  $VP_R$  menos o valor presente do custo de operação  $VP_{COP}$  aplicando-se a alíquota de impostos e encargos sociais  $I$  de aproximadamente 34%, como se mostra na equação (35):

$$VP = (VP_R - VP_{COP})(1 - I) \quad (35)$$

O valor da receita depende da produção de óleo  $Q(t)$  mais a produção *equivalente-barril* de gás ( $G(t)/1000$ ) para os tempos  $t_i$  multiplicados pelos preços do petróleo  $P_{oil}(t)$  a cada momento  $t_i$ . Neste caso, como as condições de mercado são de certeza, o preço do petróleo  $P_{oil}(t)$  é uma função constante com valor ajustado de US\$129.8 por cada  $m^3$  de óleo, o que equivale a US\$20.00 por barril. Assim, para cada tempo  $t$ , o valor da receita é obtido como:

$$R(t) = (Q(t) + G(t)/1000)P_{oil}(t) \quad (36)$$

Para obter o valor presente, aplica-se a taxa de desconto  $\rho$  como visto na equação (1). A expressão resultante é a que a equação (37) mostra

<sup>3</sup> Chama-se Árvore de Natal ao conjunto de válvulas localizado na cabeça do poço que serve para controlar o fluxo de óleo. No caso de produção *off-shore*, usam-se sempre as árvores de natal molhadas e deve ser levado em consideração seu alto custo (de 1 milhão a 2.5 milhões a unidade).

$$VP_R = \sum_{i=1}^T R(t_i).e^{-\rho.t_i} \quad (37)$$

onde  $T$  é o tempo máximo de atividade lucrativa do reservatório, determinado pelo último tempo  $t_i$  em que o valor da receita  $R(t_i)$  resulte maior que o valor do custo operacional. O valor  $t_i$  é o  $i$ -ésimo passo de tempo no valor de produção de óleo ou gás. Cabe ressaltar que  $t_i$  avança em passos de tempo que dependem das respostas obtidas pelo simulador de reservatório. Estes passos normalmente não são constantes.

O valor do custo operacional é dado pelo somatório dos custos de manutenção dos poços, custos fixos, custos variáveis dados pela produção, *royalties* da receita para cada tempo  $t_i$  e custo de retirada da água, como na equação (38) a seguir.

$$C_{OP}(t_i) = (m.n_p) \frac{t_i}{365} + C_f \frac{t_i}{365} + C_v.Q(t_i) + R_y.R(t_i) + C_w.W(t_i) \quad (38)$$

onde todos os parâmetros, conhecidos como OPEX (*Operation Expenditure*), são descritos na Tabela 3 a seguir.

Parâmetro	Descrição
$t_i$	Tempo de simulação $i$ (em dias)
$m$	Custo de manutenção anual de um poço
$n_p$	Número de poços
$C_v$	Custos variáveis que dependem da produção
$C_f$	Custos fixos anuais
$C_w$	Custo de retirada de água por cada $m^3$
$Q(t_i)$	Produção de óleo no tempo $t_i$
$W(t_i)$	Produção de água no tempo $t_i$
$R(t_i)$	Receita no tempo $t_i$

Tabela 3. Descrição dos parâmetros OPEX.

O valor presente para um custo é dado aplicando-se novamente a equação (1), resultando na seguinte expressão:

$$VP_{COP} = \sum_{i=1} C_{OP}(t_i).e^{-\rho.t_i} \quad (39)$$

Com a determinação dos valores  $VP_R$  e  $VP_{COP}$  obtém-se o valor presente  $VP$  da equação (35) e, conseqüentemente, o valor presente líquido  $VPL$  da equação (2).

Na tabela a seguir resumem-se os valores relacionados ao cenário econômico empregados para o cálculo do VPL.

Parâmetro	Descrição
$I$	Alíquota de imposto e encargos sociais
$\rho$	Taxa de desconto
$R_y$	Royalties

Tabela 4. Parâmetros do cenário econômico.

#### 4.7. Implementação do Sistema

Conforme descrito na seção 4.2, o sistema de três blocos proposto nesta tese funciona da seguinte forma: no módulo otimizador, o algoritmo genético gera um indivíduo que representa uma alternativa; em seguida, esta alternativa é submetida ao bloco de inferência de curva de produção, que fornece a curva de produção de óleo, gás e água. A curva gerada é então repassada para o bloco de cálculo do Valor Presente Líquido (VPL) junto com o arranjo dos poços, obtendo-se o valor do VPL da alternativa (o indivíduo gerado inicialmente). Para fechar a malha, o VPL calculado é inserido no otimizador como avaliação da alternativa do algoritmo genético. A Figura 30 mostra, em mais detalhes, a modelagem do sistema de otimização proposto.

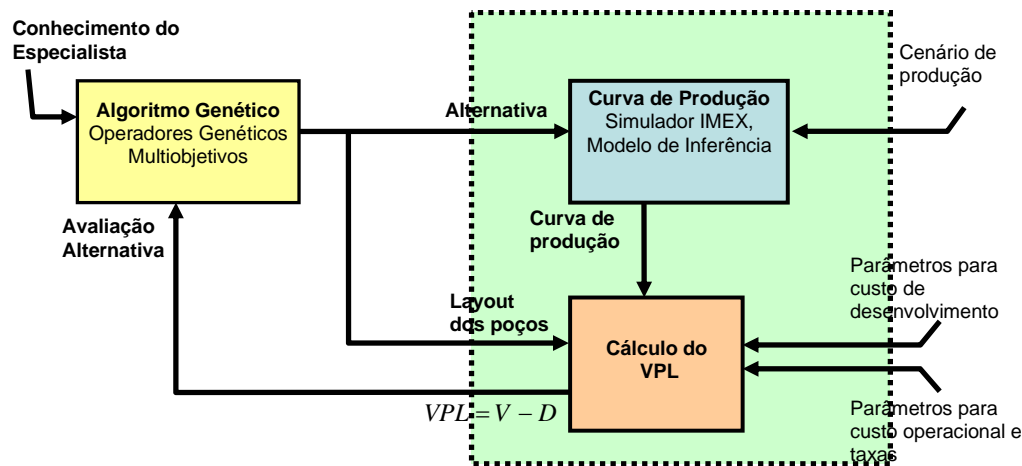


Figura 30. Modelagem do Sistema de Otimização proposto.

A Figura 30 ilustra a possibilidade de se inserir conhecimento do especialista no módulo otimizador, como por exemplo uma semente inicial no algoritmo genético ou informação sobre o campo petrolífero.

São ressaltados em verde e com quadro pontilhado os blocos de curva de produção e de cálculo do VPL, que em conjunto conformam a função objetivo do

otimizador. Analisando-se a arquitetura *master-slave* descrita na seção (3.4.7.1.1) para distribuição de AGs em ambientes paralelos ao nível de avaliação, fica claro que o modelo proposto pode se inserir nesta arquitetura destacando e replicando o bloco inteiro de cálculo da função objetivo em cada um dos processadores *slave*.

Para implementar o sistema proposto foi utilizada a linguagem de programação C++ (Stroustrup, 1999) empregando a biblioteca padrão *Standard Template Library* STL (Josuttis, 1999) e a técnica de *multithreading* (Eggers et al., 1997) para facilitar o funcionamento dos processos distribuídos. Os processos e funções envolvidos estão dispostos em 3 módulos básicos como se mostra na figura a seguir.

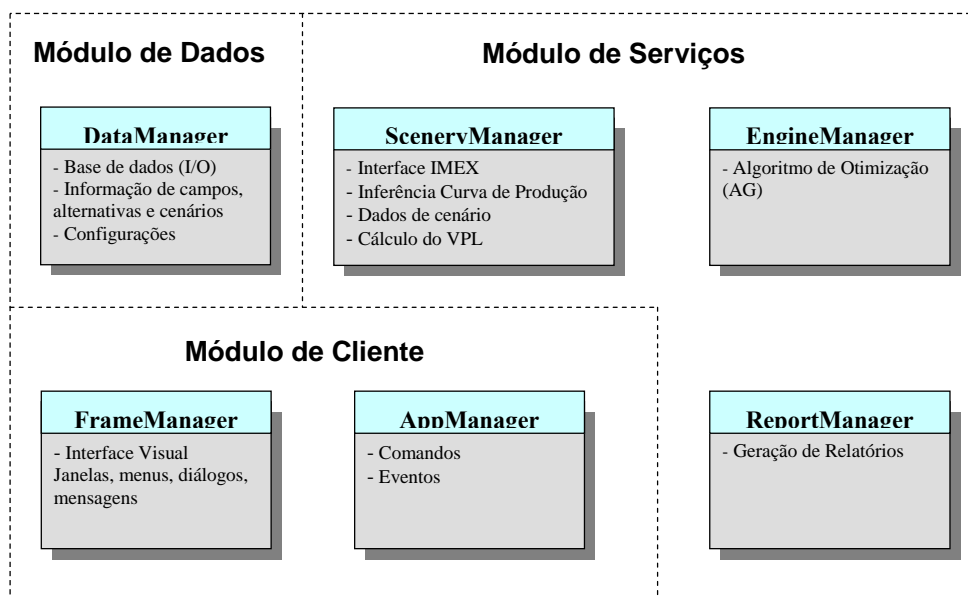


Figura 31. Módulos de Implementação do Sistema Otimizador

Como mostra a Figura 31, são três os módulos que formam o sistema:

- módulo de dados;
- módulo de serviços;
- módulo de cliente.

Nos parágrafos a seguir, são detalhados estes módulos e as classes que foram implementadas dentro deles.

#### 4.7.1. Módulo de Dados

Contém a instância da classe *DataManager* onde estão situadas rotinas e estruturas para:

- leitura e escrita da base de dados;

- estrutura de dados com informação dos campos, alternativas existentes, mapas de qualidade e aquífero;
- dados do cenário usado na avaliação;
- dados de configuração do sistema.

#### 4.7.2. Módulo de Serviços

Organizado em três blocos:

- *SceneryManager* ⇒ contém o cenário de avaliação, rotinas de interface com o simulador IMEX, rotinas de inferência de curva de produção, e cálculo do VPL;
- *EngineManager* ⇒ contém o algoritmo de otimização (algoritmo genético) com os cromossomas, população, inicialização, métodos de seleção e operadores genéticos;
- *ReportManager* ⇒ controla a visualização das informações relacionadas à evolução do AG e à geração do relatório final ao término da otimização.

Cabe ressaltar que dentro do módulo de serviços existe a instanciação múltipla no ambiente paralelo que será vista na seção 4.8.

#### 4.7.3. Módulo de Cliente

Contém os blocos *FrameManager*, que vem a ser a interface visual do sistema (janelas, menus, diálogos, mensagens); e o bloco *AppManager*, que controla as requisições do usuário, efetuadas a partir da interface visual, e as respostas do sistema a serem mostradas visualmente.

#### 4.8. Uso de Ambientes Paralelos

Como foi mostrado na seção (4.7), a avaliação pode ser separada do resto do algoritmo de otimização. Na Figura 31 foi visto que dentro do módulo de serviços existe a instância da classe *SceneryManager* que calcula a função objetivo do algoritmo genético. Portanto, esta instância interna deve ser separada do módulo de avaliação formando o denominado *Módulo de Avaliação* que pode ser instanciado de forma múltipla em diferentes processadores do

ambiente paralelo segundo a arquitetura *master-slave* da seção (3.4.7.1.1). Na Figura 32 se mostra o deslocamento e replicação do Módulo de Avaliação.

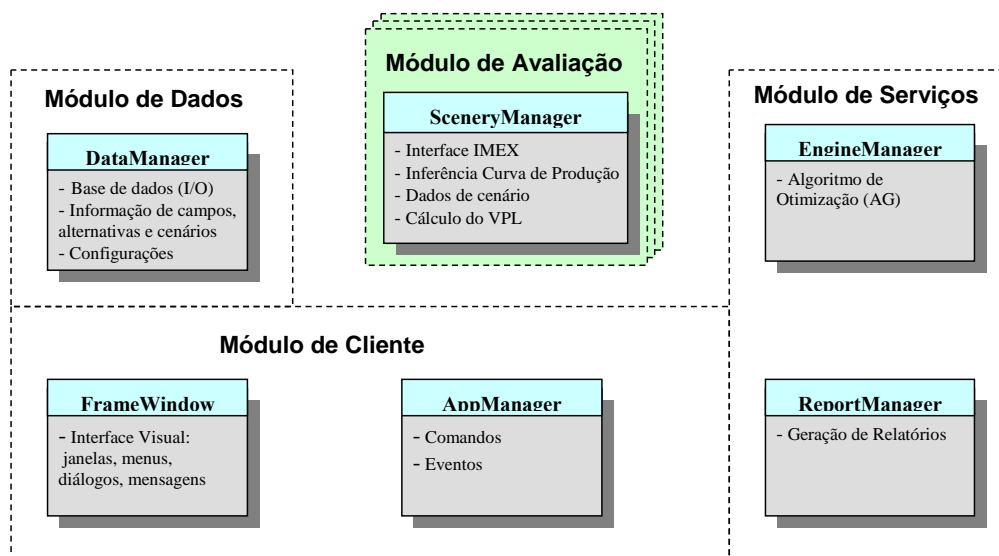


Figura 32. Módulo destacado para distribuir no ambiente paralelo

Na Figura 32 observa-se que o sistema fica dividido em duas partes: *Aplicação Master* que contém os módulos de cliente, dados e serviços sendo que, neste último módulo está instanciado o algoritmo otimizador AG; e o *Módulo de Avaliação* replicado em cada um dos processadores *slave*. Dada a instanciação distribuída e remota dos módulos de avaliação, são necessárias duas interfaces de comunicação entre cada um deles e os módulos instanciados na aplicação *Master*; estas interfaces de comunicação são implementadas utilizando CORBA: *Common Object Request Broker Architecture* (OMG, 1997), uma arquitetura que fornece modelos e interfaces para programação de aplicações orientadas a objeto portáteis e distribuídas. A compatibilidade com a linguagem de programação C++ é completa (Henning, 1999). A seguir detalha-se o funcionamento de cada uma das interfaces definidas na arquitetura CORBA e seus serviços implementados.

#### 4.8.1. Interface – Cenário

Através desta interface, os módulos de avaliação distribuídos recebem desde a o módulo de dados da aplicação *master* as informações necessárias para montar o cenário requerido na realização das avaliações. Existem os seguintes serviços implementados:

**Serviço de envio de cenário** → Através deste serviço são enviadas as seguintes informações:

- *Informações do campo petrolífero a ser utilizado:* arquivos .DAT e .INC para o simulador, parâmetros e restrições na geração dos poços e restrições a serem levadas em conta durante a avaliação;
- *Informações do cenário econômico usado no cálculo do VPL:* preço do petróleo, valores para investimento CAPEX, valores para custo operacional OPEX, alíquotas e taxas;
- *Informações de configuração* relacionadas ao simulador de reservatórios e ao uso das aproximações da curva de produção.

**Serviço de inicialização** → Através deste serviço são enviadas as requisições para montar o cenário prévio à avaliação e montar a interface de avaliação; neste caso há uma resposta sobre o *status* final (se montou corretamente, se houve algum erro, etc.).

**Serviço de Status** → Solicitar as condições atuais do cenário (sem cenário, habilitado, desabilitado, erro).

#### 4.8.2. Interface – Avaliação

Através desta interface é possível realizar a comunicação entre o algoritmo genético instanciado no módulo de serviços da aplicação *master* e cada um dos módulos avaliadores. Para isto foram definidos dois serviços:

**Serviço de Avaliação de cromossoma** → Através deste serviço, o algoritmo genético envia a lista de poços referentes a um dado cromossoma para um módulo avaliador. O módulo avaliador calcula e retorna o valor de VPL junto com outros valores necessários para os relatórios finais (óleo recuperado, custo de investimento, custo operacional, receita). Através deste serviço torna-se possível realizar avaliações simultâneas dependendo do número de processadores *slaves* disponíveis no ambiente paralelo.

**Serviço de Status** → Solicita as condições atuais do módulo avaliador: se inativa, se realizando avaliação, em erro.

Na Figura 33 apresenta-se como estes dois canais permitem a interação entre os módulos envolvidos como se mostra a seguir.



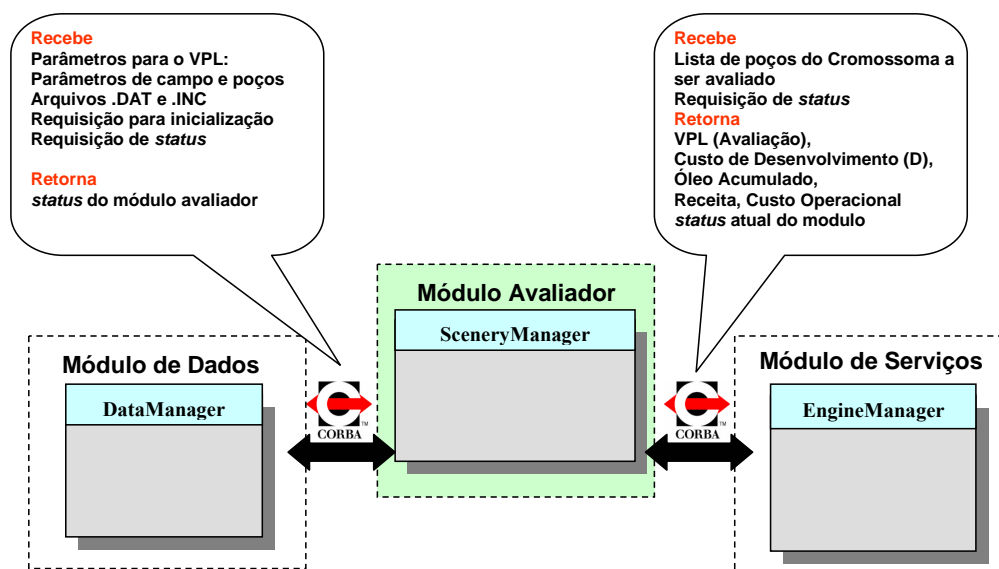


Figura 33. Canais de comunicação usados na avaliação distribuída

#### 4.8.3. Controle de erros de comunicação

Para realizar as avaliações em modo distribuído, o algoritmo genético requer mais controles do que um algoritmo iterativo. Devem ser considerados os diversos casos que podem ocorrer durante o envio do cenário de avaliação e durante a avaliação em si. Neste trabalho os seguintes erros são contemplados:

*i) Durante o envio do cenário do master para os slaves remotos*

- Erro de comunicação com o *host* (falha na rede);
- Não montou o cenário;
- Não inicializou o canal de comunicação para a avaliação;

Para qualquer caso de erro neste nível, o *master* excluirá este *host* da lista de *hosts* habilitados para avaliação.

*ii) Durante a avaliação no master*

- Erro na inicialização da *thread* associada a uma dada simulação remota;
- Erros de comunicação entre o *master* e um *slave* (como uma falha na rede); neste caso é lançada uma exceção nas rotinas do protocolo CORBA;

*iii) Durante a avaliação no slave*

- Falha ao montar o arquivo *include* (disco cheio, falta de permissões no computador remoto);
- O IMEX não recebeu a permissão do servidor de licença para rodar a simulação (*license update error*);

- IMEX finalizou com erro;
- Erro na execução/licença do *Results Report*;
- Não foi achado ou lido o arquivo de saída do *Results Report* (.RWO).

A maioria destes erros permite criar uma exceção no *host* remoto, que finaliza o processo remoto de avaliação, retornando para o processo associado no *master* um código de erro que fica armazenado em uma variável local. Desta forma, o *master* tem como gerenciar de forma *on-line* os processos que rodam nos *slaves*.

Porém, na ocorrência de uma falha na licença do IMEX ou REPORT, o próprio simulador coloca uma mensagem *modal* indicando o tipo de erro que ocorreu durante a atualização da licença CMG, sem liberar o processo se não for apertado o botão OK da mensagem. Para contornar este problema foi utilizado um critério de *time-out* para determinar se ocorreu falha de licença em um *host* remoto. O valor de tempo de expiração para cada *host* foi definido como cinco vezes a média dos tempos de simulação decorridos por cada simulação nesse *host*. Se um *host* ultrapassar este tempo, o *master* colocará um erro de *timeout*, e o cromossoma que lhe foi enviado será retirado da lista de enviados e reenviado para outro *host* disponível. Qualquer resposta de avaliação que, porventura, venha a ser retornada depois da ocorrência do erro será descartada. De fato, o cromossoma será enviado para outro *host* ativo.

Este problema pode acontecer devido às falhas reais na rede ou à necessidade de mais cálculos numéricos para essa configuração de poços; entretanto, deve-se considerar que é pouco provável que a necessidade de mais cálculos ultrapasse em 5 vezes o tempo médio das simulações realizadas anteriormente. Contudo, um *host* que entrou em *timeout* e consegue finalizar sua simulação sem outros erros, voltará novamente a estar ativo e a receber novos cromossomas para avaliar.