UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

Deivid Edson Delarota Campos

MODELAGEM DA PRESSÃO DE FUNDO DE POÇO EM SISTEMAS DE ESCOAMENTO MULTIFÁSICO: UMA ABORDAGEM UTILIZANDO PROGRAMAÇÃO GENÉTICA

Juiz de Fora
2024

Deivid Edson Delarota Campos

MODELAGEM DA PRESSÃO DE FUNDO DE POÇO EM SISTEMAS DE ESCOAMENTO MULTIFÁSICO: UMA ABORDAGEM UTILIZANDO PROGRAMAÇÃO GENÉTICA

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional. Área de concentração: Modelagem Computacional

Orientador: Prof. Dr. Leonardo Goliatt da Fonseca

Coorientadora: Profa. Dra. Camila Martins Saporetti

Juiz de Fora

2024

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Campos, Deivid.
MODELAGEM DA PRESSÃO DE FUNDO DE POÇO EM SISTEMAS DE ESCOAMENTO MULTIFÁSICO: UMA ABORDAGEM UTILIZANDO PROGRAMAÇÃO GENÉTICA / Deivid Edson Delarota Campos. – 2024.
67 f. : il.

Orientador: Leonardo Goliatt da Fonseca
Coorientadora: Camila Martins Saporetti
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Modelagem Computacional, 2024.

1. Pressão de Fundo de Poço. 2. Programação Genética. 3. Modelagem Computacional. 4. Escoamento Multifásico. I. Goliatt, Leonardo, orient. II. Dourtor III. Saporetti, Camila, coorient. IV. Doutora.

Deivid Edson Delarota Campos

Modelagem da pressão de fundo de poço em sistemas de escoamento multifásico: uma abordagem utilizando programação genética

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional. Área de concentração: Modelagem Computacional

Aprovada em 07 de maio de 2024

BANCA EXAMINADORA

Prof. Dr. Leonardo Goliatt da Fonseca - Orientador

Universidade Federal de Juiz de Fora

Profa. Dra. Camila Martins Sapore - Coorientadora

Universidade de Estado do Rio de Janeiro

Prof. Dr. Heder Soares Bernardino

Universidade Federal de Juiz de Fora

Prof. Dr. Iury Higor Aguiar da Igreja

Universidade Federal de Juiz de Fora

Prof. Dr. Wanderlei Malaquias Pereira Junior

Universidade Federal de Catalão

Juiz de Fora, 29/04/2024.

Documento assinado eletronicamente por **Leonardo Goliatt da Fonseca**, **Professor(a)**, em 07/05/2024, às 21:33, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Iury Higor Aguiar da Igreja**, **Professor(a)**, em 07/05/2024, às 22:30, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Heder Soares Bernardino**, **Professor(a)**, em 08/05/2024, às 08:30, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Wanderlei Malaquias Pereira Junior**, **Usuário Externo**, em 08/05/2024, às 12:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Camila Martins Saporetti**, **Usuário Externo**, em 09/05/2024, às 09:19, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1793851** e o código CRC **40E897D3**.

À Camilo de Lelis Paiva Campos, meu amado pai (em memória) e à Maria Aparecida Delarota minha amada mãe.

AGRADECIMENTOS

A conclusão desta jornada acadêmica marca não apenas o término de um ciclo, mas também o início de uma profunda gratidão que transcende as páginas desta dissertação. Ao expressar meus sinceros agradecimentos, desejo reconhecer as muitas pessoas e instituições que tornaram possível este capítulo enriquecedor da minha vida acadêmica.

Primeiramente, expresso minha imensa gratidão aos meus orientadores, Leonardo Goliatt, e Camila Saporetti, cuja orientação sábia e apoio incansável foram fundamentais para a realização deste trabalho. Suas valiosas contribuições não apenas moldaram o conteúdo desta dissertação, mas também inspiraram meu crescimento acadêmico e pessoal.

À fonte inesgotável de amor e apoio, meus pais e irmãs, que sempre acreditaram em mim e proporcionaram a base sólida sobre a qual construí este caminho acadêmico. Sua dedicação e carinho foram luzes orientadoras em cada passo desta jornada.

À rede de apoio que é minha família, agradeço por serem as âncoras que me mantiveram firme diante dos desafios e as estrelas que celebraram cada conquista. Seu calor e encorajamento tornaram esta jornada inesquecível.

À minha namorada, e aos amigos que estiveram ao meu lado em todos os momentos, agradeço por trazerem alegria, compreensão e apoio constante. Sua presença tornou os desafios mais leves e as vitórias mais significativas.

Aos colegas de pesquisa, cuja paixão pelo conhecimento e colaboração foram inspiradoras, agradeço por compartilharem não apenas descobertas, mas também risos e aprendizados valiosos.

Por fim, não poderia deixar de agradecer à Universidade Federal de Juiz de Fora e ao Programa de Pós-Graduação em Modelagem Computacional pela infraestrutura e pela formação acadêmica. Também expresso minha gratidão à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 - pelo suporte financeiro.

Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças. (Leon C. Megginson)

RESUMO

A modelagem da pressão de fundo de poço em sistemas de escoamento multifásico representa um desafio complexo na indústria de petróleo e gás, dado seu impacto direto na eficiência e segurança das operações de produção. Apesar da extensa literatura existente, a aplicação de técnicas de aprendizado de máquina para este propósito permanece sub explorada. Este estudo adotou uma abordagem utilizando a Programação Genética para determinar a pressão de fundo de poço. Utilizando 795 amostras de dados relacionados a testes de produtividade de poços em campos no Oriente Médio, abrangendo variáveis como fluxo de óleo, fluxo de gás, fluxo de água, densidade do óleo, profundidade de perfuração, temperatura do fundo do poço e pressão na cabeça do poço, a estratégia baseada em Programação Genética foi aplicado para desenvolver modelos simbólicos interpretáveis. Esses modelos demonstraram habilidade em descrever, de forma compreensível, a complexa relação entre variáveis operacionais, ambientais e a pressão de fundo de poço. A obtenção de modelos simbólicos compreensíveis destaca a aplicabilidade prática da pesquisa, proporcionando uma compreensão mais profunda dos fatores que influenciam a pressão de fundo de poço e facilitando uma tomada de decisão mais informada por parte dos profissionais da indústria.

Palavras-chave: Pressão de Fundo de Poço, Programação Genética, Modelagem Computacional, Escoamento Multifásico.

ABSTRACT

Wellbore pressure modeling in multiphase flow systems poses a significant challenge in the oil and gas industry due to its direct impact on production efficiency and safety. Despite extensive literature, the application of machine learning techniques for this purpose remains underexplored. This study employs a Genetic Programming (GP) approach to predict wellbore pressure. Utilizing 795 data samples from well productivity tests in Middle Eastern fields, encompassing variables such as oil flow rate, gas flow rate, water flow rate, oil density, drilling depth, wellbore temperature, and wellhead pressure, the SGP-based strategy was applied to develop interpretable symbolic models. These models demonstrated the ability to comprehensibly describe the complex relationship between operational and environmental variables and wellbore pressure. The derivation of interpretable symbolic models highlights the practical applicability of the research, providing a deeper understanding of the factors influencing wellbore pressure and facilitating more informed decision-making by industry professionals.

Keywords: Well bottom-hole pressure, symbolic genetic programming, computational modeling, multiphase flow

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANN	Rede Neural Artificial (*Artificial Neural Network*)
API	Indicie de Gravidade Específica do Petróleo (*American Petroleum Institute Gravity*)
BPNN	Rede Neural de Retropropagação (*Backpropagation Neural Network*)
ELM	*Extreme Learning Machine*
FNN	Rede Neural Funcional (*Functional Neural Network*)
FBHP	Pressão de Fundo de Poço (*Flowing Bottom-hole Pressure*)
GFR	Taxa de Fluxo de Gás (*Gas Flow Rate*)
GMDH	Método de Agrupamento de Dados (*Group Method of Data Handling*)
GRNN	Rede Neural de Regressão Geral (*General Regression Neural Network*)
IC	Inteligência Computacional
ID	Diâmetro Interno do Tubo (*Internal Diameter of Pipe*)
KNN	k-Vizinhos mais Próximos (*k-Nearest Neighbors*)
LSSVM	Máquina de Vetores de Suporte por Mínimos Quadrados (*Least Squares Support Vector Machine*)
LSTM	Modelo de Memória de Longo Prazo (*Long Short-Term Memory*)
MARS	*Multivariate Adaptive Regression Splines*
MAE	Erro Médio Absoluto (*Mean Absolute Error*)
MAPE	Erro Percentual Médio Absoluto (*Mean Absolute Percentage Error*)
ML	Aprendizado de Máquina (*Machine Learning*)
MSE	Erro Quadrático Médio (*Mean Squared Error*)
OFR	Taxa de Fluxo de Óleo (*Oil Flow Rate*)
PG	Programação Genética
PSO	Otimização por Enxame de Partículas (*Particle Swarm Optimization*)
PTC	*Probabilistic Tree-Creation*
R	Coeficiente de correlação
R^2	Coeficiente de determinação
RF	Floresta Aleatória (*Random Forest*)
RMSE	Erro Quadrático Médio Residual (*Root Mean Squared Error*)
RBFNN	Rede Neural de Função de Base Radial (*Radial Basis Function Neural Network*)
SVR	*Support Vector Regression*
TOC	Teor de Carbono Orgânico (*Total Organic Carbon*)
XGB	*Extreme Gradient Boosting*
WBHT	Temperatura na cabeça do poço (*Wellbore Head Temperature*)
WFR	Taxa de Fluxo de Água (*Water Flow Rate*)
WHP	Pressão na Cabeça do Poço (*Wellhead Pressure*)
WPD	Produção Diária de Água (*Water Production Rate*)

LISTA DE SÍMBOLOS

$a(i,t)$	Aptidão ajustada do indivíduo i na geração t
b_n	Aridade não-terminal
d	Diâmetro da tubulação
E_{tree}	Tamanho esperado de uma árvore
f	Função
f_L	Fator de atrito do líquido
f_p	Aptidão associada a um programa p_i
f_{tp}	Fator de atrito para fluxo bifásico
F	Conjunto de funções
G_m	Taxa de fluxo de massa da mistura liquido-gás
g	Aceleração da gravidade
g_c	Constante de conversão igual a $32,174 Lb_m ft/lb_f seg^2$
H_L	Fração de retenção de líquido
i	Individuo
L_b	Comprimento da bolha de Taylor
L_s	Comprimento da lesma líquida
M	Massa total de óleo, água e gás associada a 1 bbl de líquido fluindo para dentro e para fora da coluna de fluxo
N	Número total de elementos testados
$n(i,t)$	Aptidão normalizada do indivíduo i na geração t
P	População
p_i	Programa associado a uma população P
q_f	Probabilidade de escolher uma função $f \in F$
q_τ	Probabilidade de escolher um terminal $t \in T$
q_l	Taxa total de produção líquida
ρ	Probabilidade de escolher um não-terminal
ρ_g	Densidade do gás
ρ_L	Densidade do líquido
ρ_M	Densidade da mistura
S	Tamanho máximo de uma árvore
$s(i,t)$	Aptidão padronizada do indivíduo i na geração t
$\sum$	Somatório
t	Geração
τ	Terminal
θ	Angulo horizontal
$\mathcal{T}$	Conjunto de terminais
v_m	Velocidade da mistura
v_{sg}	Velocidade superficial do gás
V	Velocidade média do fluido
V_M	Velocidade média da mistura

w_s	Probabilidade associada a cada árvore com s variando de 1 até S
y_i	Saídas corretas desejadas
$\hat{y}_i$	Saídas gera pelo programa
Z	Distância do fluxo axial
α	Termo de correção do perfil de velocidade
$\cup$	União
Δh	Diferença de profundidade
ΔP	Diferença de pressão
ΔP_f	Componente de atrito de ΔP
ΔP_{HH}	Componente hidrostática de ΔP
ΔP_{KE}	Componente da energia cinética de ΔP
ΔZ	Mudança de elevação
$\bar{p}_m$	Pressão média da mistura liquido-gás para incremento

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

A exploração e produção de petróleo e gás desempenham um papel crucial na economia, contribuindo para a geração de empregos, investimentos e receitas fiscais significativas (72, 36). Estudos realizados até 2020 revelaram que o petróleo bruto ainda representava cerca de 32% da produção energética mundial (6). Embora essas atividades promovam impactos positivos nas economias locais e nacionais, é importante reconhecer que a indústria também pode acarretar consequências adversas para o meio ambiente e a saúde humana, como a poluição do ar, da água, a degradação do solo e a emissão de gases de efeito estufa. Diante desse cenário, torna-se essencial avaliar a sustentabilidade de projetos de petróleo e gás, considerando tanto os benefícios quanto os impactos negativos. Nesse contexto, a indústria está passando por uma transição significativa incorporando elementos como inovação, responsabilidade social e conformidade com regulamentações ambientais. Isso significa que as empresas estão buscando soluções mais sustentáveis e responsáveis para a exploração e produção de petróleo e gás, levando em conta os impactos ambientais e sociais de suas atividades (21).

No contexto da exploração e produção de petróleo e gás, a pressão de fundo de poço (FBHP) desempenha um papel crucial para a integridade operacional e eficiência na produção (46, 58, 64). A precisão na estimativa da FBHP é fundamental, visando evitar sobrecargas no poço, reduzir vazamentos e mitigar impactos ambientais adversos. A medição precisa da FBHP não apenas facilita o dimensionamento adequado da bomba, assegurando uma produção eficiente de petróleo, mas também proporciona um conhecimento detalhado das capacidades de produção do poço. Essa compreensão aprofundada, por sua vez, influencia a escolha do método de elevação artificial, contribuindo para uma otimização contínua do processo operacional (3).

Poços de petróleo tipicamente produzem uma combinação de líquidos e gases que são transportados à superfície, com a distribuição de fases alterando-se conforme as variações de pressão ao longo do fluxo. Quando a pressão excede o ponto de bolha da fase líquida, especialmente no fundo do poço, ocorre um fluxo monofásico, composto apenas pela fase oleosa. Contudo, à medida que o petróleo ascende no poço vertical, a redução da pressão hidrostática provoca a liberação de gases da fase oleosa, resultando no fluxo multifásico de petróleo e gás (29, 33). O fluxo multifásico, caracterizado pela coexistência de duas ou três fases, como óleo, gás e água, pode iniciar a produção em qualquer fase da vida do poço (16), ganhando destaque em diversos campos científicos, como engenharia mecânica, civil, química e nuclear (35).

Durante o fluxo multifásico de petróleo e gás no poço, a variação da pressão é crítica. A pressão no fundo do poço impacta diretamente a transição entre fluxo monofásico e

multifásico, afetando a eficiência da produção. A previsão confiável da FBHP durante o fluxo multifásico é uma necessidade reconhecida na indústria petrolífera, sendo crucial para o desenvolvimento adequado de condicionamento de poços e sistemas de elevação artificial (7, 33).

A prática comum é atualmente a aferição inteligente de poços, na qual medidores de pressão de fundo de poço são permanentemente instalados para monitorar o Fundo do Poço. No entanto, esses instrumentos demandam calibração e manutenção frequentes para evitar falhas e leituras imprecisas (2, 4). Em procedimentos convencionais, a intervenção frequente para medir o FBHP é laboriosa, associada a riscos como interrupções de produção e perdas econômicas. Nesse contexto, informações em tempo real do FBHP são extremamente valiosas para os engenheiros de produção.

1.2 Revisão Bibliográfica

Vários pesquisadores propuseram abordagens convencionais e empíricas para prever a pressão de fundo de poço a partir de dados da superfície, explorando aspectos específicos relacionados aos gradientes de pressão e retenção de líquido, elementos cruciais para otimizar a eficiência e a operação desses sistemas. Hagedorn & Brown em (33) apresentam um estudo experimental realizado em um poço de 1.500 pés para analisar os gradientes de pressão durante o fluxo bifásico em tubulações de diferentes tamanhos. Foram conduzidos testes com ampla variação de taxas de fluxo de líquido, razões gás-líquido e viscosidades do líquido. A partir dos dados coletados, foram desenvolvidas correlações e equações que permitem a previsão dos gradientes de pressão para uma ampla variedade de tamanhos de tubulação, condições de fluxo e propriedades do líquido. A equação desenvolvida para prever gradiente de pressão é dada por:

$$144\frac{\Delta p}{\Delta h} = \bar{p}_m + \frac{f q_l^2 M^2}{2,9652 \times 10^{11} d^5 \bar{p}_m} + \bar{p}_m \frac{\Delta\left(\frac{v_m^2}{2g_c}\right)}{\Delta h} \tag{1.1}$$

Onde: ΔP é a diferença de pressão, Δh é a diferença de profundidade, $\bar{p}_m$ é a pressão média da mistura liquido-gás para incremento, f é o fator de atrito, q_l é a taxa total de produção líquida, M é a massa total de óleo, água e gás associada a 1 bbl de líquido fluindo para dentro e para fora da coluna de fluxo, d é o diâmetro da tubulação, v_m é a velocidade da mistura e g_c é a constante de conversão igual a $32,174 Lb_m ft/lb_f seg^2$.

Orkiszewski em (51) aborda a previsão de quedas de pressão em sistemas de tubulação vertical com fluxo bifásico, considerando diferentes regimes de fluxo, como anular, slug e misto. Foram comparados cinco métodos iniciais de previsão de quedas de pressão, sendo os mais precisos (*Duns-Ros* e *Griffith-Wallis*) selecionados para programação computacional e testados em 148 condições de poços. Embora nenhum método tenha sido

preciso em todas as condições, o método de *Griffith-Wallis* mostrou-se mais promissor, apesar de apresentar maior erro percentual em relação ao método de *Duns-Ros*.

Aziz & Govier em (11) apresentam um esquema simples e baseado em princípios mecânicos para calcular a queda de pressão em poços de petróleo e gás. Os autores descrevem e verificam o método proposto com dados de campo independentes, comparando as previsões para 48 poços com dados de campo e outros métodos. O esquema proposto para a estimativa da queda de pressão é então descrito, baseando-se na identificação do padrão de fluxo e na aplicação do balanço de energia mecânica. Suas previsões são comparadas com outros métodos (Método de Hagedorn & Brown em (33) e Método de Orkiszewski em (51)), concluindo que o esquema proposto é mais fundamentado no mecanismo de fluxo e fornece resultados pelo menos tão bons quanto os métodos com os quais foram comparados para a estimativa da queda de pressão em poços de petróleo e gás. As equações desenvolvidas para prever o gradiente de pressão são:

$$\Delta P = \Delta P_{HH} + \Delta P_{KE} + \Delta P_f \tag{1.2}$$

$$\Delta P_{HH} = \frac{g}{g_c}(\rho_g L_b + \rho_L L_s)\frac{\Delta Z}{L_b + L_s} \tag{1.3}$$

$$\Delta P_{KE} = \frac{\Delta V^2}{2\alpha g_c}\rho \tag{1.4}$$

$$\Delta P_f = \frac{2 f_L \rho_M V_M^2}{g_c d}\Delta Z \tag{1.5}$$

Onde: ΔP é a queda de pressão sobre pequenas mudanças de elevação ΔZ, ΔZ é a mudança de elevação para a qual ΔP_{HH} é calculado, ΔP é a componente hidrostática de ΔP, ΔP_{KE} é a componente da energia cinética de ΔP, ΔP_f é a componente de atrito de ΔP, g é a aceleração da gravidade, g_c é a constante de conversão igual a $32,174 Lb_m ft/lb_f seg^2$, ρ_g é a densidade do gás, L_b é o comprimento da bolha de Taylor, ρ_L densidade do líquido, L_s comprimento da lesma líquida, ΔZ é a mudança de elevação da qual ΔP é calculada, V é a velocidade média do fluido, α é o termo de correção do perfil de velocidade, f_L é o fator de atrito do líquido, ρ_M é a densidade da mistura, V_M é a velocidade média da mistura e d é o diâmetro da tubulação.

O estudo realizado por Beggs & Brill em (16) se concentra na previsão da queda de pressão e retenção de líquido em fluxo gás-líquido de duas fases em tubulações inclinadas. O estudo aborda desafios na previsão desses parâmetros em fluxo inclinado de duas fases. Experimentos foram conduzidos com tubulações transparentes, variando parâmetros como taxa de fluxo, pressão, diâmetro e ângulo de inclinação. O estudo desenvolveu correlações e equações para prever gradientes de pressão e retenção de líquido em várias condições de fluxo, destacando a importância dessas previsões para o design de equipamentos, como separadores gás-líquido, em poços e tubulações inclinadas. A equação desenvolvida para

prever o gradiente de pressão é dada por:

$$\frac{-dp}{dZ} = \frac{\frac{g}{g_c} sen(\theta)[\rho_L H_L + \rho_g(1 - H_l)] + \frac{f_{tp} G_m v_m}{2 g_c d}}{1 - \{[\rho_L + \rho_g(1 + H_L)] v_m v_{sg}\}/g_c p} \tag{1.6}$$

onde g é a aceleração devido à gravidade, g_c é a constante gravitacional, θ é o angulo horizontal, ρ_L é a densidade do líquido, ρ_g é a densidade do gás, H_L é a fração de retenção de líquido, f_{tp} é o fator de atrito para fluxo bifásico, G_m é a taxa de fluxo de massa da mistura liquido-gás, v_m é a velocidade da mistura, p é a pressão, d é o diâmetro da tubulação, v_{sg} é a velocidade superficial do gás e Z é a distância do fluxo axial.

A maioria dos modelos empíricos e correlações foram concebidas em escala laboratorial, o que compromete sua precisão quando aplicados a situações de campo (55). Estudos indicam que essas correlações empíricas apresentam elevados erros e incertezas. Asheim em (9) destaca que as correlações empíricas podem levar a erros significativos, como indicado pelos desvios padrão relativamente altos nas comparações entre as perdas de pressão medidas e calculadas. Além disso, ele discute que as correlações empíricas podem levar a erros relativamente grandes para certos casos, refletindo uma provável imprecisão nas medições. Para Gomez *et al.* em (28) as correlações empíricas comumente usadas para prever padrões de fluxo e queda de pressão em poços e dutos apresentam elevados erros em comparação com os dados experimentais. Segundo Pucknell *et al.* em (55) a maioria dos métodos tradicionais que funcionam razoavelmente bem em poços de petróleo fornecem previsões muito ruins para poços de gás. A variabilidade no desempenho das correlações empíricas pode ser extremamente alta, com alguns métodos fornecendo bons resultados em um campo e erros significativos em outro.

Nas últimas duas décadas, observou-se um aumento significativo nas aplicações de inteligência computacional (IC) em diversas áreas das geociências e engenharia de petróleo. A relevância da IC nessas aplicações surge da sua capacidade de lidar com os vastos volumes de dados gerados no campo, como dados sísmicos, registros petrofísicos de poços e dados de injeção e produção. Cada movimento no registro e cada falha nos dados têm significado e potencial para solucionar problemas relevantes. Além disso, as suposições básicas utilizadas na derivação das equações físicas podem ser violadas devido a diversas razões, como anisotropia, heterogeneidade, não linearidade, não elasticidade e comportamento não ideal do fluido. A IC demonstra habilidade em contornar essas complexidades de maneira perspicaz, explorando as informações e relações entre elas de forma eficiente.

Técnicas avançadas de aprendizado de máquina (*Machine Learning* - ML), tem impulsionado significativamente a eficácia desses modelos na interpretação de dados históricos em diversas aplicações na engenharia de petróleo (57, 22). Sua aplicação prática tem sido evidenciada em praticamente todos os setores, apresentando vastas oportunidades para crescimento e inovação (37, 68). Alguns exemplos são, previsões do teor de carbono

orgânico (TOC) (27, 60), avaliação e otimização da produtividade (19), determinação da litologia de poços de petróleo (59, 45, 69), fator de compressibilidade do gás (48), previsão de saturação de água (61, 1), previsões de precipitação de asfaltenos e ceras (56, 15), estimativa de parâmetros de teste de bombeamento (8), estimativa de registros petrofísicos ausentes (64, 63) e identificação de unidades de fluxo hidráulico (62).

A evolução das aplicações de IC na engenharia de petróleo oferece uma base sólida para explorar a interseção dessas técnicas com um componente crítico no contexto da exploração e produção de petróleo e gás. A capacidade da IC em lidar com a complexidade dos dados gerados no campo, como registros sísmicos, dados petrofísicos de poços, informações de produção, extrair conhecimento de dados brutos, lidar com tarefas não lineares, acomodar dados defeituosos de forma tolerante a falhas e oferecer generalizações eficazes (32, 43), proporciona um terreno fértil para aprimorar a precisão da estimativa da FBHP. À medida que as inovações na IC impactam diversos domínios da engenharia de petróleo, é imperativo investigar como essas técnicas podem contribuir para a melhoria da estimativa da FBHP, garantindo assim a integridade operacional e a eficiência na produção de petróleo e gás.

A previsão da pressão de fundo de poço em poços de petróleo é um desafio complexo que tem sido abordado por meio de diversas estratégias inovadoras. Dentre essas abordagens, destaca-se o modelo proposto por Jahanandish *et al.* em (35) que desenvolveu um modelo de Rede Neural Artificial (*Artificial Neural Network* - ANN) destinado à previsão da pressão de fundo de poço em poços de fluxo multifásico vertical. O objetivo era superar as limitações de generalidade e precisão inerentes às correlações e modelos mecanísticos disponíveis. Utilizando um conjunto de dados com 413 amostras coletados em diversos campos do Irã, o modelo ANN foi desenvolvido e testado após a divisão dos dados em conjuntos de treinamento, validação e teste na proporção de 4:1:1. Os resultados surpreenderam, demonstrando desempenho excepcional ao superar correlações empíricas e modelos mecanísticos amplamente utilizados na indústria. A análise de tendência revelou que o modelo foi capaz de prever com precisão os efeitos esperados das variáveis independentes na pressão de fundo do poço, indicando uma simulação eficaz do processo físico real. Além disso, a avaliação do erro do grupo utilizando o erro percentual médio absoluto (Mean Absolute Percentage Error - MAPE) confirmou a superioridade do ANN desenvolvido em relação aos modelos existentes, proporcionando previsões com aproximadamente 3,5% de erro percentual médio absoluto e um coeficiente de correlação (R) notável de 0,9222.

Outra contribuição relevante foi feita por Awadalla & Yousef em (10) que propuseram uma abordagem para antecipar a FBHP em poços de petróleo verticais, visando desenvolver uma ferramenta confiável para estimar a queda de pressão em poços de fluxo multifásico, utilizando dados históricos de diversos campos petrolíferos. A metodologia empregada utiliza redes neurais de alimentação direta (*Feedforward Neural Network* -

FFNN) com o algoritmo de retro propagação para aprimorar a previsão da pressão no fundo do poço. Os dados utilizados foram coletados de três campos de petróleo distintos, empregando 12 variáveis de entrada e a FBHP para treinar e avaliar o modelo FFNN. Ao comparar o desempenho do modelo FFNN com outros modelos empíricos, observou-se que o modelo FFNN, especialmente aquele com duas camadas ocultas, apresentou superioridade em termos de precisão e confiabilidade. Os resultados destacaram a capacidade do modelo FFNN de prever a pressão no fundo do poço com uma precisão entre 90% e 95%. Além disso, o modelo identificou de maneira significativa as variáveis de entrada mais relevantes para a previsão da pressão no fundo do poço.

Diante da complexidade do fluxo multifásico e da escassez de dados precisos em tempo real, Tariq *et al.* em (65) propõe um modelo inovador e robusto utilizando ANN, otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) para prever a pressão de fundo de poço em tempo real, com foco em poços verticais com fluxo multifásico. Dividindo o conjunto de dados em 70% para treinamento e 30% para teste, o estudo empregou métricas de avaliação como o MAPE e o coeficiente de determinação (R^2). Os resultados obtidos destacam a superioridade do modelo em relação aos métodos convencionais, apresentando um MAPE reduzido, abaixo de 2,1%, e um elevado valor de R^2, indicando uma captura efetiva da variação na pressão de fundo do poço pelas variáveis de entrada. Adicionalmente, a identificação dos parâmetros de entrada mais relevantes, como profundidade de perfuração, taxa de fluxo de óleo, gás, água, diâmetro interno da tubulação de produção, temperatura da superfície, temperatura do fundo do poço, gravidade do óleo e pressão da cabeça do poço, destaca a utilidade do modelo na identificação de problemas de produção e no planejamento de operações de remediação.

Sami & Ibrahim em (58) propõem o uso redes neurais artificiais, floresta aleatória (*Random Forest* - RF) e k-vizinhos mais próximos(*k-Nearest Neighbors* - KNN), para prever a pressão de fundo de poço em poços de petróleo e gás. Os dados foram coletados de fontes abertas e consistem em 206 pontos de dados de fluxo multifásico de poços verticais do Oriente Médio. O estudo realizou uma seleção de recursos cuidadosa para reduzir o número de pontos de dados e melhorar a qualidade dos dados. Além disso, o estudo realizou uma análise de correlação para identificar as variáveis independentes mais importantes que afetam a pressão de fundo de poço. Os resultados mostraram que a rede neural artificial foi a técnica de aprendizado de máquina mais eficaz para prever a pressão de fundo de poço em poços de petróleo e gás, com um R^2 de 0,96 para o conjunto de teste.

Marfo *et al.* em (46) adotou a abordagem M5 prime para desenvolver um modelo de previsão de pressão no fundo do poço. Utilizando dados de campo provenientes de um poço de petróleo situado em Gana, o estudo incluiu uma avaliação da qualidade dos dados para filtrar outliers. Os dados foram pré-processados e particionados com base na conhecida abordagem de validação cruzada hold-out. Durante a fase de treinamento do modelo, foi realizado um monitoramento contínuo utilizando o MAPE como critério de

avaliação. Adicionalmente, o estudo comparou os resultados do modelo M5 prime com outras técnicas amplamente utilizadas de aprendizado de máquina, como Rede Neural de Retro propagação (*Backpropagation Neural Network* - BPNN), Rede Neural de Regressão Geral (*General Regression Neural Network*- GRNN), Rede Neural de Função de Base Radial (*Radial Basis Function Neural Network* - RBFNN), Máquina de Vetores de Suporte por Mínimos Quadrados (*xLeast Squares Support Vector Machine* - LSSVM) e Método de Agrupamento de Dados (*Group Method of Data Handling* - GMDH). Os resultados obtidos indicaram que o modelo M5 prime apresentou o melhor desempenho em termos de previsão de pressão no fundo do poço, superando as demais técnicas de aprendizado de máquina avaliadas. Além disso, a pesquisa identificou a pressão na cabeça da coluna de produção e a taxa de produção de petróleo como as variáveis mais influentes no modelo de previsão de pressão no fundo do poço.

Al Shehri *et al.* em (5) contribui significativamente ao apresentar uma solução para a previsão precisa da FBHP em poços de gás condensado não convencional. A precisão nessa previsão desempenha um papel crucial na produtividade do poço e na construção de curvas de desempenho de elevação vertical, essenciais para a identificação dos tempos de carregamento de líquidos e a previsão da necessidade de instalação de cordas de velocidade ou sistemas de elevação artificial distintos. Os autores utilizaram ANN, FNN e modelos de memória de longo prazo (*Long Short-Term Memory* - LSTM) para prever a pressão de fundo de poço, empregando mais de 30.000 pontos de dados provenientes de diversos poços com uma ampla gama de entradas. Os dados de entrada incluíram informações diárias de fluxo, como pressão de cabeça do poço, temperatura de cabeça do poço, profundidade vertical verdadeira, gravidade específica do fluido, teor de cloreto, corte de água, taxas de fluido e relação água-gás. Os modelos foram treinados e testados em novos poços para avaliar sua precisão. Os resultados revelaram que o modelo ANN destacou-se como o mais preciso em comparação com as correlações empíricas existentes, apresentando um MAPE de 0,0704% e um R de 0,999.

O estudo realizado por Nwanwe *et al.* em (50) introduziu um modelo matemático visível de ANN. Sua característica visível, permite a extração dos pesos e viés das três camadas ocultas do modelo ANN treinado. Isso possibilita que o modelo desenvolvido seja utilizado pelos usuários sem que seja necessário um framework de aprendizado de máquina, como o Neural Network and *fuzzy* toolbox no MATLAB ou outras ferramentas similares em Python. A compreensão e interpretação do modelo são facilitadas, tornando as previsões mais transparentes e compreensíveis para os usuários. Desenvolvido com base no algoritmo de otimização de Levenberg-Marquardt e na função de ativação tangente hiperbólica, o modelo utilizou 1001 pontos de dados de campo em tempo real, sendo divididos aleatoriamente em conjuntos de treinamento, validação e teste (70%, 15%, 15%, respectivamente). A análise estatística indicou que a estrutura de rede de 3 camadas ocultas com 20, 15 e 15 neurônios obteve respectivamente o melhor desempenho, com

um fator de desempenho relativo de 0,290, Erro Quadrático Médio Residual (*Root Mean Squared Error* - MSE) de 0,047 e R de 0,903. A análise gráfica de erros demonstrou que o modelo proposto apresentou um desempenho superior em relação a correlações empíricas existentes e modelos mecanicistas tradicionais, com 100% dos pontos plotados dentro das linhas de desvio de ±20%, sendo considerado um desempenho ótimo em termos de precisão na previsão de pressão de fundo de poço em fluxo multifásico.

Por fim, Goliatt *et al.* em (26) adota uma perspectiva única ao combinar algoritmos de seleção de recursos e modelos de aprendizado de máquina. Utilizando dados de 206 amostras de poços de petróleo com processos de elevação artificial, o modelo incorpora nove variáveis relacionadas à produção, como fluxo de óleo, fluxo de gás, fluxo de água, densidade do óleo, profundidade de perfuração do poço, temperatura do fundo do poço e pressão na cabeça do poço. Os resultados desse estudo revelaram que o modelo, empregando quatro diferentes algoritmos de ML (*Multivariate Adaptive Regression Splines* (MARS), *Extreme Gradient Boosting* (XGB), *Extreme Learning Machine* (ELM) e *Support Vector Regression* (SVR)), alcançou um desempenho notável. O MARS, em particular, destacou-se com métricas impressionantes, incluindo coeficiente de correlação (R) = 0,94, coeficiente de determinação (R^2) = 0,88, erro quadrático médio residual (*Root Mean Squared Error* - RMSE) = 97,88, erro médio absoluto (*Mean Absolute Error* - MAE) = 74,69 e erro percentual médio absoluto (MAPE) = 3,12%. Além disso, a flexibilidade do modelo em gerar diversas alternativas com diferentes conjuntos de variáveis de entrada oferece vantagens práticas, especialmente em situações em que informações do poço podem estar indisponíveis devido a falhas de sensores ou problemas de comunicação.

A literatura revela resultados promissores ao empregar técnicas de ML na previsão da FBHP (Tabela 1.1), estabelecendo assim uma base sólida que motiva a realização de novas pesquisas. Diante desse cenário, este estudo busca ir além, adentrando no domínio da Programação Genética (PG) como uma abordagem inovadora para aprimorar as previsões de FBHP. Na próxima seção, exploraremos as origens e a aplicação dessa metodologia, destacando seu potencial contributivo neste desafio complexo.

Tabela 1.1 – Resumo do levantamento dos trabalhos relacionados.

Autor	**Ano**	**Técnica Utilizada**	**Resultados**
Jahanandish *et al.* em (35)	2011	Rede Neural Artificial (ANN)	MAPE de 3,5%, R de 0,9222
Awadalla & Yousef em (10)	2016	Redes Neurais de Alimentação Direta (FFNN)	Precisão entre 90% e 95%
Tariq *et al.* em (65)	2020	Rede Neural Artificial com PSO	MAPE abaixo de 2,1%
Sami & Ibrahim em (58)	2021	Redes Neurais Artificiais, Floresta Aleatória, K-Vizinhos	ANN foi a técnica mais eficaz com R^2
Marfo *et al.* em (46)	2022	M5 Prime, BPNN, GRNN, RBFNN, LSSVM, GMDH	O M5prime foi o melhor desempenho comparativo nas técnicas de aprendizado
Al Shehri *et al.* em (5)	2022	ANN, FNN, LSTM	ANN foi a técnica mais eficaz com R de 0,999 e MAPE de 0,0704%
Nwanwe *et al.* em (50)	2023	Modelo matemático visível de ANN	MSE de 0,047 e R de 0,90%
Goliatt *et al.* em (26)	2023	MARS, XGB, ELM, SVR	MARS destacou-se com R de 0,94, R^2 de 0,88, MAPE de 3,12%

1.3 Objetivo

O objetivo deste estudo é determinar a Pressão de Fundo de Poço (FBHP) utilizando a Programação Genética (PG) aplicada a dados provenientes de poços de petróleo, considerando variáveis ambientais e operacionais.

1.3.1 Objetivo Específico

- Gerar um modelo simbólico que possa ser compreendido e interpretado pelos profissionais da indústria.

2 Programação Genética

2.1 Origens da Programação Genética

A programação genética (PG) é uma abordagem na área de inteligência artificial que se originou na década de 1990. E se popularizou devido aos livros publicados por John Koza (38, 39), um pesquisador e professor da Universidade de Stanford. A PG tem suas raízes na tentativa de encontrar soluções mais eficazes para a criação automática de programas de computador.

Na década de 1990, a programação evolutiva já era uma área de pesquisa estabelecida, inspirada pelos princípios da seleção natural de Charles Darwin. A ideia básica era usar algoritmos evolutivos para evoluir soluções de problemas complexos, tratando o código de computador como ”gene” que poderiam ser combinados, mutados e selecionados para melhorar ao longo do tempo (13, 14).

No entanto, a programação evolutiva enfrentava desafios significativos quando se tratava de lidar com programas complexos. A representação direta do código-fonte como cadeias de bits binários, comumente usada na programação evolutiva, tinha limitações na expressividade e na manipulação de estruturas de dados complexas (38).

Foi nesse cenário que a programação genética surgiu como uma extensão e evolução dos algoritmos genéticos propostos por John Holland em (34). O conceito central por trás da PG era representar programas como árvores de sintaxe, semelhantes às árvores de análise sintática, onde uma estrutura de dados em árvore, que representa a estrutura sintática de uma cadeia de acordo com alguma gramática formal.(39).

A PG permite que os programas sejam construídos de maneira hierárquica, com operadores e operandos representados como nós na árvore. Isso possibilita a geração de programas mais complexos e expressivos. Os algoritmos evolutivos são então aplicados para evoluir esses programas, realizando cruzamentos, mutações e seleções para melhorar as soluções ao longo do tempo (24).

Os benefícios da PG incluem a capacidade de criar automaticamente programas para resolver problemas complexos, como otimização, controle e aprendizado de máquina. Seu uso tem sido estendido a problemas de diversas áreas do conhecimento, como por exemplo: biotecnologia, engenharia elétrica, análises financeiras, processamento de imagens, reconhecimento de padrões, mineração de dados, linguagem natural, etc. (67).

A Programação Genética busca a evolução de programas de computador visando o aprendizado por indução. A ideia é ensinar os computadores a se programarem a partir de especificações de comportamento. Cada programa possui um valor de mérito (aptidão) associado, refletindo sua capacidade de resolver o problema. O processo envolve a manutenção de uma população de programas, seleção dos melhores com base na adaptabilidade,

aplicação de operadores genéticos para modificação e convergência para uma solução. O objetivo é encontrar uma solução no espaço de todos os programas possíveis(13, 24).

O mecanismo de busca opera em um ciclo ”criar-testar-modificar” (Figura 2.1), assemelhando-se ao processo humano de desenvolvimento de programas. Inicialmente, programas simbólicos são gerados com base no conhecimento do domínio, representando soluções potenciais para o problema como árvores de sintaxe. Em seguida, esses programas são avaliados quanto à sua aptidão, utilizando uma função que mede seu desempenho em relação à solução desejada. Com base nos resultados, os programas são modificados por meio de operadores genéticos, como recombinação e mutação, introduzindo variações para explorar novas possibilidades. Esse ciclo iterativo é repetido até que uma solução satisfatória seja encontrada, seguindo uma abordagem de refinamento contínuo que reflete práticas eficazes de desenvolvimento humano de programas (70).

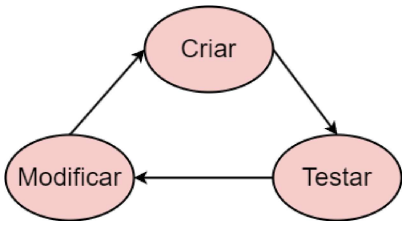

Figura 2.1 – Ciclo ”criar-testar-modificar”.
Fonte: Elaborada pelo autor (2024).

A especificação do comportamento na Programação Genética geralmente se realiza por meio de um conjunto de valores de entrada-saída conhecidos como *fitness cases*, os quais representam o conjunto de aprendizagem ou treinamento (*training set*). Com base nesse conjunto, a PG busca derivar um programa que:

- Produza de maneira não trivial as saídas corretas para cada entrada fornecida, evitando a simples atribuição por meio de uma tabela de conversão. Assim, o programa deve necessariamente aprender um algoritmo específico.
- Calcule as saídas de tal forma que, se as entradas forem selecionadas de forma representativa, o programa será capaz de produzir saídas corretas para entradas não cobertas inicialmente.

Por lidar diretamente com a manipulação de programas, a PG enfrenta o desafio de manusear uma estrutura relativamente complexa e variável. Tradicionalmente, essa estrutura assume a forma de uma árvore de sintaxe abstrata, composta por funções em seus nós internos e por terminais em seus nós folha. A especificação do domínio do problema é simplificada pela definição dos conjuntos de funções e terminais (38).

2.2 Visão Geral do Algoritmo de Programação Genética

As etapas do algoritmo de PG podem ser descritas resumidamente como (71):

1. *Inicialização da População:*

   Inicia-se gerando aleatoriamente uma população inicial de programas. Esses programas podem ser representados na forma de árvores de sintaxe abstrata, compostas por funções e terminais.

2. *Avaliação (aptidão):*

   Cada programa na população é avaliado utilizando uma função heurística, denominada função de aptidão. Essa função quantifica o desempenho de cada programa em relação ao objetivo do problema. Programas que produzem resultados mais próximos ou ideais têm uma pontuação de aptidão mais alta.

3. *Seleção:*

   Os programas são selecionados com base em seus valores de aptidão. Estratégias de seleção podem variar, mas geralmente programas com melhor desempenho têm maior probabilidade de serem escolhidos para reprodução.

4. *Operadores Genéticos:*

   Os operadores genéticos, como reprodução, cruzamento (recombinação) e mutação, são aplicados aos programas selecionados. A reprodução envolve a cópia direta de programas para a próxima geração, o cruzamento mistura partes de dois programas para criar novos, e a mutação introduz aleatoriamente alterações nos programas.

5. *Geração de Nova População:*

   A nova geração de programas resultante dos operadores genéticos forma a próxima população.

6. *Critério de Término:*

   O processo iterativo continua até que um Critério de Término seja satisfeito. Isso pode ser alcançar uma solução satisfatória, atingir um número máximo de gerações, ou qualquer outra condição específica ao problema.

7. *Retorno da Melhor Solução:*

   Após a conclusão do algoritmo, a melhor solução encontrada é retornada como resultado.

Cada execução desse loop representa uma nova geração de programas. O Critério de Término é tradicionalmente definido como alcançar uma solução satisfatória ou atingir um

número máximo de gerações. No entanto, abordagens alternativas consideram a análise do processo evolutivo, permanecendo no loop enquanto houver melhoria na população (Figura 2.2) (39). Essa flexibilidade na definição do Critério de Término destaca a adaptabilidade do algoritmo de PG às características específicas de cada problema.

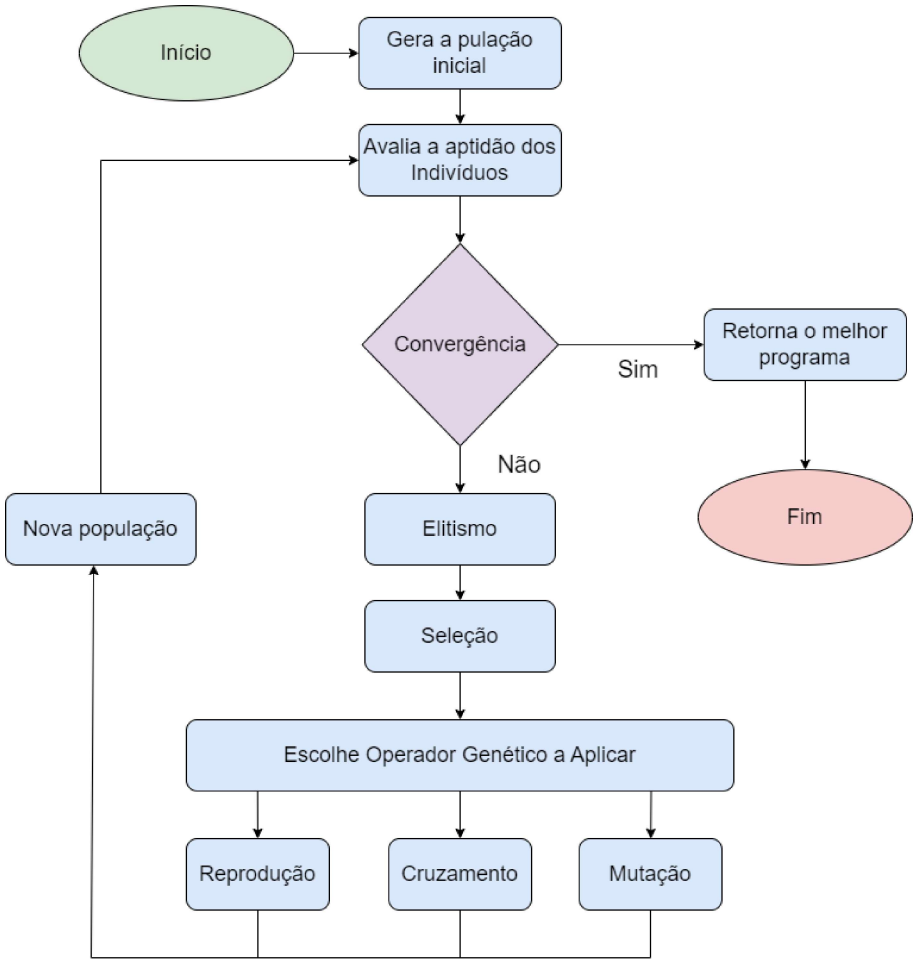

Figura 2.2 – Estrutura Básica do Algoritmo de Programação Genética.
Fonte: Elaborada pelo autor (2024).

2.3 Representação dos Programas

A PG representa tradicionalmente programas por meio de árvores de sintaxe abstrata, onde os programas são construídos através da livre combinação de funções e terminais (nós-folha das árvores e representam os valores finais ou básicos que compõem os programas) específicos para o domínio do problema (23).

Essa abordagem envolve dois conjuntos fundamentais: F, que consiste no conjunto de funções, e T, que representa o conjunto de terminais. O conjunto F pode incluir operadores aritméticos (+, -, *, etc.), funções matemáticas (seno, log, etc.), operadores lógicos (E, OU, etc.), dentre outros. Cada função $f \in F$ possui uma aridade (número de

argumentos) maior que zero. O conjunto T é composto por variáveis, constantes e funções de aridade zero (sem argumentos) (13, 41, 47, 54).

Por exemplo, considerando o conjunto de operadores aritméticos de aridade dois (2) como o conjunto de funções, e a variável y e a constante três (3) como terminais, temos:

$$\begin{aligned} F &= \{+, -, \times, \div\} \\ T &= \{y, 3\} \end{aligned} \tag{2.1}$$

então expressões matemáticas simples tais como $y * y + 3$ podem ser produzidas e sua representação é feita por uma árvore de sintaxe abstrata como mostrado na Figura 2.3.

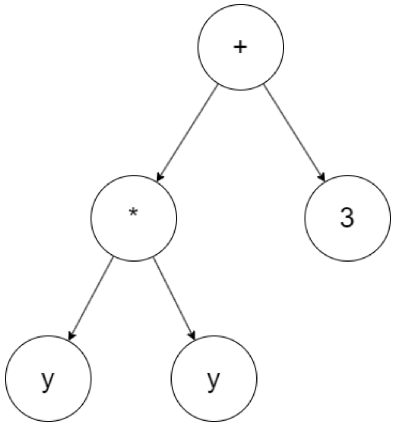

Figura 2.3 – Árvore de Sintaxe Abstrata de $y * y + 3$.
Fonte: Elaborada pelo autor (2024).

2.4 Fechamento e Suficiência

Para assegurar a viabilidade das árvores de sintaxe abstrata, John Koza (39) introduziu o conceito de Fechamento (closure). A propriedade de Fechamento estipula que cada função no conjunto F deve aceitar qualquer valor que possa ser retornado por qualquer função ou terminal, garantindo assim que todas as árvores geradas possam ser avaliadas corretamente.

Um exemplo típico que ilustra a necessidade do Fechamento é a operação de divisão, onde matematicamente não é possível dividir por zero. Uma solução para esse problema é a definição de uma função alternativa, como a função de divisão protegida (%) (38). Essa função, ao receber dois argumentos, retorna o valor 1 em caso de divisão por zero e, caso contrário, retorna o quociente normal.

Além disso, para assegurar a convergência para uma solução, Koza estabeleceu a propriedade de Suficiência. Essa propriedade requer que os conjuntos de funções F e terminais T sejam capazes de representar uma solução para o problema em questão. Em outras palavras, deve haver evidências sólidas de que alguma combinação de funções e

terminais pode produzir uma solução, podendo essa propriedade variar em sua obviedade dependendo do problema em análise, podendo exigir algum conhecimento prévio sobre a natureza da solução desejada (39).

2.5 População Inicial

Tradicionalmente, a população inicial em algoritmos para PG é formada por árvores geradas aleatoriamente a partir dos conjuntos de funções F e de terminais T. Inicialmente, uma função $f \in F$ é escolhida aleatoriamente. Para cada um dos argumentos de f, seleciona-se um elemento de $\{F \cup T\}$. O processo termina quando a árvore estiver totalmente preenchida com terminais. Geralmente, estabelece-se um limite máximo de profundidade da árvore para evitar que estas se tornem excessivamente extensas (23).

Entretanto, a qualidade da população inicial desempenha um papel crucial no sucesso do processo evolutivo, conforme destacado por Daida em 1999. A população inicial precisa representar de maneira significativa o espaço de busca, exibindo uma ampla variedade na composição dos programas. Isso possibilita a convergência para soluções através da recombinação de códigos.

Para aprimorar a qualidade dos programas na população inicial, diversos métodos são empregados, sendo os mais comuns: *ramped-half-and-half* proposto por Koza (39), *random-branch* prosporto por Chellapilla (20), *uniform* proposto por Bohm (18) e *probabilistic tree-creation* proposto por Luke (44).

O método *ramped-half-and-half*, é uma combinação dos métodos *grow* e *full* (40). O método *grow* envolve a criação de árvores com profundidade variável, escolhendo aleatoriamente nós entre funções e terminais, respeitando uma profundidade máxima estabelecida. O método *full* implica na geração de árvores completas, ou seja, todas as árvores geradas terão a mesma profundidade. Esse procedimento é facilmente implementado por meio da escolha de funções para os nós cuja profundidade é inferior à desejada e a seleção de terminais para os nós de profundidade máxima. Essa abordagem visa garantir uma estrutura de árvore regular, onde todos os caminhos da raiz até as folhas possuem o mesmo comprimento, simplificando a manipulação e avaliação dos programas gerados durante o processo evolutivo. A representação esquemática dos métodos *grow* e *full* podem ser observadas no Apêndice A nos Algoritmos 1 e 2 respectivamente.

Integrar os métodos *Full* e *Grow* com o propósito de gerar um número equitativo de árvores para cada profundidade, variando de dois até a profundidade máxima, fundamenta o método *Ramped-Half-and-Half* (40). Por exemplo, considerando uma profundidade máxima de onze, o método visa gerar árvores com profundidades de dois, até onze de maneira proporcional. Isso implica que, respectivamente, 10% das árvores terão profundidade dois, 10% terão profundidade três, e assim por diante. Para cada profundidade, 50% das árvores são geradas pelo método *Full* e 50% pelo método *Grow*, proporcionando uma distribuição

balanceada e abrangente das estruturas de árvores ao longo das diferentes profundidades. Essa abordagem visa otimizar a diversidade estrutural da população inicial, favorecendo a exploração eficiente do espaço de busca durante o processo evolutivo.

2.6 Função de Aptidão

Na natureza, a seleção natural opera identificando os seres vivos mais adaptados ao ambiente. Em Programação Genética, essa dinâmica é traduzida pela função de aptidão ou fitness. Os programas que demonstram maior eficácia na resolução do problema em questão recebem maiores valores de aptidão, proporcionando-lhes uma vantagem na probabilidade de serem selecionados para o processo reprodutivo (39).

A avaliação de aptidão, por sua vez, varia conforme o domínio do problema e pode ser quantificada de maneiras diversas, tanto de forma direta quanto indireta. No contexto deste estudo, prioriza-se a consideração apenas de domínios que possibilitem uma avaliação direta de aptidão.

Em geral, a avaliação de aptidão é conduzida mediante a utilização de um conjunto de casos de treinamento, denominados casos de aptidão, que compreendem pares de valores de entrada e saída a serem aprendidos. Cada programa recebe esses valores de entrada, e a resposta do programa é confrontada com o valor esperado de saída. Quanto mais próxima a resposta do programa estiver do valor de saída desejado, melhor será o desempenho do programa (52).

Dessa forma, a avaliação de aptidão se configura como o mecanismo diferenciador entre programas mais eficazes e menos eficazes, servindo como a força-motriz do processo evolutivo. Ela é a medida, utilizada ao longo da evolução, que indica o quanto o programa aprendeu a prever as saídas a partir das entradas dentro de um domínio de aprendizagem específico (13).

A escolha da função de aptidão, assim como a seleção do método de avaliação adotado por esta função, é intrinsecamente dependente do problema em questão. A tomada de decisões acertadas nesse contexto é crucial para alcançar resultados satisfatórios, visto que a função de aptidão atua como a força-guia que direciona o algoritmo de Programação Genética na busca pela solução (31).

Os métodos convencionalmente empregados para a avaliação de aptidão, conforme proposto por Koza (39), são:

1. **Aptidão Nata (Raw Fitness):** Representa a medida intrínseca ao domínio específico do problema. Reflete a avaliação direta do programa diante dos casos de aptidão. A métrica comumente utilizada na aptidão nata é a avaliação do erro cometido, expresso como a soma de todas as diferenças absolutas entre o resultado obtido pelo programa e seu valor correto.

A avaliação de aptidão nata é calculada considerando a soma das diferenças absolutas entre as respostas geradas pelo programa $\hat{y}_i$ e as saídas corretas desejadas y_i. Para cada programa p_i pertencente à população P, é associado um valor f_p que representa a sua aptidão, obtido por meio da avaliação dos n *fitness cases* fornecidos. O valor de f_p é determinado pela fórmula:

$$f_p = \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{2.2}$$

2. **Aptidão Padronizada (Standardized Fitness):** Dada a dependência da aptidão nata ao domínio do problema, um valor considerado bom pode ser tanto pequeno (ao avaliar o erro) quanto grande (ao avaliar a eficiência). A avaliação da aptidão padronizada é realizada por meio de uma função de adaptação do valor da aptidão nata, de modo que, quanto melhor o programa, menor deve ser a aptidão padronizada. Assim, o melhor programa apresentará o valor zero (0) como aptidão padronizada, independentemente do domínio do problema.

3. **Aptidão Ajustada (Adjusted Fitness):** Derivada da aptidão padronizada, a aptidão ajustada é calculada pela fórmula:

$$a(i,t) = \frac{1}{1 + s(i,t)}. \tag{2.3}$$

onde $s(i,t)$ representa a aptidão padronizada do indivíduo i na geração t. Essa métrica varia entre zero (0) e um (1), sendo que valores mais elevados indicam os melhores indivíduos. A aptidão ajustada possui a vantagem de amplificar a importância de pequenas diferenças nos valores de aptidão padronizada, especialmente quando esta se aproxima de zero.

4. **Aptidão Normalizada (Normalized Fitness):** Se $a(i,t)$ denota a aptidão ajustada do indivíduo i na geração t, então a aptidão normalizada $n(i,t)$ é calculada como:

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{m} a(k,t)} \tag{2.4}$$

Nota-se que a soma de todas as aptidões normalizadas dentro de uma população totaliza um (1), proporcionando uma visão proporcional e equilibrada da aptidão de cada indivíduo em relação aos demais.

2.7 Métodos de Seleção

O método de seleção desempenha um papel crucial em algoritmos genéticos, pois visa identificar quais programas devem ser submetidos aos operadores genéticos, contribuindo para a formação de uma nova geração. A qualidade de um programa é intrinsecamente

relacionada ao seu valor de aptidão, tornando imperativo que a seleção favoreça os programas que apresentem os melhores valores de aptidão.

Atualmente, diversos métodos de seleção são empregados, conforme propostos por Blickle & Thiele em (17). Dentre eles, destacam-se:

1. **Seleção Proporcional (fitness-proportionate selection):** Originalmente introduzida por John Holland para Algoritmos Genéticos, este método foi adotado por John Koza em seu pioneiro livro (39). Ele utiliza a aptidão normalizada, representada em uma ”roleta”, onde cada indivíduo da população ocupa uma ”fatia” proporcional à sua aptidão normalizada (Figura 2.4). Um número aleatório entre zero (0) e um (1) é gerado, indicando a posição da ”agulha” na roleta. Apesar de sua simplicidade e sucesso, este método pode ser sensível à escalabilidade da aptidão normalizada (Blickle 1995).

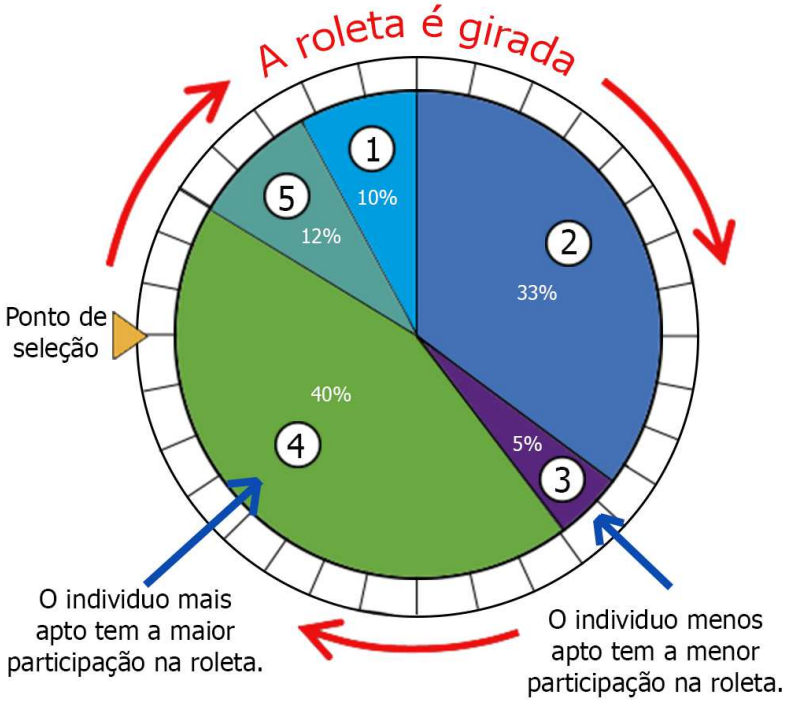

Figura 2.4 – Ilustração da Roleta de Seleção Proporcional
Fonte: Elaborada pelo autor (2024).

2. **Seleção por Torneio (tournament selection):** Desenvolvida por Goldberg & Deb em (25) para Algoritmos Genéticos, a seleção por torneio, como utilizada por John Koza em seu segundo livro (40), ocorre da seguinte maneira: t indivíduos são escolhidos aleatoriamente da população, e o melhor deles é selecionado (Figura 2.5). Esse processo é repetido até que uma nova população seja formada. O valor de t é conhecido como o tamanho do torneio.

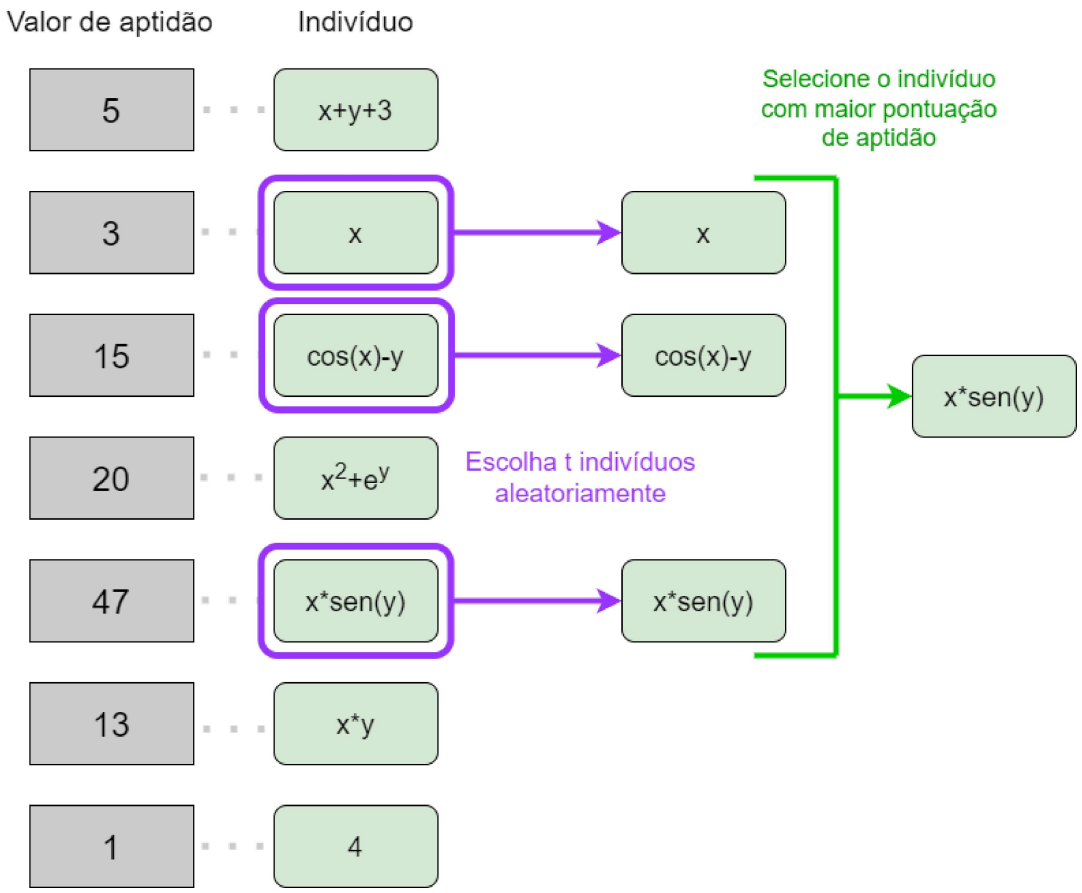

Figura 2.5 – Ilustração da Seleção Torneio
Fonte: Elaborada pelo autor (2024).

3. **Seleção por Truncamento (truncation selection):** Com base em um valor de limiar (threshold) T entre zero (0) e um (1), a seleção é realizada aleatoriamente entre os T melhores indivíduos (49). Por exemplo, se T = 0,4, a seleção ocorre entre os 40% melhores indivíduos, descartando os 60% restantes.

4. **Seleção por Ordenação Linear e Exponencial:** Propostos por Baker (12) para superar as desvantagens da seleção proporcional, esses métodos ordenam os indivíduos conforme os valores de aptidão. Em ambos os métodos, os indivíduos são ordenados com base em seus valores de aptidão, variando de um nível N (melhor indivíduo) a 1 (pior), e cada indivíduo i recebe uma probabilidade ρ_i de ser selecionado.

   Na Seleção por Ordenação Linear (linear ranking selection), cada indivíduo i recebe uma probabilidade ρ_i de ser selecionado, com um valor de $\frac{n^+}{N}$ representando a probabilidade do melhor indivíduo ser escolhido e $\frac{n^-}{N}$, a do pior. Mesmo que dois indivíduos tenham a mesma aptidão, eles apresentam probabilidades diferentes de serem escolhidos, conforme a seguinte equação:

$$\rho_i = \frac{1}{N}\left(n^- + (n^+ - n^-)\frac{i-1}{N-1}\right) \tag{2.5}$$

Onde $i \in \{1, 2, \ldots, N\}$, $n^- \geq 0$ e $n^+ - n^- = 2$.

Na Seleção por Ordenação Exponencial (exponential ranking selection), similar à Ordenação Linear, as probabilidades p_i são exponencialmente ponderadas (12). Um parâmetro c, variando entre zero (0) e um (1), é utilizado como base. Quanto mais próximo de um, menor é a "exponencialidade" da seleção. Os indivíduos são ordenados de acordo com os valores de aptidão, associados a um nível N (melhor indivíduo) a 1 (pior), e cada indivíduo i recebe uma probabilidade ρ_i de ser selecionado.

$$\rho_i = \frac{c-1}{c^{N-1}} c^{N-i}, \text{ onde } i \in \{1, 2 \ldots N\} \tag{2.6}$$

2.8 Operadores Genéticos

Após a seleção dos indivíduos, faz-se imperativo a aplicação de operadores genéticos, elementos fundamentais no processo de programação genética. Os três operadores preeminentes, conforme delineados em (39), são:

1. **Reprodução:** Consiste na seleção de um programa, então replicado integralmente para a próxima geração, mantendo inalterada sua estrutura. Suas vantagens incluem a preservação de soluções promissoras ao longo das gerações e a manutenção de características desejáveis dos indivíduos. Porém, pode levar à estagnação da população, reduzindo a diversidade genética e limitar a introdução de novas soluções.

2. **Cruzamento (Crossover):** Este operador envolve a seleção de dois programas, recombinados para gerar dois novos programas. Um ponto de cruzamento aleatório é escolhido em cada programa-pai e as subárvores abaixo desses pontos são intercambiadas (Apêndice A, Algoritmo 6). A Figura 2.6 ilustra um exemplo de cruzamento, onde programas distintos $tan(X_2) * cos(X_1)$ e $\frac{X_1}{7} + sen(X_2)$ são cruzados, resultando nos novos programas $tan(X_2) * \frac{X_1}{7}$ e $cos(X_1) + sen(X_2)$. Para assegurar a viabilidade do cruzamento, é essencial que o conjunto de funções exiba a propriedade de Fechamento (closure), permitindo que funções suportem qualquer outra função ou terminal como argumento. Caso contrário, critérios restritivos na escolha dos pontos de cruzamento tornam-se imperativos. Suas vantagens incluem a combinação de características promissoras de diferentes soluções e a introdução de diversidade na população, acelerando a busca por soluções ótimas. No entanto, pode gerar soluções inválidas caso as restrições do problema não sejam adequadamente consideradas.

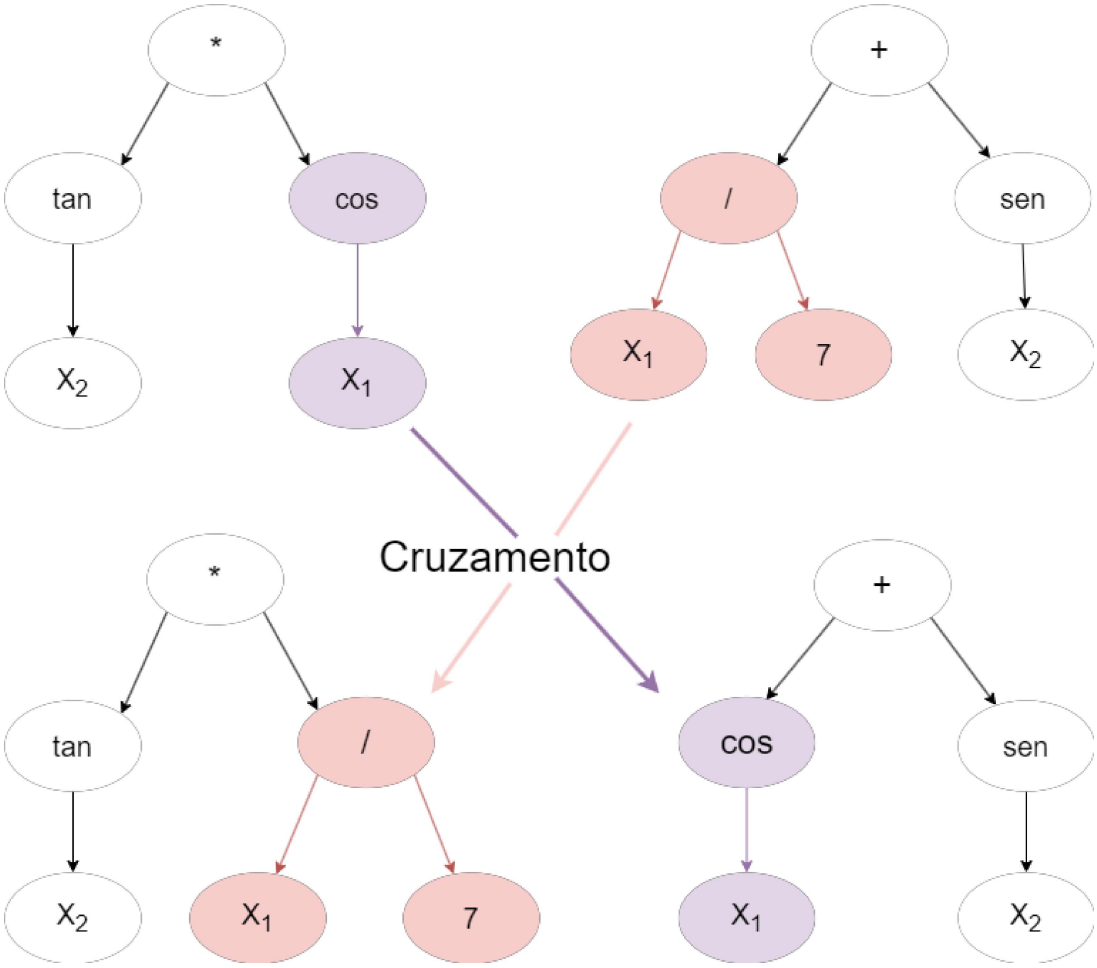

Figura 2.6 – Exemplo de Cruzamento Entre Dois Programas
Fonte: Elaborada pelo autor (2024).

3. **Mutação:** Nesse operador, um programa é selecionado, e aleatoriamente um de seus nós é escolhido. A árvore, cuja raiz é o nó selecionado, é então eliminada e substituída por uma nova árvore gerada aleatoriamente (Apêndice A, Algoritmo 7). A Figura 2.7 ilustra um exemplo de mutação, onde um programa distinto $9 + ln(X_1)$ sofre uma mutação, resultando no novo programa $\frac{X_2}{X_1} + ln(X_1)$. Este processo de mutação introduz diversidade na população, desempenhando um papel crucial na exploração do espaço de busca. Suas vantagens incluem a introdução de variação genética na população, explorando novas regiões do espaço de busca, e a capacidade de escapar de mínimos locais. No entanto, pode introduzir soluções não válidas e a taxa de mutação deve ser ajustada para evitar uma excessiva perturbação na população.

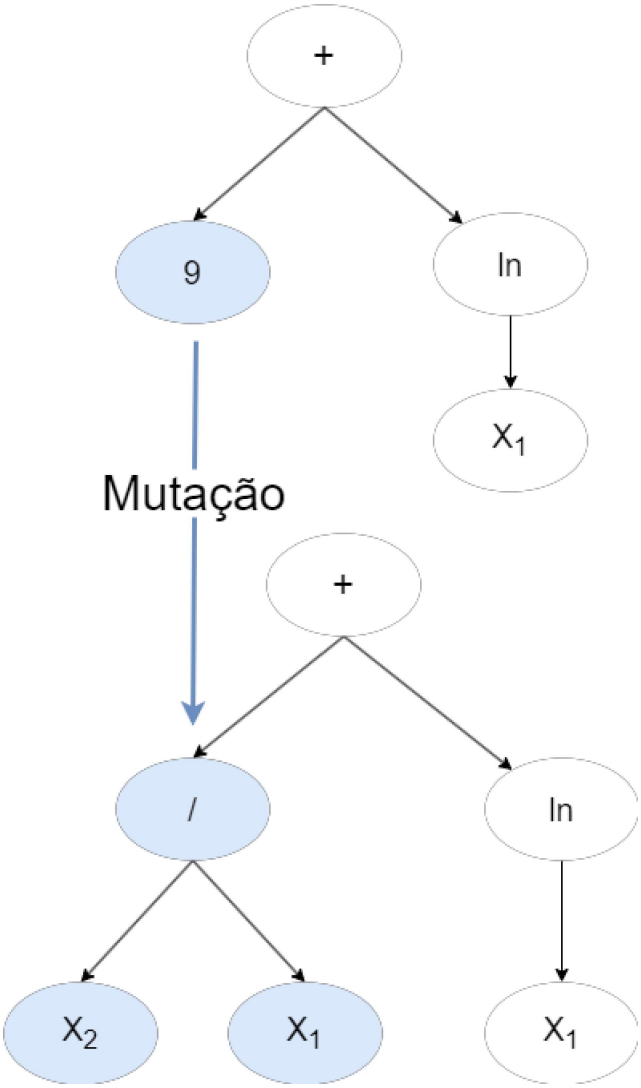

Figura 2.7 – Exemplo de Mutação de um Programa
Fonte: Elaborada pelo autor (2024)

A utilização destes operadores, em conjunto, propicia a evolução e adaptação da população de programas ao longo das gerações, visando a otimização de desempenho em tarefas específicas. Quando comparada com outras abordagens, a combinação desses operadores oferece uma estratégia adaptativa, capaz de explorar um amplo espaço de busca de soluções ótimas. Isso permite que soluções mais diversificadas sejam encontradas pelo algoritmo de programação genética em comparação com abordagens que não fazem uso desses operadores genéticos.(40)

2.9 Critério de Parada

A função de parada desempenha um papel crucial no processo evolutivo, atuando como um mecanismo para interromper o ciclo de repetição que pode perdurar indefinidamente. O critério mais comumente empregado consiste em estabelecer um limite para o número máximo de gerações ou até mesmo até que uma solução satisfatória seja alcançada (39). No entanto, é importante observar que existem critérios mais dinâmicos, fundamentados no acompanhamento contínuo do progresso evolutivo.

Seguindo essa abordagem dinâmica, proposta por Kramer & Zhang em (42), a evolução persiste enquanto houver melhorias na média da população. Este método adota uma perspectiva mais adaptativa, permitindo que o algoritmo evolutivo continue iterando até que não se observem mais ganhos substanciais na qualidade média dos indivíduos da população. Essa abordagem ajustável à evolução intrínseca do problema proporciona uma maior flexibilidade ao algoritmo, adaptando-se à complexidade e dinâmica variáveis dos cenários evolutivos. Dessa forma, a função de parada não está rigidamente atrelada a um número fixo de gerações, mas sim à dinâmica efetiva do processo evolutivo em si, buscando otimizar a eficiência na busca por soluções.

3 METODOLOGIA

3.1 Base de Dados

A fase de preparação de dados representa um componente crucial no desenvolvimento de qualquer técnica relacionada à Inteligência Artificial. O tratamento dos dados, com a exclusão criteriosa de valores discrepantes, é essencial antes da utilização destes na construção de modelos de IA. No escopo do presente estudo, foram utilizados 795 amostras referentes a testes de produtividade de poços, os quais foram coletados em diversos campos no Oriente Médio, conforme detalhado no trabalho realizado por Al-Shammari (4) e todos os dados foram devidamente submetidos a procedimentos de tratamento.

O conjunto de dados em questão engloba diversos parâmetros de entrada, dentre os quais se destacam: Pressão na cabeça do poço (*Wellhead Pressure* - WHP), Taxa de fluxo de água (*Water Flow Rate* - WFR), Taxa de fluxo de óleo (*Oil Flow Rate* - OFR), Taxa de fluxo de gás (*Gas Flow Rate* - GFR), Produção diária de água (*Water Production Rate* - WPD), Índice de gravidade específica do petróleo (*American Petroleum Institute Gravity* - API), Diâmetro interno do tubo (*Internal Diameter of Pipe* - ID), e Temperatura na cabeça do poço (*Wellbore Head Temperature* - WBHT). Como variável de saída, a FBHP. A Tabela 3.1, apresentada a seguir, exibe a faixa de dados associada aos parâmetros de entrada e saída, destacando a amplitude que esses valores podem assumir.

Tabela 3.1 – Faixas de dados coletados de parâmetros de entrada e saída (4)

			Mínimo	**Máximo**	**Média**	**Desvio Padrão**
Entrada	WHP (psi)	X_0	92	1550	423,82	253,74
	WFR (bpd)	X_1	0	11395	2215,13	2294,8
	OFR (bdp)	X_2	176	17663	2215,13	3722,13
	GFR (scf/stb)	X_3	9	17859	2699,15	2370,41
	WPD (L/dia)	X_4	4243	8620	6326,89	511,37
	API	X_5	25,4	47,5	33,86	3,11
	ID (in)	X_6	1,995	6,276	3,95	0,57
	WBHT (^{o}F)	X_7	160	233	2010,25	18,26
Saída	FBHP (psi)	y	1198	3698	2469,73	387,23

Com o intuito de mitigar possíveis desafios relacionados ao ajuste excessivo do modelo, optou-se por empregar a estratégia de normalização dos dados, evitando que variáveis com magnitudes diferentes influenciem desproporcionalmente o modelo. Adicionalmente, procedeu-se à renomeação das variáveis, designando-as de X_0 até X_7 para os parâmetros de entrada, conforme a sequência apresentada na Tabela 3.1. O parâmetro de saída foi denotado por y. Essa sistemática de nomenclatura foi adotada visando facilitar a interpretação das equações geradas pelo modelo, contribuindo para uma compreensão mais clara e intuitiva do processo analítico.

A Figura 3.1 mostra os coeficientes de correlação de Pearson entre as variáveis de entrada e a pressão de fundo de poço (FBHP). Os coeficientes variam entre +1 e -1, onde +1 representa uma correlação direta entre as variáveis e -1 uma relação indireta entre as variáveis. A figura 3.2 complementa a análise com oito histogramas que ilustram a distribuição das variáveis independentes e sua relação com a FBHP.

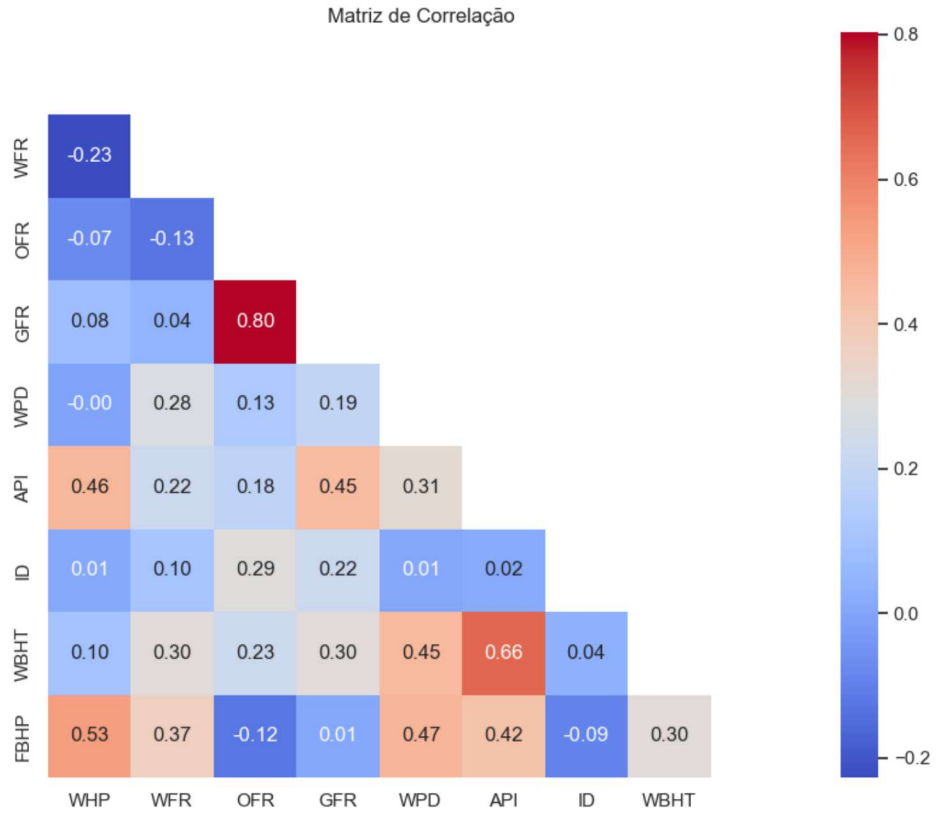

Figura 3.1 – Coeficiente de correlação entre variáveis de entrada e FBHP
Fonte: Elaborada pelo autor (2024).

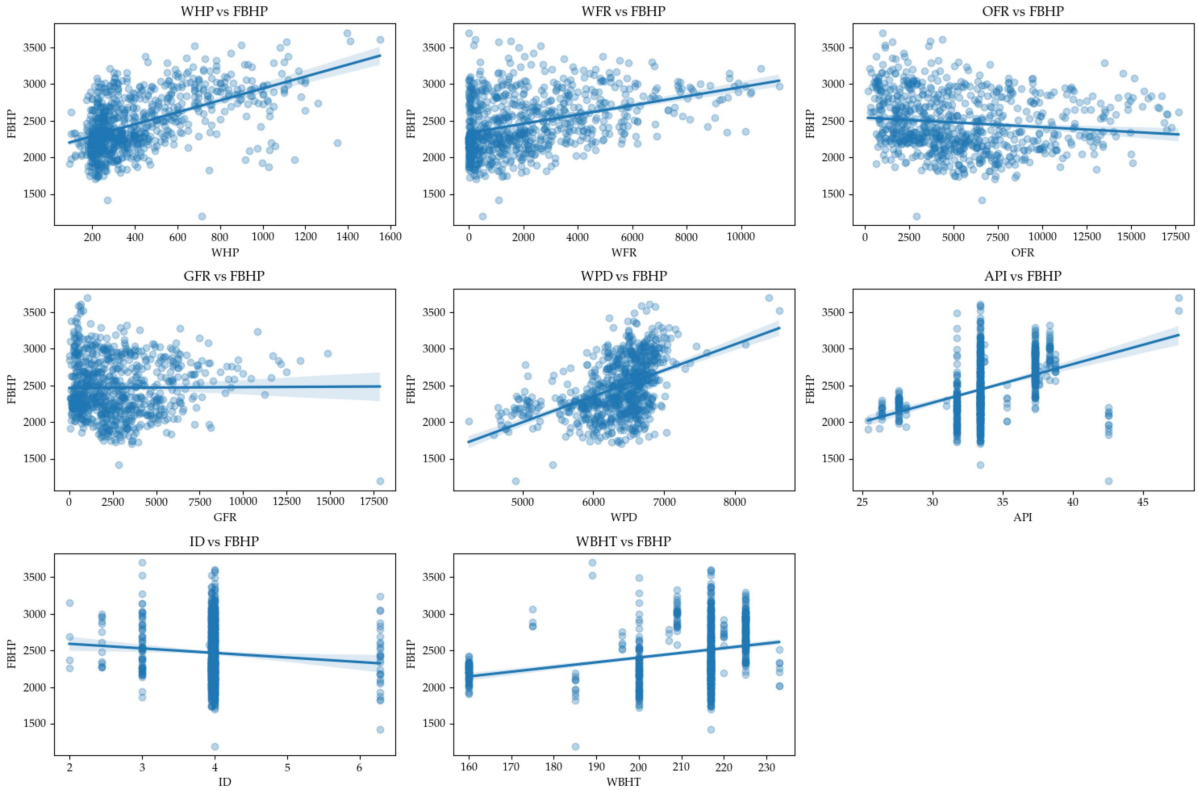

Figura 3.2 – Distribuição da pressão de fundo de poço em relação às Variáveis operacionais.
Fonte: Elaborada pelo autor (2024).

A correlação entre a pressão na cabeça do poço (WHP) e a pressão no fundo do poço (FBHP) apresentada na figura 3.1 é moderada e positiva, com um valor de 0,53. A distribuição dos pontos nos gráficos WHP vs FBHP na figura 3.2 indica que existe uma relação proporcional direta entre WHP e FBHP. No entanto, a dispersão dos pontos sugere que a correlação não é perfeita, ou seja, para um mesmo valor de FBHP, pode haver uma variação significativa na WHP, isto ocorre devido às perdas de pressão na tubulação devido ao atrito e outros fatores como a natureza do escoamento do fluido. O óleo em movimento gera atrito contra as paredes da tubulação, o que causa perda de pressão. No escoamento multifásico bolhas se formam na tubulação reduzindo a densidade do fluido, diminuindo a pressão na tubulação.

Em relação à taxa de fluxo de água (WFR) e à produção diária de água (WPD), ambas assumem uma correlação com a FBHP moderada e positiva, com um valor de 0,37 e 0,47 respectivamente. Isso indica que aumentos na WFR e na WPD tendem a elevar a FBHP. Essa relação é compreensível em poços com elevação artificial, onde a injeção de água pressurizada no poço eleva a FBHP e facilita a subida do óleo à superfície. Contudo, os gráficos de dispersão entre WFR e FBHP e entre WPD e FBHP (figura 3.2) apresentam uma dispersão moderada dos pontos. O maior agrupamento de pontos no gráfico WPD vs FBHP em comparação com o gráfico WFR vs FBHP pode ser explicada

por diversos fatores. Os processos de elevação artificial, como bombas de cavitação ou centrífugas, operam em faixas de eficiência específicas, influenciando mais diretamente a WPD e levando a um maior agrupamento de pontos no gráfico. A WFR, por sua vez, é mais sensível a flutuações momentâneas em fatores como a permeabilidade da formação geológica, a presença de gás ou outros fluidos, e as condições da tubulação e bombas, resultando em uma dispersão mais ampla dos pontos no gráfico WFR vs FBHP.

Por outro lado, a correlação entre a taxa de fluxo de óleo (OFR) e a FBHP é fraca e negativa, com um valor de -0,12. O gráfico de dispersão OFR vs FBHP da figura 3.2 apresenta uma dispersão elevada dos pontos, diferentes valores de OFR assumem os mesmo valores de FBHP explicando a correlação fraca. A relação inversa entre essas duas variáveis é esperada, pois o fluxo de óleo é uma variável de saída representando a quantidade de óleo produzida pelo poço. Assim, um aumento na taxa de fluxo de óleo está geralmente associado a uma diminuição na pressão no fundo do poço, devido ao escoamento do óleo.

No que diz respeito à taxa de fluxo de gás (GFR), a correlação com a FBHP é muito fraca e positiva, com um valor de 0,01, indicando que as variações na taxa de fluxo de gás não têm um impacto discernível na pressão no fundo do poço. A dispersão significativa dos pontos (figure 3.2) e a fraca correlação linear indicam que a relação entre as variáveis não é muito clara. O índice de gravidade específica do petróleo (API) também apresenta uma correlação moderada e positiva com a FBHP, com um valor de 0,42. Isso sugere que o API influencia a pressão no fundo do poço, indicando que a composição do petróleo extraído pode afetar a pressão no fundo do poço.

Quanto ao diâmetro interno do tubo (ID), a correlação é fraca e negativa, com um valor de -0,09. Isso sugere uma relação inversa entre o diâmetro interno do tubo e a FBHP, essa relação muito fraca, indica que o diâmetro interno do tubo tem uma influência limitada na pressão no fundo do poço. Em relação à temperatura na cabeça do poço (WBHT), a correlação é moderada e positiva, com um valor de 0,30. Isso sugere que a temperatura na cabeça do poço influencia a pressão no fundo do poço, indicando uma relação entre essas variáveis.

3.2 Avaliação do Modelo

O desempenho do modelo foi avaliado usando as seguintes métricas: c Na Tabela 3.2, é possível encontrar descrições detalhadas dessas métricas. A escolha dessas métricas baseou-se em sua capacidade de abranger os diversos comportamentos apresentados pelos modelos analisados, conforme destacado por Trujillo *et al.* em (66).

O Coeficiente de Determinação (R^2) avalia a variabilidade capturada pelo modelo em relação à variabilidade total dos dados. Um valor mais alto de R^2 indica que uma maior proporção da variabilidade na variável dependente é capturada pelo modelo. Ele é calculado como a soma dos quadrados das diferenças entre os valores observados e os

valores previstos, normalizada pela soma dos quadrados das diferenças entre os valores observados e a média dos valores observados.

O Coeficiente de Pearson (R) mede a força e a direção da relação linear entre as variáveis. Em modelos de regressão, a presença de uma correlação forte e positiva entre as variáveis de resposta e predição sugere que o modelo está capturando efetivamente as tendências lineares nos dados. Ele é calculado como a covariância normalizada pelo produto dos desvios padrão das variáveis.

O Erro Quadrático Médio (MSE) é uma métrica que calcula a média dos quadrados das diferenças entre os valores observados e os valores previstos. Essa métrica penaliza de maneira mais intensa os erros maiores, proporcionando uma avaliação mais sensível às discrepâncias significativas. O MSE é particularmente valioso ao avaliar a qualidade das previsões numéricas do modelo. Uma minimização eficaz do MSE durante o treinamento do modelo indica que a expressão simbólica gerada está otimizada para se ajustar aos dados observados. Isso é vital para garantir que o modelo gerado pela programação genética seja capaz de fornecer previsões precisas e bem ajustadas.

O Erro Quadrático Médio Relativo (RMSE) é uma métrica que representa a raiz quadrada da média dos quadrados das diferenças entre os valores observados e os valores previstos, normalizada pela média dos valores observados, avaliando a precisão das previsões em termos absolutos. Um RMSE menor indica que as previsões do modelo estão mais próximas dos valores reais.

O Erro Percentual Médio Absoluto (MAPE) fornece uma medida de precisão percentual, sendo particularmente útil para entender o desempenho do modelo em termos de erros relativos. Uma baixa porcentagem de erro absoluto médio indica que o modelo está gerando previsões precisas em relação aos valores reais.

Tabela 3.2 – Métricas de desempenho e sua expressão matemática (66)

Nome	Expressão
R	$\frac{\sum_{i=1}^{N}(y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i-\bar{\hat{y}})^2}}$
R^2	$1-\frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y})}$
MSE	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2$
RMSE	$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y})^2}}{\bar{y}}$
MAPE	$\frac{100}{N}\sum_{i=1}^{N}\left\lvert\frac{y_i-\hat{y}}{y_i}\right\rvert$

Onde N é o número total de elementos testados, y são os valores reais e $\hat{y}$ são os valores previstos.

3.3 Recursos Computacionais e Ferramentas Utilizadas

Todas as investigações realizadas neste trabalho foram feitas usando um laptop com um processador Intel Core I7-7700K, 4.2GHz com 16 GB de memória DDR3-2400 MHz. Os códigos que foram executados nesta configuração de hardware foram desenvolvidos em linguagem de programação Python (Python versão 3.11). Para a obtenção de expressões simbólicas foi utilizado um modelo de Programação Genética (biblioteca PySR (30), versão 0.16.9). Para as métricas de avaliação foi utilizada a biblioteca do scikit-learn (53) (versão 1.3).

3.4 Experimento Computacional

O PG, discutido detalhadamente no Capítulo 2, é um método evolutivo que inicia sua execução construindo uma população inicial inadequada para uma tarefa específica. No entanto, ao aplicar operadores genéticos ao longo de um número pré-definido de gerações, o algoritmo converge para a melhor solução da população, adaptando-se da melhor forma possível à tarefa designada.

A criação da população inicial envolve a definição de diversos hiperparâmetros, como o tamanho da população (*population_size*) e o conjunto de funções matemáticas (*binary_operators* e *unary_operators*). Cada membro da população é desenvolvido a partir da escolha aleatória de variáveis de entrada, constantes e funções matemáticas. Na PG, os membros da população são representados como estruturas em árvore, onde o

tamanho é determinado pela profundidade em relação ao nó raiz. O hiperparâmetro de profundidade (*maxdepth*) estabelece a profundidade máxima que uma população pode assumir durante todas as gerações, enquanto o tamanho máximo da equação, isto é, o número total de elementos, incluindo operadores, variáveis e constantes, é definido pelo hiperparâmetro (*maxsize*). Os indivíduos da população são gerados através do método *Ramped-Half-and-Half*.

Após a criação da população inicial, é necessário avaliar cada membro para calcular o valor da função de aptidão, que representa a qualidade do indivíduo. Isso envolve fornecer os valores das variáveis de entrada a cada indivíduo, calcular a saída e, em seguida, comparar essa saída com a saída real para determinar o valor da função de aptidão. A *Aptidão Normalizada* (Equação 2.4) foi utilizada em todas as avaliações.

Após a avaliação, alguns membros da população são selecionados como pais para a próxima geração, sujeitos a operações genéticas. A seleção de pais é realizada por meio do método de seleção por torneio. Quatro operações genéticas distintas são empregadas: cruzamento, mutação pontual, mutação de elevação e mutação de subárvore. A soma das probabilidades de todas as operações genéticas é mantida em 1 para controlar a reprodução dos membros da população na próxima geração. Em mutações pontuais, nós aleatórios do vencedor do torneio são alterados, substituindo constantes por valores aleatórios, variáveis por outras variáveis e funções por funções equivalentes em aridade. Na mutação de elevação, uma subárvore é escolhida aleatoriamente, e um nó aleatório nessa subárvore substitui toda a subárvore. Já na mutação de subárvore, uma subárvore aleatória é substituída por outra subárvore gerada aleatoriamente. A operação de cruzamento requer a seleção de dois vencedores do torneio, sendo uma subárvore aleatória do primeiro vencedor substituída por uma subárvore aleatória do segundo vencedor.

No modelo empregado, dois hiperparâmetros desempenham o papel crucial de determinar o encerramento do algoritmo: A função de perda (*loss*) e o número máximo de gerações (*niterations*). A função de perda (*loss*) é utilizada para avaliar o quão bem o modelo está performando em cada iteração do treinamento. O modelo busca minimizar essa função ajustando seus parâmetros. Neste trabalho, a função de perda foi definida como o MSE. Durante o treinamento, o modelo ajusta seus parâmetros de forma a minimizar o MSE entre as previsões e os valores reais do conjunto de treinamento. O modelo encerra sua execução quando não há mais melhoria significativa na função de perda ao longo de várias gerações, ou ao atingir o valor de $loss < 10^{-6}$, ou ao atingir o número máximo de gerações. Em todas as análises realizadas, a conclusão do algoritmo PG ocorreu após o alcance do número máximo de gerações predefinido.

Os hiperparâmetros empregados nesta pesquisa encontram-se detalhadamente descritos na Tabela 3.3. A busca pela formulação de uma equação que descreva a Pressão no Fundo fo Poço (FBHP), a partir de dados de poço, foi conduzida mediante a aplicação

da estratégia de complexidade das equações. Essa abordagem visa encontrar soluções interpretáveis, evitando a obtenção de equações de difícil compreensão para seres humanos sem o auxílio de máquinas.

Para alcançar esse objetivo, o modelo foi restrito a dois conjuntos de possíveis soluções: polinomiais e sem restrições a sua forma (livres). Esses conjuntos foram subdivididos em três tipos, cada um representando uma complexidade máxima que as equações poderiam atingir, sendo estas 20, 30 e 40. Assim, foram gerados seis modelos distintos, os quais serão submetidos a processos de treinamento e teste.

Com o propósito de avaliar a sensibilidade dos parâmetros nos modelos finais obtidos, cada modelo foi submetido a 30 execuções. Cada execução compreendeu a geração de um novo conjunto de dados, dividido em conjuntos de treinamento e teste, representando, respectivamente, 70% e 30% do total. Esse processo abrangeu análises de variações por meio da aplicação de diferentes sementes aleatórias. Essa abordagem sistemática foi executada para avaliar a capacidade do modelo em adaptar-se de maneira eficaz ao conjunto de dados em análise.

Tabela 3.3 – Hiperparâmetros utilizados em cada execução
Fonte: Elaborada pelo autor (2024)

	Polinomial			Livre		
Complexidade máxima	20	30	40	20	30	40
niterations	1000	1000	1000	1000	1000	1000
population_size	1000	1000	1000	1000	1000	1000
binary_operators	[’*’, ’+’, ’-’, ’/’]	[’*’, ’+’, ’-’, ’/’]	[’*’, ’+’, ’-’, ’/’]	[’*’, ’+’, ’-’, ’/’]	[’*’, ’+’, ’-’, ’/’]	[’*’, ’+’, ’-’, ’/’]
unary_operators	none	none	none	[’neg’, ’square’, ’cube’, ’exp’, ’abs’, ’sqrt’, ’sin’, ’cos’, ’tan’, ’sinh’, ’cosh’, ’tanh’, ’atan’, ’asinh’, ’acosh’]	[’neg’, ’square’, ’cube’, ’exp’, ’abs’,’sqrt’, ’sin’, ’cos’, ’tan’, ’sinh’, ’cosh’, ’tanh’, ’atan’, ’asinh’, ’acosh’]	[’neg’, ’square’, ’cube’, ’exp’, ’abs’, ’sqrt’, ’sin’, ’cos’, ’tan’, ’sinh’, ’cosh’, ’tanh’, ’atan’, ’asinh’, ’acosh’]
model_selection	”best”	”best”	”best”	”best”	”best”	”best”
maxdepth	10	10	10	10	10	10

4 RESULTADOS E DISCUSSÃO

A Tabela 4.1 apresenta resultados médios obtidos por modelos de Programação Genética (PG) utilizados para prever os valores de FBHP em um conjunto de teste. O desempenho dos modelos é avaliado através das métricas apresentadas na Seção 3.2.

Tabela 4.1 – Resultados médios para os modelos de PGRS usados para prever os valores de FBHP no conjunto de teste. Valores entre parênteses indicam o desvio padrão em 30 execuções independentes. Valores destacados em negrito indicam os melhores valores médios.
Fonte: Elaborada pelo autor (2024).

Solução	Comp.	R^2	R	MSE	RMSE	MAPE
Polinomial	20	0,651 (0,028)	0,812 (0,017)	0,047 (0,005)	0,228 (0,010)	7,685 (0,378)
	30	0,689 (0,228)	0,852 (0,061)	0,039 (0,032)	0,205 (0,049)	6,634 (0,588)
	40	0,744 (0,036)	0,866 (0,019)	0,040 (0,006)	0,196 (0,016)	6,423 (0,567)
Livre	20	0,635 (0,033)	0,802 (0,020)	0,054 (0,006)	0,233 (0,013)	7,931 (0,507)
	30	0,729 (0,042)	0,856 (0,024)	0,042 (0,006)	0,200 (0,016)	6,568 (0,547)
	40	**0,756 (0,038)**	**0,872 (0,021)**	**0,040 (0,006)**	**0,193 (0,015)**	**6,334 (0,460)**

Tabela 4.2 – Tempo médio de processamento em segundos com desvios padrão (calculado em 30 execuções independentes).
Fonte: Elaborada pelo autor (2024).

Solução	Comp.	Tempo [s]
Polinomial	20	$2908,19 \pm 108,24$
	30	$4878,68 \pm 124,32$
	40	$6144,66 \pm 183,44$
Livre	20	$5369,39 \pm 116,58$
	30	$7566,25 \pm 144,13$
	40	$10482,74 \pm 201,09$

Na análise da seção dedicada ao modelo polinomial na Tabela 4.1, observa-se uma nítida tendência de aprimoramento nas métricas à medida que a complexidade (Comp.) do modelo é incrementada. Destaca-se, de maneira significativa, o modelo cuja complexidade atinge o valor máximo de 40, evidenciando os maiores valores médios para R^2 e R e os menores valores médios para MSE, RMSE e MAPE. Tais resultados apontam para a conclusão de que o aumento da complexidade desempenha um papel crucial no aprimoramento da capacidade explicativa e preditiva do modelo.

De forma análoga, a avaliação do modelo livre reforça essa mesma tendência. Mais uma vez, o modelo composto por 40 termos na equação destaca-se como o mais eficiente, apresentando os melhores valores médios para as métricas previamente mencionadas. A consistência nos resultados entre as abordagens polinomial e livre sugere que modelos mais complexos são, de fato, mais eficazes na tarefa de prever os valores de FBHP.

Entretanto, pela Tabela 4.2, observa-se que o aumento da complexidade está associado a um significativo acréscimo no tempo de processamento. Adicionalmente, os ganhos em precisão, resultantes do aumento da complexidade, são estatisticamente pequenos quando comparados ao custo computacional agregado. A precisão dos modelos e o tempo de execução devem ser ponderados, visando a obtenção de um equilíbrio ideal entre ambos os fatores.

Modelos com maior complexidade podem comprometer a interpretabilidade das equações obtidas, dificultando a obtenção de informações plausíveis sobre os processos subjacentes. Por outro lado, modelos mais simples oferecem a vantagem de menor tempo de processamento e maior interpretabilidade das equações, facilitando a obtenção de compreensões sobre os processos subjacentes. No entanto, as equações obtidas podem não capturar totalmente a complexidade dos dados, resultando em uma menor precisão preditiva. Para uma compreensão mais aprofundada dos resultados, outras análises são necessárias.

As figuras subsequentes apresentam gráficos de dispersão dos valores previstos em relação aos valores reais. Os valores verdadeiros são representados pelos pontos no eixo x, enquanto as previsões são representadas pelos pontos no eixo y. Essa abordagem visual permite uma análise detalhada da concordância entre as previsões dos modelos e os dados reais.

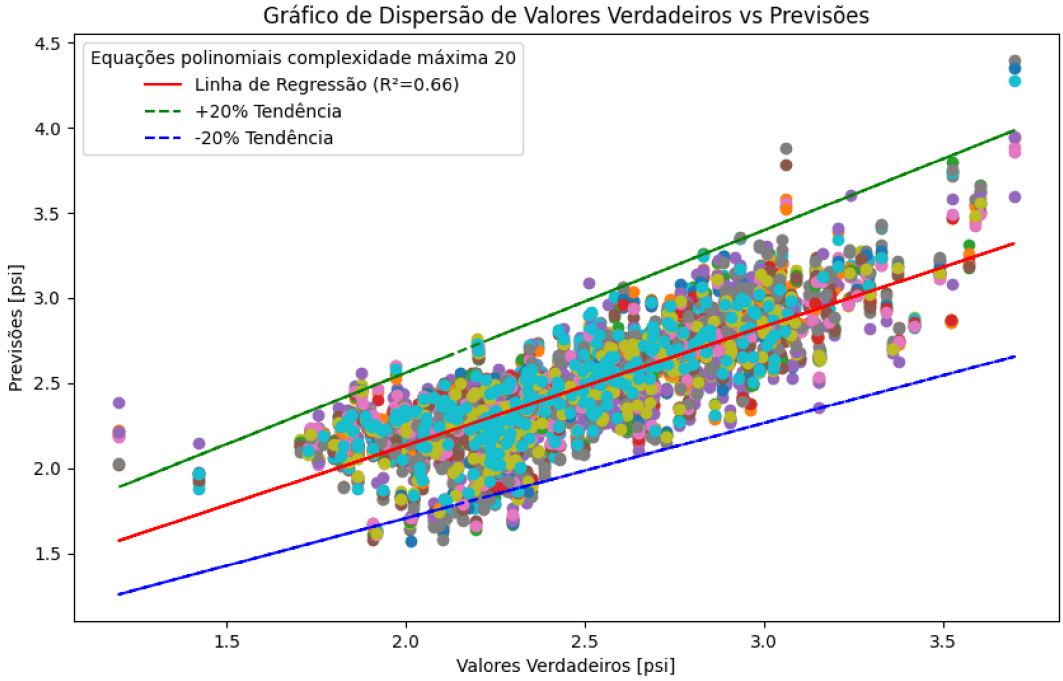

Figura 4.1 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo polinomial com complexidade máxima 20
Fonte: Elaborada pelo autor (2024).

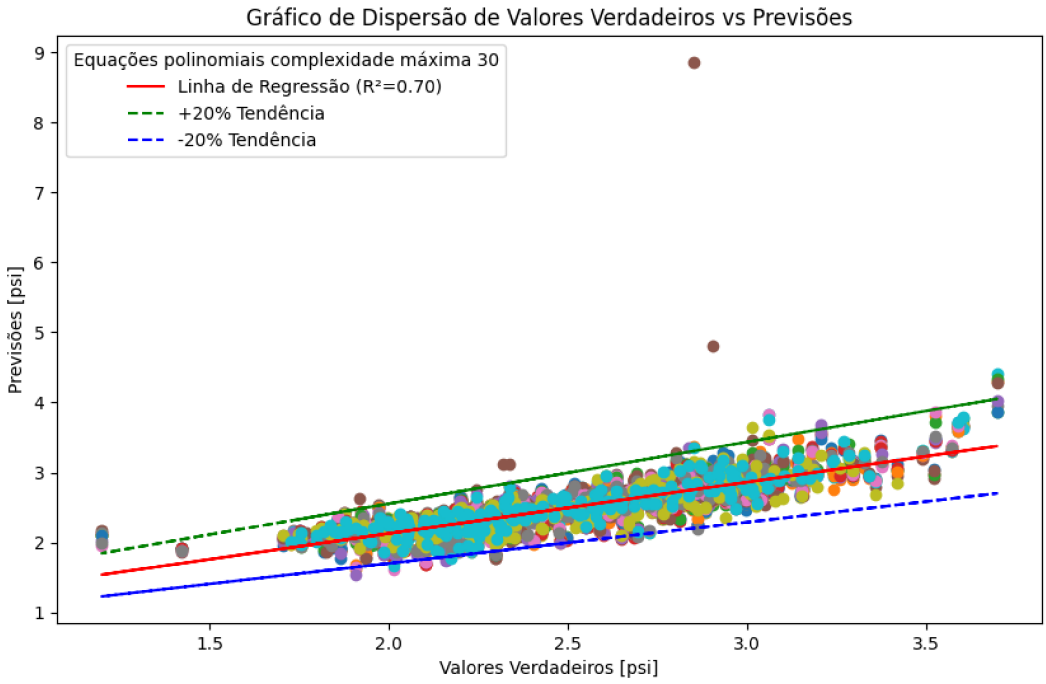

Figura 4.2 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo polinomial com complexidade máxima 30
Fonte: Elaborada pelo autor (2024).

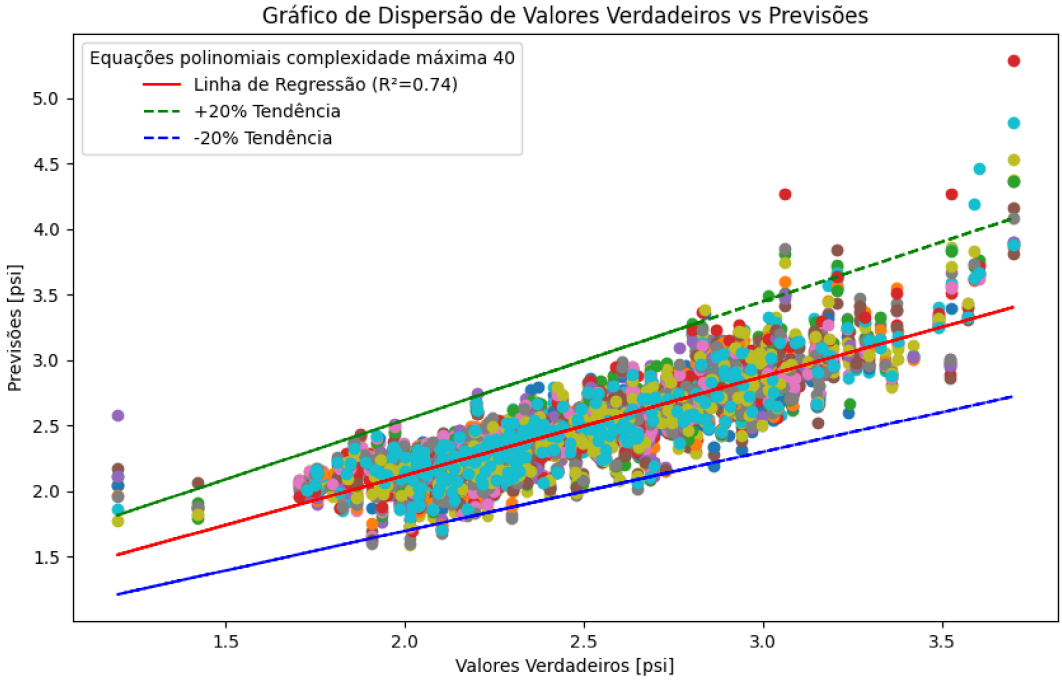

Figura 4.3 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo polinomial com complexidade máxima 40
Fonte: Elaborada pelo autor (2024).

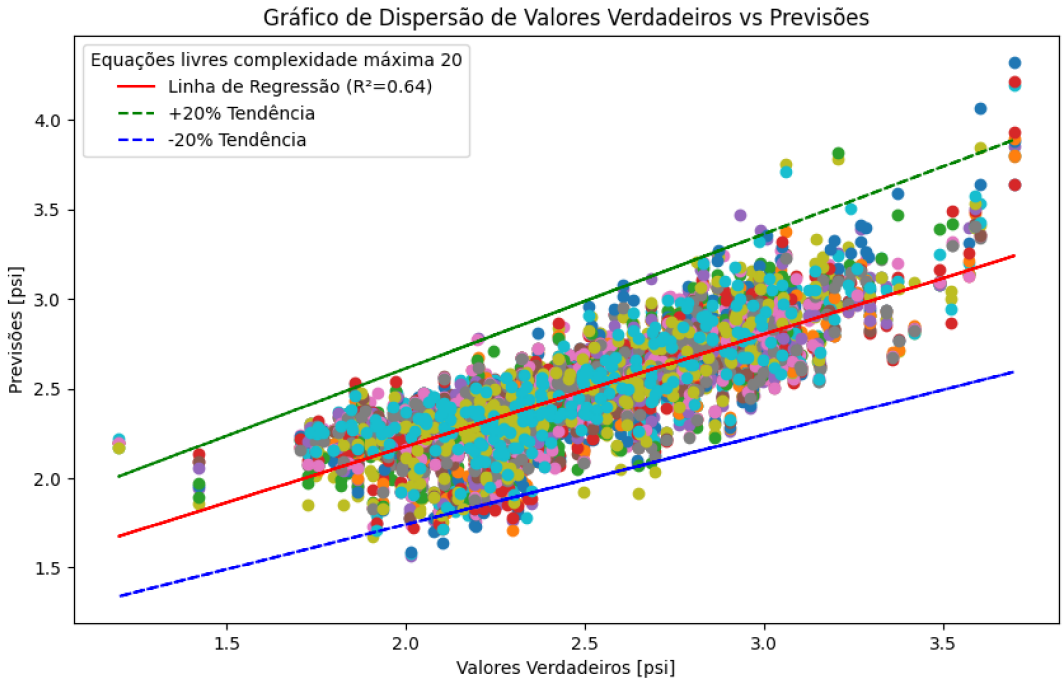

Figura 4.4 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo sem restrições à forma com complexidade máxima 20
Fonte: Elaborada pelo autor (2024).

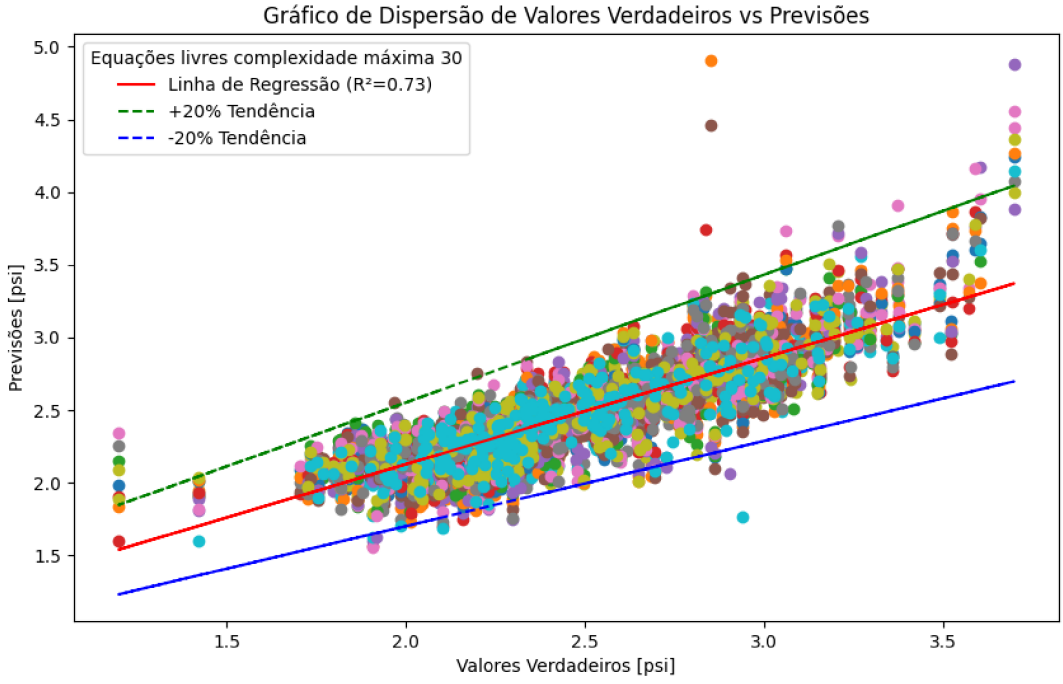

Figura 4.5 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo sem restrições à forma com máxima 30
Fonte: Elaborada pelo autor (2024).

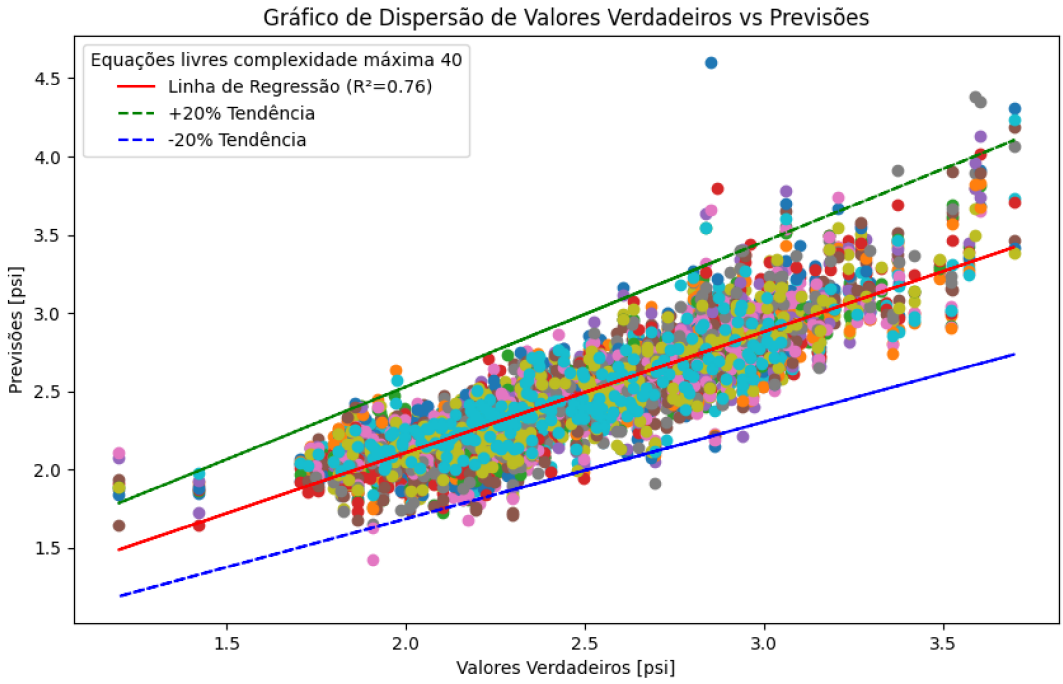

Figura 4.6 – Gráfico de dispersão de valores verdadeiros versus previsões para o modelo sem restrições à forma com complexidade máxima 40
Fonte: Elaborada pelo autor (2024).

Os gráficos das Figuras 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6 revelam uma distribuição concentrada ao longo da linha de regressão para todos os modelos. A dispersão dos resíduos se mantém constante em todo o intervalo de valores previstos, indicando a ausência de heterocedasticidade. Adicionalmente, não se observam tendências nos gráficos residuais, sugerindo a linearidade nos modelos. Em relação a pontos atípicos e influentes, todos os gráficos residuais demonstram que a maioria dos pontos estão contidos dentro da faixa de confiança de 80%.

Entretanto, ao observar a distribuição dos pontos de forma relativamente linear com uma tendência geral de aumento, é possível notar que alguns pontos estão mais distantes da linha de regressão. A dispersão dos pontos sugere que o modelo não é capaz de prever com precisão todos os valores reais de FBHP, com uma variação máxima de 20%. Essas discrepâncias indicam que o modelo de previsão pode não ser tão preciso para valores extremos, possivelmente devido ao treinamento em um conjunto de dados que não inclui valores extremos suficientes. Além disso, o R^2 variando de 0,66 a 0,76 sugere que a melhoria do valor de R^2 ocorre à medida que a complexidade dos modelos aumenta.

Porém, é importante ressaltar que o R^2 não determina se as estimativas e previsões dos coeficientes são tendenciosas. Quando um modelo possui muitos preditores e inclui polinômios de ordem superior, há o risco de que ocorra o fenômeno conhecido como overfitting do modelo. Isso pode resultar em valores de R^2 enganosamente altos e na

redução da capacidade do modelo de fazer previsões precisas. Nesse contexto, a análise gráfica de outras métricas, como o MAPE, se torna essencial.

O MAPE, por exemplo, proporciona uma visão mais direta da precisão das previsões, considerando as discrepâncias percentuais entre as previsões e os valores reais. Ao examinar graficamente o MAPE em conjunto com os gráficos residuais, podemos identificar padrões específicos, avaliar a consistência das previsões em diferentes intervalos e compreender melhor como o modelo se comporta em situações específicas.

A Figura 4.7 representa um gráfico de dispersão do erro (MAPE) versus a complexidade de cada equação obtida pelo modelo de PG limitada a somente equações polinomiais. Já a Figura 4.8 representa um gráfico de dispersão do erro versus a complexidade de cada equação obtida pelo modelo de PG livre para obter qualquer tipo de equação utilizando os operados descritos na Tabela 3.3. O erro é representado pelo eixo y, enquanto a complexidade é representada pelo eixo x. A avaliação da complexidade dos modelos de PG é realizada mediante a contabilização de todos os termos presentes na equação, ou seja, contando o total de variáveis, operadores e constantes envolvidos. A obtenção do erro é conduzida através da métrica MAPE, conforme descrita detalhadamente na Seção 3.2. As soluções apresentadas nos gráficos são as melhores soluções obtidas nas várias execuções de cada modelo.

A curva de não dominância representa a relação entre o erro e a complexidade das equações. Uma equação é considerada não dominada se não houver outra equação no gráfico que tenha um erro menor e uma complexidade menor ou igual. A análise das Figuras 4.7 e 4.8 mostra que, em geral, quanto maior a complexidade da equação, menor o erro. No entanto, existem algumas equações com complexidade menor que apresentam erro menor que outras equações com complexidade maior.

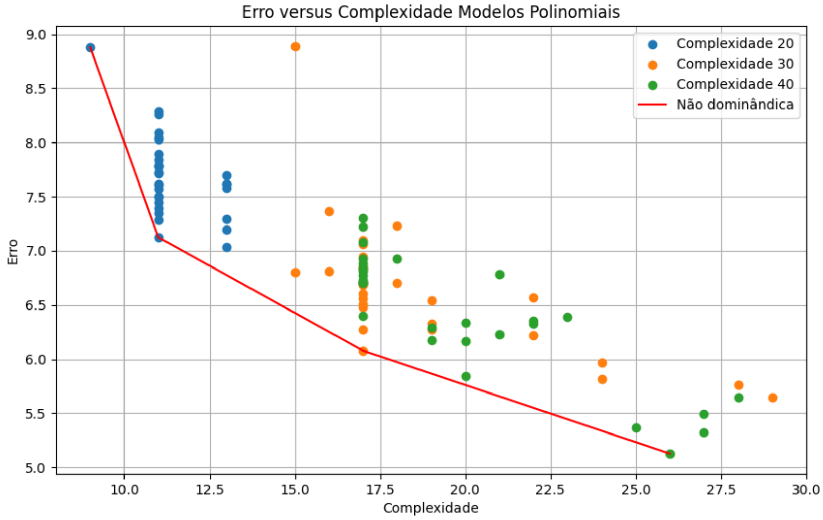

Figura 4.7 – Erro versus a complexidade para os modelos que geram equações polinomiais.
Fonte: Elaborada pelo autor (2024).

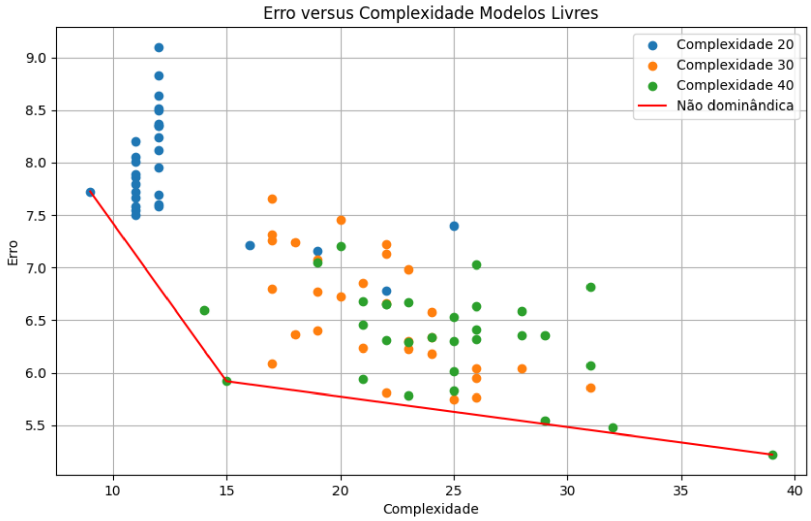

Figura 4.8 – Erro versus a complexidade para os modelos que geram equações sem restrição à forma das equações.
Fonte: Elaborada pelo autor (2024).

Uma razão é que a complexidade de uma equação não é necessariamente um indicador de sua precisão. Existem equações simples que podem capturar relações complexas entre os dados, e existem equações complexas que podem capturar relações simples. Outra razão é que a complexidade de uma equação pode ser afetada por fatores que não estão relacionados à precisão. Portanto, é possível que uma equação com complexidade menor seja mais precisa do que uma equação com complexidade maior se a equação com complexidade menor capturar as relações mais importantes entre os dados. Um exemplo disso são as Equações (4.1) e (4.2) que apresentaram os menores erros dentre todos os modelos executados. A equação (4.1) apresentou um erro 5,22069% enquanto a equação (4.2) apresentou um erro de 5,12705%. Uma análise detalhada destas equações revela características distintas fundamentais para compreender sua complexidade.

A Equação (4.1) é um exemplo de uma expressão polinomial, caracterizada pela presença de termos que envolvem potências inteiras das variáveis, incluindo um termo quadrático (X_2^2) e uma fração ($1/(X_3 + 0,0012534792)$). Essa estrutura polinomial confere à equação uma certa simplicidade em termos de avaliação numérica, facilitando a interpretação dos termos e a implementação prática. No entanto, a inclusão de uma fração adiciona uma camada de complexidade, tornando a equação uma combinação de elementos polinomiais e racionais.

Por outro lado, a Equação (4.2) apresenta uma abordagem diferente. Trata-se de uma expressão não polinomial que incorpora funções mais complexas, como a raiz quarta ($X_3^{1/4}$), a função arco seno hiperbólico ($asinh$), exponencial (e_i^X), e cosseno (cos). Essas funções adicionam uma camada de sofisticação ao modelo, permitindo uma representação mais precisa de fenômenos que podem não ser adequadamente descritos

por polinômios simples. No entanto, a complexidade introduzida por essas funções pode tornar a avaliação numérica mais desafiadora e exigir métodos específicos. Apesar das diferenças fundamentais, ambas as equações compartilham variáveis comuns, como $X_0, X_1, X_2, X_3, X_4, X_6$, e representam relações entre essas variáveis e a variável dependente. Essa semelhança fornece uma base para comparação e avaliação dos modelos em contextos específicos.

$$\begin{aligned} y = {} & 10,7035966439515X_0 + 95,9084649305018X_1 + 2632,5695548877X_2^2 \\ & + 278,16913X_4 - 0,11387765X_6 + \frac{0,0011251868}{(X_3 + 0,0012534792)} \end{aligned} \tag{4.1}$$

$$\begin{aligned} y = {} & 2,28831066837893(-X_3^{1/4} + 0,661062478136522asinh(674,378044732575X_4(X_0 + \\ & 1,4395000868357\sqrt{X_1 + 0,482588100680425X_3 + 0,00293682250964633}) + \\ & (e^{164,323209555671X_2} + \cos\left(\frac{0,10738649}{X_7}\right)e^{-X_6})^2)) \end{aligned} \tag{4.2}$$

Na equação (4.1), a pressão na cabeça do poço (WHP), simbolizada por X_0, revela uma influência moderada e positiva com a FBHP. Esta relação sugere que aumentos na WHP tendem a elevar a FBHP, indicando uma dependência essencial na pressão exercida na entrada do poço sobre a pressão no fundo do mesmo. A taxa de fluxo de água (WFR), X_1, também exibe uma influência moderada e positiva com a FBHP, refletindo o impacto direto da quantidade de água injetada no poço sobre a pressão no fundo do mesmo.

No entanto, a taxa de fluxo de óleo (OFR), representada por X_2, apresenta uma influência fraca e negativa com a FBHP. Este resultado sugere uma relação inversa entre a quantidade de óleo extraído e a pressão no fundo do poço, indicando que um aumento na produção de óleo geralmente resulta em uma diminuição na pressão. A produção diária de água (WPD), X_4, demonstra uma influência moderada e positiva com a FBHP, o que implica que aumentos na produção de água contribuem para o aumento da pressão no fundo do poço.

Por fim, o diâmetro interno do tubo (ID), X_6, mostra uma influência fraca e negativa com a FBHP, sugerindo uma relação inversa entre o diâmetro interno do tubo e a pressão no fundo do poço. Os resultados obtidos da análise da equação (4.1) são consistentes com as conclusões derivadas da matriz de correlação (Figura 3.1) e dos gráficos de dispersão da Figura 3.2, reforçando a validade das relações entre as variáveis consideradas.

Na equação (4.2), as interações múltiplas e interdependentes entre as variáveis na previsão da FBHP complicam a definição de uma influência clara e linear de cada variável isoladamente. Cada variável não apenas afeta diretamente a FBHP, mas também interage com outras variáveis de maneiras complexas e não lineares. A produção diária de água

(WPD, X_4) não só influencia diretamente a FBHP, mas também modula o impacto de variáveis como a pressão na cabeça do poço (WHP, X_0), a taxa de fluxo de óleo (OFR, X_2), e a taxa de fluxo de água (WFR, X_1). Além disso, variáveis como a taxa de fluxo de gás (GFR, X_3) e a temperatura na cabeça do poço (WBHT, X_7) são afetadas de maneira indireta, criando um sistema de relações interdependentes. A relação dinâmica e iterativa entre as variáveis, podem desencadear efeitos em cascata em todo o sistema. Por exemplo, um aumento na produção diária de água (WPD) pode levar a um aumento na a taxa de fluxo de óleo (OFR), devido ao aumento do volume de fluido deslocado, afetando a FBHP. Além disso, a presença de termos na equação, como a função trigonométricas e exponenciais, indica que as interações entre as variáveis não são simplesmente proporcionais, mas sim complexas e não lineares.

Por outro lado, as Equações (4.3) e (4.4) como menor complexidade obtida pelos modelos, demonstram uma complexidade equivalente a 9 nos gráficos das Figuras 4.7 e 4.8, revelando erros de 8,8801% e 7,7261%, respectivamente. A análise das equações (4.3) e (4.4) revela um padrão consistente na influência das variáveis na previsão da pressão no fundo do poço (FBHP). Ambas as equações utilizam três variáveis principais: a pressão na cabeça do poço (WHP, X_0), a produção diária de água (WPD, X_4), e a taxa de fluxo de água (WFR, X_1). A estrutura similar dessas equações permite uma análise conjunta de como cada variável contribui para a determinação da FBHP.

Em ambas as equações, a pressão na cabeça do poço (X_0) é somada à produção diária de água (X_4), multiplicada por um coeficiente (35,59885 na equação (4.3) e 32,964043 na equação (4.4)). Esta soma reflete uma influência direta e combinada de X_0 e X_4 sobre a FBHP, comportamento que é esperado e consistente com a análise das figuras 3.1 e 3.2. O impacto significativo do coeficiente associado a X_4 indica que a produção diária de água tem um peso considerável na determinação da FBHP, conforme previsto pela matriz de correlação.

A taxa de fluxo de água (X_1), presente no denominador de ambas as equações, introduz uma relação crítica. Especificamente, a FBHP é dividida pela diferença entre um valor constante (0,11096645 na equação (4.3) e 0,103881694 na equação (4.4)) e X_1. O posicionamento de X_1 no denominador sugere que à medida que X_1 se aproxima do valor constante, a FBHP aumenta exponencialmente, indicando uma relação direta sensível. Pequenas variações em X_1 próximas ao valor constante podem resultar em grandes mudanças na FBHP, ou seja, quando X_1 tende aos valores das constantes nas equações a FBHP tenderá a infinito.

A semelhança estrutural das equações pode ser interpretada como uma convergência de soluções em algoritmos de PG. A natureza desses algoritmos, incluindo a função de aptidão, a seleção de operadores genéticos e as características do problema, pode contribuir para a emergência de estruturas análogas. Portanto, as equações fornecem uma visão

sobre como diferentes abordagens podem convergir para soluções que compartilham uma essência matemática comum. Esse fenômeno destaca a complexidade e a flexibilidade inerentes ao modelo.

$$y = \frac{X_0 + 35,59885X_4}{0,11096645 - X_1} \tag{4.3}$$

$$y = \frac{X_0 + 32,964043X_4}{0,103881694 - X_1} \tag{4.4}$$

A Tabela 4.3 apresenta um comparativo entre os resultados obtidos pelas equações com o melhor resultado encontrado por Al-Shammari em (4) utilizando o modelo *ANFIS*.

Tabela 4.3 – Resultados da análise estatística das equações e do modelo *ANFIS*.
Fonte: Elaborada pelo autor (2024).

Modelo	MAPE (%)	R
Equação (4.1)	5,12	0,91
Equação (4.2)	5,22	0,90
Equação (4.3)	8,88	0,76
Equação (4.4)	7,72	0,77
ANFIS (4)	4,93	0,93

Em termos de desempenho preditivo, ambas as abordagens foram capazes de produzir previsões precisas da FBHP. Tanto as equações obtidas quanto o modelo *ANFIS* demonstraram baixo Erro Percentual Médio Absoluto (MAPE) e alto Coeficiente de Correlação (R) em conjuntos de teste, indicando uma boa capacidade de previsão. Esses resultados sugerem que tanto as equações quanto o modelo *ANFIS* conseguem capturar efetivamente as relações entre as variáveis de entrada e a FBHP.

No entanto, uma diferença significativa entre as duas abordagens reside na interpretabilidade do modelo. As equações desenvolvidas são expressas em termos de variáveis ambientais dados de poços conhecidos e compreensíveis. Isso permite uma interpretação direta dos resultados e uma compreensão clara das relações entre as variáveis de entrada e a FBHP. Por outro lado, o modelo *ANFIS*, embora seja capaz de fornecer previsões precisas, pode ser menos interpretável devido à sua natureza baseada em redes neurais e lógica *fuzzy*. As relações entre as variáveis de entrada e a FBHP são representadas por meio de neurônios e funções de pertinência *fuzzy*, o que dificulta a interpretação dos resultados.

Além disso, a complexidade computacional também é um aspecto a ser considerado na comparação entre as equações e o modelo *ANFIS*. As equações uma vez obtidas se tornam simples e diretas, o que resulta em baixo custo computacional para implementação e execução. Por outro lado, a execução do modelo *ANFIS* pode exigir mais recursos computacionais devido à natureza das redes neurais.

Em comparação com as correlações empíricas apresentadas na revisão bibliográfica baseadas em variáveis do fluido, as equações apresentadas neste trabalho utilizam variáveis ambientas e dados de produção, proporcionando uma representação alternativa dos fatores que influenciam a FBHP. Além disso, os modelos têm a vantagem de serem continuamente ajustáveis e aprimoráveis à medida que novos dados se tornam disponíveis. No entanto, é importante ressaltar que devido à natureza dos dados utilizados para a obtenção das equações e a falta de parâmetros do fluido presentes na base de dados disponível, não é possível realizar uma comparação direta entre a precisão das correlações empíricas e as equações obtidas.

Os resultados expostos apresentam evidências contundentes acerca da eficácia do modelo PG como uma ferramenta promissora para antecipar a pressão de fundo de poço (FBHP) em operações de extração de petróleo. Ambas as abordagens demonstram habilidade na predição da FBHP, revelando uma margem de erro pequena se comparada a outro método utilizado na mesma base de dados.

Entretanto, é importante ponderar sobre alguns compromissos essenciais ao selecionar o modelo mais adequado. A formulação que permite a inclusão de qualquer tipo de equação pode gerar expressões mais complexas e oferecer uma precisão relativa maior, permitindo uma representação mais detalhada dos fenômenos envolvidos no processo. No entanto, essa complexidade adicional pode acarretar desafios significativos na interpretação e implementação dos resultados. Por outro lado, os modelos restritos a equações polinomiais sacrificam um grau de precisão, mas são notoriamente mais simples e diretos, facilitando a interpretação dos resultados. Essa simplicidade torna o processo de validação e verificação mais eficiente, permitindo uma compreensão clara das relações entre as variáveis. Além disso, o ganho de precisão dos modelos livres é relativamente pequeno quando comparado aos modelos polinomiais, e estes últimos apresentam um menor tempo de processamento.

A decisão acerca do modelo ideal está vinculada aos requisitos específicos da aplicação em questão. Caso a simplicidade e a interpretabilidade das equações sejam prioritárias, a opção pelo modelo restrito a equações polinomiais se mostra prudente. Porém, se a eficiência computacional ou a busca pela máxima precisão absoluta forem prioritárias, a escolha pelo modelo que permite a inclusão de qualquer tipo de equação poderá ser mais apropriada.

5 CONCLUSÃO

Com base nos resultados deste estudo, a aplicação da Programação Genética revelou-se uma abordagem promissora na determinação da Pressão de Fundo de Poço em sistemas de escoamento multifásico na indústria de exploração e produção de petróleo e gás. Através da PG, foram desenvolvidos modelos simbólicos capazes de descrever, de maneira interpretável, a complexa relação entre variáveis operacionais e ambientais e a FBHP, proporcionando estimativas precisas e compreensíveis dessa variável crucial.

A obtenção de modelos simbólicos compreensíveis contribui significativamente para a aplicabilidade prática, permitindo uma melhor compreensão dos fatores que influenciam a FBHP e promovendo uma tomada de decisão mais informada por parte dos profissionais da indústria.

Em síntese, este estudo não apenas avança o conhecimento teórico, mas também oferece contribuições práticas e aplicáveis para a indústria de exploração e produção de petróleo e gás. As compreensões obtidas não só reforçam a relevância da PG, mas também destacam a importância da modelagem avançada na compreensão e otimização de sistemas complexos. Assim, este trabalho aponta para caminhos promissores e áreas de aprimoramento futuro na aplicação da PG na indústria.

REFERÊNCIAS

1 Abdulrauf R Adebayo, Abdulazeez Abdulraheem, & Sunday O Olatunji. Artificial intelligence based estimation of water saturation in complex reservoir systems. Journal of Porous Media, 18(9), 2015.

2 G. H. Aggrey & D. R. Davies. Tracking the State and Diagnosing Downhole Permanent Sensors in Intelligent-Well Completions With Artificial Neural Network. volume All Days of SPE Offshore Europe Conference and Exhibition, pages SPE–107198–MS, 09 2007.

3 Mohammad Ali Ahmadi, Morteza Galedarzadeh, & Seyed Reza Shadizadeh. Low parameter model to monitor bottom hole pressure in vertical multiphase flow in oil production wells. Petroleum, 2(3):258–266, 2016. ISSN 2405-6561.

4 Ahmed Al-Shammari. Accurate Prediction of Pressure Drop in Two-Phase Vertical Flow Systems using Artificial Intelligence. volume All Days of SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, pages SPE–149035–MS, 05 2011.

5 Fahad Hassan Al Shehri, Anton Gryzlov, Tayyar Al Tayyar, & Muhammad Arsalan. Utilizing machine learning methods to estimate flowing bottom-holepressure in unconventional gas condensate tight sand fractured wells in saudi arabia. 2020. Cited by: 7.

6 Nathan Andrews, Nathan J. Bennett, Philippe Le Billon, Stephanie J. Green, Andrés M. Cisneros-Montemayor, Sandra Amongin, Noella J. Gray, & U. Rashid Sumaila. Oil, fisheries and coastal communities: A review of impacts on the environment, livelihoods, space and governance. Energy Research & Social Science, 75:102009, 2021. ISSN 2214-6296.

7 AM Ansari, ND Sylvester, Cem Sarica, Ovadia Shoham, & JP Brill. A comprehensive mechanistic model for upward two-phase flow in wellbores. SPE Production & Facilities, 9(02):143–151, 1994.

8 Emre Artun. Characterizing interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study. Neural Computing and Applications, 28:1729–1743, 2017.

9 Harald Asheim. MONA, An Accurate Two-Phase Well Flow Model Based on Phase Slippage. SPE Production Engineering, 1(03):221–230, 05 1986.

10 Medhat Awadalla & Hassan Yousef. Neural networks for flow bottom hole pressure prediction. International Journal of Electrical & Computer Engineering (2088-8708), 6(4), 2016.

11 Khalid Aziz & George W Govier. Pressure drop in wells producing oil and gas. Journal of Canadian Petroleum Technology, 11(03), 1972.

12 James Edward Baker. An analysis of the effects of selection in genetic algorithms. Vanderbilt University, 1989.

13 Wolfgang Banzhaf, Peter Nordin, Robert E Keller, & Frank D Francone. Genetic programming: an introduction: on the automatic evolution of computer programs and its applications. Morgan Kaufmann Publishers Inc., 1998.

14 Jorge M Barreto. Inteligência artificial no limiar do século xxi. Florianópolis: PPP edições, 97, 1999.

15 Hamid Bazargan & Meisam Adibifard. A stochastic well-test analysis on transient pressure data using iterative ensemble kalman filter. Neural Computing and Applications, 31:3227–3243, 2019.

16 Dale H Beggs & James P Brill. A study of two-phase flow in inclined pipes. Journal of Petroleum technology, 25(05):607–617, 1973.

17 Tobias Blickle & Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. Evolutionary Computation, 4(4):361–394, 1996.

18 Walter Bohm. Exact uniform initialization for genetic programming. Foundations of Genetic Algorithms, pages 379–407, 1996.

19 Ahmed Buhulaigah, Ali S Al-Mashhad, Sulaiman A Al-Arifi, Mohammed S Al-Kadem, & Mohammed S Al-Dabbous. Multilateral wells evaluation utilizing artificial intelligence. In SPE Middle East Oil and Gas Show and Conference, page D031S028R005. SPE, 2017.

20 Kumar Chellapilla. Evolving computer programs without subtree crossover. IEEE Transactions on Evolutionary Computation, 1(3):209–216, 1997.

21 Alexey Cherepovitsyn, Evgeniya Rutenko, & Victoria Solovyova. Sustainable development of oil and gas resources: A system of environmental, socio-economic, and innovation indicators. Journal of Marine Science and Engineering, 9(11), 2021. ISSN 2077-1312.

22 Jiangfeng Cui, Qian Sang, Yajun Li, Congbin Yin, Yanchao Li, & Mingzhe Dong. Liquid permeability of organic nanopores in shale: Calculation and analysis. Fuel, 202: 426–434, 2017.

23 Agoston E Eiben & James E Smith. Introduction to evolutionary computing. Springer, 2015.

24 Christopher Gathercole. An investigation of supervised learning in genetic programming. 1998.

25 David E Goldberg & Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In Foundations of genetic algorithms, volume 1, pages 69–93. Elsevier, 1991.

26 Leonardo Goliatt, Reem Sabah Mohammad, Sani I. Abba, & Zaher Mundher Yaseen. Development of hybrid computational data-intelligence model for flowing bottom-hole pressure of oil wells: New strategy for oil reservoir management and monitoring. Fuel, 350:128623, 2023. ISSN 0016-2361.

27 Leonardo Goliatt, C.M. Saporetti, L.C. Oliveira, & E. Pereira. Performance of evolutionary optimized machine learning for modeling total organic carbon in core samples of shale gas fields. Petroleum, 2023.

28 LE Gomez, O Shoham, Z Schmidt, RN Chokshi, A Brown, & T Northug. A unified mechanistic model for steady-state two-phase flow in wellbores and pipelines. In SPE Annual Technical Conference and Exhibition?, pages SPE–56520. SPE, 1999.

29 George W Govier & Maria Fogarasi. Pressure drop in wells producing gas and condensate. Journal of Canadian Petroleum Technology, 14(04), 1975.

30 GP Learn Documentation. Gp learn documentation, 2023. URL `https://astroautomata.com/PySR/`. Acessado em 17 de Janeiro de 2024.

31 Larry Gritz. A C++ Implementation of Genetic Programming. Department of Electrical Engeneering and Computer Science. The George Washington University, Washington, DC, 1994.

32 M Haenlein & A Kaplan. Guest editorial to the special issue, a brief history of ai: On the past, present, and future of artificial intelligence. California Management Review, 61(4):5–14, 2019.

33 Alton R Hagedorn & Kermit E Brown. Experimental study of pressure gradients occurring during continuous two-phase flow in small-diameter vertical conduits. Journal of Petroleum Technology, 17(04):475–484, 1965.

34 JH Holand. Adaptation in natural and artificial systems, the university of michigan press. Ann Arbour, page 3l, 1975.

35 I e Jahanandish, B Salimifard, & H Jalalifar. Predicting bottomhole pressure in vertical multiphase flowing wells using artificial neural networks. Journal of Petroleum science and engineering, 75(3-4):336–342, 2011.

36 Michael Keller. Oil revenues vs domestic taxation: Deeper insights into the crowding-out effect. Resources Policy, 76:102560, 2022. ISSN 0301-4207.

37 Dmitry Koroteev & Zeljko Tekic. Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future. Energy and AI, 3:100041, 2021.

38 John R. Koza. Hierarchical automatic function definition in genetic programming. In L. DARRELL WHITLEY, editor, Foundations of Genetic Algorithms, volume 2 of Foundations of Genetic Algorithms, pages 297–318. Elsevier, 1993.

39 John R. Koza. Genetic Programming II: Automatic Discovery of Reusable Programs (Complex Adaptive Systems). The MIT Press, first edition, 1994. ISBN 9780262111898.

40 John R Koza. Genetic programming as a means for programming computers by natural selection. Statistics and computing, 4:87–112, 1994.

41 John R Koza, Martin A Keane, Matthew J Streeter, William Mydlowec, Jessen Yu, & Guido Lanza. Genetic programming IV: Routine human-competitive machine intelligence, volume 5. Springer Science & Business Media, 2005.

42 Michael D Kramer & Du Zhang. Gaps: a genetic programming system. In Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000, pages 614–619. IEEE, 2000.

43 Hong Li, Haiyang Yu, Nai Cao, He Tian, & Shiqing Cheng. Applications of artificial intelligence in oil and gas development. Archives of Computational Methods in Engineering, 28:937–949, 2021.

44 Sean Luke. Two fast tree-creation algorithms for genetic programming. IEEE Transactions on Evolutionary Computation, 4(3):274–283, 2000.

45 Ali Mahdy, Wael Zakaria, Ahmed Helmi, Ahmad Sobhy Helaly, & Abdullah M.E. Mahmoud. Machine learning approach for core permeability prediction from well logs in sandstone reservoir, mediterranean sea, egypt. Journal of Applied Geophysics, 220: 105249, 2024. ISSN 0926-9851.

46 Solomon Adjei Marfo, Solomon Asante-Okyere, & Yao Yevenyo Ziggah. A new flowing bottom hole pressure prediction model using m5 prime decision tree approach. Modeling Earth Systems and Environment, 8(2):2065–2073, 2022.

47 Julian Francis Miller & Simon L Harding. Cartesian genetic programming. In Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers, pages 3489–3512, 2009.

48 Erfan Mohagheghian, Habiballah Zafarian-Rigaki, Yaser Motamedi-Ghahfarrokhi, & Abdolhossein Hemmati-Sarapardeh. Using an artificial neural network to predict carbon dioxide compressibility factor at high pressure and temperature. Korean Journal of Chemical Engineering, 32:2087–2096, 2015.

49 Heinz Mühlenbein & Dirk Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm i. continuous parameter optimization. Evolutionary computation, 1 (1):25–49, 1993.

50 Chibuzo Cosmas Nwanwe, Ugochukwu Ilozurike Duru, Charley Anyadiegwu, & Azunna IB Ekejuba. An artificial neural network visible mathematical model for real-time prediction of multiphase flowing bottom-hole pressure in wellbores. Petroleum Research, 8(3):370–385, 2023.

51 J Orkiszewski. Predicting two-phase pressure drops in vertical pipe. Journal of Petroleum technology, 19(06):829–838, 1967.

52 Gisele Pappa, Mario Giacobini, & Zdenek Vasicek. Genetic Programming: 26th European Conference, EuroGP 2023, Held as Part of EvoStar 2023, Brno, Czech Republic, April 12–14, 2023, Proceedings, volume 13986. Springer Nature, 2023.

53 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, *et al.* Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

54 Riccardo Poli. Parallel distributed genetic programming. University of Birmingham, Cognitive Science Research Centre Birmingham, UK, 1996.

55 JK Pucknell, JNE Mason, & EG Vervest. An evaluation of recent"mechanistic"models of multiphase flow for predicting pressure drops in oil and gas wells. In SPE Offshore Europe Conference and Exhibition, pages SPE–26682. SPE, 1993.

56 A Rezaian, A Kordestany, & M Haghighat Sefat. An artificial neural network approach to formation damage prediction due to asphaltene deposition. In SPE Nigeria Annual International Conference and Exhibition, pages SPE–140683. SPE, 2010.

57 Adel M Salem, Mostafa S Yakoot, & Omar Mahmoud. Addressing diverse petroleum industry problems using machine learning techniques: literary methodology- spotlight on predicting well integrity failures. ACS omega, 7(3):2504–2519, 2022.

58 Nagham Amer Sami & Dhorgham Skban Ibrahim. Forecasting multiphase flowing bottom-hole pressure of vertical oil wells using three machine learning techniques. Petroleum Research, 6(4):417–422, 2021. ISSN 2096-2495.

59 Camila Martins Saporetti, Leonardo Goliatt da Fonseca, Egberto Pereira, & Leonardo Costa de Oliveira. Machine learning approaches for petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. Journal of Applied Geophysics, 155:217–225, 2018. ISSN 0926-9851.

60 C.M. Saporetti, D.L. Fonseca, L.C. Oliveira, E. Pereira, & L. Goliatt. Hybrid machine learning models for estimating total organic carbon from mineral constituents in core samples of shale gas fields. Marine and Petroleum Geology, 143:105783, 2022. ISSN 0264-8172.

61 Ebru Akcapinar Sezer, Hakan A Nefeslioglu, & Candan Gokceoglu. An assessment on producing synthetic samples by fuzzy c-means for limited number of data in prediction models. Applied Soft Computing, 24:126–134, 2014.

62 Syed Shujath Ali, M Enamul Hossain, Md Rafiul Hassan, & Abdulazeez Abdulraheem. Hydraulic unit estimation from predicted permeability and porosity using artificial intelligence techniques. In North Africa Technical Conference and Exhibition. OnePetro, 2013.

63 H Sonmez, ERGÜN Tuncay, & CANDAN Gokceoglu. Models to predict the uniaxial compressive strength and the modulus of elasticity for ankara agglomerate. International Journal of Rock Mechanics and Mining Sciences, 41(5):717–729, 2004.

64 Zeeshan Tariq, Abdulazeez Abdulraheem, Mohamed Mahmoud, & Adil Ahmed. A rigorous data-driven approach to predict poisson's ratio of carbonate rocks using a functional network. Petrophysics, 59(06):761–777, 2018.

65 Zeeshan Tariq, Mohamed Mahmoud, & Abdulazeez Abdulraheem. Real-time prognosis of flowing bottom-hole pressure in a vertical well for a multiphase flow using computational intelligence techniques. Journal of Petroleum Exploration and Production Technology, 10:1411–1428, 2020.

66 Leonardo Trujillo, Stephan M Winkler, Sara Silva, & Wolfgang Banzhaf. Genetic Programming Theory and Practice XIX. Springer Nature, 2023.

67 M-J Willis, Hugo G Hiden, Peter Marenbach, Ben McKay, & Gary A Montague. Genetic programming: An introduction and survey of applications. In Second international conference on genetic algorithms in engineering systems: innovations and applications, pages 314–319. IET, 1997.

68 WIPO WIPO. Technology trends 2019: Artificial intelligence. Geneva: World Intellectual Property Organization, 2019.

69 Liuqing Yang, Shoudong Wang, Xiaohong Chen, Wei Chen, Omar M. Saad, Xu Zhou, Nam Pham, Zhicheng Geng, Sergey Fomel, & Yangkang Chen. High-fidelity permeability and porosity prediction using deep learning with the self-attention mechanism. IEEE Transactions on Neural Networks and Learning Systems, 34(7): 3429–3443, 2023.

70 Tina Yu. Structure abstraction and genetic programming. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 1, pages 652–659. IEEE, 1999.

71 Fangfang Zhang, Su Nguyen, Yi Mei, & Mengjie Zhang. Genetic Programming for Production Scheduling. Springer, 2021.

72 Özge Korkmaz. Do oil, coal, and natural gas consumption and rents impact economic growth? an empirical analysis of the russian federation. Resources Policy, 77:102739, 2022. ISSN 0301-4207.

APÊNDICE A – Pseudocódigos

Algoritmo 1: *Grow* - Geração de Árvores em Programação Genética.

```
Grow(profundidade_maxima):
se profundidade_maxima é 0 então
    retorne um terminal aleatório de T
senão
    Escolha aleatoriamente entre crescer uma função (F) ou um terminal (T)
    se for uma função então
        Escolha aleatoriamente uma função f de F
        Crie um nó com a função f
        para cada argumento de f faça
            Adicione GROW(PROFUNDIDADE_MAXIMA - 1) como subárvore
             recursivamente
        fim
    senão
                                                    ▷ Se for um terminal
        retorne um terminal aleatório de T
    fim
fim
retorne a árvore gerada
```

Algoritmo 2: *Ful* - Geração de Árvores Completas em Programação Genética.

```
Full(profundidade_atual, profundidade_maxima)
se profundidade_atual é igual a profundidade_maxima então
    textbfretorne um terminal aleatório de T
senão
    Escolha aleatoriamente uma função f de F
    Crie um nó com a função f
    para cada argumento de f faça
        Adicione FULL(PROFUNDIDADE_ATUAL + 1,
         PROFUNDIDADE_MAXIMA) como subárvore recursivamente
    fim
fim
retorne a árvore gerada
```

Algoritmo 3: *Random-branch* - Geração de Árvores em Programação Genética.

RandomBranch(S)
se S é terminal **então**
  retorna Aleatoriamente um terminal
senão
  Escolha aleatoriamente um não-terminal n com aridade $a \leq S$
  Seja b_n a aridade de n
  Crie uma lista vazia $args$
  para i de 1 até b_n **faça**
    $a_i \leftarrow$ RandomBranch($\lfloor \frac{S}{b_n} \rfloor$)
    Adicione a_i à lista $args$
  fim
  retorna n com todos os argumentos preenchidos com os valores em $args$
fim

Algoritmo 4: *PTC1* - Geração de Árvores em Programação Genética.

Dados
Profundidade máxima D
Conjunto de funções F e de terminais T
A probabilidade p de escolher uma função
As probabilidades q_t e q_f para cada $t \in T$ e $f \in F$
PTC1(d)
se $d = D$ **então**
  retorna aleatoriamente um terminal de T baseado em q_t
senão
  se com probabilidade p **então**
    Escolha aleatoriamente $f \in F$ baseado em q_f
    para cada argumento a de f **faça**
      Preencha a com PTC1($d+1$)
    fim
    retorna f com todos os argumentos preenchidos
  senão
    retorna aleatoriamente um terminal de T baseado em q_t
  fim
fim

Algoritmo 5: *PTC2* - Geração de Árvore em Programação Genética

Dados
Tamanho máximo S
Distribuição de probabilidades $w_1, w_2, \ldots, w_S$ para árvores de tamanho 1 a S
PTC2(d)
Escolha aleatoriamente o tamanho da árvore s de acordo com a distribuição de probabilidades $w_1, w_2, \ldots, w_S$
se $d = s$ **então**
  return aleatoriamente um terminal de T
senão
  se com probabilidade p **então**
    Escolha aleatoriamente $f \in F$ baseado em q_f
    para cada argumento a de f **faça**
      Preencha a com PTC2$(d+1)$
    fim
    return f com todos os argumentos preenchidos
  senão
    return aleatoriamente um terminal de T
  fim
fim

Algoritmo 6: Operador de Cruzamento (Crossover) na Programação Genética

Cruzamento(programa_pai1, programa_pai2):
▷ Escolha aleatória de pontos de cruzamento ponto_cruzamento1 ← EscolhaAleatóriaPontoCruzamento(programa_pai1) ponto_cruzamento2 ← EscolhaAleatóriaPontoCruzamento(programa_pai2)
▷ Intercâmbio de subárvores abaixo dos pontos de cruzamento $programa\_filho1$ ← SubárvoreAtéPontoCruzamento(programa_pai1, ponto_cruzamento1) SubárvoreApósPontoCruzamento(programa_pai2, ponto_cruzamento2) $programa\_filho2$ ← SubárvoreAtéPontoCruzamento(programa_pai2, ponto_cruzamento2) SubárvoreApósPontoCruzamento(programa_pai1, ponto_cruzamento1)
retorne $programa\_filho1, programa\_filho2$

Algoritmo 7: Operador de Mutação na Programação Genética

Mutação(programa_pai):
▷ Escolha aleatória de um nó para mutação $no\_mutacao$ ← EscolhaAleatóriaNoMutacao(programa_pai)
▷ Geração de uma nova árvore aleatória $nova\_subarvore$ ← GereNovaSubárvoreAleatória()
// Substituição da subárvore mutada pela nova subárvore $programa\_filho$ ← SubstituirSubárvore(programa_pai, no_mutacao, nova_subarvore)
retorne $programa\_filho$