



APLICAÇÃO DE UMA PROGRAMAÇÃO GENÉTICA GRAMATICAL MULTIOBJETIVO NA INFERÊNCIA DA MÁXIMA DEFORMAÇÃO LONGITUDINAL DE DUTOS COM AMASSAMENTO

João Marcos de Freitas^a

joao@ice.ufjf.br

Heder Soares Bernardino^a

heder@ice.ufjf.br

João Nisan Correia Guerreiro^b

joao@lncc.br

Helio José Corrêa Barbosa^{a,b}

hcbm@lncc.br

^aUniversidade Federal de Juiz de Fora (UFJF)

^bLaboratório Nacional de Computação Científica (LNCC/MCTI)

Resumo. *Existem vários fatores que dificultam a extração de combustíveis, tais como os danos sofridos pelos dutos. Técnicas de programação genética vem sendo largamente adotadas para a geração de modelos em forma simbólica e neste trabalho uma variante gramatical será utilizada na busca por formas de relacionar dados de dutos e outros fatores com a sua máxima deformação longitudinal. Além disso, ideias de otimização multiobjetivo são incorporadas no mecanismo de descoberta de conhecimento afim de gerar modelos com baixa acurácia e complexidade. Para isso, um problema multiobjetivo é formulado, onde as funções objetivo são o erro quadrático médio e a profundidade da árvore de derivação do modelo candidato. A Programação Genética Gramatical Multiobjetivo desenvolvida aqui adota os procedimentos de ordenação por não dominância e distância de multidão. Os resultados encontrados nos experimentos computacionais indicam que a aplicação da técnica a esse problema é promissora.*

Keywords: *Programação Genética, Programação Genética Gramatical Multiobjetivo, Inferência de Modelos, Engenharia de Dutos*

1 INTRODUÇÃO

Durante séculos o homem buscou meios de automatizar processos. Com a ajuda de métodos computacionais foi possível aprimorar esses processos para, entre outros, auxiliar a descoberta de conhecimento em geral, e nas engenharias, em particular.

Na engenharia de tubos, por exemplo para o transporte de combustíveis, é importante saber quando um duto deve sofrer algum tipo de manutenção. Os danos sofridos pelos tubos podem levar a graves problemas ambientais e econômicos. Afim de evitar esses problemas é importante determinar a situação do duto. Para isso é necessária uma análise estrutural sofisticada, onde pequenas variações nos dados podem gerar uma grande imprecisão. Uma análise precisa pode ser obtida utilizando cálculos via o Método dos Elementos Finitos, mas este processo é computacionalmente custoso. No caso em que há deformação sem perda de material, essa informação pode ser derivada da máxima deformação longitudinal do duto, combinada ao resultados de outros modelos já conhecidos (Noronha et al., 2008). Assim, o cálculo da máxima deformação longitudinal em dutos com amassamento é um problema importante para a engenharia de tubos.

Dentre as diversas técnicas de aprendizagem de máquina para a inferência de modelos a partir de dados, a Programação Genética (PG) ganha destaque pela inferência de modelos simbólicos. A solução é um programa capaz de resolver um dado problema. A PG (Koza, 1992) é uma meta-heurística bio-inspirada baseada no princípio darwiniano da Seleção Natural.

Bernardino et al. (2011) apresenta a aplicação da técnica de Programação Imunológica Gramatical (PIG) para este problema com o uso de diversas gramáticas para gerar os modelos, obtendo resultados significativos utilizando o Método dos Mínimos Quadrados para ajustar os coeficientes lineares dos modelos candidatos. O PIG utiliza uma técnica imuno-inspirada como mecanismo de busca e uma representação binária das soluções candidatas, que são decodificadas através de mapeamentos em um programa (aqui, modelo) utilizando uma gramática formal.

A aplicação de uma Programação Genética Gramatical (PGG) capaz de resolver e reforçar informações previamente encontradas por Bernardino et al. (2011) foi apresentada por de Freitas et al. (2014). A PGG evolui modelos candidatos gerados a partir de gramáticas formais, representando os programas em estruturas de árvores de derivação, utilizando operações de seleção, aplicando operadores de recombinação e mutação, e usando elitismo.

Apesar de suas vantagens, a complexidade das soluções encontradas pode dificultar a interpretação dos modelos. Ambos trabalhos apresentados previamente (Bernardino et al., 2011; de Freitas et al., 2014) buscam por modelos que minimizem o erro quadrático médio, mas as melhores soluções encontradas segundo essa medida de qualidade são expressões muito complexas. Uma forma de buscar por modelos/programas acurados e ao mesmo tempo simples é otimizando esses dois objetivos ao mesmo tempo. Assim, alguns trabalhos propondo técnicas de PG em ambiente multiobjetivo podem ser encontrados na literatura (Bleuler et al., 2001; Nguyen et al., 2013; Zhao, 2007).

A utilização de uma técnica de Programação Genética Gramatical Multiobjetivo (PGGM) para resolver o problema da inferência da máxima deformação longitudinal de

duto com amassamento é proposta aqui. Através desse método, busca-se por modelos que minimizem tanto o erro quadrático médio quanto a complexidade (definida aqui como a profundidade da árvore de derivação). Com um conjunto de soluções de compromisso, o engenheiro de tubos pode então escolher *a posteriore* a que julgar mais adequada. Finalmente, os coeficientes lineares dos modelos são ajustados via o Método dos Mínimos Quadrados.

2 PROBLEMA

Problemas de modelagem em que há a necessidade de fórmulas aproximadas é comum e existem problemas nos quais estas são complexas ou em que há dificuldade de obter dados via medição. Pode-se destacar para esse tipo de situação as técnicas de Programação Genética (PG), por serem capazes de encontrar fórmulas e aprimorar as já existentes. Ainda é possível limitar o uso variáveis e até mesmo controlar a complexidade dos modelos. Assim, o uso de uma técnica de PG Multiobjetivo para complementar o cálculo para as componentes de flexão desenvolvidos por Noronha et al. (2008) gerando um modelo que defina a máxima deformação longitudinal de um tubo para auxiliar as tomadas de decisão referentes às manutenções dos tubos é proposto aqui.

Ao ser detectado um amassamento (deformação sem perda de material) é necessário ter uma análise rápida e precisa das condições atuais do duto para decidir as ações de manutenção. Códigos internacionais como o ASME B31.8 (2007) indicam que a decisão pode ser tomada de acordo com limites baseados na deformação equivalente de von Mises que define o estado de deformação de sólidos. Entretanto, uma análise estrutural sofisticada é necessária para que este método seja adotado. Uma alternativa é utilizar o Método dos Elementos Finitos, o que é inviabilizado por ser computacionalmente custoso. Portanto, verifica-se a necessidade de se encontrar um modelo que seja capaz de auxiliar a tomada de decisão de forma acurada, com baixa complexidade e com custo computacional reduzido.

A ASME B31.8 (2007) indica que a deformação do duto se baseia nas deformações de flexão e na maior deformação longitudinal da membrana e Noronha et al. (2008) apresentou resultados similares àqueles encontrados pelo Método de Elementos Finitos para as componentes de flexão. Entretanto, verifica-se a necessidade de se encontrar modelos que representem mais adequadamente a máxima deformação longitudinal de um tubo com amassamento (Bernardino et al., 2011; de Freitas et al., 2014).

O banco de dados utilizado aqui é o mesmo usado por Bernardino et al. (2011) e de Freitas et al. (2014). Este banco é composto por 554 registros contendo a deformação longitudinal máxima de tubos com diferentes valores de diâmetro externo (D), espessura da parede (t), profundidade de indentação (d) e o diâmetro do indentador (Φ). Os valores de deformação foram obtidos através de simulações computacionais utilizando o Método de Elementos Finitos (Bernardino et al., 2011). A Figura 1 ilustra um duto e alguns dos atributos utilizados.

Juntamente com os parâmetros descritos anteriormente são utilizados no modelo: a tensão circunferencial (σ_θ), induzida pela pressão interna, e a tensão de escoamento do material (σ_y). O conjunto de variáveis adimensionadas que seguem foram apresentadas por Bernardino et al. (2011):

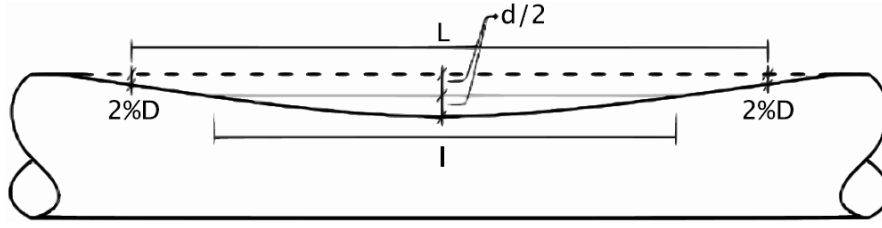


Figura 1: Ilustração do duto com amassamento (Bernardino et al., 2011).

- (i) $x_1 = t/D$
- (ii) $x_2 = \sigma_\theta/\sigma_y$
- (iii) $x_3 = d/D$
- (iv) $x_4 = \Phi/D$
- (v) $x_5 = v$, onde v é o coeficiente de Poisson
- (vi) $x_6 = E_t/E$, onde E é o módulo de Young e E_t um módulo tangente
- (vii) $x_7 = d/l$, onde l é o tamanho da deformação em $0.5 d$
- (viii) $x_8 = d/L$, onde L é o tamanho da deformação em $2\% D$

Como um mesmo material foi adotado nas simulações, então as variáveis x_5 e x_6 não são usadas aqui. Portanto, assim como nos trabalhos de Bernardino et al. (2011) e de Freitas et al. (2014), as variáveis x_1, x_2, x_3, x_4, x_7 e x_8 serão adotadas aqui para encontrar uma forma simbólica para $\epsilon : \mathbb{R}^6 \rightarrow \mathbb{R}$ tal que

$$\epsilon = f(x_1, x_2, x_3, x_4, x_7, x_8)$$

representa um modelo para máxima deformação longitudinal de tubos com amassamento. Neste trabalho, pretende-se buscar por modelos com alta acurácia e baixa complexidade.

3 PROGRAMAÇÃO GENÉTICA GRAMATICAL

A Programação Genética (PG) é uma especialização dos Algoritmos Genéticos (AGs) (Holland, 1975), uma metaheurística amplamente utilizada para resolver problemas de otimização. O AG simula a evolução natural para melhorar uma população de indivíduos baseado em algum critério, representado pela aptidão. A PG opera sobre a mesma perspectiva porém é usada para gerar modelos em forma simbólica (programas).

No processo evolutivo são criados novos indivíduos através de operações sobre indivíduos da população, como cruzamento e mutação, gerando uma nova população a cada geração. Os novos indivíduos, juntamente com os antigos, passam por um processo de seleção (substituição da população) que escolhe aqueles que continuarão o processo de otimização. Este processo evolutivo é como apresentado no Algoritmo 1.

Na PG, os indivíduos são modelos são representados comumente por estruturas de árvore. Cada modelo representa simbolicamente um programa que pode ser executado e, assim, avaliado. A população é comumente formada por um conjunto de tamanho fixo

Algoritmo 1: Pseudo-código de uma técnica de Programação Genética.

```

1 Criar aleatoriamente a população inicial  $P$ ;
2 Avaliar todos os indivíduos de  $P$ ;
3 enquanto critério de parada não for satisfeito faça
4   enquanto  $|P_{tmp}| < |P|$  faça
5     Selecionar indivíduos  $p_1$  e  $p_2$  por torneio;
6     Copiar  $p_1$  em  $p'_1$  e  $p_2$  em  $p'_2$ ;
7     se  $\text{aleatório} \leq p_{\text{cruzamento}}$  então
8       Cruzar  $p'_1$  com  $p'_2$ ;
9     se  $\text{aleatório} \leq p_{\text{mutação}}$  então
10      Aplicar operadores de mutação em  $p'_1$  e  $p'_2$ ;
11      Avaliar  $p'_1$  e  $p'_2$ ;
12      Inserir  $p'_1$  e  $p'_2$  em  $P_{tmp}$ ;
13    $P \leftarrow \text{elite de } P \cup \text{elite de } P_{tmp}$ ;
14   descartar  $P_{tmp}$ ;
15 retorna melhor indivíduo encontrado;

```

de indivíduos. Assim, como nos AGs, as operações de cruzamento e mutação compõem a PGG. No cruzamento dois indivíduos geram descendentes com a carga genética dos pais. A mutação adiciona variabilidade genética na população alterando o genótipo de um indivíduo. O processo de evolução faz uma seleção de indivíduos da população para participar do processo reprodutivo, então os indivíduos gerados são classificados e os mais aptos são mantidos para a próxima geração. Esse processo é repetido até que um critério de parada seja atendido.

Na PG, mesmo tendo uma boa representação dos modelos, ainda é possível que existam soluções inválidas. Também, em sua forma original não é possível introduzir vies ou fixar valores nos modelos. As Gramáticas Formais Livres de Contexto (GLCs) podem ser utilizadas para restringir o espaço de busca para que apenas programas válidos sejam criados e disponibilizando uma forma simples de introdução de vies. Sua aplicação no campo da PG incide diretamente sobre a estrutura dos programas definindo os componentes que podem ser utilizados (tais como funções e variáveis) e suas regras de combinação (Augusto, 2004). de Freitas et al. (2014) apresenta uma PG utilizando Gramáticas Formais para a inferência da máxima deformação longitudinal de dutos com amassamento. Através da GLC pode-se estabelecer regras nas relações entre os operadores e operandos. Por exemplo, pode-se permitir que operadores lógicos utilizem apenas operandos lógicos ou que um operador de raiz quadrada nunca atue sobre uma variável particular.

A Gramática atua como uma geradora de sentenças obedecendo as regras definidas nela (Augusto, 2004) e pode ser definida por uma quádrupla $G = (N, \Sigma, S, P)$, onde:

- N : alfabeto de não terminais (elementos auxiliares da gramática)
- Σ : alfabeto de terminais (aqueles que aparecem nas palavras da linguagem)
- S : símbolo inicial; $S \in N$
- P : produções

Pode-se definir regras para cada elemento de N que são chamadas de produções. Uma ilustração de regra de uma GLC pode ser como:

$$\langle NT \rangle ::= prod1 \mid prod2 \mid prod3$$

onde $\langle NT \rangle$ é um não-terminal, e $prod1$, $prod2$ e $prod3$ são produções de $\langle NT \rangle$. Segue um exemplo de um conjunto de regras de produção para expressões aritméticas:

$$\begin{aligned} \langle expr \rangle &::= \langle expr \rangle \mid \langle expr \rangle \langle expr \rangle \langle op \rangle \mid \langle num \rangle \\ \langle op \rangle &::= + \mid - \mid * \mid / \\ \langle num \rangle &::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

Gramáticas formais podem ser utilizadas para limitar a representação dos modelos candidatos. Ao adotar uma representação em estrutura de árvore combinada com GLC, os programas passam a ser árvores de derivação. Exemplos de indivíduos sob esta perspectiva são apresentados nas Figuras 2 e 3.

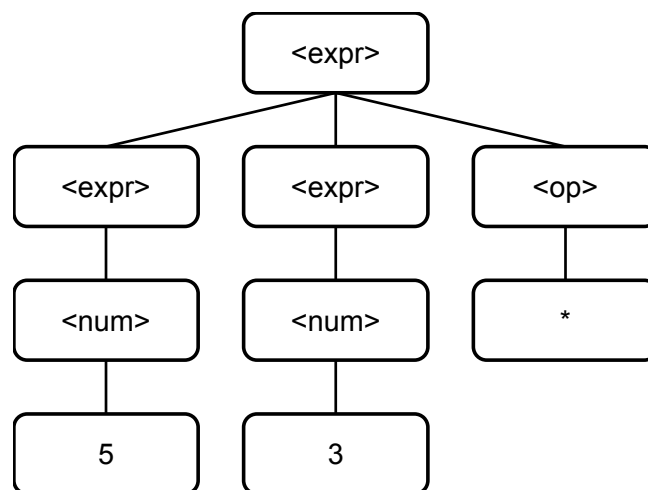


Figura 2: Exemplo de representação para o modelo “5 * 3”.

O cruzamento na PGG deve respeitar as regras da gramática formal adotada. A recombinação entre dois modelos candidatos A e B é ilustrada na Figura 4. No exemplo de cruzamento entre indivíduos A e B, uma subárvore de A é selecionada aleatoriamente e então é sorteada uma subárvore em B onde o nó raiz obedece à mesma produção da gramática. Estas subárvores são então trocadas entre A e B dando origem a dois (novos) modelos A' e B'.

A Mutação (ilustrada na Figura 5) também deve ser realizada de modo que o modelo resultante atenda às regras da gramática. Aqui a mutação é operada como segue: um nó não-terminal do indivíduo é sorteado e, a partir dele, é derivada uma nova subárvore baseada em sua regra.

Com o objetivo de limitar a complexidade dos modelos gerados é comum incorporar uma restrição de profundidade máxima nas árvores. Desta forma, os procedimentos de geração da população inicial, cruzamento e mutação estão sujeitos a essa restrição.

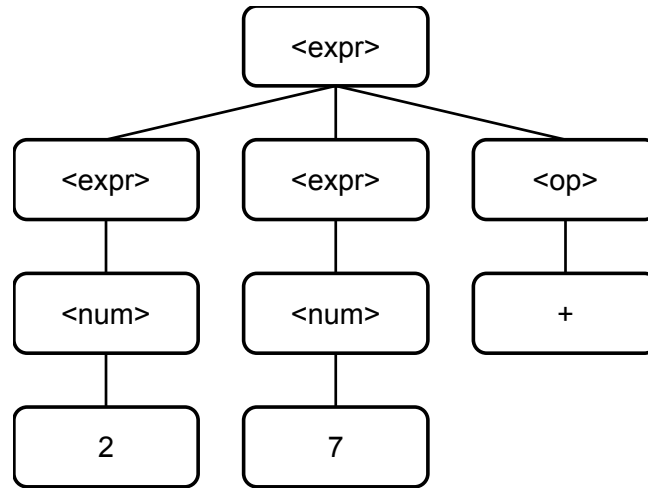


Figura 3: Exemplo de representação para o modelo “2 + 7”.

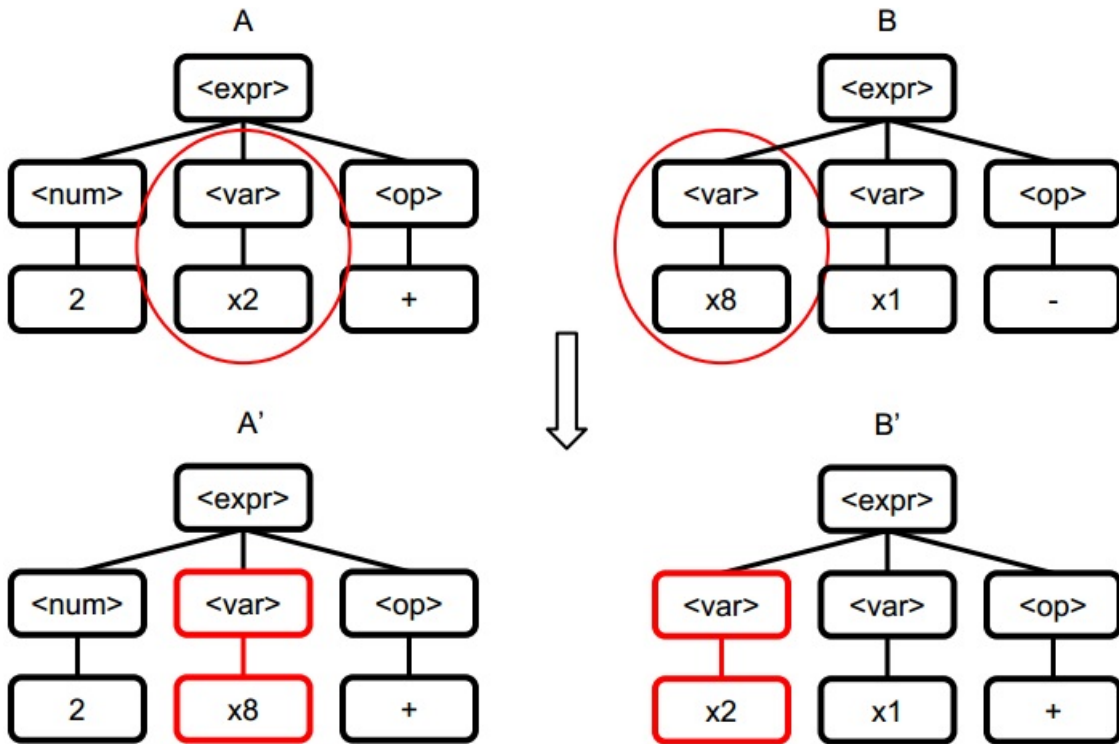


Figura 4: Exemplo de Cruzamento.

Utilizando essas ideias, a PGG é capaz de gerar modelos compiláveis e compreendíveis ao usuário, facilitando a descoberta de conhecimento. Apesar das muitas vantagens, existem limitações computacionais para a aplicação da PG em problemas reais, como aqueles em que os modelos são extensos ou requerem simulações computacionalmente muito custosas. No entanto, com a evolução da tecnologia e disponibilidade de computadores com cada vez mais memória e poder de processamento, a PG começa a se tornar uma alternativa apropriada. Dessa forma, é adequado que a PG seja executada em ambientes de alto desempenho com implementações massivamente paralelas e/ou distribuídas. Finalmente,

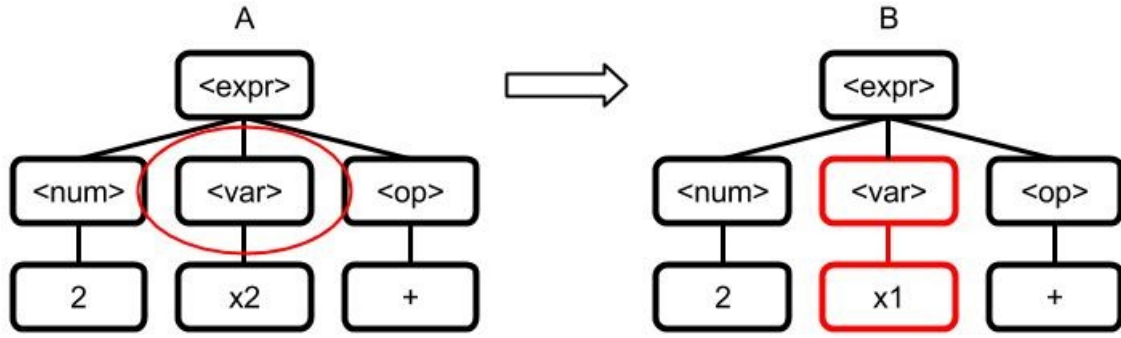


Figura 5: Exemplo de Mutação.

pode-se obter modelos não apenas acurados, mas também com baixa complexidade ao introduzir mais um objetivo no problema de otimização. Portanto, uma abordagem multiobjetivo do problema de modelagem a partir de dados aparece como uma alternativa adequada.

4 PGG MULTIOBJETIVO

A aptidão é uma medida de qualidade dos indivíduos e o processo evolutivo se dá pela seleção dos mais aptos. Na PG, a avaliação é feita através da execução do programa que o indivíduo representa.

Em problemas de regressão/classificação simbólica, o programa é executado sobre um conjunto de amostras para comparar o resultado esperado com aquele gerado pelo programa. Aqui, a acurácia de um modelo candidato é avaliada através do erro calculado como

$$\text{erro} = \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2 \quad (1)$$

onde y_i é o valor esperado, $f(x_i)$ é a avaliação do programa candidato em relação à entrada x_i (um vetor) e m é o número de dados de treinamento.

Entretanto, como já foi discutido anteriormente, a minimização desse erro é apenas uma forma de avaliar a qualidade dos modelos. Afim de obter modelos simples, verificou-se a possibilidade de buscar por soluções não apenas acuradas, mas também com baixa complexidade. Uma forma de avaliar a complexidade de um modelo representado por uma árvore é por meio de sua altura. Apesar de haver uma tendência do tamanho do modelo e a profundidade da árvore de derivação serem diretamente proporcionais, é importante ressaltar que essa medida é uma aproximação para a complexidade e que outras características de complexidade poderiam ser adotadas.

A altura da árvore é calculada como a maior distância entre a raiz (não terminal inicial) e todas as suas folhas (terminais). Como exemplo, pode-se verificar que as árvores nas Figuras 2 e 3 possuem ambas altura 3.

Pretende-se assim, buscar por modelos com menor erro (Equação 1) e menor complexidade (considerando a altura da árvore de derivação). Entretanto, não é difícil perceber

que esses objetivos são conflitantes: modelos mais complexos tentem a ser mais acurados enquanto que modelos mais simples geralmente apresentam um valor de erro alto. Portanto, pretende-se atacar aqui um problema de otimização multiobjetivo. Para isso, as componentes de seleção (para recombinação e substituição da população) presentes no Algoritmo 1 foram alteradas afim de obter um conjunto de soluções em que não é possível dizer que uma é melhor que outra, ou seja, não dominadas entre si.

4.1 Tratamento dos Múltiplos Objetivos

Introduzida por Pareto (1896), o conceito de dominância é base para muitos algoritmos evolutivos, como por exemplo para o NSGA-II proposto por Deb et al. (2002) e que é atualmente uma das metaheurísticas mais populares para problemas de otimização multiobjetivo. Sejam p_1 e p_2 dois programas candidatos, f_i a i -ésima função objetivo do problema, $i = 1, \dots, P$, e P é o número de objetivos, então

- se $f_i(p_1) \leq f_i(p_2)$, $i = 1, \dots, P$, e $\exists j \in \{1, \dots, P\} \mid f_i(p_1) < f_i(p_2)$ então p_1 domina p_2 ;
- se $f_i(p_2) \leq f_i(p_1)$, $i = 1, \dots, P$, e $\exists j \in \{1, \dots, P\} \mid f_i(p_2) < f_i(p_1)$ então p_2 domina p_1 ;
- caso contrário, p_1 e p_2 são ditos não dominados entre si.

Seguindo este conceito, Deb et al. (2002) propôs o NSGA-II, em que as soluções candidatas são comparadas utilizando (i) Classificação por Não Dominância (*Non-dominated Sorting*) e (ii) Distância de Multidão (*Crowding Distance*). Desta forma, enquanto o primeiro indica uma preferência por indivíduos mais próximos do conjunto ótimo de Pareto, o segundo seleciona aqueles que estão em regiões menos populadas (em relação ao espaço das funções objetivo).

Na Classificação por Não Dominância as soluções são separadas em dominadas e não dominadas. As soluções não dominadas são classificadas como sendo preferidas sobre as demais e retiradas do conjunto de soluções. O processo é então repetido utilizando somente os indivíduos dominados até que toda a população seja classificada. A Figura 6 ilustra a Classificação por Não Dominância.

Em um problema bi-objetivo, a Distância de Multidão é o semiperímetro de um retângulo cujos vértices são os vizinhos mais próximos (segundo os valores dos objetivos). Quanto maior este retângulo, mais distante a solução se encontra das soluções vizinhas e maior é a preferência por ela. Além disso, um valor infinito é atribuído à Distância de Multidão dos pontos extremos para que eles sejam sempre mantidos. A Figura 7 ilustra como a Distância de Multidão é calculada para o i -ésimo indivíduo numa dada classificação.

Através da Classificação por Não Dominância e da Distância de Multidão os indivíduos podem então ser comparados. Nota-se que em um problema de otimização com dois objetivos, a única comparação em que não há preferência é entre as duas soluções candidatas localizadas nos extremos de um conjunto de soluções não dominadas de uma classificação. Neste caso, um sorteio aleatório é realizado.

Sendo possível comparar as soluções candidatas, as etapas de seleção podem enfim ser realizadas. Neste trabalho foi adotado um torneio para a seleção dos progenitores e

a substituição da população (i) junta as populações de pais e filhos, (ii) classifica-a por não dominância e (iii) seleciona os melhores indivíduos seguindo a Classificação por Não Dominância e Distância de Multidão para compor a nova população. A Figura 8 ilustra a substituição da população adotada aqui.

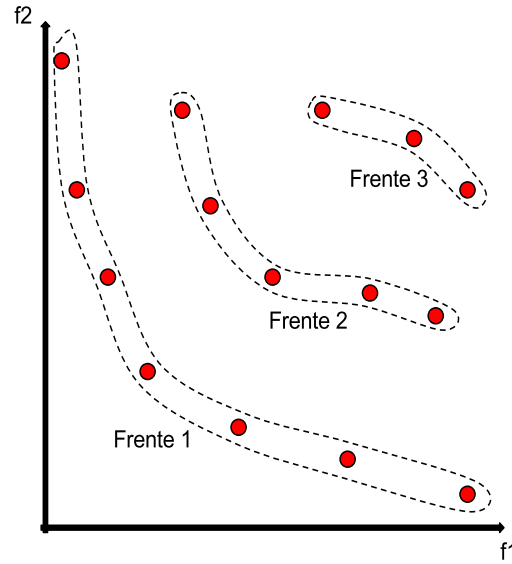


Figura 6: Ilustração da Classificação por Não Dominância.

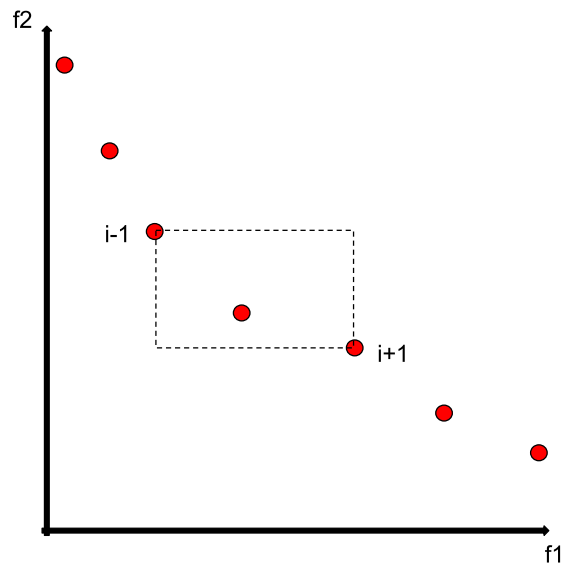


Figura 7: Ilustração do cálculo da Distância de Multidão.

Portanto, a técnica de programação genética gramatical multiobjetivo (PGGM) usada aqui combina a PGG apresentada na Seção 3 com o tratamento dos múltiplos objetivos do NSGA-II. Pretende-se assim evoluir modelos em forma simbólica para a máxima deformação longitudinal de dutos com amassamento adotando o erro quadrático médio e a altura da árvore de derivação como objetivos (a serem minimizados).

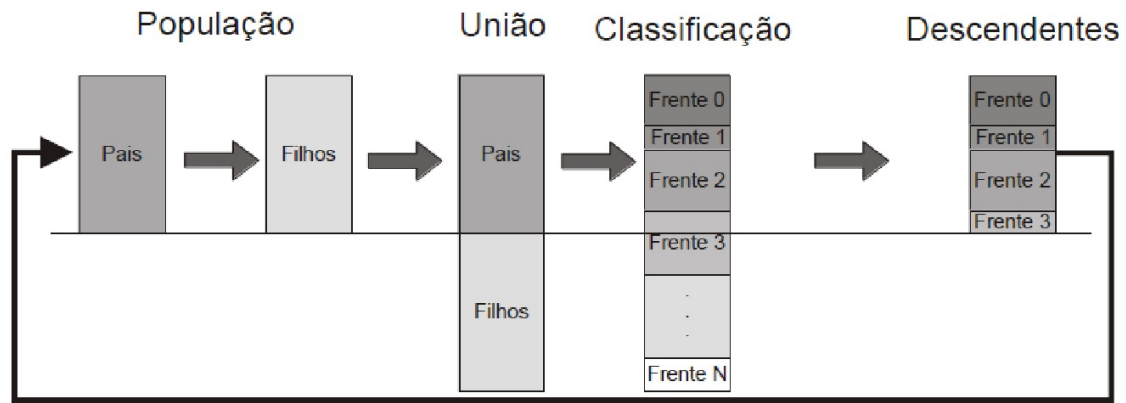


Figura 8: Ilustração da substituição da população adotada aqui.

5 EXPERIMENTOS COMPUTACIONAIS

Experimentos computacionais foram realizados afim de encontrar um conjunto de modelos (melhores segundo os dois objetivos adotados) e, quando possível, compará-los com outros encontrados na literatura. Os dados descritos na Seção 2 foram utilizados durante a evolução e para avaliar a capacidade de generalização dos modelos encontrados. Para isso, os dados foram separados em um conjunto de treinamento e outro de teste. A separação dos grupos é feita de forma aleatória sorteando uma porcentagem (90%) dos registros para treinamento e utilizando os demais (10%) como teste. Essa divisão dos 554 registros seguiu o mesmo padrão adotado por Bernardino et al. (2011). Vale ressaltar que os conjuntos de treinamento e teste são sorteados novamente em cada execução independente.

de Freitas et al. (2014) realizou um estudo para definir bons parâmetros para o tamanho da população e elitismo para o PGG ao resolver o problema atual. Os mesmos parâmetros são adotados aqui, sendo eles: tamanho da população igual a 50 indivíduos e elitismo igual a 10% da população. Por causa de limitações computacionais, a altura das árvores foi limitada a 30. Foram utilizadas duas gramáticas nos experimentos: uma mais restritiva e outra menos restritiva em relação à forma possível dos modelos.

Além disso, são realizadas comparações dos resultados encontrados pela PGG utilizando o Método dos Mínimos Quadrados (MQ), referenciada aqui por PGGM, para o ajuste dos coeficientes lineares do modelo (Bernardino et al., 2011) com a PGG apresentada por de Freitas et al. (2014). Em todos os experimentos, 30 execuções independentes foram realizadas.

As gramáticas simples e constrita elaboradas para o presente trabalho são apresentadas a seguir. Nota-se que as expressões assumem representação pós-fixada. O operador “!” indica a separação das funções de base para o ajuste dos coeficientes lineares do Método dos Mínimos Quadrados. Para os casos sem mínimos quadrados, as gramáticas apenas não incluem a geração de uma combinação linear de funções de base utilizando o operador “!”.

Gramática simples:

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle \langle \text{expr} \rangle ! \mid \langle \text{base} \rangle$
 $\langle \text{base} \rangle ::= \langle \text{base} \rangle \langle \text{base} \rangle \langle \text{op} \rangle \mid \langle \text{base} \rangle \langle \text{uop} \rangle \mid \langle \text{var} \rangle \mid \langle \text{num} \rangle$
 $\langle \text{op} \rangle ::= + \mid - \mid * \mid / \mid \text{pow}$
 $\langle \text{uop} \rangle ::= \log \mid \exp \mid \text{sqrt}$
 $\langle \text{var} \rangle ::= x_1 \mid x_2 \mid x_3 \mid x_4 \mid x_7 \mid x_8$
 $\langle \text{num} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Gramática Constrita:

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle \langle \text{expr} \rangle ! \mid \langle \text{ev} \rangle$
 $\langle \text{ec} \rangle ::= \langle \text{ec} \rangle \langle \text{ec} \rangle \langle \text{op} \rangle \mid \langle \text{ec} \rangle \langle \text{uop} \rangle \mid \langle \text{num} \rangle$
 $\langle \text{ev} \rangle ::= \langle \text{ev} \rangle \langle \text{ecv} \rangle * \mid \langle \text{ecv} \rangle \langle \text{ev} \rangle * \mid \langle \text{ev} \rangle \langle \text{ecv} \rangle / \mid \langle \text{ev} \rangle \langle \text{uop} \rangle$
 $\quad \mid \langle \text{ev} \rangle \langle \text{ec} \rangle \text{pow} \mid \langle \text{var} \rangle$
 $\langle \text{ecv} \rangle ::= \langle \text{ecv} \rangle \langle \text{ecv} \rangle \langle \text{op} \rangle \mid \langle \text{ecv} \rangle \langle \text{uop} \rangle \mid \langle \text{ecv} \rangle \langle \text{ec} \rangle \text{pow}$
 $\quad \mid \langle \text{num} \rangle$
 $\langle \text{op} \rangle ::= + \mid - \mid * \mid / \mid \text{pow}$
 $\langle \text{uop} \rangle ::= \log \mid \exp \mid \text{sqrt}$
 $\langle \text{var} \rangle ::= x_1 \mid x_2 \mid x_3 \mid x_4 \mid x_7 \mid x_8$
 $\langle \text{num} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

5.1 Análise dos Resultados

As Figuras 9, 10, 11 e 12 apresentam os conjuntos de Pareto finais, ou seja, as soluções não dominadas ao juntar os modelos encontrados ao final de todas as execuções independentes para cada combinação de (i) gramática (simples ou constrita) e (ii) uso ou não de MQ. Além do conjunto de Pareto final (em azul) os gráficos incluem: um modelo simples (ASME B31.8, 2007), o melhor modelo encontrado pelos especialistas (indicado por Bernardino et al. (2011)) e, nas Figuras 9 e 11, o resultado da PGG (de Freitas et al., 2014). Os valores de erro foram calculados utilizando os dados de teste (não utilizados durante a evolução). Assim, para gerar os gráficos (i) juntou-se os resultados encontrados ao fim das execuções independentes de um dado algoritmo, (ii) avaliou-se os modelos segundo os dados de teste e (iii) selecionou-se os não dominados.

Pode-se observar nos gráficos que: (i) a exploração biobjetivo do problema resulta em resultados mais acurados do que a PGG (mono-objetivo); (ii) melhores resultados são obtidos utilizando MQ; e (iii) foram obtidos resultados mais acurados com as gramáticas simples (com e sem MQ) do que as restritas.

As Tabelas 1 e 2 apresentam alguns modelos obtidos utilizando cada tipo de gramática e usando (ou não) MQ. Foram selecionados os melhores modelos com árvores de derivação com altura 4-8. É importante ressaltar que os modelos com árvore de derivação de alturas 4 e 5 tem a mesma forma para os casos simples e constrito da Tabela 1, bem como para ambos os casos na Tabela 2. Também, apesar de não evidenciado no texto, as variáveis x_3 ,

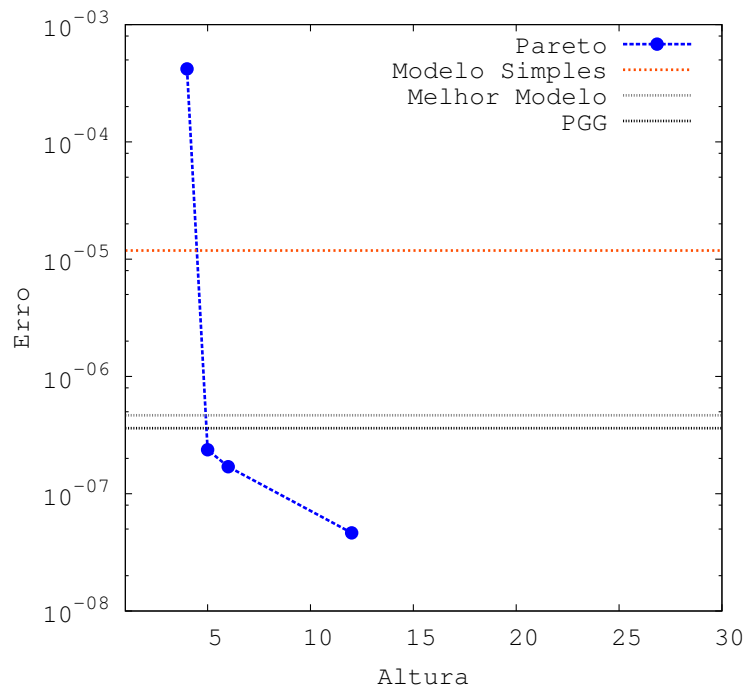


Figura 9: PGGM com Gramática Simples.

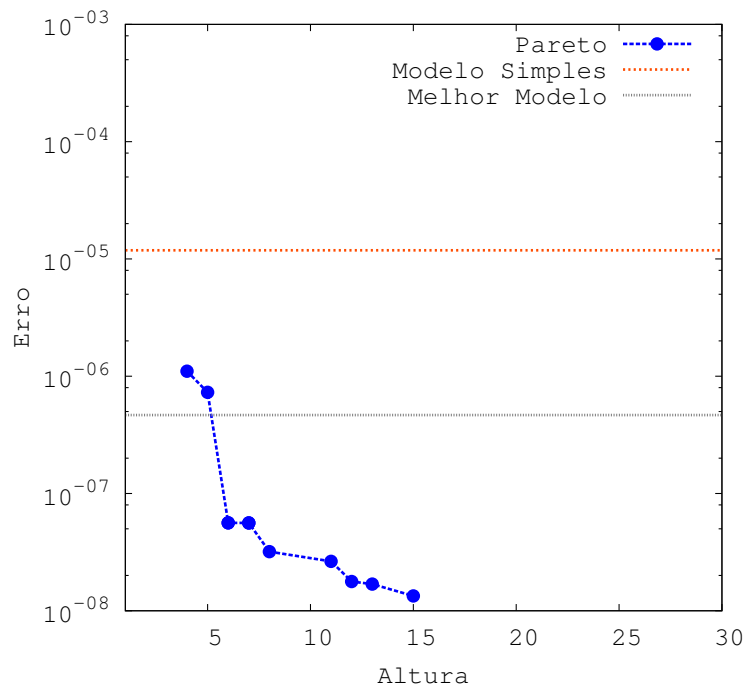


Figura 10: PGGM com Gramática Simples com MQ.

x_4 e x_7 foram combinadas na solução mais acuradas de várias execuções independentes, reforçando a informação levantada na literatura (Bernardino et al., 2011; de Freitas et al., 2014).

A Tabela 3 mostra a comparação entre os erros em relação aos dados de teste dos

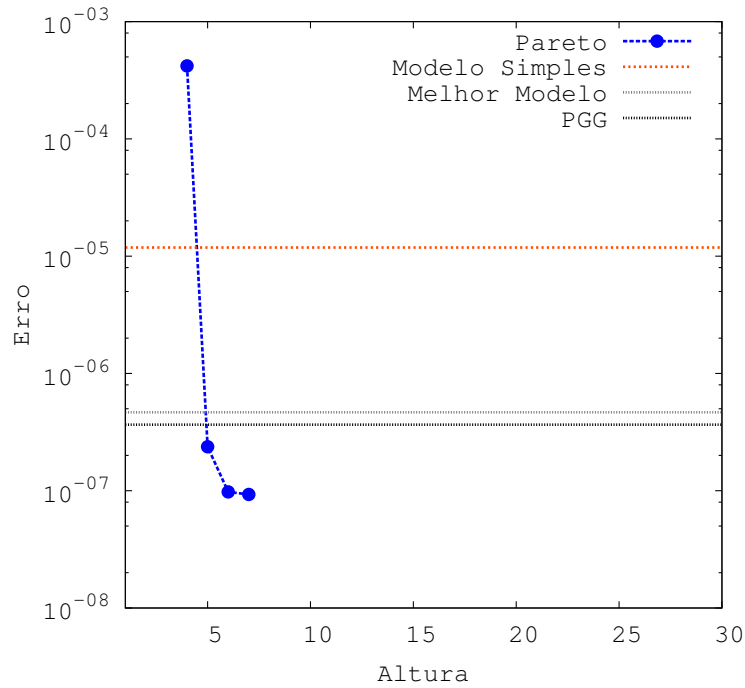


Figura 11: PGGM com Gramática Restrita.

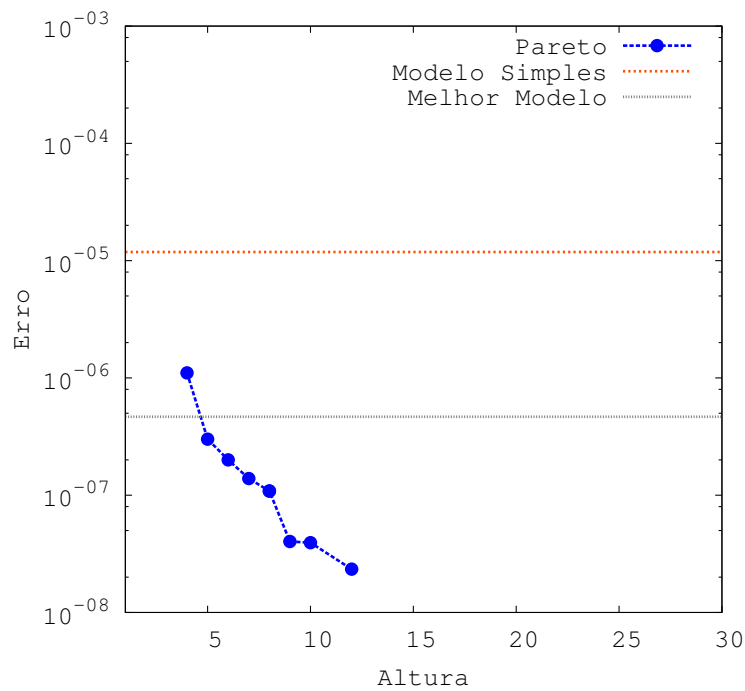


Figura 12: PGGM com Gramática Restrita com MQ.

modelos. Os resultados encontrados por de Freitas et al. (2014) com o algoritmo PGG e resultados encontrados por Bernardino et al. (2011) através do algoritmo PIG também estão incluídos. Em relação aos erros sobre os dados de teste, pode-se observar que os modelos obtidos pela PGGM (i) são melhores do que os obtidos pela PGG, com Gramática Simples e Constrita sem MQ; (ii) são melhores do que os alcançados pela PIG, com

Tabela 1: Modelos obtidos sem utilizar MQ.

Altura	Simple	Constrita
4	x_1	x_1
5	x_7^3	x_7^3
6	—	$\frac{x_7^{\sqrt{3}}}{x_4}$
8	$\frac{x_3 x_7 (\log(7) + \log(\sqrt{x_7}))}{x_4}$	—

Tabela 2: Modelos obtidos ao utilizar MQ.

Altura	Simple + MQ	Constrita + MQ
4	$0,003x_7$	$0,003x_7$
5	$118,207x_7^5$	$118,207x_7^5$
6	$0,63x_3^{x_4} + 1,05x_7^3$	$\left(\frac{30,2x_3}{x_4}\right)^{\exp(1)}$
7	$4,24\sqrt{x_4}x_3^{x_4} - 0,25x_7^{x_4} + 0,0002x_4 - 4,25$	$\frac{10,85x_3^2}{\exp(x_4)} + 0,006x_3x_4 - 0,3x_7^2$

Gramática Simple e Constrita sem MQ; (iii) são melhores do que encontrados pela PIG, utilizando MQ; e (iv) com MQ são em média piores do que aqueles obtidos sem MQ, quando uma gramática constrita é utilizada. Além disso, a variante PGGM⁽³⁾ (PGGM com Gramática Simple e Mínimos Quadrados) é a que obteve melhores modelos em relação a todas as métricas analisadas aqui. Uma análise estatística foi realizada utilizando o teste não paramétrico de Wilcoxon, indicando que as soluções geradas pela PGGM⁽³⁾ são estatisticamente diferentes do que aquelas obtidas pelas outras variantes da PGGM (p -valor $< 0,05$). Apesar de não ser objetivo do presente estudo, observou-se entretanto que o custo computacional da PGGM é maior do que o da PGG por utilizar o Método dos Mínimos Quadrados e pelas operações adicionais do tratamento dos múltiplos objetivos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Uma técnica de Programação Genética Gramatical Multiobjetivo (PPGM) foi utilizada aqui para gerar modelos de alta acurácia e baixa complexidade para inferir a máxima deformação longitudinal de dutos com amassamento. A PGGM se mostrou capaz de encontrar modelos (i) mais simples e (ii) mais acurados do que os disponíveis na literatura. Além disso, através dos resultados encontrados aqui, foram confirmadas as informações sobre a importância de algumas variáveis para o modelo, assim as combinações dessas variáveis afetam sua resposta.

Uma dificuldade encontrada aqui e em alguns trabalhos que utilizam algoritmos evolutivos é a preservação da diversidade da população. Foi observado nos experimentos computacionais um crescimento do número soluções candidatas genotipicamente diferentes (árvore) mas similares na forma do modelo (fenótipo). Assim, pretende-se estudar

Tabela 3: Resultados obtidos em relação aos dados de teste para $PGG^{(1)}$, $PIG^{(1)}$ e $PGGM^{(1)}$ com Gramática Simples, $PGG^{(2)}$, $PIG^{(2)}$ e $PGGM^{(2)}$ com Gramática Constrita, $PGGM^{(3)}$ com Gramática Simples e $PGGM^{(4)}$ com Gramática Constrita e Mínimos Quadrados e $PIG^{(3)}$ com o melhor resultado com Mínimos Quadrados.

Algoritmo	Menor	Maior	Média	Mediana	Desvio Padrão
$PGG^{(1)}$	$3,62E - 7$	$5,77E - 6$	$1,24E - 6$	$1,05E - 6$	$7,22E - 7$
$PIG^{(1)}$	$6,59E - 7$	$1,54E - 6$	$1,24E - 6$	$1,32E - 6$	$2,70E - 7$
$PGGM^{(1)}$	$4,63E - 7$	$7,71E - 7$	$2,71E - 7$	$2,37E - 7$	$1,34E - 7$
$PGG^{(2)}$	$3,66E - 7$	$8,25E - 6$	$2,03E - 6$	$1,76E - 6$	$1,27E - 6$
$PIG^{(2)}$	$5,80E - 7$	$1,89E - 6$	$1,13E - 6$	$1,08E - 6$	$3,24E - 7$
$PGGM^{(2)}$	$9,29E - 8$	$2,37E - 7$	$1,95E - 7$	$2,10E - 7$	$4,62E - 8$
$PIG^{(3)}$	$1,86E - 7$	$2,60E - 7$	$2,21E - 7$	$2,20E - 7$	$1,49E - 8$
$PGGM^{(3)}$	$1,33E - 8$	$6,03E - 8$	$3,56E - 8$	$3,59E - 8$	$1,17E - 8$
$PGGM^{(4)}$	$2,33E - 8$	$9,92E - 5$	$3,42E - 6$	$1,03E - 7$	$1,80E - 5$

alternativas de manutenção da diversidade da população neste tipo de ambiente. Além disso, pretende-se avaliar as técnicas consideradas aqui em situações mais reais, tal como adotando apenas variáveis de entrada dos modelos que possam ser coletadas em tempo real.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro do CNPq (processo 310778/2013-1) e da FAPEMIG (PCE-01337-15).

REFERÊNCIAS

- ASME B31.8. Gas transmission and distribution piping systems, 2007.
- D. A. Augusto. Co-evolução amostra-classificador integrada à programação genética gramatical para classificação de dados. Master's thesis, Universidade Federal do Rio de Janeiro, COPPE, 2004.
- H.S. Bernardino, E.S. Castro, J.N. Guerreiro, and H.J.C. Barbosa. Inferring strains on a locally deformed pipe via grammar-based immune programming. In *Proc. of the Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE)*, 2011.
- S. Bleuler, M. Brack, L. Thiele, and E. Zitzler. Multiobjective genetic programming: reducing bloat using spea2. In *Proc. of the Congress on Evolutionary Computation (CEC)*, volume 1, pages 536–543, 2001.

- J.M. de Freitas, H.S. Bernardino, J.N. Guerreiro, and H.J.C. Barbosa. Aplicação de uma programação genética gramatical na inferência da máxima deformação longitudinal de dutos com amassamento. In *Proc. of the Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE)*, 2014.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp.*, 6(2):182–197, 2002.
- J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- J. R Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.
- Anh Nguyen, Tommaso Urli, and Markus Wagner. Single- and multi-objective genetic programming: New bounds for weighted order and majority. In *Proc. of the Workshop on Foundations of Genetic Algorithms XII, FOGA XII '13*, pages 161–172. ACM, 2013.
- D.B. Noronha, R.R. Martins, B.P. Jacob, , and E. Souza. Some remarks on the strain based assessment of pipeline dents. In *Proc. of the Intl. Pipeline Conference*, 2008.
- Vilfredo Pareto. *Cours D'Economie Politique*. F. Rouge, Lausanne, Switzerland, 1896.
- Huimin Zhao. A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, 43(3):809 – 826, 2007.