

4 Modelagem e Implementação

O processo de ajuste de histórico consiste em otimizar uma grande quantidade de parâmetros. Esses parâmetros representam as propriedades das formações rochosas que constituem o reservatório, estando distribuídas por toda a malha do modelo de simulação. Portanto, para uma dada propriedade, é necessário otimizar tantos parâmetros quanto são os blocos que compõem o modelo de simulação. Dentre as várias propriedades que caracterizam um reservatório, geralmente as de maior interesse para o ajuste de histórico são a permeabilidade, a porosidade ou ainda a transmissibilidade. A determinação de qual delas deve ser ajustada depende das características do reservatório, podendo haver situações em que seja necessário ajustar mais de uma [3].

Diante disto, esse trabalho desenvolveu e implementou um modelo de solução capaz de ajustar duas ou mais propriedades simultaneamente. Esta solução é aplicada em casos onde o ajuste de apenas uma propriedade não resulte no efeito esperado, havendo assim a necessidade de ajustar duas ou mais propriedades simultaneamente.

Também foi desenvolvido um módulo capaz de aproveitar o potencial computacional paralelo dos Algoritmos Genéticos Co-Evolutivos. Este módulo possibilita a distribuição das simulações, proporcionando a capacidade de lidar com grandes quantidades de dados.

Os modelos de solução propostos buscam suprir esta necessidade, de ajuste simultâneo das propriedades, unindo, em um modelo computacional híbrido, os benefícios proporcionados pela Geoestatística de Múltiplos Pontos, através do algoritmo *FILTERSIM*, pela Inteligência Computacional, através dos Algoritmos Genéticos Coevolutivos e pela Computação Paralela, através do Módulo Distribuidor.

4.1 Modelo de Solução I

A Figura 4.1 apresenta o primeiro modelo de solução proposto. Este foi desenvolvido em três módulos principais, os quais foram subdivididos em submódulos para facilitar a implementação, o entendimento e a agregação de novas funcionalidades.

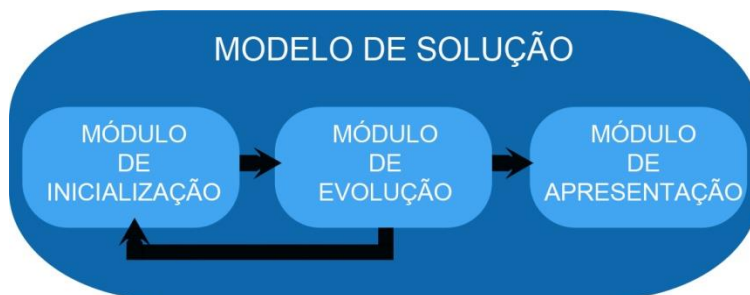


Figura 4.1: Modelo de Solução

O módulo de inicialização é responsável por configurar o algoritmo genético co-evolutivo e gerar as populações iniciais. Após preparar todo o *background*, o módulo de evolução é chamado.

O módulo de evolução é responsável por explorar todo o espaço de busca. Este é guiado pela função objetivo que avalia os resultados retornados pelo simulador de reservatório. No final de cada rodada, é requisitado ao módulo de inicialização que gere uma nova população inicial para cada propriedade a ser evoluída.

Ao final do processo evolutivo, a melhor solução é enviada ao módulo de apresentação, o qual se encarregará de processar os dados necessários para gerar o resultado final da evolução.

Os três módulos principais estão subdivididos da seguinte forma:

4.1.1

Módulo de Inicialização

O módulo de inicialização, Figura 4.2, foi subdividido em três submódulos: módulo de configuração, módulo inicializador e o *FILTERSIM*. Este módulo é responsável por preparar todo o cenário para que a evolução possa ocorrer.

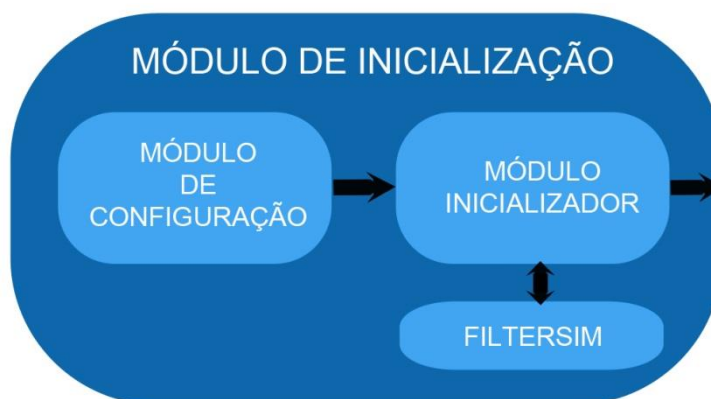


Figura 4.2: Módulo de Inicialização

4.1.1.1

Módulo de Configuração

O módulo de configuração é responsável por obter e armazenar todos os parâmetros informados pelo especialista. Estes parâmetros serão utilizados para configurar toda a solução.

Parâmetros de configuração do algoritmo evolutivo:

- Tamanho da População;
- Número de Gerações;
- Taxa de *Steady State* entre gerações;
- Número de Experimentos (Rodadas);
- Taxa de *Steady State* entre Rodadas;
- Taxa de Cruzamento;
- Taxa de Mutação I;
- Taxa de Mutação II;
- Taxa de Anulação;
- Imagens de Treinamento.

Parâmetros de configuração dos programas:

- Endereço do simulador IMEX;
- Endereço do software SGeMS;
- Endereço do distribuidor;
- Endereço de validação;
- Repositório Windows;
- Repositório Linux;
- Endereço do FTP;
- Tempo máximo por simulação.

4.1.1.2

Módulo Inicializador

O módulo Inicializador é responsável por criar e inicializar, de acordo com os parâmetros recebidos do módulo de configuração, todos os indivíduos da população inicial de cada propriedade do modelo de simulação a ser otimizada. Em um algoritmo genético coevolutivo, assim como na natureza, as espécies são geneticamente isoladas, ou seja, os indivíduos são separados em populações

distintas de acordo com as suas características. Portanto, cada indivíduo só pode se reproduzir com outros indivíduos da mesma espécie.

Nesta arquitetura, cada população representa uma espécie. Deste modo, tem-se uma população de mapas de porosidade e outra de mapas de permeabilidade. Estas duas espécies interagirão cooperativamente segundo as regras, definidas no módulo de evolução, que determinam como estas se ajudam a evoluir. O processo de inicialização das populações é exibido na Figura 4.3.

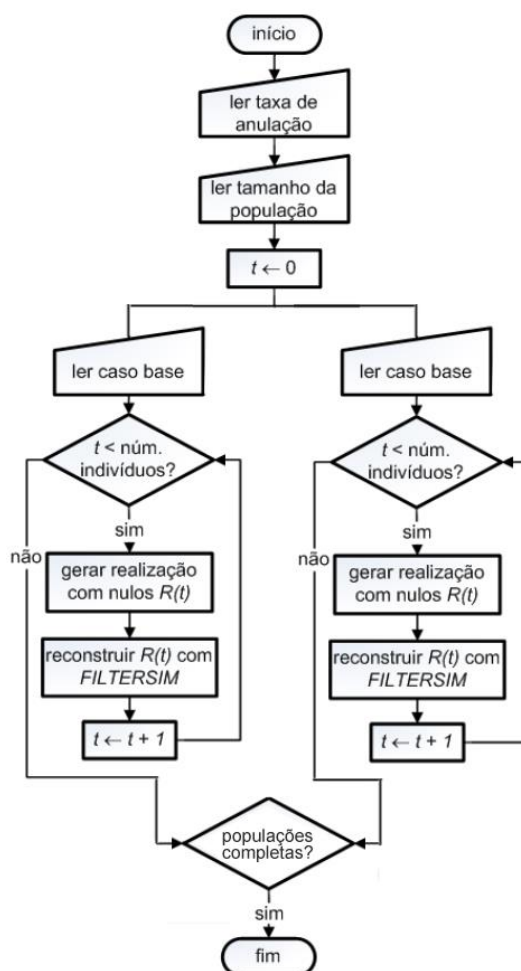


Figura 4.3: Fluxograma de inicialização das populações

O módulo inicializador recebe os parâmetros *taxa de anulação* e *tamanho da população* a partir do módulo de configuração e reserva memória para cada uma das populações a serem inicializadas. Em seguida, o caso base de cada espécie é lido e armazenado em memória. Neste momento inicia-se um ciclo de preenchimento das populações iniciais até que o número de indivíduos, em cada população, seja igual ao parâmetro *tamanho da população*. Para cada indivíduo

é realizada uma cópia do caso base e este é submetido a um processo de anulação e reconstrução de sua propriedade utilizando geoestatística de múltiplos pontos, através do algoritmo *FILTERSIM*.

O processo de anulação consiste em eliminar o valor da propriedade de interesse em determinados blocos, ou seja, anular o gene do cromossoma. O critério para seleção dos genes a serem anulados é um sorteio de um número aleatório para cada gene. Caso este número seja menor ou igual ao parâmetro *taxa de anulação* este gene será anulado para que posteriormente venha a ser reconstruído.

O processo de reconstrução consiste em enviar o cromossoma com genes anulados para o *FILTERSIM*. Através da estatística de múltiplos pontos este algoritmo reconstruirá o cromossoma utilizando os genes não anulados como dados condicionantes. Desta forma, é possível obter uma população com uma vasta diversidade genética, pois após o processo de anulação e reconstrução todos os indivíduos se tornam diferentes.

É importante destacar que durante o processo de anulação e reconstrução do indivíduo, os dados provenientes de perfis dos poços não são alterados. Estes dados são considerados confiáveis e imutáveis. Também é importante ressaltar que a utilização da estatística de múltiplos pontos poda o espaço de busca, evitando que sejam gerados indivíduos inválidos.

Em [3], é ilustrada a aplicação do procedimento de geração da população inicial. Esse procedimento foi adaptado para que pudesse ser utilizado tanto na inicialização da população de mapas de permeabilidade quanto de porosidade. No passo 1 da Figura 4.4 é apresentado um caso base genérico e, em seguida, no passo 2, são apresentados os indivíduos gerados a partir da anulação aleatória do caso base. Por fim, no passo 3, são apresentados os indivíduos após o processo de reconstrução com a estatística de múltiplos pontos através do algoritmo *FILTERSIM*.

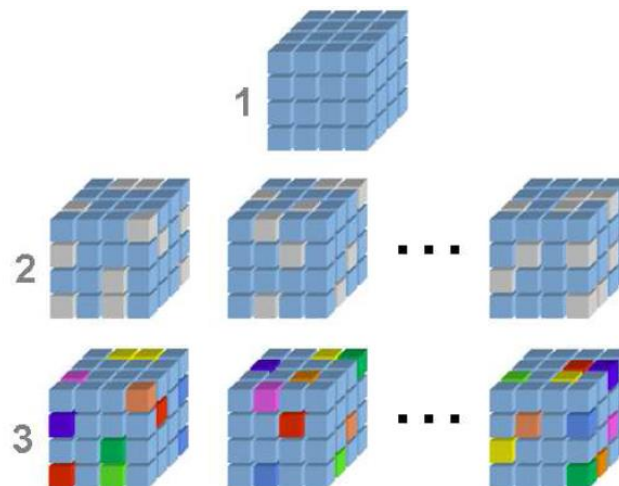


Figura 4.4: Geração da população inicial (Fonte: Silva, 2011)

4.1.2 Módulo de Evolução

O módulo de evolução, Figura 4.5, foi subdividido em cinco submódulos: módulo reprodutor, *FILTERSIM*, módulo avaliador, módulo distribuidor e o módulo interpretador. Este módulo é responsável por controlar todo o processo evolutivo, inclusive os critérios de parada do algoritmo genético.

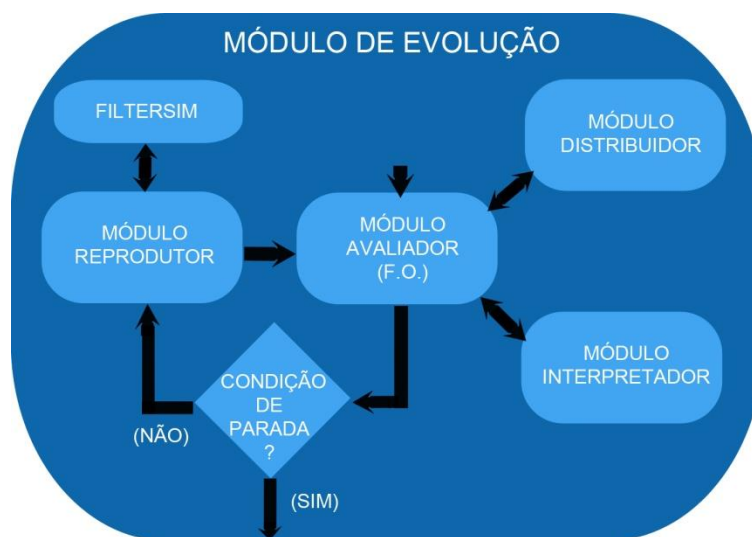


Figura 4.5: Módulo de Evolução

4.1.2.1 Módulo Reprodutor

O módulo reprodutor é responsável por evoluir as atuais soluções do problema em questão utilizando-se dos operadores genéticos de mutação e

cruzamento combinados com a estatística de múltiplos pontos, aplicada pelo *FILTERSIM*. Os operadores de cruzamento e de mutação são apresentados na Seção 4.2.

4.1.2.2 **Módulo Avaliador**

O módulo avaliador é responsável por guiar a busca realizada pelo algoritmo genético coevolutivo. Utilizando o serviço dos módulos distribuidor e interpretador, o módulo avaliador calcula e atribui um valor de aptidão a cada solução.

Em algoritmos genéticos coevolutivos, subsoluções (subcomponentes) são evoluídas separadamente em suas respectivas populações e no momento da avaliação é necessário montar uma solução completa que represente adequadamente o problema. O módulo avaliador é quem define as regras de como estes indivíduos cooperarão para a construção desta solução completa.

Durante o processo de avaliação de uma determinada população, o módulo avaliador seleciona o melhor indivíduo da população cooperadora para construir as soluções completas a serem avaliadas. Em outras palavras, durante a avaliação da população de mapas de porosidade, é selecionado o melhor indivíduo da população de mapas de permeabilidade para cooperar com cada indivíduo da população de mapas de porosidade na construção das soluções completas e vice-versa.

Após a construção de todas as soluções para uma determinada população, elas são enviadas ao módulo distribuidor que se encarregará de realizar as simulações necessárias utilizando o software IMEX [7].

Concluídas todas as simulações, o módulo avaliador utilizará o serviço do módulo interpretador para adquirir todas as informações de produção necessárias para o cálculo da aptidão de cada indivíduo da população. A função objetivo, ou seja, a função de avaliação utilizada pelo módulo avaliador está definida na Seção 4.2.2.

4.1.2.3

Módulo Distribuidor

O módulo distribuidor é responsável por fazer a integração do sistema desenvolvido com o distribuidor de tarefas *STarWeb* [4], que realizará a distribuição das simulações.

Esse módulo foi desenvolvido para permitir a distribuição das simulações dos modelos de reservatório. Os algoritmos genéticos apresentam um paralelismo nato, encontrando várias soluções durante uma única geração. Este potencial computacional paralelo é desperdiçado ao realizar todas as simulações, de uma geração, em sequência. Através da integração com o software *STarWeb* foi possível superar este gargalo imposto ao algoritmo genético, realizando várias simulações simultâneas em uma mesma geração. Desse modo, o poder computacional paralelo natural dos algoritmos genéticos foi melhor aproveitado nesta solução.

Um fluxograma, exibido na Figura 4.6, apresenta o procedimento de integração com o distribuidor de tarefas *STarWeb*.

O primeiro passo para se inicializar a distribuição de tarefas é a verificação do total de tarefas a serem realizadas. Em seguida, o módulo distribuidor testa a comunicação com o software *STarWeb*. Após a checagem da comunicação, é inicializado o procedimento de preparação do repositório. Este repositório é uma pasta compartilhada em rede, que permite a troca de informações entre o software *STarWeb* e a aplicação desenvolvida. A preparação deste repositório consiste em criar uma estrutura de diretórios, onde cada diretório representa uma solução completa do problema proposto. Dentro de cada diretório estão todos os arquivos necessários para se realizar a simulação daquele correspondente modelo de reservatório otimizado.

Terminada a estruturação do repositório, é necessário informar ao *STarWeb* as tarefas que devem ser executadas. Para isto, é gerado um arquivo *XML* de acordo com um padrão pré-definido pelo distribuidor *STarWeb*, informando-o as ordens de comando referentes a cada tarefa a ser realizada. Finalizada a geração das ordens de comando, é permitido ao *STarWeb* executar a distribuição das tarefas para todas as máquinas disponíveis.

Ao finalizar todas as tarefas, neste caso as simulações, é preciso recuperar os arquivos de saída do simulador com os resultados desejados. Estes arquivos serão copiados do repositório para as suas respectivas pastas de origem.

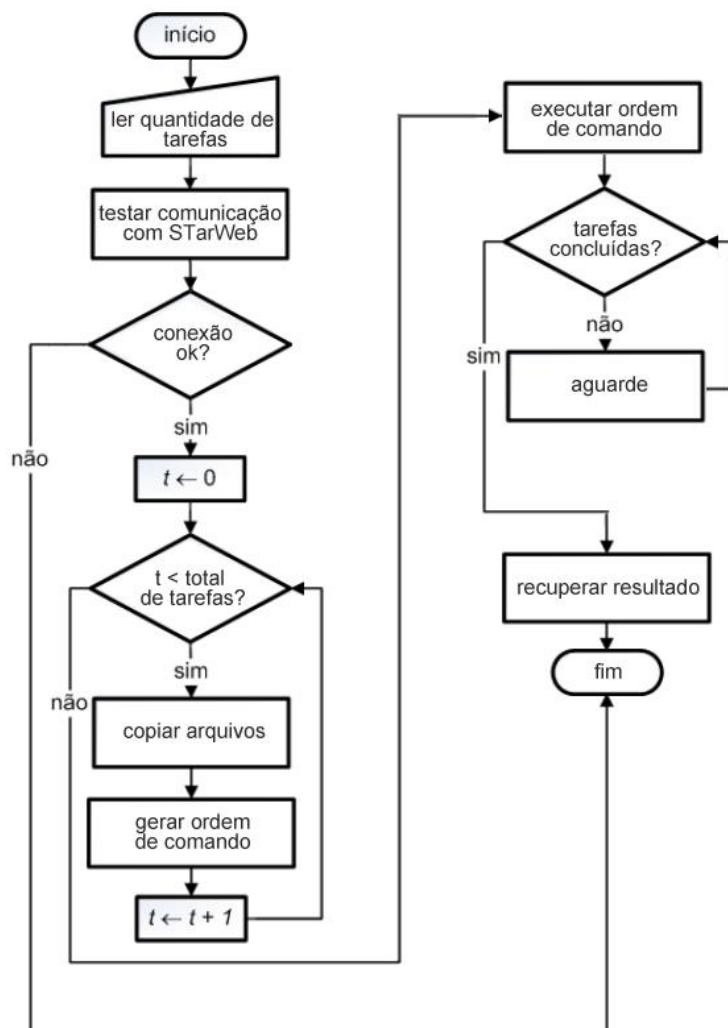


Figura 4.6: Fluxograma de integração com o distribuidor de tarefas

4.1.2.4 Módulo Interpretador

O módulo interpretador é responsável por interpretar o resultado de cada simulação e informar ao módulo avaliador o conjunto de dados necessários para o cálculo da aptidão de cada indivíduo.

Este módulo foi desenvolvido utilizando *expressões regulares* para agilizar a localização dos dados desejados. O encapsulamento dos métodos de leitura do arquivo de saída do simulador (*.out) simplifica extremamente o processo de recuperação dos dados desejados. A interface ficou muito amigável e autoexplicativa, sendo possível recuperar as informações desejadas do arquivo (*.out) com apenas três comandos.

Também é possível captar dados separados por grupos de poços, previamente definidos no arquivo (*.dat) ou (*.inc). Os dados podem ser obtidos por demanda ou por completo, flexibilizando ao usuário, um melhor gerenciamento da memória e rapidez na aquisição das informações desejadas.

A independência do módulo interpretador possibilita uma fácil adaptação a qualquer solução que deseje reutilizá-lo para obter informações do arquivo de saída do simulador CMG – IMEX (*.out). Esta reusabilidade é um ponto forte deste módulo, pois agilizará consideravelmente o tempo de implementação de novas soluções que utilizem o simulador CMG – IMEX [7].

Até o momento, o módulo interpretador disponibiliza as seguintes séries de dados:

- Nome dos grupos definidos no arquivo (*.dat);
- Datas;
- Dias;
- Grupos de poços;
- Nome dos poços;
- Número dos poços;
- Tipo dos poços;
- Quantidade de poços por grupo;
- GasProductionRates
- OilProductionRates
- WaterProductionRates
- CumulativeGasProductions
- CumulativeOilProductions
- CumulativeWaterProductions
- GasInstSurfaceInjectionRates
- GasInstSurfaceProductionRates
- GasSurfaceCumulativeInjection
- GasSurfaceCumulativeProduction
- GorSurfaceProductionRates
- OilInstSurfaceInjectionRates
- OilInstSurfaceProductionRates
- OilSurfaceCumulativeInjection
- OilSurfaceCumulativeProduction

- WaterInstSurfaceInjectionRates
- WaterInstSurfaceProductionRates
- WaterSurfaceCumulativeInjection
- WaterSurfaceCumulativeProduction
- WcutSurfaceProductionRates
- WgrSurfaceProductionRates

4.1.3

Módulo de Apresentação

O módulo de apresentação, Figura 4.7, foi subdividido em quatro submódulos: módulo gerenciador de análises, módulo analisador de produção, módulo analisador de mapas e o módulo analisador de evolução. Este módulo é responsável por todo o processamento de dados para gerar o resultado final da evolução.

4.1.3.1

Módulo Gerenciador de Análises

O módulo gerenciador de análises é responsável pela gerência de todo o processamento necessário para gerar a saída do sistema.

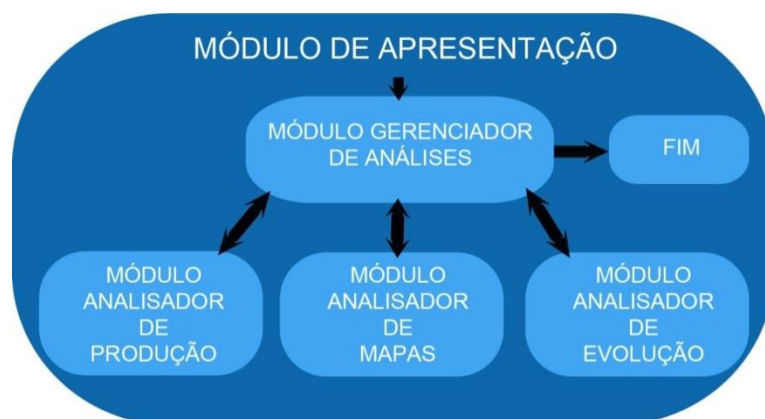


Figura 4.7: Módulo de Apresentação

4.1.3.2

Módulo Analisador de Produção

O módulo analisador de produção é responsável por processar os dados de produção gerados através da simulação do modelo de reservatório, a fim de gerar os gráficos das curvas de produção de cada poço e calcular o valor do

$RMSE$ de produção do histórico e da previsão ($RMSE_p$). Para o cálculo do $RMSE_p$ é utilizada a Equação 4.1.

4.1.3.3

Módulo Analisador de Mapas

O módulo analisador de mapas é responsável por processar os dados contidos nos mapas das propriedades de interesse, gerados após a otimização, e calcular o valor do $RMSE$ referente a cada propriedade ($RMSE_m$). Para o cálculo do $RMSE_m$ é utilizada a Equação 5.1.

4.1.3.4

Módulo Analisador de Evolução

O módulo analisador de evolução é responsável por processar os dados de evolução, gerados durante o processo de otimização, a fim de gerar o gráfico da curva de evolução da melhor solução encontrada pelo algoritmo genético coevolutivo.

4.2

Modelo Genético

A representação genética é um aspecto fundamental na modelagem de um algoritmo genético para a solução de um determinado problema. Ela define a estrutura do cromossoma de um indivíduo, com os respectivos genes que o compõem, de maneira que seja capaz de descrever todo o espaço de busca relevante do problema.

É imperativa a elaboração de operadores genéticos de cruzamento que sejam capazes de aproveitar ao máximo o material genético dos pais (geradores), levando assim, características importantes de uma geração para a subsequente. A exploração do espaço de busca está diretamente ligada à eficiência do operador de mutação, o qual inclui diversidades genéticas na população.

4.2.1

Codificação e Decodificação

Em [3], foi elaborada uma representação cromossômica que se mostrou bastante eficiente e eficaz na modelagem do problema de ajuste de histórico de modelos de reservatório de petróleo. As soluções foram representadas por

indivíduos (cromossomas) unidimensionais cujos genes são preenchidos com números reais. Cada gene armazena o valor da propriedade de interesse em um bloco do modelo do reservatório. Deste modo, o número de genes que compõem o cromossoma é igual ao número de blocos presentes no modelo de simulação.

Apesar de o cromossoma apresentar uma estrutura unidimensional, em sua decodificação é estabelecida uma correspondência entre cada gene e as respectivas coordenadas i , j e k do modelo do reservatório. Segundo [3], a inclusão dos blocos inativos e ativos na composição do cromossoma tem a finalidade de facilitar a correspondência entre as coordenadas dos blocos e os genes do cromossoma no momento da decodificação.

A Figura 4.8 exibe a decodificação de um cromossoma bem como a sua representação.

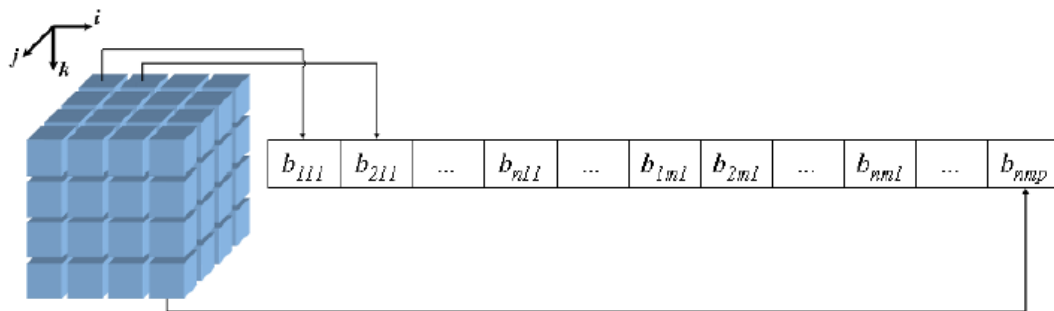


Figura 4.8: Representação e decodificação do cromossoma (Fonte: Silva, 2011)

Devido à característica genérica desta representação será possível utilizá-la em todas as populações desta solução.

4.2.2

Função Objetivo

A busca exploratória realizada pelo algoritmo genético coevolutivo é direcionada pela função objetivo, também conhecida como função de avaliação. Como o nome sugere, esta função é responsável por fornecer uma avaliação (aptidão), de acordo com o objetivo a ser alcançado, para cada solução encontrada pelo algoritmo durante o processo evolutivo.

Nesta abordagem do problema de ajuste de histórico, a avaliação é dada pela diferença entre as taxas de produção real de água e a simulada pelo modelo de simulação do reservatório.

Segundo [3], prepara-se o modelo de reservatório de maneira que a simulação sempre respeite a taxa de produção real de óleo, ou seja, aquela medida diretamente nos poços. Assim, para efeito de avaliação do indivíduo, mede-se a diferença entre as taxas de produção real e simulada de água em cada poço, ou ainda, as diferenças entre as pressões de fundo de poço reais e simuladas. A imposição da taxa de produção de óleo ao modelo é devido ao fato de haver critérios mais rígidos nessa medição, uma vez que é aferida pelo governo federal. Logo, há menos incertezas associadas a essa medição.

A Equação 4.1 assemelha-se ao procedimento prático, na qual os dados de produção observados e simulados correspondem às taxas de produção de água real e simulada, respectivamente.

$$RMSE_p = \sum_{j=1}^m \left(\sqrt{\frac{\sum_{i=1}^n (d_{obs}(i) - d_{sim}(i))^2}{n}} \right) \quad (4.1)$$

Onde:

$RMSE_p$ é o erro entre os dados de produção;

d_{obs} é o dado de produção observado;

d_{sim} é o dado de produção simulado;

n é a quantidade de dados de produção presentes na série;

m é a quantidade de poços produtores do reservatório.

4.2.3

Operador de Cruzamento

O operador de Cruzamento é responsável por promover a troca de informações genéticas entre dois indivíduos de uma população.

Em [3], foi elaborado um operador que realizava a troca de uma ou mais seções do reservatório entre dois indivíduos. Cada seção trocada é definida por um conjunto de coordenadas i , j e k iniciais e finais, que identifica exatamente a seção do reservatório. Estas coordenadas i , j e k são definidas aleatoriamente, de forma que as seções possam apresentar tamanhos diferentes e haver também sobreposições entre as seções. A Figura 4.9 ilustra o funcionamento deste operador genético.

No passo 1 são apresentados dois indivíduos genéricos e em cada um está destacada a seção a ser trocada entre os indivíduos. O passo 2 apresenta

os indivíduos após a troca da seção e, em seguida, no passo 3 parte da borda da seção é anulada em cada indivíduo. Por fim, no passo 4 são apresentados os indivíduos após a reconstrução com o *FILTERSIM* [3].

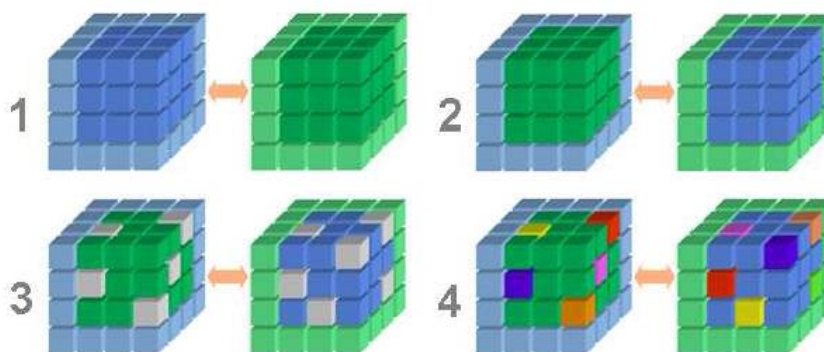


Figura 4.9: Operador Crossover (Fonte: Silva, 2011)

A finalidade da anulação e posterior reconstrução das bordas das seções utilizando o *FILTERSIM*, segundo [3], é suavizar as possíveis discontinuidades que possam surgir nos indivíduos após as trocas das seções. Indivíduos com estas discontinuidades podem representar mapas que estejam em desacordo com as características esperadas para a propriedade de interesse e, portanto, podem não preservar as estatísticas de múltiplos pontos.

A anulação aleatória das bordas das seções não garante, devido à sua característica aleatória, que todas as discontinuidades que possam surgir nas bordas das seções sejam corrigidas pela estatística de múltiplos pontos. Deste modo, poderiam ser gerados indivíduos com mapas em desacordo com as características esperadas para a propriedade de interesse. Diante disto, este trabalho propõe uma modificação neste operador de cruzamento.

A alteração no operador consiste em anular toda a borda da seção trocada, de modo que, seja dada à estatística de múltiplos pontos, através do algoritmo *FILTERSIM*, a chance de corrigir toda e qualquer discontinuidade apresentada na borda desta seção. Evitando assim, que um bloco que deveria ter sua propriedade corrigida não seja selecionado.

4.2.4 Operador de Mutação I

O operador de mutação é responsável por promover a diversificação das informações genéticas em uma população de indivíduos. Esta diversificação é

importante para que o algoritmo genético possa explorar melhor o espaço de busca das soluções do problema.

Em [3], foi apresentado um operador de mutação que promove a anulação de alguns genes do cromossoma para que este seja reconstruído utilizando a estatística de múltiplos pontos através do algoritmo *FILTERSIM*. Esse operador é aplicado aos blocos do modelo utilizando uma taxa de aplicação adaptativa que varia ao longo de todo o modelo do reservatório. O cálculo desta taxa é realizado em três etapas. Primeiro, a cada camada do modelo de simulação é associada uma distribuição normal centrada no bloco onde cada poço produtor é completado. O poço com maior erro de ajuste é tomado como referencia e a este é associada uma distribuição normal com uma amplitude igual à taxa de mutação do algoritmo genético, Equação 4.2. Este valor é utilizado como fator de normalização para as amplitudes associadas aos outros poços produtores.

$$a_k = \frac{RMSE_{pk} \times t_m}{RMSE_{pmax}} \quad (4.2)$$

Onde:

a_k é a amplitude da distribuição associada ao poço produtor k ;

$RMSE_{pk}$ é o erro de ajuste do poço produtor k ;

t_m é a taxa de mutação estabelecida para o algoritmo genético;

$RMSE_{pmax}$ é o maior erro de ajuste entre os poços produtores.

A segunda etapa é calcular as taxas de anulação de todos os blocos de cada camada do modelo para cada distribuição normal associada a um poço produtor. A Equação 4.3 apresenta este cálculo, onde o desvio padrão associado à distribuição é responsável por estabelecer a maneira como as taxas decaem na vizinhança de cada poço.

$$m_k(i, j) = a_k \times e^{\left(\frac{-\left((i-k_i)^2 + (j-k_j)^2 \right)}{2\sigma^2} \right)} \quad (4.3)$$

Onde:

$m_k(i, j)$ é a taxa de anulação no bloco (i, j) relativa ao poço k ;

a_k é a amplitude da distribuição associada ao poço produtor k ;

k_i e k_j são as coordenadas do bloco em que o poço k é completado.

A terceira e última etapa é o cálculo do valor final da taxa de anulação em um dado bloco do modelo. Após o cálculo das taxas de anulação para todas as distribuições em uma camada, o valor final da taxa de anulação em um determinado bloco é dado pelo máximo entre as taxas calculadas para todas as distribuições naquele bloco. Este cálculo é apresentado na Equação 4.4.

$$m(i, j) = \max_k(m_k(i, j)) \quad (4.4)$$

Onde:

$m(i, j)$ é a taxa de anulação final no bloco (i, j) ;

$m_k(i, j)$ é a taxa de anulação no bloco (i, j) relativa ao poço k .

Em [3], é ilustrada a distribuição das taxas de anulação ao longo de uma camada qualquer de um reservatório hipotético contendo nove poços produtores, sendo estes espaçados igualmente por toda a malha do modelo. A primeira etapa apresentada na Figura 4.10 exibe a distribuição dos poços na malha da camada do modelo enquanto a segunda etapa mostra a distribuição final das taxas de anulação. É possível notar que a distribuição associada ao poço $P3$ apresenta a maior amplitude e conseqüentemente os blocos ao redor deste poço possuem maiores taxas de anulação, pois este é o poço com maior erro de ajuste. Do mesmo modo, é possível visualizar que os blocos na vizinhança do poço $P1$ apresentam taxas de anulação mais baixas, caracterizando assim um baixo erro de ajuste em relação ao poço $P3$.

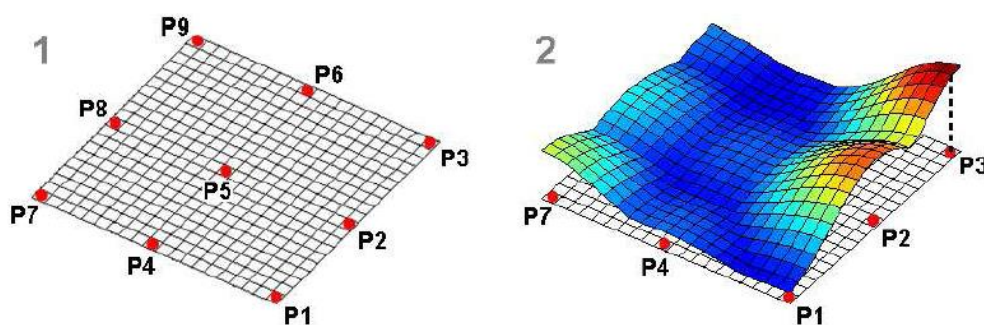


Figura 4.10: Distribuição das taxas de anulação em uma camada do reservatório
(Fonte: Silva, 2011)

A Figura 4.11 exibe um fluxograma completo para o cálculo da taxa de anulação adaptativa em uma camada do modelo.

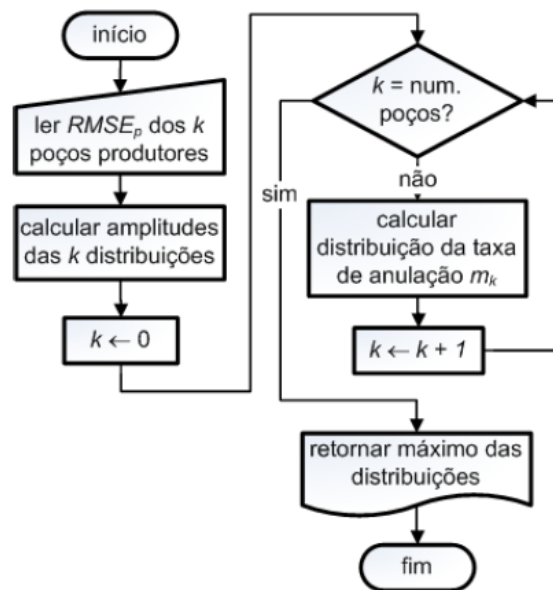


Figura 4.11: Cálculo da taxa de anulação adaptativa (Fonte: Silva, 2011)

4.2.5 Operador de Mutação II

O Operador de Mutação I mostrou-se eficiente em ajustar as propriedades próximas aos poços produtores, mas este sugere uma restrição à ação exploratória aleatória do algoritmo genético. Diante disto, este trabalho também propõe a utilização de um segundo operador de mutação que permita ao algoritmo genético coevolutivo explorar toda a área de busca de soluções para o problema. Deste modo, é possível conceder maior liberdade ao algoritmo na inclusão de diversidade genética na população.

O Operador de Mutação II procura tirar maior proveito da característica aleatória dos algoritmos evolutivos. Ele permite a anulação aleatória de qualquer gene do cromossoma. Após a anulação, este gene será reconstruído utilizando a estatística de múltiplos pontos. A Figura 4.12 apresenta o funcionamento deste operador em três passos.

É importante resaltar que durante o processo de anulação e reconstrução dos novos indivíduos, os dados provenientes de perfis dos poços continuarão inalterados, pois são considerados confiáveis e imutáveis.

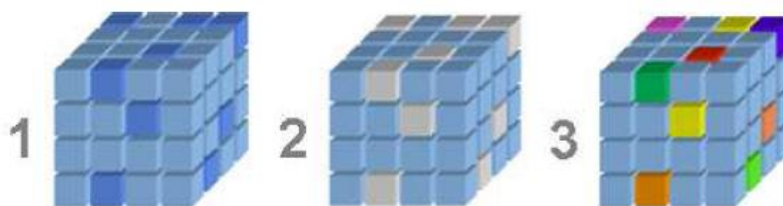


Figura 4.12: Operador de Mutação II

O primeiro passo é a seleção dos genes do cromossoma, ou seja, quais blocos do modelo de simulação serão anulados. O passo 2 realiza a anulação da propriedade em cada bloco selecionado anteriormente. Por fim, o terceiro passo realiza a reconstrução da propriedade utilizando a estatística de múltiplos pontos através do algoritmo *FILTERSIM*.

4.3 Modelo de Solução II

No modelo de solução apresentado na Seção 4.1, foi utilizada uma técnica para renovar a diversidade genética das populações de tempos em tempos sem perder o melhor indivíduo encontrado até o momento. Esta técnica baseia-se na reinicialização das populações mantendo-se um percentual dos melhores indivíduos. Este percentual é definido por uma taxa denominada *Steady State entre Rodadas (experimentos)*.

Durante o processo de otimização são realizados vários experimentos, nos quais, as populações são evoluídas de geração em geração. No início de cada experimento são geradas novas populações, sendo estas compostas por um percentual de indivíduos pertencentes às populações evoluídas do último experimento e por novos indivíduos. Estes novos indivíduos são gerados a partir do caso base, fornecido pelo especialista, conforme dissertado na Seção 4.1.1.2 e apresentado no fluxograma da Figura 4.3.

Diante disto, foi proposta uma alteração no processo evolutivo do Modelo de Solução I para melhorar o desempenho do algoritmo genético coevolutivo. Esta alteração se aplicada na geração das populações iniciais de cada novo experimento a ser realizado.

As populações iniciais de cada experimento no Modelo de Solução II são compostas por indivíduos provenientes de três métodos:

O primeiro método, assim como no Modelo de Solução I, é a seleção de um percentual de melhores indivíduos provenientes das populações evoluídas

do último experimento realizado. Este percentual também é definido pela taxa *Steady State entre Rodadas (experimentos)*.

O segundo método é a geração de indivíduos a partir do caso base, fornecido pelo especialista, através do método de anulação e reconstrução dos genes utilizando a geoestatística de múltiplos pontos (Figura 4.3).

O terceiro método é a geração de indivíduos a partir do melhor indivíduo da população evoluída do último experimento realizado até o momento, ou seja, a partir do indivíduo otimizado da mesma espécie.

A geração de novos indivíduos a partir do caso, ou indivíduo otimizado, é uma tentativa de tirar proveito do trabalho de busca já realizado pelo algoritmo genético, no experimento anterior. Gerar indivíduos a partir de um caso otimizado tem maiores chances de gerar soluções promissoras quando comparadas com indivíduos gerados a partir do caso base, apresentado pelo especialista. Entretanto não se podem descartar os indivíduos gerados a partir do caso base, pois estes contém o conhecimento do especialista e serão de grande utilidade durante o processo evolutivo.

É importante destacar que durante o processo de anulação e reconstrução dos novos indivíduos, os dados provenientes de perfis dos poços continuarão inalterados, assim como no Modelo de Solução I. Estes dados são considerados confiáveis e imutáveis. A utilização da estatística de múltiplos pontos também continuará podando o espaço de busca, evitando que sejam gerados indivíduos inválidos.

A taxa de indivíduos gerados a partir do caso base e do caso otimizado são idênticas e dada pela Equação 4.5.

$$T_{GNI} = \frac{1 - SSR}{2} \quad (4.5)$$

Onde:

T_{GNI} é a taxa de geração de novos indivíduos

SSR é a taxa *Steady State entre Rodadas (experimentos)*

O fluxograma exibido na Figura 4.13 exemplifica o procedimento para a geração de populações iniciais do Modelo de Solução II. Este procedimento deve ser realizado para cada espécie participante do processo de coevolução cooperativa.

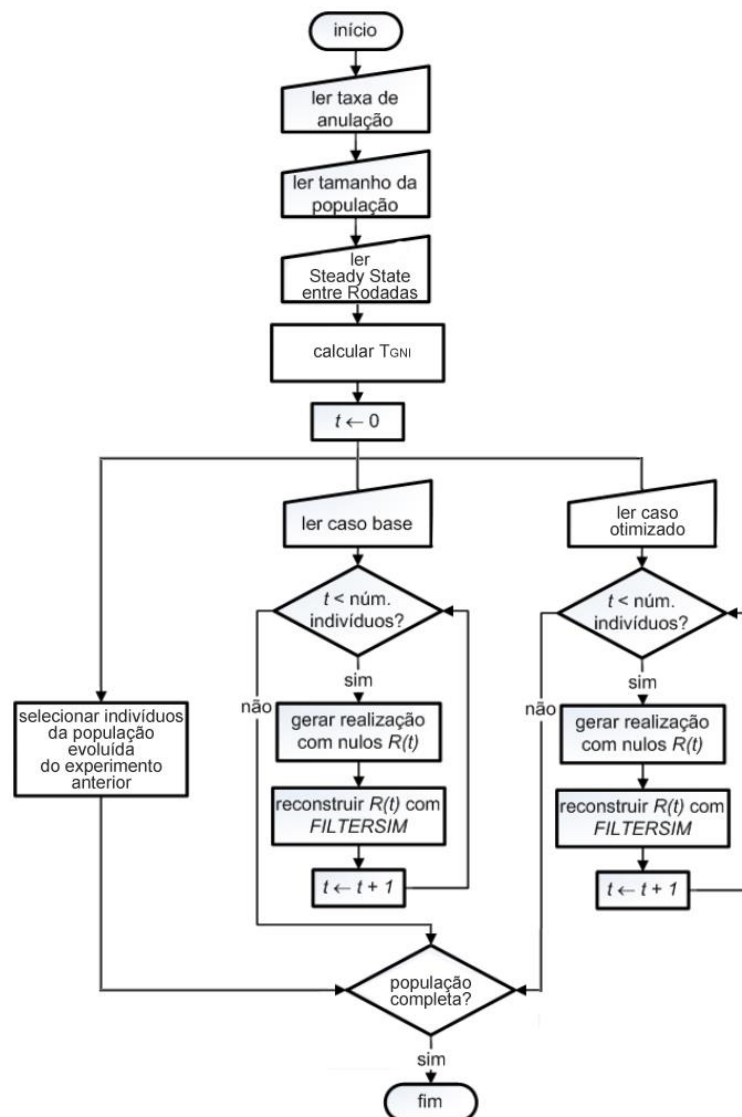


Figura 4.13: Fluxograma de geração da população inicial Modelo de Solução II

O resultado que se espera com esta alteração é uma aceleração na localização de melhores mapas de propriedades do modelo de simulação, o que consequentemente implicará na redução do erro de ajuste das curvas de produção ($RMSE_p$), melhorando assim, o desempenho do algoritmo genético coevolutivo.

4.4 Detalhes de Implementação

Para a implementação do modelo de solução proposto nesse trabalho, utilizou-se a linguagem de programação C# e o MS Visual Studio 2010. C# é

uma linguagem de programação orientada a objetos desenvolvida pela Microsoft e parte integrante de sua plataforma .NET. O MS Visual Studio 2010 é um ambiente de desenvolvimento integrado (IDE) que possui uma coleção de ferramentas expostas por meio de uma interface de usuário comum. Algumas dessas ferramentas são compartilhadas com outras linguagens do MS Visual Studio 2010.

Após o desenvolvimento do modelo de solução, foi criada uma interface gráfica utilizando o componente Windows Form. Esta interface gráfica é exibida na Figura 4.14 e permite a utilização do modelo de solução para o ajuste de histórico das curvas de produção de água através da otimização dos mapas de permeabilidade e de porosidade.

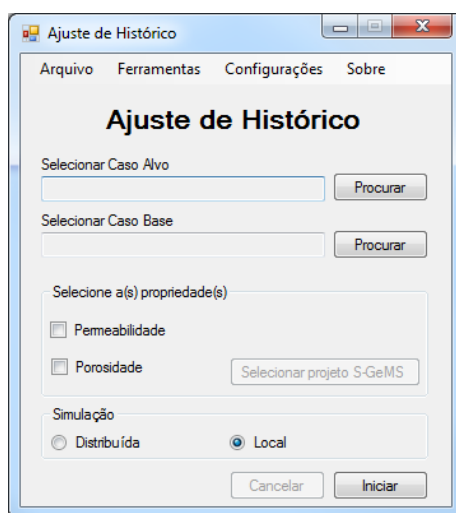


Figura 4.14: Interface gráfica do aplicativo de ajuste de histórico

Na tela principal da interface gráfica é possível selecionar o caso alvo assim como o caso base da otimização a ser realizada. Também foi disponibilizada uma seção onde o usuário deve selecionar a(s) propriedade(s) a serem otimizadas.

O usuário precisa informar através do botão “*Selecionar projeto SGeMS*”, um projeto previamente criado no software *SGeMS* [47]. O projeto deve conter quatro elementos, dois destes são as imagens de treinamento. Estas imagens serão utilizadas pelo algoritmo *FILTERSIM* para a aplicação da estatística de múltiplos pontos. Estes dois elementos devem ser do tipo *cartesian grid*. O nome da imagem de treinamento que representa o comportamento da permeabilidade deve ser *perm_ti* enquanto o nome da imagem que representa o comportamento da porosidade deve ser *por_ti*.

Outros dois elementos também necessitam estar presentes neste projeto e seus respectivos nomes devem ser *grid* e *conditional_data*. Para a criação do *grid*, é necessário instanciar um elemento do tipo *cartesian grid* que definirá os limites do modelo do reservatório, ou seja, a quantidade de blocos nas direções *x*, *y* e *z*. A criação do *conditional_data* é realizada instanciando um elemento do tipo *set point*, este elemento preencherá o *grid* com blocos nulos. Em [47], é descrito detalhadamente o processo de criação dos elementos citados acima.

Um projeto criado no SGeMS contendo os quatro elementos definidos acima é apresentado na Figura 4.15 que exhibe os elementos *conditional_data* e *grid*.

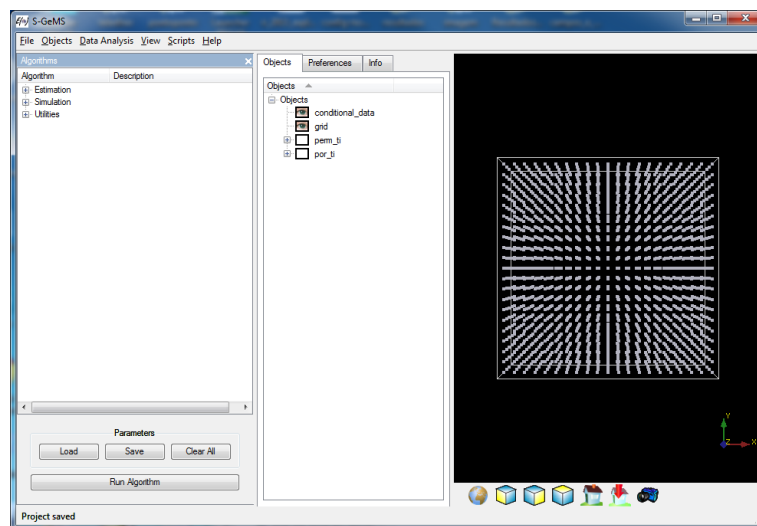


Figura 4.15: Exemplo de projeto criado no S-GeMS

Ainda na tela principal da interface gráfica, é permitido ao usuário selecionar se as simulações realizadas pelo software *CMG-IMEX* serão executadas distribuídas ou localmente. Caso o usuário selecione a opção distribuída, o aplicativo utilizará o módulo distribuidor que preparará toda a estrutura necessária para se realizar as simulações distribuídas através da comunicação com o software STarWeb.

Através do menu *Configurações* o usuário terá acesso a uma interface gráfica que permitirá a inserção das configurações necessárias para a execução do aplicativo. Esta interface gráfica, exibida na Figura 4.16, é composta por duas guias de configurações.

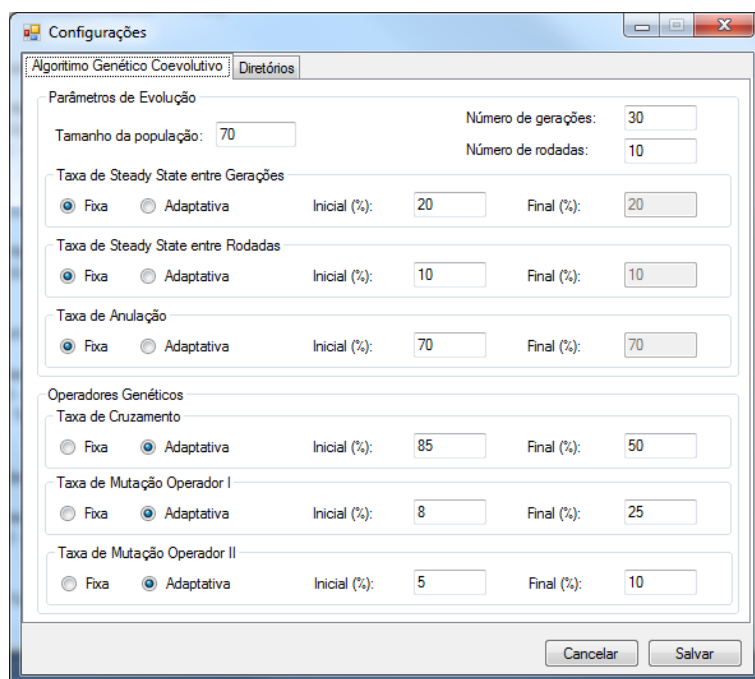


Figura 4.16: Configurações do Algoritmo Genético Coevolutivo

A primeira guia de configuração, Figura 4.16, exibe todos os parâmetros configuráveis do algoritmo genético coevolutivo. Pensando em uma flexibilidade a ser testada futuramente, a taxa de anulação dos casos base e otimizado para a geração das populações iniciais foi incluída, na interface gráfica, como uma taxa adaptativa.

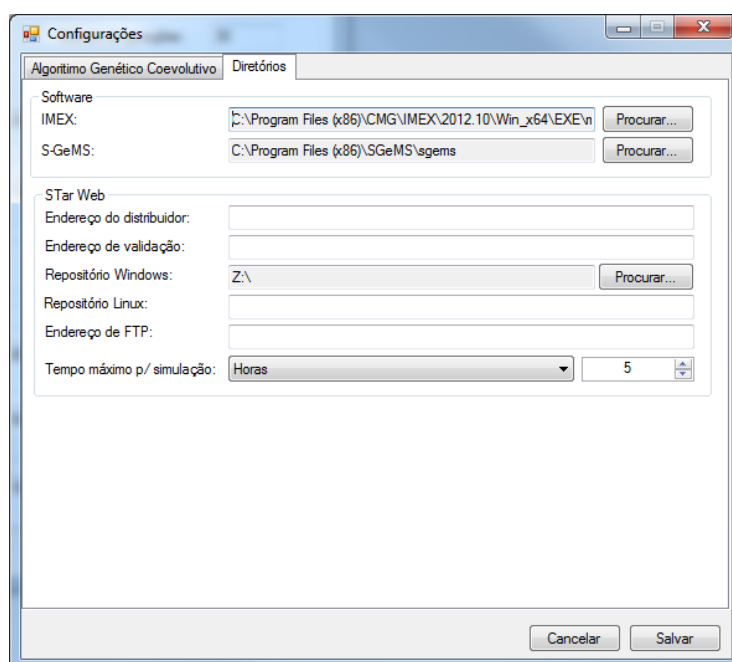


Figura 4.17: Configurações dos Diretórios

A segunda guia de configuração é exibida na Figura 4.17 e seus parâmetros configuráveis são os repositórios e o endereço de cada software utilizado pelo aplicativo.

Durante a pesquisa, desenvolvimento e teste do modelo de solução, algumas dificuldades, ou seja, trabalhos onerosos e repetitivos foram surgindo. Estas dificuldades estimularam o desenvolvimento de ferramentas de auxílio ao usuário durante o processo de ajuste de histórico. O menu *ferramentas* é exibido na Figura 4.18 e a primeira ferramenta apresentada é a criação de um arquivo do tipo *INCLUDE* a partir de um arquivo do tipo *TXT*.

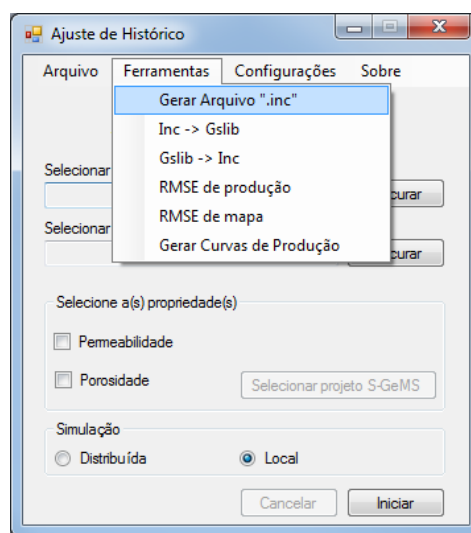


Figura 4.18: Ferramenta de geração de arquivo do tipo *INCLUDE*

A segunda e terceira ferramentas automatizam o processo de conversão dos arquivos entre os formatos *INCLUDE* e *GSLIB*. Após a conclusão do processo evolutivo do ajuste de histórico é desejável visualizar, graficamente, os mapas de permeabilidade e de porosidade. Para que seja possível carregar o mapa da propriedade de interesse no software *SGeMS*, é necessário converter o arquivo no formato *INCLUDE*, definido em [7], para o formato *GSLIB*, definido em [47]. A primeira interface gráfica exibida na Figura 4.19 é a de conversão do formato *INCLUDE* para *GSLIB* enquanto a segunda realiza a conversão inversa.

Para converter um arquivo do tipo *INCLUDE* em um *GSLIB* é necessário informar o arquivo *INCLUDE* e o número de blocos nas direções *I*, *J* e *K*. Para a conversão de um arquivo do tipo *GSLIB* em um *INCLUDE*, exige-se que seja informado um arquivo do tipo *GSLIB* e o número de colunas que se deseja no arquivo *INCLUDE*.

É importante destacar que o arquivo original é mantido intacto e um novo arquivo é gerado no formato desejado.



Figura 4.19: Ferramenta de conversão INCLUDE -> GSLIB e GSLIB -> INCLUDE

A quarta ferramenta desenvolvida realiza o cálculo do *RMSE* de produção do modelo de simulação do reservatório. A Figura 4.20 apresenta a interface gráfica que solicita ao usuário que informe o caso alvo e o caso otimizado. O usuário também precisa informar o período que deseja ser utilizado para o cálculo do *RMSE* de produção. Este período é definido por uma data inicial e final, elas devem ser selecionadas de acordo com as datas definidas no arquivo de saída do simulador *IMEX*. O *RMSE* de produção é calculado para o caso otimizado em relação ao caso alvo utilizando a Equação 4.1.

A quinta ferramenta apresenta um funcionamento parecido com a ferramenta citada acima. A interface gráfica solicita ao usuário que informe os mapas da propriedade de interesse para que se calcule o *RMSE* do caso otimizado em relação ao caso alvo. A Figura 4.21 apresenta a interface gráfica que utiliza a Equação 5.1 para calcular o *RMSE* de mapa.

A sexta e última ferramenta desenvolvida, assim como a quarta ferramenta, utiliza o módulo interpretador e o de apresentação para obter os dados de produção de cada poço definido no modelo de simulação e gerar uma saída para o usuário. Na Figura 4.22 é exibida a interface gráfica desta última ferramenta. Esta interface solicita ao usuário que informe o arquivo de saída do

simulador *IMEX* com os dados de produção a serem considerados. Também é necessário informar o período desejado através das datas inicial e final.

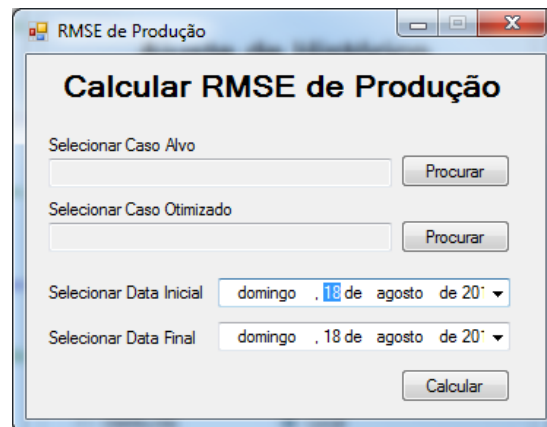


Figura 4.20: Ferramenta para o cálculo do RMSE de produção

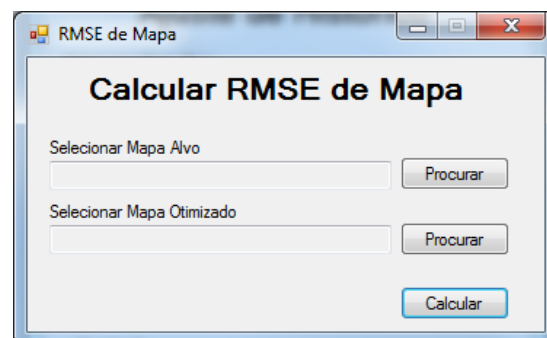


Figura 4.21: Ferramenta para o cálculo do RMSE de mapa

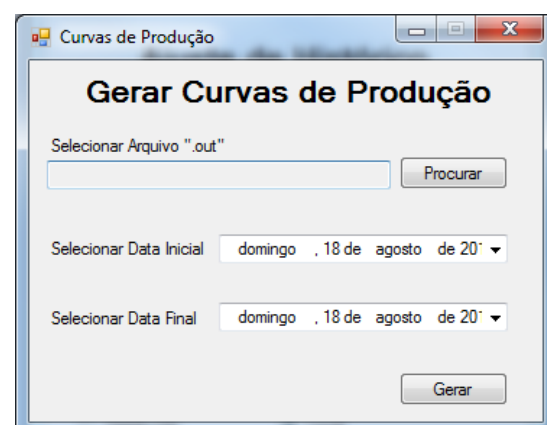


Figura 4.22: Ferramenta para a geração das curvas de produção

Este aplicativo de ajuste de histórico utilizando algoritmo genético coevolutivo, foi desenvolvido de acordo com os padrões e normas de desenvolvimento de software definidos pelo laboratório de inteligência

computacional aplicada ICA PUC-RIO. As classes de modelo e dos módulos foram definidas de acordo com a nomenclatura utilizada no projeto OCTOPUS 2.0. Portanto, assim que o Sistema Integrado de Gerência de Reservatórios – OCTOPUS 2.0 estiver disponível para inclusão de novos plug-ins, será possível à inclusão deste aplicativo como um plug-in, necessitando, basicamente, da substituição da interface gráfica por uma compatível com o padrão OCTOPUS 2.0.