Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Estatística

Robson Ortz O. Cunha

MODELO DE REGRESSÃO POR PROCESSOS GAUSSIANOS APLICADO A PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL VIA METAHEURÍSTICAS

Juiz de Fora

2018

Robson Ortz O. Cunha

MODELO DE REGRESSÃO POR PROCESSOS GAUSSIANOS APLICADO A PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL VIA METAHEURÍSTICAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade Federal de Juiz de Fora(UFJF), como parte da obtenção do grau de bacharel em Estatística.

Orientador: Heder S. Bernardino

Coorientador: Victor S. A. Menezes

Juiz de Fora

2018

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

Ortz O. Cunha, Robson.
MODELO DE REGRESSÃO POR PROCESSOS GAUSSIANOS APLICADO A PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL VIA METAHEURÍSTICAS / Robson Ortz O. Cunha. – 2018.
57 f. : il.

Orientador: Heder S. Bernardino
Coorientador: Victor S. A. Menezes
Monografia – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Departamento de Estatística, 2018.

1. Processos Gaussianos. 2. Metamodelos. 3. Metaheurísticas. 4. Evolução Diferencial.

Robson Ortz O. Cunha

MODELO DE REGRESSÃO POR PROCESSOS GAUSSIANOS APLICADO A PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL VIA METAHEURÍSTICAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade Federal de Juiz de Fora(UFJF), como parte da obtenção do grau de bacharel em Estatística.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Heder S. Bernardino - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Victor S. A. Menezes - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Camila Borelli Zeller
Universidade Federal de Juiz de Fora

Prof. Dr. Luciana Brugiolo Gonçalves
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Aos pilares deste trabalho, meus orientadores, Heder Soares Bernardino e Victor Ströele de Andrade Menezes, por toda dedicação e generosidade ao compartilharem seus conhecimentos e por toda motivação dada mesmo diante de minhas frustrações.

Ao meu tio, Antônio Roberto de Oliveira, pelo apoio e sobretudo por me mostrar, desde muito cedo, o valor da educação em nossas vidas e como ela pode nos fortificar como seres humanos. Por similar orientação em nossas conversas, agradeço ao professor Lupércio Bessegato que sempre acreditou em meu potencial.

Agradeço ainda aos meus amigos e colegas de graduação, em especial ao Rafael Rocha Garcia pelo suporte e companheirismo nesse trajeto, ao Pedro Henrique Santos Muniz e Silva, uma vez que o mesmo é parte essencial deste trabalho, compartilhando seus esforços em programação em todo processo da otimização estrutural. A Marcela Medeiros Rodrigues pelos momentos atípicos que passamos juntos e a Mariana Reis Pereira, por ser o meu melhor nesses quase cinco anos de caminhada.

Por fim, agradeço as peças mais importantes na construção do meu caráter e no meu amadurecimento, as extraordinárias mulheres que se tornaram minhas inspirações. À minha mãe, Maria Delminda de Oliveira, e à minha avó, Iracema Moreira de Oliveira, responsáveis por minha inestimável criação, me mostrando quão valiosos foram, e ainda são, seus esforços na superação de adversidades. As professoras Lucy Tiemi Takahashi e Camila Borelli Zeller por serem, de longe, as professoras mais excepcionais que tive, tornando-se exemplos por toda minha trajetória. Enfim, à todas as mulheres que conheci e admiro, por me fazerem busca uma versão melhor de mim mesmo todos os dias.

"Sic gorgiamus allos subjectatos nunc[1]."

- lema de *"The Addams Family"*

[1] em tradução livre: "Nós regozijamos no túmulo daqueles que nos subestimam."

RESUMO

A otimização estrutural é uma importante área da engenharia que vem sendo explorada, sobretudo no setor industrial, devido a sua grande gama de problemas que impactam diretamente na economia e no processo de produção como um todo. A criação de métodos que resolvam tais problemas de forma mais eficiente e/ou que produzam resultados mais fidedignos tem ganhado importância nas mais diversas áreas do conhecimento. Nesse sentido, propomos um estudo do Modelo de Regressão por Processos Gaussianos e sua utilização como metamodelo no processo de otimização estrutural via Evolução Diferencial (*Differential Evolution*, DE). A partir da construção do arcabouço teórico que promove a interdisciplinaridade entre as duas áreas (Estatística e Ciência da Computação), buscamos aplicá-lo em um problema real de engenharia, em dois diferentes casos de arranjos estruturais. Além disso, através do *software* R, usado nas análises e testes estatísticos prévios, e da linguagem de programação *Python*, usado para o desenvolvimento do processo de otimização por Evolução Diferencial assistido pelo metamodelo estatístico, apresentamos os resultados acerca dos testes e os comparamos com o método clássico da DE.

Palavras-chave: *Processos Gaussianos. Metamodelos. Metaheurísticas. Evolução Diferencial.*

ABSTRACT

The structural optimization is an important enginnering area that is being explored, specially in the industrial sector, because of its great gamma of problems that impact directly the economy and the production process as a whole. The creation of methods that solve such problems in a more effective manner and/or produce more trustworthy results has gained importance in the most diverse knowledge areas. Following this thought, we propose a Gaussian Process Regression Model study and its use as a surrogate model in the structural optimization process via Diferential Evolution (DE). From the construction of the theoretical framework that promotes the interdisciplinarity between the two areas (Statistics and Computer Science), we apply it in a real problem of enginnering, in five different cases of structural arrangements. Furthermore, through the R software, used in the previous statistical analysis and tests, and the programming language *Python*, used for the development of optimization process through Diferential Evolution assisted by the statistical surrogate model, we present the test results and compare them to the classical method of the DE.

Key-words: Gaussian Processes. Surrogate Model. Metaheuristics. Differential Evolution.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Em diversas áreas, problemas de otimização são comuns de serem enfrentados, sejam eles voltados para economia de tempo nas tomadas de decisão de um processo, ou sejam na obtenção de soluções mais precisas. No setor industrial, sobretudo no campo das engenharias, a otimização estrutural é uma área que vem sendo amplamente estudada, exigindo cada vez mais ferramentas eficientes que consigam trabalhar com sua complexidade e seu grande número de restrições.

Nesse sentido, o estudo e o desenvolvimento de métodos computacionais que contribuam para a solução de tais problemas ganham mais importância. Um dos métodos mais explorados atualmente consiste no uso de metaheurísticas que se baseiam na natureza, sobretudo em processos evolutivos, para contornar dificuldades como o cálculo da função objetivo, aquela que se deseja otimizar, quando essa apresenta mal comportamento; não-linearidade das funções de restrição, busca de gradientes caros e/ou não confiáveis, fazendo com que o processo de busca se perca em ótimos locais; dentre outros.

Em [10][14], uma das abordagens dentro do uso de metaheurísticas consiste na Evolução Diferencial (*Differential Evolution*, DE). Tratando-se de uma técnica evolutiva relativamente nova, começando a ser difundida em diversas áreas na última década, na DE uma população com indivíduos candidatos a solução ótima do problema caminha sobre o espaço de busca da função objetivo através de operações algébricas básicas. No entanto, dependendo da complexidade envolta no problema de otimização estrutural, tal técnica pode exigir um alto custo computacional para calcular os valores da função objetivo e avaliar suas restrições. Além disso, o desempenho do processo pode ser afetado pela escolha dos parâmetros das funções envolvidas.

No intuito de reduzir os exaustivos cálculos e manter os parâmetros atualizados dentro do processo, uma adaptação na DE é realizada substituindo a função objetivo por uma função de aproximação, conhecida como metamodelo, ou modelo substituto (*surrogate model*). Nesse ponto, modelos estatísticos não-paramétricos oferecem uma excelente base na metamodelagem, uma vez que podem se isentar de suposições a cerca da estrutura probabilística adjacente aos dados e por possibilitar a determinação e controle do modelo embasados pela teoria estatística.

Processos Gaussianos, segundo [6], consistem basicamente em uma das famílias de processos estocásticos observados sobre o tempo e/ou espaço de ocorrência, tal qual os dados são obtidos segundo uma distribuição normal multivariada. Assim, o estudo sobre modelos baseados em processos Gaussianos tem aparecido como uma boa escolha de metamodelo no processo de DE, como apontado em [8][24]. Em modelos de regressão, processos Gaussianos ainda ganham uma interessante abordagem sobre a perspectiva de aprendizagem de máquinas [25] facilitando a parametrização do processo como um todo.

Motivados pelo interesse interdisciplinar entre a Estatística e a Computação, além da importância na aplicabilidade dos estudos sobre os problemas de otimização estrutural, este trabalho tem por objetivo estudar e analisar a factibilidade do uso de modelos de regressão por processos Gaussianos como metamodelos no processo de otimização estrutural via Evolução Diferencial (DE). Desejamos ainda comparar os resultados do método original da DE com o proposto. Esperamos que nossa abordagem gere ganhos quanto ao valor da função objetivo, isto é, melhore o peso da estrutura analisada, e/ou melhore o tempo de processamento computacional da otimização como um todo. Também seria aceitável a melhora no cálculo da função objetivo, dado um orçamento de tempo maior no processamento computacional.

Logo, estruturamos esta monografia da seguinte forma, no Capítulo (2) partimos de uma introdução para definir a distribuição normal multivariada que serve de base para o modelo. Em seguida, tratamos da definição de processo Gaussiano, já estruturando-o sobre o modelo de regressão, partindo de uma abordagem Bayesiana. No Capítulo (3), buscamos definir o problema de otimização estrutural, o conceito de metaheurísticas e de metamodelos, além de apresentar uma abordagem por aprendizado de máquinas do modelo de regressão em questão, para enfim estruturar o método proposto pelo trabalho. Por fim, no Capítulo (4), apresentamos um rápido estudo de simulação sobre o modelo de regressão por processos Gaussianos, afim de melhor visualizar a teoria apresentada, e, mais importante, apresentamos a aplicação da metodologia num problema real de otimização estrutural sobre 5 tipos de estruturas diferentes.

2 PROCESSO GAUSSIANO

Grande parte dos modelos e metodologias estatísticas estão baseados, ou buscam trabalhar, com distribuição normal devido a sua simplicidade matemática e facilidade de tratamento. Mesmo em problemas reais, envolvendo inúmeras variáveis, a predileção pela distribuição é notável, uma vez que a mesma estende-se ao caso multivariado.

Processos Gaussianos consistem em um desses casos que busca modelar dados em inúmeras dimensões, baseando-se na distribuição normal multivariada existente sobre os mesmos. Dado nossa motivação, neste capítulo discutiremos um pouco sobre a distribuição normal multivariada, Seção (2.1), expondo suas principais propriedades, seus estimadores e apresentando um teste não-paramétrico capaz de indicar a presença da distribuição a partir de um conjunto de dados. Em seguida, Seção (2.2), trataremos do processo Gaussiano como modelo de regressão, introduzindo rapidamente conceitos de regressão linear sobre uma abordagem Bayesiana, seguido da definição do processo e suas principais propriedades. Por fim, discutiremos sobre funções Kernel, fator essencial para o ajuste do modelo, e sobre a estimação de seus hiperparâmetros.

2.1 DISTRIBUIÇÃO NORMAL MULTIVARIADA

Devido a facilidade sobre as possíveis manipulações matemáticas, a distribuição de probabilidade normal é amplamente utilizada em estatística, uma vez que podemos determiná-la conhecendo sua média e variância. A fim de generalizar seu caso para p-dimensões, nessa seção definiremos a distribuição normal multivariada, bem como suas principais propriedades e, por fim, explicitaremos os estimadores acerca de seus parâmetros.

2.1.1 **Função densidade probabilidade**

A distribuição normal multivariada consiste em uma generalização do caso univariado para $p \geq 2$ dimensões [13]. Considere X uma variável aleatória (v.a.) contínua e normalmente distribuída. Seja x uma realização dessa variável, a função densidade de probabilidade (fdp) para distrbuição normal univariada é dada por

$$f_X(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad -\infty < x < \infty, \tag{2.1}$$

onde μ e σ^2 são, respectivamente, sua média e variância.

O termo do expoente da função

$$\left(\frac{x-\mu}{\sigma}\right)^2 = (x-\mu)(\sigma^2)^{-1}(x-\mu) \tag{2.2}$$

consiste na distância quadrática entre x e μ em unidades de desvio padrão. Tal distância é conhecida na estatística como distância de Mahalanobis, utilizada para avaliação do

modelo ajustado e na detecção de observações atípicas (*outliers*). Podemos generalizar (2.2) para um vetor aleatório $\boldsymbol{X} = [X_1, X_2, \ldots, X_p]^T$ cujo vetor de observações associado é dado por $\boldsymbol{x}$

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}), \tag{2.3}$$

onde $\boldsymbol{\mu}$ é um vetor (coluna) de médias de $\boldsymbol{X}$, com tamanho p, e $\boldsymbol{\Sigma}$ é a matriz, $p \times p$, de covariância de $\boldsymbol{X}$, positiva definida, por suposição.

A distribuição normal multivariada é definida substituindo a distância univariada em (2.1) pela sua generalização em (2.3). A substituição, no entanto, exige a adequação do termo constante da fdp, uma vez que no caso multivariado as probabilidades serão representadas por volumes, segundo a superfície indicada, e não mais por área abaixo de uma curva. Dessa forma, como mostrado em [1],

$$\frac{1}{\sqrt{2\pi\sigma^2}} = \frac{1}{(2\pi)^{\frac{p}{2}}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}}.$$

Logo, a fdp para normal multivariada de um vetor aleatório $\boldsymbol{X}^T = [X_1, X_2, \ldots, X_p]$, com vetor de observações correspondente $\boldsymbol{x} = [x_1, x_2, \ldots, x_p]$, é dada por

$$f_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right\}, \tag{2.4}$$

para $-\infty < x_i < \infty$, $i = 1, 2, \ldots, p$. Denotamos (2.4) para o caso p-dimensional por $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2.1.2 Propriedades

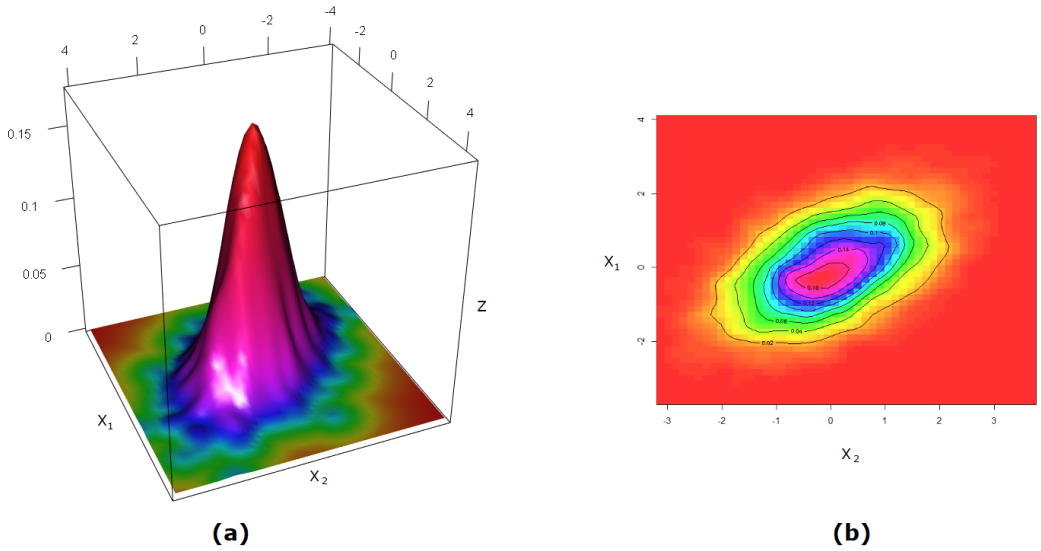
Definida a expressão (2.4), podemos citar algumas propriedades e resultados acerca da distribuição normal multivariada. Considere ainda $\boldsymbol{X}$ um vetor aleatório p-dimensional e seja $\boldsymbol{\Sigma}$ a matriz de covariância relacionada a ele, sendo esta quadrada e positiva definida, temos:

P.1. Chamamos de contorno de densidade constante de um distribuição normal p-variada todo elipsoide definido pelo vetor de observações $\boldsymbol{x}$, tal que

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) = c^2. \tag{2.5}$$

Estes elipsoides estão todos centrados na média $\boldsymbol{\mu}$ e possuem eixos $|c\sqrt{\lambda_i}\boldsymbol{\epsilon}_i|$, onde λ_i e ϵ_i denotam, respectivamente, o i-ésimo autovalor e autovetor de $\boldsymbol{X}$. A Figura (1) ilustra essa propriedade.

Figura 1 – Considere dados vindos de uma normal bivariada com $\mu_{x_1} = \mu_{x_2} = 0$ e matriz de covariância com diagonal unitária e $\sigma_{x_1,x_2} = \sigma_{x_2,x_1} = 0.1$: **(a)** representa a superfície da distribuição a cerca dos dados. **(b)** representa a projeção da superfície no eixo-X_1, X_2, evidenciando o contorno de densidade da distribuição (curvas de nível).



Fonte: Elaborado pelo autor.

P.2. Se $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, então qualquer combinação linear de seus componentes $\boldsymbol{a}^T\boldsymbol{X} = a_1X_1 + a_2X_2 + \cdots + a_pX_p$ seguem uma distribuição normal p-variada de média $\boldsymbol{a}^T\boldsymbol{\mu}$ e variância $\boldsymbol{a}^T\boldsymbol{\Sigma a}$, com recíproca verdadeira.

P.3. Se $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, então todos os possíveis subconjuntos de $\boldsymbol{X}$ também serão normalmente distribuídos.

P.4. Considere os vetores aleatórios $\boldsymbol{X_{1(q_1\times 1)}}$ e $\boldsymbol{X_{2(q_2\times 1)}}$, dizemos que ambos são independentes se:

(a) $Cov(\boldsymbol{X_1}, \boldsymbol{X_2}) = \boldsymbol{0}$ é uma matriz de zeros $q_1 \times q_2$.

(b) E somente se, $\boldsymbol{\Sigma_{12}} = \boldsymbol{\Sigma_{21}} = \boldsymbol{0}$, para

$$\begin{bmatrix} \boldsymbol{X_1} \\ \boldsymbol{X_2} \end{bmatrix} \sim N_{q_1+q_2}\left(\begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{bmatrix} \right).$$

(c) Além de independentes, X_1 e X_2 possuem distribuição normal multivariada $N_{q_1}(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_{11}})$ e $N_{q_2}(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_{22}})$, respectivamente. Então a distribuição normal multivariada conjunta entre ambos vetores aleatórios será dada por

$$N_{q_1+q_2}\left(\begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma_{22}} \end{bmatrix} \right).$$

P.5. Seja $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X_1} \\ \boldsymbol{X_2} \end{bmatrix}$ com distribuição $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, onde

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{bmatrix}.$$

Se $|\boldsymbol{\Sigma_{22}}| > 0$, então a distribuição condicional de $\boldsymbol{X_1}$, dado $\boldsymbol{X_2} = \boldsymbol{x_2}$ é normal (multivariada), com

$$\text{média: } \boldsymbol{\mu_1} + \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}}^{-1}(\boldsymbol{x_2} - \boldsymbol{\mu_2})$$

$$\text{covariância: } \boldsymbol{\Sigma_{11}} - \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}}^{-1}\boldsymbol{\Sigma_{21}}$$

P.6. Seja $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, tal que $|\boldsymbol{\Sigma}| > 0$. Então,

(a) $(\boldsymbol{X} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X} - \boldsymbol{\mu}) \sim \chi^2_p$, onde χ^2_p refere-se a a distribuição *Chi-Quadrado* com p graus de liberdade.

(b) A distribuição $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ assume probabilidade $1 - \alpha$ para um elipsoide sólido $\{\boldsymbol{x} : (\boldsymbol{X} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X} - \boldsymbol{\mu}) \leq \chi^2_p(\alpha)\}$, onde $\chi^2_p(\alpha)$ denota o (100α)-ésimo percentil superior da distribuição χ^2_p.

Demonstração. As demonstrações das propriedades anteriores podem ser encontradas em [1], para **P.1.**, e [13], **P.2.** à **P.6.**. □

A propriedade **P.6.** possibilita a interpretação da distância (2.2). Se $\boldsymbol{X}$ é normalmente distribuída (multivariada) e dado que um componente possui uma variância muito maior que os demais, então sua contribuição em (2.2) será menor. Além disso, variáveis que estão correlacionadas duas a duas também contribuirão menos para a distância quadrática, comparadas a variáveis não correlacionadas. Logo, o uso do inverso da matriz de covariância padroniza todas as variáveis envolvidas e elimina quaisquer efeitos de correlação existentes entre elas.

2.1.3 Estimadores

Seja $\boldsymbol{X}$ a matriz de uma amostra aleatória, com dimensões $n \times p$, extraída de uma população com distribuição normal multivariada, com vetor de médias $\boldsymbol{\mu}$ e matriz de covariâncias $\boldsymbol{\Sigma}$. Considerando os vetores aleatórios $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, que compõe a matriz $\boldsymbol{X}$, a distribuição conjunta para suas n observações é dada por

$$f_{\boldsymbol{X}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}} \exp\left\{-\sum_{i=1}^{n} \frac{(\boldsymbol{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})}{2}\right\}, \quad (2.6)$$

onde $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]^T$ é o vetor da i-ésima observação sobre os vetores aleatórios definidos. Diversas abordagens estatísticas buscam encontrar valores ou estimativas para os parâmetros populacionais que descrevam de forma mais fidedigna possível os dados

acerca desta população através de uma amostra, viabilizando o processo de inferência. Uma maneira de selecionar tais estimativas consiste em maximizar a densidade conjunta em uma técnica conhecida como *Estimação via Máxima Verossimilhança* [3]. Dessa forma, desejamos estabelecer uma função que maximize os valores para $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$, sobre a equação (2.6). Em [13], a função de verossimilhança para a distribuição normal multivariada é definida por

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \frac{1}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}} \exp\left\{-\frac{1}{2}tr\left[\boldsymbol{\Sigma}^{-1}\left(\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\right)\right] + \right.$$
$$\left. + \frac{1}{2}n(\overline{\boldsymbol{x}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\overline{\boldsymbol{x}} - \boldsymbol{\mu})\right\}, \tag{2.7}$$

onde $tr(\cdot)$ é denominado o traço da matriz e $\overline{\boldsymbol{x}} = (1/n)\sum_{i=1}^{n}\boldsymbol{x}_i$.

Considerando o seguinte resultado,

Resultado 1. *Seja a matriz $\boldsymbol{B}$, $p \times p$, positiva definida e considerando um escalar $b > 0$, então temos que*

$$\frac{1}{|\boldsymbol{\Sigma}|^b}\exp\left\{\frac{-tr(\boldsymbol{\Sigma}^{-1}\boldsymbol{B})}{2}\right\} \leq \frac{1}{|\boldsymbol{B}|^b}(2b)^{pb}\exp\{-pb\}$$

para toda matriz positiva definida $\boldsymbol{\Sigma}_{p\times p}$, com igualdade válida quando $\boldsymbol{\Sigma} = \frac{1}{2b}\boldsymbol{B}$.

Demonstração. Como segue em [13]. □

Dessa forma, podemos definir os estimadores de máxima verossimilhança para os parâmetros $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$ do vetor aleatório considerado, respectivamente, como

$$\begin{aligned}\hat{\boldsymbol{\mu}} &= \overline{\boldsymbol{X}} \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{X}_i - \overline{\boldsymbol{X}})(\boldsymbol{X}_i - \overline{\boldsymbol{X}})^T = \frac{(n-1)}{n}\boldsymbol{S},\end{aligned} \tag{2.8}$$

onde $\boldsymbol{S} = \frac{\sum_{i=1}^{n}\left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right)\left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right)^T}{n-1}$.

Demonstração. Os estimadores podem ser obtidos classicamente através da maximização de (2.7), utilizando o método das derivadas parciais sobre os parâmetros. Outra abordagem consiste na verificação por similaridade, sobre esta tome o núcleo do exponencial da função de verossimilhança (2.7), aplicando o conjunto de observações contidas na amostra da matriz $\boldsymbol{X}$, estruturada sobre o vetor aleatório $[X_1, X_2, \ldots, X_n]^T$, tal qual

$$tr\left[\boldsymbol{\Sigma}^{-1}\left(\sum_{j=1}^{n}(\boldsymbol{x}_j - \overline{\boldsymbol{x}})(\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T\right)\right] + n(\overline{\boldsymbol{x}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\overline{\boldsymbol{x}} - \boldsymbol{\mu}).$$

Tome ainda $\boldsymbol{\Sigma}^{-1}$ como uma matriz positiva definida, de modo que a distância $(\overline{\boldsymbol{x}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\overline{\boldsymbol{x}} - \boldsymbol{\mu}) > 0$, exceto quando $\boldsymbol{\mu} = \overline{\boldsymbol{x}}$. Dessa forma, a verossimilhança é maximizada em $\boldsymbol{\mu}$ quando $\hat{\boldsymbol{\mu}} = \overline{\boldsymbol{x}}$. Logo, nos resta maximizar

$$\mathcal{L}(\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma}|\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \frac{1}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}} \exp\left\{-\frac{tr\left[\boldsymbol{\Sigma}^{-1}\left(\sum_{j=1}^{n}(\boldsymbol{x}_j - \overline{\boldsymbol{x}})(\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T\right)\right]}{2}\right\}$$

sobre $\boldsymbol{\Sigma}$. Usando o Resultado (1) para $b = \frac{n}{2}$ e $\boldsymbol{B} = \sum_{j=1}^{n}(\boldsymbol{x}_j - \overline{\boldsymbol{x}})(\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T$, a ocorrência máxima de $\hat{\boldsymbol{\Sigma}} = \left(\frac{1}{n}\right)\sum_{j=1}^{n}(\boldsymbol{x}_j - \overline{\boldsymbol{x}})(\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T$.

Dessa forma, substituindo os vetores das observações pelo vetor aleatório levado em questão, obtemos os estimadores de máxima verossimilhança como em (2.8). □

2.1.4 **Teste de Mardia para dados normais multivariados**

Grande parte das aplicações e métodos acerca da análise multivariada e modelagem partem do princípio que os dados são provenientes de uma população normal multivariada. Se, por um lado podemos determinar o vetor de médias e a matriz de covariância de uma dada amostra, caracterizando completamente sua distribuição, por outro, esta pode ser extremamente sensível a dados discrepantes (*outliers*). Também é bem conhecido que os testes e estimativas baseados no vetor de médias da amostra e na matriz de covariância têm baixa eficiência na presença de dados ruidosos e/ou com caudas pesadas. Neste sentido, e considerando a aplicação deste trabalho, tomamos o *Teste de Mardia* para multinormalidade assintótica na tomada de decisão e modelagem dos dados.

O *Teste de Mardia* [15] baseia-se na extensão multivariada das estatística de terceira e quarta ordens, centradas na média (assimetria e curtose, respectivamente). Assim, considerando uma amostra aleatória $\boldsymbol{X}_{n\times p}$ de uma distribuição normal multivariada $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, Mardia (1970) propôs em seu trabalho coeficientes de assimetria e curtose, β_{1p} e β_{2p}, respectivamente. Ele mostrou que, a partir das propriedades de seus estimadores tais quais,

$$\begin{aligned} E[\hat{\beta_{1p}}] &= 0 & Var[\hat{\beta_{1p}}] &= \frac{6}{n} \\ E[\hat{\beta_{2p}}] &= \frac{p(p+2)(n-1)}{(n+1)} & Var[\hat{\beta_{2p}}] &= \frac{8p(p+2)}{n}, \end{aligned} \tag{2.9}$$

$\hat{\beta_{1p}}$ é um estimador não viesado para o coeficiente de assimetria da amostra e $\hat{\beta_{2p}}$ é um estimador viesado para o coeficiente de curtose da mesma.

Dessa forma, utilizando das propriedades assintóticas desses estimadores, Mardia (1970) propôs um teste de hipótese consistindo em duas partes. A primeira testa a assimetria dos dados amostrais sobre a hipótese nula primária

$$\boldsymbol{H_0'}: \beta_{1p} = 0, \text{ isto é, os dados são significativamente simétricos,}$$

com estatística de teste dada por

$$\chi_c{}^2 = \frac{n\hat{\beta_{1p}}}{6} \approx \chi_\nu{}^2, \text{ para } \nu = \frac{p(p+1)(p+2)}{6}.$$

A segunda testa a curtose dos dados sobre a hipótese nula secundária e estatística de teste dadas, respectivamente, por:

$\boldsymbol{H_0''}$: $\beta_{2p} = p(p+2)$, isto é, a distribuição é aproximadamente mesocúrtica.

$$Z_c = \frac{\hat{\beta_{2p}} - p(p+2)(n-1)(n+1)^{-1}}{[8p(p+2)n^{-1}]^{\frac{1}{2}}} \approx N(0,1).$$

Dessa forma, a um nível de significância α e assumindo o teste bilateral, rejeitamos a hipótese de que a amostra segue uma distribuição normal p-variada se ocorrer um dos casos a seguir:

(a) $\chi_c^2 \geq \chi_{\alpha,\nu}^2$ e $|Z_c| \geq Z_{\frac{\alpha}{2}}$, isto é, rejeitamos $\boldsymbol{H_0'}$ e $\boldsymbol{H_0''}$.

(b) $\chi_c^2 \geq \chi_{\alpha,\nu}^2$ ou $|Z_c| \geq Z_{\frac{\alpha}{2}}$, isto é, rejeitamos $\boldsymbol{H_0'}$ ou $\boldsymbol{H_0''}$.

Devido ao que se segue em (2.9), implicando na assintocidade do teste, o mesmo limita-se a amostras suficientemente grandes, de modo que o autor sugere amostras de tamanho mínimo entre 50 a 200 observações.

2.2 PROCESSO GAUSSIANO

Dentre as famílias de processos estocásticos mais usadas na modelagem e previsão de dados observados durante o tempo e/ou espaço de ocorrências, processos Gaussianos ganham destaque uma vez que são estruturados pela distribuição normal. Sua ampla utilização dá-se, principalmente, por duas propriedades essenciais. A primeira, podemos determinar completamente um processo Gaussiano através de suas funções de média e covariância. Essa propriedade facilita o ajuste do modelo, pois apenas os momentos de primeira e segunda ordem (média e variância, respectivamente) do processo exigem especificação. A segunda, dá-se pela simplicidade de predição, uma vez que o melhor preditor de um processo Gaussiano em um local não observado é baseado na fdp da normal multivariada, Seção 2.1.

Nesta seção vamos introduzir uma rápida abordagem em modelos de regressão linear, sob o formalismo *Bayesiano*. Em seguida buscamos definir o modelo de processos Gaussianos, juntamente as suas principais propriedades. E, finalmente, a utilização da função *Kernel* na sua modelagem.

2.2.1 O modelo de regressão via processos Gaussianos

Considere uma amostra aleatória representada pela matriz $\boldsymbol{X}_{n\times p}$. Seja $f(\cdot)$ uma função que associa $\boldsymbol{X}$ a um vetor aleatório $\boldsymbol{Y}_{n\times 1}$, cujo vetor de observações correspondente é $\boldsymbol{y}$. Assim, segundo [5], um modelo de regressão linear múltipla com erro normal é definido como

$$f(\boldsymbol{X}) = E[\boldsymbol{Y}] = \boldsymbol{X}\boldsymbol{w} \qquad \boldsymbol{Y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon}, \tag{2.10}$$

onde adotamos $\boldsymbol{w}_{p\times 1}$ como o vetor de parâmetros e $\boldsymbol{\epsilon}$ o vetor de erros aleatórios, independentes e identicamente distribuídos (iid) acerca das n observações, assumindo $\boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. Dessa forma, dado que as variáveis explicativas do modelo, que compõem $\boldsymbol{X}$, sejam independentes e sobre a suposição do vetor de erros, obtemos a função de probabilidade condicional

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) &= \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) = \prod_{i=1}^{n} \frac{1}{\sigma[2\pi]^{\frac{1}{2}}} \exp\left\{-\frac{(y_i - \boldsymbol{x}_i^T\boldsymbol{w})^2}{2\sigma^2}\right\} \\ &= \frac{1}{[2\pi\sigma^2]^{\frac{n}{2}}} \exp\left\{-\frac{1}{2\sigma^2}|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}|^2\right\} = N_n(\boldsymbol{X}\boldsymbol{w}, \sigma^2\boldsymbol{I}), \end{aligned} \tag{2.11}$$

onde $|\boldsymbol{z}|$ denota a distância euclidiana do vetor, para $\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$.

Como tratado em [25], pela ótica Bayesiana precisamos especificar uma distribuição a priori sobre o vetor de parâmetros $\boldsymbol{w}$, antes mesmo de considerarmos as covariáveis. Assim, assumimos

$$\boldsymbol{w} \sim N_p(\boldsymbol{0}, \boldsymbol{\Sigma}_w), \tag{2.12}$$

onde Σ_w é a matriz de covariâncias do parâmetros $\boldsymbol{w}$.

A inferência sobre o modelo linear Bayesiano é baseada na distribuição a posteriori dos parâmetros, obtida segundo o Teorema de Bayes.

Teorema 1. *Sejam dois eventos A e B, tal que $P(B) \neq 0$, então a probabilidade condicional de A dado B é obtida segundo a fórmula de Bayes, definida como*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

onde $P(A)$ e $P(B)$ são probabilidades a priori.

Demonstração. A prova deste teorema encontra-se em [20]. □

Dessa forma, pelo Teorema 1,

$$posteriori = \frac{condicional \times priori}{marginal} \qquad p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})} \tag{2.13}$$

cuja constante de normalização, conhecida como probabilidade marginal, é independente dos parâmetros, isto é,

$$p(\boldsymbol{y}|\boldsymbol{X}) = \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}. \tag{2.14}$$

Perceba que a posteriori em (2.13) combina a função de probabilidade condicional (2.11), a distribuição a priori (2.12) e a marginal (2.14). Assim, reescrevendo a equação (2.13) segundo as relações observadas, usando apenas o núcleo da função e completando quadrados quando necessário, obtemos

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{w} - \overline{\boldsymbol{w}})^T \boldsymbol{A}(\boldsymbol{w} - \overline{\boldsymbol{w}})\right\}. \tag{2.15}$$

Onde $\overline{\boldsymbol{w}} = \sigma^{-2}\boldsymbol{A}^{-1}\boldsymbol{X}^T\boldsymbol{y}$ e $\boldsymbol{A} = \left(\sigma^{-2}\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Sigma}_w^{-1}\right)$ são, respectivamente, o vetor médio dos parâmetros e a matriz de covariância sobre os mesmos da forma a posteriori.

Demonstração. A demonstração de (2.15) pode ser encontrada em [25]. □

A partir de (2.15), reconhecemos a forma da distribuição a posteriori (isto é, percebemos o aparecimento da distância de Mahalanobis apresentada na Seção 2.1), tal qual,

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) \sim N_p(\overline{\boldsymbol{w}}, \boldsymbol{A}^{-1}).$$

Observamos que para este modelo, e de fato para qualquer posteriori gaussiana, a média da distribuição $p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X})$ é chamada de estimativa máxima a posteriori (*maximumm a posteriori*, MAP) de $\boldsymbol{w}$.

Numa abordagem frequentista, as estimativas para os parâmetros acerca do vetor $\boldsymbol{w}$ podem ser obtidas analiticamente sobre os estimadores de máxima verossimilhança ou por métodos de controle (estimadores de mínimos quadrados, por exemplo) [5]. Ambas formas analíticas utilizam o critério das derivadas parciais sobre os parâmetros na obtenção dos estimadores, permitindo assim calcular as estimativas para $\boldsymbol{w}$. Logo, enquanto o método de estimação paramétrica Bayesiana, para o modelo de regressão, utiliza a média de vários valores obtidos pela probabilidade a posteriori, o método frequentista busca uma única estimativa para cada parâmetro.

Apesar do seu poder de interpretabilidade e predição sobre os dados ser um grande atrativo, o modelo de regressão linear possui alguns problemas quanto a sua utilização e ajuste.

O primeiro problema refere-se a sua expressividade limitada, uma vez que o ajuste do mesmo ocorre sobre uma reta. Em casos reais, isso se torna realmente problemático, já que a relação entre os dados não é necessariamente linear. O segundo problema consiste na ineficácia do uso dos métodos de estimação paramétrica discutidos quando trabalhamos com grandes bases de dados, ou com dados mal comportados (com alta variabilidade ou auto-correlacionados, por exemplo), tornando-se inviável o cálculo das derivadas matriciais bem como a inversão das mesmas. O problema analítico, por sua vez, pode ser tratado por métodos computacionais envolvendo otimização (esse tipo de tratamento será explorado na Subseção 3.2.2, após definirmos o modelo de regressão por processos Gaussianos e as implicações no processo de estimação de seus hiperparâmetros).

No intuito de manter a simplicidade e a interpretabilidade do modelo linear, uma ideia para contornar o problema de expressividade limitada consiste em projetar os dados em algum espaço com alta dimensionalidade usando um conjunto de funções bases. Sobre esse espaço, podemos aplicar o modelo de regressão em vez de diretamente nos dados em si, desde que essa projeção ocorra através de funções independentes dos parâmetros - que por sua vez devem ser lineares. Em outra palavras, ao aumentar a dimensionalidade espacial dos dados, podemos encontrar uma separação ou arranjo linear dos mesmos de modo que sua projeção não seja necessariamente linear.

Inicialmente vamos assumir uma função genérica dada $\phi(\boldsymbol{x}_i)$, para $i = 1, 2, \ldots, n$, mapeando a i-ésima observação de um espaço p-dimensional para um D-dimensional. Por convenção, tratamos de $\Phi(\boldsymbol{X})_{n\times D}$ a matriz dos dados sobre essa função. Logo, retomando $f(\boldsymbol{X})$ do modelo (2.10), temos

$$f(\boldsymbol{X}) = \Phi(\boldsymbol{X})\boldsymbol{w}, \tag{2.16}$$

onde o vetor paramétrico $\boldsymbol{w}$ agora possui comprimento D. A análise desse novo modelo será análoga ao que vimos a respeito das probabilidades a priori e a posteriori, sem perda de generalidade como discutido em [25], simplesmente substituindo $\boldsymbol{X}$ por $\Phi(\boldsymbol{X})$ e tomando o novo comprimento de $\boldsymbol{w}_{D\times 1}$ e a nova dimensão de $\boldsymbol{A}_{D\times D}$. Assim, temos

$$\overline{\boldsymbol{w}} = \boldsymbol{\Sigma}_w\Phi(\boldsymbol{X})^T(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y} \qquad \boldsymbol{A} = \boldsymbol{\Sigma}_w - \boldsymbol{\Sigma}_w\Phi(\boldsymbol{X})^T(\boldsymbol{K} - \boldsymbol{\sigma^2 I})^{-1}\Phi(\boldsymbol{X})\boldsymbol{\Sigma}_w,$$

onde $\boldsymbol{K} = \Phi(\boldsymbol{X})\boldsymbol{\Sigma}_w\Phi(\boldsymbol{X})^T$ é tida como a matriz de covariâncias dos dados mapeados na dimensão D. Detalhes sobre a equivalência dos resultados de $\overline{\boldsymbol{w}}$ e $\boldsymbol{A}$ nos casos p-dimensional e D-dimensional são discutidos em [25], fugindo do escopo desse trabalho. Além disso, na Subseção 2.2.3, trataremos mais detalhadamente sobre a matriz $\boldsymbol{K}$, chamada de *função de covariância* ou *matriz/função Kernel*.

Uma alternativa equivalente sobre a ideia de extensão do modelo de regressão linear para um espaço de maior dimensão pode ser encontrada em [25], utilizando processos Gaussianos.

Definição 1. *Um Processo Gaussiano é uma coleção finita de variáveis aleatórias, possuindo uma distribuição de probabilidade conjunta normal multivariada, sendo completamente especificado pela sua função de média e covariância. Em notação, dizemos que*

$$f_{\boldsymbol{X}}(\boldsymbol{x}_s) \sim \mathcal{GP}(m(\boldsymbol{x}_s), k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}))$$

para

$$\begin{aligned} m(\boldsymbol{x}_s) &= E[f(\boldsymbol{x}_s)] \\ k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) &= E[(f(\boldsymbol{x}_s) - m(\boldsymbol{x}_s))(f(\boldsymbol{x}_{s+h}) - m(\boldsymbol{x}_{s+h}))], \end{aligned}$$

onde $m(\cdot)$ e $k(\cdot)$ são as funções de média e covariância, respectivamente, e $\boldsymbol{x}_s$ e $\boldsymbol{x}_{s+h}$ são as observações, sobre os vetores aleatórios envolvidos, realizadas em tempos e/ou espaços diferentes.

Assim, segundo a Definição 1, podemos substituir $\phi(\boldsymbol{x})\boldsymbol{w}$ diretamente pela função $f(\cdot)$ que mapeia a observação $\boldsymbol{x}$ para esse espaço de maior dimensionalidade, desde que ela siga um processo Gaussiano com função de média $m(\cdot)$ e função de covariância (kernel) $k(\cdot, \cdot)$.

Ainda pela Definição (1) e levando em conta a estrutura do modelo de regressão linear 2.10, o modelo de regressão por Processos Gaussianos é dado por

$$\begin{aligned} y_i &= f_{\boldsymbol{X}}(\boldsymbol{x}_i) + \epsilon_i, \text{ tal que } \epsilon_i \sim N(0, \sigma^2) \\ \hat{y}_i &= f_{\boldsymbol{X}}(\boldsymbol{x}_i) \sim \mathcal{GP}(m(\boldsymbol{x}_s), k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h})), \quad \forall i = 1, 2, \ldots, n. \end{aligned} \tag{2.17}$$

Nota-se que, ainda pela Definição (1), a consistência do processo é mantida graças a propriedade **P.3.**, da Subseção 2.1.2, isto é, se um conjunto de variáveis é distribuído multinormalmente, então um subconjunto do mesmo não terá sua distribuição alterada. Não há perda de generalidade caso o processo Gaussiano seja ajustado sobre um conjunto de observações não espaciais ou temporais (caracterizado pela abordagem por aprendizado de máquinas, Capítulo 3).

2.2.2 Propriedades do modelo

Como vimos na Subseção 2.2.1, um processo Gaussiano pode ser completamente especificado apenas pela sua função de média e covariância. Dessa forma, vamos especificar algumas propriedades e/ou pressupostos que, além de caracterizar o caso tratado nesse trabalho, facilitará a determinação de tais funções.

Estacionariedade

O primeiro pressuposto considerado em nossos estudos diz respeito a estacionariedade do processo Gaussiano. Assim, tomamos as seguintes definições:

Definição 2. *Um processo estocástico $Z = \{Z(s), s \in \mathcal{S}\}$ é dito estritamente estacionário se, para todas as distribuições p-dimensionais,*

(i) $E[Z(s)] = \mu$ *é independente de* s*;*

(ii) $cov\,(Z(s), Z(s+h)) = \sigma^2$ *é independente de* s*,* $\forall h$*.*

Em outras palavras, as funções de média e covariância, bem como suas respectivas estruturas, não se alteram ao decorrer do tempo e/ou espaço.

Definição 3. *Um processo estocástico $Z = \{Z(s), s \in \mathcal{S}\}$ é considerado estacionário de segunda ordem se, e somente se,*

(i) $E[Z(s)] = \mu$ *é independente de* s*;*

(ii) $E[Z^2(s)] < \infty, \forall s$*;*

(iii) $cov[Z(s), Z(s+h)]$ *é função apenas de* h*.*

Isto é, a função de média é constante ao longo do tempo e/ou espaço, porém a função de covariância depende apenas do espaçamento temporal e/ou espacial dos dados.

Considerando um processo Gaussiano, tal qual na Definição (1), diremos que o mesmo é completamente especificado pelas suas funções de média e covariância e, se ele possuir estacionariedade de segunda ordem, então ele será estritamente estacionário (ou simplesmente estacionário).

Função de Médias

Como definido anteriormente, pela característica de estacionariedade em processos Gaussianos, a função de médias é constante para qualquer variação de tempo e/ou espaço, independente do tipo de estacionariedade adotada. Além disso, sobre o caso Gaussiano, ela é comumente igual a zero, ou seja, $m(\boldsymbol{x}) = E[\boldsymbol{x}] = 0$, para qualquer vetor de observação $\boldsymbol{x}$ contido na matriz aleatória $\boldsymbol{X}$ de dados.

Função de Covariância

Dado um processo Gaussiano estacionário com média zero, acerca da sua função de covariância $k(\cdot)$, tomamos as seguintes propriedades como em [6]:

P'.1. $k(\boldsymbol{x}, \boldsymbol{x}) \geq 0$;

P'.2. $k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) = k(h) = k(-h)$, isto é, $k(\cdot)$ é uma função par;

P'.3. $|k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h})| \leq k(\boldsymbol{x}, \boldsymbol{x})$;

P'.4. Dizemos que a função de covariância $k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h})$ é não-negativa definida, tal que

$$\sum_{i,j=1}^{n} a_i a_j k(s_i - s_j) \geq 0,$$

onde $a_i \in \mathbb{R}$, para $i, j = 1, 2, \ldots, n$, e $s \in \mathcal{S}$.

Demonstração. Seja um processo Gaussiano estacionário, com média zero, e seja $k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) = E[\boldsymbol{x}_s \boldsymbol{x}_{s+h}]$ a função de covariância associado a ele.

P'.1. e **P'.2.** decorrem imediatamente da definição de covariância.

P'.3. Basta notarmos que,

$$\begin{aligned} E[\boldsymbol{x}_{s+h} \pm \boldsymbol{x}_s]^2 &\geq 0, \\ E[\boldsymbol{x}_{s+h} \pm \boldsymbol{x}_s] E[\boldsymbol{x}_{s+h} \pm \boldsymbol{x}_s] &\geq 0, \\ E[\boldsymbol{x}_{s+h}]^2 \pm 2E[\boldsymbol{x}_{s+h} \boldsymbol{x}_s] + E[\boldsymbol{x}_s]^2 &\geq 0, \\ k(\boldsymbol{x}_{s+h}, \boldsymbol{x}_{s+h}) \pm 2k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) + k(\boldsymbol{x}_s, \boldsymbol{x}_s) &\geq 0. \end{aligned}$$

Como $k(\boldsymbol{x}_{s+h}, \boldsymbol{x}_{s+h}) = k(\boldsymbol{x}_s, \boldsymbol{x}_s) = k(\boldsymbol{x}, \boldsymbol{x}) = \sigma^2$ pela propriedade de estacionariedade, temos que

$$\begin{aligned} 2k(\boldsymbol{x}, \boldsymbol{x}) \pm 2k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) &\geq 0 \\ |k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h})| &\leq k(\boldsymbol{x}, \boldsymbol{x}). \end{aligned}$$

P'.4.

$$\sum_{i,j=1}^{n} a_i a_j k(s_i - s_j) = \sum_{i,j=1}^{n} a_i a_j E[\boldsymbol{x}_{s_i} \boldsymbol{x}_{s_j}] = E[\sum_{j=1}^{n} a_j \boldsymbol{x}_{s_j}]^2 \geq 0$$

□

2.2.3 A função Kernel

Em computação, sobretudo na área de aprendizado de máquinas, e na estatística não-paramétrica as funções Kernel são bastante utilizadas sempre que é necessário definir um espaço de características de alta dimensionalidade implícito, no qual a máquina/modelo de aprendizado (linear) opera. Segundo [9], um kernel, também chamado de função de covariância, é uma função positiva definida de duas entradas/amostras de dados, $\boldsymbol{x}_s$ e $\boldsymbol{x}_{s+h}$ por exemplo.

Em processos Gaussianos modelos kernel são utilizados para definir a covariância a priori entre duas funções/variáveis aleatórias do processo em questão

$$cov[f_{\boldsymbol{X}}(\boldsymbol{x}_s), f_{\boldsymbol{X}}(\boldsymbol{x}_{s+h})] = k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}),$$

isto é, o kernel especifica a priori quais funções estão provavelmente sob o processo Gaussiano, o que por sua vez determina as propriedades de generalização do modelo.

Ainda sobre [9], são apresentadas as principais funções kernel utilizadas no ajuste de modelos de predição por processos Gaussianos, como podem ser vistas na Tabela (1). Considerando a aplicação em [24] e dada popularidade e simplicidade da função do tipo exponencial quadrática, também denotada como função de base radial (RBF), essa foi adotada para a aplicação e ajuste do modelo de regressão nesse trabalho.

Tabela 1: Principais tipos de função kernel em processo Gaussiano.

Tipo de Kernel	$k(h = \|\|\boldsymbol{x}_{s+h} - \boldsymbol{x}_s\|\|)$
RBF	$\sigma^2 \exp\left\{\frac{-h^2}{2l^2}\right\}$
Exponencial	$\exp\left\{\frac{-\|h\|}{l}\right\}$
Periodic	$\sigma^2 \exp\left\{-\frac{1}{l^2}\sin^2\left(\frac{\pi h}{p}\right)\right\}$
Matérn ($\eta = \frac{5}{2}$)	$\left(1 + \frac{\sqrt{5}\|h\|}{l} + \frac{5\|h\|}{3l^2}\right)\exp\left\{\frac{\sqrt{5}\|h\|}{l}\right\}$
Matérn ($\eta = \frac{3}{2}$)	$\left(1 + \frac{\sqrt{3}\|h\|}{l}\right)\exp\left\{\frac{\sqrt{3}\|h\|}{l}\right\}$

Cada kernel possui um número de parâmetros que especificam a forma precisa da função de covariância. Por vezes, são referidos como hiperparâmetros, uma vez que podem ser visualizados como forma de especificar uma distribuição sobre parâmetros de função, em vez de serem parâmetros que especificam uma função diretamente. No caso, podemos entender o kernel RBF

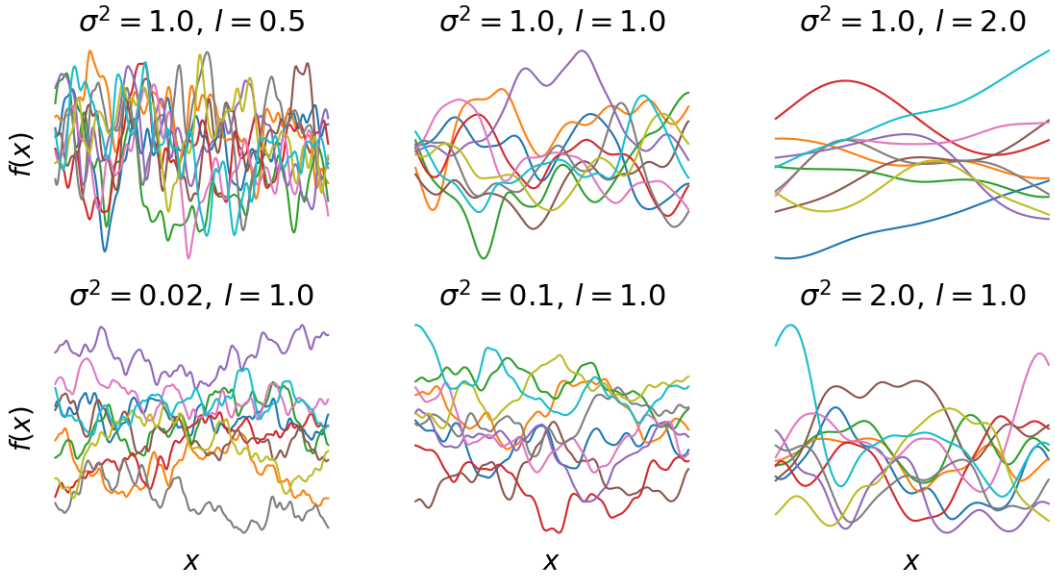
$$k(\boldsymbol{x}_s, \boldsymbol{x}_{s+h}) = \sigma^2 \exp\left\{\frac{-||\boldsymbol{x}_s - \boldsymbol{x}_{s+h}||^2}{2l^2}\right\}, \tag{2.18}$$

como uma função muito semelhante a Gaussiana, excluindo o fator de normalização, onde $\boldsymbol{x}_s$ é o ponto de máximo e distância aumenta conforme nos afastamos desse ponto. O hiperparâmetro l tem papel parecido ao da variância da Gaussiana, controlando a largura da curva. Assim, valores muito pequenos para l nos diz que a distância aumenta rapidamente quando nos afastamos de $\boldsymbol{x}_s$ fazendo com que função varie muito rapidamente, sendo pouco suave. Em contra partida, valores elevados de l introduz uma noção de distância que diminui lentamente conforme nos afastamos de $\boldsymbol{x}_s$ suavizando a função. Por sua vez, o hiperparâmetro σ^2 nos diz o quanto esperamos que nossa distância seja algo fora da média.

A Figura 2 representa o que ocorre graficamente quando se varia os hiperparâmetros do kernel. Percebemos que, ao deixarmos σ^2 fixo e variamos l (sequência gráfica horizontal

superior), a variação ocorre pela frequência das ondas nos modelos ajustados (suavidade). Em contrapartida, quando fixamos l e variamos σ^2 (sequência gráfica horizontal inferior) a amplitude das ondas geradas pelos ajustes dos modelos sofre perturbação.

Figura 2 – Dinâmica da variação dos hiperparâmetros σ^2 e l: os três gráficos superiores representam a variação de l para um dado σ^2, enquanto os três gráficos inferiores representam a variação de σ^2 para um dado l.



Fonte: Elaborado pelo autor.

Quanto a estimação dos hiperparâmetros do kernel RBF, ainda em [24], temos uma abordagem muito similar a estimação paramétrica da distribuição normal multivariada em (2.8), pelo método de máxima verossimilhança relacionada ao processo. No entanto, a determinação de uma fórmula analítica para os estimadores é, quase sempre, inviável devido a complexidade matemática gerada na sua aplicação sobre os dados. Assim, métodos computacionais baseados em otimização via gradiente descendente, nos permite obter estimativas para os hiperparâmetros σ, l de forma autoadaptativa do método da máxima verossimilhança.

Visto a necessidade de um método otimizador para obtenção das estimativas dos hiperparâmetros, adotamos nesse trabalho o algoritmo `L-BFGS`, pertencente a família dos métodos *quasi-Newton*, muito popular na estimação de hiperparâmetros em aprendizado de máquinas, além de ser uma ferramente disponível em diversas funções voltadas a processos Gaussianos em *Python*[19]. Detalhes sobre o algoritmo são tratados na Subseção 3.2.2.

3 PROCESSO DE OTIMIZAÇÃO ESTRUTURAL ASSISTIDO POR METAMODELOS

Nas engenharias, problemas envolvendo análise e otimização estrutural são recorrentes, refletindo diretamente a necessidade do setor industrial em criar ferramentas computacionais eficazes para lidar com os desafios que tais problemas oferecem. Nesse sentido, metaheurísticas podem ajudar a superar as dificuldades apresentadas pela baixa regularidade das funções objetivo, um grande número de restrições implícitas não-lineares e gradientes caros e/ou não confiáveis.

Neste capítulo, discutiremos, na Seção 3.1, os problemas acerca da otimização estrutural, definindo o conceito de metaheurística e explorando um de seus casos particulares: a Evolução Diferencial (*Differential Evolution*, DE). Em seguida, na Seção 3.2, trataremos do estudo de metamodelos, bem como a incorporação do modelo de regressão por processos Gaussianos, vistos no capítulo anterior. Por fim, na Seção 3.3, agruparemos as metodologias vistas até então para definir rapidamente a estrutura do processo de otimização via DE assistida por metamodelos.

3.1 PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL

O desenvolvimento de técnicas eficazes no tratamento de problemas de otimização estrutural tem ganhado cada vez mais atenção. Segundo [14], esse tipo de problema envolve basicamente três fatores de relevância: dimensionamento, forma e topologia.

Nos problemas relacionados a otimização de dimensionamento estrutural, considerados neste trabalho, as variáveis do modelo medem características mecânicas a respeito da estrutura considerada - em nosso caso, como pode ser visto na Seção 4.2, essa característica mecânica consiste na área transversal das barras de uma treliça em questão. Dessa forma, definimos um modelo geral para o problema de otimização estrutural como,

$$\begin{aligned} min. \quad & \mathcal{H}(\boldsymbol{x}) \\ \text{sujeito a} \quad & g_i(\boldsymbol{x}) \leq 0, \forall i = 1, 2, \ldots, m \end{aligned} \tag{3.1}$$

onde $\mathcal{H}(\cdot)$ é chamada de função objetivo, $g_i(\cdot)$ é a i-ésima restrição acerca das violações do problema, m é o número de restrições e $\boldsymbol{x}$ representa uma solução candidata.

Em otimização, há interesse no desenvolvimento de métodos com bom desempenho computacional. Entretanto, muitas técnicas requerem que a função objetivo e/ou suas restrições sejam funções explícitas das variáveis de projeto e que sejam diferenciáveis, o que não ocorre em muitos problemas práticos, como na otimização estrutural. Nesse tipo de problema é comum o uso de heurísticas que o resolva de forma genérica, já que não se conhece o algoritmo eficiente.

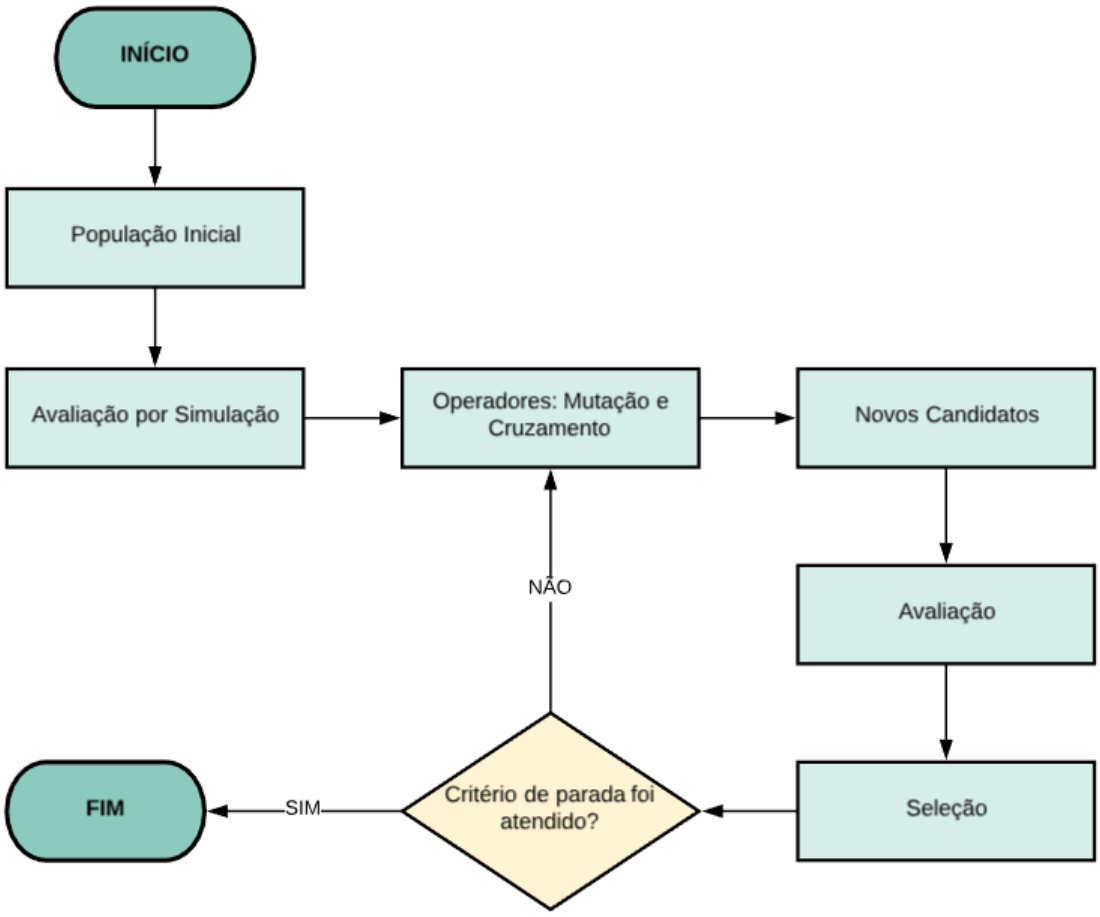
Apesar disso, essas heurísticas podem requerer muitas chamadas à função objetivo e as restrições para se obter uma boa solução, tornando, em muitos casos, proibitivo o uso desse tipo de técnica. O uso de modelos aproximados para substituir o cálculo de funções objetivo e restrições com alto custo computacional torna-se assim importante.

3.1.1 **Evolução Diferencial**

Definidas como um conjunto de processos cognitivos empregados em tomadas de decisão que ignoram parte da informação acerca do problema, afim de torná-lo mais fácil e rápido; metaheurísticas consistem em métodos heurísticos normalmente empregadas na área de otimização, aplicadas em casos onde não existe um algoritmo eficiente factível. Dentre os problemas abrangidos em otimização, as metaheurísticas coordenam, especialmente, procedimentos de busca capazes de criar um processo geral para escapar de mínimos locais e realizar uma busca robusta no espaço de soluções do problema.

Uma dessas metaheurísticas consiste na Evolução Diferencial (*Differential Evolution*, DE), popular para resolver problemas de otimização, baseando-se em métodos estocásticos. Segundo [2], DE pode ser descrito como uma manipulação de indivíduos que representam as soluções candidatas através de gerações. Isto é, considerando sucessivas gerações de um processo de DE, onde em cada uma, soluções candidatas sofrem modificações de mutação e cruzamento, gerando novas soluções. Em seguida, realiza-se uma seleção dentre essas novas candidatas, passando-as para a próxima geração, repetindo o ciclo. A Figura 3 apresenta um fluxograma simplificado que ilustra o processo de DE.

Figura 3 – Fluxograma simplificado do processo de Evolução Diferencial.



Fonte: Elaborado pelo autor.

Existem três operações básicas indicadas no processo e aplicadas sobre os indivíduos candidatos à solução ótima:

(i) **Mutação** - o operador de mutação promove uma modificação em cada indivíduo pela adição da diferença vetorial ponderada entre dois indivíduos aleatórios a um terceiro candidato (base), todos pertencentes a população inicial, finalmente gerando vetores modificados. Em notação,

$$\boldsymbol{x}_{mutação} = \boldsymbol{x}_{base} + F(\boldsymbol{x}_1 - \boldsymbol{x}_2),$$

onde $\boldsymbol{x}_{mutação}$ é o novo indivíduo gerado, F é a ponderação associada a diferença vetorial entre dois indivíduos aleatórios e $\boldsymbol{x}_{base}$ é o candidato onde será realizada a perturbação.

(ii) **Cruzamento** - após a operação de mutação, introduz-se o operador de cruzamento afim de aumentar a variabilidade dos indivíduos relacionados, promovendo troca de atributos entre os vetores mutantes e membros da população não modificados. Além

disso, introduzimos uma constante arbitrária `k` (não confundir a constante `k` com a função kernel $k(\cdot,\cdot)$) uniformemente distribuída por toda dimensão do problema, afim de garantir que o pai seja diferente de seu descendente. Definimos ainda que o novo vetor experimental é formado, tal que,

$$u_j = \begin{cases} x_{j_{mutação}}, & r \leq CR. \text{ ou } j = \texttt{k} \\ x_{j_{base}}, & c.c. \end{cases}$$

onde r é um número gerado aleatoriamente, CR é dito operador de cruzamento, sendo um valor real determinado dentro de um intervalo informado pelo usuário, $x_{j_{base}}$ é a j-ésima componente do indivíduo alvo da população, que competirá com o novo indivíduo gerado.

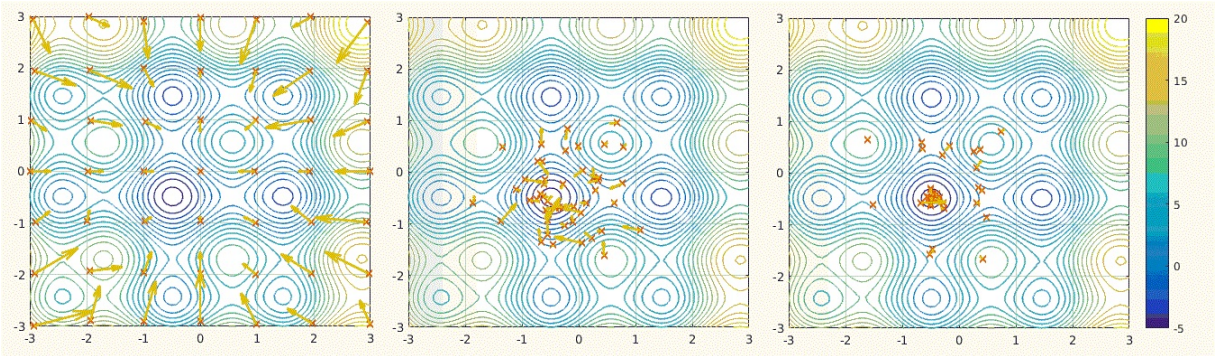
(iii) **Seleção** - o operador de seleção busca selecionar os indivíduos com melhores atributos após a realização de uma avaliação, sendo estes mantidos para a sucessiva geração. Para isso utilizamos o seguinte critério de escolha,

$$\boldsymbol{x}_{base} = \begin{cases} \boldsymbol{u}, & \mathcal{H}(\boldsymbol{u}) \leq \mathcal{H}(\boldsymbol{x}) \\ \boldsymbol{x}_{base}, & c.c. \end{cases}$$

onde $\boldsymbol{x}_{base}$ denota o candidato atual e $\boldsymbol{u}$ o candidato que sofreu cruzamento de atributos.

A nível de ilustração, a Figura 4 trás uma representação do que ocorre na DE quando são aplicadas as três operações básicas citadas. Basicamente elas alteram a direção e sentido do vetor gradiente dos pontos, de modo a sempre conduzir os candidatos a solução ótima ao ponto ótimo desejado.

Figura 4 – Ilustração das operações básicas em um processo de DE de minimização: os vetores gradientes gerados tem sua direção e sentido modificados toda vez que aplicamos as operações de mutação, cruzamento e seleção nos candidatos, buscando convergi-los a um ponto ótimo.



Fonte: Wikipidia (adaptado)

Como já citado, o que foi apresentado na Figura 3 trata-se da forma mais básica do processo de DE, uma vez que o mesmo apresenta variações sobre a aplicação das

diferenças, a seleção e a distribuição de recombinação. Na literatura, e como apresentado em [14], podemos citar quatro principais variações do processo (escritas sobre a forma *DE/mecanismo-de-seleção/número-de-diferenças/modelo-de-recombinação*) baseadas apenas no mecanismo de seleção, sendo elas:

(1) *DE/rand/1/bin* - trata-se da forma básica de DE que seleciona aleatoriamente todos os indivíduos envolvidos ao longo do processo, realizando a subtração da etapa de mutação por um único par de vetores, com operador de cruzamento do tipo binomial.

(2) *DE/best/1/bin* - possui estrutura similar ao caso básico, no entanto a seleção busca o melhor indivíduo da população, servindo como vetor base ($\boldsymbol{x}_{base}$) no processo de mutação, mantendo a seleção dos demais aleatória.

(3) *DE/target-to-best/1/bin* - esta variante usa o melhor indivíduo da população e um indivíduo alvo, que será usado na comparação após a mutação.

(4) *DE/target-to-rand/1/bin* - similar a variante anterior, no entanto substitui a seleção do melhor indivíduo por um aleatório.

Ainda sobre o funcionamento da DE, neste trabalho, vamos considerar apenas casos onde as variáveis de projeto são contínuas. Dessa forma, são especificados previamente os limites inferior e superior, tal que se um determinado componente x_j, para $j = 1, 2, \ldots, p$, de uma solução candidata $\boldsymbol{x}$ é gerado fora de sua faixa prescrita, uma operação padrão de projeção é executada, re-selecionando-o até que esteja entre seus limitantes. Essa operação segue de [14],

$$\begin{cases} x_j = x_j^{sup}, \text{ se } x_j > x_j^{sup}, \\ x_j = x_j^{inf}, \text{ se } x_j < x_j^{inf}, \end{cases} \tag{3.2}$$

onde x_j^{sup} e x_j^{inf} são os limitantes superior e inferior fixados para a j-ésima componente, respectivamente.

Em muitos casos, inclusive o que discutiremos na Seção 4.2, o processo de otimização por DE pode se tornar preocupante quanto ao tempo do processamento computacional, pois o mesmo pode envolver simulações computacionalmente custosas (tal como o método dos elementos finitos) na avaliação (função objetivo e/ou restrições) dos indivíduos em cada geração. Além disso, ainda na Seção 4.2 explicitaremos as funções aqui tratadas especificamente para o nosso problema de otimização estrutural.

3.1.2 **Tratamento de Restrições**

Muitos problemas de busca e otimização no mundo real envolvem restrições de desigualdade e/ou igualdade sendo colocados como problemas de otimização restritos. No

sentido de tratar tais restrições uma técnica muito popular que prioriza indivíduos factíveis no problema como critério de seleção é conhecida como Método Deb [7].

O Método Deb consiste inicialmente na contabilização de todas as restrições envolvidas no problema sobre um indivíduo/candidato, isto é

$$c_{sum}(\boldsymbol{x}) = \sum_{i=1}^{m} \max[0, g_i(\boldsymbol{x})], \tag{3.3}$$

onde $g(\cdot)_i$ é a i-*ésima* restrição de desigualdade e $\boldsymbol{x}$ o indivíduo analisado.

Em seguida, um processo de seleção do candidato é realizado sobre c_{sum}, visando a seleção do melhor indivíduo para resolver o problema de otimização (no geral, o indivíduo escolhido é o que apresenta menor valor de c_{sum}). Um escopo do processo pode ser visualizado pelo pseudocódigo do Algoritmo 1.

Algorithm 1 Método Deb

Entrada candidatas a solução ($\boldsymbol{x}_1$ e $\boldsymbol{x}_2$, por exemplo)

início
  se $c_{sum}(\boldsymbol{x}_1) < c_{sum}(\boldsymbol{x}_2)$ **então**
    selecione $\boldsymbol{x}_1$
  fim se
  se $c_{sum}(\boldsymbol{x}_1) > c_{sum}(\boldsymbol{x}_2)$ **então**
    selecione $\boldsymbol{x}_2$
  fim se
  se $c_{sum}(\boldsymbol{x}_1) = c_{sum}(\boldsymbol{x}_2)$ **então**
    se $\mathcal{H}(\boldsymbol{x}_1) < \mathcal{H}(\boldsymbol{x}_2)$ **então**
      selecione $\boldsymbol{x}_1$
    fim se
    se $\mathcal{H}(\boldsymbol{x}_1) > \mathcal{H}(\boldsymbol{x}_2)$ **então**
      selecione $\boldsymbol{x}_2$
    fim se
  fim se
fim

Saída melhor candidato entre $\boldsymbol{x}_1$ e $\boldsymbol{x}_2$

3.2 METAMODELOS

Os metamodelos [10], ou modelos substitutos, são conhecidos como aproximações da função de avaliação/objetivo em um processo de otimização via metaheurísticas sendo significativamente mais simples e mais baratos que esta. Esse termo apareceu pela primeira vez em 1970, a partir dos estudos realizados pelo físico-computacional Robert W. Blanning, e suas primeiras aplicações buscavam auxiliar o cálculo de sensibilidade de modelos de simulação, onde era preciso executá-lo diversas vezes. Atualmente, além desta, é comum utilizar metamodelos na substituição direta de simulações que demandam

recursos computacionais em excesso e na melhora do desempenho de algoritmos iterativos de otimização, mantendo fixo o custo computacional ligado ao processo.

Ainda segundo [10], de maneira formal, um metamodelo pode ser definido como:

- uma simplificação de um modelo de simulação, construído sobre hipóteses físicas rígidas ou hipóteses numéricas mais flexíveis;
- uma construção feita segundo um número limitados de elementos resultante de algum experimento, provindas de um domínio/amostra de dados $\mathcal{D} = \{(\boldsymbol{x}_i, \mathcal{H}(\boldsymbol{x}_i))\}$, para $i = 1, \ldots, n$.

Em notação,

$$\mathcal{H} \approx \hat{\mathcal{H}} = \hat{\mathcal{H}}(\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{x}_i), \tag{3.4}$$

onde $\mathcal{H}$ denota o modelo original (função objetivo), $\hat{\mathcal{H}}$ o seu modelo aproximado (substituto), $\boldsymbol{\theta}$ é o vetor de parâmetros do modelo associado e $\boldsymbol{x}_i$ o vetor de observação indexado.

Perceba que o ajuste do metamodelo só ocorre quando obtemos uma amostra $\mathcal{D}$, como definida, uma vez que nela conhecemos os valores das entradas (variáveis explicativas do modelo) e saída (variáveis respostas) dos dados. Em termos computacionais, mais precisamente em aprendizado de máquinas, entendemos essa amostra como conjunto de treinamento (como veremos na próxima Seção) capaz de ajustar um metamodelo inicial para predições aproximadas.

Uma vez que o metamodelo foi ajustado e fornece uma aproximação para o modelo de simulação, o erro da aproximação realizada para etapa de avaliação de uma solução é dado por

$$E_{\hat{\mathcal{H}}}(\boldsymbol{x}_i) = ||\mathcal{H}(\boldsymbol{x}_i) - \hat{\mathcal{H}}(\mathcal{D}, \boldsymbol{\theta}, \boldsymbol{x}_i)||,$$

para $\boldsymbol{x}_i$ o i-ésimo indivíduo (estrutura) considerada, com $i = 1, 2, \ldots, n$.

Existem diversos tipos de metamodelos derivados dos modelos de simulação. Tendo em vista nossa aplicação, consideramos *metamodelos de ajuste de dados*, construídos com base em dados obtidos de simulações prévias, sendo eles gerados especificamente para construção do metamodelo ou usados nas primeiras iterações da otimização. Metamodelos desse tipo independem do modelo de simulação que irão substituir, exigindo apenas o ajuste de seus parâmetros. O procedimento de ajuste pode ser realizado seguindo um sub-problema de otimização ou por um processo de aprendizado do modelo. Além da facilidade de compreensão e aplicação, esse tipo de metamodelo pode ser construído segundo modelos baseados em processos Gaussianos com funções de base radial [18].

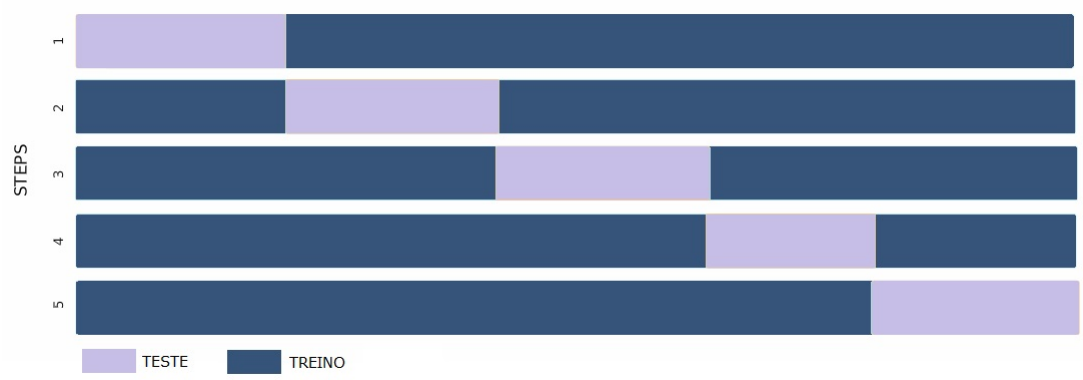
3.2.1 Regressão por Processos Gaussianos sobre a Perspectiva de Aprendizagem de Máquinas

Atualmente, *machine learning*, ou aprendizado de máquinas, tem se tornado um termo muito utilizado e de grande atratividade na ciência da computação e em áreas relacionadas. Em termos gerais, o aprendizado de máquinas evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial. Além disso, suas tarefas são classificadas em três tipos: *aprendizado supervisionado*, *aprendizado não-supervisionado* e *aprendizado por reforço* [12].

Nossa abordagem se concentrará no aprendizado supervisionado, que por sua vez funciona através de um conjunto de dados com entradas e saídas conhecidas, com o objetivo de aprender uma regra geral de mapeamento entre os dois conjuntos Na literatura, dá-se o nome de conjunto de treinamento a essa base de dados. O aprendizado é legitimado através de técnicas de validação cruzada, sendo a mais comum o método de *k-fold*.

O método de validação cruzada por *k-fold* consiste no particionamento aleatório dos dados de treinamento em k grupos disjuntos (*folds*) de tamanhos aproximadamente iguais. Após a divisão, selecionamos k-1 grupos para a realização do ajuste do modelo (simulando seu treinamento na base de dados) e o grupo restante para a validação dos resultados (simulando a etapa de teste em bases onde não se conhece as saídas dos dados). Isto é, nesse grupo isolado é onde comparamos as predições realizadas sobre o modelo treinado (ajustado) e extraímos uma medida de qualidade desse ajuste. Esse processo é realizado até que os k grupos tenham sidos usados como conjunto de validação (ou de teste). No final do procedimento teremos k medidas de qualidade de modo que possamos obter sua média e desvio padrão, verificando assim a consistência e a capacidade de generalização do modelo. A Figura 5 ilustra o processo para $k = 5$. Perceba que a cada um dos 5 passos, um novo conjunto é escolhido para simular os dados de teste, sendo os outros 4 restantes responsáveis por simular os dados de treinamento do modelo.

Figura 5 – Representação do processo de *5-fold*: as barras horizontais representam o conjunto de dados com saídas conhecidas.



Fonte: Elaborado pelo autor.

Sobre as medidas utilizadas para quantificar a consistência e bondade de ajuste do modelo sobre suas predições, no caso de regressão, é comum o uso do score de validação, construído sobre o erro quadrático médio (EQM)

$$cv_k = \frac{1}{K}\sum_{k=1}^{K} EQM_k = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{n}\right),$$

onde n é o número de indivíduos do grupo de validação, y_i e $\hat{y}_i$ são os valores exato e estimado da característica de interesse (variável resposta), respectivamente, e K o número de grupos da validação. Como $EQM \geq 0$, quanto mais próximo o score de validação estiver de zero, mais acurado é o nosso modelo.

Em alguns casos, a adoção do score baseado na medida de erro quadrático médio é um tanto quanto desinteressante, por exemplo, se a escala dos dados for pequena ou se o EQM não for conveniente para explicitar a qualidade do modelo. Dessa forma, uma boa alternativa consiste no uso do coeficiente médio de variação explicada,

$$CV = \frac{1}{K}\sum_{k=1}^{K}\frac{s_k}{\overline{\boldsymbol{X}}_K}, \tag{3.5}$$

onde s e $\overline{\boldsymbol{X}}$ são as estimativas do desvio padrão e média dos dados do k-ésimo grupo de validação, respectivamente. O CV varia dentro de um intervalo $[0, 1]$ de modo que quanto mais próximo de 1 melhor é a captação de informação do modelo e sua explicação sobre os dados. Se o valor for muito próximo de zero, então há indícios de que o modelo está mal especificado.

Vamos agora retomar a Definição 1, onde o processo Gaussiano é estabelecido sobre o vetor de observações $\boldsymbol{x}$ obtido em um tempo e/ou espaço s e $s + h$. O uso do modelo de regressão por processos Gaussianos sobre a perspectiva de aprendizado de máquinas supervisionado consiste em uma etapa de treinamento, ou ajuste do modelo, e uma de teste, onde o mesmo será validado e teremos uma noção sobre sua qualidade de predição. Assim, simplificando a notação tempo-espacial como proposto em [25], podemos reescrever o modelo da seguinte forma[1],

$$\begin{aligned} y_i &= f_i(\boldsymbol{x}) + \epsilon_i \\ f_i(\boldsymbol{x}) &\sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \end{aligned} \tag{3.6}$$

onde $\boldsymbol{x}$ e $\boldsymbol{x}'$ são vetores de observações sobre dois indivíduos distintos, pertencentes a base de dados, possuindo um grau de similaridade tempo-espacial, levado em consideração no processo pela função *Kernel*. Generalizando essa ideia para uma amostra aleatória (iid), representada pela matriz $\boldsymbol{X}$ da Seção (2.2), temos que o treinamento do modelo leva em conta o grau de similaridade existente nas interações dois a dois de todos os indivíduos observados na amostra.

[1] Lembrando que $f_i(\boldsymbol{x}) = \phi(\boldsymbol{x})\boldsymbol{w}$ e $\boldsymbol{w} \sim N_p(\boldsymbol{0}, \boldsymbol{\Sigma})_w$ sobre a abordagem realizada na Seção 2.2.

3.2.2 Uma Introdução ao Algoritmo L-BFGS

No início da Seção 3.2, discutimos rapidamente sobre a estimação dos hiperparâmetros e do ajuste dos metamodelos por sub-otimizações e via processo de aprendizado. Essa necessidade foi vista também na especificação da função *Kernel* na Subseção 2.2.3.

Dessa forma, tome como base o Método de Newton para problemas de otimização, baseado na obtenção de soluções via Método do Gradiente e na redução do erro de determinação. O Método de Newton tem uma boa convergência, no entanto sofre com a presença de mínimos locais, isto é, os valores tendem a se concentrar nas vizinhanças dos limitantes da função objetivo. Logo, se tomarmos um valor distante daqueles presentes na vizinhança da função objetivo, o método de otimização pode não convergir [16]. Essa constatação pode ser observada no Teorema 2, onde $G(\cdot)$ denota o domínio de vizinhança de raio r da solução x_* e $Lip_\lambda(\cdot)$[2] é dita família de Lipschitz.

Teorema 2. *Suponha $f : \mathbb{R}^n \to \mathbb{R}$ uma função duas vezes diferenciável num conjunto aberto e convexo $\mathcal{D} \subset \mathbb{R}^n$ e que $\nabla^2 f(x) \in Lip_\lambda(G(x_*, r))$. Então, existe $\epsilon > 0$, tal que $\forall x_o \in G(x_*, \epsilon)$, se $\nabla^2 f(x)$ é positiva definida para todo $x \in G(x_*, \epsilon)$, o método de Newton está bem definido, a sequência gerada pelo método converge para x_*, e existe $c > 0$, tal que*

$$||x_{k+1} - x_*|| \leq c||x_k - x_*||^2.$$

Demonstração. A demonstração do teorema encontra-se em [17]. □

Sobre essa restrição, métodos Quasenewtonianos aparecem na tentativa de contornar ou suavizar a limitação do método de Newton, procurando manter a boa convergência local do mesmo. Esse método pode ser usado toda vez que a matriz Hessiana, ou o Jacobiano, do processo - isto é, a matriz positiva definida formada pelo gradiente da função objetivo ($\nabla^2 f(x)$) - for inviável ou muito cara de se computar a cada iteração do processo, uma vez que tal matriz é substituída por avaliações aproximadas do gradiente. Dentre os algoritmos baseados no método de quase-newton, o `L-BFGS` é um dos mais utilizados.

Sendo uma variante do método `BFGS`, desenvolvido pelos matemáticos Charles G. Broyden , Roger Fletcher , Donald Goldfarb e David Shanno, em 1970, o algoritmo `L-BFGS` busca minimizar a função objetivo ($\mathcal{H}(\boldsymbol{x})$), de onde se deseja estimar as variáveis de interesse (em nosso caso, os hiperparêmtros da função Kernel do metamodelo) usando uma quantidade limitada (fator `L`) da memória do computador.

Sobre [16] e tendo como objetivo maximizar a verossimilhança presente no processo de ajuste dos parâmetros do modelo (2.7), o algoritmo se inicia com um valor aleatório

[2] $Lip_\lambda(\cdot)$ refere-se a família de funções de Lipschitz que assegura critérios para suavidade da função indexada a esta, sendo mais robusta que as condições de continuidade uniforma. Para maiores detalhes, consulte [23]

acerca dos hiperparâmetros, $\hat{\boldsymbol{\theta}}_0$, prosseguindo iterativamente para refinar essa estimativa com uma sequência de melhores estimativas da variável. As derivadas da função de verossimilhança para variável na k-ésima iteração, $\nabla\mathcal{L}_k(\boldsymbol{\theta}|\boldsymbol{x})$, são usadas como critério do algoritmo na identificação do gradiente descendente, formando assim uma estimativa para matriz Hessiana, pela segunda derivada de $\mathcal{L}$.

Tradicionalmente, o `BFGS` tem cada passo definido da seguinte maneira,

$$\begin{aligned} \hat{\theta}_{k+1} &= \hat{\theta}_k - \alpha_k H_k \nabla\mathcal{L}_k \\ H_{k+1} &= V_k^T H_k V_k + \rho_k s_k s_k^T, \end{aligned} \tag{3.7}$$

onde $H_k = B_k^{-1}$ representa a aproximação da inversa da Hessiana, α_k é o tamanho do passo escolhido do algoritmo de acordo com alguma estratégia e

$$\begin{aligned} \rho_k &= \frac{1}{y_k^T s_k}, \qquad & V_k &= I - \rho_k y_k s_k^T, \\ s_k &= \theta_{k+1} - \theta_k, \qquad & y_k &= \nabla\mathcal{L}_k + 1 - \nabla\mathcal{L}_k. \end{aligned} \tag{3.8}$$

O `L-BFGS` estima implicitamente a inversa da matriz hessiana, $H_k = B_k^{-1}$, através de m pares de vetores $\{s_k, y_k\}$ afim de direcionar a busca da solução sobre os espaço da variável. O produto $H_k\nabla\mathcal{L}_k$ é substituído por uma sequência de produtos internos e somas de vetores sobre $\nabla\mathcal{L}_k(\boldsymbol{\theta}|\boldsymbol{x})$ e $\{s_k, y_k\}$. Além disso, como qualquer método Quasenewtoniano, o `L-BFGS` exige que a função objetivo seja ao menos duas vezes diferenciável.

Apesar da nossa discussão ser apenas introdutória, tornando-se o tratamento matemático do processo superficial, apresentamos o pseudocódigo sobre o funcionamento do `L-BFGS`, Algoritmo 2 e Algoritmo 3, tal como segue em [16].

Algorithm 2 Produto $H_k\nabla\mathcal{L}_k$

```
1:  INÍCIO
2:  q ← ∇L_k
3:      para i ← k-1 até k-m faça
4:          v_i ← ρ_i s_i^T q
5:          q ← q − v_i y_i
6:      fim para
7:  r ← H_k^0 q
8:      para i ← k-m até k-1 faça
9:          β ← ρ_i y_i^T r
10:         r ← r + s_i(v_i − β)
11:     fim para
12: fim
```

Algorithm 3 L-BFGS

1: Início
2: escolha $\theta_0, m > 0$
3: $k \leftarrow 0$
4: **para** k $\leftarrow 0$ **até** convergência **faça**
5: escolha H_k^0
6: calcule $p_k \leftarrow -H_k \nabla \mathcal{L}_k$ usando o Algoritmo (2)
7: $x_{k+1} \leftarrow x_k + \alpha_k p_k$
8: **se** $k > m$ **então**
9: descarte o vetor $\{s_{k-m}, y_{k-m}\}$
10: **fim se**
11: calcule e armazene $s_k \leftarrow \theta_{k+1} - \theta_k$, $y_k \leftarrow \nabla \mathcal{L}_{k+1} - \nabla \mathcal{L}_k$
12: $k \leftarrow k + 1$
13: **fim para**
14: **fim**

3.3 APLICAÇÃO DO METAMODELO NO PROCESSO DE OTIMIZAÇÃO ESTRUTURAL

Vimos na Seção 3.2 que metamodelos buscam substituir a função de avaliação original em um processo de otimização via metaheurísticas por uma função aproximada, menos complexa e mais barata de se computar.

Levando em conta o método de Evolução Diferencial (DE), do tipo *DE/rand/1/bin*, visto na Subseção 3.1.1, e o modelo de regressão via processos Gaussianos, discutido na Seção 2.2 e na Subseção 3.2.1, desejamos incorporar a estrutura do modelo de regressão por processos Gaussianos como metamodelo na solução de problemas de otimização estrutural via DE. Assim, temos:

$$\hat{\mathcal{H}} = f(\boldsymbol{X}) \sim \mathcal{GP}(m(\boldsymbol{X}), k(\boldsymbol{X}, \boldsymbol{X}')) \tag{3.9}$$

onde $\boldsymbol{X}$ e $\boldsymbol{X}'$ são as bases dados (matrizes) de treinamento e teste, respectivamente.

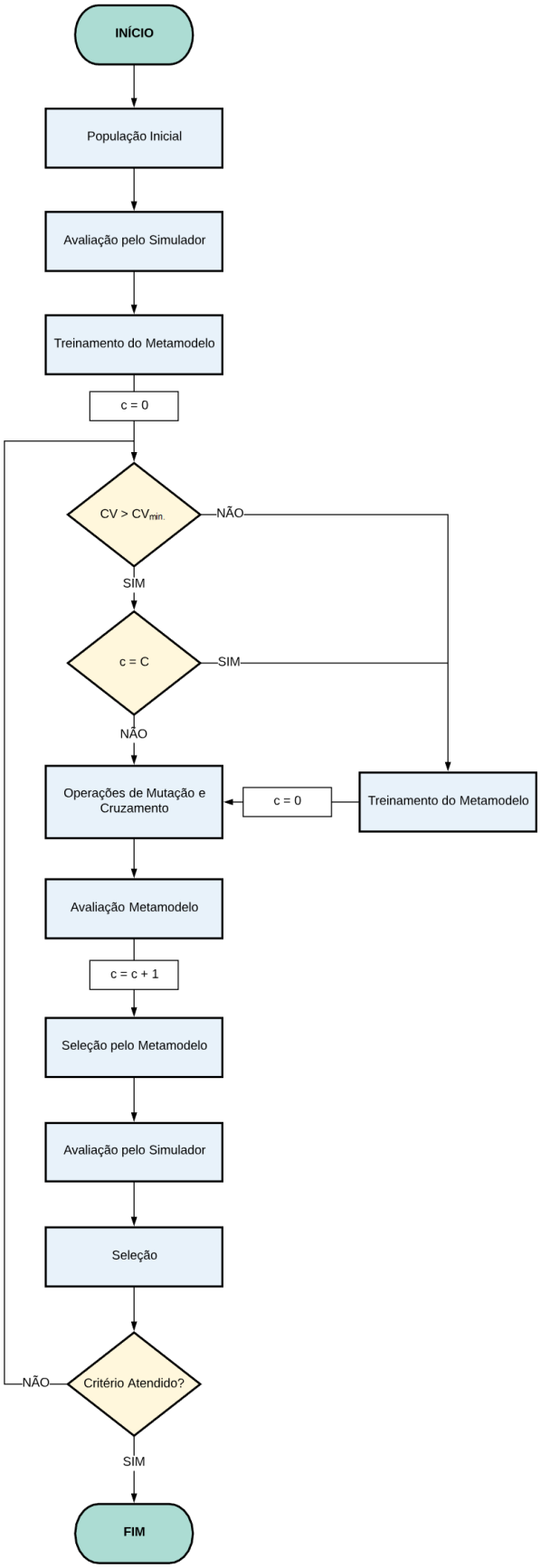
Seguindo o que foi proposto em [14], no método de DE assistido por metamodelo, o modelo de regressão por processos Gaussianos será treinado/ajustado considerando a população inicial avaliada a priori sobre a função original. Após a seleção dos quatro melhores indivíduos (chamados de 'mães') e sua passagem pelas operações básicas de mutação e cruzamento, ocorrerá a geração de uma nova amostra de descendentes a partir desses quatro pais. O metamodelo treinado entra então como avaliador, predizendo os novos valores para os indivíduos alterados. Assim, seleciona-se quatro novos candidatos, um herdado de cada pai, e se a avaliação exata de um deles for melhor que a de seus pais, o mesmo é substituído pelo seu descendente.

No processo de treinamento bem como nas avaliações, usaremos como medida de qualidade do metamodelo o coeficiente médio de variação explicada ($CV_{médio}$) (3.5). Dessa forma se para uma dada geração, após a etapa de seleção, o metamodelo apresentar um

$CV_{médio} < CV_{min.}$, o mesmo será retreinado. Quando essa especificação for atendida, então um número fixo de gerações C serão realizadas sem a atualização do modelo. Todo esse procedimento é ilustrado pelo fluxograma da Figura 6.

Considerando a influência da metamodelagem sobre o desempenho da DE, essa proposta foi feita não só para reduzir o número de avaliações que usam o simulador/modelo original, mas também como fator importante, do ponto de vista estatístico, na melhora da acurácia do processo.

Figura 6 – Fluxograma do processo de Evolução Diferencial assistido por metamodelo baseado em um modelo de regressão por processos Gaussianos.



Fonte: Elaborado pelo autor.

4 APLICAÇÕES E RESULTADOS

Neste Capítulo, introduziremos uma rápida ilustração de como funciona o processo Gaussiano como interpolador e quando indexado ao modelo de regressão segundo a metodologia de aprendizado de máquinas (Seção 4.1) Em seguida, na Seção 4.2 trataremos do processo de otimização estrutural sobre as cinco estruturas propostas em [14], aplicando toda a metodologia discutida até então, segundo o objetivo do trabalho.

4.1 ILUSTRAÇÃO DO PROCESSO GAUSSIANO SOBRE MODELOS DE REGRESSÃO

Vamos ilustrar rapidamente como obtemos realizações a partir de um processo Gaussiano, que resulta na avaliação de uma função sobre um conjunto de pontos amostrados sobre uma função específica, determinando o processo Gaussiano a priori.

Consideramos a função que se segue:

$$y_i = f(x_i) = x_i^2 - a\cos(2\pi x_i) + 2a + \epsilon_i,$$

onde $a \geq 0$ e $\epsilon_i \sim N(0, 1)$.

Em seguida, sobre a estrutura da função de covariância, tal qual em (2.18), determinamos arbitrariamente os seguintes hiperparâmetros $\sigma^2 = 30$ e $l = 10$. Isto é, estamos determinando que nossa função de covariância vai captar tendências com altas amplitudes ou distanciamento da média e com alta suavidade (baixas frequências). Consideramos também que a constante da função escolhida seja $a = 10$.

Tomamos um ponto conhecido (x_i, y_i), amostrado aleatoriamente sobre a função, e usamos a propriedade da distribuição condicional da normal multivariada - propriedade **P.5.**, na Subseção 2.1.2 - para prever novos pontos condicionados aos elementos anteriores, tal que

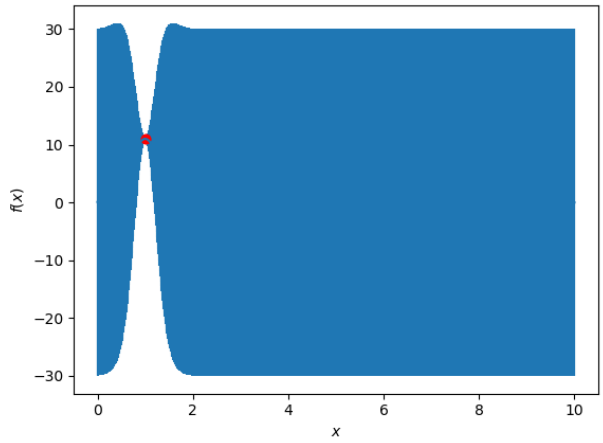
$$p(x'|y, x) = N(\mu_x + \Sigma_{xy}\Sigma_y^{-1}(y - \mu_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T),$$

onde x' é a nova observação.

Por fim, vamos determinar uma banda representando um desvio padrão da média entre os valores $y = 30$ e $y = -30$, isso nos ajuda a determinar um envelopamento da função condicionada aos pontos dados (em outras palavras, definimos um intervalo de confiança para funções ajustadas sobre o processo Gaussiano).

A Figura 7 ilustra o ponto amostrado sobre a função, percebemos que ao inseri-lo as bandas de confiança se atualizaram em torno do mesmo.

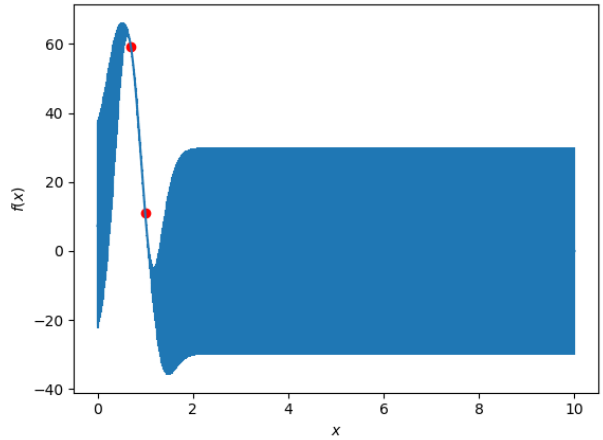
Figura 7 – Ponto $(1.0, 11.0)$ definido arbitrariamente, segundo as definições impostas para o processo Gaussiano em questão.



Fonte: Elaborado pelo autor.

Condicionado a esse ponto gerado e à estrutura de covariância especificada, restringimos essencialmente a provável localização de pontos adicionais. Dessa forma, tomando x_{i+1} com um novo valor arbitrário no espaço podemos obter sua previsão segundo a sua probabilidade condicional. Isso faz com que a banda de confiança seja novamente atualizada, reconfigurando a estrutura de covariância dos dados bem como a localização de novos pontos inseridos próximos a x_i e x_{i+1} (Figura 8).

Figura 8 – Novo ponto inserido no processo Gaussiano $(0.7, 59.2)$ a partir da distribuição condicional.

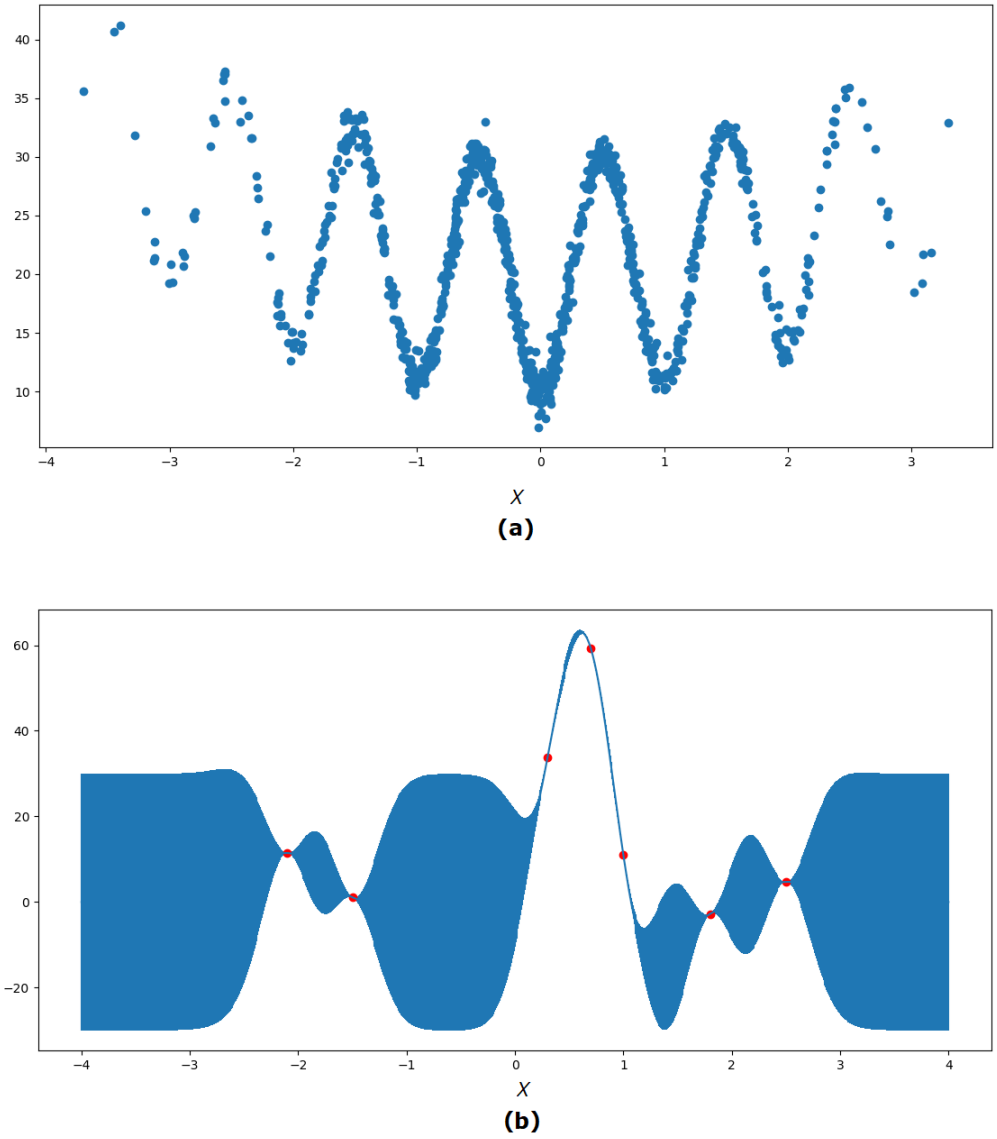


Fonte: Elaborado pelo autor.

Efetuando esse processo inúmeras vezes, através de amostragens, podemos mostrar como a função de covariância influencia e auxilia o modelo a captar a estrutura da função

original. Logo, para uma amostra de tamanho $n = 10$ obtemos o que segue na Figura (9).

Figura 9 – Processo Gaussiano sobre uma amostra de 10 observações, todas estimadas segundo a probabilidade condicional. **(a)** Forma da função $f(x)$, com $a = 10$, para uma amostra de 1000 dados simulados. **(b)** ajuste do processo Gaussiano via probabilidade condicional.

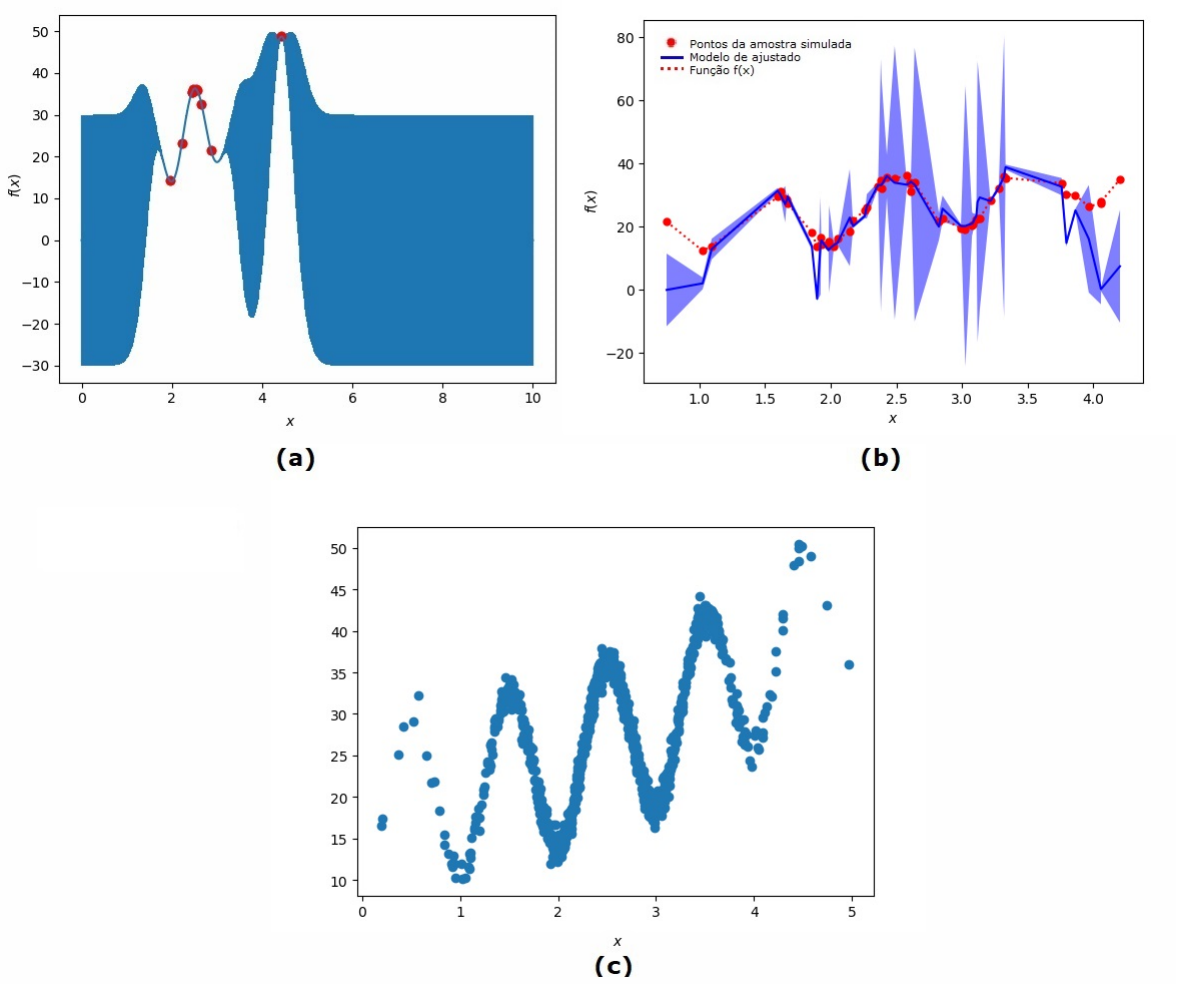


Fonte: Elaborado pelo autor.

Tomamos agora a uma amostra $\boldsymbol{x}$, tal que $\boldsymbol{x} \sim N(2.5; 0.5)$. Aplicando a função f sobre ela, com $a = 10$ e seguindo a abordagem de aprendizado de máquinas, obtemos o conjunto de treinamento do modelo, de tamanho $n = 200$, definido por $\mathcal{D} = (\boldsymbol{x}, f(\boldsymbol{x}))$. Nesse conjunto, utilizando cerca de 75% dos dados de treinamento, ajustamos o modelo de regressão por processos Gaussianos utilizando a biblioteca `sklearn` [21], que contém a ferramenta de ajuste adequada para grandes amostras e dados de alta dimensionalidade (caso que será tratado na Seção 4.2). Após a etapa de ajuste, simulamos a fase de

teste/validação do modelo sobre os 25% dos dados restantes. Além disso, 20 observações selecionadas aleatoriamente desta amostra foram usadas no ajuste do processo Gaussiano segundo a abordagem anterior.[1] O resultados da predição podem ser vistos na Figura 10.

Figura 10 – **(a)** Interpolação do processo Gaussiano para uma subamostra, de tamanho 20, dos dados. **(b)** predição e ajuste do modelo por processos Gaussianos sobre o conjunto de teste gerado por 25% da amostra aleatória dada (ou dados de treinamento) de tamanho 200. **(c)** Comportamento geral dos dados sobre a função f para uma mesma amostra de tamanho 1000.



Fonte: Elaborado pelo autor.

Notamos que pelo gráfico **(b)** da Figura 10 as linhas dos modelos ajustado (em azul) acompanham bem a do modelo teórico (pontilhada) captando a tendência dos dados de teste (pontos), com exceção de seus extremos. O envelope em azul foi ajustado considerando um intervalo de confiança de 95%, isso nos diz que qualquer outro ajuste

[1] A amostra de tamanho reduzido foi realizada uma vez que o algoritmo programado para abordagem de predição por probabilidade condicional não trabalhou bem sobre amostras superiores a 50 observações.

feito dentro desse intervalo, terá a mesma bondade e características do modelo considerado (linha azul).

Efetuamos o processo de validação cruzada via *k-fold*, para $k = 5$, e calculamos os coeficientes de variabilidade explicada do modelo em cada rodada. Os resultados podem ser vistos na Tabela 2.

Tabela 2: Resultado do coeficiente de variação explicada (CV) para cada rodada da validação cruzada via *5-fold* ($CV_{médio} = 0.8666 \pm 0.1419$).

k-fold	rodada 1	rodada 2	rodada 3	rodada 4	rodada 5
CV	0.8341	0.9863	0.9572	0.6342	0.9210

Na quarta rodada da validação por *k-fold*, percebemos um valor do coeficiente de variabilidade bastante diferente dos demais. Esse fato pode ser explicado quando o conjunto de teste da rodada engloba boa parte dos valores atípicos da amostra. Novamente, ao olharmos o gráfico **(b)** da Figura 10, percebemos que nos extremos da distribuição dos dados os pontos encontram-se mais esparsos, além do desnivelamento presente entre a função ajustada e a função teórica.

4.2 PROBLEMA DE OTIMIZAÇÃO DE ESTRUTURAS METÁLICAS NA ENGENHARIA

Vamos assumir o problema de otimização estrutural tal qual discutido em [14], formulado por

$$\begin{aligned} min. \quad & \mathcal{H}(\boldsymbol{x}) = \sum_{k=1}^{p} \gamma A_k \left(\sum_{j=1}^{N_k} L_j \right) \\ rest. \quad & \frac{|s_{j,l}|}{s_{adm}} - 1 \leq 0, \\ & \frac{|u_{i,l}|}{u_{adm}} - 1 \leq 0, \quad A_i^{inf} \leq A_i \leq A_i^{sup}, \end{aligned} \tag{4.1}$$

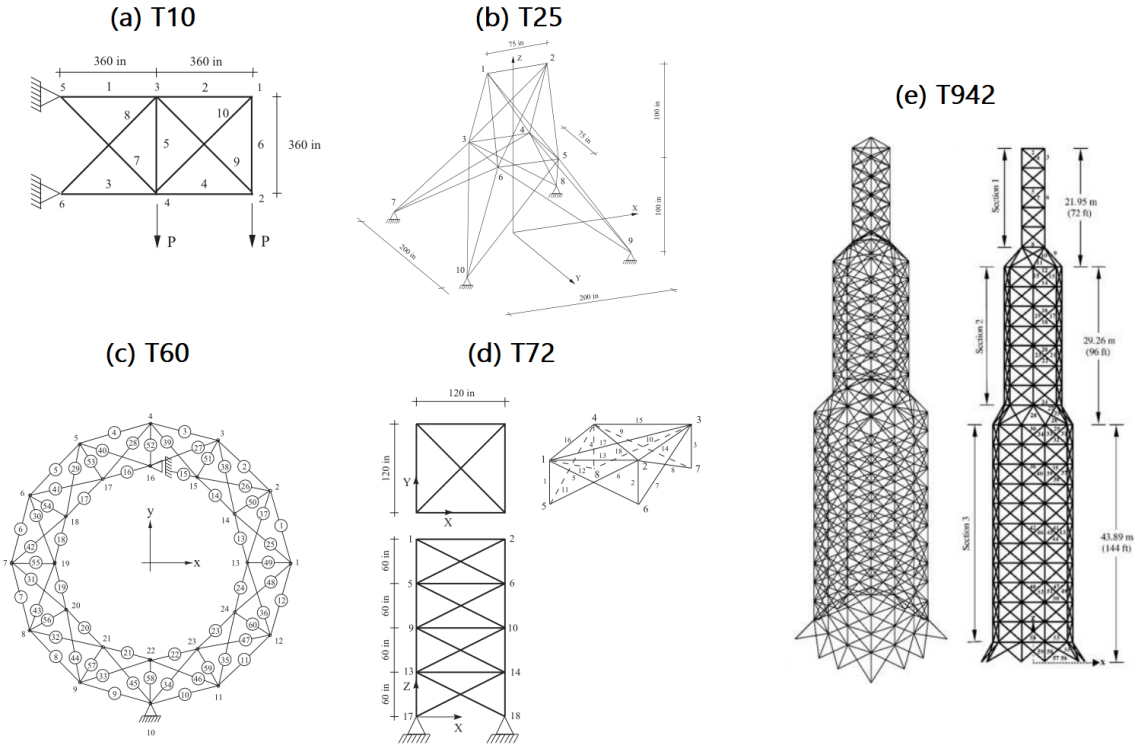
para $j = 1, 2, \ldots, N$, $i = 1, 2, \ldots, M$ e $l = 1, 2, \ldots, N_L$. Chamamos $\boldsymbol{x} = \{A_1, A_2, \ldots, A_n\}$ o conjunto de variáveis de projeto representando as áreas transversais das barras que constituem as estruturas tratadas. Sobre os dados e variáveis do problema, temos:

- s e u são as medidas de tensão e deslocamento de eixo das barras que compõem as funções de restrição, com valores máximos admissíveis representados por s_{adm} e u_{adm}, respectivamente.
- γ é o peso específico do material.
- L_j é o comprimento da j-ésima barra da estrutura.

- N é o número total de barras da estrutura.
- N_k é o número de membros do k-ésimo grupo que compartilham da mesma área transversal.
- N_L é o número de cargas aplicadas sobre a estrutura.

Vamos considerar as cinco estruturas metálicas - T10, T25, T60, T72 e T942 - tratadas em [14], sendo elas compostas por 10, 25, 60, 72 e 942 barras, respectivamente, como mostra a Figura 11.

Figura 11 – Ilustração das estruturas metálicas consideradas, neste trabalho, para aplicação do método de Evolução Diferencial assistido pelo modelo de regressão por processos Gaussianos.



Fonte: Extraído de [14].

As estruturas são modeladas segundo (4.1) e utilizamos o Método Deb (Subseção 3.1.2) para o tratamento das restrições. Buscamos, segundo o objetivo proposto, aplicar o processo de otimização estrutural via DE assistido pelo modelo de regressão por processos Gaussianos (2.17), discutido na Seção 3.3. Para avaliar os resultados, comparamos o tempo de processamento da abordagem proposta neste trabalho, bem como o valor final da função objetivo (peso da estrutura), com aqueles obtidos segundo o método da DE com ausência do modelo substitutivo, visto na Subseção 3.1.1 e ilustrado pela Figura 3. Nesse intuito, o metamodelo busca prever os valores de c_{sum} acerca da estrutura, segundo

as p áreas transversais das barras, substituindo os cálculos da função objetivo e suas restrições. Esperamos, sobre a abordagem proposta, que haja ganho na determinação do peso estrutural, isto é, ele diminua se comparado ao DE original, e/ou haja uma queda no tempo de processamento ao compará-lo.

Dessa forma, iniciamos a aplicação com uma análise previa das estruturas através de uma amostra contendo 1000 observações em ambiente *R* e *Python*. No primeiro *software* tratamos da verificação de normalidade dos dados, como a aplicação do Teste de Mardia (Subseção 2.1.4), sobre um nível de significância de 0.05. A segunda linguagem de programação, utilizamos no ajuste do modelo e a verificação da qualidade e comportamento do mesmo sobre a perspectiva de aprendizagem de máquinas. Como medidas de qualidade do modelo, levamos em conta o coeficiente de variabilidade explicada e a frequência relativa da ordenação entre os dados previstos e os dados originais (isto é, verificamos o quão os valores preditos mantinham a ordenação dos dados segundo o vetor de saídas de teste/validação do processo).

Finalmente, ainda sobre a parte de análise prévia das estruturas, propomos dois tipos de tratamento dos dados:

- Ponderação segundo pesos amostrais devido ao excesso de zeros contido nos dados de saída (c_{sum}) do conjunto/amostra de treinamento. Essa ponderação é feita sobre a frequência relativa entre as saídas (c_{sum}) nulas e não-nulas, tal como em [22].
- Transformação logarítmica nos dados, afim de obter uma tendência homoscedástica e potencializar a qualidade do ajuste e o aumento da taxa de ordenação das saídas preditas.

O processo de otimização foi realizado em linguagem de programação *Python*, integrado a etapa de avaliação pelo simulador em linguagem `C/C++`, sobre um ambiente computacional NP550PC-ADBR1 *Samsumg*, com processador Intel Core i7-3630QM. SO Manjaro x86, Kernel 4.14.83-1 com Xfce 4.12.4. 8Gb RAM e placa de vídeo Nvidia GeForce GT 630M. HD 1Tb. Nele, partimos de dados simulados segundo uma distribuição normal multivariada contendo 50 mães que gerarão 4 filhos cada, compondo uma amostra inicial com 200 indivíduos. Seguindo a estrutura do fluxograma da Figura 6, os 200 indivíduos, após o treinamento, passam pelas operações básicas do processo e são avaliados pelo metamodelo, desses são selecionados os 50 melhores (um de cada mãe) que são avaliados pelo simulador original. A etapa seguinte consiste na comparação dos 50 herdeiros com suas respectivas genitoras, se o herdeiro for melhor que sua genitora, então ele a substitui, caso contrário o mãe é mantida.

Para o processo de otimização estrutural, tanto sobre o DE assistido pelo modelo de regressão, quanto pelo DE original, consideramos:

- Como critério de parada do processo usamos um total de 15 mil avaliações realizadas pelo simulador.
- As constantes $CR = 0.9$ e $F = 0.5$, como em [14].
- Como medidas de controle assumimos $CV_{min.} = 0.6$ e $C = 5$.
- Para a estimação dos hiperparâmetros a sub-otimização `L-BFGS`, visto em (3.2.2), associada ao método de busca aleatória (*random search*), selecionando os hiperparâmetros com maior frequência/taxa da ordenação dos dados para atualizar/ajustar o modelo.

Os resultados são apresentados a seguir.

Análise de Normalidade dos Dados

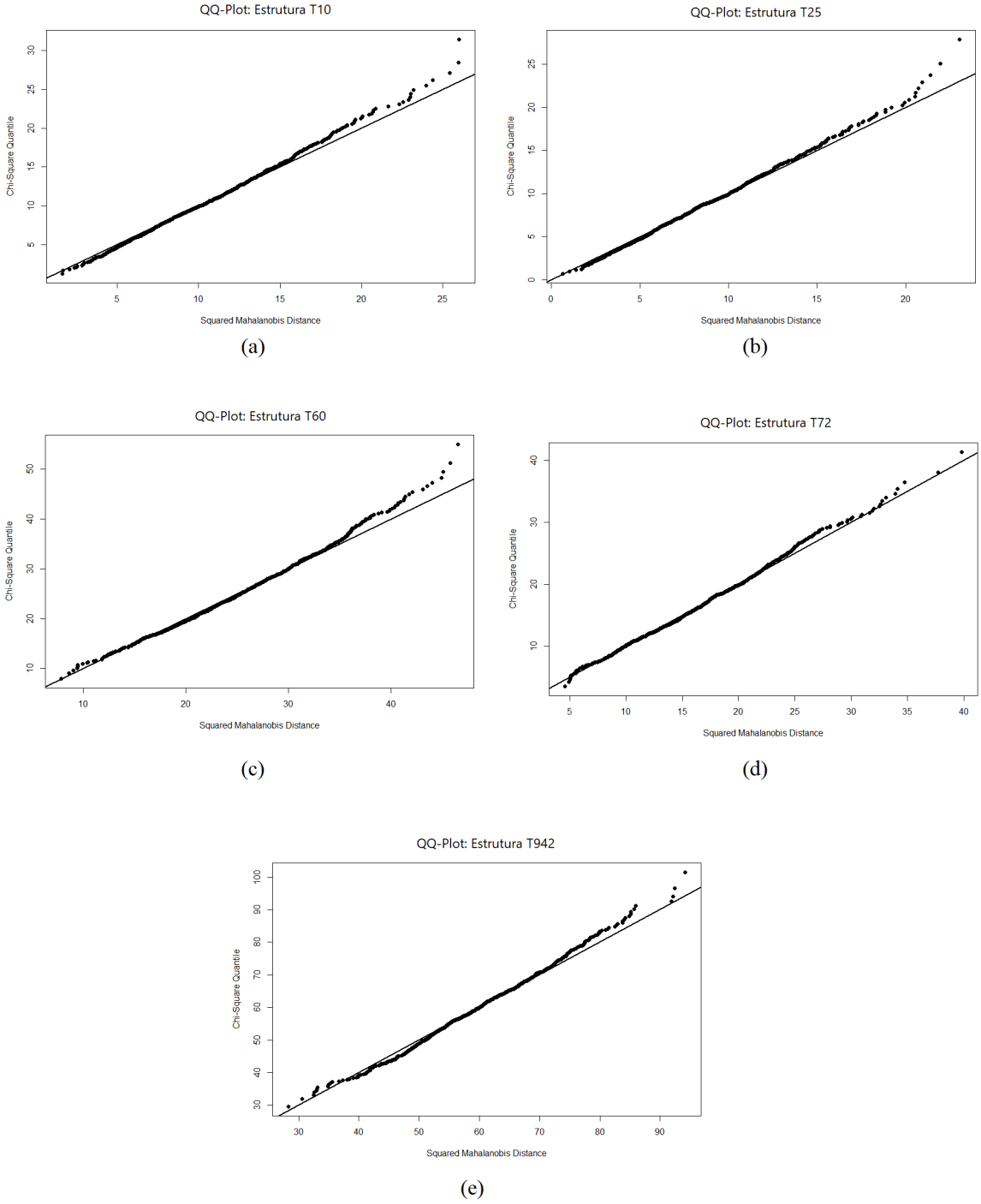
Aplicando o teste de Mardia sobre a amostra de análise, obtivemos resultados significativos em quase todas estruturas, como pode ser visto na Tabela 3. Apenas na estrutura T924 rejeitamos a etapa de análise da curtose, uma vez que seu p-valor é muito inferior ao nível de significância estabelecido.

Tabela 3: Resultados do Teste de Mardia para a análise de multinormalidade das estruturas, a um nível de significância de 0.05.

Mardia	**T10**	**T25**	**T60**	**T72**	**T924**
$Assimetria_{p-valor}$	0.9893	0.4571	0.4641	0.8501	0.999
$Curtose_{p-valor}$	0.1622	0.1256	0.1254	0.3260	2.6338e-5

Durante a aplicação do teste, construímos os *qq-plots* para cada caso (Figura 12). Notamos que, apesar do relativo bom comportamento sobre a reta de normalidade, a extremidade superior de todos os casos apresentou certo desvio. Porém, esse desvio também ocorreu na extremidade inferior da estrutura T924 (há uma leve “barriga"formando-se entre a reta e os pontos). Quando o desvio ocorre sobre ambas as extremidades (inferior e superior) da reta, em relação aos pontos, comumente refere-se ao fato da distribuição possuir caudas pesadas. Tal fato, pode explicar a rejeição do teste de Mardia sobre a curtose da estrutura em questão.

Figura 12 – QQ-Plot sobre multinormalidade dos dados: **(a)** estrutura T10 (10-barras). **(b)** estrutura T25 (25-barras). **(c)** estrutura T60 (60-barras). **(d)** estrutura T72 (72-barras). **(e)** estrutura T942 (942-barras)



Fonte: Elaborado pelo autor.

Efeito sobre a Transformação Logarítmica no Ajuste Prévio do Modelo

Ajustamos o modelo de regressão por processos Gaussianos sobre os dados originais e sobre a transformação logarítmica dos mesmos. De modo geral notamos uma melhora nos medidores de qualidade dos modelos (taxa de ordenação e coeficiente de variabilidade explicada) após a transformação.

Tabela 4: Resultado das análises prévias para o modelo da estrutura de 10 e 25 barras.

	T10			**T25**		
Dados	**Tx.**	$CV_{médio}$	*sd*	**Tx.**	$CV_{médio}$	*sd*
originais	0.37	0.20	0.31	0.26	0.91	0.02
log.	0.41	0.85	0.17	0.68	0.96	0.02

Enquanto sobre as estruturas T10 e T25 (Tabela 4) a melhora foi observada sobre todas as medidas de qualidade estipuladas, nas estruturas T60, T72 e T942, por sua vez, observamos a piora do coeficiente de variabilidade do modelo sobre T942 (Tabela 5). Isso já era esperado, uma vez que optamos por ignorar a rejeição no teste de Mardia dessa estrutura.

Tabela 5: Resultado das análises prévias para o modelo da estrutura de 60, 72 e 942 barras.

	T60			**T72**			**T942**		
Dados	**Tx.**	$CV_{médio}$	*sd*	**Tx.**	$CV_{médio}$	*sd*	**Tx.**	$CV_{médio}$	*sd*
originais	0.44	0.31	0.14	0.37	0.20	0.31	0.05	0.40	0.49
log.	0.44	0.65	0.01	0.41	0.85	0.17	0.55	0.14	0.01

Ainda sobre a transformação logarítmica, apesar da melhora de qualidade no ajuste do modelo sobre os dados, a distribuição normal multivariada desejada é perdida, isto é, não temos mais a garantia da distribuição probabilística em questão sobre os dados. Apesar disso, devido a importância em se potencializar a pedição do modelo para o processo de otimização, optamos trabalhar com os dados transformados.

Otimização Estrutural

Considerando o método proposto (Evolução Diferencial assitida pelo Modelo estatístico, DEM) e a Evolução Diferencial original (DEO), utilizamos cinco medidas para compará-los, sendo elas: coeficiente de variabilidade médio ($CV_{\text{méd.}}$), o desvio padrão relacionado a essa média (sd), o valor predito de $c_{\text{sum.}}$, a valor ótimo encontrado para função objetivo ($peso$) e, por fim, o tempo médio de execução de cada processo ($\text{tempo}_{\text{méd.}}$, em segundos).

Como vimos no processo da DEM, ilustrado pelo fluxograma da Figura 6, temos duas constantes de controle do processo. Como havíamos definido no início dessa subseção, o coeficiente de variabilidade explicada mínima do processo foi definido como 0.60 (valor que, pela literatura, é considerado como um bom valor da medida). Já a constante c, que refere-se a janela de folga do processo (onde o modelo vai trabalhar iterativamente sem receber atualizações dos dados, até que esta seja cumprida) foi definida de modo experimental. Variamos o valor de c entre 5, 10 e 15 iterações em cada caso, comparando se houve diferença entre as medidas de qualidade dos três casos e se houve queda no tempo de processamento.

Por fim, comparamos o melhor caso do método DEM, observado sobre a variação de c (isto é, consideramos a janela que nos desse o menor tempo de processamento sem alterar significativamente as outras quatro medidas de qualidade do modelo), com o procedimento original da DE. Para todos esses testes utilizamos o teste estatístico não-paramétrico de comparações de média de Wilcoxon [11], cuja hipótese nula (H_0) nos diz que não há diferença significativa entre as médias comparadas, a um nível de significância de 0.05.

Os resultados são observados nas Tabelas 6 a 10. De modo geral, o valor de $c = 15$ conseguiu diminuir significativamente o tempo de execução do método DEM, sem agregar perdas significativas as medidas de qualidade do modelo, em todos os casos tratados. Assim, tomando DEM-15 na comparação com o procedimento original DEO, observamos uma relativa piora no valor da função objetivo ($peso$) e no tempo médio de processamento. Apenas na estrutura T60 (Tabela 8), observamos uma melhora significativa do valor da função objetivo do método proposto (DEM), custeado pelo aumento do tempo de processamento.

Por simplificação, apenas os resultados do teste de Wilcoxon para a comparação entre DEM-15 e DEO foram apresentados nas tabelas. Em todos os casos rejeitamos H_0, ou seja, há evidências que indicam diferenças significativas entre o valor ótimo da função objetivo comparado entre os dois métodos.

Tabela 6: Resultado do processo de otimização para o modelo estrutural de 10 barras. Teste de Wilcoxon: foi significativo, isto é, rejeitamos H_0, a um nível de significância de 0.05, com p-valor = 0.0256, sobre o *peso* médio obtido pelo método DEM-15 se comparado ao DEO.

T10				
Atributos	**DEM-5**	**DEM-10**	**DEM-15**	**DEO**
$CV_{méd.}$	1.00	0.99	0.99	$*$
sd	$3.81e-9$	$5.84e-5$	$3.96e-6$	$*$
$c_{sum.}$	0.00	0.00	0.00	0.00
$peso$	5465.27	5476.90	5457.60	5062.16
$tempo_{méd.}(s)$	181.08	123.97	80.41	1.52

Tabela 7: Resultado do processo de otimização para o modelo estrutural de 25 barras. Teste de Wilcoxon: foi significativo, isto é, rejeitamos H_0, a um nível de significância de 0.05, com p-valor = 0.0008, sobre o *peso* médio obtido pelo método DEM-15 se comparado ao DEO.

T25				
Atributos	**DEM-5**	**DEM-10**	**DEM-15**	**DEO**
$CV_{méd.}$	0.99	0.99	0.99	$*$
sd	$2.9e-7$	$5.32e-6$	$1.52e-3$	$*$
$c_{sum.}$	0.00	0.00	0.00	0.00
$peso$	510.82	516.43	517.25	484.05
$tempo_{méd.}(s)$	209.96	134.23	94.44	1.69

Tabela 8: Resultado do processo de otimização para o modelo estrutural de 60 barras. Teste de Wilcoxon: foi significativo, isto é, rejeitamos H_0, a um nível de significância de 0.05, com p-valor = 0.0001, sobre o *peso* médio obtido pelo método DEM-15 se comparado ao DEO.

T60				
Atributos	**DEM-5**	**DEM-10**	**DEM-15**	**DEO**
$CV_{méd.}$	1.00	1.00	1.00	$*$
sd	0.00	0.00	0.00	$*$
$c_{sum.}$	0.00	0.00	0.00	0.00
$peso$	310.87	311.09	310.93	334.03
$tempo_{méd.}(s)$	59.35	43.55	37.24	4.07

Tabela 9: Resultado do processo de otimização para o modelo estrutural de 72 barras. Teste de Wilcoxon: foi significativo, isto é, rejeitamos H_0, a um nível de significância de 0.05, com p-valor = 0.0001, sobre o *peso* médio obtido pelo método DEM-15 se comparado ao DEO.

T72				
Atributos	**DEM-5**	**DEM-10**	**DEM-15**	**DEO**
$CV_{méd.}$	0.99	0.99	0.99	$*$
sd	$2.67e-6$	$1.70e-5$	$8.24e-6$	$*$
$c_{sum.}$	0.00	0.00	0.00	0.00
$peso$	488.05	487.93	488.20	380.98
$tempo_{méd.}(s)$	236.68	134.43	91.30	3.82

Tabela 10: Resultado do processo de otimização para o modelo estrutural de 942 barras. Teste de Wilcoxon: foi significativo, isto é, rejeitamos H_0, a um nível de significância de 0.05, com p-valor = 0.0003, sobre o *peso* médio obtido pelo método DEM-15 se comparado ao DEO.

T942				
Atributos	**DEM-5**	**DEM-10**	**DEM-15**	**DEO**
$CV_{méd.}$	0.99	0.99	0.99	$*$
sd	$1.17e-6$	$7.01e-7$	$4.7e-7$	$*$
$c_{sum.}$	0.00	0.00	0.00	0.00
$peso$	487542.50	505009.90	496480.30	399560.46
$tempo_{méd.}(s)$	249.90	173.54	142.32	26.88

5 CONSIDERAÇÕES FINAIS

Incorporamos neste trabalho a abordagem de um modelo estatístico por processos Gaussianos, baseado em uma das distribuições mais importantes da área, a normal multivariada, em um método reconhecido na Computação para solução de problemas de otimização, a metaheurística baseada em Evolução Diferencial (DE). Dessa forma, propomos um novo processo para solução de problemas de otimização estrutural, nos permitindo medir a qualidade das soluções durante todo o processamento. No fim, foi possível compará-lo com a abordagem original da DE.

Segundo os resultados apresentados pelas análises, o modelo de regressão por processos Gaussianos apresentou um bom ajuste sobre os dados, melhorando seu poder preditivo após a transformação logarítmica proposta. Sobre o processo de otimização, o método proposto (Evolução Diferencia assistida pelo modelo de regressão por processos Gaussianos) não correspondeu as nossas expectativas. Com exceção da estrutura T60, onde obtivemos uma significativa melhora no peso da estrutura, em todos os outros casos houve uma piora bastante acentuada no tempo médio de processamento, bem como no valor da função objetivo (peso da estrutura), quando comparado ao DE original.

Durante toda a pesquisa, mais especificamente nos testes realizados sobre o processo de otimização via DE assistido pelo modelo de regressão por processos Gaussianos, encontramos alguns problemas que podem explicar os resultados inesperados:

- O processo de sub-otimização dos hiperparâmetros do modelo, o `L-BGFS`, apresentou problemas numéricos de convergência. Sempre que a não-convergência ocorria, o algoritmo recomeçava o processo de busca até que esse convergisse de fato para as estimativas dos hiperparâmetros. Isso acarretou, sem dúvidas, no aumento do tempo de processamento computacional.

- Ainda sobre o `L-BGFS`, notamos que as estimativas dos hiperparâmetros, em todos os testes, convergiam sempre para os limitantes do espaço de busca do código. Isto é, dado um 'chute inicial' para os valores dos hiperparâmetros, o processo de busca do valor ótimo iria ocorrer sobre um espaço limitado ente $[1.00 \cdot 10^{-3}, 1.00 \cdot 10^{3}]$. Dessa forma o hiperparâmetro σ^2 da função kernel convergia sempre para o limitante inferior e o hiperparâmetro l da mesma função convergia para o limitante superior. Isso pode ter afetado no cálculo da função objetivo pelo método, além de impossibilitar a interpretação sobre esses hiperparâmetros.

- Como o procedimento de otimização não pode ser interrompido, devendo priorizar o tempo de processamento, não podemos efetuar uma análise residual do modelo durante o processo. Isso pode nos acarretar perda de informação e/ou da identificação de estruturas (como heterocedasticidade nos erros, por exemplo) que vão de encontro

aos pressupostos feitos, que deveriam ser tratadas para uma melhor especificação do modelo.

Visto isso, como trabalhos futuros, propomos um estudo residual indexado ao processo de otimização estrutural, afim de verificar como os resíduos do modelo se comportam sobre a mudança do espaço de busca. Isso, além de contribuir para uma especificação do modelo de regressão por processos Gaussianos, poderia diminuir significativamente o valor da função objetivo, melhorando os resultados do método proposto. Outra abordagem, seria propor um novo sub-otimizador no processo de estimação dos hiperparâmetros da função kernel, de modo a contornar o problema de convergência observado. Por fim, desejamos ainda analisar o método proposto sobre os dados não transformados, sendo mantida a estrutura de multinormalidade sobre os mesmo.

Além disso, seria interesse trabalhar sobre um modelo de regressão em duas etapas. Ao invés de utilizarmos as áreas transversais das barras como covariáveis para resposta c_{sum}, a primeira etapa utilizaríamos essas covariáveis para prever as violações através de um modelo de regressão multivariado. Na segunda parte, após prever os valores de violações, aplicaríamos uma regressão linear múltipla sobre esses resultados para, finalmente, predizer o valor de c_{sum}. Em casos onde o número de restrições é muito grande, poderíamos realizar ainda uma análise fatorial ou testes estatísticos que verifiquem a relevância de cada restrição, afim de reduzir o seu número para as predições realizadas.

De maneira geral, apesar de não corresponder as nossas expectativas, o trabalho propõe uma nova ferramenta de conhecimento interdisciplinar sobre duas importantes áreas científicas, contribuindo em novas pesquisas, para ambas as áreas, que visem o seu aprimoramento e novas aplicações. É importante ressaltar também sobre o tempo da pesquisa realizada. Todo esse trabalho é fruto de apenas 6 meses de estudos. No entanto, nos 2 últimos meses, devido a inviabilidade da aplicação, o modelo que havíamos proposto (regressão *kriging*) foi mudado para a regressão via processos Gaussianos, tratado neste trabalho. Isso mostra que, sobre pouco tempo de modelagem, ainda sim obtivemos resultados consistentes (mesmo que não satisfatórios), tendo altas chances de se sobressaírem quando os problemas identificados forem trabalhados.

REFERÊNCIAS

[1] ANDERSON, T. W. *An Introduction to Multivariate Statistical Analysis* (3rd ed.). John Wiley, 2003.

[2] ARAUJO, R. L.; BARBOSA, H. J. C.; BERNARDINO, H. S. (2016) Evolução Diferencial para Problemas de Otimização com Restrições Lineares. *Tese de Mestrado em Modelagem Computacional*, Universidade Federal de Juiz de Fora. Páginas 25-29.

[3] CASELLA, G.; BERGER, R. *Inferência Estatística* (1ª ed.). Cengage Learning, 2010.

[4] COELHO, C. A. C. (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering, ELSEVIER*, **v. 191**. Páginas 1245-1287.

[5] CHARNET, T.; FREIRE, C. A. L.; CHARNET, E. M. R., BONVINO, H. *Análise de modelos de regressão linear - Com aplicações* (2rd ed.). Unicamp, 2015.

[6] DAVIS, R. A. (2014) Gaussian Process: Theory. *Wiley StatsRef: Statistics Reference Online*. `https://doi.org/10.1002/9781118445112.stat07472`.

[7] DEB, K. (2000) An Efficient Constrait Handlinb Method for Generic Algorithms. *Comput. Methods Appl. Mech. Engrg., ELSEVIER*, **v.186**. Páginas 311-338.

[8] DO, C. B. (2007) Gaussian Processes. *Stanford University, Stanford, CA*. `https://see.stanford.edu/materials/aimlcs229/cs229-gp.pdf` (acesso: novembro/2018).

[9] DUVENAUD, D. K. (2014) Automatic Model Construction with Gaussian Processes. *Doctoral Thesis, University of Cambridge*. Páginas 08-30. `https://www.cs.toronto.edu/~duvenaud/thesis.pdf` (acesso: novembro/2018).

[10] FONSECA, L. A. (2009) Algoritmos Genéticos Assistidos por Metamodelos Baseados em Similaridade. *Tese de doutorado, Laboratório Nacional de Computação Científica (LNCC)*. Páginas 25-45.

[11] GIBBONS, J. D.; CHAKRABORTI, S. *Nonparametric Statistical Inference* (4rd ed.). Marcel Dekker, 2003.

[12] JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. *An Introduction to Statistical Learning: with Applications in R* (7rd ed.). Springer , 2017.

[13] JOHNSON, R. A.; WICHER, D. W. *Applied Multivariate Statistical Analysis*. Pearson, 2007. Páginas 149-175.

[14] KREMPSER, E.; BERNARDINO, H. S., BARBOSA, H. J. C.; LEMONGE, A. C. C. (2017) Performance evaluation of local surrogate models in differential evolution-based optimum design of truss structures. *Engineering Computations, EmeraldPublishingLimited*, **v.34**(N0. 2).

[15] MARDIA, K. V. (1970) Measures of multivariate skewness and kurtosis with applications. *Biometrika*, **v57** (March). Páginas 519–530.

[16] MENDONÇA, M. W. (2005) O Método L-BFGS com Fatoração Incompleta para a Resolução de Problemas de Minimização. *Tese de mestrado, Pós-Graduação em Matemática e Computação Científica, Universidade Federal de Santa Catarina.* `http://mtm.ufsc.br/~melissa/arquivos/dissertacao_mestrado.pdf`.

[17] NOCEDAL, J.; WRIGHT, S. J.*Numerical Optimization* Springer Series In Operations Research,Springer, 1999.

[18] PRAVEEN, C.; DUVIGNEAU, R. (2007) Radial basis functions and kriging metamodels for aerodynamic optimization. *Relatório técnico, INRIA*.

[19] Python Software Foundation. Python Language Reference, version 3.6. Available at `http://www.python.org`

[20] ROSS, S. *Probabilidade: Um Curso Moderno com Aplicações* (8rd ed.) Bookman, 2010.Páginas 89-106.

[21] PEDREGOSA, F. at. all (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**. Páginas 2825-2830.

[22] SILVA, D. G. C.; PESSOA, P. L. N. (1998) Análise de dados amostrais complexos.. *IBGE*.

[23] SOHRAB,H. H. *Basic Real Analysis* (2rd ed.).Birkhäuser Boston, 2003.

[24] SU, G.; PENG, L.; HU, L. (2017) A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Structural Safety, ELSEVIER*, **v.68**. Páginas 97-109.

[25] RASMUSSEN, C. E.; WILLIAMS, C. K. I. *Gaussian Processes for Machine Learning* (2rd ed.). The MIT Press, 2006.