5
Modelo de Solução

5.1
Consideraçãoes Iniciais

O processo de ajuste de histórico quando aplicado a um modelo de reservatório real, ou até mesmo a um modelo sintético, consiste em otimizar uma enorme quantidade de parâmetros. Esses parâmetros representam propriedades das formações rochosas que constituem o reservatório e estão distribuídos por toda a malha do modelo de simulação. Portanto, para uma dada propriedade, é necessário otimizar tantos parâmetros quantos são os blocos que compõem o modelo. 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.

Devido às complexidades mencionadas, o ajuste de histórico exige uma estratégia de otimização que seja, ao mesmo tempo, eficaz e eficiente. Isso significa que, além de proporcionar bons resultados, a estratégia deve ser capaz de lidar com grandes quantidades de variáveis. Além disso, deve ser suficientemente flexível para poder ser aplicada ao ajuste de qualquer propriedade de interesse. O modelo de solução proposto busca suprir essas necessidades unindo, em um modelo computacional híbrido, os benefícios proporcionados pela Geoestatística de Múltiplos Pontos, através do algoritmo *FILTERSIM*, e pela Inteligência Computacional, através dos Algoritmos Genéticos. O que caracteriza a união das duas técnicas no modelo de solução é a inclusão do algoritmo *FILTERSIM* tanto na geração da população inicial quanto na aplicação dos operadores genéticos de cruzamento e mutação do algoritmo genético. O fluxograma da Figura 5.1 ilustra o acoplamento entre as técnicas.

5.2
Representação da Solução

A aplicação de Algoritmos Genéticos à resolução de problemas de otimização está condicionada à obtenção de uma representação adequada para a

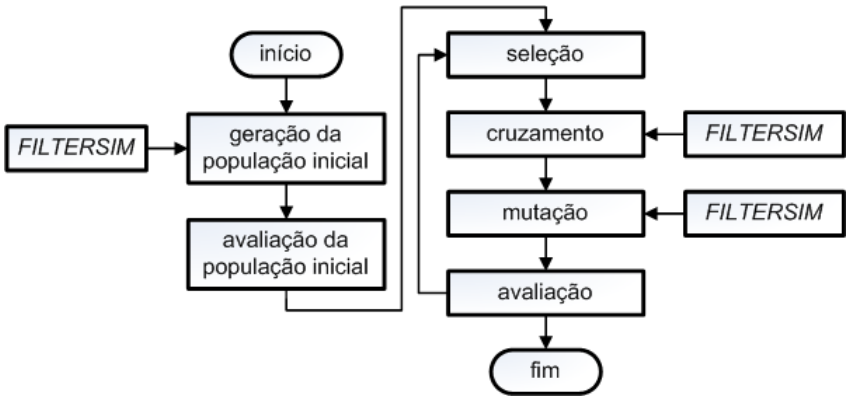

Figura 5.1: Fluxograma do modelo de solução

solução do problema e também de uma função objetivo que norteie a busca pelas melhores soluções. Aqui, as soluções são representadas por indivíduos (cromossomos) unidimensionais cujos genes são preenchidos com números reais, onde cada gene armazena o valor da propriedade de interesse em um bloco do modelo do reservatório. Dessa forma, o número de genes que compõe o cromossomo é sempre igual ao número de blocos presentes no modelo de simulação.

Além da representação, é necessário estabelecer uma regra de decodificação que permita a construção da solução real a partir do cromossomo, de forma que possa ser efetivamente avaliada. Apesar de o cromossomo apresentar uma estrutura unidimensional, na sua decodificação é estabelecida uma correspondência entre cada gene e as respectivas coordenadas i, j e k do modelo do reservatório. Na Figura 5.2 estão ilustradas a representação e a decodificação de um cromossomo do modelo de solução proposto.

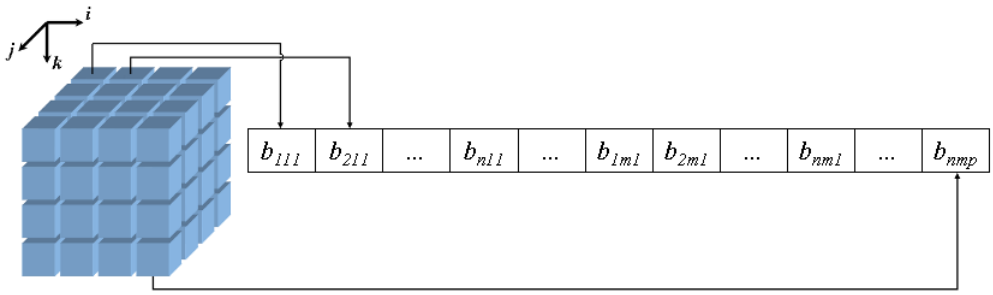

Figura 5.2: Representação e decodificação do cromossomo

É importante observar que tanto os blocos ativos quanto os inativos compõem o cromossomo. Todavia, a inclusão dos blocos inativos tem a finalidade apenas de facilitar a correspondência entre as coordenadas dos blocos e os genes do cromossomo no momento da decodificação. Durante a evolução,

os blocos inativos são totalmente desprezados e, portanto, não são modificados pelo algoritmo genético.

5.3 Função Objetivo

A função objetivo é responsável por fornecer o valor da avaliação de cada solução proposta pelo algoritmo genético ao longo do processo evolutivo. No ajuste de histórico, geralmente a avaliação é dada pela diferença entre as taxas de produção real e simulada dos fluidos do reservatório. Na prática, prepara-se o modelo do 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 devida 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. Aqui, conforme ilustra a Equação 5-1, adota-se uma abordagem semelhante à usada na prática, em que os dados de produção observados e simulados correspondem às taxas de produção de água real e simulada, respectivamente.

$$RMSE_p = \sum_{j=1}^{m} \left(\sqrt{\frac{\sum_{i=1}^{n} \left(d_{obs}\left(i\right) - d_{sim}\left(i\right) \right)^2}{n}} \right) \tag{5-1}$$

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.

5.4 Geração da População Inicial

A geração da população inicial de um algoritmo genético é fundamental para a eficiência do processo de otimização. Uma boa população inicial é aquela formada por indivíduos válidos, com boa avaliação e com ampla diversidade genética. Partindo-se de uma população inicial de indivíduos mais promissores,

há uma chance maior de se encontrar soluções melhores com uma quantidade menor de ciclos de evolução (gerações) do algoritmo genético. Porém, nem sempre é fácil obter indivíduos iniciais com essas características, especialmente em problemas cujos espaços de soluções são extremamente grandes, como é o caso do problema do ajuste de histórico.

Por isso, adotou-se uma estratégia que tira proveito das informações presentes no modelo de simulação atual (caso base), ou seja, aquele correntemente usado pelo especialista. Afinal, esse modelo é fruto de um trabalho bastante criterioso realizado por profissionais altamente especializados e, portanto, carrega muita informação útil a respeito da caracterização da reserva. Levando em consideração a importância dessas informações, cada indivíduo da população inicial é formado pelos valores da propriedade de interesse do caso base. Logo, num primeiro momento, todos os indivíduos da população inicial são idênticos.

Em seguida, a fim de garantir a diversidade genética, todos os indivíduos são submetidos a um processo de anulação e reconstrução. A anulação consiste em eliminar o valor da propriedade de interesse em determinados blocos do caso base, ou seja, anular determinados genes do cromossomo. O critério de escolha dos blocos a serem anulados consiste em sortear um número aleatório para cada gene do cromossomo e anular os genes cujos números sorteados sejam menores ou iguais a uma taxa de anulação previamente estabelecida. A taxa de anulação deve ser informada como entrada para o modelo de solução e a determinação de seu valor depende diretamente do conhecimento do especialista em relação ao caso base. Para situações em que o caso base se encontra mais desajustado é necessário estabelecer valores mais altos para a taxa de anulação e, obviamente, para um caso base não tão desajustado estabelece-se taxas de anulação menores. No Capítulo 6 é apresentado um estudo que analisa a sensibilidade dos resultados da otimização em relação à taxa de anulação e às características do caso base.

Vale destacar que, partindo da premissa de que os dados provenientes dos perfis dos poços são considerados confiáveis, o processo de anulação impede a eliminação do valor da propriedade em um bloco onde um poço é completado. Após a anulação, os blocos não anulados de cada indivíduo são usados como dados condicionantes para a reconstrução do próprio indivíduo através do algoritmo *FILTERSIM*.

Dessa forma, obtém-se uma população de indivíduos que, apesar de terem a mesma origem, se tornam diferentes após a anulação e a reconstrução. Além da diversidade genética necessária ao algoritmo genético, os indivíduos reconstruídos pelo algoritmo *FILTERSIM* tendem a ser promissores, uma vez que respeitam as estatísticas de múltiplos pontos. A Figura 5.3 apresenta um

fluxograma com o procedimento de geração dos indivíduos da população inicial.

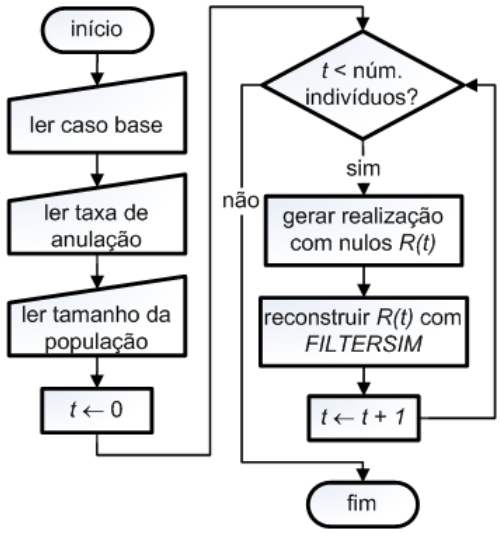

Figura 5.3: Fluxograma de geração da população inicial

Na Figura 5.4 está ilustrada a aplicação do procedimento de geração da população inicial. No passo 1 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. No passo 3, finalmente, são apresentados os indivíduos após o processo de reconstrução com o algoritmo *FILTERSIM*.

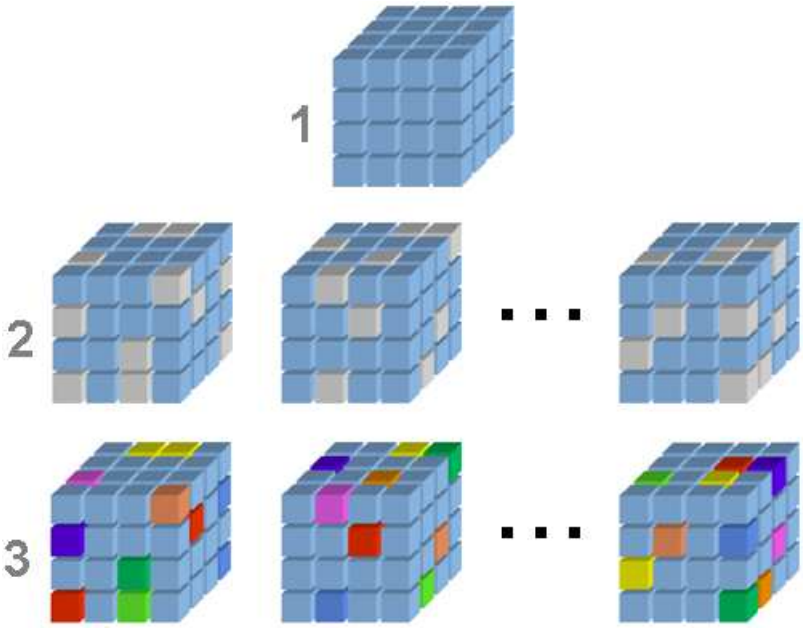

Figura 5.4: Geração da população inicial

5.5 Descrição dos Operadores Genéticos

Numa otimização por Algoritmos Genéticos a definição dos operadores genéticos é fundamental para a qualidade do processo evolutivo. Uma vez definida a representação cromossômica, é necessário estabelecer operadores que estejam de acordo tanto com a representação escolhida quanto com as características do problema de otimização. No caso do ajuste de histórico foram criados operadores que, assim como na geração da população inicial, se beneficiam da Geoestatística Multiponto para promover modificações nos indivíduos, sem que essas modificações comprometam a consistência geológica desses indivíduos.

5.5.1 Cruzamento

O operador de cruzamento consiste em trocar, entre dois indivíduos, um ou mais conjuntos de genes que representem seções do reservatório. Cada seção a ser trocada é definida por um conjunto de coordenadas i, j e k iniciais e finais, que identifica exatamante a seção do reservatório. Essas coordenadas i, j e k são definidas aleatoriamente, de forma que as seções podem apresentar tamanhos diferentes e pode haver também sobreposições entre as seções.

Após a troca entre os indivíduos, as bordas das seções trocadas são anuladas de acordo com o mesmo princípio adotado na geração da população inicial, ou seja, anula-se aleatoriamente as bordas de acordo com uma taxa de anulação previamente determinada. Em seguida, os indivíduos são reconstruídos através do algoritmo *FILTERSIM*. A Figura 5.5 ilustra o funcionamento do operador proposto para uma situação hipotética em que apenas uma seção é trocada entre os indivíduos. No passo 1 são apresentados dois indivíduos genéricos e em cada um está destacada a seção a ser trocada. 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*.

A finalidade da anulação e posterior reconstrução das bordas das seções é suavizar as possíveis descontinuidades que possam surgir nos indivíduos após as trocas das seções. Indivíduos com essas descontinuidades 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.

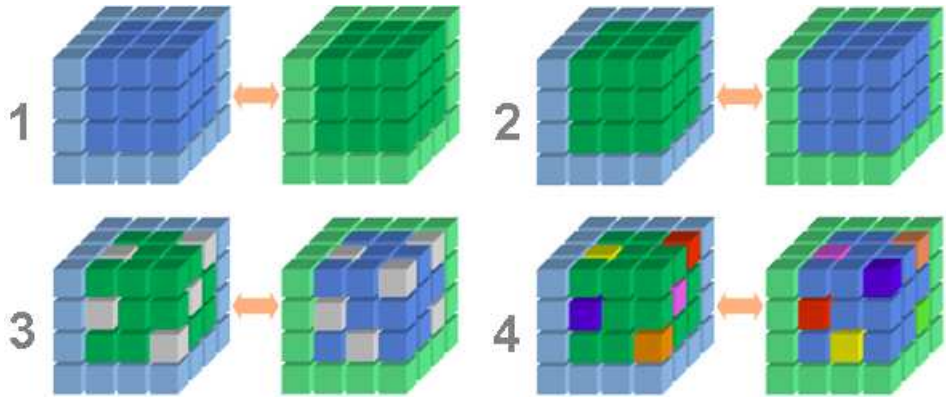

Figura 5.5: Operador de cruzamento

5.5.2 Mutação

O operador de mutação funciona de forma semelhante aos dois últimos passos do operador de cruzamento, ou seja, consiste em anular alguns genes do indivíduo para que este seja reconstruído pelo algoritmo *FILTERSIM*. Porém, nesse caso, a anulação pode ser aplicada a qualquer bloco do modelo, não se restringindo às bordas de uma ou mais seções, como acontece no cruzamento. A Figura 5.6 ilustra o funcionamento do operador para um caso hipotético. No passo 1 estão destacados os blocos do modelo selecionados para a anulação. No passo 2 são exibidos os blocos selecionados após a anulação. Finalmente, no passo 3, é apresentado o indivíduo reconstruído com o *FILTERSIM*. Assim como no cruzamento, a reconstrução com o *FILTERSIM* tem a finalidade de gerar um novo indivíduo que também preserve as estatísticas de múltiplos pontos.

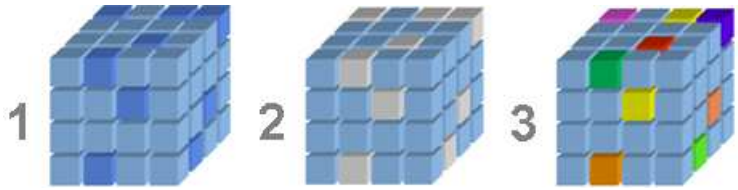

Figura 5.6: Operador de mutação

Da mesma maneira que na geração da população inicial e no operador de cruzamento, o critério de seleção dos blocos a serem anulados também é estabelecido de acordo com uma taxa de anulação. Nesse caso, porém, trata-se de uma taxa de anulação adaptativa que varia ao longo de todo o modelo do reservatório. De acordo com esse critério, cada bloco do modelo tem a sua própria taxa de anulação e o seu valor depende dos erros de ajuste medidos

nos poços produtores espalhados pelo reservatório. Conforme apresentado na Seção 5.3, em linhas gerais, o erro de ajuste é dado pela diferença entre o volume real de um fluido produzido por um poço e o volume de fluido obtido, para este mesmo poço, após a simulação do modelo do reservatório.

Considerando uma camada do modelo do reservatório, para o cálculo das taxas de anulação, associa-se a cada poço produtor uma distribuição normal, centrada no bloco em que o poço é completado, que determina as taxas de anulação dos blocos em torno do bloco central. O poço com o maior erro de ajuste é tomado como referência e a ele é associada uma distribuição normal cuja amplitude é igual à taxa de mutação estabelecida para o algoritmo genético. Esse valor é utilizado como fator de normalização para as amplitudes das distribuições associadas aos outros poços, de acordo com seus respectivos erros de ajuste. O cálculo das amplitudes das distribuições é dado pela Equação 5-2.

$$a_k = \frac{RMSE_{p_k} \cdot t_m}{RMSE_{p_{max}}} \tag{5-2}$$

Onde:

a_k é a amplitude da distribuição associada ao poço produtor k;
$RMSE_{p_k}$ é o erro de ajuste do poço produtor k;
t_m é a taxa de mutação estabelecida para o algoritmo genético;
$RMSE_{p_{max}}$ é o maior erro de ajuste entre os poços produtores.

Uma vez calculadas as amplitudes, para cada distribuição as taxas de anulação de todos os blocos da camada são calculados segundo a Equação 5-3. O desvio padrão associado à distribuição é responsável por estabelecer a maneira como as taxas decaem na vizinhança de cada poço.

$$m_k(i,j) = a_k \cdot e^{\left(\frac{-\left((i-k_i)^2+\left(j-k_j\right)^2\right)}{2\sigma^2}\right)} \tag{5-3}$$

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.

Após o cálculo das taxas de anulação para todas as distribuições em uma camada, o valor final da taxa em um bloco é dado pelo máximo entre as taxas de anulação calculadas para todas as distribuições naquele bloco. Dessa forma, garante-se que, ainda que muito baixa, sempre haverá uma probabilidade de anulação diferente de zero associada a todos os blocos. A Equação 5-4 apresenta o cálculo da taxa de anulação final. O fluxograma completo para o cálculo da taxa de anulação adaptativa em uma camada do modelo está ilustrado na Figura 5.7.

$$m\left(i,j\right) = \max_{k}\left(m_k\left(i,j\right)\right) \tag{5-4}$$

Onde:

$m\left(i,j\right)$ é a taxa de anulação final no bloco (i,j);
$m_k\left(i,j\right)$ é a taxa de anulação no bloco (i,j) relativa ao poço k.

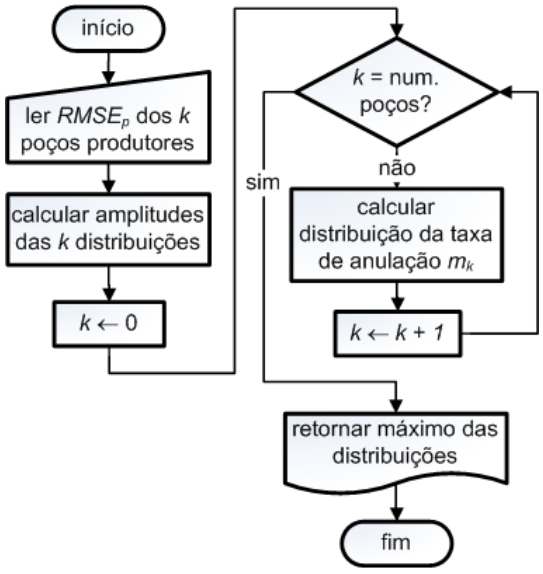

Figura 5.7: Fluxograma de cálculo da taxa de anulação adaptativa

A Figura 5.8 ilustra a distribuição das taxas de anulação ao longo de uma camada de um reservatório hipotético com 9 poços produtores igualmente espaçados. O passo 1 mostra a distribuição dos poços no modelo e o passo 2 mostra a distribuição final das taxas de anulação. Nota-se que a distribuição associada ao poço *P3* é a de maior amplitude, logo, esse é o poço com o maior erro de ajuste. Por isso, os blocos na vizinhança do poço *P3* apresentam as maiores taxas de anulação. Por outro lado, os blocos em torno do poço *P1*

apresentam taxa de anulação mais baixas, o que caracteriza um baixo erro de ajuste em relação ao poço *P3*.

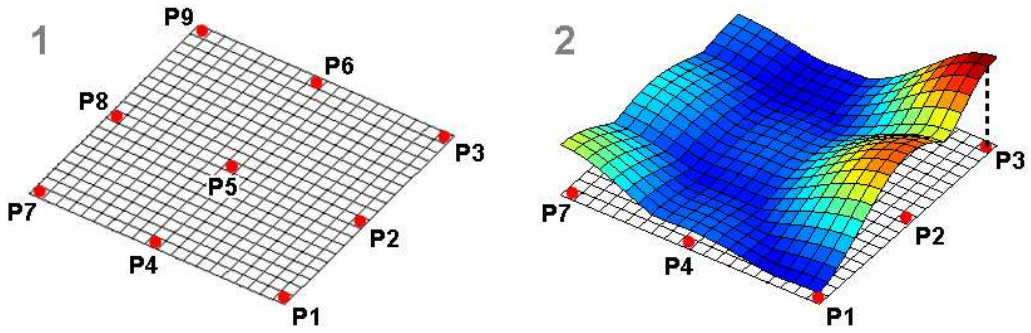

Figura 5.8: Distribuição das taxas de anulação em uma camada do reservatório

Com base nesse critério, o operador privilegia a ocorrência de mutações em blocos cujos valores da propriedade de interesse, provavelmente, contribuam mais para o erro de ajuste de um determinado poço e tende a manter inalterados os blocos cuja influência não é tão representativa. Esse mesmo procedimento é aplicado a todas as camadas do modelo do reservatório, de maneira que as distribuições das taxas de anulação variam de acordo com a geometria e com o modo de completação dos poços nas camadas. Portanto, em situações em que os poços produtores são verticais e completados em todas as camadas, a distribuição das taxas de anulação é a mesma em todas as camadas.

5.6 Características do Modelo de Solução

A inclusão da Geoestatística Multiponto, tanto na geração da população inicial quanto nos operadores genéticos, confere ao modelo de solução proposto características que o tornam interessante para ser aplicado ao problema de otimização por trás do ajuste de histórico. Devido ao fato de sempre gerar realizações que respeitam as estatísticas de múltiplos pontos, o *FILTERSIM* funciona como um mecanismo de "poda" do espaço de busca, evitando que o algoritmo genético caminhe em busca de soluções que, mesmo sendo interessantes do ponto de vista do valor da função de avaliação, não estejam de acordo com as características reais da propriedade em estudo. Isso significa que a busca se restringe a uma porção do espaço onde provavelmente estão as soluções mais promissoras.

Essa restrição imposta ao tamanho do espaço de busca também viabiliza a representação cromossômica adotada. Sabe-se que quanto maior a quantidade de genes do cromossomo, maior é o espaço de busca a ser explorado. Contudo,

apesar de a representação adotada implicar na criação de cromossomos com grandes quantidades de genes, na prática o espaço de busca a ser explorado é reduzido em função da restrição imposta pela Geoestatística Multiponto.

Além disso, a Geoestatística Multiponto viabiliza a representação cromossômica adotada, pois a restrição imposta ao espaço de busca também permite que o modelo de solução seja capaz de lidar com cromossomos com grandes quantidades de genes.

Outra característica interessante é a maneira como o *FILTERSIM* completa os pontos não amostrados do mapa de propriedades. Sabendo-se que os valores são provenientes da imagem de treinamento e que essa imagem é construída de forma a não violar as restrições de domínio da propriedade em estudo, tem-se a garantia de que todos os indivíduos gerados durante a evolução são válidos. Isso é bastante atrativo para o problema do ajuste de histórico, pois as avaliações dos indivíduos demandam um alto custo computacional e, por questões de eficiência, é fundamental que o processo evolutivo não avalie indivíduos inválidos.

Além das características apresentadas, vale destacar que a forma como o modelo de solução foi elaborado dá a ele flexibilidade suficiente para ser aplicado à otimização de qualquer propriedade de interesse. Para isso, basta que haja a disponibilidade de uma imagem de treinamento adequada. Contudo, durante uma execução o modelo proposto permite a otimização de apenas uma propriedade. A otimização de mais de uma propriedade na mesma execução é possível desde que se conheceça, de antemão, uma lei (função) que relacione as propriedades envolvidas. Dessa forma, o processo de otimização transcorre normalmente para uma propriedade e as propriedades correlatas são determinadas pela aplicação da função previamente estabelecida.

5.7 Detalhes de Implementação

Para a implementação do modelo de solução proposto foi construído um sistema no ambiente *MS Visual Studio 2008*, usando a linguagem orientada a objetos *C#*. O sistema, denominado *OCTOPUS 2.0*, é um avanço em relação ao sistema *OCTOPUS* (versões *1.x*) que se encontra implantado em alguns dos ativos da Petrobras. O *OCTOPUS* é resultado de uma parceria de pesquisa e desenvolvimento, entre a Petrobras e o Laboratório *ICA* (Inteligência Computacional Aplicada) da *PUC-Rio*, que teve início em 2000. Trata-se de um sistema desenvolvido especificamente para a solução do problema de otimização de planos de drenagem. A partir da descrição de um reservatório (modelo de simulação), o sistema busca encontrar, por meio de um algoritmo genético,

uma configuração de poços que maximize o *VPL* (Valor Presente Líquido) do projeto. No processo de busca pela melhor configuração são levados em consideração a quantidade, o tipo (injetor ou produtor), a trajetória (vertical, horizontal ou direcional) e a localização dos poços. O sistema *OCTOPUS* foi inspirado em [70] e [71] e teve a sua versão preliminar disponível para testes no *CENPES* (Centro de Pesquisas e Desenvolvimento da Petrobras) em 2007. Em 2008 foi disponibilizada a primeira versão do sistema para uso nos ativos da empresa. A partir de então, o sistema vem sendo aperfeiçoado e modificado de acordo com as necessidades. Em [72] são apresentados alguns resultados importantes obtidos com a aplicação do *OCTOPUS* na otimização de planos de drenagem em casos reais da Petrobras.

Neste trabalho, iniciou-se o desenvolvimento do *OCTOPUS 2.0*, que representa uma mudança de conceito em relação às versões *1.x*. A partir da versão *2.0*, o *OCTOPUS* deixa de ser um sistema específico para a otimização de planos de drenagem e se torna um ambiente integrado para a gerência de reservatórios. Segundo esse novo conceito, o sistema permite, a qualquer tempo, o acoplamento de novas funcionalidades que se mostrem úteis para as atividades relacionadas à gerência de reservatórios. Para a construção do novo sistema, adotou-se uma arquitetura modular, a fim de tornar mais simples as tarefas futuras de extensão e de manutenção. A arquitetura do sistema está ilustrada no diagrama da Figura 5.9 e nas seções seguintes são apresentados os detalhes dos seus módulos.

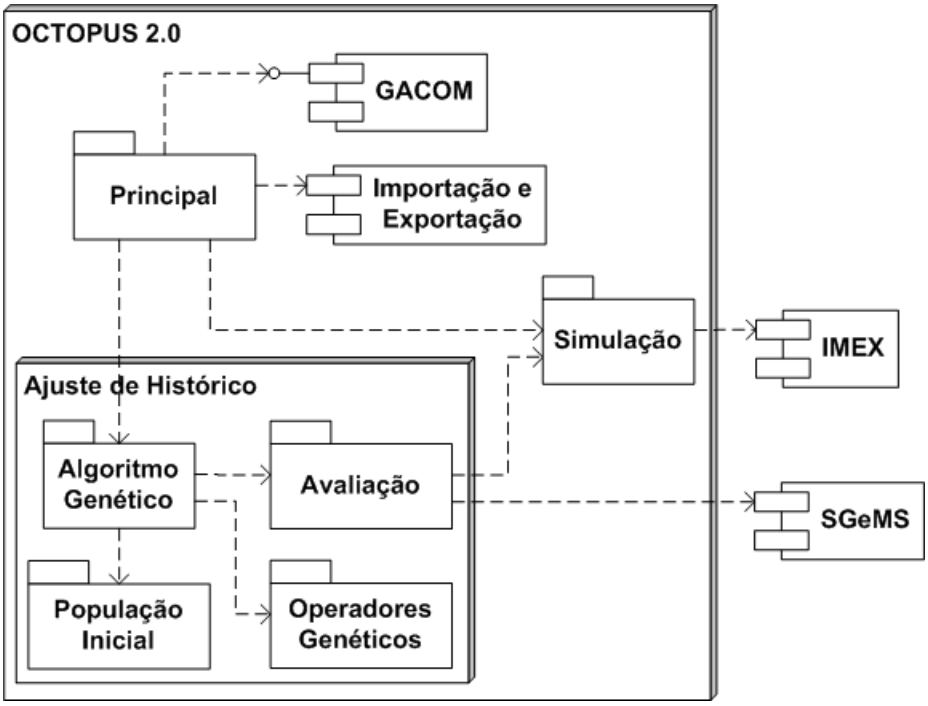

Figura 5.9: Arquitetura do sistema *OCTOPUS 2.0*

5.7.1 Módulo Principal

O módulo *Principal* é responsável pela apresentação, pela persistência e recuperação de dados, e também pelo fornecimento de outros serviços que são necessários ao funcionamento dos demais módulos que compõem o sistema. Na construção desse módulo foi utilizado o padrão de arquitetura *MVC* (Modelo-Visão-Controle) que permite dividir a implementação em 3 camadas. A camada de visão contém todos os elementos de interface com o usuário, enquanto a camada de modelo contém toda a lógica de negócio. A camada de controle é responsável pela gerência da comunicação entre o modelo e a visão e estabelece uma independência entre essas duas camadas. A separação entre visão e modelo permite a realização de modificações na interface, ou até mesmo a substituição do modo de apresentação (gráfico, texto, ou *web*), sem que a lógica de negócio precise ser alterada.

No módulo *Principal* foi destinada uma atenção maior à implementação das camadas de modelo e de controle. A implementação da camada de modelo exigiu, inicialmente, uma reformulação do modelo de dados do sistema *OCTOPUS 1.x*, a fim de que se tornasse suficientemente genérico para atender às necessidades de persistência e recuperação de dados de todas as funcionalidades previstas para o *OCTOPUS 2.0*. A partir do modelo de dados, foi gerado um novo banco de dados e, em seguida, foram implementadas todas as classes das entidades de negócio e suas respectivas classes de acesso e manipulação do banco. Logo depois foram implementadas as classes de controle referentes às classes de entidades. Para a camada de visão, foram implementados apenas os elementos de interface estritamente necessários para a realização dos testes com o modelo de solução proposto neste trabalho.

Além da integração com o banco de dados, o módulo *Principal* também está integrado ao componente de *Importação e Exportação* de modelos. Trata-se de uma *DLL* (*Dynamic Link Library*) que fornece métodos capazes de acessar e percorrer os arquivos textos que descrevem o modelo do reservatório (extensões *.dat* e *.inc*) e recuperar deles os dados do reservatório que são relevantes para o sistema. Os dados recuperados são endereçados aos respectivos objetos de entidades para que estejam disponíveis a todas as funcionalidades do sistema. A principal característica oferecida por esse componente é a capacidade de importar e exportar dados de arquivos de modelos pertencentes a diferentes tipos de simuladores de reservatórios, uma vez que, para cada tipo de simulador, o arquivo que descreve o modelo é escrito segundo uma sintaxe própria. Assim, a origem dos dados do modelo do reservatório se torna transparente para a camada de modelo, que se encarrega apenas de manipular os

dados que são retornados pelo componente.

Ao módulo *Principal* também está acoplado o componente *GACOM* ([73]) que é desenvolvido e mantido pelo Laboratório *ICA*. O *GACOM* também se apresenta na forma de uma *DLL* e fornece todos os métodos necessários para a construção de modelos de soluções baseados em Algoritmos Genéticos. O *GACOM* é constituído de vários módulos que podem ser reutilizados e agrupados conforme as necessidades do problema para o qual se deseja construir um modelo de solução. Além disso, essa organização modular permite também a prototipação fácil e rápida de novas soluções. Apesar de ser um componente com código fechado, o *GACOM* disponibiliza um conjunto de interfaces que possibilita a sua extensão por meio do acoplamento de módulos específicos que sejam necessários para o desenvolvimento de um determinado modelo de solução. Assim, qualquer funcionalidade que seja acrescentada ao sistema, e que necessite de um algoritmo genético, pode instanciá-lo a partir do módulo *Principal*, configurá-lo e estendê-lo conforme as necessidades. Exemplos diretos dessas funcionalidades são as de alocação de poços e de ajuste de histórico.

5.7.2 Módulo de Simulação

Assim como o módulo *Principal*, o módulo de *Simulação* é de extrema importância para as funcionalidades que venham a ser incluídas no sistema. Ele é responsável por receber, de algum módulo, o pedido de simulação de uma ou mais instâncias de um modelo de reservatório e encaminhar essas instâncias ao simulador. Após o término das simulações, o módulo se encarrega também de disponibilizar ao módulo solicitante os arquivos resultantes das simulações. Vale destacar que o módulo de *Simulação* está disponível também para o módulo *Principal*, que tem autonomia para submeter um pedido de simulação sem que haja a solicitação de alguma funcionalidade do sistema.

Apesar de o componente de *Importação e Exportação* permitir a manipulação de modelos de diferentes tipos de simuladores, atualmente, é possível simular apenas modelos escritos para o componente *IMEX* (*IMplicit-EXplicit Black Oil Simulator*), que está acoplado ao módulo de *Simulação*. O *IMEX* ([74]), desenvolvido e comercializado pela *CMG* (*Computer Modelling Group*), é um simulador de fluxo que, a partir do modelo do reservatório, gera os dados de produção de fluidos que são necessários para o cálculo da função de avaliação durante o processo evolutivo do algoritmo genético. O *IMEX* é parte de um pacote de aplicações desenvolvidas pela *CMG* que são voltadas para a modelagem e simulação de reservatórios. Apesar de o pacote apresentar uma interface

gráfica própria, ele permite o acesso a suas aplicações através de chamadas por linha de comando, o que torna a sua integração com o módulo de *Simulação* razoavelmente simples. Para isso, ao receber um pedido de simulação, o módulo executa um comando semelhante a:

mx200711 -f "..\Property_0\modelo_1.dat" -wd "..\Property_0"

Onde:

- *mx200711* é o arquivo executável do simulador *IMEX*;
- o parâmetro após o *-f* é o caminho completo do arquivo (extensão *.dat*) que descreve o modelo de simulação do reservatório;
- o parâmetro após o *-wd* é o diretório em que serão armazenados os arquivos (extensões *.irf*, *.mrf* e *.out*) resultantes da simulação.

5.7.3 Módulo de Ajuste de Histórico

O módulo de *Ajuste de Histórico* corresponde ao modelo de solução proposto neste trabalho. Internamente esse módulo é composto por módulos que utilizam e estendem o componente *GACOM*, a fim de contemplar as particularidades do modelo de solução. Os detalhes sobre esses módulos são apresentados nas seções seguintes.

Módulo de Algoritmo Genético

O módulo *Algoritmo Genético* é responsável por toda a configuração do algoritmo genético a ser aplicado à solução do problema do ajuste de histórico. Essa configuração compreende não só a definição dos valores dos parâmetros de entrada do algoritmo como, por exemplo, tamanho da população, número de gerações e taxas de *steady state*, cruzamento e mutação, mas também a definição dos módulos do *GACOM* que devem ser utilizados na montagem do algoritmo genético. Nesse processo de montagem são estabelecidos, por exemplo, os módulos de geração da população inicial, de avaliação dos indivíduos e também dos operadores de cruzamento e mutação. O *GACOM* dispõe de um conjunto de opções para esses módulos e, geralmente, essas opções são suficientes para a implementação de soluções para uma grande quantidade de problemas de otimização. Contudo, no caso do modelo proposto, foi necessário estender o *GACOM* a fim de acoplar módulos específicos para atender às necessidades do problema em questão.

Módulos de População Inicial e Operadores Genéticos

Conforme descrito nas Seções 5.4 e 5.5 e ilustrado na Figura 5.1, conceitualmente, o algoritmo *FILTERSIM* está associado à geração da população inicial e aos operadores genéticos. Todavia, na construção do modelo de solução, o conceito foi implementado de forma a valorizar a eficiência e a organização do código. Por isso, em vez de efetuar as chamadas ao *FILTERSIM* a partir dos módulos de *População Inicial* e de *Operadores Genéticos*, esses módulos, após o processo de anulação, enviam seus indivíduos ao módulo de *Avaliação*. Assim, o módulo de *Avaliação* se encarrega de reunir todos os indivíduos que precisam ser reconstruídos e efetuar as chamadas necessárias ao *FILTERSIM* para a reconstrução desses indivíduos.

Para exemplificar o processo de construção da população inicial, no passo 1 da Figura 5.10 é apresentado um modelo hipotético a ser otimizado e, no passo 2, um indivíduo que foi gerado após a aplicação de uma taxa de anulação de 50% ao modelo inicial. Nota-se que os blocos nulos estão distribuídos de maneira bastante homogênea ao longo do indivíduo. Essa característica dá ao *FILTERSIM* a flexibilidade suficiente para criar indivíduos que sejam diferentes do caso base, mas que continuem respeitando as estatísticas de múltiplos pontos.

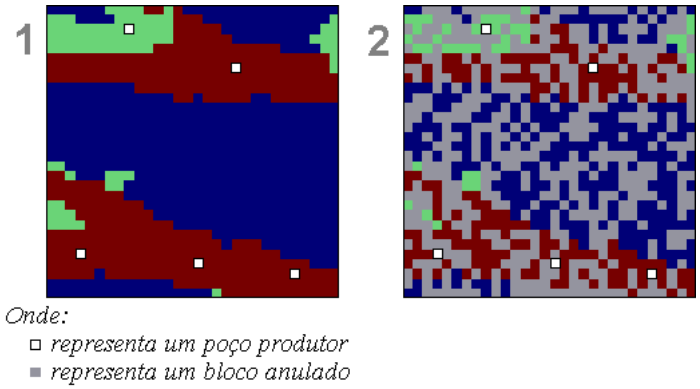

Figura 5.10: Exemplo da geração de um indivíduo da população inicial

Na Figura 5.11 é apresentado um exemplo de cruzamento entre dois indivíduos durante o processo evolutivo. No passo 1 estão destacadas as seções que devem ser trocadas e, no passo 2, são apresentados os indivíduos após as trocas das seções. No passo 2 destacam-se ainda algumas descontinuidades que podem surgir nos indivíduos após as trocas das seções. A ocorrência de descontinuidades pode comprometer a qualidade dos indivíduos gerados, uma vez que esses novos indivíduos podem não respeitar as estatísticas de múltiplos pontos. Para evitar essas descontinuidades, no passo 3 efetua-se a anulação de

parte das bordas das seções trocadas. Isso dá ao *FILTERSIM* a possibilidade de reconstruir esses indivíduos de tal maneira que essas descontinuidades sejam suavizadas.

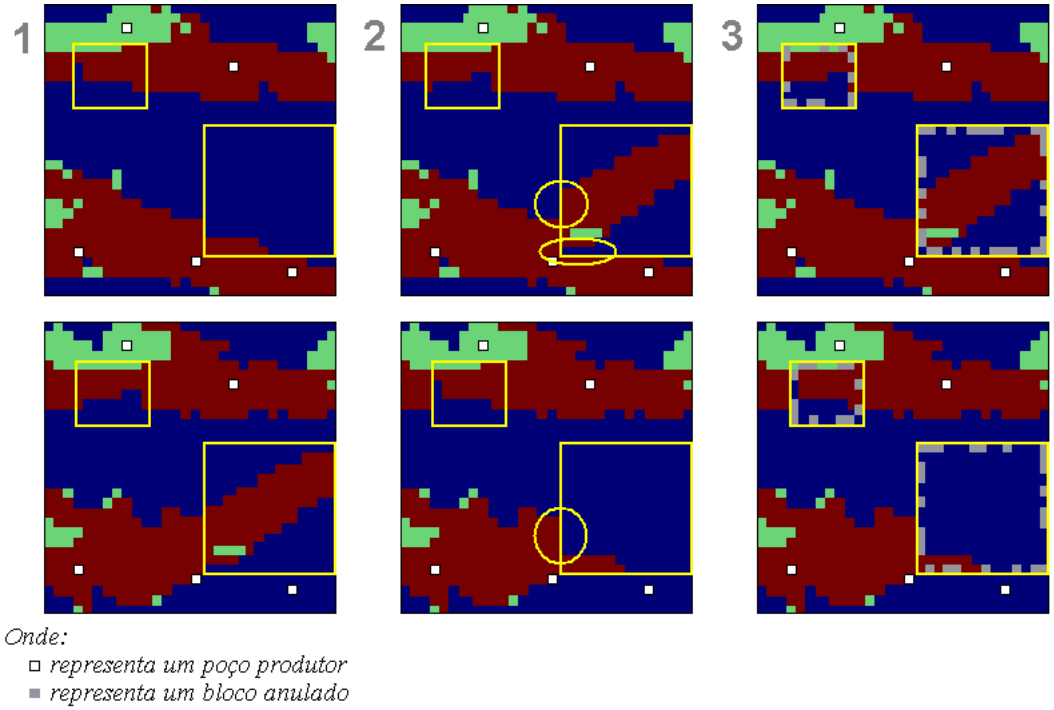

Figura 5.11: Exemplo da aplicação do operador de cruzamento

O passo 1 da Figura 5.12 apresenta um exemplo de um indivíduo sobre o qual é aplicado o operador de mutação. O passo 2 mostra a distribuição das taxas de anulação calculadas para todos os poços produtores. Nota-se que as maiores taxas de anulação estão concentradas em torno de apenas dois, dentre os 5 poços produtores existentes. Com base nessa distribuição, observa-se no passo 3 a concentração de blocos nulos em torno dos poços que apresentam os maiores erros de ajuste.

Nos exemplos apresentados, observa-se que os procedimentos de geração da população inicial e de aplicação dos operadores genéticos são finalizados após o processo de anulação. Conforme o critério adotado na implementação, a reconstrução desses indivíduos é de responsabilidade do módulo de *Avaliação*.

Módulo de Avaliação

A cada geração do algoritmo genético, todos os indivíduos da população são enviados ao módulo de *Avaliação*. Dentre esses indivíduos há um percentual que não precisa ser avaliado. Esses indivíduos correspondem àqueles que são transferidos de uma geração para outra através de *steady state*. O módulo se encarrega de identificar esses indivíduos e evitar que sejam desnecessariamente

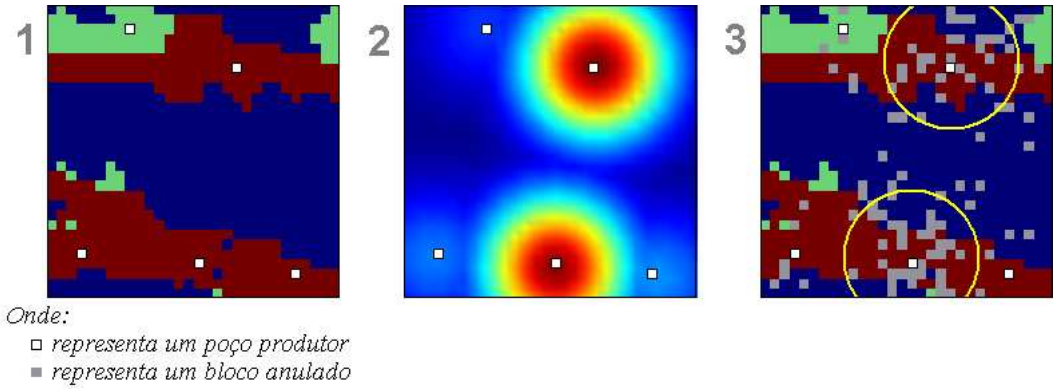

Figura 5.12: Exemplo da aplicação do operador de mutação

avaliados. Essa medida é fundamental para a diminuição do custo computacional da otimização, uma vez que a etapa de avaliação representa o gargalo de todo o processo.

Para que a Equação 5-1 seja efetivamente utilizada para avaliar os indivíduos ao longo do processo evolutivo, é necessária a interação do módulo de *Avaliação* com o módulo de *Simulação* e com o componente *SGeMS* (*Stanford Geostatistical Modeling Software*). O *SGeMS* ([75], [76]) é um sistema de código aberto, desenvolvido pela *Stanford University* (*EUA*), que fornece diversos métodos de modelagem geoestatística, dentre eles o *FILTERSIM*, que é utilizado na reconstrução dos indivíduos ao longo do processo de otimização. Semelhantemente ao pacote de aplicações da *CMG*, o *SGeMS* também apresenta uma interface gráfica própria e também possibilita o acesso aos seus métodos por linha de comando, porém, a integração com o módulo de *Avaliação* não é tão simples.

A primeira etapa do processo de integração consiste em, antes da execução do *OCTOPUS 2.0*, criar um projeto no *SGeMS* a partir de sua própria interface gráfica. Nesse projeto é necessário definir os elementos "grade", "dados condicionantes" e "imagem de treinamento" do modelo a ser otimizado. A criação da grade consiste em instanciar um elemento do tipo *cartesian grid* que define os limites do modelo de reservatório, ou seja, estabelece as quantidades de blocos nas direções x, y e z.

Para os dados condicionantes é necessário instanciar um elemento do tipo *set point* que preenche toda a grade com pontos (blocos) nulos. Para isso, é necessário importar os pontos nulos a partir de um arquivo texto (extensão *.gslib*) específico para o *SGeMS*, cujo *layout* é apresentado na Seção A.1. A cada simulação, o conjunto de pontos nulos é substituído pelos pontos do indivíduo a ser reconstruído, de maneira que permanecem nulos apenas aqueles pontos que também são nulos no indivíduo. Os pontos preenchidos são utilizados

pelo *FILTERSIM* como dados condicionantes para a estimativa dos valores da propriedade de interesse nos pontos nulos.

Por fim, é necessário instanciar outro elemento do tipo *cartesian grid* que define a imagem de treinamento da propriedade de interesse. Nesse caso, a imagem de treinamento é definida por um conjunto de valores que também são importados a partir de um arquivo *.gslib*. Porém, há diferenças entre os *layouts* dos arquivos *.gslib* utilizados para instanciar elementos do tipo *cartesian grid* e *set point*. O *layout* do arquivo *.gslib* para *cartesian grid* é apresentado na Seção A.2.

A Figura 5.13 exibe um fragmento de tela do *SGeMS* em que são mostrados os pontos nulos iniciais (*conditional_data*) e a grade (*grid*) em torno dos pontos para um modelo de reservatório hipotético. Na Figura 5.14, o mesmo fragmento de tela mostra um exemplo de uma imagem de treinamento (*training image*).

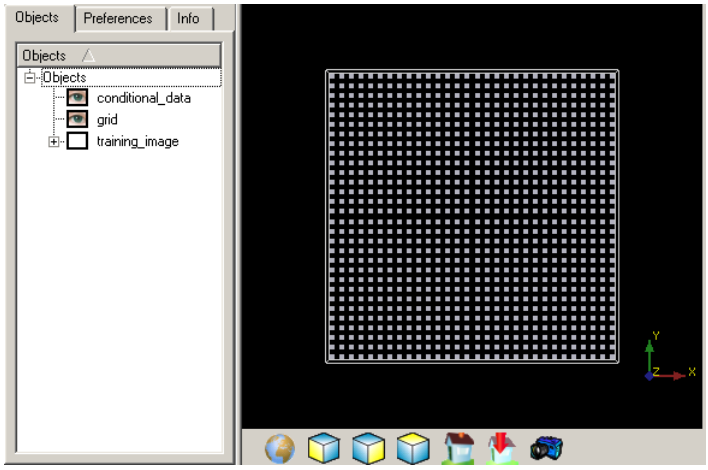

Figura 5.13: Grade e dados condicionantes nulos criados no *SGeMS*

A etapa seguinte da integração consiste em estabelecer um modo de comunicação entre o módulo de *Avaliação* e o *SGeMS*. Para isso, dado um indivíduo, inicialmente cria-se um diretório para servir de repositório de todos os arquivos necessários para essa comunicação. Em seguida, cria-se um arquivo de lote (extensão *.bat*) cujo conteúdo é semelhante a:

RunScript "..\Property_0\filtersim.py"

Onde:

- o parâmetro após o *RunScript* é o caminho completo de um *script* escrito em linguagem *Python* (extensão *.py*), cujo propósito é apresentado mais adiante.

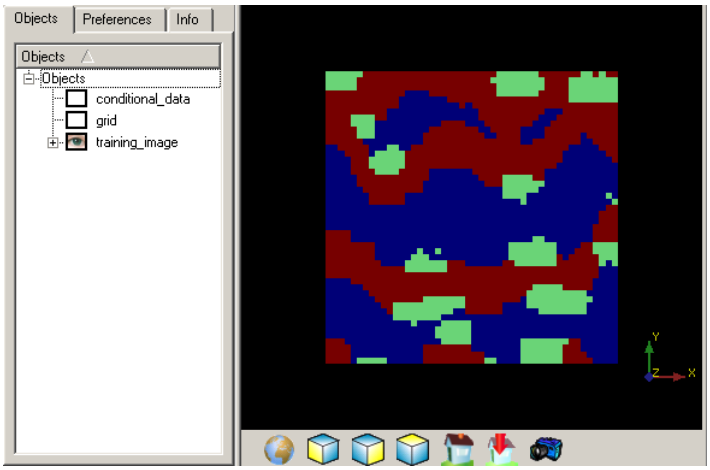

Figura 5.14: Imagem de treinamento criada no *SGeMS*

Após a criação do arquivo de lote, os valores dos genes que compõem o cromossomo são armazenados em um arquivo *.gslib* segundo o *layout* da Seção A.1. O arquivo *.gslib* criado contém a distribuição dos valores da propriedade de interesse ao longo do reservatório e, dentre esses valores, alguns são nulos e precisam ser determinados pelo *FILTERSIM*. Os valores não nulos, obviamente, se mantêm inalterados. O conteúdo desse arquivo deve ser importado pelo *SGeMS* como um elemento do tipo *set point* e substituir os pontos nulos (*conditional_data*) criados inicialmente. A importação como *set point* é uma condição imposta pelo *SGeMS* para que os dados possam ser utilizados como condicionantes no processo de simulação com o *FILTERSIM*.

Em seguida cria-se o *script* em *Python* com as instruções e os parâmetros necessários à execução do *FILTERSIM*. Os detalhes do conteúdo do *script* são apresentados na Seção A.3. Após a criação do *script* efetua-se a chamada ao *SGeMS* por meio da execução do arquivo de lote criado anteriormente. Ao final da execução do *FILTERSIM* é retornado um arquivo *.gslib*, com o *layout* da Seção A.2, em que todos os valores da propriedade de interesse são não nulos.

A partir do arquivo *.gslib* de saída, atualiza-se o cromossomo preenchendo todos os genes nulos com os valores calculados pelo *FILTERSIM*. Após a atualização, gera-se, a partir do cromossomo, um arquivo texto (extensão *.inc*) com os valores da propriedade de interesse em todos os blocos do modelo do reservatório. Esse arquivo *.inc* é chamado pelo arquivo *.dat* quando o simulador *IMEX* é executado.

Todos esses passos, desde a criação do diretório até a geração do arquivo *.inc* para o *IMEX*, devem ser realizados para todos os indivíduos da população que precisam ser avaliados. A criação do arquivo *.inc* representa o término do processo de reconstrução de um indivíduo enviado ao módulo de *Avaliação*.

Assim, para os exemplos apresentados nas Figuras 5.10, 5.11 e 5.12, os indivíduos resultantes da reconstrução são apresentados nas Figuras 5.15, 5.16 e 5.17, respectivamente. Em particular, no caso da Figura 5.16, observa-se que o *FILTERSIM* foi capaz de reduzir a descontinuidade provocada pela aplicação do operador de cruzamento aos indivíduos.

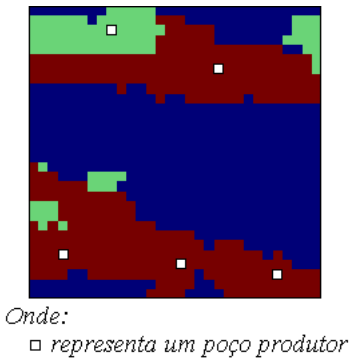

Figura 5.15: Resultado da reconstrução do indivíduo da população inicial

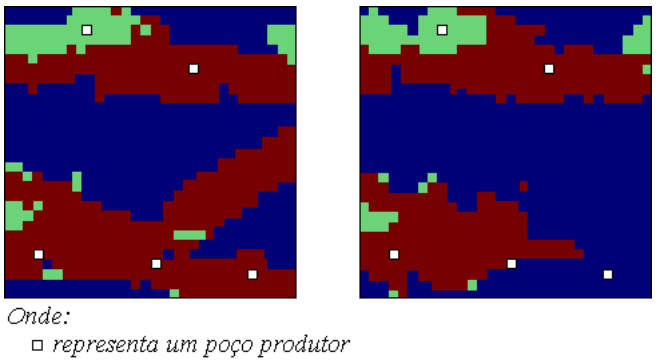

Figura 5.16: Resultado da reconstrução dos indivíduos submetidos ao operador de cruzamento

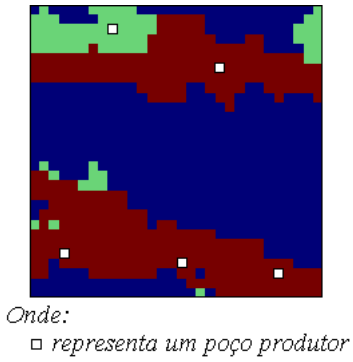

Figura 5.17: Resultado da reconstrução do indivíduo submetido ao operador de mutação

Após a criação dos arquivos *.inc* de todos os indivíduos, o módulo de *Simulação* é acionado para que simule o modelo do reservatório em cada diretório, considerando o seu respectivo arquivo *.inc*. Cada simulação do *IMEX* retorna um conjunto de arquivos e, dentre eles, se encontra um arquivo texto (extensão *.out*) com os dados de produção de fluidos da simulação. A partir do arquivo *.out* são extraídos todos os dados necessários para o cálculo da função objetivo e, finalmente, aplica-se a Equação 5-1 a esses dados para se obter a avaliação do indivíduo.

Terminada a avaliação de todos os indivíduos, seus respectivos diretórios (juntamente com seus arquivos) são apagados. A remoção dos arquivos entre uma geração e outra é importante devido ao fato de alguns deles, especialmente os arquivos com extensões *.out* e *.mrf*, ocuparem muito espaço em disco. Assim, o armazenamento de todos os arquivos gerados por todas as simulações efetuadas ao longo do processo evolutivo pode ser inviável.

Antes da remoção dos diretórios, porém, toma-se o cuidado de sempre armazenar, em outro diretório, os arquivos do melhor indivíduo encontrado até aquele momento. Dessa forma, a cada nova geração, se o melhor indivíduo da geração atual é melhor que o melhor indivíduo encontrado até o momento, seus arquivos são substituídos. Caso contrário, os arquivos anteriores são mantidos. Portanto, ao final do processo evolutivo sempre se tem o melhor indivíduo e seus respectivos arquivos resultantes da simulação.

A Figura 5.18 apresenta um fluxograma que resume todo o procedimento de avaliação. O passo de simulação dos indivíduos corresponde à chamada do módulo de *Simulação* e o passo de remoção dos diretórios se encarrega também de atualizar o diretório que armazena os arquivos do melhor indivíduo.

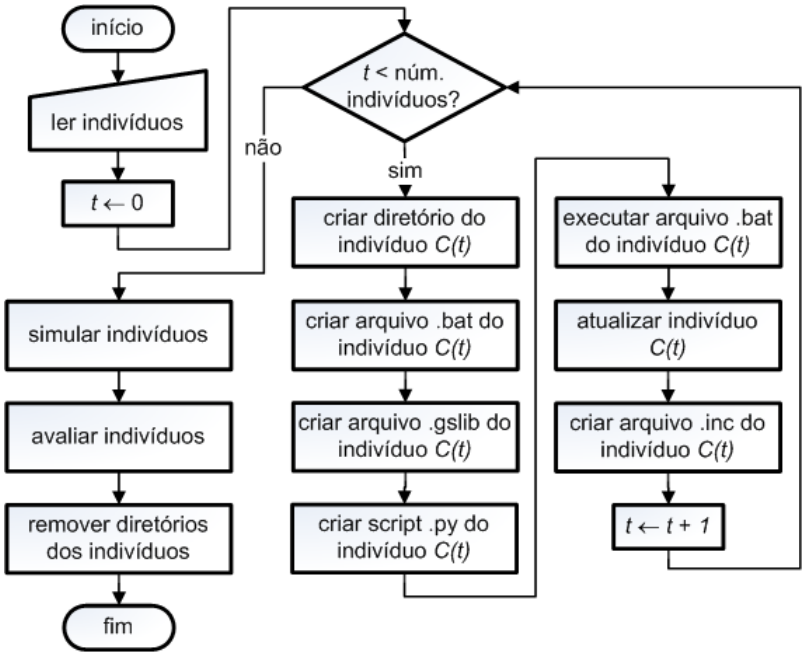

Figura 5.18: Fluxograma do módulo de avaliação