



COMPARAÇÃO DE ALGORITMOS EVOLUTIVOS PARA O PROBLEMA DE AJUSTE DE HISTÓRICO USANDO O MODELO IMPES

Iago Augusto de Carvalho

iagoac@ufmg.br

Departamento de Ciência da Computação, Universidade Federal de Minas Gerais

Daniel Gonçalves Rocha

Vinicius da Fonseca Vieira

Carolina Ribeiro Xavier

Departamento de Ciência da Computação, Universidade Federal de São João Del Rei

Resumo. Neste trabalho apresentamos um estudo dos parâmetros de entrada de um Algoritmo Genético e um algoritmo de Evolução Diferencial para o ajuste de histórico automático em reservatórios de petróleo. O problema de Ajuste de Histórico é um problema inverso que procura por um conjunto de parâmetros que minimiza a diferença entre o comportamento do modelo desenvolvido e o comportamento histórico do objeto modelado. Ambos os algoritmos foram implementados em um modelo paralelo de ilhas e uma série de testes variando os parâmetros dos algoritmos foram conduzidos. Os resultados demonstram que o algoritmo de Evolução Diferencial paralelo em ilhas é superior ao Algoritmo Genético.

Palavras chave: Simulação de reservatórios, Ajuste de histórico, Algoritmos evolucionários, Engenharia de reservatórios, Paralelização

1 Introdução

A simulação de reservatórios é uma poderosa ferramenta e que vem sendo amplamente utilizada na engenharia de reservatórios. Ela combina áreas como a física, matemática, engenharia de reservatórios e computação, tendo como principal objetivo construir um modelo computacional que pode ser utilizado para prever o comportamento de um reservatório real. Infelizmente, estes modelos computacionais são dependentes de vários parâmetros e características do reservatório em estudo, de forma que a performance da predição realizada é dependente de uma boa estimativa de alguns parâmetros físicos, como a distribuição de permeabilidade do reservatório. Muitas dificuldades surgem durante a validação do modelo, pois a maioria dos reservatórios de óleo estão inconvenientemente enterrados milhares de pés abaixo da superfície (Oliver et al., 2008).

Uma alternativa para a validação do modelo é a estimativa das propriedades relevantes do reservatório através do *Ajuste de Histórico* (Watson et al., 1994). O processo de Ajuste de Histórico consiste em um problema inverso que minimiza a diferença entre a performance entre um modelo construído e os dados históricos de um sistema real. Tipicamente, modelos de reservatórios possuem uma complexidade muito elevada, desta forma sendo necessário a construção de métodos computacionais para sua execução. Ainda assim, o processo consome uma enorme quantidade de tempo, sendo indispensável o uso de computação paralela em sua implementação.

Métodos baseados em derivadas muitas vezes não obtêm bons resultados, devido as características do problema e seu incontável número de ótimos locais. Os trabalhos (Rosa et al., 1983, Rosa et al., 1996) utilizam os métodos de funções de penalidades e métodos dos mínimos valores absolutos modificado, respectivamente, visando resolver o problema de ajuste de histórico em reservatórios de petróleo. Métodos livres de derivadas e baseados em *Algoritmos Evolutivos* foram propostos na literatura. Em (Santos et al., 2009, Soleng, 1999, Xavier et al., 2013), um Algoritmo Genético (Holland, 1992) foi proposto, e foi demonstrado que este método supera alguns métodos *Newton-like* para o problema de Ajuste de Histórico. Uma comparação entre um Algoritmo Genético e o algoritmo de Evolução Diferencial (Storn and Price, 1995, Price et al., 2005) foi desenvolvida em (Santos et al., 2012) e demonstrou que a Evolução Diferencial tem melhor performance que Algoritmos Genéticos para problemas de Ajuste de Histórico. Este artigo apresenta um largo estudo sobre o ajuste dos parâmetros destes dois algoritmos aplicados ao ajuste de histórico automático em um modelo de fluxo de petróleo 2D. Nós implementamos os algoritmos em um ambiente distribuído distribuindo espacialmente a população dos Algoritmos Evolutivos em várias subpopulações, utilizando o *modelo paralelo de ilhas* (Tomassini, 2006), utilizado para avaliar as diferentes combinações de parâmetros e realizando, deste modo, uma comparação mais justa entre os algoritmos que em (Santos et al., 2012).

O trabalho está organizado como se segue: na próxima Seção é apresentado o modelo de simulação de reservatórios utilizado neste trabalho.

2 Problema direto

O problema tratado neste artigo consiste em um fluxo de meios porosos incompressível e imiscível de duas dimensões e duas fases (água/óleo) em um ambiente livre de gravidade (Kassem et al., 2006). O sistema de equações diferenciais parciais que rege este fluxo é derivado

da lei de conservação de massas e a Lei de Darcy. A lei de conservação de massa para ambas as fases pode ser escrita como demonstrado na Equação (1).

$$\phi \partial_t (\rho_\alpha s_\alpha) + \nabla \cdot (\rho_\alpha v_\alpha) = Q_\alpha, \quad (1)$$

onde $\alpha = w$ denota a fase de água, $\alpha = o$ denota a fase de óleo, ϕ é relativo a porosidade do meio poroso, e ρ_α , s_α , v_α e Q_α são, respectivamente, a densidade, saturação, velocidade volumétrica e taxa de vazão em poços da fase α . A velocidade volumétrica (v_α) pode ser dada pela Lei de Darcy, como escrita na Equação (2).

$$v_\alpha = \frac{K k_{r\alpha}(s_\alpha)}{\mu_\alpha} \nabla p_\alpha, \quad (2)$$

onde K é a permeabilidade efetiva do meio poroso, $k_{r\alpha}$ é a permeabilidade relativa na fase α , que é uma função dependente da saturação, e μ_α e p_α são, respectivamente, a viscosidade e a pressão na fase α . Neste trabalho, é considerado que a pressão dos capilares é nula, ou seja, $p_w = p_o$. Desta forma, podemos nos referir a pressão sempre por p . Além disso, seja $s_w + s_o = 1$.

Nós introduzimos a mobilidade de fase e a função de transmissibilidade nas Equações (3) e (4), respectivamente.

$$\lambda_\alpha(s) = \frac{k_{r\alpha}(s)}{\mu_\alpha} \quad (3)$$

$$T_\alpha(s) = K \lambda_\alpha, \quad (4)$$

onde $s = s_w$ a partir deste momento. A velocidade volumétrica pode ser descrita como $v_\alpha = -T_\alpha \nabla p$.

Assumindo que a densidade de fase e a viscosidade sejam constantes, obtemos

$$\begin{cases} \phi \rho_w \partial_t s_w + \rho_w \nabla v_w = Q_w \\ \phi \rho_o \partial_t s_o + \rho_o \nabla v_o = Q_o \end{cases} \quad (5)$$

Pode-se dividir as Equações em 5 por ρ_α e somar ambas para obter a Equação (6).

$$\begin{cases} \phi \partial_t s + \nabla v_w = q_w \\ \nabla v_t = q_t \end{cases} \quad (6)$$

onde $q_\alpha = \frac{Q_\alpha}{\rho_\alpha}$ é a nova densidade de taxa de vazão da fase α , $q_t = q_w + q_o$ e $v_t = v_w + v_o$. Definindo a mobilidade total como $\lambda_t = \lambda_w + \lambda_o$ podemos introduzir a função de fluxo fracionado como $f(s) = \frac{T_w}{T_t} = \frac{\lambda_w}{\lambda_t}$. A Equação 5 pode ser reescrita como a Equação (7).

$$\begin{cases} \phi \partial_t s - \nabla (f(s) T_t(s) \nabla p) = q_w \\ -\nabla (T_t(s) \nabla p) = q_t \end{cases} \quad (7)$$

Para completar o modelo, as condições de limite devem ser especificadas. Neste trabalho é considerada a condição de limite de vazão, $v_\alpha \cdot \nu = 0, x \in \partial\Omega$, onde ν é a unidade externa normal ao limite $\partial\Omega$ do domínio Ω . Finalmente, a condição inicial pode ser definida como $s(x, 0) = s_0(x), x \in \Omega$.

O problema direto tratado neste trabalho é dado como o sistema de equações diferenciais parciais definido na Equação (7) com os limites e condições iniciais acima descritos.

2.1 Implementação

As equações diferenciais descritas na Seção 2 acopladas e não-lineares. Neste trabalho utilizamos o método *IMPES* para resolver estas equações. A implementação do método *IMPES* adota um esquema adaptativo de passo de tempo. A ideia básica por trás do método *IMPES* é separar os cálculos computacionais da equação de pressão da equação de saturação e resolver as equações de pressão e saturação usando aproximações de tempo implícitas e explícitas, respectivamente. Desacoplando as Equações 7 é possível obter uma equação elíptica para a pressão como descrito na Equação (8) e uma equação não-linear hiperbólica para a saturação, dada pela Equação (9).

$$-\nabla(T_t(s)\nabla p) = q_t . \quad (8)$$

$$\phi\partial_t s - \nabla(f(s)T_t(s)\nabla p) = q_w . \quad (9)$$

Para o cálculo da pressão, a saturação s da Equação (8) é suposta como conhecida e a Equação (8) é resolvida implicitamente para p . Neste trabalho, o método de volumes finitos foi utilizado para a discretização espacial do meio poroso (Watson et al., 1994).

Como mencionado anteriormente, a Equação (9), que representa a saturação, é resolvida explicitamente.

O método *IMPES* prossegue como a seguir: dado um s^0 ; para todo $n = 0, 1, \dots$ é utilizada a Equação (8) e s^n para avaliar p^n . Então, a Equação 9, s^n e p^n são utilizados para avaliar s^{n+1} . De forma a garantir a estabilidade desta equação, o passo de tempo Δt deve ser suficientemente pequeno, o que requer um alto custo computacional. Para minimizar este problema, é utilizado um método *IMPES* Aperfeiçoado (Chen et al., 2004). Este método utiliza o fato de que a saturação é alterada mais rapidamente que a pressão. Com este conhecimento, é possível calcular a pressão utilizando um passo de tempo muito superior ao passo de tempo da saturação.

Utilizando o método *IMPES* Aperfeiçoado nós temos dois diferentes passos de tempo: Δt^n para a pressão e $\Delta t^{n,l}$ para a saturação. A pressão p^n corresponde ao instante $t^n = \sum_{1 \leq i \leq n} \Delta t^i$ e a saturação $s^{n,l}$ corresponde ao instante $t^{n,l} = t^n + \sum_{1 \leq j \leq l} \Delta t^{n,j}$. Então, as condições CFL dadas nas duas próximas equações podem ser deduzidas.

Para as células que possuem poços injectores, as condições CFL são dadas pela Equação (10).

$$\Delta t^{n,l} \leq \frac{\phi(1 - s_{o,res} - s_{i,j}^{n,l})}{\beta_1 q_w^{n,l}(1 - f(s_{i,j}^{n,l}))} \quad (10)$$

onde $\beta_1 > 1$. As outras células, que não possuem poços injectores, as condições CFL são dadas pela Equação (11)

$$\max f'(s_m) \sum_m \frac{\Delta t^{n,l}}{\phi \Delta_m} |v_m^n| \leq \beta_2 \quad (11)$$

onde $0 < \beta_2 < 1$ e m correspondem as interfaces pelas quais o fluxo entra no bloco.

Para controlar a pressão no passo de tempo Δt^n podemos calcular a porcentagem de variação da pressão como $VP^n = \frac{\|p^{n+1} - p^n\|}{\|p^n\|}$. Se $VP^n > VP_{max}$, para uma dada VP_{max} , então o passo de tempo da pressão é reduzido. Caso $VP^n < VP_{min}$, para um dado VP_{min} , então o passo de tempo da pressão é aumentado.

3 Problema inverso

O problema de Ajuste de Histórico é um conhecido problema inverso na indústria de petróleo (Oliver et al., 2008). No problema direto, as propriedades físicas do reservatório são conhecidas e um simulador é utilizado para calcular o comportamento de produção do reservatório. No problema inverso, o objetivo é obter uma boa estimativa das propriedades físicas do reservatório, dado um conjunto de observações históricas sobre a produção do reservatório em estudo. O esquema dos problemas direto e inverso podem ser exemplificados pela Figura (1).

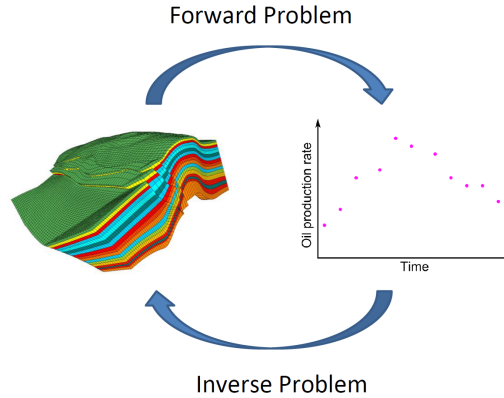


Figura 1: Ajuste de histórico: um problema inverso

Como as propriedades físicas do reservatório não podem ser diretamente medidas por toda sua extensão, o processo de ajuste de histórico é utilizado para estimar suas propriedades. Estas propriedades estimadas são utilizadas como parâmetros do simulador com o objetivo de prever a produção deste reservatório em diferentes cenários de produção.

Neste trabalho, o problema inverso proposto objetiva estimar a permeabilidade absoluta de um reservatório realizando o ajuste histórico de seus dados de produção, dados como a taxa de vazão de óleo e a pressão medida na boca do cano por onde o óleo é extraído em vários passos de tempo.

O objetivo é determinar o vetor de permeabilidades K . Deixe que $S(K)$ seja o vetor de dados simulados pelo método IMPES para dados valores K e $\bar{O}$ o vetor de dados historicamente observados. O problema consiste em encontrar K de forma a minimizar a formulação de mínimos quadrados descrita na Equação (12).

$$f(K) = \|S(K) - \bar{O}\|^2. \quad (12)$$

O problema a ser resolvido pode ser formulado como um problema de minimização irrestrito, como demonstrado na Equação (13).

$$f_x = \min_K f(K). \quad (13)$$

No contexto de Algoritmos Evolucionários, $f(K)$ é denominado como a *função de aptidão*. Neste trabalho, nós modelamos esta função de aptidão como uma função de medição de erro relativo, como demonstrado na Equação (14).

$$f(K) = \frac{\|S_o(K) - O_o\|^2}{\|O_o\|^2} + \frac{\|S_r(K) - O_r\|^2}{\|O_r\|^2}, \quad (14)$$

onde os parâmetros subscritos o e r denotam a vazão de óleo e a pressão medida na boca do cano de extração de óleo, respectivamente.

4 Algoritmos evolutivos

Algoritmos evolutivos tem como ideia básica o fato de que, na natureza, os indivíduos competem entre si por uma variedade de recursos naturais, tais como alimentos, água e abrigo. Entre os animais de uma mesma espécie, aqueles que não obtêm êxito nesta competição tendem a ter um número reduzido de descendentes e, conseqüentemente, uma menor probabilidade de terem seus genes propagados com o passar das gerações (Mitchell, 1998, Eiben and Smith, 2008).

Estes algoritmos utilizam uma analogia direta do fenômeno natural da evolução, em que cada indivíduo representa uma possível solução para o problema dado. Os principais conceitos que serão abordados nesta seção estão brevemente descritos a seguir (Mitchell, 1998).

- Indivíduo ou Cromossomo – conjunto de parâmetros que descrevem um indivíduo.
- Gene – representação de cada parâmetro de acordo com o alfabeto utilizado (binário, inteiro ou real).
- População – conjunto de indivíduos que formam uma geração.
- Geração – iteração completa do algoritmo genético que determina os indivíduos que irão compor a nova população.

Para cada indivíduo é atribuída uma pontuação referente a sua função aptidão, neste problema dado pela função descrita na Equação (14). Aos indivíduos mais adaptados é dada a oportunidade de reproduzir-se mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes (Eiben and Smith, 2008).

Um algoritmo evolutivo segue basicamente os passos descritos no Algoritmo 1. Primeiramente, uma população inicial é gerada com valores aleatórios dentro do espaço de busca da função objetivo. Em seguida, cada indivíduo tem calculado seu valor de aptidão. O processo de evolução tem início logo após, e consiste nos operadores de *seleção*, *cruzamento*, *mutação* e *evolução*. O processo de seleção consiste na escolha de alguns indivíduos da população sobre os quais atuarão os processos de cruzamento e mutação. O operador de cruzamento consiste na combinação de informações genéticas (genes) de dois ou mais indivíduos selecionados, dando origem a um novo indivíduo. O operador de mutação consiste em inserir uma alteração aleatória em um ou mais genes do novo indivíduo gerado. O operador de evolução define quais indivíduos serão levados para a próxima geração (Dumitrescu et al., 2000).

Algorithm 1 Algoritmo Evolutivo

- 1: *Inicialize* a população com indivíduos gerados aleatoriamente
 - 2: *Avalie* cada indivíduo
 - 3: **while** Condição de parada não satisfeita **do**
 - 4: *Selecione* os pais
 - 5: Realize o *cruzamento* entre os pais selecionados;
 - 6: Execute a *mutação* nos indivíduos gerados
 - 7: *Avalie* os novos indivíduos
 - 8: *Evolua* a população para uma nova geração
 - 9: **end while**
-

Algoritmos evolutivos se diferenciam de métodos matemáticos tradicionais de busca e otimização principalmente em quatro aspectos:

- Podem trabalhar com uma codificação do conjunto de parâmetros, e não com os próprios parâmetros.
- Trabalham com uma população e não só com um ponto único da solução.
- Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
- Utilizam regras de transição probabilística e não determinadas.

Estes aspectos fazem com que o algoritmo não se prenda facilmente em ótimos locais do problema, sempre tendendo a convergir para o ótimo global da função objetivo.

4.1 Representação dos indivíduos

Os indivíduos podem ter diversas representações e, sobre estas é que são realizadas as operações genéticas. A escolha de uma representação adequada é muito importante para o sucesso de um algoritmo evolucionário. Todo cromossomo representa um indivíduo candidato a solução do problema. As representações mais utilizadas para os cromossomos são a binária e a real (Eshelman and Schaffer, 1993, Janikow and Michalewicz, 1991). Neste trabalho, foi utilizada a representação real de um cromossomo. A representação real de um cromossomo pode ser observada na Figura 2, representando o vetor K . Cada posição do vetor é denominado como sendo um *gene*, e representa um valor de permeabilidade de uma fração do reservatório a ser estimada.

2	1.5	3.1	4.2	5.1	6.3	4.6	3.1	2.5	1.2	3.4	5.3
---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Figura 2: Representação do vetor K

$$x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}], i = (1, 2, \dots, Np) \quad (15)$$

Uma população é composta de Np diferentes indivíduos, como demonstrado na Equação (15) e são inicializados como se segue.

4.2 Inicialização da população

No processo de inicialização da população, para cada um dos Np indivíduos gerados, cada gene é inicializado aleatoriamente definido entre um conjunto de valores $\underline{x}_k$ e $\bar{x}_k$, como demonstrado pela Equação (16). Cada indivíduo possui n genes, sendo $n = |K|$.

$$x_{i,k} \in [\underline{x}_k, \bar{x}_k], k = (1, 2, \dots, n) \quad (16)$$

Uma vez que a população inicial é gerada, cada indivíduo é avaliado de acordo com a Equação (14) e sua aptidão é armazenada para posterior uso (Mitchell, 1998).

4.3 Seleção

O operador de seleção tem como função selecionar os indivíduos que vão se perpetuar para a geração seguinte. Os indivíduos selecionados sofrerão os operadores de cruzamento e mutação.

Indivíduos podem ser selecionados aleatoriamente dentro da população atual, ou algum método pode ser utilizado de forma a escolher preferencialmente indivíduos com um melhor valor de aptidão.

Vários métodos podem ser utilizados para a seleção. Podemos destacar o método da *roleta* (Tang et al., 1996), no qual os indivíduos são ordenados de acordo com sua aptidão e probabilidades decrescentes de serem escolhidos lhe são atribuídas, proporcionais a razão entre sua aptidão e a soma das aptidões de todos os indivíduos da população. Além da roleta, podemos destacar o método de *ranking* (Baker, 1985). O método de ranking é semelhante ao método da roleta, mas difere no sentido que a probabilidade de seleção é relacionada à sua posição na ordenação dos indivíduos na população e não a sua adequação, propriamente dita.

Existem ainda vários outros métodos de seleção na literatura (Goldberg and Deb, 1991, Bäck, 1994, Hancock, 1994).

4.4 Cruzamento e Mutação

O operador de cruzamento, juntamente com o operador de mutação, tem como principal objetivo explorar regiões desconhecidas do espaço de soluções do problema (Holland, 1992).

O operador de cruzamento atua misturando os genes de dois indivíduos selecionados, dando origem a um ou mais filhos (i.e. novos cromossomos). Uma lista de indivíduos selecionados é embaralhada criando-se, desta forma, uma lista de parceiros. Cada indivíduo selecionado na lista original é então cruzado com o indivíduo de mesmo índice na lista de parceiros. O operador de cruzamento é responsável por intensificar a busca em boas soluções já existentes na população atual (Dumitrescu et al., 2000).

Altera de forma aleatória um ou mais genes dos indivíduos gerados a partir do operador de cruzamento. Estas mutações são responsáveis por inserir uma maior variabilidade na população, impedindo uma estagnação prematura do algoritmo. O operador de mutação é responsável por diversificar a população, explorando novos espaços de soluções do problema inatingíveis pelo operador de cruzamento (Eiben and Smith, 2008).

Ambos os operadores atuam de forma estocástica, e estão sujeitos a uma probabilidade de sucesso, definidas como *probabilidade de cruzamento* (P_c) e *probabilidade de mutação* (P_m). Quanto maior a probabilidade de cruzamento, mais o algoritmo genético intensificará a busca por melhores soluções a partir das soluções atuais. Já a probabilidade de mutação controla a diversificação das soluções de um algoritmo evolutivo: um alto valor deste parâmetro resulta em uma grande diversificação das soluções dentro da população (Holland, 1992).

Para o sucesso de um algoritmo evolutivo é de vital importância um balanço entre intensificação e diversificação no processo de busca. Desta forma, uma escolha precisa para o valor destes dois parâmetros tem grande influência na performance e convergência do algoritmo.

4.5 Estratégias de evolução

Dentre os vários métodos de evolução apresentados na literatura, os mais comuns são o método geracional e o esquema *steady-state* (Dumitrescu et al., 2000).

O processo de evolução pode ser encarado como um processo de otimização, aproximando cada vez mais os valores obtidos pelo algoritmo ao valor da função objetivo. Alternativamente, um algoritmo evolutivo pode ser visto como um processo de adaptação, fazendo com que a cada geração os indivíduos se adaptem melhor ao ambiente em que estão inseridos (Holland, 1992).

No esquema de reprodução geracional, a população é completamente substituída ao final de cada geração. Este método tem como desvantagem a possibilidade de perda de bons indivíduos. Para contornar este fato, o elitismo pode ser utilizado, ou seja, os melhores indivíduos podem se perpetuar de uma geração para outra (Mitchell, 1998).

Já no esquema de evolução *steady-state*, somente uma parcela de indivíduos da população são criados a cada geração. Cada novo indivíduo é avaliado e inserido na população atual seguindo uma dada política de inserção. Várias políticas de inserções podem ser utilizadas. Pode-se, por exemplo, inserir um novo indivíduo na posição do indivíduo com pior valor de aptidão da geração anterior ou inserir o novo indivíduo no lugar do indivíduo presente a mais tempo da população atual (Mitchell, 1998, Eiben and Smith, 2008).

4.6 Condições de parada

Diferentes critérios de parada podem ser utilizados para interromper

Em nosso trabalho, três critérios de parada foram utilizados: ao atingir um número máximo de gerações estipuladas; quando não ocorrer melhora na população durante 20 gerações consecutivas; ou quando $f(x) < e$, onde e é uma tolerância prescrita.

5 Algoritmo Genético

Um Algoritmo Genético (AG) é uma variante de um Algoritmo Evolutivo, e seguem a ideia básica acima apresentada. A representação real de cromossomos é utilizada em nosso AG. A implementação é dada como se segue.

5.1 Seleção

O operador de seleção utilizado é a roleta. Ela torna-se uma boa opção de forma a ser possível escolher com maior probabilidade como pais os indivíduos de melhor aptidão, selecionando algumas vezes outros indivíduos da população. Na Tabela 1 fica evidenciado um exemplo do método de seleção por roleta em uma população com quatro indivíduos utilizando representação binária.

Na Tabela 1, P_d representa as probabilidades de distribuição de um cromossomo, Pa_d representa as probabilidades de distribuição acumulada, sendo dada pela Equação (17).

$$Pa_d = \sum_{z=1}^d P_z, d = 1, \dots, N_p \quad (17)$$

Para representação na roleta dos dados contidos na Tabela 1, multiplica-se a P_d de um cromossomo por 360, como pode ser observado na Figura 3.

Nº	Cromossomo	Aptidão(P_d)	P_d	Pa_d	Graus
1	101110110	10,0	0,500	0,500	180
2	001011001	5,0	0,250	0,750	90
3	111010101	2,5	0,125	0,875	45
4	100100010	2,5	0,125	1,000	45
Σ		20	1,00	1,000	360

Tabela 1: Seleção por roleta em um algoritmo genético

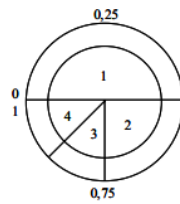


Figura 3: Exemplo de distribuição visualizado na roleta (Linden, 2008)

5.2 Cruzamento

O AG implementado utilizou o cruzamento de múltiplos pontos. Na Figura 4 podemos notar esta operação de cruzamento entre dois cromossomos, resultando em dois filhos.

Pais	2	1.5	3.1	4.2	5.1	6.3	4.6	3.1	2.5	1.2	3.4	5.3
	3.2	4.1	2.4	6.2	1.4	2.2	4.4	8.1	6.7	8.3	9.4	2.8
Filhos	3.2	1.5	2.4	6.2	5.1	6.3	4.4	8.1	6.7	1.2	3.4	5.3
	2	4.1	3.1	4.2	1.4	2.2	4.6	3.1	2.5	8.3	9.4	2.8

Figura 4: Ilustração do cruzamento entre dois indivíduos

O operador de cruzamento é acionado com uma probabilidade igual a P_c %.

5.3 Mutação

O operador de mutação é então aplicado aos cromossomos gerados pelo operador de cruzamento. Nossa implementação do AG utiliza um operador de mutação individual sobre cada gene.

Na Figura 5 podemos observar o processo de mutação em um cromossomo com representação real.

O operador de mutação é acionado com uma probabilidade de P_m % para cada gene do cromossomo alvo.

Pai	2	1.5	3.1	4.2	5.1	6.3	4.6	3.1	2.5	1.2	3.4	5.3
Filho	2	0.5	0.8	4.2	1	6.3	4.6	0.1	0.7	0.9	3.4	5.3

Figura 5: Ilustração da mutação de um cromossomo

6 Evolução diferencial

O algoritmo de Evolução Diferencial (ED) utiliza uma abordagem um pouco diferente dos algoritmos evolucionários tradicionais. O ED trata-se de um método de busca direta estocástica, desenvolvido a partir de tentativas de resolver o problema de ajuste polinomial de Chebychev. A ideia de usar diferenças de vetores para perturbar a população foi introduzida, resultando em um método que requer poucas variáveis de controle, robusto, de fácil utilização e de rápida convergência (Storn and Price, 1995, Price et al., 2005).

A ideia crucial por trás do ED é na geração de um novo indivíduo i pelo operador de mutação. A mutação realiza a adição da diferença vetorial ponderada entre dois indivíduos escolhidos aleatoriamente dentre a população a um terceiro indivíduo. Logo após, o indivíduo i gerado é comparado com um outro indivíduo j , também selecionado aleatoriamente dentre a população, e é gerado um novo indivíduo k a partir do operador de cruzamento aplicado aos i e j (Storn and Price, 1995).

O ED utilizado neste trabalho, assim como o AG, utiliza a representação real de indivíduos. A implementação é dada como se segue.

6.1 Seleção

O operador de seleção torna possível com que o ED produza filhos melhores. Ao contrário de outros algoritmos evolutivos, não existem procedimentos de seleção proporcional ou elitismo.

Após a determinação do valor de aptidão do vetor experimental, ele é então comparado ao vetor inicial. Deste modo, o vetor com melhor valor de aptidão é então escolhido para persistir para a geração $(q + 1)$ (Storn and Price, 1995, Price et al., 2005).

O processo de seleção é dado como se segue:

$$\begin{aligned} \text{If } f(U^{(q+1)}) &\leq f(x_s^{(q)}), & x_s^{(q+1)} &= U^{(q+1)} \\ \text{If } f(U^{(q+1)}) &> f(x_s^{(q)}), & x_s^{(q+1)} &= x_s^{(q)} \end{aligned} \quad (18)$$

6.2 Mutação

A mutação consiste em um operador de adição entre a diferença de dois indivíduos por um terceiro, contidos na mesma população. Em uma geração q , três indivíduos x_i , x_j e x_k são escolhidos aleatoriamente dentre a população, sendo

$$\alpha, \beta, \gamma \in [0, Np - 1], \quad \alpha, \beta, \gamma \in \mathbb{N}; \quad \alpha \neq \beta \neq \gamma \quad (19)$$

Um novo vetor de parâmetros $V^{(q+1)}$, denominado vetor doador, é gerado através da operação de mutação, que pode ser escrita matematicamente como nas Equações (20) e (21).

$$V^{(q+1)} = x_{\alpha}^{(q)} + F(x_{\beta}^{(q)} - x_{\gamma}^{(q)}) \quad (20)$$

$$V^{(q+1)} = x_{best}^{(q)} + F(x_{\beta}^{(q)} - x_{\gamma}^{(q)}) \quad (21)$$

onde F é denominado como *fator de perturbação*, um fator escalar pertencente ao intervalo $[0, 2]$, q representa a geração atual, $q + 1$ representa a próxima geração e $x_{best}^{(q)}$ o indivíduo com melhor aptidão da geração atual (Price et al., 2005).

O operador de mutação é acionado com uma probabilidade de Pm %. Da mesma forma que os fatores Pc e Pm , o fator de perturbação F tem grande influência na execução do ED, devendo ser precisamente escolhido.

6.3 Cruzamento

Considere que para cada vetor alvo $x_s, s \in 1, \dots, Np, s \neq \alpha \neq \beta \neq \gamma$ foi gerado um vetor doador. Assim, as componentes do vetor experimental $U^{(q+1)}$ são geradas conforme a Equação (22).

$$U_i^{(q+1)} = \begin{cases} V_i^{(q+1)}, & \text{se } r_i \leq Pc \\ x_{s,i}^{(q)}, & \text{se } r_i > Pc \end{cases}, i = 1, 2, \dots, n; \quad r_i \in [0, 1] \quad (22)$$

onde $V_i^{(q+1)}$ a i -ésima componente do vetor doador $V^{(q+1)}$, $x_{s,i}^{(q)}$ a i -ésima componente do vetor alvo x_s e $Pc \in [0, 1]$ representando a probabilidade de cruzamento entre os vetores $V_i^{(q+1)}$ e $x_{s,i}^{(q)}$, ou seja, é a probabilidade do vetor experimental herdar os valores das variáveis do vetor doador, devendo Pc ser fornecida pelo usuário. Como exemplo, quando $Pc = 1$, todas as componentes do vetor experimental virão do vetor doador $V^{(q+1)}$. Porém, quando $Pc = 0$, o vetor experimental terá todas suas variáveis vindo do vetor alvo $x_s^{(q)}$. Este cruzamento é definido como cruzamento binomial, devido aos experimentos binomiais independentes.

O operador de cruzamento é acionado com uma probabilidade igual a Pc %.

7 Paralelização

Algoritmos Evolucionários são implicitamente paralelos graças a independência da avaliação da função objetivo. Neste trabalho, foi desenvolvido um método de paralelização e distribuição por ilhas (Tomassini, 1999, Tomassini, 2006).

A ideia do modelo de ilhas é dividir uma população em várias subpopulações. Cada subpopulação é enviada a uma ilha. A paralelização se dá no fato de cada processador ser responsável pela execução do Algoritmo Evolucionário de cada ilha.

Este modelo, além de simplesmente paralelizar o algoritmo, consegue aumentar a variabilidade de indivíduos. Quando temos uma população única, um indivíduo bom, em um ótimo local, pode propagar suas características para os demais indivíduos da população, eliminando assim a diversidade de características da população. O modelo de ilhas desenvolvido reduz este

efeito, aumentando a diversidade dos indivíduos, já que cada subpopulação evolui separadamente das outras (Tomassini, 2006).

Entretanto, o mesmo problema de uniformização das características dos indivíduos pode ocorrer dentro de cada subpopulação. Para isto, um operador de migração foi utilizado. A cada n gerações, um indivíduo qualquer migra de uma ilha para outra ilha vizinha, favorecendo assim a variabilidade genética entre as populações (Tomassini, 2006). Este esquema de ilhas e migração pode ser exemplificado pela Figura 6.

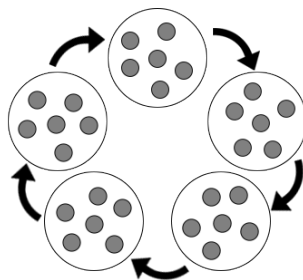


Figura 6: Esquema de migração de indivíduos entre ilhas vizinhas

8 Experimentos computacionais

Devido as características estocásticas dos algoritmos evolutivos, é necessário realizar uma série de testes com uma grande variação de parâmetros de cada algoritmo.

Para o AG, os valores de P_c , P_m e o tamanho da população foram variados, além da variação dos esquemas de evolução, entre os esquemas *steady-state* e geracional, gerando um conjunto de configurações iniciais a serem testadas. Na Tabela 2 pode-se observar os valores que cada parâmetro assumiu para a realização dos testes.

Para o ED, foram variados os valores de P_c e Fator de Perturbação (F), além do tamanho do *steady-state* (SS) e o tamanho da população, gerando um conjunto de configurações iniciais a serem testadas. Este conjunto foi testado em algoritmos de Evolução Diferencial que utilizam como abordagem de mutação a Equação (21), gerando o conjunto de testes final para este algoritmo. Na Tabela 3 pode-se observar os valores que cada parâmetro foi variado para a realização dos testes.

P_c	P_m	População
20%	5%	64
40%	10%	120
60%	15%	-
80%	20%	-
100%	-	-

Tabela 2: Parâmetros variados no AG

P_c	F	SS	População
20%	0,50%	12,5%	64
40%	0,60%	50%	120
60%	0,70%	80%	-
80%	0,80%	-	-
100%	-	-	-

Tabela 3: Parâmetros variados no ED

Cada teste foi realizado sempre variando um único parâmetro e mantendo todos os outros fixos, fazendo assim uma combinação de todos para todos, com todos os parâmetros variados.

Cada um destes conjuntos iniciais foi testado utilizando o modelo IMPES, executando um reservatório sintético com dois valores de permeabilidade a serem estimadas. Após a execução neste problema, o melhor conjunto de parâmetros obtidos através dos testes foi aplicado a outro reservatório sintético com quatro valores de permeabilidade a serem estimadas através do ajuste de histórico.

Por se tratarem de métodos estocásticos, é necessário repetir cada teste várias vezes, a fim de obter um resultado mais confiável e preciso. Desta maneira, cada item dos conjuntos de testes descritos foi executado dez vezes, tornando possível um estudo estatístico dos resultados, com a obtenção dos valores médios de número de iterações necessárias para convergência e aptidão, bem como o desvio padrão para cada conjunto de parâmetros.

Com base neste estudo estatístico, foi possível dizer qual a configuração de parâmetros inicial gera um melhor resultado para o problema a solução do problema de ajuste de histórico, bem como comparar os algoritmos para cada conjunto testado.

9 Resultados

Para dois valores de permeabilidade, tanto o AG quanto o ED se comportaram de maneira satisfatória, obtendo bons resultados para o ajuste de histórico. As Tabelas 4 e 5 exibem, respectivamente, o comportamento do AG e do ED com um variado conjunto de parâmetros, variando desde o pior até o melhor conjunto de parâmetros encontrado, quanto ao número de avaliações.

Pc	Pm	Melhor	Pior	Media	Desvio Padrão	Avaliações
60%	5%	9,006e-08	1,761e-05	3,935e-06	4,680e-06	23417
80%	5%	8,737e-07	7,517e-06	3,850e-06	2,076e-06	22022
60%	5%	2,380-e10	3,647e-06	3,173e-07	2,371e-07	19801
40%	10%	4,757e-11	1,145e-05	9,866e-05	1,286e-05	9406
40%	5%	2,201e-07	1,101e-05	2.845e-06	3,296-e06	8281

Tabela 4: Comportamento do AG para alguns conjuntos de parâmetros

Para ambos os algoritmos, todos os resultados apresentados são para uma população de 64 indivíduos. Para o AG, todos os resultados apresentados utilizam o esquema geracional.

Para o AG, o melhor conjunto de parâmetros encontrado foi uma Pc de 40% e uma Pm de 5%, utilizando o esquema geracional, sem a distribuição da população em ilhas.

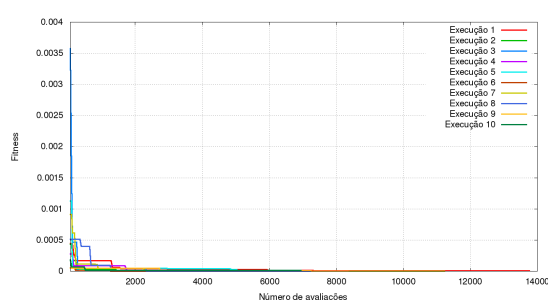
Para o ED, o melhor conjunto de parâmetros encontrado foi uma Pc de 80% e F de 0,5, além do tamanho de *steady-state* correspondente a 50% da população. A população do ED, no melhor conjunto de parâmetros, foi dividida em oito ilhas, contendo oito indivíduos cada, com migração a cada 10 gerações.

Pc	F	SS	Melhor	Pior	Media	Desvio Padrão	Avaliações
20%	70%	80%	1,964e-06	3,786e-05	1,159e-05	9,324e-06	4544
20%	50%	80%	1,745e-07	4,322e-05	1,187e-05	1,576e-05	3900
40%	60%	50%	2,962e-07	1,544e-06	7,167e-07	4,012e-07	2146
40%	50%	50%	6,401e-08	8,757e-07	5,361e-07	3,345e-07	1268
80%	50%	50%	7,887e-08	9,525e-07	4,881e-07	2,744e-07	1180,8

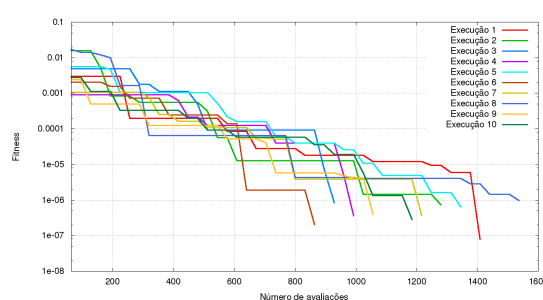
Tabela 5: Comportamento do ED para alguns conjuntos de parâmetros

Para ambos os algoritmos, valores médios para todos os parâmetros apresentam melhores resultados globais.

Nas Figuras 7(a) e 7(b) podemos observar a convergência do AG e do ED para o ajuste de histórico do IMPES com duas permeabilidades, comparando o número de avaliações com o melhor aptidão encontrado pelo algoritmo até o momento.



(a) Convergência dos testes do AG para o melhor conjunto de parâmetros encontrado



(b) Convergência dos testes do ED para o melhor conjunto de parâmetros encontrado

Figura 7: Convergência do AG e do ED para 2 valores de permeabilidade

Para o ajuste de histórico do IMPES com quatro permeabilidades não foi feito um novo estudo de parâmetros. O melhor conjunto de parâmetros obtidos na solução de problemas com duas permeabilidades foi utilizado para a solução do IMPES com quatro valores de permeabilidade.

Nas Figuras 8(a) e 8(b) podemos observar a convergência do AG e do ED para o ajuste de histórico do IMPES com quatro valores de permeabilidade, comparando o número de avaliações com a melhor aptidão encontrado pelo algoritmo até o momento.

O AG utilizando o esquema geracional, sem ilhas, demonstrou melhores resultados do que o AG utilizando o esquema *steady-state* com ilhas, devido a baixa variabilidade dos indivíduos em cada subpopulação no esquema de ilhas. O ED não demonstrou sofrer com uma baixa variabilidade, obtendo os melhores resultados com uma população de 64 indivíduos dividida em oito subpopulações de oito indivíduos cada.

No AG, valores mais elevados de P_m e de valores de P_c levavam o algoritmo a uma enorme diversidade de resultados, mas sem a intensificação da busca a partir uma boa solução já encon-

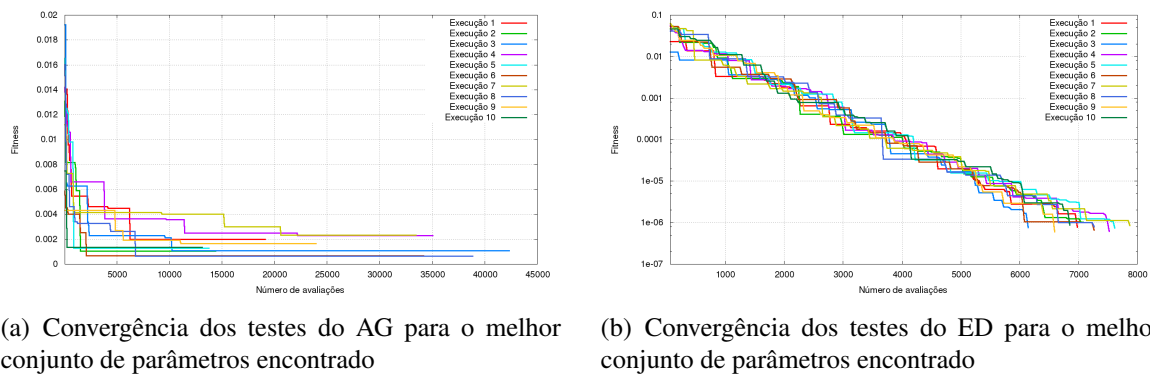


Figura 8: Convergência do AG e do ED para 4 valores de permeabilidade

trada. Valores muito baixos destes parâmetros diminuíam muito a diversidade do algoritmo, fazendo com que ele ficasse preso em ótimos locais.

No ED, assim como no AG, valores elevados do fator de perturbação F e de P_c não obtiveram bons resultados, devido a alta variabilidade dos indivíduos, e valores baixos fazem com que o algoritmo fique preso em ótimos locais e não consiga convergir para o ótimo global. O número do *steady-state* influenciou muito no número total de avaliações necessárias para a convergência do algoritmo, sendo que um número mais alto de *steady-state* implicou em um número maior de avaliações necessárias para a obtenção do resultado desejado.

É fácil perceber que o ED gastou um número muito menor de avaliações que o AG, tanto para dois quanto para quatro valores de permeabilidade. O ED convergiu em 100% dos testes com o melhor conjunto de parâmetros, enquanto o AG convergiu somente em 50% dos testes com o melhor conjunto de parâmetros para o ajuste de histórico do valor de duas permeabilidades e não convergiu em nenhum dos testes para quatro permeabilidades.

Para dois parâmetros, apesar de não ter convergido para o valor esperado em muitos casos, o AG converge rapidamente para um valor muito próximo do desejado com menos de 2000 avaliações, como pode ser observado na Figura 9.

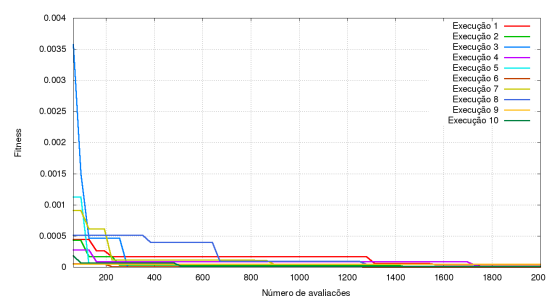


Figura 9: Corte no gráfico de convergência dos testes do AG para o melhor conjunto de parâmetros encontrado em duas permeabilidades

Já para quatro valores de permeabilidade a serem ajustados, o AG não conseguiu obter resultados satisfatórios, ficando estagnado em valores muito abaixo do desejado, como pode ser observado na Figura 8(a).

10 Conclusão

Neste trabalho foram implementados dois Algoritmos Evolutivos para realizar o ajuste de histórico de um simulador de reservatórios de petróleo, denominado IMPES.

Foram realizados inúmeros testes para o problema de ajuste de histórico do simulador utilizado, obtendo resultados satisfatórios..

Indica-se que para o ajuste de histórico sejam utilizados, para o ED, um fator de perturbação F de 0,50 e uma P_c de 80%, com a separação da população em ilhas. Para o AG, recomenda-se que utilize o esquema de reprodução geracional, com P_c de 40% e P_m de 5%.

Conclui-se que o método de paralelização em ilhas contribui para a diversidade da população, fazendo com que execuções de Algoritmos Evolutivos com populações pequenas possam escapar de ótimos locais, assim como Algoritmos Evolutivos com populações maiores, e ainda assim convergir rapidamente.

Ao analisar os resultados, fica claro que o ED se mostrou superior ao AG, para os conjuntos de parâmetros testados. O ED convergiu em 100% das execuções, tanto para dois quanto para quatro valores de permeabilidade a serem ajustados, enquanto o AG teve algumas dificuldades para convergir com dois valores de permeabilidade e não conseguiu obter resultados satisfatórios com quatro valores de permeabilidade.

Como trabalhos futuros, pode-se citar algumas modificações nos algoritmos desenvolvidos, para tentar diminuir o número médio de avaliações do ED e melhorar a convergência do AG.

Podem ser realizados testes no AG com um tamanho maior de população, para que ele não sofra com os ótimos locais quando utilizando o esquema de ilhas para divisão da população. Ainda sobre o AG, pode-se propor a implementação de outros métodos de seleção, como o Torneio, além de mudanças no esquema de evolução, substituindo todos os indivíduos a cada geração.

No ED, podem ser realizados testes utilizando outras fórmulas de mutação, como a Equação (20), ou outras presentes na literatura.

A maior contribuição deste trabalho foi uma comparação justa de ambos os algoritmos, demonstrando a superioridade do ED sobre o AG para problemas de ajuste de histórico com base em um grande estudo empírico dos parâmetros de ambos os algoritmos.

Agradecimentos

Os autores agradecem às agências de fomento: Capes, CNPq e FAPEMIG pelo apoio financeiro.

Referências Bibliográficas

- Bäck, 1994 **Bäck, T.** (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*; pages 57–62. IEEE.
- Baker, 1985 **Baker, J. E.** (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications*; pages 101–111. Hillsdale, New Jersey.
- Chen et al., 2004 **Chen, Z., Huan, G., and Li, B.** (2004). An improved impes method for two-phase flow in porous media. *Transport in Porous Media.*; pages 261–276.
- Dumitrescu et al., 2000 **Dumitrescu, D., Lazzerini, B., Jain, L. C., and Dumitrescu, A.** (2000). *Evolutionary computation*. CRC press.
- Eiben and Smith, 2008 **Eiben, A. E. and Smith, J. E.** (2008). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer.
- Eshelman and Schaffer, 1993 **Eshelman, L. J. and Schaffer, J. D.** (1993). Real-coded genetic algorithms and interval-schemata. *Foundation of Genetic Algorithms 2*; pages 187 – 202.
- Goldberg and Deb, 1991 **Goldberg, D. E. and Deb, K.** (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*; 1:69–93.
- Hancock, 1994 **Hancock, P. J.** (1994). An empirical comparison of selection methods in evolutionary algorithms. In *Evolutionary Computing*; pages 80–94. Springer.
- Holland, 1992 **Holland, J.** (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.
- Janikow and Michalewicz, 1991 **Janikow, C. Z. and Michalewicz, Z.** (1991). An experimental comparison of binary and floating point representations in genetic algorithms. In *ICGA*; pages 31–36.
- Kassem et al., 2006 **Kassem, J. H. A., Ali, S. M. F., and Islam, M. R.** (2006). *Petroleum Reservoir Simulation: A Basic Approach*. Gulf Publishing Company.
- Linden, 2008 **Linden, R.** (2008). *Algoritmos genéticos (2a edição)*. Brasport.
- Mitchell, 1998 **Mitchell, M.** (1998). *An Introduction to Genetic Algorithms*. The MIT Press.
- Oliver et al., 2008 **Oliver, D. S., Reynolds, A. C., and Liu, N.** (2008). *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press; 1

edition.

Price et al., 2005 **Price, K., Storn, R., and Lampinen, J.** (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. U.S. Government Printing Office.

Rosa et al., 1983 **Rosa, A., Horne, R., et al.** (1983). Automated type-curve matching in well test analysis using laplace space determination of parameter gradients. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Rosa et al., 1996 **Rosa, A. J., Horne, R. N., et al.** (1996). New approaches for robust nonlinear parameter estimation in automated well test analysis using the least absolute value criterion. *SPE Advanced Technology Series*; 4(01):21–27.

Santos et al., 2012 **Santos, E. P., Xavier, C. R., Campos, R. S., and Weber Dos Santos, R.** (2012). Comparison between genetic algorithms and differential evolution for solving the history matching problem. In *Computational Science and Its Applications – ICCSA 2012*; volume 7333 of *Lecture Notes in Computer Science*; pages 635–648. Springer Berlin Heidelberg.

Santos et al., 2009 **Santos, E. P., Xavier, C. R., Goldfeld, P., Dickstein, F., and Weber Dos Santos, R.** (2009). Comparing genetic algorithms and newton-like methods for the solution of the history matching problem. In *Proceedings of the 9th International Conference on Computational Science: Part I*; ICCS '09; pages 377–386; Berlin, Heidelberg. Springer-Verlag.

Soleng, 1999 **Soleng, H. H.** (1999). Oil reservoir production forecasting with uncertainty estimation using genetic algorithms. *Evolutionary Computation*.

Storn and Price, 1995 **Storn, R. and Price, K. V.** (1995). Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces.

Tang et al., 1996 **Tang, K.-S., Man, K.-F., Kwong, S., and He, Q.** (1996). Genetic algorithms and their applications. *Signal Processing Magazine, IEEE*; 13(6):22–37.

Tomassini, 1999 **Tomassini, M.** (1999). Parallel and distributed evolutionary algorithms: A review.

Tomassini, 2006 **Tomassini, M.** (2006). *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time*; volume 1 of *Natural Computing Series*. Springer.

Watson et al., 1994 **Watson, A. T., G., W. J., , and E., E. R.** (1994). Parameter and system identification for fluid flow in underground reservoirs. *Proceedings of the Conference, Inverse Problems and Optimal Design in Industry*.

Xavier et al., 2013 **Xavier, C. R., dos Santos, E. P., da Fonseca Vieira, V., and dos Santos, R. W.** (2013). Genetic algorithm for the history matching problem. *Procedia Computer Science*; 18(0):946 – 955.