

Otimização de alternativas para o desenvolvimento de campos de petróleo

Luciana Faletti Almeida

Yván Túpac Valdivia

Marley Maria Bernardes Rebuszi Vellasco

Marco Aurélio Cavalcanti Pacheco



Resumo

Este artigo apresenta um sistema de suporte à decisão para a otimização de alternativas de desenvolvimento de campos de petróleo. O desenvolvimento de um campo de petróleo consiste na escolha de uma alternativa para a exploração de um reservatório petrolífero, conhecido e delimitado, que permita produzir a maior quantidade de hidrocarboneto possível, dentro dos limites físicos e econômicos existentes, isto é, maximizar o valor presente líquido (VPL). O VPL é calculado a partir da produção de óleo, a qual é obtida com o uso de um simulador de reservatório. Cada execução do simulador pode demorar desde alguns segundos até várias horas, dependendo da complexidade do reservatório modelado. Isto reduz o número de alternativas que podem ser geradas e avaliadas pelo especialista na busca da melhor solução. Deste modo, o objetivo deste trabalho é propor e avaliar um sistema inteligente de otimização que emprega: algoritmos genéticos (AGs), algoritmos culturais (ACs) e coevolução para a busca de uma alternativa de desenvolvimento ótima; e um ambiente de computação paralela para a simulação de reservatório e cálculo do VPL das alternativas. O sistema resultante permite que o especialista obtenha, em tempo hábil, a alternativa ótima (ou quase-ótima) para o desenvol-

vimento de um campo de petróleo conhecido. Os resultados obtidos nos estudos de casos apresentados demonstram que o sistema proposto, baseado em técnicas inteligentes, obtém boas alternativas de desenvolvimento de campos petrolíferos com uma grande redução do tempo computacional, redução esta obtida a partir do aproveitamento do poder computacional de um ambiente de computação paralela e do aproveitamento de conhecimento do especialista, por meio das sementes iniciais.

Palavras-chave: Algoritmo genético. Algoritmos culturais. Coevolução. Otimização. Desenvolvimento de campos petrolíferos. Engenharia de reservatório. Algoritmos genético distribuídos. Computação paralela.

1 Introdução

Na engenharia de reservatórios, uma das primeiras tarefas a serem realizadas é o desenvolvimento de uma estratégia para alcançar a produção da maior quantidade de hidrocarboneto possível, dentro dos limites físicos e econômicos existentes (BITTENCOURT, 1997; BITTENCOURT; HORNE, 1997; GUYAGULER et al., 2000; GUYAGULER; HORNE, 2000). A determinação dessa estratégia é tratada como um problema de otimização de alternativas para o desenvolvimento de um campo de petróleo, o qual envolve a análise do sistema de exploração (determinação da disposição dos poços de petróleo e da plataforma) e um reservatório geológico a ser explorado.

Freqüentemente a solução operacional para o desenvolvimento de um campo de petróleo não é a ótima, mas

apenas um resultado viável que satisfaz as restrições operacionais e econômicas em um momento específico. Isto se deve ao grande número de variáveis que um engenheiro de reservatório deve considerar de forma a alcançar a melhor estratégia para o desenvolvimento do campo. Atualmente o processo de escolha do valor de cada uma dessas variáveis é feito de forma manual.

Deste modo, um processo de otimização que permita avaliar diferentes perfis de produção, para diferentes configurações, de forma a determinar a alternativa ótima do ponto de vista econômico sob as restrições técnicas existentes, torna-se bastante importante. Trabalhos que abordam processos de otimização, para esse problema, são encontrados na literatura (BITTENCOURT, 1997;

BITTENCOURT; HORNE, 1997; GUYAGULER et al., 2000; GUYAGULER; HORNE, 2000; TÚPAC et al., 2002; BECKNER; SONG, 1995; PEDROSO; SCHIOZER, 2000). Entretanto, a maioria destes trabalhos restringem-se a otimizar apenas um tipo de poço, ou a localização de poços convencionais pré-determinados, ou ainda usam métodos de otimização clássicos que, para se tornarem viáveis, restringem-se a modelos analíticos de reservatório bastante simplificados. Para se aplicar um processo mais completo de otimização nesse problema, é necessário definir uma metodologia de otimização apropriada e uma função objetivo a ser maximizada ou minimizada.

Este artigo apresenta um sistema de suporte à decisão para a otimização de alternativas de desenvolvimento de campos de petróleo. O sistema emprega algoritmos evolucionários na busca de uma alternativa de desenvolvimento ótima, em um ambiente de computação paralela, que forneça ao especialista, em tempo hábil, a melhor configuração de poços para o desenvolvimento do campo de petróleo conhecido.

Neste trabalho, utilizaram-se algoritmos genéticos, algoritmos culturais e coevolução para encontrar uma alternativa, a qual é definida pela quantidade, localização, tipo (vertical/horizontal) e função (injetor/produzidor) de poços. Dentre as técnicas de computação, esses algoritmos são métodos de busca e otimização utilizados para procurar soluções de problemas complexos ou com espaço de soluções muito grande, problemas estes de difícil modelagem e solução quando se aplicam métodos de otimização convencionais.

A função objetivo a ser maximizada neste problema é o valor presente líquido (VPL) (TÚPAC et al., 2002). O valor presente líquido é uma função do fluxo de caixa esperado, obtido a partir do custo de investimento inicial, do preço do petróleo, da produção de óleo, dos custos de operação, das alíquotas de impostos e dos *royalties* pagos durante o tempo de produção. Devido à complexidade do perfil de produção de petróleo, é necessária a utilização de simuladores de reservatório para obtenção das curvas de produção de óleo e gás. Deste modo, o simulador de reservatórios é parte integrante da função de avaliação. Entretanto, o uso de simuladores eleva muito o custo computacional, fazendo com que a obtenção do resultado seja muito demorada. Torna-se importante a implementação de uma técnica que permita contornar esse problema. Neste trabalho, foi utilizado processamento distribuído para aproveitar o poder computacional de vários processadores e realizar várias simulações de produção de óleo e cálculos do VPL em um tempo menor. Com o uso do processamento distribuído e do poder computacional de uma rede de computadores, conseguiu-se reduzir consideravelmente o tempo computacional necessário para realizar a otimização.

Na próxima seção, apresenta-se o sistema de otimização de alternativa proposto neste trabalho, detalhando cada um dos seus módulos. Em seguida, na seção 3, apresenta-se o estudo de caso com seus resultados e discussões. Por fim, as conclusões e os trabalhos futuros são apresentados na seção 4.

2 Sistema de otimização de alternativas

O sistema proposto é composto de três módulos principais: o módulo otimizador, o simulador de reservatórios IMEX (IMEX, 2000) e o módulo de cálculo do valor presente líquido. O módulo otimizador, que contém os algoritmos evolucionários (genético, cultural e coevolução), determina os valores adequados para cada uma das variáveis e parâmetros que configuram uma certa alternativa. Essa alternativa é então submetida ao simulador de reservatórios, o qual fornece as previsões de produção de óleo, gás e água da alternativa. Estas, por sua vez, são utilizadas pelo módulo de cálculo do valor presente líquido (VPL), de forma a obter o valor de avaliação da alternativa. Finalmente, o VPL calculado é enviado para o módulo otimizador que o utiliza como valor associado à avaliação da alternativa (cromossomo). A Figura 1 apresenta a arquitetura do sistema de otimização desenvolvido. As seções seguintes descrevem em detalhes cada módulo separadamente.

2.1 Módulo otimizador

O módulo otimizador é composto pelo algoritmo evolucionário que irá otimizar a estratégia de exploração de um dado reservatório. Neste módulo, foram implementados três modelos de algoritmos evolucionários: algoritmo genético (AG), modelo híbrido AG/algoritmo cultural (AC) e modelo de Coevolução, de forma a avaliar o desempenho de cada algoritmo na solução do problema. A seguir, serão descritos o funcionamento e a modelagem de cada algoritmo evolucionário avaliado.

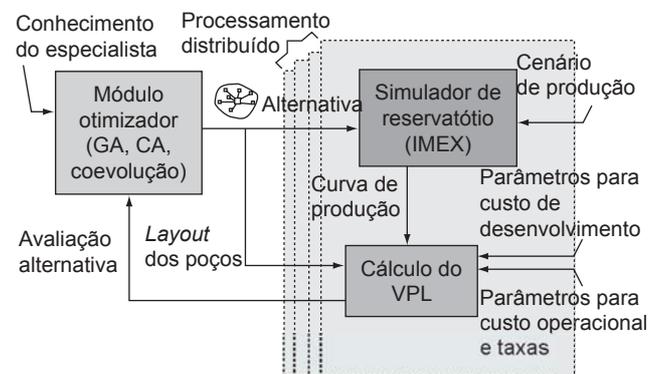


Figura 1. Módulos que compõem o sistema de otimização de alternativas.

2.1.1 Modelagem por algoritmo genético

Algoritmos genéticos (AGs) são algoritmos probabilísticos que fornecem um mecanismo de busca paralela e adaptativa baseado no princípio Darwiniano de sobrevivência dos mais aptos e na reprodução genética (MITCHELL, 1996; GOLDBERG, 1989; BACK, 1996; MICHALEWICZ, 1996). O mecanismo de busca é obtido a partir de uma população de indivíduos (soluções), representados por cromossomos (palavras binárias, vetores, etc.), cada um associado a uma aptidão (avaliação da solução), que são submetidos a um processo de evolução (seleção, reprodução, cruzamento e mutação) por vários ciclos.

2.1.1.1 Representação do cromossomo

A representação do cromossomo na modelagem do AG deve ser capaz de descrever corretamente o espaço de busca relevante do problema. Deste modo, a representação escolhida neste trabalho possui tamanho variável (ZEBULUM et al., 2001), uma vez que não se conhece *a priori* o número de parâmetros a serem otimizados (número de poços da alternativa). Assim, o comprimento do cromossomo é alterado no decorrer da evolução, sendo este comprimento interpretado como o número de genes ativos.

O cromossomo é formado por uma dupla estrutura (Figura 2): uma que contém a descrição da alternativa (cromossomo); e outra que define a ativação dos genes (máscara de ativação). A estrutura do cromossomo é dividida em duas partes: a primeira contendo informações sobre os poços verticais e a segunda sobre os poços horizontais. Cada uma dessas partes é, por sua vez, subdividida em poços injetores e poços produtores. A função da máscara de ativação é exatamente indicar, dependendo de seu valor (0 ou 1), quais genes dos cromossomos estão ativos e quais estão inativos (ALMEIDA, 2003). Cada gene do cromossomo contém informações referentes à localização de um certo poço, sendo que essas informações são representadas por números inteiros.

Para poços verticais, a informação refere-se apenas à posição (x, y) do poço. Já para os poços horizontais, além da posição (x, y), é necessário especificar também a

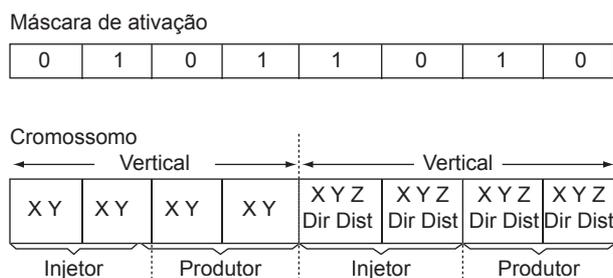


Figura 2. Representação do cromossomo do AG.

profundidade em que o poço se encontra (z); a sua direção horizontal (Dir), que pode ser x ou y; e o tamanho total (comprimento do poço - Dist) na direção horizontal.

A Figura 2 ilustra o cromossomo desenvolvido para este problema.

2.1.1.2 Operadores genéticos

Os operadores mais conhecidos nos AGs são os de cruzamento (*crossover*), que é um operador baseado na troca de partes dos cromossomos selecionados (pais), formando-se duas novas soluções (filhos); Mutação, que é a troca aleatória do valor contido nos genes de um cromossomo por outro valor válido do alfabeto; e Reprodução, que é o processo de selecionar e copiar um determinado cromossomo para a população seguinte de acordo com sua aptidão.

Neste trabalho foram utilizados os seguintes operadores:

- cruzamento aritmético (MICHALEWICZ, 1996); mutação *LocalMove* (ALMEIDA, 2003);
- mutação *FlipWell* (ALMEIDA, 2003);
- reprodução *Steady State* (elitismo) (MICHALEWICZ, 1996);
- mutação uniforme (MICHALEWICZ, 1996);
- mutação *AddWell* (ALMEIDA, 2003); e
- mutação *AddFiveSpotWell* (ALMEIDA, 2003);

Os operadores cruzamento aritmético e mutação uniforme são operadores tradicionais de algoritmos genéticos, descritos em Michalewicz, 1996. Já os operadores *Local Move*, *AddWell*, *FlipWell* e *AddFiveSpotWell*, foram projetados especificamente para este projeto. Todos os operadores são descritos a seguir:

- cruzamento aritmético:** os dois cromossomos escolhidos para sofrer o cruzamento têm todos os seus genes alterados da seguinte maneira: $x'_1 = a * x_1 + (1 - a) * x_2$ e $x'_2 = a * x_2 + (1 - a) * x_1$, em que x'_j é um gene alterado do cromossomo j, e a é um número real aleatório compreendido entre zero e um, garantindo assim que os valores de x'_1 e x'_2 continuem pertencendo ao domínio. Vale ressaltar que, como o cromossomo trabalha com números inteiros, os valores resultantes do cruzamento são truncados, sendo considerada somente a parte inteira.
- mutação uniforme:** o valor do gene i do cromossomo que sofre a mutação é alterado por: $x(i) = left + (right - left) * a$, em que *left* é o limite inferior do domínio do gene; *right* é o limite superior; e a é um número aleatório entre zero e um. Da mesma forma que no caso anterior, como o cromossomo trabalha apenas com números inteiros, o valor resultante da mutação é truncado, considerando somente a parte inteira.
- LocalMove:** o valor do gene do cromossomo sofre um deslocamento de 1 para mais ou para menos,

dependendo do sorteio feito a partir de uma variável aleatória. Essa mutação tem como objetivo fazer pequenos deslocamentos com os poços, explorando assim sua vizinhança.

- d) **AddWell:** atua sobre a máscara do cromossomo somente no sentido de ativação dos genes (mudança $0 \Rightarrow 1$ na máscara de ativação).
- e) **FlipWell:** o operador permite a ativação e desativação de genes de forma oscilatória ao longo das gerações (mudança $0 \Rightarrow 1$ ou $1 \Rightarrow 0$ na máscara de ativação).
- f) **AddFiveSpotWell:** tem como objetivo ativar uma configuração pré-estabelecida de poços conhecida como “*five spot*”. Essa configuração de poços é composta de um poço produtor cercado por quatro poços injetores equidistantes a ele, formando um quadrado. Na opinião de especialistas, esta estrutura costuma dar bons resultados quando inserida em locais adequados. A Figura 3 ilustra uma configuração *five spot*; e
- g) **SteadyState:** troca parcial da população, na qual os M melhores indivíduos da população corrente são copiados para a população seguinte.

2.1.1.3 Validação das alternativas

Durante a evolução do AG podem ser geradas configurações com posições duplicadas de poços verticais e/ou horizontais, além de poços horizontais muito próximos uns dos outros ou mesmo com alguma interseção. Estas configurações são inaceitáveis e, na maioria dos casos, não podem ser simuladas.

Para evitar que essas configurações inválidas sejam enviadas ao Simulador de Reservatórios, é necessário efetuar uma validação das trajetórias de todos os poços gerados pelo AG previamente, antes que o cromossomo seja avaliado. Esta função de validação segue as seguintes heurísticas:

- a) poços horizontais têm prioridades sobre poços verticais: isto significa que caso um poço horizontal e um vertical estejam formando uma configuração inválida (por exemplo, distância entre os poços inferior à mínima permitida), o poço vertical será eliminado

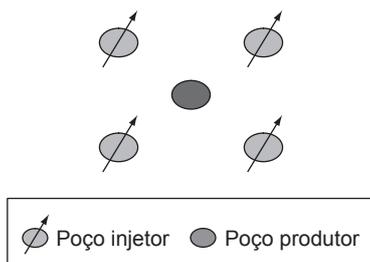


Figura 3. Configuração *Five Spot*.

(por meio da máscara de ativação) e não será considerado na avaliação;

- b) poços produtores têm prioridade sobre poços injetores: neste caso, se poços do mesmo tipo (horizontais ou verticais) estiverem em configuração inválida, o produtor será mantido e o injetor terá seu *bit* de máscara de ativação colocado em zero;
- c) os poços devem respeitar uma distância mínima entre si, isto é, dois poços quaisquer da alternativa não podem estar a uma distância inferior a um raio de distância mínima especificado, raio este geralmente fornecido pelo especialista. Esta restrição, além de evitar poços muito próximos, também evita a superposição de poços (o que significaria valor de distância zero entre dois poços quaisquer);
- d) poços horizontais têm um comprimento máximo, também especificado pelo especialista; e
- e) não podem ser criados poços no local definido como posição da plataforma.

As restrições descritas acima também são usadas nos demais modelos de otimização descritos nas seções 2.1.2 e 2.1.3.

2.1.1.4 Uso de conhecimento prévio

O sistema de otimização de alternativas proposto neste trabalho permite a utilização de conhecimento prévio, ou seja, podem-se aproveitar alternativas fornecidas por especialistas ou alternativas obtidas a partir de experimentos anteriores e efetuar o processo de evolução dos algoritmos genéticos a partir delas, introduzindo estas alternativas na população inicial do novo experimento. Esse conhecimento é chamado de Semente Inicial, a qual é adicionada à primeira população. Isto é de grande importância quando a avaliação tem um alto custo computacional, pois, além de possibilitar a adição do conhecimento do especialista, permite iniciar novos experimentos a partir de resultados obtidos em experimentos anteriores, favorecendo a busca de melhores soluções. Dada a característica elitista dos algoritmos genéticos, a solução resultante não poderá ser pior do que as inseridas como semente inicial. Assim, a influência da semente inicial no resultado da otimização vai depender diretamente de sua qualidade como solução, ou seja, se a semente inicial for a solução ótima ou quase-ótima, o algoritmo dificilmente será capaz de encontrar uma melhor resposta; já se a semente estiver longe das soluções quase-ótimas, rapidamente será substituída por outras soluções obtidas na evolução

2.1.2 Modelagem híbrida algoritmos genéticos - algoritmos culturais

Algoritmos Culturais (ACs) (REYNOLDS, 1994; REYNOLDS; CHUNG, 1996a; REYNOLDS; CHUNG, 1996b) consistem em três componentes básicos (REYNOLDS; CHUNG, 1996a): a população a ser

evoluída, juntamente com seus operadores e mecanismo de avaliação; o espaço de crença, o qual representa a experiência que está sendo adquirida pela população durante o processo de solução do problema; e o protocolo de comunicação que determina a interação entre a população e o espaço de crença.

No modelo utilizado neste estudo, a componente população do AC é modelada pelo AG descrito na seção 2.1.1.

O Espaço de Crenças é composto por padrões que representam as características dos melhores indivíduos da população. O espaço de crença se comporta como um cromossomo que, ao invés de ter um valor para cada informação de cada gene, possui um ou mais intervalos de valores (ALMEIDA, 2003). Estes intervalos influenciam a avaliação dos indivíduos da população: caso algum indivíduo tenha o valor de um gene dentro de um dos intervalos presentes no espaço de crenças, este indivíduo ganha um bônus na sua avaliação.

O protocolo de comunicação é composto por dois processos:

- o processo de votação:** escolhe os melhores indivíduos de uma geração do AG para que estes influenciem o espaço de crença. Os melhores indivíduos são escolhidos em termos de porcentagem, e o parâmetro $pBest$ representa o percentual dos melhores indivíduos da população que irão influenciar o Espaço de Crença; e
- o processo de promoção:** determina como o espaço de crenças é capaz de influenciar um componente da população. Essa influência ocorre pela alteração das avaliações dos indivíduos da população. Essa alteração é feita com o uso de um parâmetro chamado bônus, o qual possui um valor que influencia a avaliação do cromossomo no sentido de melhorá-la.

A influência dos melhores indivíduos no espaço de crença é feita por meio do uso de operadores (REYNOLDS; CHUNG, 1996b) descritos abaixo. Um exemplo ilustrativo sobre cada um dos operadores descritos abaixo é apresentado na Figura 4.

- especialização:** Se os genes pertencentes a algum intervalo que compõe o espaço de crença estão concentrados em algum ponto deste intervalo, este intervalo deve ser reduzido.

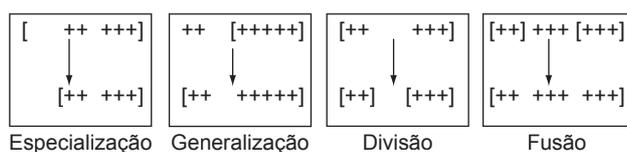


Figura 4. Operadores de atualização do Espaço de Crenças.

- generalização:** Se há uma evidência de indivíduos aceitáveis fora do intervalo atual, então o intervalo deve ser ampliado para incluir os novos indivíduos.
- divisão:** Este é um caso particular de especialização, no qual os genes pertencentes a um determinado intervalo estão concentrados em mais de um ponto no interior deste intervalo; e
- fusão:** Este é um caso especial de generalização em que existem genes na vizinhança de dois intervalos bastante próximos e, por isso, ambos os intervalos devem ser fundidos.

2.1.3 Modelagem com coevolução

Na modelagem por coevolução, tem-se a decomposição do problema em submódulos, de forma que se possa ter a evolução de subcomponentes interdependentes, em que cada um deles pode evoluir independentemente dos demais. Essa interdependência é normalmente complexa, isto é, o efeito de mudar um subcomponente provoca uma “deformação” no espaço de buscas dos outros subcomponentes que estão acoplados por esta interdependência. (DURHAM, 1991; POTTER; DE JONG, 2000).

Neste artigo, foi utilizado um modelo coevolucionário cooperativo genérico (DURHAM, 1991). Nesta arquitetura, duas ou mais espécies diferentes formam um ecossistema. Como na natureza, as espécies são geneticamente isoladas, ou seja, os indivíduos só podem se reproduzir com outros indivíduos da mesma espécie. Isto é feito simplesmente isolando-se as espécies em populações separadas. As espécies somente interagem umas com as outras por um domínio compartilhado e têm uma relação apenas de cooperação. Um diagrama do modelo coevolucionário cooperativo proposto para o problema de otimização de alternativas é mostrado na Figura 5.

O modelo de coevolução utilizado neste estudo possui duas populações a serem evoluídas: uma composta pelos poços horizontais e outra pelos poços verticais. Na Figura 5, mostra-se a população de poços verticais sendo avaliada, utilizando a população de poços horizontais como colaboradora. A escolha destas duas populações deveu-se às diferenças das capacidades de obtenção de

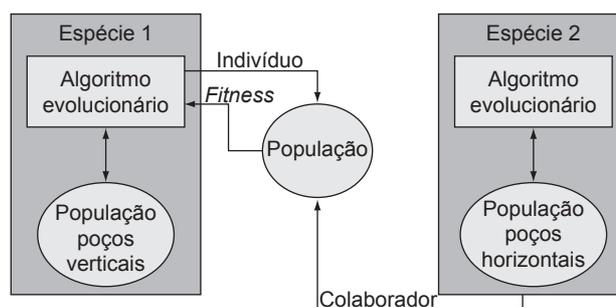


Figura 5. Modelo coevolucionário cooperativo.

óleo e custo de perfuração e manutenção entre poços verticais e horizontais.

A Figura 6 ilustra os cromossomos referentes às duas populações da coevolução. O significado de cada parâmetro nos genes dos dois cromossomos é o mesmo descrito na seção 2.1.1. Da mesma forma, os operadores utilizados aqui são os mesmos descritos na seção 2.1.1.

2.2 Módulo simulador

Para se obterem os perfis de produção de óleo, gás e água a serem utilizados na avaliação de cada alternativa, utiliza-se o Simulador (IMEX : *Advanced Oil/ Gas Reservoir Simulator Version 2002* da CMG) (IMEX, 2000) que possui muitas características relevantes, sendo as mais importantes para este trabalho as seguintes:

- gerenciamento detalhado de poços;
- simulação de poços verticais, horizontais e diagonais; e
- porosidade e permeabilidade duplas.

Dado que o simulador é um módulo separado, a comunicação entre o algoritmo otimizador e o simulador de reservatórios dá-se por meio de arquivos, havendo comunicação nos dois sentidos (escrita de arquivos na linguagem do simulador e leitura de alguns arquivos fornecidos pelo simulador). Deste modo, o algoritmo otimizador acessa informações sobre o modelo de reservatório especificado para o simulador, recuperando dados sobre a geometria do reservatório tais como: dimensões da malha; tipo de malha; e número de blocos nas direções i , j , k , que servem para obter uma matriz de distâncias, necessária para o cálculo das trajetórias dos poços e para a definição dos domínios de cada gene. Todos estes dados

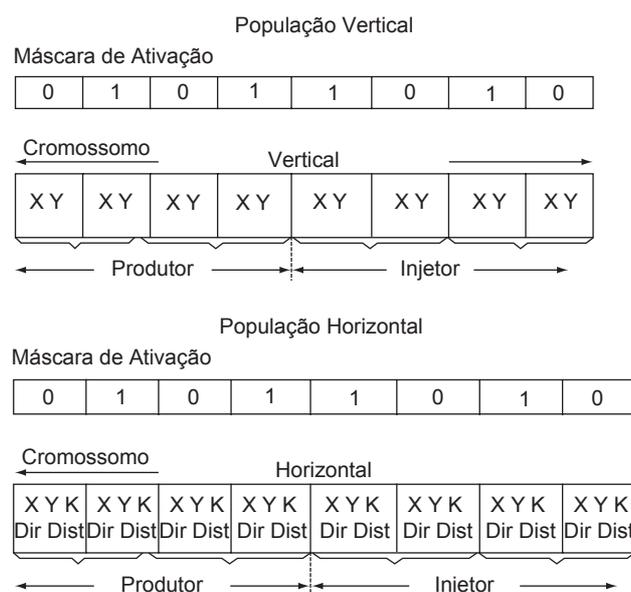


Figura 6. Representação dos cromossomos das populações da modelagem de coevolução.

são lidos do arquivo de especificação uma única vez e usados para iniciar o algoritmo otimizador, permanecendo fixos durante toda a evolução.

Além dessas especificações geométricas, o modelo otimizador também obtém do arquivo de especificação (em uma seção denominada *NULL) informações sobre os blocos inativos da malha de reservatório. Estes blocos são utilizados para determinar posições onde não se podem perfurar poços, devido, por exemplo, a condições como porosidade nula. Caso essa seção seja encontrada, o algoritmo otimizador mapeia toda a área inválida, utilizando-a como restrição na inicialização da população e no uso de operadores de crossover e mutação, evitando assim que poços sejam criados nessas áreas impróprias para exploração.

Após a criação de uma população, o algoritmo otimizador especifica uma determinada configuração de poços, utilizando a sintaxe padrão do IMEX, por meio de um arquivo *include.inc*, o qual é lido pelo simulador.

Após o término da simulação da alternativa fornecida pelo algoritmo otimizador, o simulador fornece um arquivo de saída contendo informações sobre a produção de óleo, gás e água em condições de superfície e referentes ao campo em sua totalidade. Esses dados são empregados pelo módulo cálculo do valor presente líquido, para a determinação da avaliação da alternativa, conforme descrito na seção a seguir.

2.3 Cálculo do valor presente líquido (VPL)

Conforme mencionado, no sistema de otimização de alternativas proposto neste trabalho, cada cromossomo representa uma alternativa a ser avaliada. Esta avaliação ocorre da seguinte forma:

- a alternativa gerada pelo módulo otimizador é inserida no módulo simulador;
- o módulo simulador fornece um perfil de produção do campo (curvas de produção de óleo, gás e afluência de água) dependendo da configuração de poços encontrada e do tempo simulado em anos; e
- o perfil de produção obtido é inserido no módulo de cálculo do VPL, o qual fornece a avaliação da alternativa (cromossomo) especificada pelo módulo otimizador.

O valor presente representa uma precificação no dia de hoje para algum objeto ou evento que ocorrerá somente no futuro. Para isto, torna-se necessário ter uma boa previsão do comportamento deste evento futuro no dia de hoje. Para o problema de alternativas de produção, os eventos são os fluxos de caixa (BREALEY; MYERS, 1981) para determinados espaços de tempo. Estes espaços de tempo ficam atrelados aos valores que o simulador de reservatórios irá fornecer. O ajuste ao tempo presente é feito por um decremento do valor previsto no futuro, o qual depende da diferença temporal entre hoje e o tempo

de ocorrência futuro e uma taxa de desconto exponencial. Como é um evento segmentado no tempo, o valor presente será o somatório de todos estes valores descontados exponencialmente no tempo, conforme especificado na expressão seguinte:

$$VP_X = \sum_{i=0}^n x_i e^{-\rho t_i} \quad (1)$$

em que x_i é o evento no tempo i , t_i é a distância temporal do evento x_i e ρ é o valor da taxa de desconto exponencial.

O Valor Presente Líquido (VPL) é a diferença entre o Valor Presente (VP) da alternativa e o custo de desenvolvimento desta (D), como se mostra na expressão seguinte:

$$VPL = VP - D \quad (2)$$

Os custos de desenvolvimento (D), isto é, dos investimentos efetuados para iniciar a produção, são calculados a partir dos poços a serem perfurados e das dimensões das linhas de condução, risers, planta e plataforma, como mostra a expressão a seguir:

$$D = (a * f_{ph} + r)n_p + b + \sum_{j=1}^{n_p} |p_j - P_{pl}| * c \quad (3)$$

em que os parâmetros, conhecidos como CAPEX (*Capital Expenditure*), estão descritos a seguir: a - Custo médio de perfuração de um poço mais custo da Árvore de Natal; r - Custo do riser para uma linha d'água de 1 km; b - Custo de transferência mais custo da plataforma e planta; p_j - Posição do poço p_j ; P_{pl} - Posição da planta mais plataforma; c - Custo de linha por quilômetro; n_p - Número de poços por alternativa; e f_{ph} - Fator de custo de poços horizontais.

O valor do fator de custo por poços horizontais f_{ph} da expressão anterior é calculado como segue.

$$f_{ph} = \left(1 + \frac{n_{ph}}{n_p} f_h \right) \quad (4)$$

sendo n_{ph} o número de poços horizontais que existem na alternativa e o fator f_h é o fator de custo adicional para poços horizontais BITTENCOURT; HORNE, 1997). Com essa formulação evita-se separar o número de poços verticais e horizontais, usando-se somente o número total de poços da alternativa e o número de poços horizontais.

O valor presente da alternativa é composto pela diferença entre o valor presente da receita (VP_R) menos o valor presente do custo operacional (VP_{COP}), aplicando ao resultado a alíquota de imposto I de aproximadamente 34%. Assim, a expressão para obtenção do valor presente da alternativa é dada por:

$$V = (V_R - V_{COP}) (1 - I) \quad (5)$$

em que o valor presente da receita VP_R depende da produção de óleo $Q(t)$ mais a produção de barril-equivalente de gás $G(t)/1000$, do preço de petróleo $P_{oil}(t)$, nos diferentes intervalos de tempo t , e da constante de amortecimento ρ . Como neste caso as condições de mercado são de certeza, o preço do petróleo no tempo t pode ser expresso como uma função $P_{oil}(t)$ constante. Assim, para cada intervalo de tempo o valor da receita pode ser obtido como:

$$R(t) = (Q(t) + G(t) / 1000) P_{oil}(t) \quad (6)$$

e o valor presente como:

$$V_R = \sum_{i=1}^T R(t_i) e^{-\rho t_i} \quad (7)$$

em que a constante de amortecimento ρ é 0,1 (valor indicado pelo especialista), e o valor T representa o tempo máximo de atividade lucrativa do reservatório, isto é, enquanto o valor da receita for maior que o custo de operação. O valor t_i é o passo de tempo que foi determinado pelo simulador de reservatório para montar as curvas de produção de óleo, gás e afluência de água.

O valor presente do custo operacional representa os custos para manter a operação de produção de óleo. Esses custos dependem da produção obtida e de outros custos mais fixos, como por exemplo, a manutenção periódica dos poços. Desta forma, o custo operacional para o tempo t é dado por:

$$C_{OP}(t_i) = \quad (8)$$

$$(m.n_p) \frac{t_i}{365} + C_f \frac{t_i}{365} + C_v \cdot Q(t_i) + R_y R(t_i) + C_w W(t_i)$$

sendo t_i - Tempo de simulação i (em dias); m - custo de manutenção anual de um poço (considerado igual para todos os poços); c_v - constante para cálculo de custos variáveis no tempo que dependem da produção; C_f - custos fixos anuais; C_w - custo de retirada da água/m³; $Q(t_i)$ - produção de óleo no tempo t_i ; $W(t_i)$ - afluência de água no tempo t_i ; R_y - alíquota de *royalties*; e n_p - número de poços.

Assim, o valor presente do custo operacional é calculado conforme a expressão abaixo:

$$V_{COP} = \sum_{i=1}^T C_{OP}(t_i) e^{-\rho t_i} \quad (9)$$

Foram descritas nesta seção todas as variáveis necessárias para o cálculo do VPL, usado como função de avaliação que será maximizada pelo algoritmo otimizador.

2.4 Ambiente de processamento distribuído

Dada a evolução atual dos sistemas de rede de alto desempenho e, com esta evolução, a possibilidade de

aproveitar a grande capacidade de computação conjunta de ambientes colaborativos de pesquisa e desenvolvimento, a computação paralela e os sistemas distribuídos cada vez mais atendem à crescente demanda por recursos computacionais a baixo custo.

O sistema de otimização proposto é iterativo e requer uma grande quantidade de simulações de reservatório. Dado que a execução de uma destas simulações pode levar minutos ou horas para ser realizada, surge a necessidade de aproveitar o poder computacional de um grupo de processadores pelo uso de ambientes de computação paralela e distribuída aplicada a sistemas de otimização evolucionária.

Na literatura encontram-se diferentes modelos de algoritmos genéticos distribuídos (TOMASSINI, 1999; COHOON et al., 1987; MANDERICK, 1989) que vêm a ser os próprios algoritmos genéticos inseridos de diferentes formas em uma arquitetura de computação paralela ou distribuída. Dentro destes algoritmos ressaltam-se os seguintes:

- a) **algoritmo trivial** (CANTÚ-PAZ, 1998): realiza a execução simultânea de várias versões de um mesmo problema de otimização em vários processadores independentes, com diferentes populações iniciais ou diferentes valores de parâmetros do AG. Apesar desta ser a forma mais simples de paralelismo, já se obtêm reduções no tempo de execução e obtenção de estatísticas para um problema baseado em AG; e
- b) **algoritmo global de população simples** (CANTÚ-PAZ, 1998): é um algoritmo genético com uma única população em que apenas o módulo de avaliação é executado no ambiente distribuído. A grande vantagem está na independência existente entre as avaliações de cada indivíduo da população, a qual viabiliza a realização de múltiplas avaliações distribuídas ao longo dos diversos processadores existentes na rede de computadores. A arquitetura de comunicações empregada é a *master-slave*. O processador *master* contém a população, operadores e métodos de seleção do algoritmo genético, e os processadores *slave* contêm a função de avaliação. Assim, com a possibilidade de se realizar avaliações simultâneas, há um ganho considerável no tempo de otimização, já que a avaliação normalmente consome o maior tempo no processamento de um AG.

Existem outros modelos que distribuem a população entre os processadores do ambiente distribuído, visando se assemelhar ao processo de evolução natural das espécies com característica de separação geográfica entre indivíduos e grupos de indivíduos, como ocorre nos algoritmos em ilhas (COHOON et al., 1987) e os algoritmos celulares (MANDERICK, 1989).

Neste trabalho, foi empregado o algoritmo genético global de população simples com arquitetura de comu-

nicações *master-slave*, adequada para os modelos de algoritmos genéticos, algoritmos culturais e os algoritmos de coevolução cooperativa utilizados neste trabalho. A Figura 7 mostra a arquitetura *master-slave* aqui empregada.

Na Figura 7, o *master* e cada *slave* são, normalmente, aplicações instanciadas em diversos computadores de uma rede local de alto desempenho. Pode-se notar que o computador *master* se comunica com cada um dos *slaves* enviando um indivíduo (cromossomo) a ser avaliado, e fica aguardando o valor de aptidão calculado pelo *slave*. Assim, vários indivíduos podem ser avaliados simultaneamente, por diferentes *slaves*, pois o *master* fica aguardando a finalização de cada avaliação para enviar um novo indivíduo a cada *slave*, até se esgotarem os indivíduos da população.

Com o propósito de adequar na implementação o sistema desenvolvido ao modelo do algoritmo global, o ambiente de processamento distribuído foi dividido em três grandes módulos:

- a) **módulo de dados:** que contém as informações necessárias para realizar a otimização: campos, alternativas existentes e parâmetros para o cálculo do VPL;
- b) **módulo de engenharia:** que contém os algoritmos utilizados na otimização, algoritmos genéticos, algoritmos culturais e coevolução; e
- c) **módulo de cenários:** que contém as rotinas referentes à avaliação de alternativas, como o *link* com o simulador de reservatórios e o cálculo do VPL.

Os módulos de dados e engenharia estão instanciados no *master* e o módulo de cenários é replicado e instanciado em cada um dos *slaves* existentes.

De forma a permitir a comunicação entre os três módulos, foram criadas duas interfaces de comunicação:

- a) interface para envio de dados de cenário e inicialização; e

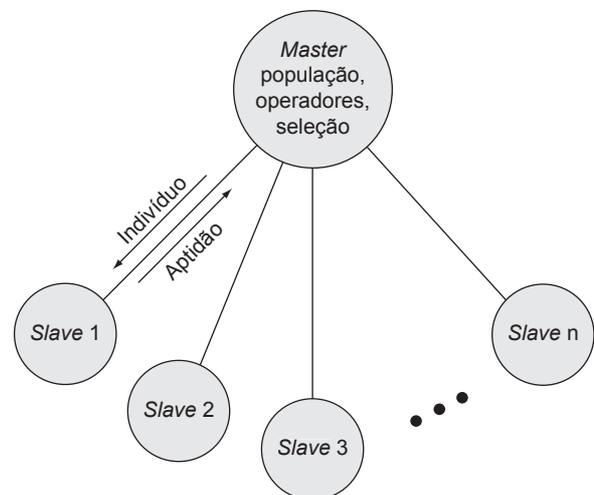


Figura 7. Arquitetura *master-slave*.

b) interface para envio de cromossomo e retorno da avaliação.

Estas interfaces foram implementadas utilizando a arquitetura CORBA (OMG, 2003; HENNING, 1999). Na Figura 8, mostram-se os módulos e as interfaces de comunicação implementadas com seus respectivos serviços.

3 Estudo de caso

Para a realização dos testes do sistema de otimização de alternativa proposto neste trabalho, foi utilizado um modelo de reservatório sintético, porém com características próximas às de campos existentes, denominado *campo-1*. A seguir é feita uma breve descrição das dimensões e características principais deste reservatório.

Este modelo de reservatório consiste em uma malha de 33 x 57 x 3 blocos, isto é, 33 blocos na direção i, 57 blocos na direção j e 3 na direção k. As dimensões de cada bloco são de aproximadamente 100,0 x 100,0 x 8,66 m. Este modelo não apresenta blocos nulos. Os valores geológicos do reservatório são os seguintes: permeabilidade: 575,0 (md) nas direções i, j; e 57,40 (md) na direção k; porosidade: 0,23; e pressão inicial: de 390 até 420 kg/cm², dependendo inversamente da altura.

Uma característica importante deste modelo é a existência de uma região com 100% de saturação de água na parte mais funda, formando um aquífero. As demais regiões

possuem saturação de água no valor de 0,25. Este fato cria a condição de heterogeneidade da reserva. A Figura 9 ilustra o campo-1 com uma vista em três dimensões.

Os testes realizados neste trabalho foram feitos no campo descrito acima e utilizando os três modelos propostos neste artigo. A Tabela 1 apresenta as melhores configurações encontradas dos parâmetros dos modelos de algoritmos evolucionários. Essas configurações de parâmetros foram obtidas de forma empírica. Devido à complexidade do problema, foram testadas configurações nas quais o número de gerações e o número de indivíduos variou entre 50 e 150. Em todos os algoritmos evolucionários foi utilizado como critério de parada o número máximo de gerações.

Os testes 1 e 2 foram realizados com o algoritmo genético descrito na seção 2.1.1. Foram usados ambos os tipos de poços (vertical e horizontal) durante a evolução do AG. Os testes 3, 4 e 5 foram realizados com o modelo híbrido do AG/AC, sendo que no teste 3 foram usados poços verticais e horizontais, no teste 4 somente poços verticais e no teste 5 somente poços horizontais. Os testes 6 e 7 foram realizados com o modelo coevolucionário e foram usados poços verticais e horizontais. Somente o teste 7 foi testado com o processamento distribuído.

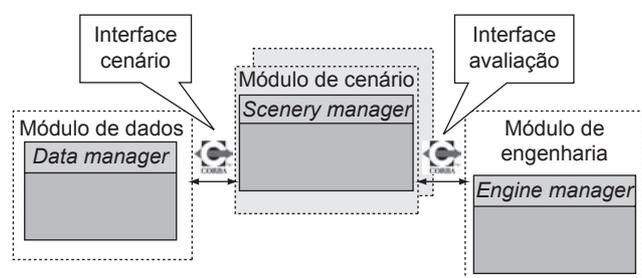


Figura 8. Módulos e interfaces de comunicação implementados no sistema.

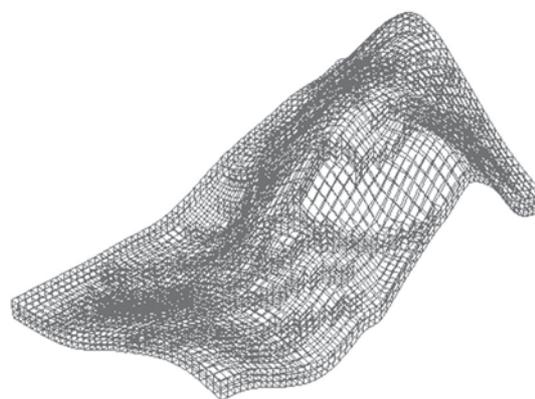


Figura 9. Visão em 3D do campo-1.

Tabela 1. Parâmetros da evolução.

Parâmetros	Teste AG		Teste AG /AC		Teste coevolução		
	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste6	Teste7
Tipo pPoço	Vert/hori	Vert/hori	Vert/hori	Vert	Hori	Vert/hori	Vert/hori
Número de gerações	150	80	150	100	100	50	70
Tamanho da população	100	80	100	50	50	50	60
Número total de genes	80	80	80	50	50	80	40
Steady State	0,4	0,4	0,4	0,4	0,4	0,4	0,4
pWell	0,3	0,3	0,3	0,3	0,3	0,3	0,3
Bônus	-	-	3	3	3	-	-
pBest	-	-	0,1	0,1	0,1	-	-
Operador	Todos	Todos	Todos	Todos	Todos	Todos	Todos
Proc. Distribuído	Não	Não	Não	Não	Não	Não	Sim

3.1 Apresentação dos resultados

A Tabela 2 contém os melhores valores de VPL obtidos e o número de poços encontrados em todos os testes do Campo-1. Novamente, a tabela é dividida entre os modelos apresentados (AG, AG/AC, Coevolução). Pelos resultados apresentados na Tabela 2, percebe-se que os melhores VPLs foram obtidos pelo algoritmo de Coevolução. Os testes 1 e 2, nos quais se utiliza um Algoritmo Genético puro, a convergência para valores positivos é mais lenta, devido à complexidade do problema e à falta de recurso capazes de agilizar a convergência. O teste 2 utilizou a melhor alternativa encontrada no teste 1 como semente inicial (ver seção 2.1.1-d), começando a evoluir a partir desta melhor solução.

Pelos resultados apresentados na Tabela 2, percebe-se que os melhores VPLs foram obtidos pelo algoritmo de Coevolução. Os testes 1 e 2, em que se utiliza um Algoritmo Genético puro, a convergência para valores positivos é mais lenta, devido à complexidade do problema e à falta de recursos capazes de agilizar a convergência. O teste 2 utilizou a melhor alternativa encontrada no teste 1 como semente inicial (ver seção 2.1.1-d), começando a evoluir a partir desta melhor solução.

O modelo híbrido do AG/AC demonstrou uma melhor capacidade de convergência para bons valores de VPL, mostrando-se mais eficiente que o modelo AG. Os testes 4 e 5 usaram somente um tipo de poço, tendo sido propostos por especialistas e obtendo resultados como o esperado.

A seguir são detalhados os resultados de alguns testes descritos nas Tabelas 1 e 2. Para cada teste apresentado a seguir, são fornecidos: o VPL alcançado de cada teste; a curva de desempenho dos algoritmos de otimização; e a distribuição de poços encontrada. A curva de desempenho dos algoritmos de otimização, denominada *Score*, mostra a avaliação da melhor alternativa a cada geração. Por ela pode-se verificar como ocorreu a convergência do algoritmo.

3.1.1 Algoritmo genético

O teste 1 foi realizado somente com o uso de Algoritmos Genéticos e contendo poços verticais e hori-

zontais. O melhor VPL encontrado neste teste foi de US\$ -13.788.509,70 correspondente a: 5 poços injetores e 6 poços produtores.

A Figura 10 mostra a configuração 3D encontrada no teste 1. A Figura 11 apresenta a curva de desempenho do algoritmo. O tempo gasto neste experimento foi de aproximadamente 27 horas, sendo o tempo médio de cada simulação de 20 segundos.

3.1.2 Algoritmo genético/algoritmo cultural

O teste 3 foi realizado com o uso do algoritmo híbrido AG/CA, também contendo poços verticais e horizontais. O melhor VPL encontrado neste teste foi de

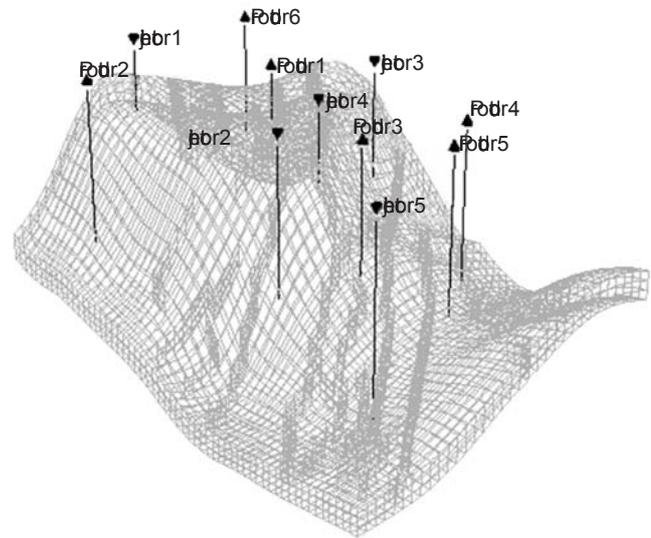


Figura 10. Configuração teste 1.

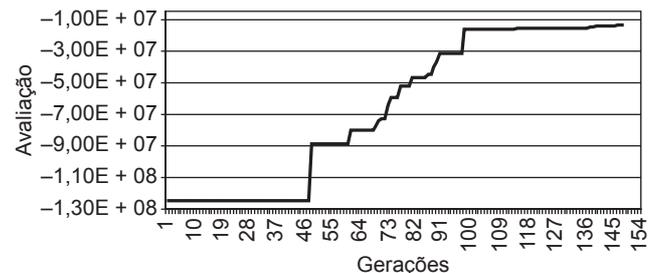


Figura 11. Gráfico *score* teste 1.

Tabela 2. Resultados.

Algoritmo genético			
Teste	Tipo poço	Nº poços	Melhor VPL
1	Vertical/horizontal	5 injetores/6 produtores	-1,38.10 ⁷
2	Vertical/horizontal	6 injetores/12 produtores	9,47.10 ⁷
Algoritmo genéticos/algoritmo cultural			
Teste	Tipo poço	Nº poços	Melhor VPL
3	Vertical/horizontal	5 injetores/11 produtores	1,55.10 ⁸
4	Vertical	15 injetores/16 produtores	-1,9.10 ⁸
5	Horizontal	1 injetor/2 produtores	1,9.10 ⁸
Coevolução			
Teste	Tipo poço	Nº poços	Melhor VPL
6	Vertical/horizontal	2 injetores/6 produtores	1,97.10 ⁸
7	Vertical/horizontal	2 injetores/5 produtores	2,25.10 ⁸

U\$ 155.414.702,70, correspondente a: 5 poços injetores e 11 poços produtores.

A Figura 12 mostra a configuração de poços encontrada e a Figura 13 apresenta o gráfico de desempenho do algoritmo. O tempo gasto neste experimento foi de aproximadamente 25 horas, sendo o tempo médio de cada simulação, como no caso anterior, igual a 20 segundos.

3.1.3 Coevolução

O teste 6 foi realizado com o uso de coevolução com poços verticais e horizontais. O melhor VPL encontrado neste teste foi de U\$ 197.649.844,80, correspondente a: 2 poços injetores e 6 poços produtores.

As Figuras 14 e 15 apresentam a configuração de poços encontrada e o gráfico de desempenho do algoritmo neste teste, respectivamente. O tempo gasto neste experimento foi de aproximadamente 30 horas, sendo o tempo médio de cada simulação também igual a 20 segundos.

3.1.4 Processamento distribuído

O processamento distribuído foi aplicado no algoritmo de melhor desempenho, isto é, Coevolução, teste 7. Foram utilizados 20 computadores. A Figura 16 mostra o gráfico

de desempenho do algoritmo (*Score*) e a Figura 17 o gráfico do percentual de uso dos computadores durante o processo de evolução distribuído. O tempo gasto para rodar o algoritmo de Coevolução com 70 gerações foi de 3 horas.

Na Figura 17, as alturas das barras representam a quantidade relativa de avaliações realizadas por cada um dos 20 computadores. Assim, barras mais altas correspondem a computadores mais potentes que realizaram um maior percentual do total de avaliações da otimização.

3.2 Discussões

Os resultados obtidos com os testes apontam, pelas curvas *score*, uma otimização consistente durante a evolução e indicam que as técnicas de Computação Evolucionária utilizadas - Algoritmos Genéticos, Algoritmos Culturais e Coevolução -, são adequadas à otimização do problema de alternativas para desenvolvimento de campos de petróleo. Tipicamente, essas curvas denotam maior inclinação no início e uma convergência no final da evolução, isto é, um número maior de gerações não seria vantajoso tendo em vista o alto custo computacional

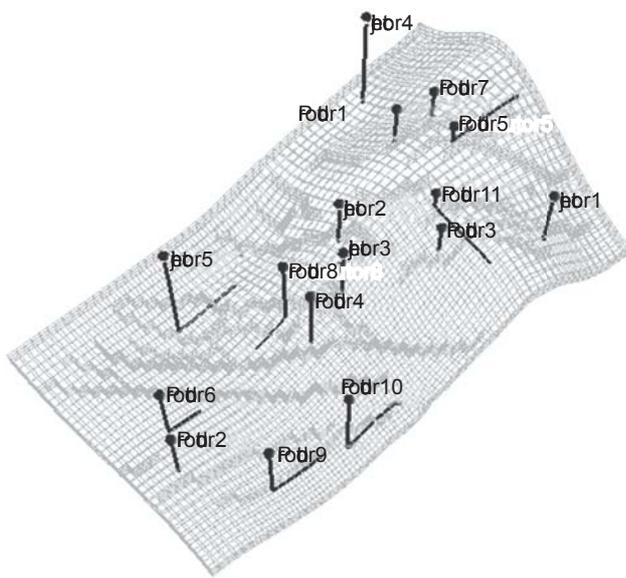


Figura 12. Configuração teste 3.

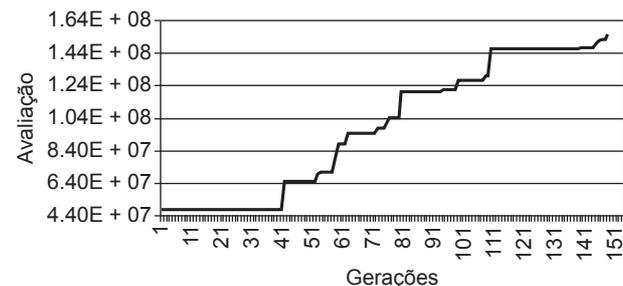


Figura 13. Gráfico *score* teste 3.

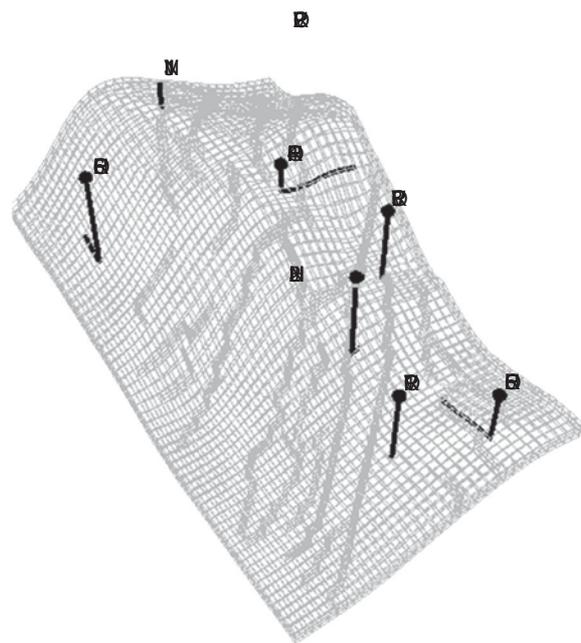


Figura 14. Configuração teste 6.

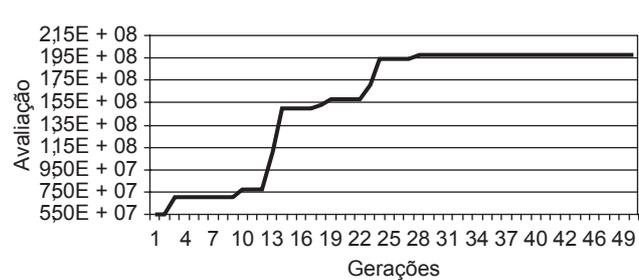


Figura 15. Gráfico *score* teste 6.

dos experimentos. Este custo computacional é, essencialmente, devido à execução constante do simulador de reservatórios. Em média, cada experimento levou 30 horas num PC de 1000 MHz. Caso exista interesse em se obterem soluções mais rapidamente, aceitando uma redução na precisão do modelo, pode-se utilizar um número menor de gerações e indivíduos.

Das técnicas inteligentes utilizadas neste artigo, a Coevolução ofereceu maior desempenho (melhores resultados com um menor número de gerações), alcançando alternativas que apresentaram maiores valores de VPL. Isso pode ser visto comparando-se os testes feitos por meio das curvas de desempenho. Assim, a convergência para melhores resultados ocorre de forma mais rápida para o algoritmo de Coevolução.

Os testes 4 e 5 foram propostos por um especialista, e os resultados encontrados foram compatíveis com o esperado. No teste 5, no qual se tem somente o uso de poços horizontais, os valores de VPL encontrados foram mais altos do que no teste 3, em que se tem o uso de poços verticais e horizontais. Comparando o teste 5 com o teste 4, a diferença é ainda maior. No teste 4, com apenas poços verticais, o VPL não saiu de valores negativos. Esse resultado era esperado, pois o uso de poços

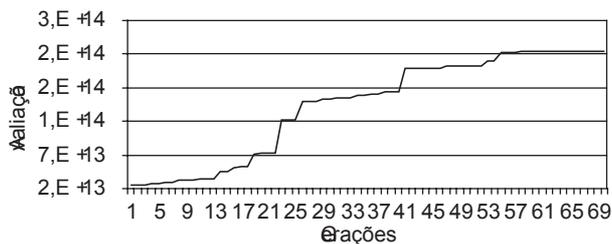


Figura 16. Gráfico score teste 7.

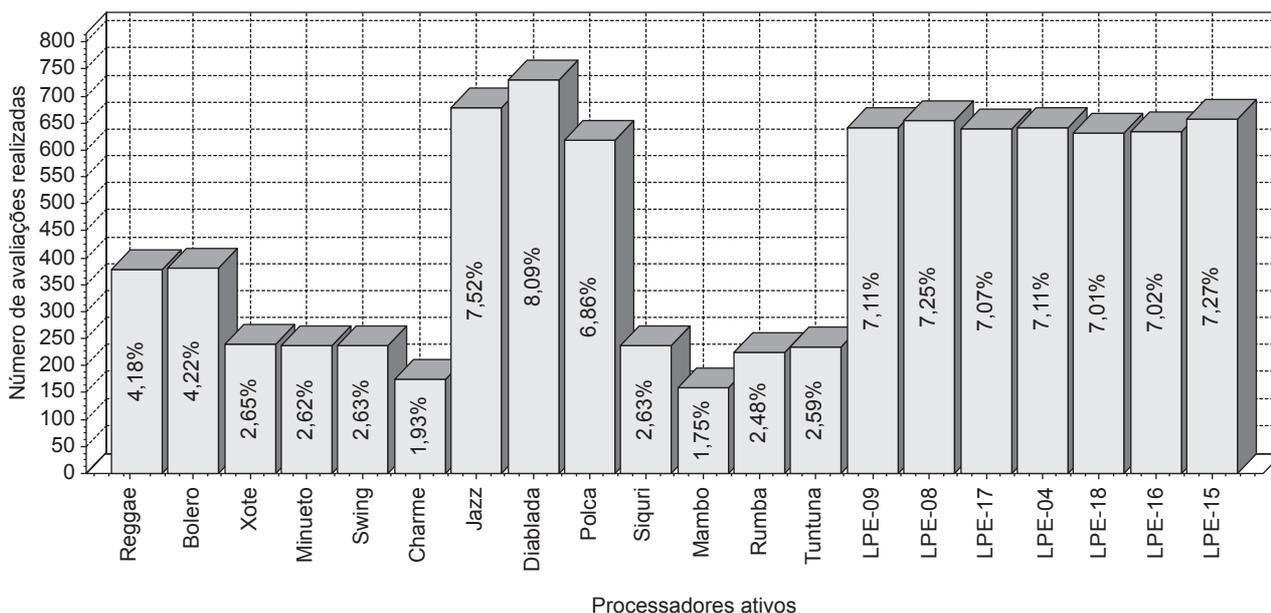


Figura 17. Computadores usados na avaliação distribuída.

horizontais permite uma maior obtenção de óleo com um custo de perfuração relativamente menor.

Outro requisito importante foi observado por um especialista: a proporção do número de poços injetores e produtores, alcançado nos testes, indica que o número de poços injetores nunca é maior que o número de poços produtores, no máximo deve existir uma proporção de 1 para 1. Isso foi validado por especialistas como uma boa característica do algoritmo.

4 Conclusões e trabalhos futuros

Os resultados encontrados indicam de forma promissora que as técnicas de computação evolucionária usadas, algoritmo genético, algoritmo cultural e coevolução, podem se tornar boas ferramentas de apoio para encontrar melhores alternativas de exploração de um campo de petróleo, confirmando o objetivo deste estudo.

A representação do cromossomo proposta e utilizada no sistema consegue tratar várias particularidades do problema:

- permite a criação de poços verticais e/ou horizontais;
- permite delimitar um número máximo de poços a ser empregado na exploração do campo, evitando soluções indesejáveis como poços em todas as localizações da malha;
- considera a existência de áreas impróprias para exploração dentro dos limites da malha do reservatório e não deixa que poços sejam criados nessas regiões; e
- permite a utilização de conhecimento prévio, ou seja, é possível aproveitar alternativas dadas pelo especia-

lista e evoluir a partir delas no sentido de encontrar melhores alternativas, reforçando a utilidade deste sistema como uma ferramenta de apoio à decisão.

Foram feitos testes com um sistema utilizando AG, um sistema híbrido utilizando AG/AC e um sistema utilizando Coevolução. Os três sistemas apresentaram bom desempenho, encontrando alternativas com altos valores de VPL, sendo que o sistema com Coevolução consegue convergir para melhores valores com um menor número de ciclos. Com o uso de avaliações distribuídas, o tempo computacional foi reduzido consideravelmente.

O sistema híbrido AG/AC oferece uma melhora no desempenho, uma vez que na evolução cultural não se consideram apenas cromossomos promissores, mas também genes promissores, evitando assim que informações de bons genes, contidos em cromossomos não tão bem avaliados, sejam desperdiçadas.

A evolução com Coevolução conseguiu chegar aos melhores resultados. Existem duas razões principais pelas quais algoritmos evolucionários convencionais não são totalmente adequados para resolver esse problema de grande complexidade. Em primeiro lugar, os algoritmos genéticos convencionais impedem, a longo prazo, a preservação de certos componentes da solução, pois, por estarem codificados por completo em um indivíduo, eles são avaliados como um todo e apenas os subcomponentes que pertencem a indivíduos com avaliações altas serão

preservados. Em segundo lugar, o fato da representação estar relacionada a uma solução completa e por não haver interações entre os membros da população, não existe pressão evolucionária para a ocorrência de co-adaptação, ou seja, não existe pressão para a adaptação de um subcomponente dada a ocorrência de uma mudança em outro subcomponente (DURHAM, 1991).

Com relação ao algoritmo otimizador, propõe-se a investigação nos seguintes itens:

- a) Alterar o modelo para evoluir não só localizações, quantidades e tipos de poços, mas também o momento em que estes devam iniciar a sua produção;
- b) Considerar, na avaliação da alternativa, o custo da retirada da água que vem junto com o óleo em sistemas de extração de óleo de tipo *Water Flooding*;
- c) Com relação à trajetória dos poços, propõe-se estender o sistema existente de forma tal que os poços sejam definidos pela trajetória sem importar o tipo (vertical ou horizontal), visando com isso suportar a otimização de configurações angulares e multilaterais; e
- d) Criar um Aproximador de Função para inferir a resposta do simulador de reservatórios, reduzindo-se significativamente o tempo de processamento requerido.

Todos esses itens já foram iniciados e encontram-se em andamento.

Selection of alternatives for the development of oil fields

Abstract

This paper presents an optimization system for the development of petroleum fields. Developing a petroleum field consists of choosing an alternative exploitation of an already known and delimited petroleum reservoir allowing the maximum hydrocarbon production within the physical and economical limitations i.e., maximizing the net present value (NPV). The net present value is calculated according to the oil production, which is obtained with the use of a reservoir simulator. Each reservoir simulation can take from few seconds to several hours, depending on the complexity of the reservoir being modeled. This reduces the total number of configurations that can be generated and evaluated by the user in search for the best solution. Therefore, this work proposes and evaluates a new intelligent, optimization system that employs genetic algorithms (GA), cultural algorithms (CA), and co-evolution in order to search for an optimal development alternative in a parallel computing environment for reservoir simulations and NPV calculation. The proposed system provides the user, in a reasonable time, with the optimum (or sub-optimum) configuration for the development of the petroleum field. The results obtained in the case studies demonstrate that the proposed system, based on intelligent techniques, enable good configurations for the development of petroleum fields with a great reduction in computational time. This reduction is obtained from the computational power of the parallel computing environment and from the expert knowledge, through the initial configuration of the optimizing system (initial seed).

Keywords: Genetic algorithms. Cultural algorithms. Co-evolution. Optimization. Petroleum reservoir development. Reservoir engineering. Distributed genetic algorithm. Parallel computing.

Referências bibliográficas

- ALMEIDA, L. F. **Otimização de alternativas para desenvolvimento de campo de petróleo utilizando computação evolucionária**. 82 f.. Dissertação (Mestrado) – Puc-RJ, Rio de Janeiro, fev. 2003.
- BACK, T. **Evolutionary algorithms in theory and practice**. Oxford, UK: Oxford University Press, 1996, 314 p, ISBN 0-19-509971-0.
- BECKNER, B. L.; SONG, X. **Field development planning using simulated annealing**: Optimal economic well scheduling and placement. SPE Annual Technical Conference & Exhibition, Dallas, Texas, oct. 22-25, 1995, Paper No. SPE 30650, p. 209-221.
- BITTENCOURT, A. C. **Optimizing hydrocarbon field development using a genetic algorithm based approach**. PhD thesis, Stanford University, CA, 1997.
- BITTENCOURT, A. C.; HORNE, R. N. **Reservoir development and design optimization**. SPE Annual Technical Conference and Exhibition, San Antonio, Texas, October 5-8, 1997, Paper No. SPE 38895, 14p.
- BREALEY, R.; MYERS, S. **Principles of corporate finance**. New York: McGraw-Hill series in finance, 1981, ISBN 0070074178.
- CANTÚ-PAZ, E. Designing efficient master-slave parallel genetic algorithms. Genetic Programming 1998. In: ANNUAL CONFERENCE, 3., 1998, San Francisco, CA, USA. **Proceedings...** Madison, Wisconsin, USA: University of Wisconsin, Morgan Kaufmann, jul. 22-25, 1998. p. 455-467.
- COHOON, J. P. et al. Punctuated equilibria: A parallel genetic algorithm. In: INTERNATIONAL CONFERENCE OF GENETIC ALGORITHMS, CAMBRIDGE, 2., 1987, , USA. **Proceedings...** Massachusetts: Lawrence Erlbaum Associates, 1987, p. 148-154, ISBN 0-8058-0158-8.
- DURHAM, W. H. **Co-Evolution**: Genes, culture and human diversity. Stanford, CA: Stanford University Press, 1991, 656 p.
- GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley Publishing Company, Inc. 1989, 372 p, ISBN 0201157675.
- GUYAGULER, B., et al. **Optimization of well placement in a gulf of Mexico water flooding project**. SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 2000, Paper No SPE 63221, 10p.
- GUYAGULER, B.; HORNE, R. Optimization of well placement. **Journal of Energy Resources Technology**, v. 122, n. 2, p. 64-70, jun. 2000.
- HENNING, M.; VINOSKI, S. **Advanced CORBA® Programming with C++**. Addison-Wesley Professional Computing Series, Pearson Education, USA, February 1999, ISBN 0-20137-927-9, ISBN-13 9780201379273, 1120p, Upper Saddle River, NJ.
- IMEX Advanced Oil/Gas Reservoir Simulator version 2000 User's Guide. Computer Modeling Group LTD, 2000.
- MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Springer, USA, 1996, 387p, ISBN 3540606769.
- MITCHELL, M. **An Introduction to Genetic Algorithms**. Boston, Massachusetts, USA: The MIT Press, 1996, 217p, ISBN-10 0-262-13316-4, ISBN-13 978-0-262-13316-6.
- MANDERICK, B.; SPIESSENS, P. Fine-grained parallel genetic algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989. San Francisco. **Proceedings...** San Francisco, CA, USA: Morgan Kaufmann, 1989, p. 428-433, ISBN:1-55860-066-3.
- OBJECT MANAGEMENT GROUP. **CORBA Services**: Common Object Services Specification. Disponível em: <ftp://ftp.omg.org/pub/docs/formal/98-07-05>. Acesso em: 23 jul. 2003.
- POTTER, M. A.; DE JONG, K. A. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. **Evolutionary Computation**, v. 8, n. 1, p. 1-29. Boston, Massachusetts: MIT Press, 2000.
- PEDROSO, C. J.; SCHIOZER, D. J. **Otimização de locações de poços usando simulação numérica de reservatórios e computação paralela**. Rio Oil & Gas Expo and Conference, Rio de Janeiro, October, 2000.
- REYNOLDS, G. R. An introduction to cultural algorithms. In: ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING, 3., 1994. **Proceedings...** World Scientific Publishing, 1994, p. 131-139.
- REYNOLDS, G. R.; CHUNG, J. C. A self-adaptive approach to representation shifts in cultural algorithms. In: IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION (ICEC '96), 1996. **Proceedings...** may, 20-22, 1996, Nagoya University, Japan: IEEE, 1996a, ISBN 0-7803-2902-3.
- _____. A testbed for solving optimization problems using cultural algorithms. Evolutionary Programming V. In: ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING, 5., 1996, San Diego. **Proceedings...** San Diego CA, USA: MIT Press, 1996b, ISBN: 0-262-06190-2.
- TOMASSINI, M. Parallel and distributed evolutionary algorithms: a review. In: MIETTINEN, K.; MKEL, M.; NEITTAANMKI, P.; PERIAUX, J. (Ed.). **Evolutionary Algorithms in Engineering and Computer Science**. New York, NY: Wiley, p. 113-133, 1999.
- TÚPAC, Y. J.; VELLASCO, M. M. B. R.; PACHECO, M. A. C. Selection of alternatives for oil field development by genetic algorithms. **Revista de Engenharia Térmica - RETERM**. Edição Especial n. 2, p. 51-54, Curitiba, PR, Janeiro, 2002, ISSN 1676-1790.
- ZEBULUM, R. S.; PACHECO, M. A. C.; VELLASCO, M. M. B. R. **Evolutionary Electronics**: Automatic Design of Electronic Circuits and Systems by Generic Algorithms. Florida: CRC press, 2001, 320 p, ISBN 0849308658.

Sobre os autores

Luciana Faletti Almeida**Yván Túpac Valdivia****Marley Maria Bernardes Rebuszi Vellasco****Marco Aurélio Cavalcanti Pacheco**

Inteligência Computacional Aplicada, ICA: Laboratório de Inteligência Computacional Aplicada,
Departamento de Engenharia Elétrica, Centro Técnico Científico,
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio,
R. Marques de S. Vicente 225, CEP 22453-900, Gávea, Rio de Janeiro, RJ, Brasil,
e-mails: faletti@ele.puc-rio.br; yvantv@ele.puc-rio.br; marley@ele.puc-rio.br; marco@ele.puc-rio.br

Agradecimentos: À CENPES/PETROBRAS pelo suporte financeiro para a realização deste trabalho por meio dos projetos de pesquisa ANEPI – Análise Econômica de Projetos de E&P sob Incerteza. À CMG pelo suporte oferecido com o software de simulação de reservatórios CMG/IMEX.

Recebido em 03/3/2006
Aceito em 03/4/2007

