



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ESTRUTURA E CONSTRUÇÃO CIVIL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL: ESTRUTURAS E
CONSTRUÇÃO CIVIL

JORGE ARTUR FRANÇA DE MENDONÇA

APLICAÇÃO DE MODELOS SUBSTITUTOS BASEADOS EM REDES NEURAIAS
ARTIFICIAIS NA OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS

FORTALEZA

2017

JORGE ARTUR FRANÇA DE MENDONÇA

APLICAÇÃO DE MODELOS SUBSTITUTOS BASEADOS EM REDES NEURAIIS
ARTIFICIAIS NA OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia Civil. Área de concentração: Estruturas.

Orientador: Prof. Dr. Evandro Parente Junior
Coorientador: Prof. Dr. Antônio Macário
Cartaxo de Melo

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- M495a Mendonça, Jorge Artur França de.
Aplicação de modelos substitutos baseados em redes neurais artificiais na otimização de estruturas laminadas / Jorge Artur França de Mendonça. – 2017.
98 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Civil: Estruturas e Construção Civil, Fortaleza, 2017.
Orientação: Prof. Dr. Evandro Parente Júnior.
Coorientação: Prof. Dr. Antônio Macário Cartaxo de Melo.
1. Otimização Estrutural. 2. Modelos Substitutos. 3. Redes Neurais Artificiais. 4. Funções de Bases Radiais. 5. Materiais Compósitos. I. Título.

CDD 624.1

JORGE ARTUR FRANÇA DE MENDONÇA

APLICAÇÃO DE MODELOS SUBSTITUTOS BASEADOS EM REDES NEURAIIS
ARTIFICIAIS NA OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Civil. Área de concentração: Estruturas.

Aprovada em: 31/ 08/ 2017.

BANCA EXAMINADORA

Prof. Dr. Evandro Parente Junior (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antônio Macário Cartaxo de Melo (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Batista Marques de Sousa Júnior
Universidade Federal do Ceará (UFC)

Prof. Dr. Anselmo Cardoso de Paiva
Universidade Federal do Maranhão (UFMA)

À minha mãe, Maria José Carvalho de França.

AGRADECIMENTOS

Antes de mais nada, agradeço à vida. Se não fosse por ela eu jamais teria feito algo desse tipo.

À minha mãe, Maria José Carvalho de França, pelo esforço contínuo e dedicação.

À minha namorada, Leticia Alexandre Lima, pelo companheirismo, amizade e amor.

A meu orientador Evandro Parente Junior, pela paciência de Buda, pelos inúmeros ensinamentos e compreensão.

Ao meu coorientador Antônio Macário Cartaxo de Melo, pelas contribuições pertinentes.

Aos professores Anselmo Paiva e João Batista por aceitarem participar do meu julgamento final e pelas contribuições sugeridas.

Aos colegas do LMCV, em especial a Elias Barroso, que contribuiu lindamente nos códigos do BIOS.

Aos colegas da turma, Jovyntino, Michelle, Eric e Savanna. We have a dream!

“Onde há dúvida, há liberdade.”

- Provérbio Latino

RESUMO

O campo da otimização estrutural vem ganhando importância devido à busca por projetos mais eficientes. Contudo, o custo computacional para resolver um problema de otimização pode ser muito elevado, principalmente quando se trabalha com estruturas sem solução analítica, sendo necessário uma análise por métodos numéricos como o Método dos Elementos Finitos (MEF) ou Análise Isogeométrica (AIG). Assim, novas técnicas estão sendo desenvolvidas, como é o caso dos modelos substitutos, para redução do custo de processamento. Os modelos substitutos baseiam-se em técnicas estatísticas e computacionais para substituir os métodos numéricos na avaliação das respostas estruturais. Uma técnica bastante utilizada são as Redes Neurais Artificiais (RNA), pela capacidade de trabalhar com problemas não lineares. O objetivo do presente trabalho é estudar a aplicação de Redes de Múltiplas Camadas (*Multilayer Perceptron* - MLP) e Redes de Funções de Base Radial como modelo substitutos, avaliando a influência de cada técnica no tempo de otimização e a capacidade de reconhecer e simular o comportamento de um software de análise numérica de estruturas laminadas. O primeiro problema foi o de maximização da carga crítica, cuja solução analítica é conhecida. Com esse problema foi possível identificar como as técnicas de amostragem (i.e. Hipercubo latino, Hammersley e Hipercubo latino otimizado) e o tamanho da amostra influenciaram na qualidade do modelo substituto. Em seguida, considerou-se também a otimização de uma casca cilíndrica laminada com objetivo de minimizar os seus deslocamentos. Neste caso, os modelos substitutos foram treinados utilizando resultados obtidos pela Análise Isogeométrica. Pode-se notar, para problemas de baixa complexidade, que as técnicas de amostragem pouco influenciaram, pois todas chegaram a bons resultados. Além disso, as redes RNA e RBF por interpolação apresentaram melhores resultados para um problema de alta complexidade. Com base nos treinos, observou-se que a RBF por interpolação foi mais vantajosa, visto que o tempo de treinamento era menor. Quanto ao número de pontos amostrais, a relação fornecida por Amouzgar e Strömberg (2016) foi suficiente para os problemas estudados. O Hipercubo latino otimizado apresentou os melhores resultados para os casos estudados. Finalmente, verificou-se que o uso de modelos substitutos possibilitou uma redução significativa no tempo de processamento.

Palavras-chave: Otimização Estrutural, Modelos Substitutos, Redes Neurais Artificiais, Funções de Bases Radiais, Materiais Compósitos.

ABSTRACT

The field of structural optimization is becoming increasingly present in engineering offices in the search for more efficient designs. However, the computational cost to perform an optimization problem can be very high, especially when dealing with complex structures without analytical solution, where numerical methods such as the Finite Element Method (MEF) or Isogeometric Analysis (IGA) are necessary. Thus, new approaches have been developed, to reduce the cost of processing. Surrogate models are based on statistical and computational techniques to replace costly numerical analyses by efficient and accurate approximate solutions. One well-known and applied technique is the Artificial Neural Networks (RNA), since they can handle nonlinear problems. The objective of the present work is to study the application of Multilayer Perceptron (MLP) Networks and Radial Basis Functions (RBF) Networks to replace the numerical analysis methods, evaluating the influence of each technique on the optimization time and the ability to recognize and simulate the behavior of a software of analysis of laminated structures. The first problem was that of maximizing the critical load, whose analytical solution is known. With this problem, it was possible to identify how sampling techniques (i.e. Latin Hypercube, Hammersley and Optimized Latin Hypercube) and the sample size influenced the accuracy of the surrogate model. Then, the optimization of a laminated cylindrical shell with the objective of minimizing its displacements was also considered. In this case, the surrogate models were trained using results obtained by Isogeometric Analysis (IGA). It can be noticed, that the sampling techniques had little influence for low complexity problems, since all methods lead to good results. In addition, the MLP and RBF networks by interpolation presented better results for a highly complex problem. Based on the training, it was observed that the RBF by interpolation was more advantageous, since the training time was shorter. As for the number of sampling points, the relationship provided by Amouzgar and Strömberg (2016) was sufficient for the problems studied. Optimized Latin Hypercube presented the best results for the cases studied. Finally, use of surrogate models allowed a significant reduction in processing time.

Keywords: Structural Optimization, Surrogate Models, Artificial Neural Networks, Radial Basis Functions, Composite Materials.

LISTA DE FIGURAS

Figura 1 – Tipo de materiais compósitos.....	19
Figura 2 – Sistema global (x, y, z) e local (x_1, x_2, x_3)	20
Figura 3 – Esquema de laminação.....	22
Figura 4 – Tipos de laminação.....	23
Figura 5 – Representação dos parâmetros de resistência.	26
Figura 6 – Envoltória de falha para um estado biaxial.....	26
Figura 7 – Envoltória de falha para o critério de máxima deformação.	27
Figura 8 – Fluxograma para determinação do Sf.	29
Figura 9 – Estrutura de um Algoritmo Genético.	32
Figura 10 – Sistema de codificação e decodificação.....	33
Figura 11 – Mapeamento da função objetivo.	36
Figura 12 – Distribuição de probabilidade de cada indivíduo para roleta (a) e ranking (b).....	37
Figura 13 – processo de cruzamento de um laminado com duas camadas.....	38
Figura 14 – Mutação no indivíduo gerado pelo cruzamento.....	39
Figura 15 – operador troca de lâminas.	39
Figura 16 – Operadores de eliminação e adição.....	40
Figura 17 – Estrutura de um algoritmo PSO.	41
Figura 18 – Topologia do PSO.	42
Figura 19 – Diagrama do aprendizado supervisionado.	44
Figura 20 – Estrutura da criação de um modelo substituto.	46
Figura 21 – Amostragem pelo experimento fatorial completo.....	49
Figura 22 – Amostra gerada de forma aleatória.	51
Figura 23 – Amostra obtida por Hipercubo Latino.	52
Figura 24 – Amostragem pela Sequência de Hammersley.....	53
Figura 25 – Representação de um neurônio biológico.	55
Figura 26 – Modelo de um neurônio artificial.....	56
Figura 27 – Representação das funções de ativação.....	57
Figura 28 – Arquitetura de redes neurais alimentadas adiante (Feedforward).....	57
Figura 29 – Rede neural com arquitetura recorrente.	58
Figura 30 – Aproximando de uma função senoidal usando RBF.....	61
Figura 31 – Funções de base radial.	62
Figura 32 – Função Gaussiana para diferentes valores de c	63
Figura 33 – Estrutura de uma rede RBF.	65

Figura 34 – Nova estrutura de uma rede RBF.	65
Figura 35 – Modelo de neurônio de uma RNA.	66
Figura 36 – Código MATLAB para construção de uma RNA.	67
Figura 37 – Estrutura de um neurônio para RBFN.....	68
Figura 38 – Comportamento da função <i>radbas(n)</i>	68
Figura 39 – Influência do <i>SPREAD</i> na função <i>radbas()</i>	70
Figura 40 – Arquitetura de uma RRBF.....	70
Figura 41 – Comandos para construção de uma RBFN.	71
Figura 42 – Comandos para construção de uma RBFN com variação de neurônios.	71
Figura 43 – Parada precoce de uma rede.....	72
Figura 44 – Placa laminada com carregamento biaxial.....	74
Figura 45 – Valor comparativo dos erros percentuais para a RNA.	77
Figura 46 – Comparativo entre os erros percentuais para a RBFi.....	78
Figura 47 – Comparativo entre os erros percentuais para a RBFr.....	81
Figura 48 – Geometria e carregamento da casca.....	87
Figura 49 – performance das redes.....	89

LISTA DE TABELAS

Tabela 1 – Probabilidade de seleção por roleta.	36
Tabela 2 – Probabilidade de seleção por Ranking.	37
Tabela 3 – Experimento fatorial $k=n=2$ e $N=2^2=4$	49
Tabela 4 – Propriedades geométricas e do material (grafite-epóxi).	74
Tabela 5 – Valores do NRMSE da RNA.	76
Tabela 6 – Resultado para AG + RNA.	77
Tabela 7 – Valores do NRMSE para a RBFi.	78
Tabela 8 – Resultado para AG + RBFi.	79
Tabela 9 – Valores do NRMSE para a RBFr.	80
Tabela 10 – Resultado para AG + RBFr.	81
Tabela 11 – Resultado do treinamento para a RNA.	82
Tabela 12 – Resultado da otimização RNA.	83
Tabela 13 – Resultados do treinamento da rede RBFi.	83
Tabela 14 – Resultados da otimização da rede RBFi.	84
Tabela 15 – Resultados do treinamento da rede RBFr.	85
Tabela 16 – Resultados da otimização da rede RBFr.	85
Tabela 17 – Propriedades elásticas do material.	87
Tabela 18 – Resultado de treinamento da RNA.	89
Tabela 19 – Otimização das RNAs.	90
Tabela 20 – Comparativo entre resultado do FAST e da rede para a laminação aproximada. .	90
Tabela 21 – Resultados da otimização da rede RBFi.	90
Tabela 22 – Resultados da otimização para rede RBFr.	91

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo	17
1.2	Organização do trabalho	17
2	MATERIAIS COMPÓSITOS	19
2.1	Comportamento mecânica de uma lâmina	19
2.1.1	<i>Comportamento macromecânico</i>	20
2.2	Teoria de laminação	22
2.3	Teoria Clássica da Laminação	23
2.4	Critério de falha	25
3	OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS	30
3.1	Algoritmos Bioinspirados	31
3.1.1	<i>Algoritmos Genéticos</i>	31
3.1.1.1	<i>Codificação</i>	33
3.1.1.2	<i>Tratamento das restrições</i>	34
3.1.1.3	<i>Função aptidão e Seleção</i>	35
3.1.1.4	<i>Cruzamento</i>	38
3.1.1.5	<i>Mutação</i>	38
3.1.1.6	<i>Operadores específicos para laminados</i>	39
3.1.2	<i>Otimização por Nuvem de Partículas</i>	40
4	MODELOS SUBSTITUTOS	43
4.1	Parâmetros de desempenho	45
4.2	Construindo um modelo substituto	46
4.3	Projeto de Experimento	47
4.3.1	<i>Amostragem Aleatória</i>	50
4.3.2	<i>Hipercubo Latino</i>	51
4.3.3	<i>Sequências de baixa discrepância</i>	52
4.4	Redes Neurais Artificiais	53
4.4.1	<i>Histórico</i>	54
4.4.2	<i>Neurônios biológicos x Neurônios Artificiais</i>	55
4.4.3	<i>Arquitetura de uma RNA</i>	57
4.4.4	<i>Processo de aprendizado de um perceptron simples</i>	58
4.4.5	<i>Processo de aprendizado de uma rede MLP</i>	59
4.5	Funções de Base Radial	60

4.5.1	Redes de funções de bases radiais.....	64
4.6	Implementação computacional	66
5	EXEMPLOS.....	73
5.1	Placa laminada sob compressão biaxial no plano.....	73
5.1.1	<i>Placa com 8 lâminas.....</i>	<i>75</i>
5.1.2	<i>Placa com 40 lâminas.....</i>	<i>82</i>
5.1.3	<i>Discussão</i>	<i>86</i>
5.2	Minimização do deslocamento de uma casca cilíndrica.....	87
5.2.1	<i>Comentários.....</i>	<i>91</i>
6	CONCLUSÃO	92
	REFERÊNCIAS.....	94

1 INTRODUÇÃO

Um material compósito é constituído de dois ou mais materiais, e suas características são, em geral, melhores que as características de seus constituintes. Por esse motivo, o uso de compósitos em estruturas se tornou comum, principalmente em estruturas que precisam de alta resistência e peso reduzido (HAFTKA; GÜRDAL, 1992). Existem diversos tipos de materiais compósitos, porém neste trabalho foram utilizados os compósitos reforçados por fibras unidirecionais.

Estes compósitos são normalmente utilizados na forma de laminados constituídos por um conjunto de lâminas de material compósito empilhadas em uma determinada sequência. Estes laminados são utilizados na fabricação de estruturas de parede fina, como placas, cascas, vigas e colunas.

O projeto de uma estrutura laminada deve considerar a espessura das lâminas, a orientação das fibras e o material de cada lâmina. Essa combinação dá aos compósitos uma grande flexibilidade de adaptação a diferentes aplicações estruturais. Porém devido à natureza combinatória, esse tipo de projeto só se torna viável por meio da otimização.

No caso de laminados, as variáveis de projeto geralmente são discretas, sendo necessário o uso de algoritmos bioinspirados, como é o caso dos Algoritmos Genéticos e da Otimização por Nuvens de Partículas. Tais algoritmos são métodos estocásticos capazes de encontrar o ótimo global de um problema com múltiplos ótimos locais, além de serem capazes de resolver problemas de otimização com variáveis discretas e funções descontínuas.

Os Algoritmos Genéticos (AG) foram desenvolvidos com base na teoria da seleção natural de Charles Darwin, onde o mais apto sobrevive e deixa descendentes. A ideia básica é iniciar com um conjunto de dados (população) gerados aleatoriamente e avaliar cada um desses pontos (indivíduos) na busca por características positivas (aptidão) que serão passadas adiante. A cada geração, novos indivíduos são criados através da combinação (cruzamento) de indivíduos selecionados, onde os mais aptos têm maior probabilidade de seleção.

A Otimização por Nuvens de Partículas (*Particle Swarm Optimization* - PSO), desenvolvida por Kennedy e Eberhart (1995), simula o comportamento de grupos de animais (e.g. bando de aves ou cardumes de peixes) em busca de alimento. Neste processo, o movimento das partículas é guiado pelo aprendizado individual e do grupo.

O uso dessas técnicas nas mais diversas aplicações, mostra a robustez desses algoritmos (REID, 1995; KONA *et al.*, 2006; HERATH *et al.*, 2013; ROCHA; PARENTE; MELO, 2014 e LIU; WANG, 2014).

Apesar dos algoritmos bioinspirados conseguirem encontrar com certa facilidade o ótimo global, o custo computacional associado a esses algoritmos é bastante elevado, pois requerem um grande número de avaliações da função objetivo e restrições. Este número é proporcional ao número de indivíduos (ou partículas) e de geração (ou iterações).

Em problemas de otimização estrutural, a avaliação de cada indivíduo ou partícula envolve a realização de uma análise estrutural, o que tipicamente é feito pelo Método dos Elementos Finitos (MEF) ou outro método computacional, como a Análise Isogeométrica (AIG) (Barroso, 2015). A AIG é caracterizada pelo uso de funções do tipo B-Splines e NURBS, para representar a geometria da estrutura. Isso significa que essa geometria será modelada exatamente, e não de forma aproximada como ocorre no MEF.

O problema está na combinação dos algoritmos evolutivos com as análises numéricas, pois essa combinação torna o processo computacional oneroso. Uma forma de contorná-lo é substituindo o modelo de alta fidelidade baseado no MEF ou AIG por um modelo mais simples (modelo de baixa fidelidade) e computacionalmente mais eficiente. Estes modelos são chamados de modelos substitutos (*surrogate models*) ou ainda, metamodelos (BARTHELEMY; HAFTKA, 1993; FORRESTER; KEANE, 2009).

Um modelo substituto pode ser utilizado para reconhecimento de dados, classificação e aproximação de funções. Este trabalho se deteve ao último caso, pois é necessário que o modelo seja capaz de aproximar o comportamento de uma função analítica ou de um método de análise numérica, como é o caso da substituição do MEF ou AIG por modelos substitutos na otimização.

O primeiro passo que deve ser dado para a construção do modelo substituto é a geração de amostras usadas no treinamento. Aqui, as amostras foram geradas por meio do Hipercubo Latino, Sequência de Hammersley e o Hipercubo Latino Otimizado (FORRESTER; SÓBESTER; KEANE, 2008; HAMMERSLEY, 2006; KALAGNANAM; DIWEKAR, 1997; VIANA; VENTER; BALABANOV, 2010). Estas três técnicas foram utilizadas com o intuito de estudar como o tipo de amostragem influencia na resposta final.

O treinamento de um modelo substituto pode ser, de modo geral, supervisionado ou não-supervisionado. O primeiro apresenta um conjunto de dados (entrada-saída), que representa o ambiente a ser substituído, usado no treinamento do metamodelo. No segundo caso, não há um conjunto de entrada-saída para ser usado pelo metamodelo (HAYKIN, 2001). Dessa forma, optou-se por usar o aprendizado supervisionado.

Neste trabalho, será estudado o uso de Redes Neurais Artificiais e Redes de Funções de Bases Radiais como modelos substitutos. Quanto à arquitetura, a rede neural pode ser do

tipo perceptron simples ou perceptron multicamadas (*Multlayer Perceptron* - MLP). A primeira limita-se aos problemas linearmente separáveis, enquanto a segunda pode ser usada em qualquer caso, tornando-se a melhor solução para os problemas não-lineares. Uma rede de funções de base radial, diferencia-se da MLP em alguns pontos, pois possui apenas uma camada escondida e as funções dessa camada são de bases radial.

1.1 Objetivo

Devido ao alto custo computacional gerado pelo processo de otimização por algoritmos bioinspirados combinado com análise numérica, o objetivo geral deste trabalho é estudar a aplicabilidade de modelos substitutos na otimização de estruturas laminadas para reduzir esse tempo de processamento, por meio da substituição dos modelos numéricos por técnicas mais eficientes para avaliação do comportamento estrutural.

Para atingir o objetivo do trabalho se faz necessária a realização de alguns objetivos específicos:

- a) Estudar os efeitos do tamanho da amostra e do método de geração da amostra na qualidade do modelo substituto.
- b) Comparar os modelos substitutos criados utilizando Redes Neurais Artificiais do tipo MLP e Redes de Funções de Bases Radiais.
- c) Otimizar estruturas de material compósito utilizando AIG combinado com algoritmos bioinspirados (PSO e AG).
- d) Comparar a eficiência e precisão da otimização utilizando diferentes modelos substitutos.

1.2 Organização do trabalho

A dissertação está organizada em 6 capítulos. O Capítulo 2 apresenta os conceitos básicos relacionados aos materiais compósitos laminados, incluindo as características necessárias à otimização.

O Capítulo 3 aborda do processo de otimização, apresentando conceitos referentes aos algoritmos genéticos e enxame de partículas, incluindo suas implementações no programa BIOS.

O Capítulo 4 trata dos modelos substitutos, apresentando as técnicas necessárias para geração de amostras para treinamento, criação de um modelo substituto, treinamento, e um aprofundamento em Redes Neurais Artificiais e Redes de Funções de Bases Radiais.

O Capítulo 5 apresenta os resultados obtidos na otimização de uma placa cuja solução analítica é conhecida e de uma casca cilíndrica onde é necessária uma análise numérica.

Por fim, o Capítulo 6 traz as conclusões e comentários a respeito dos resultados obtidos no Cap. 5, bem como sugestões para trabalhos futuros.

2 MATERIAIS COMPÓSITOS

Materiais compósitos são formados pela união de dois ou mais materiais, em escala macroscópica, formando um novo material. Em geral, o compósito possui características melhores do que os seus constituintes considerados isoladamente (JONES, 1999)

Segundo Reddy (2004), muitos compósitos construídos pelo homem possuem duas fases características e distintas: a matriz e a fibra. Sendo a primeira utilizada como material de base e a segunda utilizada para material de reforço, ou seja, elemento que melhora o comportamento mecânico do material base. Os materiais em discussão são classificados, por Reddy (2004), conforme a combinação entre os componentes, em: Compósitos Fibrosos; Compósitos Particulados e Compósitos Laminados. O primeiro tipo é caracterizado pela presença de fibras como material resistente, as fibras de carbono e vidro são as mais utilizadas. O segundo, é caracterizado pela presença de partículas como material resistente (e.g. concreto) e o terceiro, o laminado, é caracterizado pela união de lâminas de diferentes materiais. Esta classificação é apresentada na Figura 1.

Figura 1 – Tipo de materiais compósitos.



Fonte: Adaptado de Gürdal, Haftka e Hajela (1998).

Este trabalho tem foco nos compósitos laminados unidirecionais, aqueles nos quais as fibras de uma lâmina são paralelas entre si e dispostas em uma única direção, resultando em um material ortotrópico com maior resistência e rigidez na direção das fibras. As seções seguintes abordam o comportamento mecânico da lâmina e do laminado.

2.1 Comportamento mecânica de uma lâmina

O estudo do comportamento mecânico de uma lâmina, estrutura básica do laminado, baseia-se em duas hipóteses (REDDY, 2004): a lâmina é contínua, isto é, não há espaços vazios e seu comportamento atende à lei de Hooke, antes da falha do material.

Devido à natureza heterogênea, o material compósito pode ser estudado por meio de duas abordagens (JONES, 1999):

- A abordagem *micromecânica* estuda os materiais, fibra e matriz, isoladamente, e avalia a interação entre eles.
- A abordagem *macromecânica* considera a combinação dos materiais da lâmina de maneira homogênea. Essa abordagem foi a utilizada no presente trabalho.

2.1.1 Comportamento macromecânico

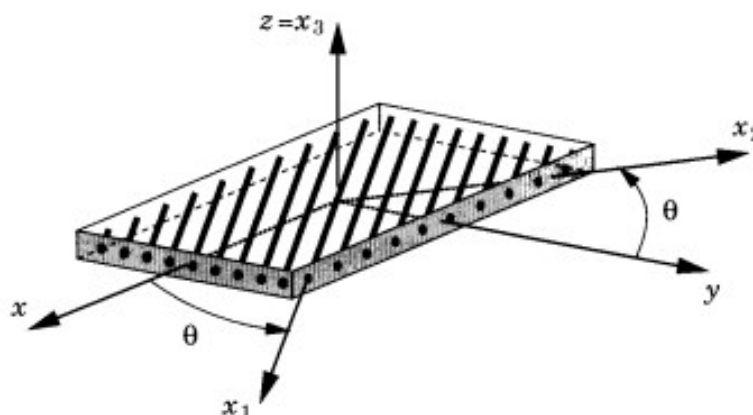
Uma das hipóteses de análise é de que o material possui comportamento linear elástico, portanto, pode-se aplicar a lei de Hooke generalizada, dada por:

$$\sigma_i = Q_{ij}\varepsilon_j, \quad (i,j = 1,2,\dots,6). \quad (1)$$

onde σ_i são as tensões na lamina, Q_{ij} é a matriz constitutiva da lâmina no sistema local, e ε_j são as deformações na lâmina.

Em laminados com fibras unidirecionais, cada lâmina apresenta um comportamento ortotrópico, isso significa que o material da lâmina possui propriedades diferentes nas três direções mutuamente perpendiculares (JONES, 1999). Neste caso, as propriedades do material são definidas em um sistema de coordenadas local (x_1, x_2, x_3), sendo o eixo x_1 na direção das fibras, x_2 transversal às fibras no plano da lâmina e x_3 perpendicular ao plano da lâmina, como mostrado na Figura 2.

Figura 2 – Sistema global (x, y, z) e local (x_1, x_2, x_3).



Fonte: Reddy (2004).

Dessa forma, a Equação (1) pode ser representada da seguinte maneira:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & 0 & 0 & 0 \\ Q_{12} & Q_{22} & Q_{23} & 0 & 0 & 0 \\ Q_{13} & Q_{23} & Q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} \Rightarrow \boldsymbol{\sigma}_1 = \mathbf{Q}\boldsymbol{\varepsilon}_1 \quad (2)$$

sendo os coeficientes da matriz \mathbf{Q} definidos a partir das propriedades mecânicas do material compósito

$$\begin{aligned} Q_{11} &= E_1 \frac{1 - \nu_{23}\nu_{32}}{\Delta} & Q_{12} &= E_1 \frac{\nu_{21} - \nu_{31}\nu_{23}}{\Delta} & Q_{13} &= E_1 \frac{\nu_{31} - \nu_{21}\nu_{32}}{\Delta} \\ Q_{22} &= E_2 \frac{1 - \nu_{13}\nu_{31}}{\Delta} & Q_{23} &= E_2 \frac{\nu_{32} - \nu_{12}\nu_{31}}{\Delta} & Q_{33} &= E_3 \frac{1 - \nu_{12}\nu_{21}}{\Delta} \\ Q_{66} &= G_{12} & Q_{44} &= G_{13} & Q_{55} &= G_{23} \end{aligned} \quad (3)$$

$$\Delta = 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}$$

As equações apresentadas até agora representam o material no sistema da lâmina, também conhecido como sistema local ou do material, porém as equações de equilíbrio e relações cinemáticas do laminado são escritas no sistema global (x, y, z). Sendo assim, se faz necessário uma lei de transformação que leve as equações constitutivas de um sistema para outro. De acordo com Cook *et al.* (2002), a transformação das deformações pode ser realizada como

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & 0 & \sin \theta \cos \theta & 0 & 0 \\ \sin^2 \theta & \cos^2 \theta & 0 & -\sin \theta \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -2\sin \theta \cos \theta & 2\sin \theta \cos \theta & 0 & \cos^2 \theta - \sin^2 \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos \theta & -\sin \theta \\ 0 & 0 & 0 & 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} \Rightarrow \boldsymbol{\varepsilon}_1 = \mathbf{T}\boldsymbol{\varepsilon} \quad (4)$$

onde \mathbf{T} é a matriz de transformação. Através do Princípio dos Trabalhos Virtuais (PTV) é possível obter a transformações das tensões, expressa por:

$$\boldsymbol{\sigma} = \mathbf{T}^T \boldsymbol{\sigma}_1 \quad (5)$$

Finalmente, substituindo a Equação (2) e a Equação (4) na Equação (5), chega-se a relação constitutiva no sistema global:

$$\boldsymbol{\sigma} = \mathbf{T}^T \mathbf{Q} \mathbf{T} \boldsymbol{\varepsilon} \Rightarrow \boldsymbol{\sigma} = \bar{\mathbf{Q}} \boldsymbol{\varepsilon} \quad (6)$$

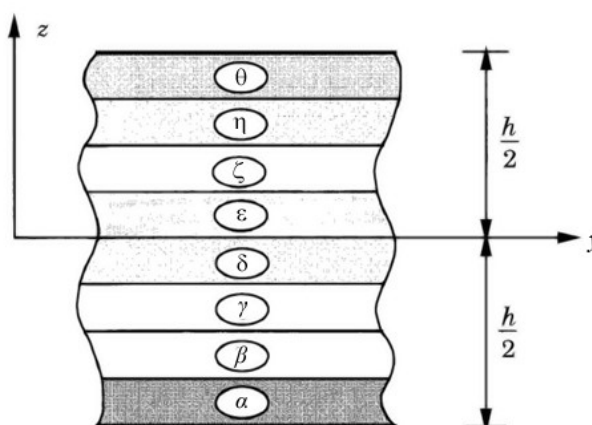
2.2 Teoria de laminação

Segundo Ghugal e Shimpi (2002), as teorias de laminação são classificadas em duas teorias gerais. A primeira contempla as teorias de lâmina equivalente (*Equivalent Single Layer*), o laminado é tratado como uma lâmina equivalente. Essas teorias são desenvolvidas pela expansão do campo de deslocamento. A segunda são as teorias de lâminas discretas (*layer-wise*), onde cada lâmina é tratada individualmente, conseguindo representar o campo de deslocamento com maior acurácia.

Entre as diversas teorias existentes na literatura, foi considerado, para o desenvolvimento do trabalho, o uso da Teoria Clássica de Laminação (TCL). A TCL pertence às teorias de lâmina equivalente, sendo bastante eficiente do ponto de vista computacional.

O comportamento mecânico de estruturas laminadas é determinado pelo o esquema de laminação (*layup*). No caso de laminados com apenas um material, esse esquema é definido pela sequência da orientação das camadas, na forma ($\alpha/\beta/\gamma/\delta/\varepsilon/\dots$), onde α é a orientação da primeira camada, β é a orientação da segunda camada e assim sucessivamente (REDDY, 2004). A Figura 3 apresenta um esquema de laminação.

Figura 3 – Esquema de laminação.

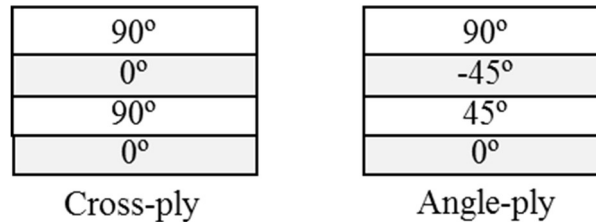


Fonte: Adaptado de Reddy (2004).

O esquema de laminação pode ser do tipo *cross-ply* ou *angle-ply*. A diferença entre os dois está na orientação das lâminas. No primeiro caso, a orientação da camada só aceita dois valores: 0° ou 90° . No segundo caso, *angle-ply*, outros ângulos podem ser adotados, como é o caso de camadas com 45° . A

Figura 4 apresenta esses dois tipos de combinação.

Figura 4 – Tipos de laminação.



Fonte: Elaborada pelo autor.

A laminação ainda pode ser classificada quanto a simetria dos ângulos, em:

- i) Laminados simétricos: Possuem simetria de material, espessura e orientação das fibras em relação ao plano médio, ex. $[90^\circ/0^\circ/0^\circ/90^\circ]_s$ equivale ao seguinte laminado $[90^\circ/0^\circ/0^\circ/90^\circ/90^\circ/0^\circ/0^\circ/90^\circ]$. O índice s representa a simetria do laminado.
- ii) Laminado balanceado: Nesse esquema, cada lâmina, com orientação, α , diferente de 0° e 90° , possui uma correspondente, $-\alpha$. Ex. $[+45^\circ/-45^\circ/0/90]$.
- iii) Laminado antissimétrico: São os que possuem simetria apenas em relação aos materiais e espessuras, mas com orientações das lâminas opostas em relação à superfície média. Ex. o laminado $[45^\circ/-45^\circ]_3$, equivalente a $[45^\circ/-45^\circ/45^\circ/-45^\circ/45^\circ/-45^\circ]$, é antissimétrico.
- iv) Laminados assimétricos: É o caso geral de laminação, onde não existe simetria em relação à superfície média.

2.3 Teoria Clássica da Laminação

A Teoria Clássica de Laminação (TCL) baseia-se na teoria de placas de Kirchhoff que leva em consideração as seguintes hipóteses (REDDY, 2004):

- i) As linhas normais transversais permanecem transversais após a deformação;
- ii) As linhas normais transversais são inextensíveis e
- iii) As linhas normais transversais rotacionam de modo que permaneçam perpendiculares à superfície média.

Assim, as deformações $\varepsilon_z, \gamma_{xz}$ e γ_{yz} são desprezadas. Com bases nessas informações é possível reescrever a Equação (2) de maneira simplificada, considerando a relação tensão-deformação no plano:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} \Rightarrow \boldsymbol{\sigma}_1 = \mathbf{Q}\boldsymbol{\varepsilon}_1 \quad (7)$$

onde os termos da matriz \mathbf{Q} são expressos em função do coeficiente de Poisson, do módulo de elasticidade longitudinal e do módulo de elasticidade transversal, da seguinte forma:

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}; Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}}; Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}; Q_{66} = G_{12} \quad (8)$$

Assim como nas Equações (5) e (6), é preciso aplicar uma matriz de transformação

T. Para uma lâmina qualquer, com orientação θ , esta matriz é dada por:

$$\mathbf{T} = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & \sin\theta\cos\theta \\ \sin^2 \theta & \cos^2 \theta & -\sin\theta\cos\theta \\ -2\sin\theta\cos\theta & 2\sin\theta\cos\theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix}. \quad (9)$$

Desse modo, a matriz constitutiva de cada lâmina no sistema global será obtida aplicando a Equação (6), onde os coeficientes da matriz constitutiva transformada são expressos na forma:

$$\begin{aligned} \bar{Q}_{11} &= Q_{11} \cos^4 \theta + (Q_{12} + 2Q_{66}) 2\sin^2 \theta \cos^2 \theta + Q_{22} \sin^4 \theta \\ \bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{12} (\sin^4 \theta + \cos^4 \theta) \\ \bar{Q}_{16} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin\theta \cos^3 \theta + (Q_{12} - Q_{22} - 2Q_{66}) \sin^3 \theta \cos \theta \\ \bar{Q}_{22} &= Q_{11} \sin^4 \theta + 2(Q_{12} + 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{22} \cos^4 \theta \\ \bar{Q}_{26} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin^3 \theta \cos \theta + (Q_{12} - Q_{22} - 2Q_{66}) \sin\theta \cos^3 \theta \\ \bar{Q}_{66} &= (Q_{11} + Q_{22} - 2Q_{12} - Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{66} (\sin^4 \theta + \cos^4 \theta) \end{aligned} \quad (10)$$

As deformações, em um laminado, podem ser reduzidas a $\varepsilon_x, \varepsilon_y$ e γ_{xy} devido às hipóteses de Kirchhoff (JONES, 1999). Logo:

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \varepsilon_x^o \\ \varepsilon_y^o \\ \gamma_{xy}^o \end{bmatrix} + z \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} \Rightarrow \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^o + z\boldsymbol{\kappa}, \quad (11)$$

onde $\boldsymbol{\varepsilon}^o$ são as deformações na superfície média e $\boldsymbol{\kappa}$ a curvatura na superfície média. Já no caso dos esforços resultantes, força e momentos, podem ser obtidos integrando as tensões ao longo da espessura t (JONES, 1999):

$$\mathbf{N} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} dz \quad \mathbf{M} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} z dz. \quad (12)$$

Uma relação entre os esforços resultantes e as deformações generalizadas do laminado pode ser expressa pela combinação das Equações (6) e (12), na forma:

$$\begin{bmatrix} \mathbf{N} \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \varepsilon^o \\ \kappa \end{bmatrix}, \quad (13)$$

Sendo

$$A_{i,j} = \sum_{k=1}^n \int_{z_k}^{z_{k+1}} \bar{Q}_{i,j}^k dz \quad B_{i,j} = \sum_{k=1}^n \int_{z_k}^{z_{k+1}} \bar{Q}_{i,j}^k z dz \quad D_{i,j} = \sum_{k=1}^n \int_{z_k}^{z_{k+1}} \bar{Q}_{i,j}^k z^2 dz \quad (14)$$

Integrando ao longo de cada lâmina tem-se:

$$A_{i,j} = \sum_{k=1}^n \bar{Q}_{i,j}^k (z_{k+1} - z_k) \quad B_{i,j} = \frac{1}{2} \sum_{k=1}^n \bar{Q}_{i,j}^k (z_{k+1}^2 - z_k^2) \quad D_{i,j} = \frac{1}{3} \sum_{k=1}^n \bar{Q}_{i,j}^k (z_{k+1}^3 - z_k^3) \quad (15)$$

2.4 Critérios de falha

Existem critérios que podem ser utilizados para verificar a falha de materiais compósitos reforçados por fibras, como o critério da máxima tensão, o critério da máxima deformação e o critério de Tsai-Wu. Segundo Jones (1999), alguns critérios sofrem inadequações em suas representações experimentais e uma maneira de evitar isso é aumentar o número de termos na equação de predição.

No critério de máxima tensão, a falha ocorre quando a tensão em uma dada direção se torna maior que a resistência do material nessa respectiva direção. Essa condição é expressa por:

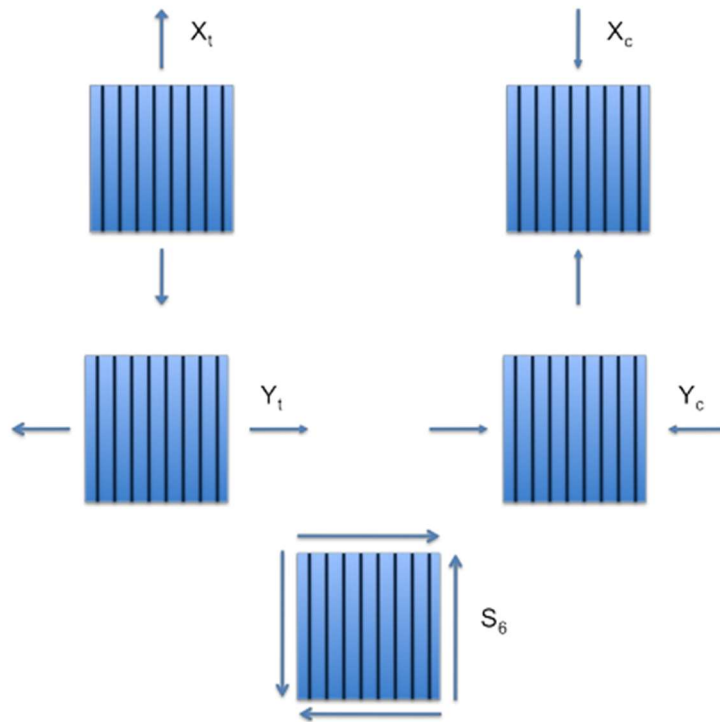
$$\sigma_1 = \begin{cases} X_t, & \text{quando } \sigma_1 > 0 \\ -X_c, & \text{quando } \sigma_1 < 0 \end{cases} \quad (16)$$

$$\sigma_2 = \begin{cases} Y_t, & \text{quando } \sigma_2 > 0 \\ -Y_c, & \text{quando } \sigma_2 < 0 \end{cases}$$

$$|\tau_6| = S_6.$$

Onde X_t , X_c , Y_t , Y_c e S , representados na Figura 5, são os parâmetros que caracterizam a resistência do compósito quando submetido a um estado plano de tensões.

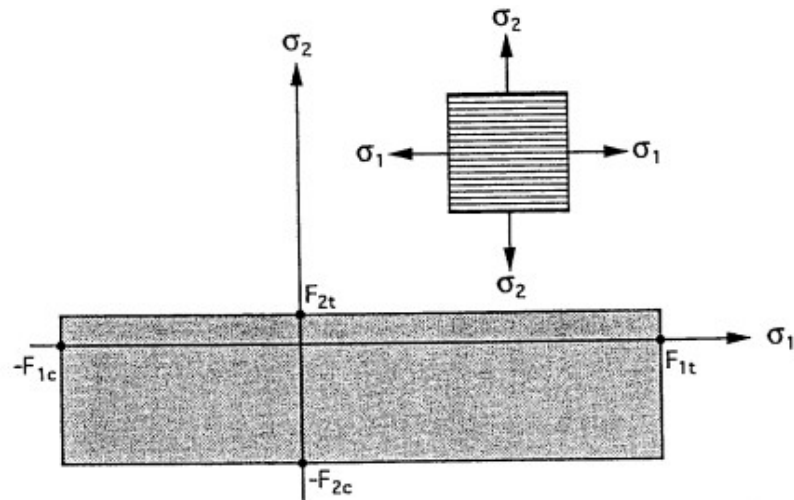
Figura 5 – Representação dos parâmetros de resistência.



Fonte: Rocha (2013).

Para um estado de tensão biaxial, $\tau_6 = 0$, a envoltória apresenta uma forma retangular, como representado na Figura 6 (DANIEL; ISHAI, 1994). Esse critério é mais adequado para os modos frágeis de falha do material.

Figura 6 – Envoltória de falha para um estado biaxial.



Fonte Daniel e Ishai (1994)

O critério de máxima deformação tem uma abordagem similar ao critério explicado anteriormente, ou seja, as deformações são tratadas independentemente e a falha ocorre quando

ao menos uma deformação ultrapassa a resistência do material. Matematicamente, temos (DANIEL e ISHAI, 1994):

$$\varepsilon_1 = \begin{cases} \varepsilon_{1t}^u, & \text{quando } \varepsilon_1 > 0 \\ \varepsilon_{1c}^u, & \text{quando } \varepsilon_1 < 0 \end{cases}$$

$$\varepsilon_2 = \begin{cases} \varepsilon_{2t}^u, & \text{quando } \varepsilon_2 > 0 \\ \varepsilon_{2c}^u, & \text{quando } \varepsilon_2 < 0 \end{cases} \quad (17)$$

$$|\gamma_6| = 2|\varepsilon_{12}| = \gamma_6^u.$$

Os termos ε são referentes aos componentes de deformação do material no eixo principal.

Escrevendo a Equação (17) em termos de tensão, devido ao efeito Poisson, temos:

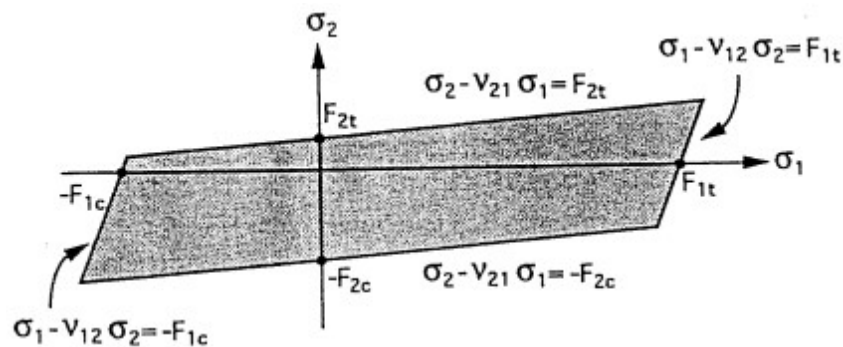
$$\sigma_1 - \nu_{12}\sigma_2 = \begin{cases} F_{1t}, & \text{quando } \varepsilon_1 > 0 \\ -F_{1c}, & \text{quando } \varepsilon_1 < 0 \end{cases}$$

$$\sigma_2 - \nu_{21}\sigma_1 = \begin{cases} F_{2t}, & \text{quando } \varepsilon_2 > 0 \\ -F_{2c}, & \text{quando } \varepsilon_2 < 0 \end{cases} \quad (18)$$

$$|\tau_6| = F_6.$$

Dessa forma, para um estado biaxial de tensões, onde $|\tau_6| = 0$, a envoltória se apresentará como na Figura 7, mostrando que este critério incorpora uma certa interação das tensões sobre a falha do material.

Figura 7 – Envoltória de falha para o critério de máxima deformação.



Fonte: Daniel e Ishai (1994)

Existem situações onde há uma necessidade de se obter o Fator de Segurança (S_f). No caso do critério da máxima tensão, o fator de segurança é dado por:

$$S_f = \min \left[\frac{X_t}{\sigma_1}, -\frac{X_c}{\sigma_1}, \frac{Y_t}{\sigma_2}, -\frac{Y_c}{\sigma_2}, \frac{S_6}{|\tau_6|} \right]. \quad (19)$$

Por outro lado, para o critério da máxima deformação:

$$S_f = \min \left[\frac{\varepsilon_{1t}^u}{\varepsilon_1}, -\frac{\varepsilon_{1c}^u}{\varepsilon_1}, \frac{\varepsilon_{2t}^u}{\varepsilon_2}, -\frac{\varepsilon_{2c}^u}{\varepsilon_2}, \frac{\gamma_6^u}{|\gamma_6|} \right]. \quad (20)$$

Tsai e Wu (1971), consideram o efeito da interação das componentes de tensão adotando a hipótese de que existe uma superfície de falha em um espaço de tensão, descrito na forma:

$$F_i \sigma_i + F_{ij} \sigma_i \sigma_j = 1 \quad (i, j, k = 1, 2, \dots, 6). \quad (21)$$

Expandido, o critério de falha, para materiais ortotrópicos em um estado plano de tensão, assume a seguinte forma (GÜRDAL; HAFTKA; HAJELA, 1998):

$$F_{11} \sigma_1^2 + F_{22} \sigma_2^2 + F_{66} \tau_{12}^2 + 2F_{12} \sigma_1 \sigma_2 + 2F_{16} \sigma_1 \tau_{12} + 2F_{26} \sigma_2 \tau_{12} + F_1 \sigma_1 + F_2 \sigma_2 + F_6 \tau_{12} = 1. \quad (22)$$

Os tensores de forças de segunda e quarta ordem, F_i e F_{ij} , são determinados pelas seguintes relações:

$$\begin{aligned} F_1 &= \frac{1}{X_t} - \frac{1}{X_c} & F_{11} &= \frac{1}{X_t X_c} \\ F_2 &= \frac{1}{Y_t} - \frac{1}{Y_c} & F_{22} &= \frac{1}{Y_t Y_c} \\ F_6 &= F_{16} = F_{26} = 0 & F_{66} &= \frac{1}{S^2} \end{aligned} \quad (23)$$

De acordo com Daniel e Ishai (1994), o coeficiente F_{12} , na maioria dos casos, não é crítico, podendo ser aproximado por:

$$F_{12} \cong -\frac{1}{2} (F_{11} F_{22})^{1/2}$$

O fator de segurança (S_f) é o valor que multiplica as tensões de modo a chegar no limite da falha. Assim, com base na Equação (21), temos:

$$F_i (S_f \sigma_i) + F_{ij} (S_f \sigma_i) (S_f \sigma_j) - 1 = 0 \quad (i, j = 1, 2, \dots, 6). \quad (24)$$

Esta pode ser reescrita na forma:

$$aS_f^2 + bS_f - 1 = 0 \quad (25)$$

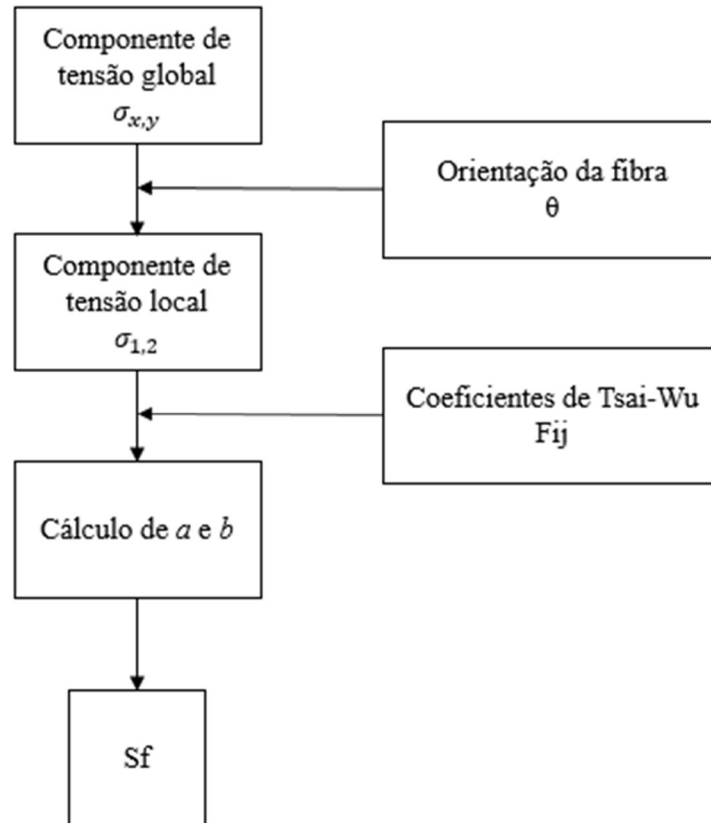
com

$$\begin{aligned} a &= F_{11} \sigma_1^2 + F_{22} \sigma_2^2 + F_{66} \tau_6^2 + F_{12} \sigma_1 \sigma_2 + \\ b &= F_1 \sigma_1 + F_2 \sigma_2 \end{aligned}$$

Nota-se que a Equação (25) é de segundo grau na variável S_f , nesse caso, é suficiente determinar as raízes da equação para encontrar o fator de segurança. No projeto de laminados,

o fator de segurança calculado deve ser comparado com o fator de segurança requerido. De forma resumida, a Figura 8 mostra como determinar o fator de segurança quando as tensões globais são fornecidas.

Figura 8 – Fluxograma para determinação do S_f .



Fonte: Adaptado de Daniel e Ishai (1994).

3 OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS

Quando se trabalha com estruturas laminadas de material compósito reforçados por fibras, muitas variáveis influenciarão no resultado final, seja na resistência ou no peso dos elementos. Encontrar a melhor configuração do laminado pode se tornar uma tarefa difícil. Nesse momento, utilizar ferramentas de otimização é a melhor saída.

Para Haftka e Gürdal (1992), a otimização se concentra em buscar o melhor projeto que satisfaz a um conjunto de restrições, representando condições previamente estabelecidas que precisam ser atendidas.

Um projeto eficiente de uma estrutura laminada precisa harmonizar o dimensionamento das seções transversais e espessura das camadas com as propriedades dos materiais, por meio da orientação, número e sequência de empilhamento das camadas (GÜRDAL; HAFTKA; HAJELA, 1998). Segundo os mesmos autores, no contexto de otimização, as propriedades do material estão associadas com a maximização ou minimização de um critério específico. Esse critério pode ser o peso do laminado, considerando, por exemplo, a rigidez como restrição do problema. Outro caminho é a otimização de uma característica específica, geralmente a resistência, usando o peso como restrição.

Os problemas de otimização citados no parágrafo anterior são formulados como:

$$\begin{aligned}
 & \text{Encontre } \mathbf{x} = [(e_1, \theta_1, m_1), (e_2, \theta_2, m_2), \dots, (e_{NI}, \theta_{NI}, m_{NI})] \\
 & \text{Que maximize/minimize } f(\mathbf{x}) \\
 & \text{Sujeito a } g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, Ng
 \end{aligned}
 \tag{26}$$

Sendo \mathbf{x} o conjunto de variáveis, e a espessura da lâmina, θ a direção das fibras na lâmina e m o material de cada camada, $f(\mathbf{x})$ a função objetivo (resistência ou função custo) e $g(\mathbf{x})$ as restrições do problema.

Por questões de fabricação, a direção das fibras no laminado normalmente assume valores discretos entre 0° e 90° , variando em 45° . Em alguns casos, como em Nicholas, Babu e Sofia (2016), a variação também pode ocorrer em intervalos menores, de 15° a 5° , sendo este tipo de laminado classificado como disperso.

Na literatura é possível encontrar diversas técnicas de otimização. Aqui serão utilizados os algoritmos bioinspirados: Algoritmos Genéticos (AGs) e Otimização por Nuvem de Partículas (PSO), dois dos algoritmos implementados no BIOS (programa desenvolvido no Laboratório de Mecânica Computacional e Visualização - LMCV). Os tópicos seguintes tratam com mais clareza esses dois assuntos.

3.1 Algoritmos Bioinspirados

Os algoritmos aqui apresentados pertencem ao que se conhece por Computação Natural (FACELI *et al.*, 2011). O termo “Computação Natural” se refere às técnicas de programação que envolvem algoritmos desenvolvidos com base nos processos ocasionados na natureza. Essa inspiração, segundo os autores, é oriunda das mais diversas atividades, por exemplo, o Algoritmo de Enxame de Partículas (*Particle Swarm Optimization* – PSO) foi desenvolvido a partir do comportamento de bandos de animais em busca de alimentos. Outro exemplo são os Algoritmos Genéticos (AG’s), que se baseiam na teoria evolutiva da seleção natural, onde o mais apto tem maior probabilidade de passar seus genes adiante. Esses algoritmos podem ser chamados de algoritmos bioinspirados.

Uma característica dessas técnicas é a robustez em relações aos modelos de programação matemática, além do que aqueles podem ser utilizados para qualquer problema, contínuo ou não, uma vez que estes algoritmos não utilizam os gradientes da função objetivo e das restrições. Um dos problemas relacionado aos algoritmos bioinspirados é o elevado tempo de processamento, que depende do tamanho do problema.

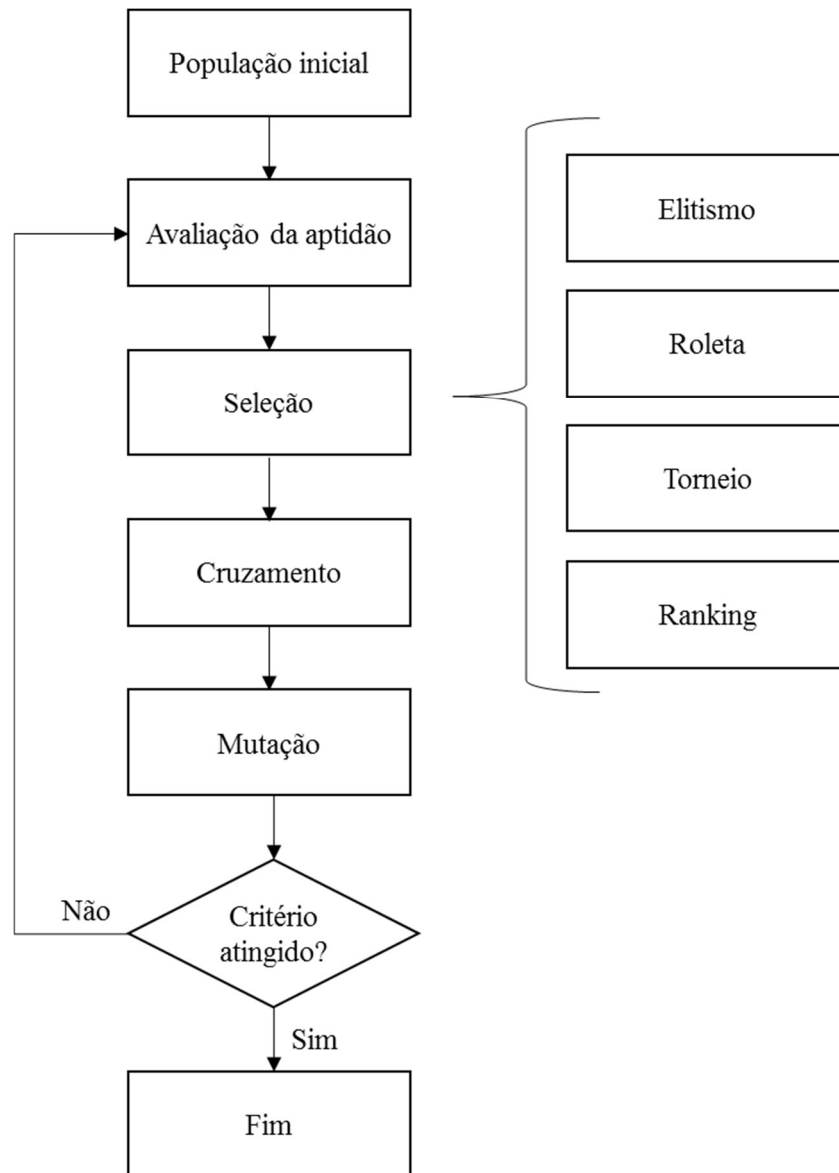
3.1.1 Algoritmos Genéticos

Os Algoritmos Genéticos constituem uma técnica de otimização desenvolvida por Holland (1975). São algoritmos de otimização global baseados na teoria da seleção natural de Darwin, segundo a qual o indivíduo mais bem adaptado ao ambiente tem maior probabilidade de sobreviver e, portanto, maior chance de propagar suas características genéticas.

Os AGs partem de um conjunto de pontos no espaço de projeto escolhidos aleatoriamente, a população inicial, e geram novos indivíduos através de operadores genéticos, como cruzamento e mutação.

O que define a capacidade de um indivíduo passar suas informações para as próximas gerações é a sua função aptidão. Em problemas irrestritos a função aptidão é igual à função objetivo. No caso de problemas com restrições, a função aptidão será uma combinação entre a função objetivo e uma penalidade adotada. Essa penalidade é aplicada nos indivíduos que violaram as restrições, pois eles podem apresentar características boas para serem repassadas as próximas gerações. A Figura 9 apresenta uma estrutura simplificada de um AG.

Figura 9 – Estrutura de um Algoritmo Genético.



Fonte: Elaborada pelo autor.

No AG existem tr s componentes b sicos que ser o definidos de acordo com cada problema a ser otimizado: gera o da popula o, fun o aptid o e os operadores gen ticos. A gera o da popula o inicial normalmente   aleat ria, e vai depender do problema a ser otimizado. A maioria dos AGs utilizam uma popula o fixa, ou seja, o n mero de indiv duos escolhidos inicialmente determinar  o tamanho da popula o em todas as outras gera es. A cada gera o, a popula o passa por tr s operadores gen ticos: a sele o, o cruzamento e a muta o (ARORA, 2004; VIANA, 1998).

Os algoritmos gen ticos t m sido utilizados com sucesso na otimiza o de estruturas no Programa de P s-Gradua o em Engenharia Civil: Estruturas e Constru o Civil da UFC. Silva (2012) desenvolveu ferramentas computacionais em AG e aplicou em risers

compósitos. Rocha, Parente e Melo (2014) trabalharam com AG híbrido na otimização de cascas laminadas.

Nesse trabalho, foi utilizado o Bio-Inspired Optimization System (BIOS), desenvolvido no LMCV, e escrito em C++. Segundo Rocha (2013), que implementou o AG, o BIOS possui técnicas de paralelização híbrida, visando a redução do tempo de execução do BIOS. Além disso, Barroso (2015) implementou o PSO e aplicou na otimização de placas laminadas. O processo de codificação do BIOS, aqui explicado, foi feito com base em Rocha (2013).

3.1.1.1 Codificação

Uma das características do BIOS, é que ele não trabalha com números binários, mas com um sistema de codificação inteira das variáveis, organizando esses valores em matrizes. Na matriz, o número de colunas representa o número de laminadas e cada linha representa um parâmetro da lâmina (espessura, orientação, material). Um exemplo dessa codificação é apresentado na Figura 10.

Figura 10 – Sistema de codificação e decodificação.

Código	Espessura	Código	Ângulo	Código	Material
0	0.0mm	1	-45°	1	Material 1
1	0.1mm	2	-25°	2	Material 2
2	0.2mm	3	-5°	3	Material 3
3	0.3mm	4	+25°	4	Material 4
4	0.4mm	5	+45°	5	Material 5

Espessura	←	$\left \begin{array}{ccccc} 4 & 1 & 4 & 2 & 1 \\ 1 & 2 & 5 & 1 & 1 \\ 2 & 3 & 3 & 5 & 4 \end{array} \right $	$\xrightarrow{\text{Decodificação}}$	$\left \begin{array}{ccccc} 0.4 & 0.1 & 0.4 & 0.2 & 0.1 \\ -45^\circ & -25^\circ & 45^\circ & -45^\circ & -45^\circ \\ \text{Mat 2} & \text{Mat 3} & \text{Mat 3} & \text{Mat 5} & \text{Mat 4} \end{array} \right $
Orientação	←			
Material	←			
		Genótipo	Fenótipo	

Fonte: Rocha (2013).

Para os laminados em geral, o número de colunas é igual ao número de camadas, como ilustrado na Figura 10. Para os laminados simétricos, o número de colunas é igual a $\frac{1}{2}$ do número de lâminas. Finalmente, para os laminados balanceados-simétricos, o número de colunas é igual a $\frac{1}{4}$ do número de lâminas. O processo de decodificação é capaz de tratar estes casos de maneira simples, gerando laminados com a sequência de laminação correspondente.

A representação genotípica é usada para aplicar os operadores genéticos (seleção, cruzamento mutação). A representação fenotípica é usada para avaliar a função objetivo e a função aptidão, ou seja, é a representação utilizada para análise estrutural.

3.1.1.2 Tratamento das restrições

A necessidade de utilizar técnicas para tratar restrições surge pelo fato de que os AGs não conseguem fazer isso diretamente. Nesse caso, é preciso garantir que o processo de otimização gere projetos viáveis. Para tal, existem os métodos diretos e os indiretos.

Os métodos diretos, trabalham somente com os pontos viáveis. O primeiro é o método de pena de morte, caracterizado pela rejeição dos pontos inviáveis e geração de novos pontos até que seja gerado um ponto viável. O problema desse método é que dependendo do tamanho do espaço de projeto ele pode se tornar computacionalmente caro (ROCHA, 2013).

Outra alternativa é a utilização de uma estratégia de codificação que garanta a geração de fenótipos viáveis para qualquer genótipo fornecido. Esse método é vantajoso para restrições simples, como as restrições laterais. Existe ainda o método de reparo, que traz um ponto inviável de volta ao espaço de projeto. Este tipo de estratégia pode ser eficiente, mas geralmente só pode ser aplicado a restrições simples, como aquelas que possuem expressões explícitas em função das variáveis de projeto.

No caso de restrições que dependem de forma implícita das variáveis de projeto, como as de rigidez e resistência, o método de penalidade é o mais utilizado. Este método consiste em penalizar o valor da função objetivo dos projetos inviáveis, de forma a reduzir a sua probabilidade de seleção. Esta abordagem permite que indivíduos inviáveis, mas com características interessantes para o projeto, participem do processo de seleção.

Existem diversas funções de penalidade, sendo a mais comum a função de penalidade estática:

$$f_p = f(\mathbf{x}) + \sum_{l=1}^m k \max(g_l(\mathbf{x}), 0) \quad (27)$$

onde f é a função objetivo original, f_p a função penalizada, m o número de restrições e k o fator de penalidade. A desvantagem desta função é a dificuldade de escolher um fator de penalidade adequado.

Um outro método penaliza a função objetivo na forma (DEB, 2000):

$$f_p = \begin{cases} f(\mathbf{x}), & \text{se } \mathbf{x} \text{ for viável} \\ f_{max} + \sum_{l=1}^m \max(g_l(\mathbf{x}), 0) & \text{se } \mathbf{x} \text{ for inviável.} \end{cases} \quad (28)$$

Ou seja, quando inviável, atribui-se ao indivíduo o valor mais alto de toda a população, f_{max} , e em seguida, as violações de todas as restrições são adicionadas a esse valor. Como observou Rocha (2013), esse método é de fácil implementação e não requer a escolha de nenhum

parâmetro por parte do usuário.

A função de penalidade adaptativa proposta por Lemonge e Barbosa (2004) e envolve o cálculo de um fator de penalidade, k_j , para cada restrição. Uma característica desse método é a existência de uma função aplicada somente aos pontos inviáveis, expressa por:

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{se } f(\mathbf{x}) > f_m(\mathbf{x}) \\ f_m(\mathbf{x}) & \text{se } f(\mathbf{x}) < f_m(\mathbf{x}). \end{cases} \quad (29)$$

A Eq. (29) mostra que a função objetivo dos pontos inviáveis é levada ao valor médio, f_m , de toda a população. Os valores de cada coeficiente de penalidade são obtidos pela seguinte equação,

$$k_j = |f_m| \frac{g_{m(j)}}{\sum_{l=1}^m g_{m(l)}^2}. \quad (30)$$

Onde g_m é o valor médio das violações de cada restrição.

Com as Equações (29) e (30) é possível escrever a função objetivo penalizada, na forma,

$$f_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{se } \mathbf{x} \text{ for viável} \\ \bar{f}(\mathbf{x}) + \sum_{j=1}^m k_j \max(g_j(\mathbf{x}), 0) & \text{se } \mathbf{x} \text{ for inviável.} \end{cases} \quad (31)$$

3.1.1.3 Função aptidão e Seleção

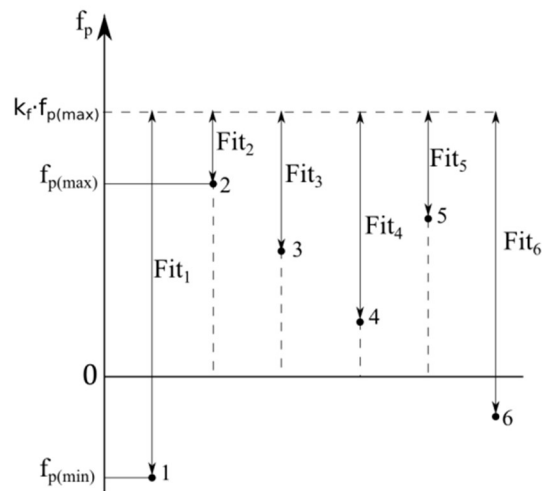
Segundo Rocha (2013), o processo de seleção é realizado em duas etapas. A primeira etapa consiste na formação de uma população intermediária, os indivíduos para essa população são escolhidos a partir da função aptidão, que permite avaliar o quanto um indivíduo é melhor ou pior que os outros. A função aptidão implementada no algoritmo é dada por:

$$Fit_i = k_f \max(|f_{p(\min)}|, |f_{p(\max)}|) - f_{pi}, \quad (32)$$

sendo $f_{p(\min)}$ e $f_{p(\max)}$ os valores mínimo e máximo das funções objetivo penalizadas.

O mapeamento linear da função objetivo é apresentado na Figura 11. Observe, que esse mapeamento garante um valor sempre positivo para a função aptidão.

Figura 11 – Mapeamento da função objetivo.



Fonte: Rocha (2013).

A segunda etapa, consiste na *seleção* propriamente dita, onde é feito a escolha de indivíduos (pais) para o cruzamento. Depois de escolhido, cromossomos de cada pai são usados para gerar um novo indivíduo (filho). Essa seleção pode ser feita, além do *elitismo* que seleciona os dois melhores indivíduos para passar adiante, por *roleta* ou *ranking*. A primeira, mais utilizada, refere-se as roletas do cassino. Sendo a probabilidade de cada indivíduo (P_i) dependente da função aptidão (f_i). A Tabela 1 mostra um exemplo com a probabilidade de cada indivíduo ser selecionado na roleta.

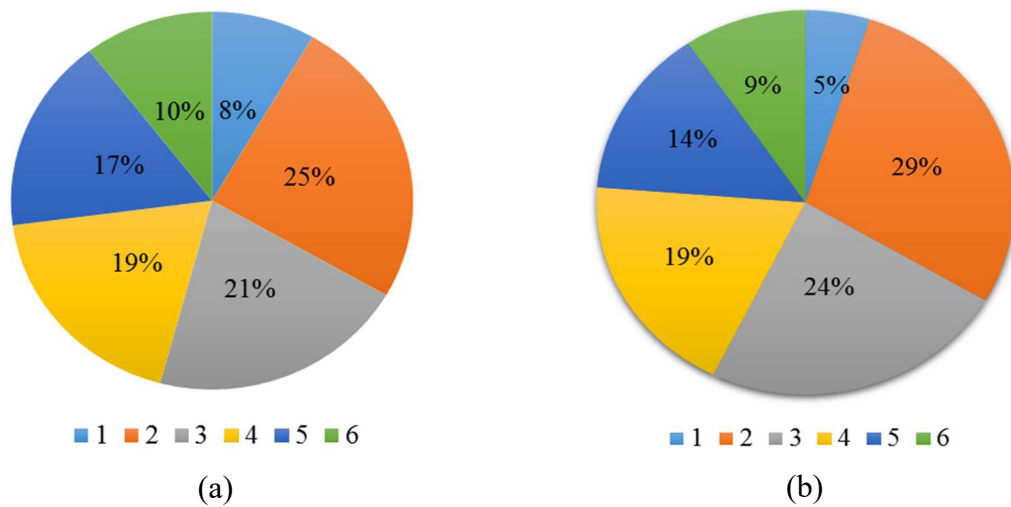
Tabela 1 – Probabilidade de seleção por roleta.

Indivíduo	f_i	$P_i = F_i / \Sigma f$ (%)
1	8	8,3
2	24	25,0
3	20	20,8
4	18	18,7
5	16	16,7
6	10	10,4

Fonte: Elaborada pelo autor.

A distribuição na roleta é apresentada na Figura 12, onde é visto a porcentagem de cada indivíduo (representado pelo número de 1 a 6). Essa porcentagem representa as chances que um indivíduo tem de ser escolhido. Observando a Figura 12 – (a), nota-se que o indivíduo 2 é o que apresenta maior probabilidade de ser selecionado.

Figura 12 – Distribuição de probabilidade de cada indivíduo para roleta (a) e ranking (b).



Fonte: Elaborada pelo autor.

A seleção por *ranking* classifica cada indivíduo de acordo com sua aptidão, em ordem crescente. Desse modo o indivíduo com maior aptidão apresentará maior valor no ranking e o indivíduo com menor aptidão terá valor de ranking igual a 1. Continuando com o exemplo usado anteriormente, pode-se observar o ranking de cada indivíduo na Tabela 2. A distribuição da probabilidade é mostrada na Figura 11 – (b)

Tabela 2 – Probabilidade de seleção por Ranking.

Indivíduo	F_i	Ranking	$P_i = R_k / \sum R_k (\%)$
1	8	1	5,0
2	24	6	28,6
3	20	5	23,8
4	18	4	19,0
5	16	3	14,3
6	10	2	9,0

Fonte: Elaborada pelo autor.

A seleção por ranking reduz a possibilidade de convergência prematura, pois a pressão de seleção se mantém constante, já que a probabilidade de seleção depende da posição do indivíduo no ranking. Enquanto que na seleção por roleta, o final do processo pode apresentar muitos indivíduos parecidos, com probabilidades de seleção bem próximas entre si,

fazendo com que a probabilidade de seleção do melhor indivíduo seja pouco superior ao dos demais.

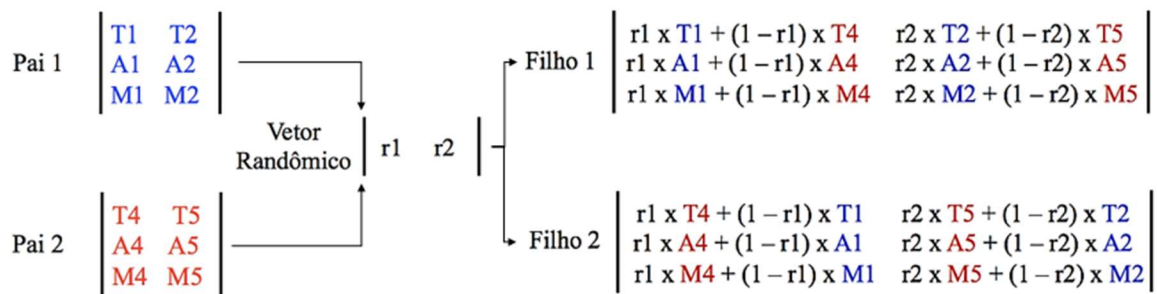
3.1.1.4 Cruzamento

A etapa seguinte é o *cruzamento* (*crossover*), que gera um novo indivíduo (filho) através do cruzamento das características de outros dois indivíduos (pais), aumentando a variabilidade genética. A quantidade de indivíduos selecionados é definida pela taxa de cruzamento (t_c). O cruzamento é um operador genético essencial para a convergência do processo de otimização, pois busca gerar novos indivíduos combinando as melhores características dos pais.

No algoritmo implementado por Rocha (2013), foi utilizada uma combinação linear dos pais, gerando novos filhos. O processo, exemplificado na

Figura 13, consiste em gerar um número, $r \in [0,1]$, aleatoriamente, que será utilizado no cálculo de uma média ponderada dos genes paternos. Quando fracionado, arredonda-se para o inteiro mais próximo.

Figura 13 – processo de cruzamento de um laminado com duas camadas.



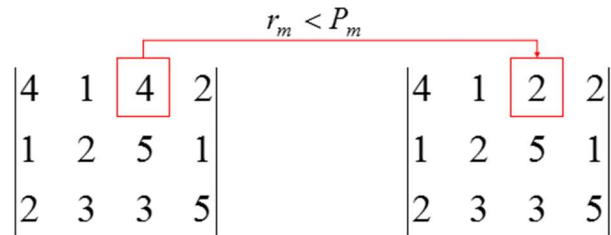
Fonte: Adaptada de Rocha (2013).

3.1.1.5 Mutação

A *mutação*, processo que segue o cruzamento, é uma perturbação na configuração do indivíduo, realizada de maneira aleatória. Esta etapa evita que informações genéticas importantes sejam perdidas, além de manter a diversidade da população reduzida pela seleção e cruzamento, impedindo a convergência prematura do algoritmo para mínimos locais.

A mutação é implementada por meio de uma geração aleatória de um número $r_m \in [0,1]$, se esse número for menor que a taxa de mutação (P_m) o indivíduo sofrerá mutação (GÜRDAL; HAFTKA; HAJELA, 1998). Um exemplo de mutação é apresentado na Figura 8.

Figura 14 – Mutaç o no indiv duo gerado pelo cruzamento.



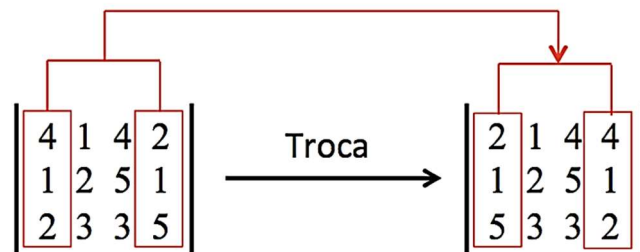
Fonte: Adaptada de Rocha (2013).

3.1.1.6 Operadores espec ficos para laminados

Para problemas de laminados, tr s outros operadores entram em quest o. O primeiro   o operador de permuta o ou troca de l minas (*layer swap*) que troca de posi o duas l minas de um laminado. Este operador   importante em problemas de flambagem, deflex es e vibra es, pois ele altera a rigidez de flex o (**D**) mantendo constante a rigidez de membrana (**A**).

Esse operador tamb m possui uma probabilidade de ocorr ncia, chamada de probabilidade de troca (P_s). Um exemplo da atua o desse operador   apresentado na Figura 15.

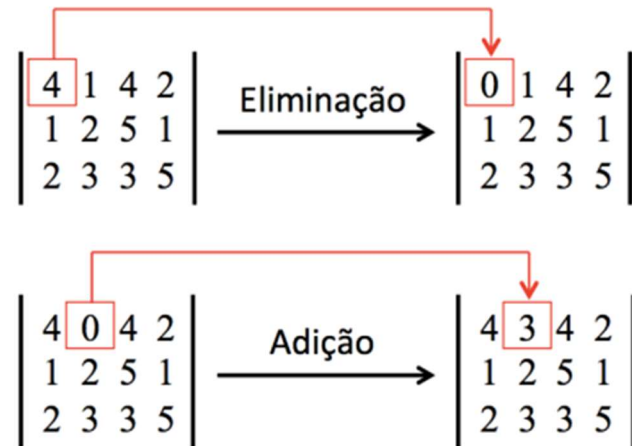
Figura 15 – operador troca de l minas.



Fonte: Rocha (2013).

O segundo operador   o de elimina o, e sua fun o   remover uma camada espec fica, zerando sua espessura. A probabilidade de ocorr ncia   denominada Probabilidade de elimina o (P_e). Por fim, tem-se o operador de adi o de camadas, que atua contrariando o operador de elimina o. Isso significa que, para cada camada eliminada, um valor r_a no intervalo $[0,1]$   determinado aleatoriamente, se r_a for menor que a probabilidade de adi o (P_a), um inteiro   atribu do   espessura da camada. Esses dois operadores s o observados na Figura 16.

Figura 16 – Operadores de eliminação e adição.



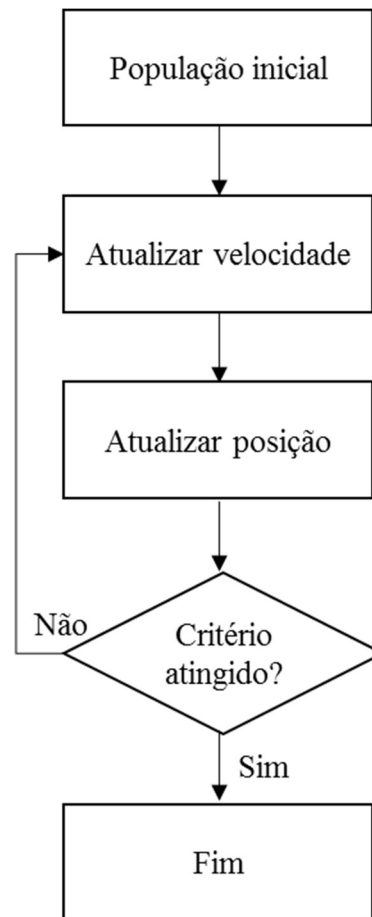
Fonte: Rocha (2013).

Observa-se que no processo de eliminação, as informações sobre o ângulo e material continuam armazenados, porém a espessura se torna nula. Para esse trabalho, os operadores de adição e eliminação não foram utilizados.

3.1.2 Otimização por Nuvem de Partículas

A Otimização por nuvem de Partículas, desenvolvidas por Kennedy e Eberhart (1995), foi criada com base no comportamento social de organismos biológicos, semelhante ao comportamento de bando de aves ou cardume de peixes em busca de alimentos, ou seja, inicialmente dispersos, mas com o passar do tempo (de busca) convergem para o local de alimentação (o ponto ótimo). As partículas desse algoritmo apresentam comportamento semelhante, ou seja, utilizam o aprendizado próprio e do conjunto, diferente dos AGs, onde os indivíduos competem entre si, para alcançarem o ótimo da função. O algoritmo básico de um PSO é ilustrado na Figura 17.

Figura 17 – Estrutura de um algoritmo PSO.



Fonte: Adaptado de Siciliano (2007).

A proposta do PSO é, por meio de um conjunto de operadores, mover a partícula para um local promissor no espaço de projeto, em busca da solução ótimo. A responsabilidade do processo está no vetor velocidade de cada partícula, pois este vai determinar a direção que será feito o movimento. Esse vetor (v_{ik}) é atualizado de acordo com as melhores posições da partícula e de sua vizinhança, conforme a equação que segue:

$$v_{i(k+1)} = w \cdot v_{ik} + c_1 \cdot e_1 \cdot (p_{ik} - x_{ik}) + c_2 \cdot e_2 \cdot (p_{gk} - x_{ik}), \quad (33)$$

onde, w representa a inércia da partícula, p_{ik} é melhor posição encontrada pela partícula i , p_{gk} é a melhor posição das partículas vizinhas na iteração k , c_1 e c_2 representam a confiança da partícula em si e no bando, respectivamente e e_1 e e_2 são números aleatórios no intervalo $[0,1]$.

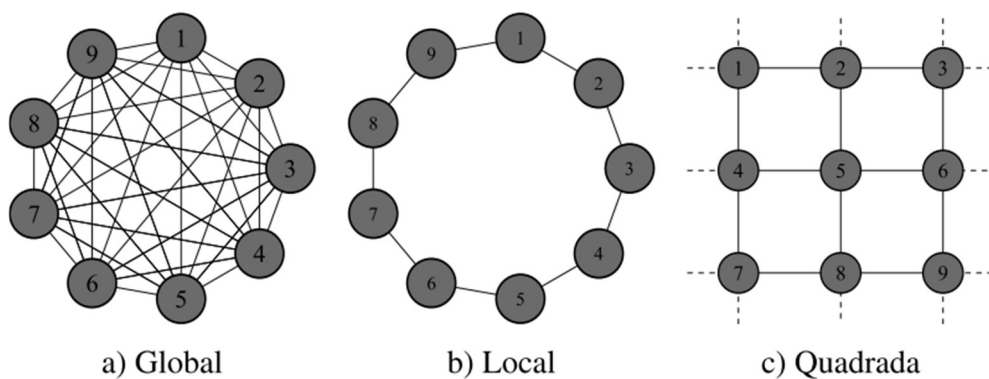
A posição (x_i) de cada partícula na próxima iteração ($k + 1$) depende da velocidade calculada na Equação (33), sendo expressa por:

$$x_{i(k+1)} = x_{ik} + v_{i(k+1)} \quad (34)$$

A população do PSO pode ser organizada por uma topologia global, local ou

quadrada (KENNEDY; MENDES, 2002). Na topologia global, todas as partículas possuem informações a respeito das demais partículas. Enquanto que na topologia local e quadrada, cada partícula só possui informação de suas vizinhas. Desta forma, a topologia global possui uma taxa de convergência maior que as topologias local e quadrada, porém corre maior risco de convergir prematuramente para mínimos locais (BRATTON; KENNEDY, 2007). Já as topologias local e quadrada, possuem uma maior capacidade de percorrer todo o espaço de projeto, com menores chances de convergirem prematuramente.

Figura 18 – Topologia do PSO.



Fonte: Barroso (2015).

As restrições de projeto são tratadas de maneira semelhante aos métodos utilizados para os AGs. Já as restrições laterais, segundo Barroso (2015), são tratadas de forma diferenciada. Nesse caso, o algoritmo não permite que as partículas saiam do espaço de projeto. Dessa forma, quando uma restrição é violada, o valor da variável violada é alterado de forma que a partícula não saia da região viável.

4 MODELOS SUBSTITUTOS

Os algoritmos bioinspirados discutidos no capítulo anterior, apesar de apresentarem robustez na busca de um ótimo global, deixam a desejar no que diz respeito à eficiência computacional, devido ao grande número de avaliações da função objetivo e das restrições requerido durante o processo de otimização.

Isso significa que, para problemas complexos, o tempo de processamento pode ser muito elevado. Isto ocorre porque em muitos casos de Engenharia, os problemas não possuem soluções analíticas simples, sendo necessário o uso de técnicas numéricas como o Método dos Elementos Finitos (MEF). Wang e Shan (2007) exemplificaram esse problema usando o caso da indústria automotiva, que enfrenta problemas com simulações computacionais de colisões (*crash simulations*), pois tais simulações, cujos processamentos duravam horas, tornavam inviável a utilização de técnicas de otimização.

De acordo com Pina (2009), mesmo com os avanços dos computadores, o custo computacional para simulações de problemas de engenharia ainda é muito alto. Nesse contexto o uso de modelos substitutos torna-se uma forma de reduzir o tempo desse processo.

Simpson *et al.* (2001) se referem ao modelo substituto como sendo um modelo aproximado e computacionalmente eficiente usado para substituir uma análise onerosa, seja ela computacional ou experimental. Wang e Shan (2007) afirmam que os esses modelos têm sido desenvolvidos por meio da Estatística, Matemática, Engenharia e Ciência da Computação.

Hoje a maior parte das análises de engenharia consiste em rodar códigos complexos em computadores, com objetivo de obter um conjunto de respostas (\mathbf{y}) para um conjunto de variáveis de entrada (\mathbf{x}). Isto porque, as análises de engenharia, quando feitas por meio do MEF, que hoje é uma das principais ferramentas de análise, além da AIG que vem se desenvolvendo, não fornecem uma relação explícita entre as variáveis. É importante notar que, uma análise pelo MEF se torna onerosa para problemas complexos, principalmente quando incorpora efeitos dinâmicos e não linearidades física e geométrica.

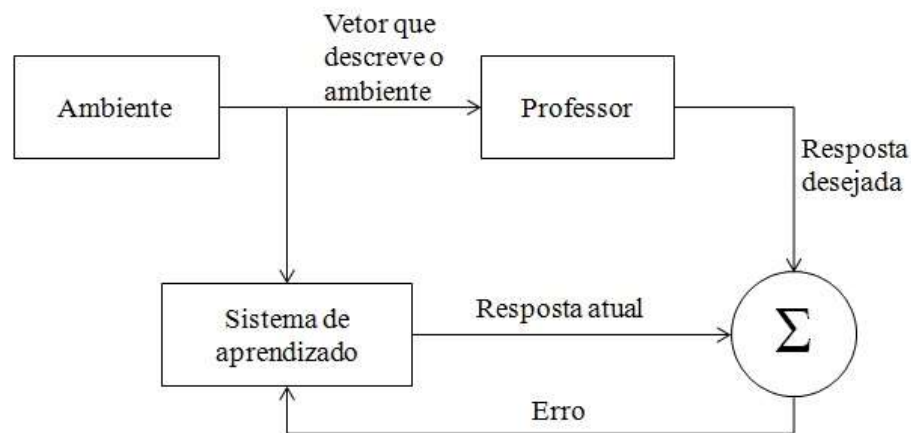
Nesses casos, a abordagem básica é construir um modelo aproximado, chamado de modelo de baixa fidelidade (*low fidelity model*) a partir do modelo original de alta fidelidade (*high fidelity model*), que seja mais eficiente na execução e ainda por cima, permita encontrar uma relação funcional entre \mathbf{x} e \mathbf{y} . Essa aproximação é dada por $\hat{\mathbf{y}} = g(\mathbf{x})$, sendo $\mathbf{y} = \hat{\mathbf{y}} + erro$. Esse erro é inerente ao uso de modelos substitutos, visto que são técnicas aproximadas, sendo a eficiência da aproximação dependente de vários fatores, inclusive da técnica escolhida para

construção do modelo substituto. As maneiras de avaliar o erro de um modelo aproximado serão discutidas mais adiante.

A função do modelo substituto pode ser entendida como um processo que ensina o computador a “pensar” de maneira similar à função que será substituída. Esse aprendizado pode ser, de modo geral, dividido em: aprendizado supervisionado e não supervisionado (DUDA; HART; STORK, 2000; COPPIN, 2013). Haykin (2001) divide em aprendizado com professor e aprendizado sem professor, já que ele subdivide o aprendizado sem professor em dois: aprendizado por reforço e aprendizado não supervisionado.

Para Coppin (2013), no aprendizado supervisionado o algoritmo aprende devido a informações de treinamento pré-classificadas (obtidas pela avaliação do modelo de alta fidelidade). Nesse caso existe um “professor” que vai avaliar a resposta do algoritmo de aproximação e estimar o erro de aproximação. Os parâmetros do modelo substituto são alterados de modo que as saídas se aproximem das informações preestabelecidas. A Figura 19 apresenta um diagrama que resume o aprendizado supervisionado.

Figura 19 – Diagrama do aprendizado supervisionado.



Fonte: Adaptado de Haykin (2001).

Na Figura 19, como é possível perceber, o “professor” recebe um conjunto de entrada-saída que representa o ambiente estudado (chamado também de conjunto de treinamento). Com esses dados o “professor”, então, ajusta a resposta do sistema de aprendizado com base no erro existente entre a sua resposta e a resposta do modelo.

No aprendizado não supervisionado, os modelos aprendem sem qualquer intervenção do homem, ou seja, não existe “professor”. Esse tipo de treinamento é caracterizado pela ausência do vetor resposta do problema de alta fidelidade, sendo responsabilidade do

modelo encontrar parâmetros adequados ao treinamento. Nesse trabalho, foi considerado apenas o aprendizado supervisionado para o treinamento da rede.

4.1 Parâmetros de desempenho

Já foi explicado que ao trabalhar com modelos substitutos estamos trabalhando com erros, devido sua natureza de aproximação. Dessa forma, uma análise desses parâmetros permite medir a adequação dos modelos substitutos. Alguns dos parâmetros de desempenho encontrados na literatura, são:

- A diferença entre os pontos reais (y_i) e os aproximados (\hat{y}_i):

$$e_i = y_i - \hat{y}_i, \quad (35)$$

- O erro máximo absoluto (*Maximum Absolute Error* - MAE):

$$MAE = \max |e_i|, \quad (36)$$

- O erro quadrático médio (*Mean Square Error* - MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2, \quad (37)$$

- A raiz quadrada do erro quadrático médio (*Root Mean Square Error* - RMSE):

$$RMSE = \sqrt{MSE}, \quad (38)$$

- A raiz quadrática do erro quadrático médio normalizado (*Normalized Root Mean Square Error* - NRMSE):

$$NRMSE = \sqrt{\frac{\sum_{i=1}^n e_i^2}{\sum_{i=1}^n (\hat{y}_i)^2}}. \quad (39)$$

Essas relações são utilizadas por Amouzgar e Strömberg (2016) para avaliar modelos substitutos baseados em Funções de Bases Radiais (*Radial Basis Function* – RBF). De acordo com o autor, quanto menor o valor dos erros, mais acurado é o modelo. Outra maneira de verificar a acurácia do modelo substituto é por meio da regressão, calculada da seguinte forma:

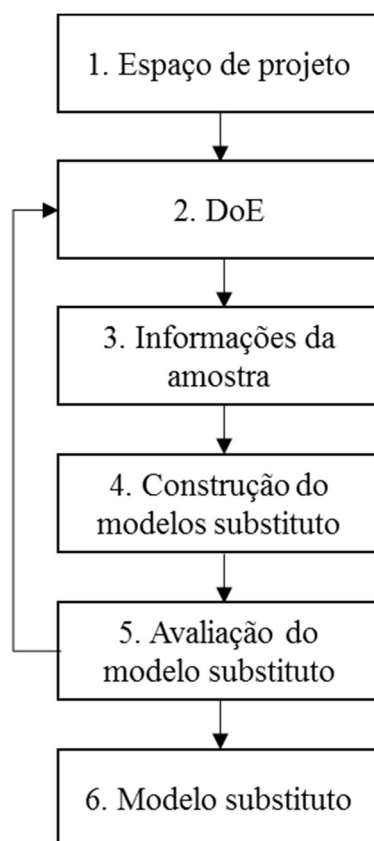
$$r^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (40)$$

Sendo \bar{y} a média dos pontos reais. Nesse caso quanto mais próximo de 1 mais acurada será o modelo. Forrester, Sóbester e Keane (2008) sugerem $r^2 > 0,8$ como sendo um limite ideal.

4.2 Construindo um modelo substituto

O fluxograma apresentado na Figura 20 elenca as etapas de criação de um modelo substituto. Essa sequência vale tanto para treinamentos supervisionados quanto para os não-supervisionados.

Figura 20 – Estrutura da criação de um modelo substituto.



Fonte: adaptado de Han e Zhan (2012).

Inicialmente é preciso determinar o espaço de projeto, ou seja, todo o espaço trabalhável. Esta etapa envolve tanto a definição das variáveis que controlam o modelo quanto a determinação dos limites inferiores e superiores destas variáveis.

O segundo passo, corresponde ao Projeto de Experimentos (*Design of Experiments* – DoE), utilizado para construir uma amostragem adequada do espaço de projeto. Essa determinação é feita por meio de técnicas de amostragem (fatorial completo, fatorial fracionado, aleatório, hipercubo latino e outros).

O fluxograma segue com o treinamento e validação do modelo substituto. Para isso, costuma-se adotar um valor mínimo para o parâmetro de desempenho. Assim é possível avaliar a qualidade do modelo. Caso não seja obtido uma boa aproximação faz-se necessário retornar ao passo 2, ou realizar alterações nos parâmetros de treinamento.

Na literatura há inúmeras técnicas de modelos substitutos, algumas são bem conhecidas, por exemplo, Método de Superfície de Resposta, Redes Neurais, Kriging, Funções de Base Radial e Máquina de Vetor Suporte.

Em Berthelemy e Haftka (1993) encontra-se uma abordagem dos conceitos de aproximação (Redes Neurais e Superfície de Resposta) no projeto estrutural ótimo. Uma revisão e comparação entre os métodos pode ser encontrada em Simpson *et al.* (2001) que aponta algumas recomendações para o uso dos modelos substitutos e algumas armadilhas que devem ser evitadas. Forrester e Keane (2009) fazem uma apresentação dos avanços dos modelos substitutos (Kriging, RBF, MVS). Algumas aplicações da técnica de Krigagem podem ser vistas em: Leary *et al.* (2004), que utilizaram esta técnica para otimizar problemas complexos; Souza *et al.* (2010) que aplicaram o Kriging em problemas de gerenciamento de reservatórios e Gaspar *et al.* (2014), onde essa técnica foi aplicado em problemas de reabilitação estrutural.

Quanto às técnicas de Máquina de Vetor Suporte e Funções de Base Radial (RBF), cita-se: Jayawardena e Fernando (1998) que usaram RBF em uma rede neural e aplicaram em simulações de escoamento em sistema hidrológico; Tiago e Leitão (2002) aplicaram RBF em problemas unidimensionais de engenharia, como treliças e vigas; Awruch *et al.* (2010) utilizaram RBFs na construção de um modelo substituto de painéis sujeitos à flambagem; Pan *et al.* (2010) aplicaram as técnicas de MVS de regressão na otimização de um carro sujeito a colisões e Pal e Deswal (2011) que utilizaram MVS de regressão para analisar a resistência ao cisalhamento em vigas.

4.3 Projeto de Experimentos

Quando se trabalha com modelos aproximados é necessário seguir alguns passos como explicado no fluxograma da Figura 20. Depois de obter um espaço de projeto, o passo seguinte é realizar um DoE para determinar a amostragem que interessa. De acordo com Frisso, Scarpel e Ferrari (2011), o Projeto de Experimentos busca selecionar pontos que serão avaliados e servirão como base para a construção do modelo substituto.

Segundo Mason, Gunst e Hess (2003), alguns pontos relevantes para entender o uso do DoE são:

- Os procedimentos estatisticamente projetados são mais econômicos;
- Permitir a mensuração de influência de um ou mais fatores em uma resposta;
- Permitir uma estimativa da magnitude do erro experimental;
- Experimentos projetados sem a adesão de princípios estatísticos geralmente violam uma ou mais metas de projetos desejáveis.

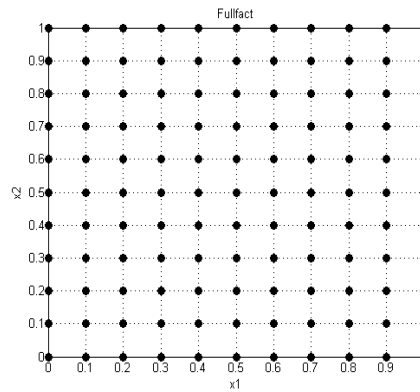
Os autores supracitados afirmam ainda que a economia de esforço, que os procedimentos experimentais fornecem reduzindo o número de testes conduzidos, pode ser enganadora por dois motivos. O primeiro motivo é que ele vai limitar o uso de fatores que serão estudados efetivamente. O segundo é que nos testes pode ocorrer que um fator seja avaliado por vez, impedindo que uma avaliação conjunta seja realizada.

De acordo com Coleman e Montgomery (1993) os passos para o experimento são:

- Reconhecimento e declaração do problema;
- Escolha dos fatores e níveis;
- Seleção das variáveis de resposta;
- Escolha do projeto de experimentos;
- Condução do experimento;
- Análise das informações;
- Conclusão do experimento.

A escolha do experimento é feita de acordo com a necessidade do projeto. Mason, Gunst e Hess (2003) explicam que quando o intuito é avaliar todas as interações entre os fatores (variáveis que serão avaliadas) e os níveis (como as variáveis serão alteradas), a melhor opção é o experimento fatorial completo (FC). Nesse experimento o número de amostras é dado por todas as combinações possíveis, afirma Cavazzuti (2013). Ou seja, para um caso mais simples de experimento, com k fatores e n níveis ($n = 2$), o tamanho da amostra será $N = k^n = k^2$, a Figura 21 apresenta um exemplo para $k = 2$ e $n = 11$, resultando em $N=121$ pontos de amostragem. Os níveis, para esse problema, são chamados de “+1” e “-1” ou “h” (high) e “l” (low), esses níveis representam o menor valor e o maior valor que um fator pode obter. No experimento fatorial completo é feito a alteração de um fator por vez e depois verifica a influência dessa alteração na resposta.

Figura 21 – Amostragem pelo experimento fatorial completo.



Fonte: Elaborada pelo autor.

O efeito principal (E) de um fator é explicado, matematicamente, pela diferença das respostas médias da variável de controle, da seguinte forma:

$$E = \bar{y}_h - \bar{y}_l \quad (41)$$

onde,

\bar{y}_h é a média das respostas dos níveis superiores (high ou “+1”) e \bar{y}_l é a média das respostas níveis inferiores (low ou “-1”).

Para exemplificar, tomemos o caso mais simples, um experimento com dois fatores (A e B) e dois níveis (+1 e -1). Nesse exemplo, o tamanho da amostra será $N = 2^2 = 4$. A Tabela 3 apresenta o modelo de um experimento.

Tabela 3 – Experimento fatorial $k=n=2$ e $N=2^2=4$.

Nº do experimento	Níveis do fator		Iterações	Variável de resposta
	A	B		
1	-1	-1	1	$y(l,l)$
2	-1	1	-1	$y(l,h)$
3	1	-1	-1	$y(h,l)$
4	1	1	1	$y(h,h)$

Fonte: Elaborada pelo autor.

Assim, o efeito principal do fator A (E_A), de acordo com Santanna-Greco e Fabbri (2003), será,

$$E_A = \frac{y(h,l) + y(h,h)}{4} - \frac{y(l,l) + y(l,h)}{4}, \quad (42)$$

e o efeito principal do fator B (E_B) é determinado de maneira similar à Equação (42). O efeito da interação, E_{AB} , é dado por (SANTANNA-GRECO; FABBRI, 2003),

$$E_{AB} = \frac{E_A^{B,h} - E_A^{B,l}}{2} \quad (43)$$

onde,

$E_A^{B,h}$ é o efeito principal de A quando B está com o nível mais alto e $E_A^{B,l}$ é o efeito principal de A quando B está no nível mais baixo.

Para o caso de experimentos fatoriais com três níveis ou mais, recomenda-se a leitura de Mason, Gunst e Hess (2003). Um exemplo pode ser visto em Santanna-Greco e Fabbri (2003).

O problema desta abordagem, como explica Cavazzuti (2013) e Santanna-Greco e Fabbri (2003), é a possibilidade de cada fator possuir diversos níveis de variação e por esse motivo a combinação de cada fator com os diferentes níveis pode se tornar computacionalmente cara.

Desse modo, entra em cena o experimento fatorial fracionado, usado quando questões orçamentárias e de tempo impedem o uso do fatorial completo. Nesse caso, não é necessário avaliar todas as interações para estimar a importância de um fator (MASON; GUNST; HESS, 2003).

Essa técnica permite que a resposta seja encontrada realizando apenas uma fração do experimento, explorando a redundância que existe nos experimentos com muitos fatores (SANTANNA-GRECO; FABBRI, 2003)

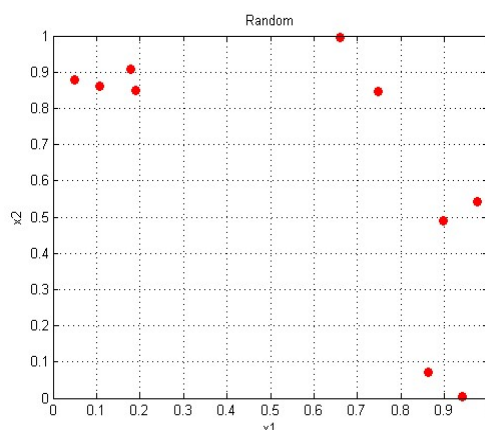
O tamanho da amostra (ou fracionamento do experimento) pode ser de um-meio ou um-quarto do fatorial completo, informa Cavazzuti (2013). No caso de um fracionamento de um-meio ($1/2p$) do fatorial completo, $2k$, teremos o experimento fatorial $2k-1$. Essa técnica pode ser encontrada em Mason, Gunst e Hess (2003) e em Simpson *et al.* (2001).

O experimento fatorial completo fornece uma amostra muito grande e isso vai exigir mais da máquina. Alternativamente, pode-se usar outras técnicas de amostragem, como a amostragem aleatória, hipercubo latino ou as sequências de baixas discrepância.

4.3.1 Amostragem Aleatória

A amostra usada na construção do modelo substituto pode ser definida aleatoriamente, porém, pode acarretar em um aglomerado de pontos em locais distantes do ótimo global. Essa é a técnica mais simples, mas requer atenção na escolha dos pontos de informação. A Figura 22 representa uma escolha aleatória, nota-se claramente uma região sem qualquer ponto representativo.

Figura 22 – Amostra gerada de forma aleatória.



Fonte: Elaborada pelo autor.

4.3.2 *Hipercubo Latino*

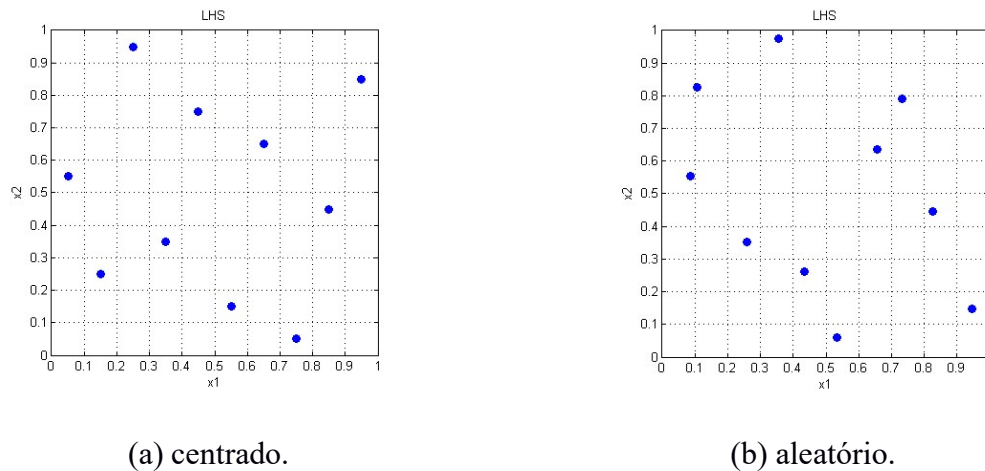
Segundo Frisso, Scarpel e Ferrari (2011), o Hipercubo Latino (HCL) é uma técnica de amostragem estratificada, esse procedimento garante a representação de cada variável ao longo do domínio.

Viana (2013) afirma que o HCL divide o domínio em níveis com igual probabilidade e retira uma amostra de cada nível, que representará toda a fatia de onde foi selecionada tanto no sentido horizontal quanto no sentido vertical. O mesmo autor afirma que a proposta inicial é usar um processo aleatório para determinar a posição dos pontos que irão compor a amostra. O projeto de experimentos por Hipercubo Latino fica sujeito a escolhas pobres em informações, devendo-se tomar cuidados ao escolher os pontos, dando preferência aos pontos que estão uniformemente distribuídos ao longo do domínio.

Viana (2013) ainda informa que essa técnica tem sido amplamente utilizada, pois permitir a criação de um DoE com a quantidade de pontos necessária (evitando redundâncias e inferências) e por apresentar grande flexibilidade.

Por possuir um ponto em cada faixa do domínio, o HCL garante que toda a região seja avaliada com um número menor de pontos. Outra característica do HCL é que ele pode gerar a amostra de duas formas: com os pontos centrados na região limitada ou com os pontos posicionados aleatoriamente na região limitada. Amostras geradas utilizando a ferramenta *lhsdesign* do MATLAB são apresentadas na Figura 23.

Figura 23 – Amostra obtida por Hipercubo Latino.



Fonte: Elaborada pelo autor.

Nesse caso, o HCL se torna a melhor opção para um experimento computacional, frente aos outros métodos explicados (fatorial e aleatório), visto que vai abranger todo o domínio com poucos pontos, não exigindo muito computacionalmente.

Porém, um problema que esse tipo de amostragem pode trazer é a má distribuição dos pontos, sendo, em alguns casos, pouco representativo. Para suprir essa falha Forrester, Sóbester e Keane (2008), propuseram o uso de um algoritmo em Linguagem MATLAB, o *bestlh*. Basicamente, o *bestlh* utiliza os AGs para otimizar o hipercubo latino. O que o algoritmo faz é encontrar a melhor posição dos pontos, buscando a maior distância entre eles. O problema desse algoritmo está no tempo de busca, pois quanto maior o número de gerações e população, melhor será a resposta, porém maior será o tempo que o algoritmo leva para encontrar a melhor solução.

4.3.3 Sequências de baixa discrepância

De acordo com Kalagnanam e Diwekar (1997), a discrepância é uma medida quantitativa do desvio de uma sequência distribuída uniformemente. Eles usaram a Sequência de Hammersley (SH) que pertence ao grupo de sequências de baixa discrepância (*low discrepancy sequences*) (KALAGNANAM; DIWEKAR, 1997).

A SH parte de uma sequência de baixa discrepância para distribuir N pontos em um espaço k -dimensional. Então, para um inteiro n , escrito na base R ,

$$n = n_i R^i, \quad i = 0, 1, 2, \dots, m. \quad (44)$$

sendo $m = \lceil \log_R n \rceil$. A distribuição de Hammersley no espaço k -dimensional é expressa na forma

(KALAGNANAM; DIWEKAR, 1997):

$$\mathbf{z}_k(n) = \left(\frac{n}{N}, \Phi_{R_i}(n) \right), \quad i = 1, 2, \dots, k-1 \quad \text{e} \quad n = 1, 2, \dots, N. \quad (45)$$

onde,

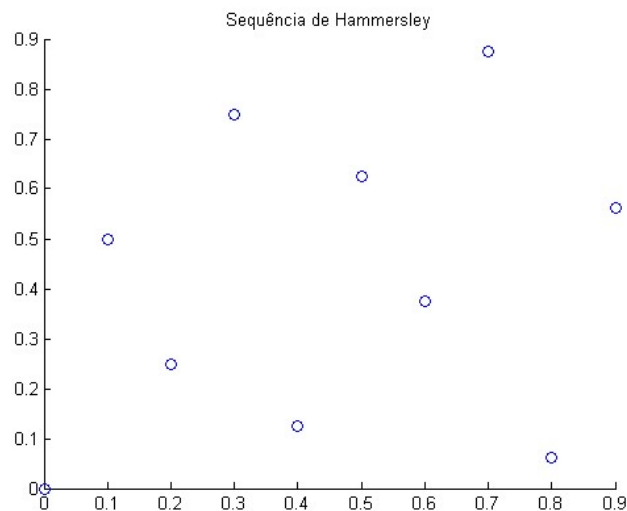
$$\Phi_R = \frac{n_0}{R} + \frac{n_1}{R^2} + \dots + \frac{n_m}{R^{m+1}} \quad (46)$$

e R_i ($i = 1, 2, \dots, k-1$) são os primeiros $k-1$ números primos. Sendo assim, os pontos de Hammersley são dados por:

$$\bar{x}_k(n) = 1 - \bar{z}_k(n). \quad (47)$$

Essa técnica, que se baseia na distribuição dos números primos, fornece pontos igualmente espaçados no domínio. Outro ponto importante é que, para problemas com muitas dimensões (i.e. grande número de variáveis), a Sequência de Hammersley tende a fornecer amostras mais uniformes que o Hipercubo Latino (AMOUZGAR; STRÖMBERG, 2016). Um exemplo dessa amostragem, para 10 pontos, é apresentado na Figura 24.

Figura 24 – Amostragem pela Sequência de Hammersley.



Fonte: Elaborada pelo autor.

4.4 Redes Neurais Artificiais

Rede Neural Artificial (RNA) é uma técnica de aprendizagem de máquina bastante conhecida e foi desenvolvida com o intuito de “imitar” a capacidade de aprendizado de um cérebro humano. Haykin (2001) define uma rede neural como sendo um processador maciço, trabalhando com uma estrutura altamente paralelizada, com uma disposição natural a aprender

e tornar esse aprendizado disponível para uso. O autor ainda afirma que uma rede neural se assemelha ao cérebro em dois aspectos: (a) o conhecimento da RNA ocorre através de um processo de aprendizagem e (b) pesos sinápticos são utilizados para armazenar o conhecimento adquirido.

Haykin (2001) apresentou algumas propriedades úteis que uma RNA pode oferecer, entre elas, a capacidade de trabalhar com problemas não lineares; e a adaptabilidade dos pesos sinápticos, que vai garantir o aprendizado e mapeamento dos dados de entrada e saída do problema.

A maior aplicação das RNAs é no campo da Inteligência Artificial, onde uma máquina é ensinada a reconhecer padrões por meio de um conjunto de dados e se comportar de maneira adequada. No âmbito da Otimização Estrutural, as RNAs podem ser utilizadas com o intuito de reproduzir o comportamento de uma etapa do processo, como na substituição da análise por elementos finitos.

4.4.1 Histórico

O estudo de redes neurais artificiais teve início em 1943 com McCullosh e Pitts. De acordo com Haykin (2001) e Braga, Carvalho e Ludermir (2007), MacCullosh e Pitts publicaram um trabalho apresentando o primeiro modelo artificial de um neurônio biológico. Na publicação, os pesquisadores buscaram descrever o modelo artificial do neurônio e suas capacidades computacionais, afirmam Braga, Carvalho e Ludermir (2007), sem se deterem tanto ao aprendizado da rede.

Somente em 1949, com o estudo de Hebb, o aprendizado começou a ser estudado. Haykin (2001) e Braga, Carvalho e Ludermir (2007), afirmam que Hebb apresentou uma formulação de uma regra de aprendizado, afirmando que o neurônio de McCullosh e Pitts poderia aprender por meio da variação dos pesos de entrada do neurônio.

Segundo Coppin (2013) e Braga, Carvalho e Ludermir (2007), Roseblatt, em 1958, propôs um novo modelo neuronal denominado perceptron, demonstrando que a regra de aprendizado associada ao neurônio de McCullosh e Pitts, poderia ser treinada para classificar alguns padrões.

Em 1969 Minsky e Papert apontaram falhas no modelo de Roseblatt. Segundo os autores, tal modelo só funcionava para problemas linearmente separáveis. Eles comprovaram isso aplicando o perceptron simples em problema OU-exclusivo.

Hopfield, em 1982, apresenta o perceptron de múltiplas camadas (do Inglês,

Multilayer Perceptron – MLP), afirmam Braga, Carvalho e Ludermir (2007). Esse modelo melhorou o perceptron simples (de uma camada), possibilitando a resolução de problemas não lineares.

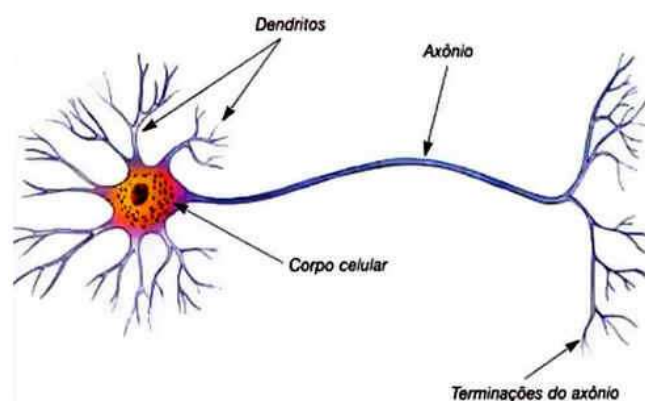
Diversos estudos sobre MLP foram realizados, implementado e melhorando sua forma de aprendizado e criando algoritmos mais eficientes. Atualmente as RNAs são usadas para as mais diversas aplicações, como já foi mencionado.

Na engenharia estrutural encontra-se, por exemplo, trabalhos na área de Patologia e Otimização. Kapania e Liu (1998) criaram um modelo contínuo de alta fidelidade para estruturas aeroespaciais usando redes neurais. Lastiri e Pauletti (2004) estudaram as RNAs para dimensionamento de viga em concreto armado e na análise de chapa com furo. Barbosa *et al.* (2005) usaram RNAs em conjunto com o método de Monte Carlo para resolver problemas de confiabilidade estrutural. Lopes *et al.* (2010) analisaram essa técnica na identificação de falhas em compósitos laminados de alumínio e epóxi reforçados por fibras de vidro. Sliseris e Rocens (2012) aplicaram as RNAs na otimização de placas compósitas com rigidez variável discreta.

4.4.2 *Neurônios biológicos x Neurônios Artificiais*

As RNAs baseiam-se completamente no neurônio biológico, na estrutura de suas conexões e no comportamento conjunto desses elementos, afirmam Braga, Carvalho e Ludermir (2007). O neurônio biológico, apresentado na Figura 25, é uma unidade do sistema neurológico, que se conecta com outros neurônios por meio de sinapses.

Figura 25 – Representação de um neurônio biológico.



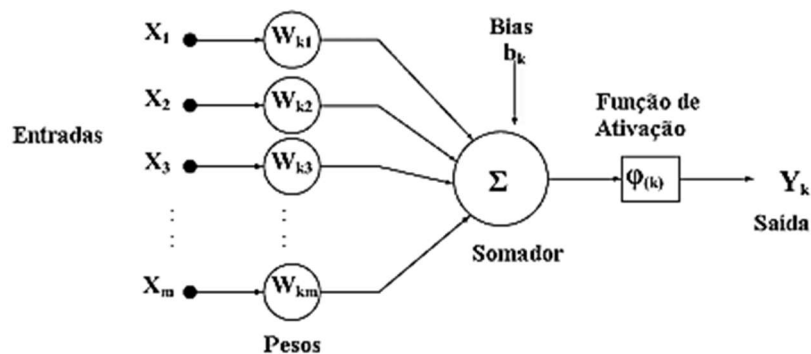
Fonte: Google imagens.

Como pode ser visto, o neurônio se divide em três partes: dendritos, corpo celular (ou soma) e axônio. De acordo com Coppin (2013) e Braga, Carvalho e Ludermir (2007), o estímulo/sinal vai chegar ao dendrito e quando este ultrapassa o limiar do neurônio, um impulso

elétrico (potencial de ação) é produzido e propagado pelo axônio. De maneira simples, o que vai ocorrer é uma ativação desse neurônio que vai passar a informação adiante.

Analogamente, o neurônio artificial, Figura 26, vai receber estímulos/sinais (dados de entrada) que serão multiplicados pelos respectivos pesos e depois somados. Se essa soma for maior que o limiar do neurônio, o neurônio será ativado e resultará em um impulso de saída (dados de saída).

Figura 26 – Modelo de um neurônio artificial.



Fonte: Lopes *et al.* (2010).

Os três componentes básicos do neurônio artificial, mostrados na Figura 26, são: o conjunto de sinapses que envolvem o sinal de entrada x_m e o peso w_{km} do neurônio; o somador, que tem o objetivo de somar os sinais de entrada multiplicados pelos respectivos pesos e por fim, a função de ativação para limitar a saída de um neurônio (HAYKIN, 2001).

De acordo com Haykin (2001), a resposta (y_k) de um neurônio k é determinado por meio da seguinte equação:

$$y_k = \varphi(u_k + b_k), \quad (48)$$

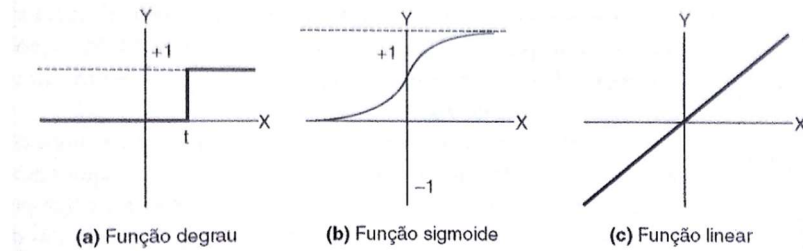
e

$$u_k = \sum_{j=1}^m w_{kj} x_j. \quad (49)$$

Onde x_j são as variáveis de entrada, w_{kj} são os pesos, b_k é o bias e tem o efeito de aumentar ou diminuir a entrada da função de ativação e $\varphi(\cdot)$ é a função de ativação.

As funções de ativação definirão a saída do neurônio a partir dos vetores \mathbf{w} e \mathbf{x} . As mais comuns são a função linear, função degrau e a função sigmoide (BRAGA *et al.*, 2007; COPPIN, 2013; HAYKIN, 2001), representadas na Figura 27.

Figura 27 – Representação das funções de ativação.



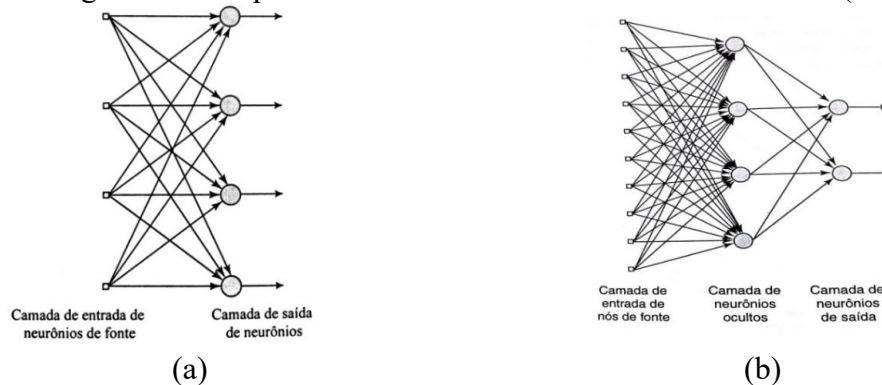
Fonte: Coppin (2013).

4.4.3 Arquitetura de uma RNA

As RNAs possuem em geral três tipos de arquitetura: As redes alimentadas diretamente (feedforward) com uma camada; Rede alimentada diretamente com múltiplas camadas; e as redes recorrentes (HAYKIN, 2001).

A arquitetura mais simples é apresentada na Figura 28 (a). Por ser de uma única camada (camada de saída), possui restrições quanto à complexidade do problema, sendo aplicada apenas em problemas linearmente separáveis. Na Figura 28 (b), é possível observar uma rede alimentada diretamente com múltiplas camadas, uma camada de entrada, uma de saída e uma camada responsável por intervir entre a entrada dos dados e a saída do neurônio, chamada de camada oculta (HAYKIN, 2001). Segundo Braga, Carvalho e Ludermir (2007) a camada oculta (também chamada de intermediária) garante uma maior capacidade computacional à RNA, pois consegue trabalhar com problemas não-lineares.

Figura 28 – Arquitetura de redes neurais alimentadas adiante (Feedforward).



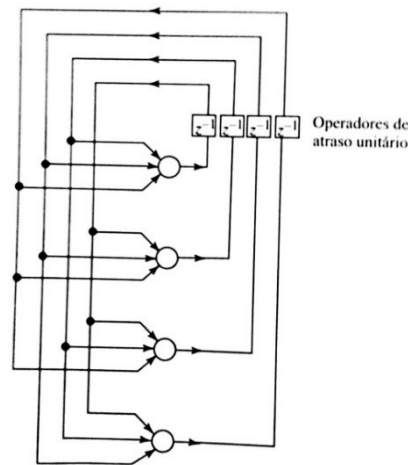
Fonte: Haykin, 2001.

As duas redes apresentadas na Figura 22 são exemplos de, respectivamente, um perceptron simples e uma rede de múltiplas camadas (*Multilayer Perceptron* – MLP), o treinamento dessas redes será explicado a seguir.

Por fim, têm-se as redes recorrentes, que diferente das redes feedforward, apresenta

pelo menos um laço de realimentação. Sua estrutura é apresentada na Figura 29.

Figura 29 – Rede neural com arquitetura recorrente.



Fonte: Haykin (2001).

Em uma rede de realimentação, a saída de um neurônio realimenta a entrada dos outros neurônios. Na rede de auto-realimentação, a saída de um neurônio realimenta sua própria entrada, afirma Haykin (2001). Braga, Carvalho e Ludermir (2007) explica que uma rede recorrente é útil na resolução de problemas que envolvam processamento temporal, como a previsão de eventos futuros.

4.4.4 Processo de aprendizado de um perceptron simples

O aprendizado de um neurônio, de acordo com Braga, Carvalho e Ludermir (2007), está relacionado com a melhoria do desempenho segundo algum critério, por exemplo, a redução do erro de aproximação a medida que o neurônio vai melhorando o desempenho. Como já foi comentado anteriormente, o aprendizado pode ser supervisionado ou não supervisionado.

Em uma rede neural artificial, o processo de aprendizagem implica na busca dos valores dos pesos que forneçam as melhores respostas. Para Braga, Carvalho e Ludermir (2007)), o vetor peso (\mathbf{w}) no instante $t+1$ é dado por,

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w}(t), \quad (50)$$

onde, $\mathbf{w}(t)$ é o vetor peso no tempo t e $\Delta \mathbf{w}(t)$ é o ajuste aplicado ao peso. A maneira como será calculado esse ajuste definirá o algoritmo de aprendizado.

Para um modelo de um Perceptron simples (Figura 26), tem-se como principais

dados o vetor de entrada, \mathbf{x} , o vetor de pesos, \mathbf{w} , e o bias, \mathbf{b} . Dessa forma, a ativação do neurônio será dada pelo produto desses vetores e a condição de disparo será (BRAGA; CARVALHO; LUDERMIR, 2007):

$$w_{k1}x_1 + w_{k2}x_2 + w_{k3}x_3 + \dots - b = 0. \quad (51)$$

Por conveniência, adota-se um peso $w_0 = -1$ e uma entrada $x_0 = b$. Assim, a condição de disparo, na forma matricial, será:

$$w_{k1}x_1 + w_{k2}x_2 + w_{k3}x_3 + \dots + w_{k0}x_0 = \mathbf{w}^T \mathbf{x}. \quad (52)$$

Onde, $\mathbf{w} = \{w_0 = -1, w_1, \dots, w_k\}$ e $\mathbf{x} = \{x_0 = b, x_1, \dots, x_k\}$.

Braga, Carvalho e Ludermir (2007) afirma que, de acordo com o Teorema de convergência, a atualização dos pesos, para problemas linearmente separáveis, pode ser dada por:

$$w(t+1) = w(t) + \alpha e x(t) \quad (53)$$

sendo e o erro de aproximação e α a taxa de aprendizado.

4.4.5 Processo de aprendizado de uma rede MLP

Um dos algoritmos mais utilizados na literatura, para treinamento de MLP, é o algoritmo de retropropagação (*Backpropagation*). Este, de acordo com Barbosa *et al.* (2005), baseia-se no princípio do aprendizado por correção do erro. Para Coppin (2013), a ideia do algoritmo é a atribuição de responsabilidade para os pesos de cada neurônio. A regra de aprendizado mais comum é pela minimização do erro quadrático médio, apresentado na Equação (37), reescrevendo esta equação em termos dos pesos, temos:

$$\bar{e}_d = \frac{1}{n} \sum (f(\mathbf{w}^T \mathbf{x}) - \hat{y}^{(i)})^2. \quad (54)$$

Sabendo que o intuito da rede neural é reduzir esse erro pode-se utilizar técnicas como a regra delta generalizada (também conhecida por gradiente descendente com momento), que busca minimizar o erro quadrático (ou outra medida de erro) que é expresso em função dos pesos, assim, chegará a um vetor de pesos que conseguirão calibrar adequadamente a rede, encerrando o processo de treinamento (BARBOSA *et al.*, 2005). Coppin (2013) explica que essa regra é usada para melhorar o desempenho do treinamento. Incluindo um fator de momento na equação que modifica os pesos. A nova equação seria (BARBOSA *et al.*, 2005):

$$\Delta \mathbf{w}(t) = \beta \Delta \mathbf{w}(t-1) + \alpha \frac{\partial \bar{e}_d}{\partial \mathbf{w}}, \quad (55)$$

substituindo na Equação (50), tem-se a nova expressão para atualização dos pesos:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \beta \Delta \mathbf{w}(t-1) + \alpha \frac{\partial \bar{e}_d}{\partial \mathbf{w}}, \quad (56)$$

onde, α é o passo de aprendizagem e β é o valor do fator de momento, cujo domínio é $[0,1]$

No processo de treinamento a escolha da amostra é muito importante visto que, uma amostra com poucos pontos pode resultar em um baixo aprendizado, por sua vez, uma amostra com muitos pontos pode superdimensionar o aprendizado da rede, esse efeito é chamado de overfitting (BRAGA; CARVALHO; LUDERMIR, 2007). O overfitting ocorre quando o número de neurônios na camada escondida é grande para o número de pontos na amostra, reproduzindo fielmente o comportamento dos dados. Dessa forma a rede, reconhece o padrão dos dados de treinamento, mas não reconhecerá pontos fora do padrão amostral. Além disso, nesse algoritmo, a escolha do passo de aprendizado é muito importante, visto que, um passo pequeno pode fazer a rede convergir muito lentamente e um passo muito grande pode deixar o algoritmo instável.

4.5 Funções de Base Radial

Assim como as Redes Neurais Artificiais, as funções de base radial (*Radial Basis Functions*, RBF) podem ser utilizadas para construção de modelos aproximados, substituindo os modelos computacionais de alta fidelidade por funções aproximadas através de interpolações. Desse modo a RBF utiliza-se de uma superfície multidimensional para interpolar outros pontos que estão fora do conjunto de treino, de forma a aproximar respostas de sistemas de engenharia que envolva simulações numéricas custosas (YIN *et al.*, 2016).

Uma função radial pode ser descrita como sendo uma função que possui simetria radial, dependendo apenas da distância, $r = \|\mathbf{x} - \mathbf{x}_i\|$, entre o centro da função, \mathbf{x}_i , e o ponto genérico, \mathbf{x} .

O problema da interpolação, para caso de interpolação multivariável em alta dimensão, descrito em Mulgrew (1996), e também encontrado em Haykin (2001), pode ser formulado da seguinte maneira:

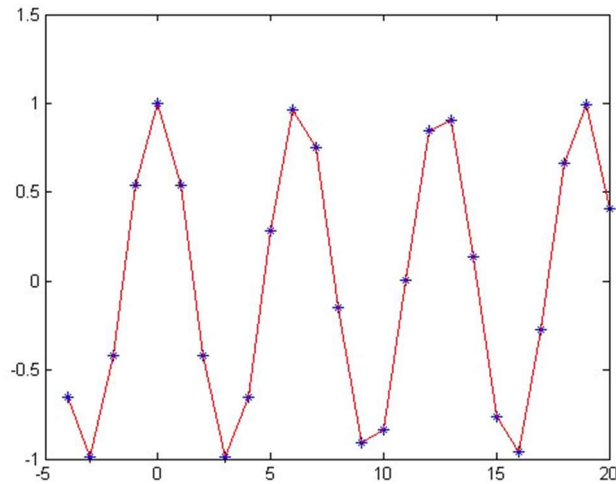
Dados um conjunto de N diferentes pontos $\{\mathbf{x}_i \in \mathbb{R}^m \mid i = 1, 2, \dots, N\}$ e um conjunto de N números reais $\{\mathbf{d}_i \in \mathbb{R} \mid i = 1, 2, \dots, N\}$. Procure uma função $F : \mathbb{R}^N \rightarrow \mathbb{R}^1$, tal que satisfaça a condição de interpolação, dada por:

$$F(\mathbf{x}_i) = \mathbf{d}_i. \quad (57)$$

Sendo esse, um problema de interpolação estrita. Isso significa que a superfície de interpolação

necessariamente passa por todos os pontos de dados de treinamento. A Figura 30 apresenta uma aproximação por RBF de uma função senoidal, onde os pontos de amostragem estão marcados por um *.

Figura 30 – Aproximando de uma função senoidal usando RBF.



Fonte: Elaborada pelo autor.

O uso das RBFs consiste em escolher uma função que tenha a forma (MULGREW, 1996 e HAYKIN, 2001):

$$F(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (58)$$

Onde $\phi(\|\mathbf{x} - \mathbf{x}_i\|)$ com $i=1,2,\dots,N$, é um conjunto com N funções RBF, \mathbf{x}_i são os centros das RBFs e w_i os pesos.

Substituindo a Equação (57) na Equação (58) chega-se na seguinte equação:

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}. \quad (59)$$

Escrevendo esta equação na forma matricial:

$$\Phi \mathbf{w} = \mathbf{d}. \quad (60)$$

A solução deste sistema de equações lineares fornece o vetor de pesos.

Para que a solução da Equação (60) seja possível, é necessário que a matriz de interpolação seja não singular e que sua inversa exista. Micchelli (1986) publicou um teorema que fornece uma solução para a matriz de interpolação:

Seja $\{\mathbf{x}_{ij}\}_{i,j=1}^N$ um conjunto de pontos reais. Então a matriz de interpolação, cujo ij -ésimo

elemento é $\varphi(\|x_i - x_j\|)$, é não singular.

Algumas funções que atendem esse teorema são apresentadas em Mulgrew (1996), Yin *et al.* (2016) e Haykin (2001). As mais utilizadas são a função Gaussiana:

$$\varphi(r) = \exp\left(-\frac{r^2}{2c^2}\right), \quad (61)$$

a função multiquadrática:

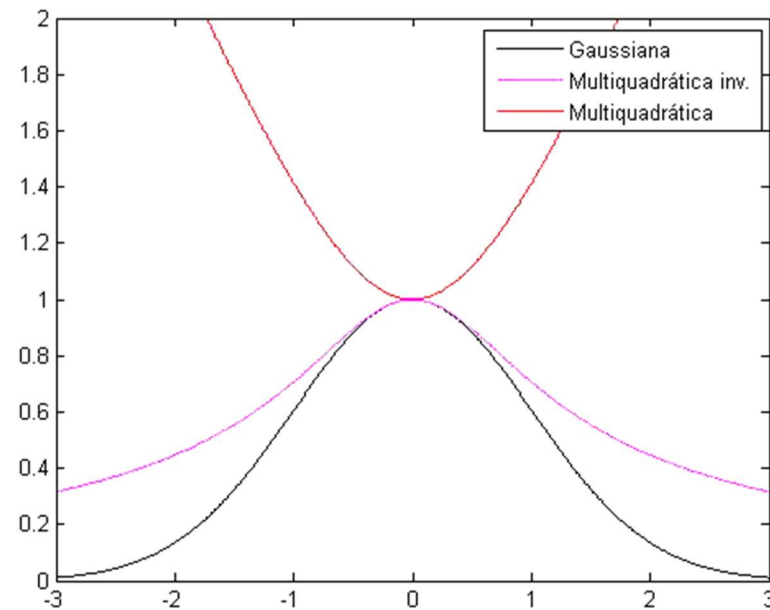
$$\varphi(r) = \sqrt{r^2 + c^2}, \quad (62)$$

e a função multiquadrática inversa:

$$\varphi(r) = \frac{1}{\sqrt{r^2 + c^2}}. \quad (63)$$

Sendo c um parâmetro que controla o espalhamento de cada função de base.

Figura 31 – Funções de base radial.

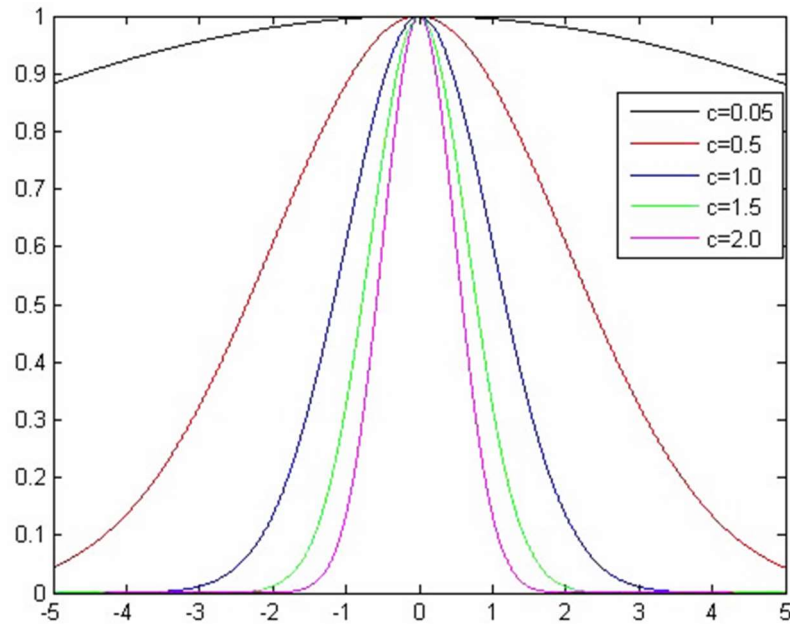


Fonte: Elaborada pelo autor.

A

Figura 31 ilustra o comportamento gráfico dessas três funções de base radial descritas anteriormente e a Figura 32 mostra a influência do parâmetro c na função Gaussiana, mais adiante serão apresentadas maneiras de estimar esse parâmetro. Observando a Figura 32, nota-se que para valores muito pequenos de c , tem-se uma função de interpolação menos suave.

Figura 32 – Função Gaussiana para diferentes valores de c .



Fonte: Elaborada pelo autor.

Para que a matriz de interpolação seja não-singular é suficiente que as funções radiais sejam não-singulares, ou seja, os pontos $\{\mathbf{x}_i\}_{i=1}^N$ precisam ser todos distintos. De acordo com Orr (1996), as funções gaussianas tendem a ser locais, garantindo resposta mais significativa em regiões próximas ao centro e são mais utilizadas que outras funções como a multiquadrática, que fornece respostas globais.

O uso da interpolação pode não ser a melhor opção para a construção de um modelo substituto, visto que para problemas com número de pontos na amostra for maior que o número de graus de liberdade, a generalização pode ser pobre. Outro ponto que merece atenção é o fato de que para pontos distintos a matriz de interpolação da função Gaussiana é não singular, porém, quando esses pontos são relativamente próximos esta matriz se torna quase singular, dificultando o cálculo dos pesos. Este problema pode ser resolvido com o uso de um termo de regularização.

Considerando o coeficiente de regularização (λ), o treinamento de uma rede RBF é caracterizado pelo problema:

$$E = \sum_{i=1}^N (y_i - F(x_i))^2 + \sum_{j=1}^N \lambda_j w_j^2 \rightarrow \min. \quad (64)$$

Assim, a determinação dos pesos corresponde a uma aplicação do método dos mínimos quadrados. A solução para a Equação (64) é (ORR, 1996):

$$\mathbf{w} = (\mathbf{\Phi}^T \mathbf{\Phi} + \lambda \mathbf{I})^{-1} \mathbf{\Phi}^T \mathbf{y}. \quad (65)$$

Onde,

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_m(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_m(x_2) \\ \vdots & \cdots & \ddots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \cdots & \varphi_m(x_m) \end{bmatrix},$$

$\mathbf{y} = (y_1 \ y_2 \ \cdots \ y_m)^T$ e λ é o termo de regularização.

O parâmetro referente à largura da função de base radial (em algumas referências chamados de *spread*), pode ser estimado por tentativa e erro, por validação cruzada (FORRESTER,) ou por meio de expressões simples. A expressão proposta por Nakayama *et al.* (2002) é:

$$c = \frac{d_{max}}{\sqrt[n]{nm}}, \quad (66)$$

onde n é o número de variáveis, m é o número de centros (i.e. RBFs) e d_{max} é a maior distância entre estes centros. Outra alternativa (KITAYAMA; ARAKAWA; YAMAZAKI, 2011) é:

$$c = \frac{d_{max}}{\sqrt{n} \sqrt[n]{m}}. \quad (67)$$

É importante frisar que estas duas equações foram propostas para uma função Gaussiana escrita como:

$$\varphi(r) = \exp\left(-\frac{r^2}{c^2}\right). \quad (68)$$

Finalmente, a expressão sugerida por Haykin (2001) para uso com a Equação (61) é:

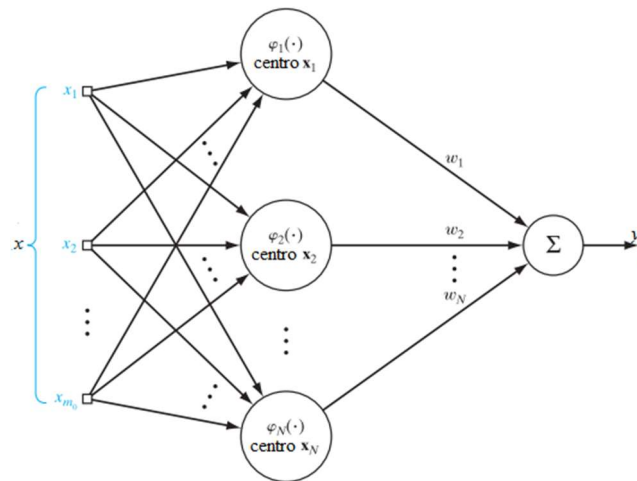
$$c = \frac{d_{max}}{\sqrt{2k}}. \quad (69)$$

onde k é o número de centros. No caso de uma RBF interpoladora (RBFi), o número de centros será igual ao número de pontos na amostra.

4.5.1 Redes de funções de bases radiais

Uma Rede de Funções de Bases Radiais (*Radial Basis Function Network* – RBFN) é caracterizada pela união das técnicas explicadas anteriormente. A estrutura de uma RBFN, de interpolação exata ($N=m_0$), pode ser observada na Figura 33.

Figura 33 – Estrutura de uma rede RBF.

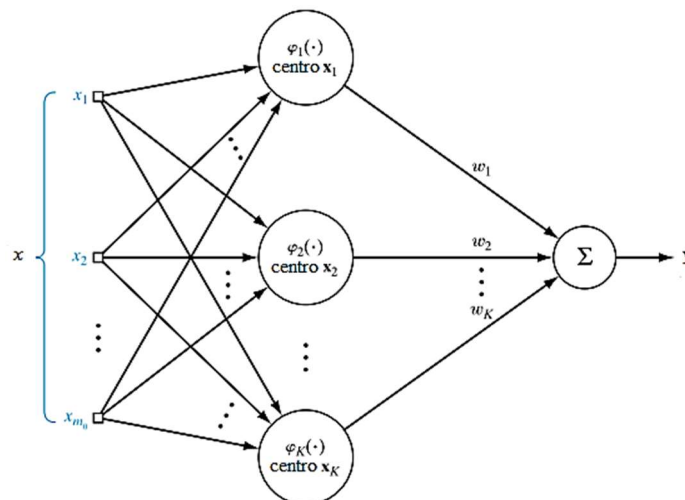


Fonte: Adaptado de Haykin (2001).

Segundo Haykin (2001), há uma questão prática que precisa de atenção: uma rede com camada escondida do mesmo tamanho da camada de entrada (N é o número de pontos no conjunto de treinamento) pode ser considerado um desperdício de recursos computacionais, principalmente para problemas com grandes amostras de treinamento.

O autor ainda afirma que quando a camada escondida da rede RBF é especificada por uma função Gaussiana, há redundância de neurônios na camada escondida em virtude da redundância que pode inerentemente existir na amostra de treinamento. O autor propõe que, em situações desse tipo, o tamanho da camada escondida seja uma fração do tamanho da amostra de treinamento, dessa forma o número de neurônios será menor. A rede com essa nova camada é apresentada na Figura 34.

Figura 34 – Nova estrutura de uma rede RBF.



Fonte: Adaptada de Haykin (2001).

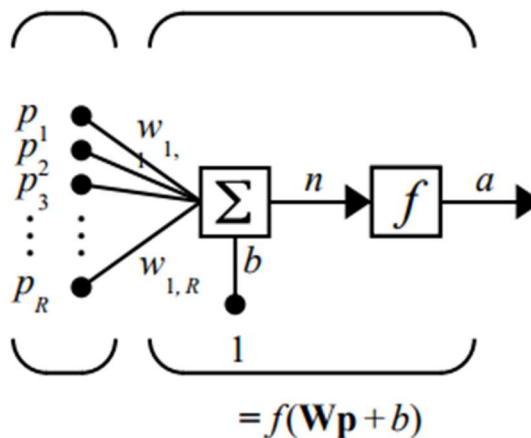
Observa-se que a rede da Figura 34 possui $K < N$ pontos na camada escondida, essa modificação é feita justamente para evitar redundância ao se utilizar a função gaussiana, como explica Haykin (2001). Um ponto em comum entre as redes e que diferem de uma MLP, é o treinamento. No caso das redes de base radial não há envolvimento de retropropagação do erro.

4.6 Implementação computacional

O software MATLAB possui uma caixa de ferramentas (*toolbox*) relacionada às redes neurais (*Neural Network Toolbox*) bastante completa. O toolbox fornece algoritmos, modelos pré-treinados e aplicativos para criar, treinar, visualizar e simular redes neurais. Com esta ferramenta, é possível realizar classificação de dados, regressão, agrupamento, previsão de séries temporais e modelagem de sistemas dinâmicos. Vale ressaltar que a caixa de ferramentas oferece diversos tipos de redes, treinamentos e funções de ativação, porém nos deteremos às MLPs e RBFNs, modelos usados no trabalho.

Para uma RNA, o modelo de neurônio é do tipo apresentado na Figura 35. Observe que nesse caso, o bias (b) é incorporado em um somatório e a saída é resultado da função f , chamada de função de ativação. Tais funções, mais comuns no MATLAB, são: a sigmoide (*logsig*), a linear (*purelin*) e a tangente hiperbólica (*tansig*).

Figura 35 – Modelo de neurônio de uma RNA.



Fonte: MATHWORKS (2017).

Nas versões mais recentes do MATLAB, como a versão R2010b, a função *newff*, responsável por criar uma MLP alimentada adiante, tornou-se obsoleta, podendo ser utilizado em seu lugar a função *feedforwardnet*(), que permite a criação de uma rede com mais de uma camada escondida, ou a função *fitnet*(), essa função constrói redes com apenas uma camada

escondida. A rede usada nesse trabalho foi construída com a função a *fitnet()*. Sua sintaxe é:

```
net = fitnet(hiddenSizes)
net = fitnet(hiddenSizes,trainFcn)
```

onde *net* é o nome da rede, *hiddenSizes* define o número de neurônios na rede e *trainFcn* é qualquer uma das funções de treinamento. Entre as funções de treinamento, pode-se citar a *trainlm*, onde o treinamento é feito com base no algoritmo de Levenberg-Marquardt, a *traingd*, que aplica o gradiente descendente e a *trainbr* que treina redes regularizadas utilizando uma abordagem Bayesiana (MATHWORKS, 2017).

A regularização é realizada substituindo, no treinamento na rede, o erro médio quadrático (MSE) pela função:

$$E = (1-\gamma) \left(\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \right) + \gamma \left(\frac{1}{n} \sum_{j=1}^n \lambda_j w_j^2 \right) \quad (70)$$

onde γ é o parâmetro de regularização, N é o número de pontos da amostra e n é o número de neurônios da camada linear. É importante notar que o primeiro termo corresponde ao erro médio quadrático (MSE), logo $\gamma = 0$ corresponde a uma rede convencional (não regularizada). Além dos pesos, o algoritmo *trainbr* determina também o parâmetro de regularização ótimo.

A Figura 36 apresenta os comandos necessários para criação, treinamento e simulação de uma RNA. Nesse caso, a função de treinamento não foi definida, dessa forma a função utilizada é a padrão, Levenberg-Marquardt.

Figura 36 – Código MATLAB para construção de uma RNA.

```
% processo de criação de uma rede neural.
hiddenLayerSize = i;
RNA{i,1} = fitnet(hiddenLayerSize);

% Porcentagem da amostra usada em cada etapa (treino, validação, teste)
RNA{i,1}.divideParam.trainRatio = 100/100;
RNA{i,1}.divideParam.valRatio   = 0/100;
RNA{i,1}.divideParam.testRatio  = 0/100;

% treino da rede
RNA{i,1} = train(RNA{i,1}inputs,targets);

% simulação
simulaRNA = sim(RNA{i,1},xval_norm)';
```

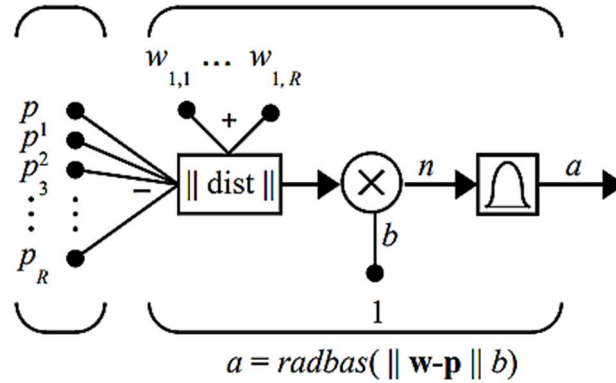
Fonte: Elaborada pelo autor.

Para o código da Figura 36, o número de neurônios está variando, conforme o valor adotado por i . A rede, $RNA\{i\}$, foi treinada com todos os pontos amostrais, sendo a simulação feita posteriormente com uma amostra menor e gerada aleatoriamente.

A estrutura do neurônio para uma RBFN é caracterizada pela presença de uma

função radial como função de ativação. Na Figura 37 pode-se ver a estrutura de um neurônio para uma RBFN.

Figura 37 – Estrutura de um neurônio para RBFN.



Fonte: MATHWORKS (2017).

Como já foi explicado, a caixa **dist** calcula a distância Euclidiana entre os pesos (\mathbf{w}) e os dados de entrada (\mathbf{p}). A saída do neurônio é o resultado de uma função radial, no MATLAB, essa função é da forma:

$$\text{radbas}(n) = e^{-n^2}. \quad (71)$$

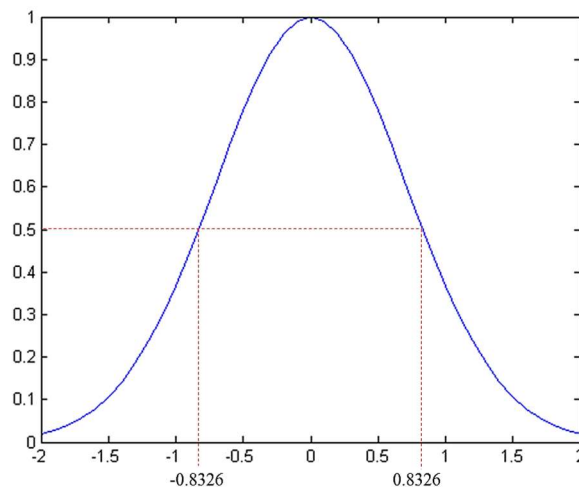
onde

$$n = \|\mathbf{w} - \mathbf{p}\| b \quad (72)$$

O comportamento desta função é expresso na

Figura 38.

Figura 38 – Comportamento da função $\text{radbas}(n)$.



Fonte: Elaborada pelo autor.

De acordo com a curva da

Figura 38, a saída da rede tende a 1 quando a distância entre os pesos e as entradas tendem a zero, ou seja, quando o vetor de entradas for igual ao vetor pesos, pois $n = \|\mathbf{w} - \mathbf{p}\|b = 0 \Rightarrow \mathbf{p} = \mathbf{w}$.

Outro ponto importante é que o bias influencia diretamente na sensibilidade da curva, pois ele define o valor da distância para os valores da função *radbas()*. Observe na função da Figura 37, o valor $n = 0,8326$ correspondente a uma redução de 50% no valor da função Gaussiana em relação ao centro. Partindo desse valor, temos:

$$\|\mathbf{w} - \mathbf{p}\|b = 0,8326 \Rightarrow \|\mathbf{w} - \mathbf{p}\| = 0,8326/b \quad (73)$$

Assim, se adotarmos um $b = 0,1$, qualquer valor de \mathbf{p} e \mathbf{w} que resulte na distância $8,326$ ($0,8326/b$), fornece uma saída igual a $0,5$.

Fazendo um comparativo entre as funções Gaussianas encontradas em Haykin (2001) e no MATLAB, e considerado que o vetor de pesos (\mathbf{w}) representa os centros da RBF, teremos:

$$e^{-r^2 b^2} = e^{-\left(\frac{r^2}{2\sigma}\right)} \quad (74)$$

Portanto,

$$b = \frac{1}{\sqrt{2}\sigma}. \quad (75)$$

No MATLAB não se fornece diretamente o valor do bias (b), mas o valor do parâmetro *SPREAD*, sendo a relação entre eles dada por:

$$b = 0,8236/SPREAD. \quad (76)$$

Sendo assim, é possível escrever o *SPREAD* em função do parâmetro σ utilizado na Eq. (61):

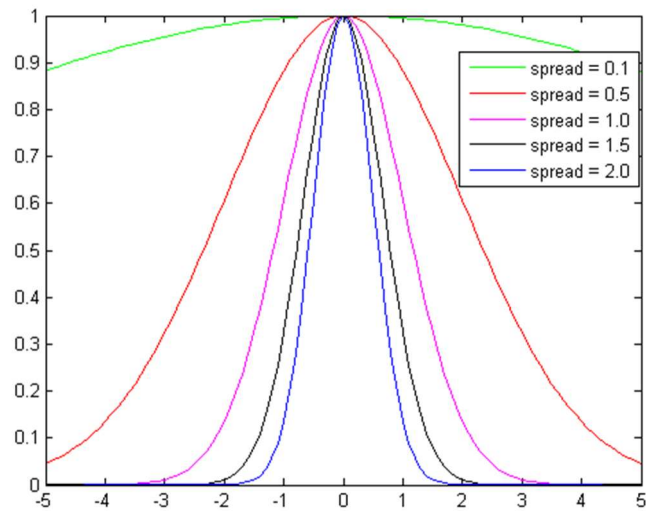
$$SPREAD = 1,165\sigma. \quad (77)$$

Quando comparado com o fator de espalhamento (c) utilizado na Eq. (68), o *SPREAD* corresponde a:

$$SPREAD = 0,8326c. \quad (78)$$

Como o próprio nome indica, o parâmetro *SPREAD* vai influenciar no espalhamento da função *radbas()* em torno do centro, como ilustrado na Figura 39.

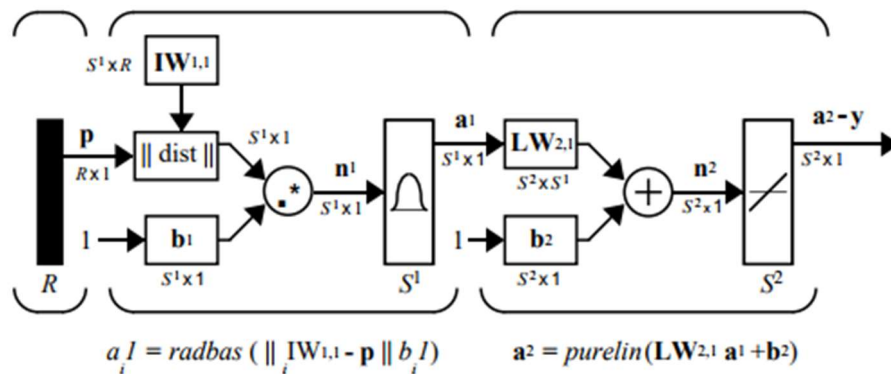
Figura 39 – Influência do *SPREAD* na função *radbas*().



Fonte: Elaborada pelo autor

A arquitetura de uma RBFN no MATLAB é apresentada na Figura 40. É importante notar que a saída da rede é uma função linear (*purelin*), essa função é usada nos problemas de regressão, tanto para a RNA quanto para a RBFN.

Figura 40 – Arquitetura de uma RRBF.



Fonte: MATHWORKS (2017).

É importante salientar que o MATLAB fornece a possibilidade de trabalhar com dois tipos de rede RBF. A primeira por interpolação exata,

Figura 41, onde o número de neurônios na camada escondida é fixado a um valor igual ao número de entradas (i.e. número de pontos da amostra).

Figura 41 – Comandos para construção de uma RBFN.

```
% construção e treinamento da rede
RBFfe{i,1} = newrbe(xnorm, ynorm, spread(i));
% simulação da rede
simulaRBFfe = sim(RBFfe{i,1}, xval_norm)';
```

Fonte: Elaborada pelo autor.

A segunda rede é uma melhoria da primeira, pois permite variação no número de neurônios, acrescentando um neurônio a cada treinamento, nessa rede, o valor do erro (MSE) é estimado e usado para treinamento, os comandos são apresentados na Figura 42. Observe que no caso da RBFN, a etapa de treinamento é realizada no processo de construção da rede.

Figura 42 – Comandos para construção de uma RBFN com variação de neurônios.

```
% construção e treinamento da rede
RBFm{i,1} = newrb(xnorm, ynorm, goal, spread(i), MN);
% simular a rede
simulaRBFm = sim(RBFm{i,1}, xval_norm)';
```

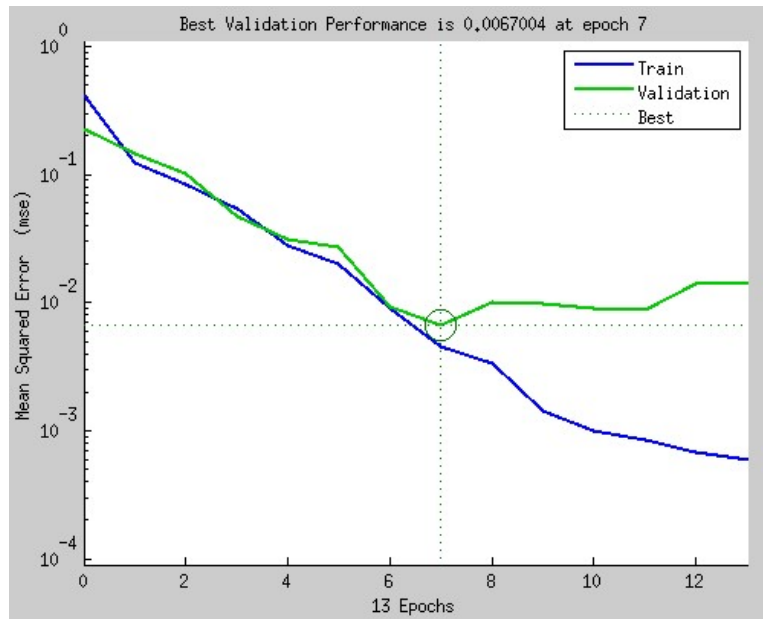
Fonte: Elaborada pelo autor.

Os parâmetros *MN* e *goal* na Figura 42, correspondem ao número máximo de neurônios e o valor do erro que deseja obter, respectivamente. Esses dois parâmetros são fornecidos pelo usuário.

Com já foi discutido, um problema comum que costuma ocorrer em redes neurais é o *overfitting*, para contorná-lo, o *toolbox* dispõe de algumas estratégias. Entre elas, a repetição do treinamento das redes, pois os parâmetros como peso e bias, sofrem alteração a cada treino, dessa forma, diferentes treinos geram diferentes resultados (essa estratégia não tem validade para as redes RBF).

Outra estratégia é da parada precoce (*Early Stopping*). Nesse caso, a validação da rede ocorre em paralelo com treinamento e os erros desses processos são computados, de modo que, se o erro do treinamento continua reduzindo e o erro de validação aumentando (atingindo uma quantidade estabelecida pelo usuário), o treinamento é cessado, pois há indícios de *overfitting*. Um exemplo desse processo é apresentado na Figura 43.

Figura 43 – Parada precoce de uma rede.



Fonte: Elaborada pelo autor.

Nota-se que a partir do melhor ponto (*Best*) o erro do treinamento continua caindo, enquanto o erro de validação começa a aumentar, esse comportamento é um indício de overfitting.

5 EXEMPLOS

Neste capítulo serão apresentados dois exemplos de otimização de laminados, onde aplica-se os conceitos de modelos substitutos no processo de otimização. O primeiro exemplo é usado para avaliar o comportamento de três técnicas amostrais, bem como a influência do tamanho da amostra na qualidade do modelo substituto. Em seguida aplica-se as informações pertinentes ao segundo problema, que tratará de uma casca cilíndrica, onde uma análise isogeométrica foi realizada para avaliar as respostas da estrutura para os projetos correspondentes aos pontos de treinamento.

Para a criação dos modelos substitutos, foram utilizadas técnicas de amostragem do tipo hipercubo latino (HCL), hipercubo latino otimizado (HCLO) e a sequência de Hammersley (SH). O número de pontos na amostragem mudou conforme o problema (i.e. número de variáveis). Na primeira parte do primeiro exemplo, adotou-se uma amostra com 30, 60 e 90 pontos. Nos demais casos optou-se pela determinação do tamanho da amostra por meio do coeficiente k , dado pela relação (AMOUZGAR; STRÖMBERG, 2016):

$$k = (m+1)(m+2)/2 \quad (79)$$

onde m é o número de variáveis. Este parâmetro corresponde ao número de coeficientes que definem de um polinômio quadrático, que corresponde à superfície de resposta mais simples para fins de otimização. Com k é possível calcular o número de pontos na amostra, fazendo, para problemas de baixa dimensão: a) $1,5k$ - amostra pequena; b) $2,5k$ - amostra média; c) $5k$ amostra grande (AMOUZGAR; STRÖMBERG, 2016).

Como já foi explicado, o treino e validação da rede foram feitos no MATLAB, gerando um arquivo *.mat*, que armazenava o modelo treinado. Por meio dos comandos: *engOpen()*; *engPutVariable()*; *engEvalString()*, foi possível, respectivamente, executar o MATLAB, fornecer os dados de entrada e executar o modelo substituto, tudo isso através do BIOS.

5.1 Placa laminada sob compressão biaxial no plano

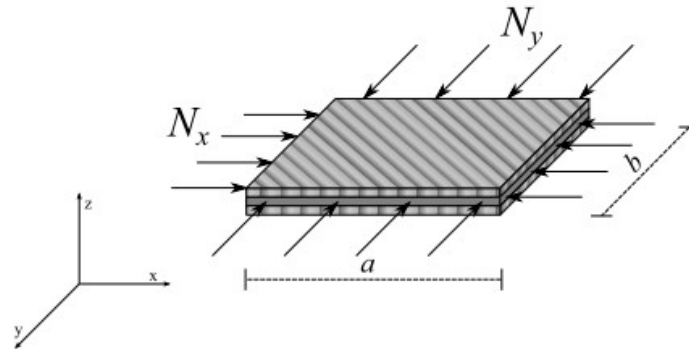
Este exemplo trata da otimização de uma placa laminada, simplesmente apoiada em todas as bordas, de grafite-epóxi submetida a carregamento biaxial mostrada na Figura 44. A geometria e as características do material são apresentadas na Tabela 4.

Tabela 4 – Propriedades geométricas e do material (grafite-epóxi).

Geometria		Material	
a (m)	0,508	E_1 (GPa)	181
b (m)	0,508	E_2 (GPa)	10,3
t (m)	1,27E-04	ν_{12}	0,28
		G_{12}	7,17

Fonte: Elaborada pelo autor.

Figura 44 – Placa laminada com carregamento biaxial.



Fonte: Barroso (2015).

Nesse exemplo, foram consideradas todas as lâminas com a mesma espessura e material. Dessa maneira, as variáveis de projeto são apenas as orientações das fibras de cada lâmina. O problema de otimização busca determinar o esquema de laminação que maximize a carga de flambagem da placa. Matematicamente, tem-se:

Determinar $\mathbf{x} = [\theta_1, \dots, \theta_{n_v}]$ que

$$\text{Maximize } \lambda_{cr} \quad (80)$$

$$\text{Com } 0 \leq \theta_i \leq 90,$$

onde n_v é o número variáveis do problema de otimização. Serão consideradas apenas laminações simétricas e balanceadas. Assim, o n_v será igual a $1/4$ do total de lamina.

Como a geometria, condições de contorno e carregamento deste exemplo são simples, o fator de carga crítica pode ser calculado de forma analítica utilizando a expressão (REDDY, 2004):

$$\lambda_{cr}(m, n) = \frac{\pi^2 \left[D_{11}m^4 + 2(D_{12} + 2D_{66})m^2n^2 + D_{22}n^4 \right]}{a^2 (m^2 + kn^2)}, \quad (81)$$

onde m e n são o número de meias ondas nos eixos x e y , respectivamente, e $k = N_y/N_x$. Desta forma, $k = 0$ implica em um carregamento uniaxial e $k = 1$ em um carregamento biaxial com

$N_x = N_y$. Esta solução é exata para laminação cross-ply e fornece bons resultados para laminação angle-ply com várias lâminas.

Devido à existência de solução analítica, este exemplo pode ser resolvido de forma eficiente sem a utilização de modelos substitutos. Por outro lado, esta eficiência facilita a realização de vários estudos importantes. O primeiro, é verificar se a filosofia dos modelos substitutos é consistente com as estruturas laminadas. O segundo, consiste em estudar como o tamanho da amostra e a técnica de amostragem utilizada influenciam no resultado final. Finalmente, o terceiro consiste em determinar qual técnica para construção do modelo substituto tem melhor desempenho desde a etapa de treinamento até os testes feitos com a placa (RNA ou RRBF).

Vale salientar que neste trabalho foram treinadas redes para cada laminado (i.e. para o laminado com 8 e 40 lâminas), pois o número de variáveis aumentou. Uma alternativa à esse método é apresentado em Koide, Ferreira e Luersen (2014), que estudaram os parâmetros de laminação como uma maneira de evitar treinar várias redes, visto que esses parâmetros permitem um número de entrada constante. Contudo, a abordagem utilizada neste trabalho é mais simples e intuitiva, por trabalhar diretamente com as mesmas variáveis utilizadas na otimização.

5.1.1 Placa com 8 lâminas

Este exemplo consiste na otimização de uma placa com 8 lâminas, o que corresponde a um problema de otimização com 2 variáveis de projeto. A otimização foi realizada utilizando o AG com 50 indivíduos e 40 gerações, uma taxa de cruzamento de 90%, a taxa de mutação de 5%. Para a seleção foi utilizado o *ranking*.

Para a otimização dessa placa foram treinadas três redes, uma RNA, uma rede RBF interpoladora (RBFi) e uma rede RBF regularizada (RBFr). Três técnicas de amostragem foram consideradas: Hipercubo Latino Padrão (HCL), a versão melhorada do Hipercubo Latino (HCLO) e Sequência de Hammersley (SH).

Para a otimização com AG, os resultados de laminação ótima, bem como suas respectivas cargas de flambagem, para os modelos substitutos, são apresentados a seguir. A construção do modelo substituto foi feita no MATLAB, utilizando os valores *default* dos parâmetros. Dessa forma, apenas o número de neurônios na camada oculta foi modificado.

A Tabela 5 apresenta os resultados obtidos no algoritmo de escolha da RNA. O parâmetro de escolha foi o valor no NRMSE, ou seja, a rede escolhida para cada amostra foi

aquela que apresentou o menor NRMSE. Para validação, foi considerado uma amostra de 10 pontos, gerados de maneira aleatória.

Tabela 5 – Valores do NRMSE da RNA.

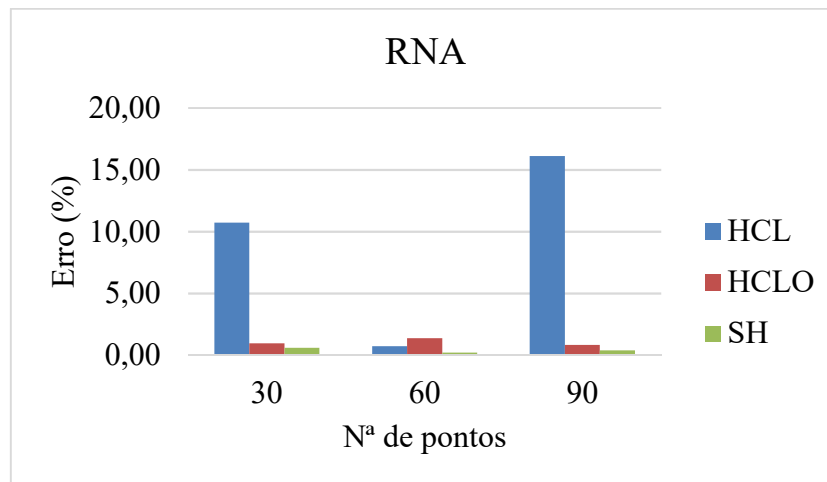
Nº de pontos	Tipo de amostra	Nº de neurônios				
		5	6	7	8	9
RNA30	HCL	0,401	0,532	0,827	1,656	0,472
	HCL0	0,042	0,109	0,115	0,707	0,120
	SH	0,054	0,156	0,075	0,028	0,038
RNA60	HCL	0,186	0,244	0,228	0,239	0,197
	HCL0	0,032	0,032	0,034	0,032	0,044
	SH	0,186	0,244	0,228	0,239	0,197
RNA90	HCL	0,243	0,410	0,227	0,359	0,358
	HCL0	0,057	0,071	0,053	0,057	0,059
	SH	0,048	0,050	0,039	0,044	0,040

Fonte: Elaborada pelo autor.

Nota-se que, nesse caso, aumentar o número de neurônios não implicou na redução do erro. Possivelmente, esse problema ocorreu por deficiência no treinamento da rede. Observe também que, na maioria dos casos, o HCL apresentou o pior desempenho entre as técnicas de amostragem utilizadas para uma RNA.

Os melhores resultados, para uma otimização baseada na combinação de AG com RNA, foram obtidos através da SH, que apresentou uma média de erro percentual de 0,38%. Enquanto para o HCL0 foi de 1,05% e para o HCL foi de 9,17%. A evolução dos erros percentuais para cada técnica amostral, considerando o número de pontos na amostra, é apresentado na Figura 45. Observa-se que os erros obtidos para o HCL, são consistentes com os resultados obtidos no treinamento da rede.

Figura 45 – Valor comparativo dos erros percentuais para a RNA.



Fonte: Elaborada pelo autor.

A Tabela 6 apresenta os resultados da otimização do modelo substituto (MS), percebe-se que as redes treinadas usando SH obtiveram sucesso tanto na aproximação da laminação ótima ($[\pm 45_2]_s$), quanto no valor da carga de flambagem (645,33 N), para o modelo exato (ME). Apesar de não ter obtido sucesso em todas as laminações, o HCLO apresentou erros aceitáveis para os problemas de engenharia. Já o HCL, apresentou o melhor resultado para uma rede com 60 pontos. Outras verificações como o valor da carga crítica do modelo substituto usando a laminação ótima (MSLOt).

Tabela 6 – Resultado para AG + RNA.

Rede	Amostra	Laminação	Neur ônios	Ncr*		MSLOt	Erro(%) MS-ME
				MS	ME		
RNA30	HCL	$[\pm 45 \ 90_2]_s$	5	714,5	645,3	478,589	10,72
	HCLO	$[\pm 45 \ 90_2]_s$	8	639,1	645,3	637,801	0,96
	SH	$[\pm 45_2]_s$	5	649,1	645,3	649,143	0,59
RNA60	HCL	$[\pm 45 \ 0_2]_s$	5	640,8	645,3	493,109	0,70
	HCLO	$[\pm 45_2]_s$	5	636,5	645,3	636,457	1,37
	SH	$[\pm 45_2]_s$	6	646,6	645,3	646,567	0,19
RBF90	HCL	$[\pm 45 \ 0_2]_s$	7	541,5	645,3	346,758	16,09
	HCLO	$[\pm 45_2]_s$	7	650,6	645,3	650,55	0,81
	SH	$[\pm 45_2]_s$	7	643,0	645,3	642,951	0,37

Fonte: Elaborada pelo autor.

A escolha da rede RBFi foi baseada na escolha do *spread*, avaliando o NRMSE para a amostra de validação. Adotou-se valores, para o *spread*, variando de 0,1 a 2,0 e computou a medida do erro, o resultado está apresentado na Tabela 7.

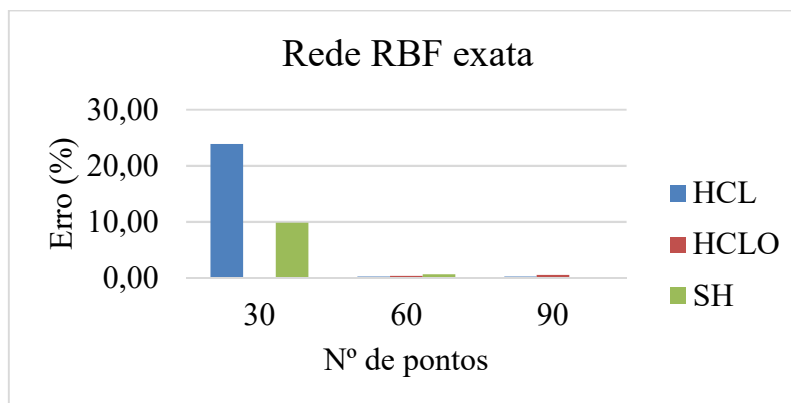
Tabela 7 – Valores do NRMSE para a RBFi.

Spread	30 pontos			60 pontos			90 pontos		
	HCL	HCLO	SH	HCL	HCLO	SH	HCL	HCLO	SH
0,1	0,079	0,187	0,083	0,067	0,084	0,086	0,042	0,045	0,045
0,2	0,113	0,135	0,202	0,078	0,023	0,038	2,565	0,426	0,925
0,3	0,104	0,070	0,530	0,326	0,098	0,420	39,663	2,048	2,909
0,4	0,119	0,200	0,689	0,843	0,473	1,321	6,695	4,387	5,334
0,5	0,123	0,363	1,083	1,659	0,756	1,903	0,457	0,147	3,476
0,6	0,119	0,431	1,799	3,699	0,992	2,427	0,382	0,081	0,262
0,7	0,113	0,519	2,865	1,605	0,581	2,860	0,145	0,110	0,194
0,8	0,108	0,607	2,200	1,077	0,285	2,890	0,046	0,036	0,061
0,9	0,105	0,678	2,321	0,138	0,074	0,991	0,038	0,031	0,055
1,0	0,103	0,766	2,411	0,188	0,194	0,304	0,022	0,039	0,047
1,1	0,102	0,817	2,479	0,063	0,190	0,225	0,063	0,040	0,060
1,2	0,101	0,797	2,532	0,104	0,078	0,208	0,033	0,061	0,056
1,3	0,101	0,827	2,574	0,137	0,228	0,212	0,044	0,128	0,088
1,4	0,102	0,853	2,607	0,033	0,151	0,138	0,057	0,133	0,075
1,5	0,102	0,875	2,634	0,050	0,123	0,086	0,121	0,068	0,087
1,6	0,103	1,337	2,655	0,040	0,124	0,096	0,075	0,065	0,074
1,7	0,036	1,353	2,675	0,038	0,186	0,069	0,045	0,065	0,066
1,8	0,0344	1,752	2,689	0,064	0,065	0,091	0,045	0,065	0,062
1,9	0,0341	1,770	2,709	0,060	0,061	0,091	0,045	0,066	0,061
2,0	0,0338	1,788	0,127	0,061	0,059	0,078	0,049	0,074	0,049

Fonte: Elaborada pelo autor.

Para a rede RBFi, a técnica amostral que apresentou melhores resultados foi a HCLO, com erro médio de 0,31%. Enquanto que a SH apresentou um erro médio de 3,50% e para o HCL, a média foi 6,60%. A Figura 46 apresenta um comparativo entre a evolução dos erros das técnicas abordadas em relação ao número de pontos na amostra.

Figura 46 – Comparativo entre os erros percentuais para a RBFi.



Fonte: Elaborada pelo autor.

Nota-se que, ao aumentar o número de pontos no conjunto de treinamento, o erro percentual referente às técnicas Hipercubo Latino e sequência de Hammersley reduz significativamente, enquanto que, para o Hipercubo Latino Otimizado, o erro apresentou um pequeno aumento. A redução do erro se dá pelo fato de ter mais pontos representado o espaço de projeto, sendo assim, o comportamento da função será melhor representado, facilitando o aprendizado da rede.

Considerando os resultados apresentados na Tabela 8, o HCLO apresentou melhor comportamento, aproximando tanto a laminação ótima quanto a carga de flambagem. Os outros dois métodos, HCL e SH, forneceram erros elevados para uma amostra com poucos pontos, porém, esse erro diminuiu a medida que o número de pontos amostrais aumentou, tal comportamento sustenta a ideia de que aumentar o número de pontos em uma amostra, permite que o modelo substituto reconheça melhor a função que será aproximada.

Tabela 8 – Resultado para AG + RBFi.

Rede	Amostra	Laminação	Neur ônios	Ncr*		MSLOt	Erro(%) LF-HF
				MS	ME		
RBFi30	HCL	[90 ₄] _s	30	799,46	645,3	659,20	23,88
	HCLO	[±45 ₂] _s	30	644,83	645,3	644,83	0,07
	SH	[±45 0 ₂] _s	30	581,87	645,3	565,89	9,83
RBFi60	HCL	[±45 ₂] _s	60	646,96	645,3	646,96	0,25
	HCLO	[±45 ₂] _s	60	647,55	645,3	647,55	0,35
	SH	[±45 ₂] _s	60	649,46	645,3	649,46	0,64
RBFi90	HCL	[±45 ₂] _s	90	646,96	645,3	646,96	0,25
	HCLO	[±45 ₂] _s	90	648,62	645,3	648,62	0,51
	SH	[±45 ₂] _s	90	645,46	645,3	645,46	0,02

Fonte: Elaborada pelo autor.

Para o treinamento da rede RBF regularizada (RBF_r) foi testado além dos spreads o número de neurônios para cada rede. Porém, ao executar o treinamento, a RBF_r faz sua própria avaliação do erro, através do MSE. Dessa forma, torna-se necessário determinar apenas o *spread*. Os resultados dos treinamentos foram organizados na Tabela 9.

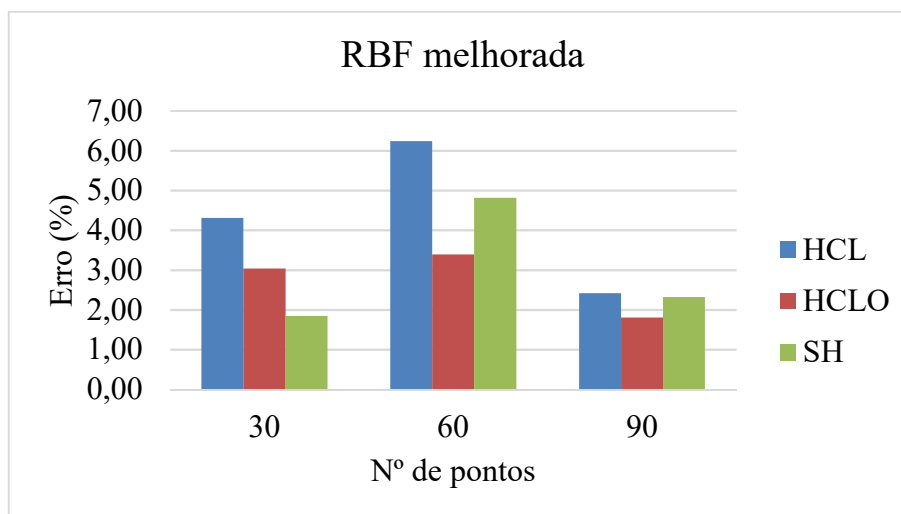
Tabela 9 – Valores do NRMSE para a RBFr.

Spread	30 pontos			60 pontos			90 pontos		
	HCL	HCLO	SH	HCL	HCLO	SH	HCL	HCLO	SH
0,1	0,273	0,261	0,132	0,126	0,143	0,095	0,114	0,154	0,093
0,2	0,209	0,138	0,089	0,100	0,176	0,070	0,112	0,102	0,081
0,3	0,172	0,151	0,099	0,118	0,132	0,072	0,171	0,084	0,125
0,4	0,266	0,367	0,134	0,118	0,100	0,107	0,137	0,108	0,093
0,5	0,151	0,581	0,177	0,095	0,094	0,207	0,119	0,136	0,098
0,6	0,225	0,411	0,228	0,114	0,088	0,153	0,111	0,160	0,107
0,7	0,158	0,791	0,585	0,122	0,157	0,164	0,105	0,137	0,078
0,8	0,186	0,653	0,143	0,169	0,076	0,106	0,121	0,146	0,096
0,9	0,147	0,612	0,657	0,127	0,130	0,105	0,086	0,120	0,133
1,0	0,153	0,713	0,532	0,095	0,125	0,164	0,116	0,150	0,094
1,1	0,122	0,711	0,292	0,154	0,112	0,127	0,141	0,108	0,068
1,2	0,166	1,215	0,285	0,108	0,158	0,355	0,082	0,121	0,149
1,3	0,127	0,759	0,339	0,173	0,067	0,113	0,127	0,120	0,189
1,4	0,117	0,767	0,167	0,119	0,067	0,101	0,049	0,124	0,144
1,5	0,138	0,866	0,131	0,133	0,060	0,234	0,152	0,098	0,112
1,6	0,137	1,562	0,386	0,133	0,106	0,102	0,234	0,166	0,132
1,7	0,137	1,572	0,371	0,085	0,119	0,151	0,209	0,131	0,091
1,8	0,136	1,367	1,128	0,116	0,089	0,230	0,134	0,077	0,090
1,9	0,136	1,377	0,096	0,217	0,086	0,123	0,275	0,077	0,075
2,0	0,136	1,166	0,331	0,163	0,180	0,118	0,169	0,127	0,126

Fonte: Elaborada pelo autor.

Como esperado, o NRMSE diminuiu com o aumento de pontos amostrais. Os resultados obtidos para essa técnica foram mais consistentes, apresentado pouca variação no erro percentual. Nesse caso, os melhores resultados foram obtidos pela rede treinada com a amostra obtida pelo HCLO, com uma média do erro de 2,75%, enquanto a SH obteve um erro médio de 2,99% e o HCL chegou à 4,32%. A variação do erro percentual é apresentada Figura 47.

Figura 47 – Comparativo entre os erros percentuais para a RBFr.



Fonte: Elaborada pelo autor.

Apesar da pouca variação nos erros percentuais, a rede treinada com amostras geradas pelo HCLO não aproximou a laminação ótima para uma amostra com 60 pontos. Vale notar também que a quantidade de neurônios utilizados foi bem inferiores à quantidade de neurônios para a rede interpoladora (RBFi), que utiliza a mesma quantidade de pontos da amostra.

Observando a Tabela 10, é possível verificar que o melhor resultado foi obtido pela rede treinada com o HCLO e com 90 pontos amostrais. Isso pode ser explicado pelo fato que, ao aumentar o número de pontos amostrais, o modelo passa a ter mais informações sobre o espaço de projeto.

Tabela 10 – Resultado para AG + RBFr.

Rede	Amostra	Laminação	Neurônios	Ncr*		MSLOt	Erro(%) LF-HF
				MS	ME		
RBFr30	HCL	$[0_2 \pm 45]_s$	18	617,5	645,3	597,42	4,30
	HCLO	$[\pm 45_2]_s$	14	625,7	645,3	625,69	3,04
	SH	$[0_2 \pm 45]_s$	14	633,4	645,3	603,41	1,84
RBFr60	HCL	$[\pm 45_2]_s$	24	605,1	645,3	605,10	6,24
	HCLO	$[\pm 45 \ 90_2]_s$	25	667,2	645,3	662,65	3,39
	SH	$[\pm 45_2]_s$	11	614,3	645,3	614,30	4,82
RBFr90	HCL	$[\pm 45_2]_s$	26	660,9	645,3	660,91	2,42
	HCLO	$[\pm 45_2]_s$	25	633,7	645,3	633,66	1,81
	SH	$[\pm 45_2]_s$	22	630,4	645,3	630,40	2,32

Fonte: Elaborada pelo autor.

De modo geral, as técnicas HCLO e SH, utilizadas para gerar a amostra de treinamento, apresentaram melhores resultados, a primeira se mostrou superior quando trabalhamos com uma rede de interpolação (RBFi), e a segunda quando se trabalha com RNA. Quanto ao tamanho do espaço amostral, verificou-se que aumentando o número de pontos, os resultados apresentaram melhoras significativas.

5.1.2 Placa com 40 lâminas

Nesse caso, a placa possui as mesmas características da placa anterior, mas com um total de 40 lâminas. Por se tratar de um laminado balanceado-simétrico, o problema passa a ter 10 variáveis, o que representa um aumento significativo do espaço de projeto. Ressalta-se ainda, que os parâmetros de otimização permaneceram os mesmos.

Para o treinamento da rede neural, além do tamanho das amostras, o número de neurônios também mudou, de acordo com o menor valor da regressão. Os resultados obtidos pela RNA são apresentados na Tabela 11.

Tabela 11 – Resultado do treinamento para a RNA.

Tamanho da amostra	Técnica	Neurônios	R²
99	HCL	10	0,83
	HCLO	15	0,76
	SH	10	0,85
165	HCL	15	0,95
	HCLO	20	0,93
	SH	-	-
330	HCL	20	0,97
	HCLO	20	0,96
	SH	-	-

Fonte: Elaborada pelo autor.

Como esperado, valores do coeficiente de regressão (R^2) aumentaram conforme o tamanho da amostra aumentou. Além disso, como o aumento do tamanho da amostra, foi necessário aumentar o número de neurônios para melhorar o resultado do treinamento. Nenhuma rede treinada usando SH com 165 e 330 pontos forneceu um valor aceitável (nem próximo) para a regressão, por esse motivo não foi computada.

Os resultados da otimização obtidos para as RNA treinadas são apresentados na

Tabela 12. Os resultados foram comparados com o valor ótimo obtido pela função analítica, sendo $[\pm 45_{20}]_s$ a laminação ótima e $N_{cr} = 80665,8$ N, o valor ótimo da carga de flambagem.

Tabela 12 – Resultado da otimização RNA

Tamanho da amostra	Técnica	Laminação	Ncr	Diferença (%)
99	HCL	$[90_2 \pm 45_3 \ 90_4 \pm 45_3 \ 0_2]_s$	77719,99	3,65
	HCLO	$[\pm 45_4 \ 90_2 \ \pm 45_2 \ 0_4 \ \pm 45_2]_s$	86853,33	7,67
	SH	$[0_4 \ \pm 45 \ 90_2 \ 0_2 \ \pm 45 \ 90_2 \ \pm 45 \ 0_2 \ \pm 45]_s$	79634,00	1,28
165	HCL	$[\pm 45_{10}]_s$	79136,05	1,90
	HCLO	$[\pm 45_7 \ 90_2 \ \pm 45 \ 90_2]_s$	79988,41	0,84
	SH	-	-	-
330	HCL	$[\pm 45_9 \ 90_2]_s$	79554,17	1,38
	HCLO	$[\pm 45_{10}]_s$	80249,42	0,52
	SH	-	-	-

Fonte: Elaborada pelo autor.

Nota-se que a laminação ótima foi obtida pelo HCL com 165 pontos, apesar do erro ter sido maior que o encontrado pelo HCLO. Quando o tamanho da amostra aumentou, o HCLO apresentou melhores resultados, resultando na laminação ótima e um erro de apenas 0,52 %.

Partindo para as redes neurais com bases radiais (RRBF), o treinamento foi feito de maneira similar ao problema com 8 lâminas. Ou seja, a escolha da rede RBFi foi baseada no valor do *spread*, avaliando a regressão. A Tabela 13 apresenta os resultados da regressão para a RBFi.

Tabela 13 – Resultados do treinamento da rede RBFi.

Tamanho da amostra	Técnica	Spread	R ²
99	HCL	1,0	0,88
	HCLO	2,0	0,92
	SH	0,6	0,77
165	HCL	2,0	0,89
	HCLO	0,7	0,87
	SH	0,3	0,73
330	HCL	1,2	0,88
	HCLO	1,8	0,91
	SH	0,3	0,70

Fonte: Elaborada pelo autor.

A rede RBFi apresentou bons (i.e. $R^2 > 0,8$) resultados de regressão, com exceção da rede treinada com SH. No geral, o HCLO mostrou-se melhor que o HCL. A Tabela 14 apresenta os resultados da otimização para essa rede.

Tabela 14 – Resultados da otimização da rede RBFi.

Tamanho da amostra	Técnica	Laminação	Ncr	Diferença (%)
99	HCL	$[\pm 45_{10}]_s$	80839,89	0,22
	HCLO	$[\pm 45_{10}]_s$	80243,69	0,52
	SH	$[90_2 \pm 45_3 \ 0_2 \pm 45_2 \ 0_2 \pm 45 \ 0_2]_s$	72234,64	10,45
165	HCL	$[\pm 45_{10}]_s$	78639,42	2,51
	HCLO	$[\pm 45_{10}]_s$	82522,98	2,30
	SH	$[\pm 45_5 \ 0_4 \ 90_2 \ \pm 45 \ 90_2]_s$	64637,18	19,87
330	HCL	$[\pm 45_{10}]_s$	83832,03	3,93
	HCLO	$[\pm 45_4 \ 0_4 \ 90_2 \ \pm 45 \ 0_2 \ 90_2]_s$	81181,42	0,64
	SH	$[\pm 45 \ 90_2 \ \pm 45 \ 0_2 \ \pm 45_2 \ 90_4 \ 0_4]_s$	68392,79	15,21

Fonte: Elaborada pelo autor.

A associação feita com a rede RBFi e com o HCL apresentou os melhores resultados, tanto para a laminação quanto para a função objetivo. Mais uma vez, as redes treinadas com o SH apresentaram comportamento pior que as demais. Note que, diferente da RNA, a RBFi não precisou de tantos pontos para fornecer resultados aceitáveis, com 99 a rede já conseguiu reproduzir o comportamento da função analítica.

A outra rede treinada, foi a rede regularizada, a característica dessa rede está no fato de que ela incorpora neurônios a medida que o treinamento avança, dessa forma, a rede pode atingir resultados aceitáveis com uma quantidade de neurônios inferior ao número de neurônios da rede RBFi. As quantidades de neurônios, bem como o spread e a regressão, foram organizadas na Tabela 15.

Tabela 15 – Resultados do treinamento da rede RBFr.

Tamanho da amostra	Técnica	Spread	R²	Neurônios
99	HCL	0,4	0,83	26
	HCLO	1,1	0,57	14
	SH	1,0	0,74	20
165	HCL	1,9	0,90	25
	HCLO	1,2	0,80	26
	SH	1,8	0,77	55
330	HCL	1,2	0,88	22
	HCLO	1,9	0,85	28
	SH	1,4	0,70	85

Fonte: Elaborada pelo autor.

Observe que o número de neurônios aumentou significativamente para a técnica SH, e mesmo assim o valor da regressão caiu. Um ponto fora da reta foi o $R^2 = 0,57$, resultado da combinação da rede com HCLO, gerando uma rede com apenas 14 neurônios. Verificando os resultados da otimização, apresentados na Tabela 16, nota-se que mesmo com uma regressão baixa, a rede RBFr combinada com o HCLO conseguiu aproximar o comportamento do laminado.

Tabela 16 – Resultados da otimização da rede RBFr.

Tamanho da amostra	Técnica	Laminação	Ncr	Diferença (%)
99	HCL	$[\pm 45_{10}]_s$	82850,05	2,71
	HCLO	$[\pm 45_{10}]_s$	78706,95	2,43
	SH	$[\pm 45_2 0_2 \pm 45 0_2 \pm 45_4 90_2]_s$	71458,23	11,41
165	HCL	$[\pm 45_{10}]_s$	77130,90	4,38
	HCLO	$[\pm 45_{10}]_s$	78709,64	2,43
	SH	$[90_4 \pm 45 90_6 0_2 90_6]_s$	91019,29	12,84
330	HCL	$[\pm 45_{10}]_s$	77239,59	4,25
	HCLO	$[\pm 45_{10}]_s$	77419,16	4,02
	SH	$[0_4 \pm 45_2 0_2 90_2 \pm 45 0_4 90_2]_s$	86712,43	7,50

Fonte: Elaborada pelo autor.

A rede treinada com HCL e HCLO apresentaram resultados mais consistentes em relação à SH, cujo menor erro foi 7,5%. A combinação da rede RBFr com o HCLO forneceu resultados melhores do que com HCL.

5.1.3 *Discussão*

Como foi exposto, a escolha desse problema aconteceu pelo fato do mesmo possuir uma solução analítica. Dessa forma, seria possível avaliar alguns parâmetros importantes na construção de um modelo substituto. O primeiro parâmetro foi o tamanho da amostra, o segundo a técnica de amostragem e o terceiro o tipo de modelo substituto.

O tamanho da amostra teve uma influência maior na RNA, reduzindo o erro percentual e aumentando o número de neurônios. Para a RBFr o número de neurônios manteve-se entre 14 e 30 neurônios, com exceção da combinação com SH para 165 e 330 pontos amostrais, que resultou em números mais elevados. Além disso, o erro percentual aumentou com o número de pontos amostrais. Já na RBFi o erro percentual do HCL sofreu um aumento, porém os resultados continuaram consistentes.

Para um espaço de projeto pequeno, todas as técnicas de amostragem se comportaram de maneira se similar, aproximando a laminação ótima e gerando erros percentuais baixos. Quando o espaço de projeto aumentou, todas as redes combinadas com a SH apresentaram respostas ruins, tanto na laminação quanto na função objetivo. Amouzgar e Strömberg (2016) que trabalhou com a Sequência de Hammersley para aproximar funções matemáticas fechadas, mostrou que essa sequência gerou resultados melhores que o HCL. Porém, para problemas de laminados com grande dimensão, a SH não apresentou um bom comportamento.

O HCLO apresentou resultados melhores que o HCL, porém a técnica de geração da amostra é mais demorada, pois se trata de uma otimização e por esse motivo precisa avaliar todos os pontos considerando a quantidade de geração e de indivíduos, tornando-se mais lenta que o HCL. Desta forma, como as redes combinadas com HCL apresentaram bons resultados, optou-se por usar esta técnica no exemplo seguinte.

Quanto ao modelo substituto, as três técnicas conseguiram aproximar bem a função objetivo, porém a RNA não apresentou resultados satisfatórios para a laminação, diferentemente das redes RBFs. Além disso, o treinamento de uma RNA é um processo de tentativa e erro, sendo necessário, em muitos casos, uma repetição dos treinamentos, a alteração da topologia da rede, ou alteração de algum parâmetro específico. Quanto às redes RBFs, a construção e treinamento são mais simples, pois o processo é mais automático e fácil de implementar. É importante ressaltar que o treinamento foi mais rápido para a RBFi, mesmo usando grandes quantidades de neurônios, do que para a RBFr.

5.2 Minimização do deslocamento de uma casca cilíndrica

Esse exemplo foi trabalhado por Barroso (2015) como um problema de minimização dos deslocamentos com a restrição do critério de falha e do número máximo de camadas adjacentes na mesma direção. Aqui, iremos considerar um problema de minimização dos deslocamentos considerando apenas a restrição que limita o número máximo de camadas adjacentes, uma vez que a restrição de resistência não estava ativa em Barroso (2015). O material considerado foi o carbono-epóxi, cujas propriedades do material estão descritas na

Tabela 17.

Tabela 17 – Propriedades elásticas do material.

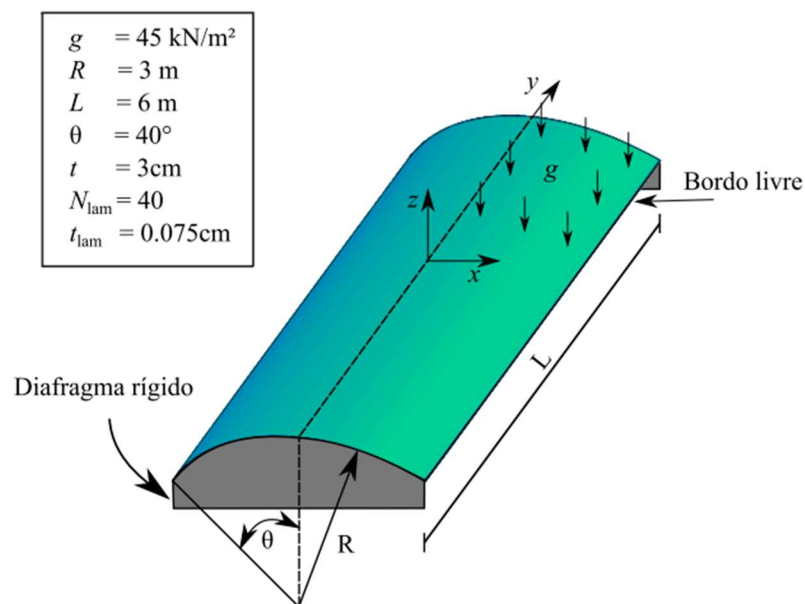
E_1 (GPa)	E_2 (GPa)	E_3 (GPa)	ν_{12}	ν_{13}	ν_{23}	G_{12} (GPa)	G_{13} (GPa)	G_{23} (GPa)
147	10,3	10,3	0,27	0,27	0,54	7	7	3,7

Fonte: Elaborada pelo autor.

A geometria e o tipo de carregamento são apresentados na

Figura 48. O objetivo aqui, é minimizar o deslocamento de uma estrutura com espessura de 30mm e laminação balanceada e simétrica com 40 camadas.

Figura 48 – Geometria e carregamento da casca.



Fonte: Barroso (2015).

Uma característica desse exemplo, é que ele não possui solução analítica, como no Exemplo 1, sendo necessário uma análise numérica para obter os pontos amostrais, além disso esse problema apresenta várias laminações ótimas com o mesmo valor da função objetivo, isto é, vários mínimos globais.

Nesse exemplo, as respostas nos pontos de amostragem foram avaliadas utilizando o *FAST*, software de análise numérica desenvolvido no LMCV. Os deslocamentos foram obtidos por meio da Análise Isogeométrica (AIG), implementada por Barroso (2015). Esta técnica foi escolhida porque é capaz de representar exatamente a geometria da estrutura.

O problema de otimização é definido como:

$$\begin{aligned}
 & \text{Deter min ar } x = [\theta_1, \theta_2, \dots, \theta_{N_v}] \text{ que} \\
 & \text{Minimize } w_b \\
 & \text{Sujeito a} \\
 & \frac{N_{cp}}{cp_{max}} - 1 \leq 0.
 \end{aligned} \tag{82}$$

Onde cp_{max} é o número de camadas adjacentes permitidas, aqui $cp_{max}=4$.

A otimização foi feita com o AG e o PSO (topologia quadrada), considerando 5 otimizações com 50 gerações e 40 indivíduos na população e laminação simétrica-balanceada. Para o treino do modelo substituto, optou-se por usar uma amostra intermediária (165 pontos), gerada pelo HCL. Vale a ressalva de que neste trabalho as amostras são contínuas, desses pontos 90% foram usados no treinamento da rede e 10% na validação. Neste caso, não foi utilizada uma amostra separada para validação, devido ao alto custo das análises estruturais. Foram utilizadas e comparadas as mesmas técnicas para o modelo substituto (RNA, RBFi e RBF_e). O parâmetro de qualidade da rede foi a regressão dos pontos de validação (R^2).

A primeira rede treinada foi uma RNA. Para tal foi utilizado a função *fitnet()*, com 10, 15 e 20 neurônios na camada escondida. A rede foi treinada para 1000 épocas, o número máximo de verificações de validação foram 6. O algoritmo de treinamento foi o *'trainlm'*. Optou-se por esse algoritmo, pois ao treinar a rede, nas mesmas condições, com outros algoritmos, como o *'trainbr'*, o tempo de treinamento aumentava e a convergência tornava-se mais lenta, não fornecendo uma regressão aceitável para a amostra em questão. Por fim, para avaliar a performance foi considerado o MSE, apesar de ser a regressão o fator dominante na escolha da rede. Os resultados para essas redes estão organizados na Tabela 18.

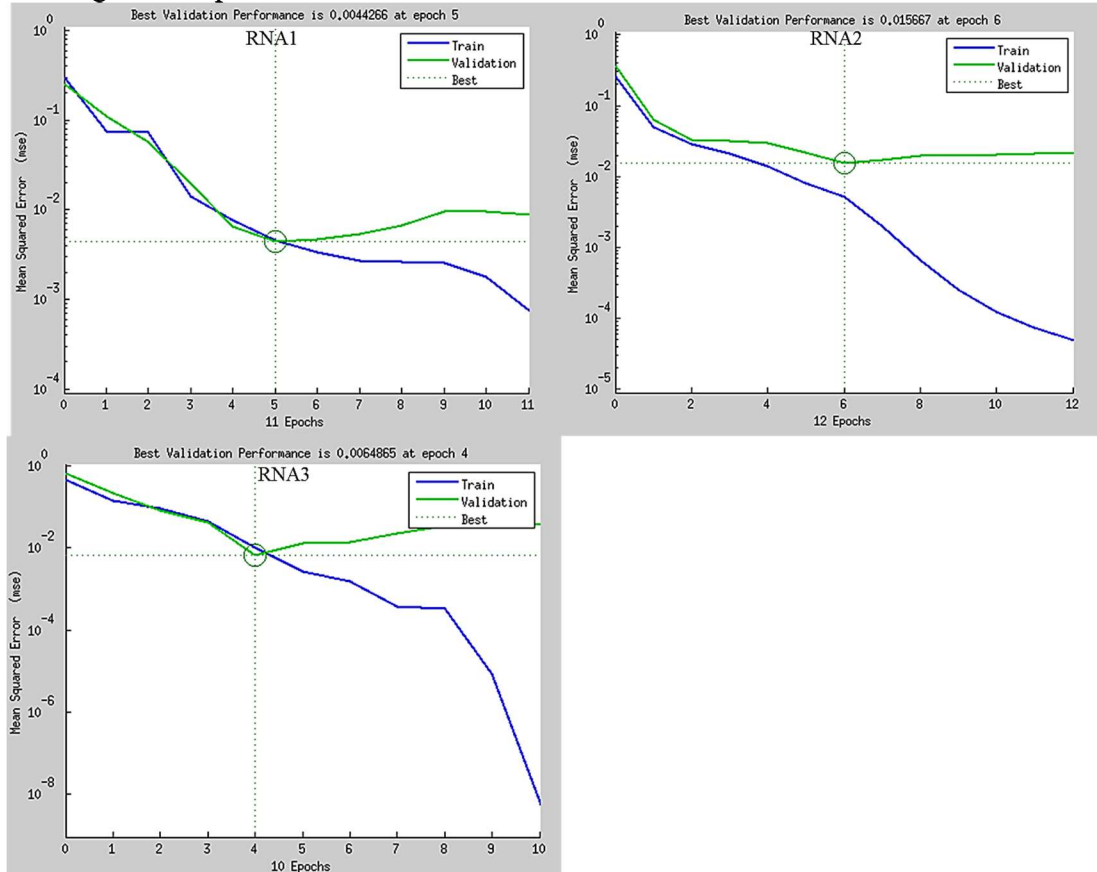
Tabela 18 – Resultado de treinamento da RNA.

Rede	Neurônios	MSE	R ² treino	R ² val	VC	Épocas
1	10	0,004	0,93	0,92	6	13/1000
2	15	0,016	0,92	0,70	6	12/1000
3	20	0,006	0,85	0,85	6	10/1000

Fonte: Elaborada pelo autor.

Os gráficos do comportamento do MSE são apresentados na Figura 49.

Figura 49 - performance das redes.



Fonte: Elaborada pelo autor.

A RNA2 foi a única que apresentou uma regressão abaixo do ideal, mesmo assim, foi considerada para otimização, pois na construção da rede esse foi o melhor resultado obtido.

A Tabela 19 apresenta os resultados para a otimização da estrutura utilizando as três RNAs. O resultado obtido por Barroso (2015), foi $[(90_4 \pm 45)_2 \ 0_4 \pm 45 \ 0_2]_s$ com $w_b = 11,58$ cm.

Tabela 19 – Otimização das RNAs.

Tipo	Laminação	w_b (cm)	Diferença (%)	Tempo (s)
RNA1 – AG	[90 ₂ 0 ₂ ±45 90 ₂ 0 ₂ ±45 ₃ 0 ₂ ±45] _s	11,0243	4,799	337
RNA1 - PSO	[90 ₂ 0 ₂ ±45 90 ₂ 0 ₂ ±45 ₃ 0 ₂ ±45] _s	11,0243	4,799	395
RNA2 – AG	[0 ₂ 90 ₄ 0 ₄ ±45 90 ₄ 0 ₂ 90 ₂] _s	10,9679	5,286	346
RNA2 - PSO	[0 ₂ 90 ₄ 0 ₄ ±45 90 ₄ 0 ₂ 90 ₂] _s	10,9679	5,286	390
RNA3 – AG	[0 ₂ 90 ₄ ±45 90 ₄ 0 ₂ ±45 0 ₂ 90 ₂] _s	11,3119	2,315	341
RNA3 - PSO	[0 ₂ 90 ₄ ±45 90 ₄ 0 ₂ ±45 0 ₂ 90 ₂] _s	11,3119	2,315	386

Fonte: Elaborada pelo autor.

Observe que as redes RNA1 e RNA2 conseguiram resultados aceitáveis para a função objetivo, mesmo a laminação não sendo aproximada. Outro ponto importante é o tempo de otimização. As redes otimizadas com AG, foram relativamente mais rápidas que aquelas otimizadas com PSO. Essa diferença no tempo ocorre, pois, o PSO avalia todos os indivíduos enquanto o AG avalia um número menor (90%) devido à taxa de cruzamento.

Substituindo a laminação encontrada pelas redes e substituindo no FAST, é possível avaliar o deslocamento “real” da laminação e desse modo comparar com o valor da rede. Em todos os casos os deslocamentos das redes foram menores que os deslocamentos do FAST, como pode ser visto na Tabela 20.

Tabela 20 – Comparativo entre resultado do FAST e da rede para a laminação aproximada.

Rede	Laminação	w_{FAST}	w_b	Diferença (%)
RNA1	[90 ₂ 0 ₂ ±45 90 ₂ 0 ₂ ±45 ₃ 0 ₂ ±45] _s	13,6959	11,0243	19,51
RNA2	[0 ₂ 90 ₄ 0 ₄ ±45 90 ₄ 0 ₂ 90 ₂] _s	15,0815	10,9679	27,28
RNA3	[0 ₂ 90 ₄ ±45 90 ₄ 0 ₂ ±45 0 ₂ 90 ₂] _s	13,8299	11,3119	18,21

Fonte: Elaborada pelo autor.

A técnica seguinte foi a rede RBFi, o mesmo processo de variação do *SPREAD* foi adotado aqui, porém considerando a regressão dos dados de validação como parâmetro. Com isso obteve-se uma rede com *SPREAD* = 1,7 e $R^2 = 0,81$. Os resultados estão organizados na Tabela 21.

Tabela 21 – Resultados da otimização da rede RBFi.

Tipo	Laminação	w_b (cm)	Diferença (%)	Tempo (s)
RBFi - AG	[0 ₂ 90 ₂ ±45 90 ₄ ±45 90 ₄ ±45 90 ₂] _s	11,2865	2,535	793
RBFi - PSO	[0 ₂ 90 ₂ ±45 90 ₄ ±45 90 ₄ ±45 90 ₂] _s	11,2865		940

Fonte: Elaborada pelo autor.

Ao substituir a laminação da rede no FAST, obteve-se $w_b = 15,6856$ cm, uma

diferença de 28% entre os deslocamentos da rede e do FAST.

Para a rede RBFr foi treinada com uma variação de SPREAD diferente (3 a 4), isso porque a regressão para *SPREADs* pequenos era muito baixa. Dessa forma, o SPREAD para a rede foi 3,5 e $R^2 = 0,84$. Nesse caso, 29 neurônios foram utilizados na camada escondida. A Tabela 22 apresenta os resultados da otimização para a rede RBFr.

Tabela 22 – Resultados da otimização para rede RBFr.

Tipo	Laminação	w_b (cm)	Diferença (%)	Tempo (s)
RBFr - AG	[90 ₂ 0 ₄ 90 ₂ 0 ₄ ±45 90 ₂ 0 ₂ ±45]s	13,9253	20,253	1941
RBFr- PSO	[90 ₂ 0 ₄ 90 ₂ 0 ₄ ±45 90 ₂ 0 ₂ ±45]s			2205

Fonte: Elaborado pelo autor.

Utilizando o FAST para analisar a laminação encontrada pela RBFr, obtêm-se um deslocamento de 14,392 cm, o que corresponde a um erro de 24,3%.

5.2.1 Comentários

Esse é um problema que permite várias laminacões ótimas. Sendo assim, com exceção da RBFr, as redes apresentaram resultados próximos aos exatos para o deslocamento. Porém, quando se compara o deslocamento do modelo substituto com o deslocamento obtido pelo FAST, utilizando a mesma laminação encontrada pelas redes, percebe-se um erro percentual alto, chegando a 27% para a RNA. Provavelmente, isto se deve ao pequeno tamanho da amostra utilizada.

Em todos os casos, o tempo de otimização do PSO foi maior que o tempo do AG, pois como explicado o AG não avalia toda a população. Ambos os algoritmos levaram à mesma solução, mostrando a consistência dos algoritmos utilizados para otimização. Por outro lado, às soluções variaram de um modelo substituto para outro.

Contudo, a utilização de modelos substitutos levou a uma significativa redução no tempo de processamento em relação à otimização utilizando diretamente à AIG, que levou 6133s. Isso mostra o potencial dos modelos substitutos para problemas de otimização de estruturas laminadas.

6 CONCLUSÃO

Esse trabalho abordou o uso de modelos substitutos no processo de otimização de estruturas laminadas. Esses modelos foram baseados nas técnicas de Redes Neurais Artificiais e Redes de Funções de Bases Radiais.

Para a construção e treinamento dos modelos substitutos, foi utilizado o software comercial MATLAB, pois este apresenta um toolbox voltado para esse tipo de aplicação. Vale ressaltar que, para o problema da placa laminada, foram usadas três técnicas de amostragem diferente, a hipercubo latino padrão, a sequência de Hammersley e a uma versão melhorada do hipercubo latino. O número de pontos na amostra sofreu variação, conforme o número de variáveis aumentava.

No primeiro exemplo, foi possível notar que, para problemas de pequena dimensão, a técnica de geração da amostra não é tão significativa, pois todas as redes apresentaram boas aproximações, tanto para a laminação quanto para o ótimo. Para o problema com maior espaço de projeto, a sequência de Hammersley não apresentou bons resultados.

O segundo exemplo tratou da minimização dos deslocamentos em uma casca cilíndrica, tendo sido necessário utilizar um software de análise numérica como modelo de alta fidelidade para obter a resposta da estrutura nos pontos amostrais. Os deslocamentos foram obtidos por meio da análise isogeométrica, através do software acadêmico FAST. Uma particularidade desse problema é que ele admite várias laminações ótimas. Os resultados obtidos mostraram que as redes RNA e RBFi conseguiram aproximar muito bem a função objetivo, já a RBFr apresentou um erro maior que as demais. Outra avaliação que pode ser feita, foi em relação ao tempo, todas as redes conseguiram uma função objetivo ótima em um intervalo de tempo bem menor que o modelo de alta fidelidade.

Das técnicas estudadas, a que apresentou mais vantagens foi a RBFi, pois é mais fácil de se trabalhar, o treinamento é direto (i.e. com um único treinamento é possível obter resultados favoráveis), diferente da RNA que trabalha com tentativa e erro. Quanto ao número de pontos amostrais, a relação que Amouzgar e Strömberg (2016) apresenta foi suficiente para os laminados estudados. Em relação a técnica de amostragem, a melhor foi o HCLO, seguido do HCL. Como foi dito, a SH não apresentou bons resultados para os laminados.

Dessa forma pode-se afirmar que o uso de modelos substitutos nos processos de otimização é aceitável e merece ser estudado cada vez mais. Note que aqui, foram estudadas apenas três técnicas que podem ser utilizados nos modelos substitutos, com técnicas de

treinamento específica do MATLAB. Com base nisso, algumas sugestões para trabalhos futuros são:

- a) Utilizar como treinamento das RBFs a técnica de validação cruzada, de modo a determinar o parâmetro de espalhamento;
- b) Estudar a aplicação dos modelos substitutos em problemas de otimização sequencial aproximada;
- c) Utilizar restrições de critério de falha no processo de otimização;
- d) Estudar o comportamento de outras técnicas de modelos substitutos, como Máquina de Suporte Vetorial e Kriging.

REFERÊNCIAS

- AMOUZGAR, K.; STRÖMBERG, N. Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias. **Structural and Multidisciplinary Optimization**, p. 1–17, 2016.
- ARORA, J. S. **Introduction to Optimum Design**. 2.ed. Elsevier, 2004.
- AWRUCH, M. D. F.; HERNANDES, J. A.; BUSSAMRA, F. L. S. **Utilização de metamodelos de painéis reforçados integrais otimizados sujeitos à flambagem**. VI Congresso Nacional de Engenharia Mecânica, Campina Grande – PB, 2010.
- BARBOSA, A. H.; FREITAS, M. S. R.; NEVES, F. A. Confiabilidade estrutural utilizando o método de Monte Carlo e redes neurais. **R. Esc, Minas**. v. 58, p. 247-255, 2005.
- BARROSO, E. S. **Análise e otimização de estruturas laminadas utilizando a formulação isogeométrica**. 2015. 156 f. Dissertação (Mestrado em Engenharia Civil: Estruturas e Construção Civil) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2015.
- BARTHELEMY, J. F. M.; HAFTKA, R. T. **Structural Optimization structural design**. **Structural Optimization**, v. 144, n. 6, p. 129–144, 1993.
- BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. **Swarm Intelligence Symposium, 2007. SIS 2007. IEEE**, n. Sis, p. 120–127, 2007.
- BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T.B. **Redes Neurais Artificiais: Teoria e Aplicações**. 2. ed. Rio de Janeiro: LTC, 2007.
- CAVAZZUTI, M. **Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics**. 1.ed. Springer-Verlag Berlin Heidelberg, 2013.
- COLEMAN, D. E.; MONTGOMERY, D. C. A Systematic Approach to Planning for a Designed Industrial Experiment. **Technometrics**. v.35, No.1. p. 1-12, 1993.
- COPPIN, B. **Inteligência Artificial**. Tradução de Jorge Duarte Pires. Rio de Janeiro: LTC, 2013.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. John Wiley & Sons, INC, 2000.
- DEB, K. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, v. 186, n. 2–4, p. 311–338, 2000.
- FORRESTER, A. I. J.; KEANE, A. J. Recent advances in surrogate-based optimization. **Progress in Aerospace Sciences**, v. 45, n. 1–3, p. 50–79, 2009.
- FORRESTER, A. I. J.; SÓBETER, A.; KEANE, A. J. **Engineering Design via Surrogate Modelling**. John Wiley & Sons Ltd. 2008.

FRISSO, C.; SCARPEL, R. A.; FERRARI, D. B. T. P. A. Metamodelagem de funções determinísticas por composição integrada de especialistas locais. **XLIII Simpósio Brasileiro de Pesquisa Operacional**, Ubatuba-SP, 2011.

GASPAR, B.; TEXEIRA, A. P.; SOARES, C. G. Assessment of the efficiency of Kriging surrogate models for structural reliability analysis. **Probabilistic Engineering Mechanics**. v.37, p. 24-34, 2014.

GRECO-SANTANNA, J. A.; FABBRI, G. T. Considerações sobre planejamento e análise de experimentos fatoriais fracionários assimétricos – exemplos de aplicação a um caso de infraestrutura de transportes. **IV Encontro Tecnológico da Engenharia Civil e Arquitetura-ENTECA**, 2003.

GHUGAL, Y. M. M.; SHIMPI, R. P. P. A Review of Refined Shear Deformation Theories for Isotropic and Anisotropic Laminated Beams. **Journal of Reinforced Plastics and Composites**, v. 20, n. 9, p. 255–272, 2002.

GÜRDAL, Z.; HAFTKA, R. T.; HAJELA, P. Design and optimization of laminated composite materials. [s.l: s.n.].

HAFTKA, R. T.; GÜRDAL, Z. **Elements of Structural Optimization**. 3. ed. kluwer Academic Publishers, 1991.

HAMMERSLEY, J. M. Monte Carlo Methods For Solving Multivariable Problems. **Annals of the New York Academy of Sciences**, v. 86, n. 3, p. 844–874, 15 dez. 2006.

HAN, Z. H. Surrogate - Based Aerodynamic Optimization via Gradient - Enhanced Kriging and Hierarchical Kriging, 2013.

HAN, Z. -H.; ZHANG, K-S. Surrogate-Based Optimization, **Real-World Applications of Genetic Algorithms**. Dr. Olympia Roeva (Ed.), InTech, DOI: 10.5772/36125, 2012.

HAYKIN, S. **Redes Neurais: Princípio e prática**. Tradução de Paulo Martins Engel. 2. ed. Porto Alegre: Bookman, 2001.

HERATH, M. T.; et al. Smoothed finite elemento and genetic algorithm based optimization for shape adaptative composite marine propellers. **Composite Structures**. v. 109, p. 189-197, 2014.

JAYAWARDENA, A. W.; FERNANDO, A. K. Use of radial basis function type artificial neural networks for runoff simulation. **Computer-Aided Civil and Infrastructure Engineering**. v. 13, p.91-99, 1998.

JONES, R. M. Mechanics of Composite materials. 2nd. ed. Taylor & Francis, 1999.
KALAGNANAM, J.; DIWEKAR, U. An efficient sampling technique for off-line quality control. **Technometrics**, v. 39, n. 3, p. 308–319, 1997.

KAPANIA, R. K.; Liu, Y. Applications of Artificial Neural Networks in Structural Engineering with Emphasis on Continuum Models , Relatório técnico, Virginia Polytechnic

Institute and State University, Blacksburg, VA, 1998.

KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. **Neural Networks**, v. 4, p. 1942, 1995.

KENNEDY, J.; MENDES, R. Population Structure and Particle Swarm Performance. In: c2002. v. 2. p. 1671 – 1676.

KITAYAMA, S.; ARAKAWA, M.; YAMAZAKI, K. Sequential Approximate Optimization using Radial Basis Function network for engineering optimization. **Optimization and Engineering**, v. 12, n. 4, p. 535–557, 2011.

KOIDE, R. M.; FERREIRA, A. P. C. S.; LUERSEN, M. A. Laminated Composites Buckling Analysis Using Lamination Parameters, Neural Networks and Support Vector Regression. **Latin American Journal of Solids and Structures**, v. 12, n. 1999, p. 271–294, 2014.

KONAK, A.; COIT, D. W.; SMITH, A. E. **Multi-objective optimization using genetic algorithms: A tutorial**. Reliability Engineering & System Safety. v.91, p. 992-1007, 2006.

LASTIRI, C. P.; PAULETTI, R. M. O. Aplicação de Redes Neurais Artificiais à Engenharia de Estruturas. **XXXI Jornadas Sud-Americanas de Ingeniería Estructural**. Mendoza-Argentina. 2004.

LEARY, S. J.; BHASKAR, A.; KEANE, A. J. A derivative Based Surrogate Model for Approximating and Optimizing the Output of an Expensive Computer Simulation. **Journal of Global Optimization**. v.30. p. 39-58, 2004.

LEMONGE, A. C. C.; BARBOSA, H. J. C. An adaptive penalty scheme for genetic algorithms in structural optimization. **International Journal for Numerical Methods in Engineering**, v. 59, n. 5, p. 703–736, 2004.

LIU, Y. Y.; WANG, S. A scalable parallel genetic algorithm for the Generalized Assignment Problem. **Parallel Computing**. v.46, p.98-119, 2015.

LOPES, D. B. P.; et al. Uso de rede neural artificial aplicada à classificação automática de descontinuidades em chapas de compósito laminado de alumínio e epóxi reforçado por fibra de vidro. **VI Congresso Nacional de Engenharia Mecânica – CONEM**. Campina Grande-PB. 2010.

MASON, R. L.; GUNST, R. F.; HESS, J. L. **Statistical Design and Analysis of Experiments: with applications to engineering and Science**. 2. ed. John Wiley & Sons, INC, 2003.

MULGREW, B. Applying radial basis functions. **Signal Processing Magazine, IEEE**, v. 13, n. 2, p. 50–65, 1996.

NICHOLAS, P. E.; BABU, M. C. L.; SOFIA, A. S. Mechanics of Materials and Structures. **Journal of Mechanics of Materials and Structures**, v. 11, n. March, p. 157–172, 2016.

ORR, M. J. L. **Introduction to radial basis function networks**. Time, p. 1–67, 1996.

PAL, M.; DESWAL, S. Support vector regression based shear strength modelling of deep beams. **Computers and Structures**. v.89, p. 1430-1439, 2011.

PAN, F.; ZHU, P.; ZHANG, Y. Metamodel-based lightweight design of B-pillar with TB structure via support vector regression. **Computers and Structures**. v.88, p. 36-44, 2010.

PINA, A. A. **Metodologias de análise, síntese e otimização de sistemas para produção de petróleo offshore através de metamodelos e ensaio de partículas**. 2009. 176p. Tese (Doutorado em ciências em Engenharia Civil) – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

REID, D. J. Genetic Algorithms in Constrained Optimization. **Mathl. Comput. Modelling**. v.23, N° 5, p. 87-111, 1996.

REDDY, J. N. **Mechanics of laminated composite plates and shells: Theory and Analysis**. 2. ed. CRC Press, 2004.

ROCHA, I. B. C. M. **Análise e otimização de cascas laminadas considerando não-linearidade geométrica e falha progressiva**. Dissertação (Mestrado em Engenharia Civil). Universidade Federal do Ceará, 2013.

ROCHA, I. B. C. M.; PARENTE, E.; MELO, A. M. C. A hybrid shared/distributed memory parallel genetic algorithm for optimization of laminate composites. **Composite Structures**, v. 107, n. 1, p. 288–297, 2014.

SILVA, R. F. **Otimização de Risers de materiais compósitos**. 2012. 96f. Dissertação (Mestrado em Engenharia Civil), Universidade Federal do Ceará, Fortaleza, 2012.

SIMPSON, T. W. et al. Metamodels for Computer-based Engineering Design: Survey and recommendations. **Engineering With Computers**, v. 17, n. 2, p. 129–150, 2001.

SLISERIS, J.; ROCENS, K. Optimal design of composite plates with discrete variable stiffnesses. **Composite Structures**. v.98, p.15-23, 2013

SOUZA, S. A.; OLIVEIRA, L. C.; AFONSO, S. M. B. Otimização de estratégia de produção de petróleo utilizando algoritmo evolucionário e modelos substitutos. **VI Congresso Nacional de Engenharia Mecânica**, Campina Grande - PB, 2010.

The Mathworks, Inc. **Neural Network Toolbox™**. Disponível em: <<http://www.mathworks.com/products/neural-network/>> Acesso: 28/08/2017.

TIAGO, C. M.; LEITÃO, V. M. A. Utilização de Funções de base radial em problemas unidimensionais de análise estrutural. **Métodos numéricos en ingeniería V**, España, 2002.

TSAI, S. W.; WU, E. M. A General Theory of Strength for Anisotropic Materials. **Journal of Composite Materials**, v. 5, n. 1, p. 58–80, 1971.

VIANA, F. A. C.; VENTER, G.; BALABANOV, V. An algorithm for fast optimal latin hypercube design of experiments. **International Journal for Numerical Methods in Engineering**, v. 82, n. 2, p. 135–156, 2010.

VIANA, G. V. R. **Meta-heurística e programação paralela em otimização combinatória**. Fortaleza: EUFC, 1998.

WANG, G. G.; SHAN, S. Review of Metamodeling Techniques in Support of Engineering Design Optimization. **Journal of Mechanical Design**, v. 129, n. 4, p. 370, 2007.

YIN, H. et al. Design optimization of a MASH TL-3 concrete barrier using RBF-based metamodels and nonlinear finite element simulations. 2016.