

**UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA MECÂNICA**

**DISSERTAÇÃO DE MESTRADO**

**Desenvolvimento e Aplicação do Algoritmo  
de Espectros de Lichtenberg em Otimização  
de Sistemas Mecânicos**

**Autor: João Luiz Junho Pereira**

**Orientador: Prof. Dr. Guilherme Ferreira Gomes**

**Coorientador: Prof. Dr. Sebastião Simões da Cunha Jr.**

Janeiro de 2020  
Itajubá - MG

**UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA MECÂNICA**

**João Luiz Junho Pereira**

# **Desenvolvimento e Aplicação do Algoritmo de Espectros de Lichtenberg em Otimização de Sistemas Mecânicos**

**Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica como parte dos requisitos para obtenção do título de Mestre em Ciências em Engenharia Mecânica.**

**Área de Concentração: Projeto, Materiais e Processos**

**Orientador: Prof. Dr. Guilherme Ferreira Gomes**

**Coorientador: Prof. Dr. Sebastião Simões da Cunha Jr.**

Janeiro de 2020  
Itajubá – MG

## **Dedicatória**

*Dedico este trabalho a meu pai, João Batista de Melo Pereira; À minha tia, Maria Mônica e ao meu irmão Carlos cujos apoios transcenderam a esfera física.*

*Aos Estudantes que cumulativamente desenvolvem a Ciência e a Tecnologia em prol do Desenvolvimento Humano!*

*"Daria tudo o que sei pela metade do que ignoro"*

René Descartes

## Agradecimentos

Neste período repleto de dificuldades, desafios e superações que repercutiram no andamento deste trabalho, quero agradecer:

Ao professor Dr. Guilherme Ferreira Gomes pela orientação, paciência e apoio técnico essencial para o desenvolvimento desta Dissertação.

Ao professor Dr. Sebastião Simões pelas coorientação técnica, pessoal, amizade, paciência, conselhos de vida e motivação que trarão boas recordações para toda a vida.

Aos amigos Matheus Ramos (e família, em especial à Márcia) e Leonardo Huhn pelo suporte vital durante este período. Também aos amigos do GEMEC, em especial à Camila e aos demais professores do Instituto de Engenharia Mecânica.

Aos meus familiares pelo carinho e suporte, principalmente meu pai e irmão, que sempre me apoiaram em minhas decisões e me ouviram. À minha tia Mônica, que sempre enfatizou a importância do estudo e aos primos-irmãos Carol e Bruno pelo carinho e inspiração. À minha tia Lúcia e marido que colaboraram para as dificuldades deste período.

Aos profissionais do Hospital Escola de Itajubá, equipe de Ortopedia, enfermeiros e técnicos de enfermagem pelo cuidado e atenção que se mostrou além de profissional. Em especial à Dra. Elisa, ao Dr. Rodrigo, Dr. Edvaldo, Dr. Luciano e Leandro.

Por fim, agradeço a todos que torcem pelo meu sucesso.

## Resumo

**PEREIRA, J. L. J. (2019)**, *Desenvolvimento e Aplicação do Algoritmo de Espectros de Lichtenberg em Otimização de Sistemas Mecânicos*, Itajubá, 120p. Dissertação (Mestrado em Engenharia Mecânica) – Instituto de Engenharia Mecânica, Universidade Federal de Itajubá.

A Otimização é uma ferramenta imprescindível para minimizar ou maximizar funções, obtendo resultados ótimos sobre custos, massa, energia, ganhos, entre outros. Problemas reais podem ser multimodais, não lineares e descontínuos e podem não ser minimizados por métodos analíticos clássicos que dependem de gradiente. Neste contexto, existem algoritmos metaheurísticos inspirados em fenômenos da natureza para otimização de problemas reais. Não existe algoritmo melhor ou pior, mas mais eficiente para determinado tipo de problema. Assim, um inédito algoritmo metaheurístico foi criado neste trabalho inspirado no fenômeno físico de relâmpagos intra-nuvem radiais e Figuras de Lichtenberg, explorando com sucesso o poder dos fractais e é diferente de muitos da literatura por ser um algoritmo híbrido composto por métodos de busca baseados em população e trajetória. O Algoritmo de Lichtenberg (LA) desenvolvido foi testado por funções de teste, onde foi validado com sucesso. Foi aplicado no problema-projeto da viga soldada, onde encontrou resultados com excelentes valores de função objetivo ao comparar aos da literatura. Finalmente e sendo o maior fim, o LA foi acoplado ao Método de Elementos Finitos (MEF) para identificação de danos do tipo delaminação e foi comparado com outros algoritmos como o Algoritmo Genético (AG) e o *SunFlower Optimization* (SFO), tendo excelentes resultados. O LA se mostrou como ferramenta poderosa na identificação de danos, pois não errou a identificação do dano mesmo em situações muito ruidosas (10% de ruído) e com baixa severidade do dano (10% de redução de rigidez somente).

*Palavras-chave:* Algoritmo de Lichtenberg, Figuras de Lichtenberg, Otimização, Meta-Heurísticas, Identificação de Danos.

## Abstract

**PEREIRA, J. L. J. (2019)**, *Development and Application of the Lichtenberg Spectrum Algorithm in Mechanical Systems Optimization*, Itajubá, 120 f. Thesis (Master in Mechanical Engineering) - Institute of Mechanical Engineering, Federal University of Itajubá.

Optimization is an essential tool to minimize or maximize functions, obtaining optimal results on costs, mass, energy, gains, among others. Actual problems may be multimodal, nonlinear, and discontinuous and may not be minimized by classical analytical methods that depend on the gradient. In this context there are metaheuristics algorithms inspired by natural phenomena to optimize real problems. There is no better or worse algorithm, but more efficient for a given type of problem. Thus, an unprecedented metaheuristic algorithm was created in this work inspired by the physical phenomenon of radial intra-cloud lightning and Lichtenberg Figures, successfully exploiting the fractal power and it is different from many in the literature as it is a hybrid algorithm composed of methods of search based on population and trajectory. The Lichtenberg Algorithm (LA) developed was tested in test functions, where it was successfully validated it. Was applied to the design problem of the welded beam, where it found results with excellent objective function values when compared to those in the literature. Finally and being the ultimate goal, LA was coupled with the Finite Element Method (FEM) for delamination damage identification and was compared with other algorithms such as Genetic Algorithm (GA) and SunFlower Optimization (SFO), yielding excellent results. LA has proven to be a powerful damage identification tool because it has not misplaced damage even in very noisy (10% noise) and low damage severity (10% stiffness reduction only) situations.

*Keywords:* Lichtenberg Algorithm, Lichtenberg Figures, Optimization, Metaheuristics, Damage Identification.

# SUMÁRIO

<b>Lista de Figuras .....</b>	<b>xii</b>
<b>Lista de Tabelas.....</b>	<b>xivv</b>
<b>Lista de Símbolos.....</b>	<b>xvii</b>
<b>Lista de Abreviaturas.....</b>	<b>xvi</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1 Objetivos.....	4
1.2 Justificativas .....	5
1.3 Estrutura da Dissertação .....	5
<b>2. REFERENCIAL TEÓRICO.....</b>	<b>6</b>
2.1 Otimização.....	23
2.2 Identificação de Danos .....	11
2.2.1 Identificação de danos utilizando o MEF .....	28
2.2.2 Posicionamento dos sensores na identificação de danos .....	33
2.2.3 Monitoramento da integridade estrutural.....	35
2.2.4 Monitoramento por meio de vibrações mecânicas .....	35
2.2.5 Delaminação .....	35
2.2.6 Método da identificação do dano.....	20
<b>3. ALGORITMO DE LICHTENBERG.....</b>	<b>23</b>
3.1 Relâmpagos .....	23
3.2 Figuras de Lichtenberg .....	25



3.3 Métodos para Construir uma Figura de Lichtenberg.....	28
3.3.1 Difusão limitada por agregação .....	28
3.3.2 Modelo da quebra do dielétrico .....	33
3.3.3 Busca na literatura por otimizadores baseados em FL .....	35
3.4 Algoritmo de Lichtenberg .....	36
<b>4. METODOLOGIA - APLICAÇÕES.....</b>	<b>46</b>
4.1 Funções de Teste com Restrições Lineares.....	46
4.2 Problema Teste Com Restrições Não Lineares.....	48
4.2.1 Projeto da viga soldada .....	48
4.3 Metodologia na Identificação de Danos .....	50
4.3.1 Modelagem do problema direto .....	50
4.3.2 Modelagem do problema inverso .....	52
4.3.3 Modelagem de ruídos.....	53
4.3.4 posição dos sensores .....	54
<b>5. RESULTADOS E DISCUSSÕES.....</b>	<b>55</b>
5.1 Desempenho nas Funções de Teste .....	55
5.1.1 Duas Variáveis .....	56
5.1.2 Três Variáveis .....	61
5.1.3 Demais Variáveis .....	63
5.1.4 Comparação dos Menores Valores da Função Objetivo.....	64
5.2 Desempenho no Problema Teste .....	64
5.3 Desempenho na Identificação da Delaminação.....	67
5.3.1 Sinais sem Ruídos.....	67
5.3.1 Sinais com Ruídos .....	75
<b>6. CONCLUSÃO .....</b>	<b>83</b>
6.1 Conclusões.....	83
6.2 Sugestões Futuras .....	85
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>86</b>
<b>APÊNDICE A - Programa LA .....</b>	<b>94</b>
<b>APÊNDICE B - Resultados LA sem Restrição.....</b>	<b>108</b>

<b>APÊNDICE C - Resultados LA - Projeto de Viga Soldada.....</b>	<b>113</b>
<b>APÊNDICE D - Resultados LA- Identificação de Danos .....</b>	<b>115</b>
<b>ANEXO A - Registro no INPI.....</b>	<b>119</b>

# Lista de Figuras

Figura 1.1 – Fractal formado devido à propagação de descarga elétrica por um meio resistente...	3
Figura 1.2 – Placas com modos de vibração definidos quando excitada com frequências próximas – fenômeno de ressonância.....	4
Figura 2.1 – <i>Bird Function</i> – Função Multimodal com Mínimo Global em (-1.58, -3.13).....	7
Figura 2.2 – Diagrama relacionando as metaheurísticas, as fontes de inspiração e os algoritmos criados .....	10
Figura 2.3 – Quadro Multidisciplinar da concepção de um sistema SHM. ....	13
Figura 2.4 – Delaminação em Material Compósito. ....	20
Figura 2.5 – Fluxograma resumido da identificação de danos por otimização.....	22
Figura 3.1 – Relâmpago intra-nuvem.....	24
Figura 3.2 – Exemplos de figuras oriundas de descargas elétricas em meios resistentes diferentes, sendo (a) em humanos e (b) em acrílicos .....	25
Figura 3.3 – FL criada sobre placa de resina.....	26
Figura 3.4 – FL em gás tetra fluoreto sob tensão de 30kV e pressão 30,3atm.....	27
Figura 3.5 – Iterações iniciais da formação do aglomerado segundo o método do DLA .....	29
Figura 3.6 – Aglomerado gerado pelo modelo da agregação limitada por difusão para 3600 partículas .....	29
Figura 3.7 – Função de Correlação de Densidade do aglomerado em função do seu raio geométrico de criação.....	30

Figura 3.8 – Exemplificação Numérica da Equação diferencial, sendo $i$ a direção horizontal e $j$ a vertical.....	31
Figura 3.9 – Influência do Coeficiente de Aderência ( $S$ ) na densidade do aglomerado e, portanto na Dimensão do Fractal ( $D$ ) .....	33
Figura 3.10 – Evolução do aglomerado a partir do Modelo de Quebra do Dielétrico.....	34
Figura 3.11 – Algoritmos com inspirações semelhantes às do LA: (a) LSA e (b) FSA.....	36
Figura 3.12 – Figuras de Lichtenberg em <i>bitmap</i> geradas no início da simulação do LA.....	38
Figura 3.13 – Transformação das figuras <i>bitmap</i> em coordenadas plotadas no intervalo .....	39
Figura 3.14 – Exemplificação das escalas e rotações randômicas para uma mesma figura de lichtenberg.....	40
Figura 3.15 – Figura Local com 30% do tamanho da Global e algumas Iterações.....	41
Figura 3.16 – Distribuição da população nas Figuras de Lichtenberg ( $pop = 10$ e $ref = 0.3$ ).....	42
Figura 3.17– Distribuição da população no LA para três variáveis e dois exemplos de iterações ( $pop = 10$ e $ref = 0.3$ ) .....	42
Figura 3.18 – Fluxograma do LA.....	44
Figura 4.1 – Variáveis do Projeto da Viga Soldada .....	48
Figura 4.2 – Modelo utilizado no MEF e identificação dos elementos.....	52
Figura 4.3 – Localização dos Danos a serem encontrados pelo LA .....	53
Figura 4.4 – Localização dos sensores .....	54
Figura 4.5 – Resumo da Metodologia da Identificação de Danos .....	54
Figura 5.1 – Gráficos de Convergência da função objetivo avaliando o desempenho do LA para geração ou não de FL diferentes a cada iteração.....	60
Figura 5.2 – Influência negativa do aumento do número de variáveis na precisão do LA .....	64
Figura 5.3 – Convergência da Função Objetivo para o Cenário I.....	69
Figura 5.4 – Gráficos de convergência da identificação do dano para o Cenário I. ....	69
Figura 5.5 – Distância entre o dano calculado pelo LA e o dano real para algumas iterações. ....	70
Figura 5.6 – Convergência da Função Objetivo para o Cenário II .....	70
Figura 5.7 – Gráficos de convergência da identificação do dano para o Cenário II .....	71
Figura 5.8 – Convergência da Função Objetivo para o Cenário III .....	71

Figura 5.9 – Gráficos de convergência da identificação do dano para o Cenário III.....	72
Figura 5.10 – Resultados visuais da identificação de dois danos.....	73
Figura 5.11 – Convergência da Função Objetivo para o Cenário VIII .....	74
Figura 5.12 – Gráficos de convergência da identificação do dano para o Cenário III.....	74
Figura 5.13 – Convergências das funções objetivos para o Cenário I sem e com adição de ruídos .....	76
Figura 5.14 – Convergências das localizações do dano para o Cenário I sem e com adição de ruídos.....	77
Figura 5.15 – Convergências das taxas de dano para o Cenário I sem e com adição de ruídos ...	78
Figura 5.16 – Influência negativa do acréscimo de ruídos na precisão da identificação do dano	78
Figura 5.17 – Poluição ruidosa nos seis primeiros modos de vibração.....	80
Figura A-1 – Certificado de Registro do LA ao INPI.....	119

# Lista de Tabelas

Tabela 3.1 – Resultados Figuras <i>Bitmap</i> .....	38
Tabela 3.2 – Parâmetros Recomendados do LA .....	45
Tabela 4.1 – Funções de Teste para Validação do LA .....	47
Tabela 4.2 – Propriedades do Material.....	51
Tabela 4.3 – Cenários de Identificação de danos propostos ao LA .....	52
Tabela 5.1 – Parâmetros do LA para otimização de funções bidimensionais .....	56
Tabela 5.2 – Resultados de otimização de funções de teste bidimensionais.....	56
Tabela 5.3 – Análise dos resultados de busca para funções bidimensionais.....	57
Tabela 5.4 – Parâmetros do LA para funções tridimensionais.....	62
Tabela 5.5 – Resultados das funções tridimensionais .....	62
Tabela 5.6 – Parâmetros do LA para demais variáveis .....	63
Tabela 5.7 – Resultados da otimização para demais variáveis .....	63
Tabela 5.8 – Parâmetros utilizados no LA para o Projeto de Viga Soldada .....	65
Tabela 5.9 – Variáveis ótimas de Projeto (LA).....	65
Tabela 5.10 – Comparação dos Resultados do Projeto de Viga Soldada entre diferentes algoritmos.....	66
Tabela 5.11 – Relação do LA com a média dos outros resultados.....	66

Tabela 5.12 – Comparação Estatística dos menores valores de função objetivo do Projeto de Viga Soldada .....	66
Tabela 5.13 – Parâmetros do LA para identificação de um elemento.....	68
Tabela 5.14 – Resultados da busca pelo dano localizado no elemento número 19 com diferentes taxas de severidade.....	68
Tabela 5.15 – Resultados visuais da busca do dano para todos os cenários da tabela anterior ....	68
Tabela 5.16 – Resultados da identificação do dano do Cenário I .....	72
Tabela 5.17 – Parâmetros do LA para identificação de danos em dois elementos .....	73
Tabela 5.18 – Resultados da identificação de dois danos e mesma severidade .....	73
Tabela 5.19 – Influência do Ruído na média dos resultados.....	75
Tabela 5.20 – Desempenho da identificação de dano do Cenário I com níveis diferentes de ruídos .....	81
Tabela 5.21 – Resultados pela busca do dano em condição muito ruidosa .....	81
Tabela B-1 – Simulações da Validação para uma variável com $f_1$ e $f_2$ .....	108
Tabela B-2 – Simulações da Validação para três variáveis com $f_1$ e $f_2$ .....	109
Tabela B-3 – Simulações da Validação para quatro variáveis com $f_1$ e $f_2$ .....	110
Tabela B-4 – Simulações da Validação para cinco variáveis com $f_1$ e $f_2$ .....	111
Tabela C-1 – Problema Teste - Projeto Viga Soldada .....	113
Tabela D-1 – Simulações de Identificação de Delaminação – Dano induzido no elemento 19 com severidades de 0.2, 0.5 e 0.9.....	115
Tabela D-2 – Simulações de Identificação de Delaminação – Dano induzido nos elementos 19 e 65 com severidade de 0.2. ....	116
Tabela D-3 – Simulações de Identificação de Delaminação – Dano induzido no elemento 19 com severidade de 0.2 e adição de ruídos nos sinais .....	117

# Lista de Símbolos

## Simbolos Latinos

$t$	Iteração
$f(x)$	Função Objetivo ou Função Custo
$f'(x)$	Derivada primeira da Função Objetivo (se houver)
$C(r)$	Correlação da FL
$N$	Número de Partículas DLA
$r$	Raio Geométrico do espaço de construção da FL no DLA
$r'$	Raio da FL no DLA
$u$	Probabilidade de uma partícula atingir um local
$k$	passo seguinte do algoritmo DLA
$c$	Vizinhos disponíveis do algoritmo DLA
$\vec{x}$	Local qualquer
$a$	Distância entre partículas no DLA
$D$	Dimensão do Fractal
$N_p$	Número de Partículas
$R_c$	Raio de Criação
$S$	Coefficiente ou Probabilidade de Aderência
$R_{cluster}$	Raio do aglomerado
$N_{cluster}$	Número de Partículas do aglomerado
$ref$	Refinamento da LA
$M$	Parâmetro de troca de figura no LA
$pop$	Número de população utilizado no LA
$N_{iter}$	Número de Iterações do LA
$rand$	Número aleatório entre 0 e 1
$x_i$	Variáveis de projeto, $i = 1, 2, \dots, d$
$h$	Largura solda
$l$	Comprimento solda



$t$	Profundidade da Viga
$b$	Espessura da Viga
$L$	Comprimento da Viga
$P$	Carga de Flambagem
$P_c$	Carga Crítica de Flambagem
$E$	Módulo de Elasticidade Longitudinal
$G$	Módulo de Elasticidade Transversal
[M]	Matriz Massa
[K]	Matriz Rigidez
$J(x)$	Função Objetivo avaliada pelas variáveis
$J(\omega)$	Função Objetivo avaliada pelo modo de vibração
$N_e$	Número do Elemento

## Simbolos Gregos

$\gamma$	Ângulo de rotação do eixo x do Plano Cartesiano
$\alpha$	Ângulo de rotação do eixo y do Plano Cartesiano
$\beta$	Ângulo de rotação do eixo z do Plano Cartesiano
$\rho$	Densidade específica
$\eta$	Constante de Difusão no DLA
$\nabla^2$	Laplaciano
$\tau$	Tensão de Cisalhamento
$\tau_{\text{máx}}$	Tensão de Cisalhamento Máxima
$\sigma$	Tensão de Flexão
$\sigma_{\text{máx}}$	Tensão de Flexão Máxima
$\delta$	Deflexão
$\delta_{\text{máx}}$	Deflexão Máxima
$\nu$	Coefficiente de Poisson
$\omega$	Frequência Natural
$\phi$	Modo de Vibração
$\lambda$	Autovalor
$\alpha$	Taxa de Dano

## Subscritos e Sobrescritos

<i>c</i>	criação
<i>cluster</i>	aglomerado
<i>e</i>	elemento
<i>FEM</i>	obtido pelo método de elementos finitos
<i>i</i>	i-enésimo
<i>p</i>	partícula
<i>ruido</i>	Sinal com adição de Ruído
<i>real</i>	Representa dados com valores conhecidos
<i>S</i>	Laminado Simétrico
<i>T</i>	Transposto
<i>t</i>	Iteração Atual
<i>t + 1</i>	Iteração seguinte
min	mínimo
max	máximo

# Lista de Abreviaturas

AG	Algoritmo <b>G</b> enético
ACO	<i>Ant Colony Optimization</i>
PSO	<i>Particle Swarm Optimization</i>
SFO	<i>Sunflower Optimization</i>
SHM	<i>Structural Health Monitoring</i>
MEF	<i>Método de Elementos Finitos</i>
MSE	Energia de Deformação Modal
NDI	<i>Non Destructive Inspection</i>
LA	Lichtenberg Algoritmo
FL	Figura de Lichtenberg
DLA	Difusão Limitada por Agregação
DBM	<i>Dielectric Break Model</i>
UP	<i>Uper Bound</i>
LB	<i>Lower Bound</i>
SFS	<i>Stochastic Fractal Search</i>
LSA	<i>Lightning Search Algorithm</i>
DE	Evolução <b>D</b> iferencial
SPO	<i>Sensor Placement Optimization</i>
Efi	Indenpedência Efetiva
KE	Energia Cinética
EVP	Produto de Autovetores

ADPR	Ponto de Resíduo Médio
BVID	<i>Barely Visible Impact Damage</i>
CAE	<i>Computer Aided Engineering</i>
INPI	Instituto Nacional de Propriedade Intelectual
SFS	<i>Stochastic Fractal Search</i>
HTS	<i>Heat Transfer Search</i>
BH	<i>Black Hole</i>
SFS	<i>Stochastic Fractal Search</i>
GSA	<i>Gravitational Search Algorithm</i>
WOA	<i>Whale Optimization Algorithm</i>
HS	<i>Harmony Search</i>
SEO	<i>Social Emotional Optimization</i>

# Capítulo 1

## INTRODUÇÃO

A otimização pode ser descrita como um processo de busca da melhor solução dentro de um conjunto de possíveis soluções (LOPES, 2007). A otimização clássica consta de métodos que dependem da continuidade e do gradiente de funções, sendo solucionadas na maioria das vezes por métodos analíticos e matemáticos. Como no mundo real a maioria das aplicações normalmente são não lineares e possuem soluções analíticas complicadas ou inexistentes, elas requerem ferramentas sofisticadas de otimização para serem enfrentadas (YANG, 2014).

Atualmente as soluções dos problemas de otimização reais se dão por ferramentas numéricas e algoritmos que passam por funções testes complexas, problemas de reconhecimento de padrões, identificação de danos, entre outros (CHENG *et al.*, 2018).

Alguns comportamentos encontrados na natureza como comportamentos dos animais ou plantas, conduções de exames de seres vivos, fenômenos físicos, características psicológicas ou biológicas relacionadas a humanos são fontes de inspiração para o desenvolvimento de algoritmos de otimização numérica que visam obter soluções ótimas com menor custo computacional e tempo de operação possível (MIRJALILI & LEWIS, 2016).

Este tipo de algoritmos gera soluções aleatórias (e por isto são chamados de estocásticos) no espaço de busca para um dado problema e continuamente as melhora a cada iteração, o explorando não ao acaso, mas com compensações que o faz se aproximar de ótimos globais.

Devido a esta característica, estes algoritmos são chamados de metaheurísticos (YANG, 2014). Cada metaheurística possui seus parâmetros próprios que regulam seu processo de otimização (NABIL, 2016).

As maiores dificuldades destes algoritmos estão em ter um correto balanço entre exploração e refinamento, isto é, conseguir escapar de mínimos locais e ainda conseguir melhorar a precisão das soluções já encontradas (OLORUNDA & ENGELBRECHT, 2008).

Para Yang (2014), dentre os diversos algoritmos não existem algoritmos bons ou ruins, mas um mais apropriado para determinado problema de otimização. Este fato sugere a necessidade de desenvolvimento de algoritmos que possam ser mais eficientes em determinadas situações. Ou seja, os processos de otimização numérica ainda possuem muito espaço para o desenvolvimento de novos algoritmos metaheurísticos.

Neste cenário, uma nova metaheurística é inspirada nos fenômenos físicos das tempestades. É comum em localizações próximas aos trópicos terem-se tempestades com muitos relâmpagos e quem já os observou já pôde verificar a sua velocidade, alcance e energia. Desta forma foi idealizada uma nova metaheurística capaz de trazer estas características para a busca numérica de soluções ótimas de problemas reais.

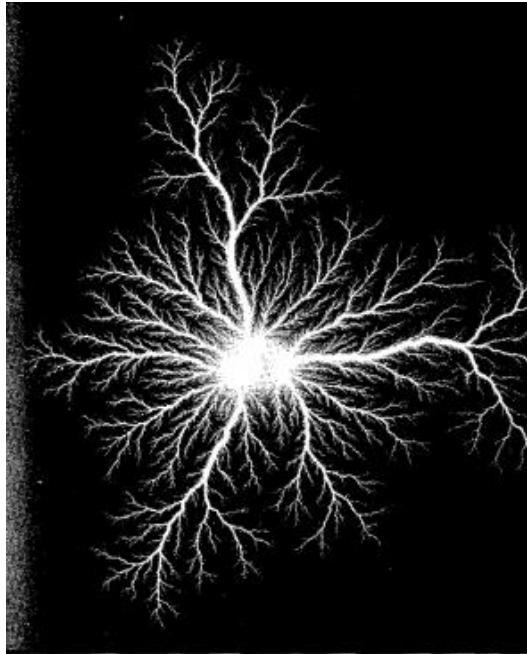
Na revisão de literatura pertinente ao tema, uma ferramenta para combinar com esta inspiração foi muito importante: as figuras de Lichtenberg. Estas são reproduções de propagações de descargas elétricas radiais em meios com resistência elétrica (dielétrico). Essas figuras trazem uma característica importante para o novo otimizador deste trabalho: o poder dos fractais.

Para Di Rocco (2009), fractais são formas irregulares que apresentam os mesmos padrões em qualquer escala na mesma figura. Portanto o otimizador foi programado para aproveitar desta característica e conseguir explorar e refinar o espaço de busca na procura do ótimo global. A Figura 1.1 mostra um exemplo de fractal encontrado na natureza devido à fenômenos físicos.

A metaheurística desenvolvida deve passar por uma série de funções de teste usuais na literatura para sua validação: dez funções de teste com restrições simples serão alvos do Lichtenberg Algoritmo e seus resultados serão analisados. Entre estas funções terão unimodais e multimodais. Esta é uma fase importante que marca o início de todo algoritmo de otimização numérica e é essencial para obter informações para aprimorar o algoritmo logo no seu início.

Validado, o algoritmo pode começar a enfrentar problemas mais complexos, como os de projeto na engenharia mecânica, onde a obtenção das variáveis ótimas de projeto ou próximas a ela podem implicar em menor massa ou menor custo, tendo-se como restrições não somente o

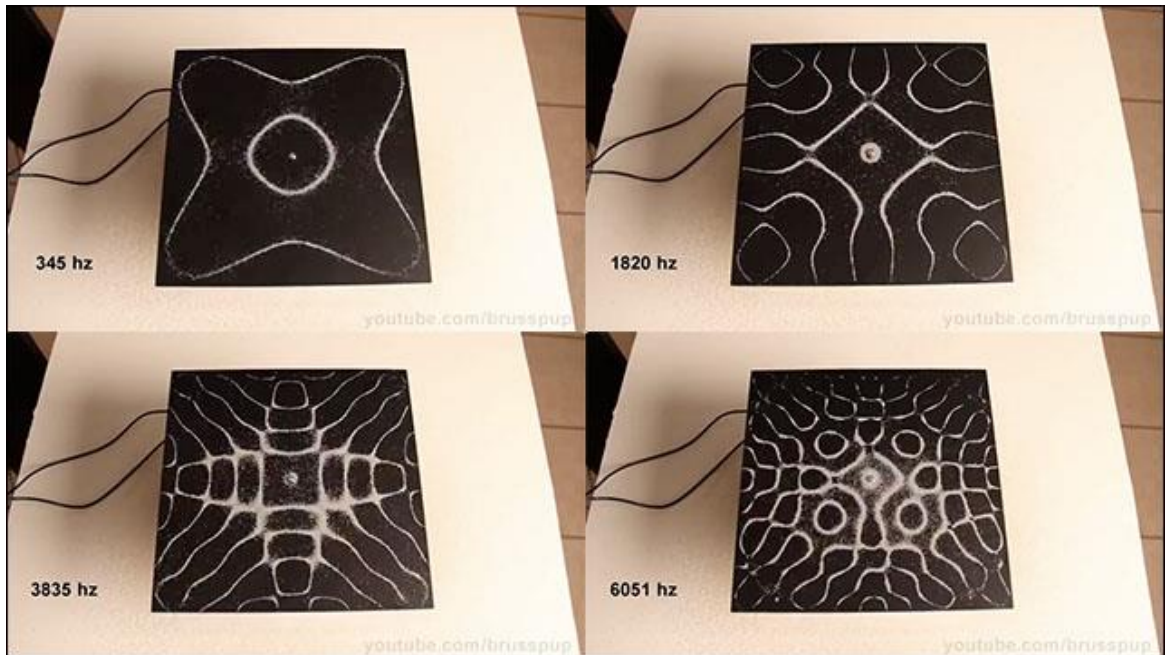
espaço de busca, mas algumas igualdades ou desigualdades que devem ser obedecidas. E como já dito, a maior parte dos problemas reais de engenharia são deste tipo.



**Figura 1.1** – Fractal formado devido à propagação de descarga elétrica por um meio resistente (MERRILL, 1939)

Por fim, o Lichtenberg Algoritmo será testado na aplicação de identificação de danos, onde o problema é ainda mais complexo e envolve uma metodologia mais elaborada onde o algoritmo sozinho é incapaz de solucionar. Ele precisa de um programa de elementos finitos e de leitura de sinais experimentais (ou não) para aquisição e processamento dos sinais de formas de vibração de uma estrutura. Isto, pois estruturas intactas vibram de maneira diferente se estão danificadas e existe uma metodologia que tendo referências destes modos de vibração, o otimizador pode identificar o dano (VO-DUY *et al.*, 2016).

Com base nisto, existem metodologias de monitoramento da saúde estrutural de estruturas, assunto que será melhor discutido em breve e tem suma importância na engenharia para melhorar a segurança do equipamento, reduzir tempo de inspeções que normalmente são demoradas e geram custos pela parada do equipamento ou por se negligenciar manutenções preventivas em tempos inapropriados (GOMES, 2019). A Figura 1.2 mostra um experimento com os modos de vibração de uma placa ao ser excitada com diferentes frequências.



**Figura 1.2** – Placas com modos de vibração definidos quando excitada com frequências próximas – fenômeno de ressonância. (ILLUSIONS AND SCIENCE, 2019)

## 1.1 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de um novo algoritmo de otimização que pode melhorar a eficiência e a eficácia na solução de determinados tipos de problemas. Após investigação destes algoritmos e estudo em alguns fenômenos encontrados na natureza, uma nova metaheurística foi idealizada e desenvolvida hibridizando classificações em seu processo de otimização como não visto antes em outros algoritmos.

Após o término, esta será validada com funções de teste usuais na literatura, será executado com um problema teste clássico de otimização em projeto de Engenharia Mecânica e comparada com outros algoritmos. Finalmente, será aplicada em um problema complexo de identificação de danos, com dano do tipo delaminação com foco em Material Compósito, sendo este um dos principais fins da metaheurística criada, mas não se limitando a ele. Com este propósito, ter-se-á revisões importantes e pertinentes sobre o assunto como Métodos de Monitoramento da Integridade Estrutural, posicionamento ótimo de sensores, delaminação e a formulação numérica do problema de identificação de danos.



## 1.2 Justificativas

Como mencionado, ainda há muito espaço para o desenvolvimento de novas metaheurísticas já que não existem algoritmos que possuam eficiência em todos os problemas reais de otimização. Trazendo-se o poder dos fractais através das ramificações de uma descarga elétrica e até mesmo da distribuição destas figuras em tamanhos diferentes no espaço de busca, vislumbrou-se a possibilidade de criar um algoritmo que une população e trajetória. A distribuição de pontos no espaço de busca pode assim ocorrer ora no espaço de busca inteiro e ora em regiões muito pequenas, conforme a iteração, o que faz deduzir a obtenção de excelentes resultados por balancear refinamento e exploração.

Não foram encontrados na literatura algoritmos deste tipo. Os que envolviam figuras de lichtenberg, relâmpagos ou fractais não eram otimizadores híbridos como o proposto neste trabalho, estes serão melhor explicados no Capítulo 3.

## 1.3 Estrutura da Dissertação

A distribuição dos conteúdos abordados nesta dissertação se dá por:

No Capítulo 1 fez-se uma breve introdução sobre a otimização, sua importância e algumas de suas aplicações. Também foi definido o escopo, a justificativa e a organização do trabalho.

O Capítulo 2 contém o referencial teórico pertinente para o desenvolvimento deste trabalho, contendo conceitos importantes de otimização e da identificação de danos em uma breve revisão da literatura sobre os temas e tópicos relevantes do assunto.

A nova metaheurística desenvolvida é apresentada detalhadamente no Capítulo 3.

O Capítulo 4 aborda a Metodologia do trabalho. Estão apresentados os problemas que a nova metaheurística enfrentou. São eles Problemas Lineares, Não Lineares e de Identificação de Danos, onde em alguns casos é analisada a sensibilidade de identificação de dano do algoritmo com severidades diferentes de dano e com a adição de ruídos nos sinais.

No Capítulo 5 são apresentados os resultados e as discussões dos três casos descritos anteriormente.

Por fim, no Capítulo 6 tem as conclusões e recomendações para trabalhos futuros.

## Capítulo 2

# REFERENCIAL TEÓRICO

Os referenciais teóricos mais importantes para o desenvolvimento deste trabalho consistem nos conceitos de otimização e da aplicação na identificação de danos – com ênfase em materiais compósitos. Este é o problema de otimização na engenharia mecânica escolhido para a aplicação da nova metaheurística desenvolvida neste trabalho, não limitando a isto.

### 2.1 Otimização

A Otimização é essencial em muitas aplicações, na engenharia, atividades de negócios (mercado) e indústria. Pode ser usada para minimizar a energia de consumo e custos, para maximizar o lucro, a saída, o desempenho ou a eficiência na solução de problemas. (YANG, 2014). Antes dos computadores, prevaleciam-se os métodos clássicos de otimização que eram baseados em gradientes ou derivadas.

Um exemplo de método clássico de otimização é o Método de Newton (Equação 1), que pode ser usado para funções de uma variável (objetivo simples) em que  $X_{t+1}$  é o valor futuro da variável na próxima iteração,  $X_t$  é o valor presente e que para a primeira iteração pode ser um

chute,  $f(x_t)$  é a “função objetivo” ou “função custo” a ser minimizada avaliada em  $X_t$  e  $f'(x)$  é a derivada de ordem um da função a ser minimizada avaliada em  $X_t$  (YANG, 2014):

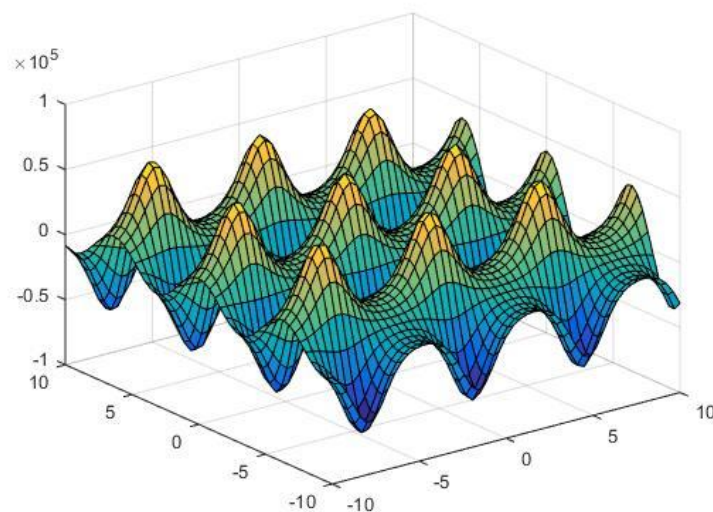
$$X_{t+1} = X_t - \frac{f(X_t)}{f'(X_t)} \quad (1)$$

O processo iterativo leva a função  $f(x)$  a um valor mínimo próximo do chute inicial de  $X_t$ , podendo ser um “mínimo local” ou “global”. Derivando deste procedimento, pode-se ter o método de Newton-Raphson em que se usa a derivada segunda da função a ser analisada e ainda, pode-se introduzir um parâmetro  $p$  para aumentar a convergência do método.

Estes são uns dos métodos mais antigos e clássicos de otimização de uma função, que como se pode observar, exige que a função seja contínua e derivável próxima do mínimo local e quanto mais variáveis, mais complexa fica a solução analítica. Esses tipos de algoritmos são frágeis e insuficientes para solucionar grande parte dos problemas reais.

Segundo Yang (2014), problemas reais podem ser compostos por funções descontínuas e não deriváveis em uma série de locais, podem apresentar mais de uma variável e composição de funções objetivos (“multiobjetivos”). Possuem “espaço de busca” determinado pelo número de variáveis e pelas “restrições” do problema. Estas restrições podem ser “lineares”, como o simples intervalo de busca da melhor solução para cada uma das variáveis, e/ou “não lineares”, em que esses limites de busca devem obedecer a combinações de variáveis (equações ou inequações).

Ainda, funções objetivo reais podem ser “multimodais”, isto é, apresentam múltiplos picos e vales, como mostrado da Figura 2.1, que fazem métodos clássicos caírem em mínimos ou máximos locais e ficarem presos, já que são baseados em gradientes ou derivadas.



**Figura 2.1** – *Bird Function* – Função Multimodal com Mínimo Global em (-1.58, -3.13)

Caminhar em direção ao ótimo global nem sempre é tarefa fácil. Embora haja problemas mono-objetivos em que se possa ter uma única solução ótima, os problemas multiobjetivos apresentam diversas soluções ótimas (CHIANDUSSI *et al.*, 2012). Um problema de otimização multiobjetivo consiste em encontrar um vetor de variáveis de decisão que otimize uma função vetorial cujos elementos representam funções objetivos originais, que geralmente estão em conflito entre si e ainda satisfaçam as restrições (OSYCZKA & KINDU, 1995).

Não existe um método direto para determinar se uma solução é melhor que outra. O método mais comumente utilizado na otimização multiobjetivo para comparar soluções é o chamado de relação de dominância de Pareto que em vez de determinar uma única solução ótima, leva a um conjunto de alternativas entre os objetivos. Estas soluções são chamadas de soluções ótimas de Pareto ou soluções não dominadas (JAIMES *et al.*, 2009). Gomes (2013) explica que uma solução ótima de Pareto é aquela em que não existe outra solução que irá provocar a redução em alguma função objetivo sem causar um aumento simultâneo em pelo menos um dos outros objetivos. Esta é a frente de Pareto.

A solução de um problema multiobjetivo é a determinação de um vetor de variáveis de decisão  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  que otimiza o vetor de funções objetivos  $F(\mathbf{X}) = \{f_1(x), f_2(x), \dots, f_n(x)\}$  dentro de uma região viável de soluções (BARIL *et al.*, 2011) sujeitas às restrições de igualdade  $h_i(x)$  ou desigualdades  $g_i(x)$  onde  $x_{min}$  e  $x_{max}$  são os limites que determinam o espaço de busca para cada uma das variáveis, ou vetor de variáveis. Como descrito na Equação (2) (GOMES *et al.*, 2013):

$$\begin{aligned} \min F(\mathbf{X}) &= \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})\} \\ \text{sujeito a: } &h_i(\mathbf{x}) = 0, i=1, 2, \dots, p \\ &g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, q \\ &\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \end{aligned} \quad (2)$$

Portanto, o desenvolvimento de algoritmos que independem do gradiente, da continuidade, do número de variáveis, que possam escapar de mínimos locais e que possam lidar com mais de um objetivo é necessário para o desenvolvimento dos métodos de otimização. Nesta perspectiva e com auxílio de ferramentas computacionais, diversos algoritmos inspirados em comportamentos encontrados na natureza e de iteração estocástica foram criados. Estes algoritmos possuem no processo de execução parâmetros ou ações que permitem exploração mais completa e variada do espaço de busca, além dos mínimos locais. Apresentam funções randômicas durante sua execução e cada execução do algoritmo leva a variáveis e ótimos

diferentes, mas podendo ser próximos. Ao contrário de “algoritmos determinísticos”, cujos resultados são sempre os mesmos independentemente do número de execuções. (YANG, 2014)

Ainda, alguns desses algoritmos estocásticos mais básicos podem ser chamados de *heurísticos*, pois funcionam simplesmente à base de tentativa e erro. Outros ainda mais completos e utilizados na otimização podem ser chamados de “metaheurísticos”, já que além de funcionarem à base de tentativa e erro, possuem certas compensações de randomização e pesquisa local que aprimoram a busca na direção do ótimo global. Alguns autores não consideram estas diferenças e utilizam o nome de algoritmos metaheurísticos para todos.

Para Yang (2014), existe ainda uma classificação para os algoritmos metaheurísticos que podem ser “baseados em trajetórias”, como o *Simulated Annealing* – criado por Kirkpatrick *et al.* (1983) e os “baseados em população” que são mais populares, tais como Algoritmo Genético (AG) - Um dos primeiros a serem criados e foi por Holland (1975); Colônia de Formigas (ACO) - criado por Marco Dorigo (1992); Enxame de Partículas (PSO) – desenvolvido por Kennedy & Heberhart (1995); *SunFlower Optimization* (SFO) – desenvolvido por Gomes (2018b). Ainda, podem existir os baseados em vetores, como o algoritmo de Evolução Diferencial (DE) – criado por Storn & Price (1997).

Estes algoritmos geram soluções aleatórias no espaço de busca para um dado problema e a melhora a cada iteração com o menor custo computacional possível. Os parâmetros que limitam o tempo, como pontos a serem avaliados na função objetiva e número de iterações dão à metaheurística capacidade de gerar melhores resultados para a função objetivo. Além destes, cada metaheurística possui seus próprios parâmetros que regula seu processo de otimização conforme foi desenvolvida (NABIL, 2016).

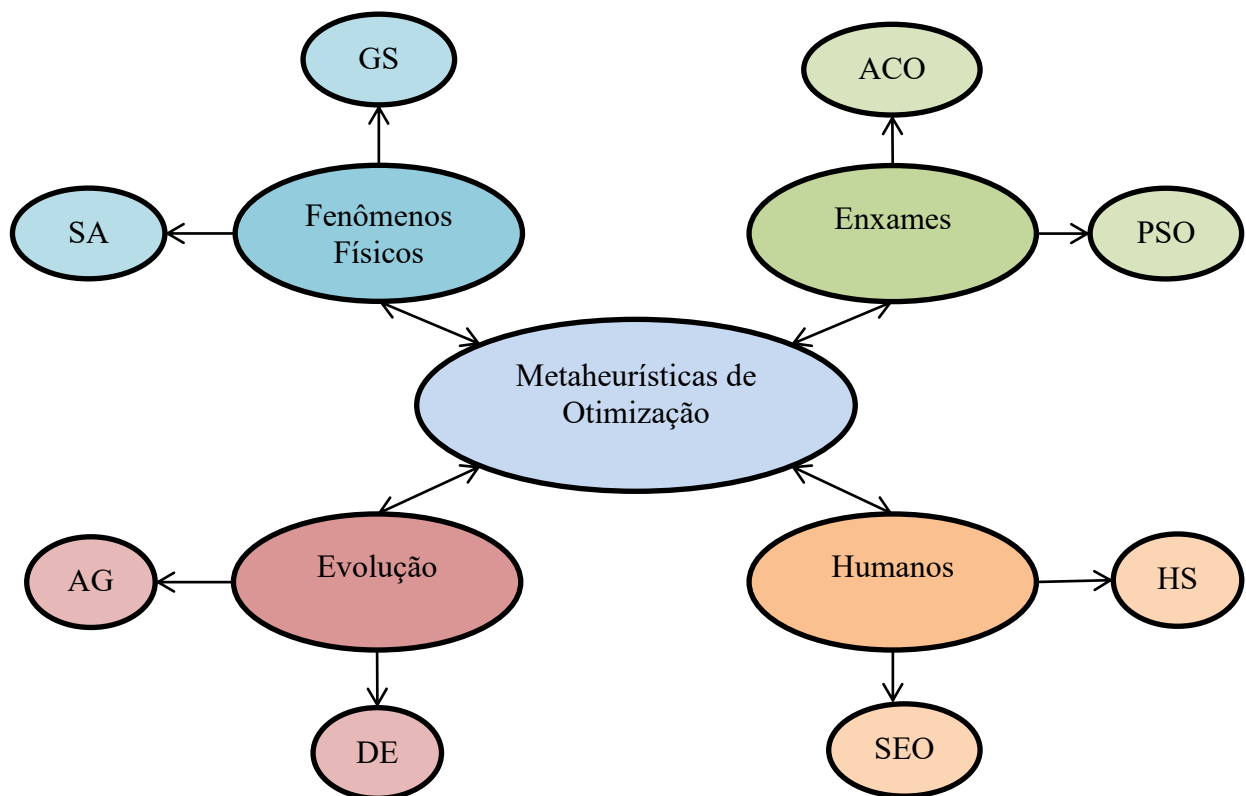
Cada metaheurística nasce com o propósito de reduzir a complexidade e o tempo de execução de um processo de otimização e por isto sempre há espaço para o desenvolvimento de novas técnicas. Portanto as metaheurísticas são desenvolvidas para solucionar problemas de computação sofisticados com tempo razoável e boa precisão (CHOPARD & TOMASSINI, 2018).

A natureza é rica de fontes de inspiração para a elaboração de metaheurísticas. Geralmente são elaboradas por processos de otimização encontrados na natureza. De maneira geral há quatro principais grupos que dividem as metaheurísticas conforme a inspiração para a sua criação: baseados em evolução (natureza), baseados em fenômenos físicos, baseados em comportamentos relacionados a humanos e baseados em enxames (HEIDARI *et al.*, 2019).

Os algoritmos evolucionários são inspirados em plantas, comportamento de animais, tais como o AG, o DE e o SFO já ditos anteriormente. Os baseados em fenômenos físicos são

inspirados em leis da física, como transferência de calor, força gravitacional, movimento de partículas, propagação de ondas, etc. Tais como o SA (mencionado anteriormente); o *Gravitational Search Algorithm* (GSA) – criado por Rashedi *et al.* (2009); o *Heat Transfer Search* (HTS) – criado por Patel & Savsani (2015); e o *Black Hole* – criado por Hatamlou (2013).

Os baseados em enxames são aqueles inspirados em comportamentos inteligentes de movimentos coletivos vistos em colônias ou sociedades. Esses comportamentos coletivos foram inspiração para o ACO; PSO; o *Whale Optimization Algorithm* (WOA) – criado por Mirjalili & Lewis (2016), entre outros. Finalmente, temos como exemplos de algoritmos inspirados em comportamentos humanos o *Search Harmony* (HS) – criado por Geem *et al.* (2001) e o *Social Emotional Optimization* (SEO) – criado por Xu *et al.* (2010). A Figura 2.2 resume o que foi exposto.



**Figura 2.2** – Diagrama relacionando as metaheurísticas, as fontes de inspiração e os algoritmos criados.

De maneira genérica os algoritmos baseados em enxames tem uma vantagem a mais que os baseados em evolução: eles preservam informações do espaço de busca em iterações subsequentes enquanto que os baseados em evolução descartam qualquer informação assim que

uma nova população é formada. Ainda possuem menos parâmetros de execução e são mais fáceis de implementar (MIRJALILI & LEWIS, 2016).

As maiores dificuldades destes algoritmos estão em ter um correto balanço entre exploração e refinamento (OROLUNDA & ENGELBRECHT, 2008). A primeira se refere à capacidade de sair de mínimos locais e explorar melhor todo o espaço de busca a fim de novas soluções. A segunda se refere à exploração localizada para melhorar a precisão das soluções que já foram encontradas (ELAZIZ & MIRJALILI, 2019). Desta dificuldade nasceu a proposta da nova metaheurística deste trabalho, a ter seus detalhes melhores explicados em breve.

Cada um destes algoritmos possuem modos de operações e parâmetros que os controlam quando estão sendo executados. Para Yang (2014), não existem algoritmos bons ou ruins, mas um mais apropriado para determinado problema de otimização. Este fato sugere a necessidade de desenvolvimento de algoritmos que possam ser mais eficientes em determinadas situações.

Segundo Fister (2013), no ano da publicação do seu trabalho existiam cerca de 40 algoritmos de otimização relevantes inspirados na natureza e nenhum deles é 100% eficiente para problemas complexos, concordando com a problemática levantada por Yang (2014). Hoje existem ainda mais algoritmos. É necessário aplicar algoritmos diferentes para um mesmo caso para poder avaliar a eficiência deles diante de um problema e compará-los entre si, além de se ter a possibilidade da criação de novos algoritmos para problemas específicos, já que na natureza é encontrada uma rica fonte de inspiração de diversos fenômenos que levaram milhões de anos para se desenvolverem e se aperfeiçoarem.

## **2.2 Identificação de Danos**

Materiais compósitos têm sido amplamente usados ao longo dos anos na indústria aeroespacial e em outras aplicações de engenharia onde o peso estrutural é uma das principais justificativas do seu emprego. Isto é devido às suas excelentes vantagens, tais como: elevada resistência e notável rigidez em relação à sua massa específica, além da elevada capacidade de suportar a fadiga e corrosão (KAW, 2005).

Porém, em serviço, podem apresentar mecanismos de falha, tais como ruptura de fibra, trincas na matriz ou delaminações. Sobrecarga estática, impacto, fadiga, erros de projeto e superaquecimento são alguns dos causadores destas falhas. A delaminação é considerada a maior “fraqueza” dos materiais compósitos laminados, pois pode se espalhar por todo o laminado de

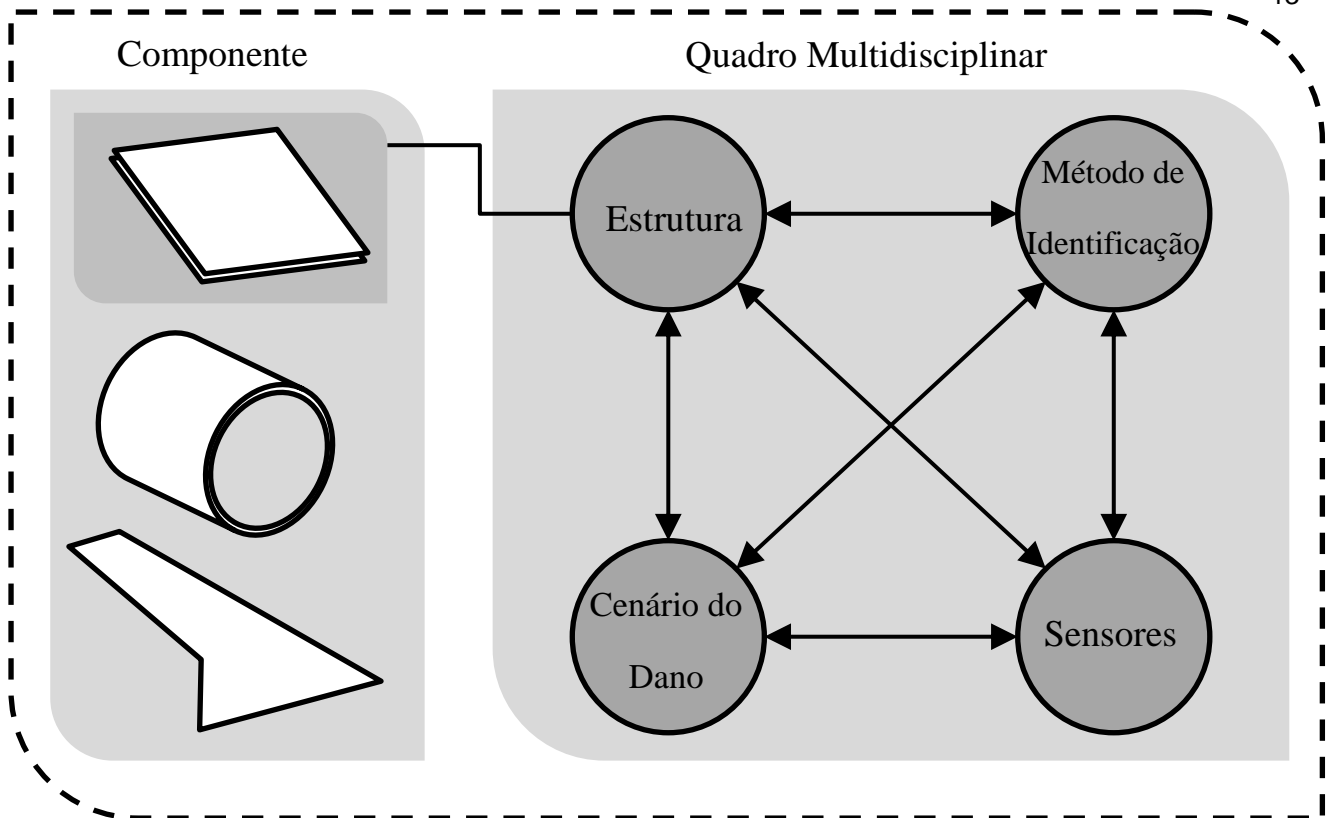
uma estrutura composta e levar a falhas catastróficas se não detectadas (CHAKRABORTY, 2005). Deve-se existir ações que avaliem esporadicamente estes sistemas para se evitar desastres.

Segundo Gomes (2017), inspeções visuais programadas podem ser demoradas, caras e exigem que os componentes sejam acessíveis. Assim, novas tecnologias são necessárias para evitar esses tipos de falhas e garantir economia, agilidade e segurança. Sistemas de monitoramento eficientes, tais como os métodos de monitoramento e inspeção de integridade estrutural (*Structural Health Monitoring* – SHM) permitem identificação precoce de danos ou degradação de componentes críticos, permitindo a adoção de ações corretivas por meio de ensaios não destrutivos (Non Destructive Inspection – NDI) e sensores integrados (HESLERHURST, 2014).

Para Gomes (2017), inspeções SHM que exploram métricas de vibração são métodos baseados no princípio que a degradação devido a danos em uma estrutura altera os parâmetros de vibração como frequências naturais, modos de vibração e amortecimento estrutural. Então, analisando-se os parâmetros de vibração de saída de um sistema, é possível identificar a presença de danos usando técnicas de modelagem inversa e inteligência computacional.

Neste trabalho, o método de identificação de danos já parte do pressuposto que o dano está na estrutura. Um ou mais danos do tipo delaminação com localização e intensidade conhecidas é induzido e o método de rastreamento de danos é testado quanto à capacidade de encontrar e quantificar este dano. Ambos os trabalhos são dedicados à identificação de danos estruturais em estruturas compósitas laminadas, mas não se restringindo a elas, principalmente na identificação de delaminações, que são uma perda local de rigidez. A Figura 2.3 traz os mecanismos de uma metodologia SHM onde é possível notar a inter-relação entre o método de identificação, os sensores, a estrutura e o cenário do dano para qualquer componente.





**Figura 2.3** – Quadro Multidisciplinar da concepção de um sistema SHM (GOMES, 2017)

O SHM aplicado nesta circunstância é de extrema relevância, já que segundo Garg (1988), a delaminação é uma importante forma de falha em materiais compósitos, que pode não ser visível na superfície estrutural e é capaz de afetar resistência e rigidez.

Então, o objetivo desta parte do trabalho é utilizar o método de identificação que se baseia na solução do problema inverso usando-se inteligência computacional e parâmetros modais estruturais com a intenção de identificar a possível localização e severidade de um dano estrutural. Enquanto Gomes (2017) utilizou o AG e a nova metaheurística SFO (GOMES, 2018d) no item “Métodos de identificação”, aqui será utilizado o LA e estes resultados serão comparados. O método de identificação que utilizou metaheurísticas é acoplado ao método de elementos finitos (MEF) como forma de se obter os parâmetros de resposta de vibração informados pelos sensores.

### 2.2.1 Identificação de danos utilizando o MEF

Segundo Worden & Friswell (2009), o monitoramento da saúde estrutural por meio do MEF geralmente trata da minimização de uma ou mais funções objetivo complexas, não lineares, não convexas e com vários mínimos locais usando dados de uma estrutura real (possivelmente

danificada) e dados oriundos do modelo de otimização onde muitas ferramentas podem oferecer a melhor solução. Dependendo do critério adotado, o método inverso tem a habilidade de detectar e identificar danos estruturais. Assim, algoritmos de otimização estocástica com desempenho promissor na resolução de problemas de otimização global têm recebido considerável atenção para fins de atualização de modelos de elementos finitos.

O MEF é um modelo matemático que traz respostas necessárias para a localização do dano. Segundo Gopalakrishnan *et al.* (2011a), o MEF é uma técnica numérica poderosa para resolver problemas governados por equações diferenciais em domínios complexos. É normalmente adotado para resolver problemas diretos em estruturas. Isto é, para um determinado carregamento e geometria (entrada), existem as saídas (deformação, tensão, frequências naturais, modos de vibração, etc). Reconhecer a saúde da estrutura a partir das saídas por meio do SHM é um problema chamado problema inverso.

Sabendo-se que o MEF se baseia na divisão de uma geometria em  $n$  elementos, a matriz de rigidez do elemento quando o MEF trata de dados modais é expressa na Equação 3 (YAN *et al.*, 2004):

$$[K_e] = \int_{V_c} [B]^T [A][C][A]^{-1} [B] dV \quad (3)$$

Sendo  $[C]$  matriz de constante elástica do material,  $[A]$  a matriz de transformação de coordenadas e  $[B]$  representa a matriz relacionada à função de forma do elemento.

Neste contexto, Jafarkhani & Masri (2011) observaram o desempenho de uma estratégia evolutiva na abordagem da atualização de modelos em Identificação de danos. O algoritmo de atualização do modelo foi capaz de detectar, localizar e quantificar com precisão os danos.

Nanda *et al.* (2012) usaram o PSO incremental para determinar a localização e a quantidade de trincas. A função objetivo foi formulada usando a frequência natural medida da estrutura intacta e a frequência obtida a partir da simulação por elementos finitos. O método desenvolvido foi capaz de detectar e estimar a extensão dos danos com boa precisão.

Mohan *et al.* (2013) utilizaram o PSO e o AG considerando frequência natural como resposta estrutural relacionada ao modo e teve resultados melhores. Cha & Buyukozturk (2015) utilizaram energia de deformação modal (MSE), que é sensível, em identificação de danos em estruturas tridimensionais de aço e afirmaram que danos pequenos e espalhados são difíceis de detectar utilizando-se como resposta propriedades modais. No entanto, o AG multiobjectivo híbrido proposto foi capaz de detectar a localização exata de pequenos danos induzidos na estrutura.

Braun *et al.* (2015) utilizaram diferentes versões do ACO para identificação de danos em ambiente ruidoso tendo-se bons resultados. Vo-Duy *et al.* (2016) utilizaram um algoritmo de evolução diferencial (DE) para identificar danos em uma placa laminada utilizando-se MSE e minimizando uma função que utilizava modos de vibração. Mesmo em condições de identificação de danos com cenários ruidosos tiveram-se excelentes resultados. Para cada um destes trabalhos, funções objetivos diferentes foram utilizadas, mas todas com o mesmo propósito: ser minimizada e identificar o dano.

Neste trabalho, o MEF é acoplado a técnicas inteligentes de processamento de sinais e ambas trabalharão em conjunto para localizar danos estruturais. Este modelo pode ser chamado de *finite element updating*, já que o MEF é executado várias vezes fornecendo a resposta do sistema (saída) a um algoritmo de otimização.

### **2.2.2 Posicionamento dos sensores na identificação de danos**

O problema básico da identificação de danos se dá a partir de respostas obtidas por meio de sensores distribuídos em estruturas. A qualidade destas respostas e portanto, a qualidade da identificação de danos, depende fortemente da localização e do número de sensores utilizados. Determinar esta configuração é um problema de otimização complexo (GOMES *et al.*, 2019)

Métodos práticos para determinar a localização dos sensores existem, como posicionamento dos sensores próximos aos anti-nós dos modos de vibração de baixa frequência, entretanto não apresenta resultados tão satisfatórios (BARTHORPE & WORDEN, 2009).

Diversos estudos existem na literatura acerca disto. Guo *et al.* (2004) apresentaram um índice de desempenho de otimização de posicionamento de sensores com base na identificação de danos com emprego de AG e maximização da matriz de informação de Fisher e obtiveram um algoritmo melhorado que foi capaz de determinar posições ótimas baseando em informações modais.

Kammer & Tinker (2004) estudaram o posicionamento de acelerômetros triaxiais e apresentaram uma nova técnica baseada em independência efetiva. Rao & Anandakumar (2007) sugeriram um algoritmo de posicionamento ótimo de sensores utilizando PSO e tiveram excelentes resultados. Yi *et al.* (2011) propuseram um método híbrido baseado em múltiplos métodos de otimização que se baseava em AG e obteve bons posicionamentos ótimos.

Gu *et al.* (2016) propuseram um algoritmo para um *layout* de sensores que integrou o método de energia cinética, método de independência efetiva e o critério de garantia modal e teve-se bons resultados.

Gomes (2017) buscou definir o número de sensores (que não pode ser inferior ao número de modos que se deseja identificar e com limite superior ao custo do sensoriamento), suas localizações e como deve ser medido o desempenho das diferentes configurações. Na otimização utilizou-se o AG para estas condições e sob quantidade pré-definida de 4, 6 e 9 sensores.

### 2.2.3 Monitoramento da Integridade Estrutural

O Monitoramento da Saúde ou da Integridade estrutural (SHM) trata de métodos que monitoram a estrutura, integridade e desempenho sem afetar a própria estrutura ou prejudicar sua operação. Esta tecnologia integra técnicas de avaliação não destrutivas utilizando sensores e materiais inteligentes para criar mecanismos de automonitoramento caracterizados por maior confiabilidade e longa vida útil estrutural. Aplica-se em sistemas com exigências críticas relativas ao desempenho estrutural e onde a parada da estrutura tem custos elevados, difíceis ou até mesmo impossíveis. (STEPINKSKY *et al.*, 2013). O assunto é tão estudado que hoje existem alguns axiomas que foram propostos por Worden *et al.* (2007):

- **Axioma I:** Todos os materiais são suscetíveis a falhas.
- **Axioma II:** A avaliação de danos requer a comparação entre dois estados.
- **Axioma III:** A identificação do dano pode ser feita por métodos de aprendizagem não supervisionada, porém, identificar o tipo de dano e sua severidade requer aprendizagem supervisionada.
- **Axioma IV:** Sensores não conseguem medir danos. Aquisição e processamento de sinais sim.
- **Axioma V:** Sem “processamento inteligente”, quanto mais sensível a medição é em relação ao dano, mais sensível também será submetido às condições ambientais.
- **Axioma VI:** O tamanho e a escala de tempo associadas ao início e a evolução do dano são as prerrogativas necessárias para o sistema SHM.
- **Axioma VII:** Existe uma relação entre a sensibilidade de um algoritmo ao dano e o ruído.

- **Axioma VIII:** O tamanho (ou dimensão) de um dano pode ser determinada por mudanças na dinâmica do sistema e é inversamente proporcional à faixa de frequência de excitação.

O procedimento deste trabalho obedece a todos estes axiomas. O axioma I é a razão da proposta. O Axioma II demonstra que se deve ter um estado de referência (podendo ser uma estrutura intacta) e outras. Os Axiomas III, IV, V e VI tratam da importância da boa aquisição de dados pelos sensores, utilizando-se algoritmos inteligentes. Técnicas computacionais serão utilizadas para contornar e satisfazer todos estes axiomas.

Segundo Stepinsky *et al.* (2013), o dano é uma excitação exógena que resulta em fluxo de energia e transformação, levando a modificações nos sinais de saída. É um problema inverso que analisa os sinais de saída e permite três níveis de ação: identificação, localização e avaliação. Diversos podem ser estes sinais de saída, mesmo que todos se baseiem em mudanças no comportamento dinâmico dos componentes monitorados.

## 2.2.4 Monitoramento por meio de vibrações mecânicas

Segundo Montalvão (2006), a principal ideia atrás das técnicas de identificação de danos baseadas em mudanças dinâmicas estruturais é o fato de que os parâmetros modais (frequências naturais, modos de vibração e amortecimento modal) são funções de parâmetros físicos (massa, rigidez e amortecimento). Assim é razoável assumir que a existência de um dano leva a mudanças nas propriedades modais da estrutura. Ainda, o monitoramento da estrutura permite detectar, localizar e até rever danos em estruturas mecânicas (GOMES *et al.*, 2018a).

Segundo Gopakal Krishnan *et al.* (2011b), estudos de técnicas baseadas em vibração podem avaliar a influência das reduções de rigidez nas frequências naturais causadas por danos. Estes estudos demonstraram que as frequências naturais não são indicadores melhores de danos, que geralmente mostram baixa sensibilidade e não permitem a determinação da localização dos danos. Também investigaram efeitos de danos localizados nos modos de vibração, formas de deflexão operacional e curvaturas correspondentes. Estes métodos baseados em propriedades modais vêm sendo muito atraentes, pois fornecem informações sobre o estado geral de saúde estrutural, bem como a possibilidade de identificar danos em regiões aproximadas.

Uma das vantagens de se usar Frequências Naturais é que estas podem ser obtidas sem um recurso de teste modal completo, uma análise espectral básica de um único teste de excitação

aleatório com um sensor de resposta pode ser suficiente. Também, segundo Worden & Friswell (2009), elas podem ser estimadas com muita precisão, cerca de 1% de erro. Com o dano, a perda de rigidez leva à queda da frequência natural.

Hearn & Testa (1991) demonstraram que a mudança da  $i$ -ésima frequência natural pode ser aproximada pela Equação 4:

$$\Delta\omega_i^2 = \frac{(\epsilon_N(\Phi_i))^T \Delta K_N (\epsilon_N(\Phi_i))}{\Phi_i^T M \Phi_i} \quad (4)$$

Sendo que  $M$  é a matriz massa,  $\Phi_i$  define o  $i$ -ésimo modo de vibração e  $\epsilon_N(\Phi_i)$  o vetor de deformação que é calculado pelo modo de vibração.  $\Delta k_N$  é a variação da matriz de rigidez devido à presença do dano. É considerada uma variação nula para a matriz massa.

Segundo Guan *et al.* (2017), o método que utiliza medições nas variações de frequências naturais estruturais de baixa frequência são feitas em um único ponto da estrutura em conjunto em conjunto com uma análise dinâmica do sistema para detectar o dano. A vantagem do método é que a frequência natural e a razão de amortecimento podem ser obtidas usando medição num único ponto na estrutura e são independentes da posição de medição escolhida. Porém, requer análise dinâmica suficientemente precisa da estrutura para obter as formas modais para estimar a localização e severidade do dano.

Por outro lado, segundo Gomes *et al.* (2018a), os modos de vibração são características de resposta estrutural que se opõem às frequências naturais, pois estes sofrem alterações locais na presença de danos localizados. Assim, podem ser mais eficazes na localização de danos. Porém, a tarefa de sensoriamento é mais árdua e completa com sensores de múltiplas respostas e instrumentação complexa e sofisticada.

Outro problema, segundo Worden & Friswell (2009), a precisão da medição é inferior em relação às frequências naturais e possui erro em torno de 5 – 10%. Mesmo assim tem sido mais usado devido sua sensibilidade maior a danos estruturais e por poder prover informações detalhadas a respeito da extensão e localização do dano estrutural.

O problema de autovalor de um sistema dinâmico pode ser definido pela Equação 5 (RAO & YAP, 2011):

$$[K - \lambda_i M] \Phi_i = 0 \quad (5)$$

Sendo  $K$  a matriz de rigidez,  $M$  a matriz de massa,  $\lambda_i$  o autovalor e  $\Phi_i$  o autovetor associados ao  $i$ -ésimo modo de vibração. Se alguma dessas propriedades estruturais (material ou geometria) é modificada devido a danos, as matrizes de massa ou rigidez também são afetadas. O autovalor

nada mais é que a frequência natural ao quadrado e os autovetores associados são os modos de vibração.

No caso de delaminação e alguns tipos de danos, assume-se que somente a matriz rigidez é afetada, então, segundo Gomes (2017), a variação no  $i$ -ésimo modo de vibração pode ser deduzida pela Equação (6):

$$\Delta\Phi_i = \sum_{r=1, r \neq i}^p \frac{-\Phi_r^T \Delta K \Phi_i}{\lambda_r - \lambda_i} \quad (6)$$

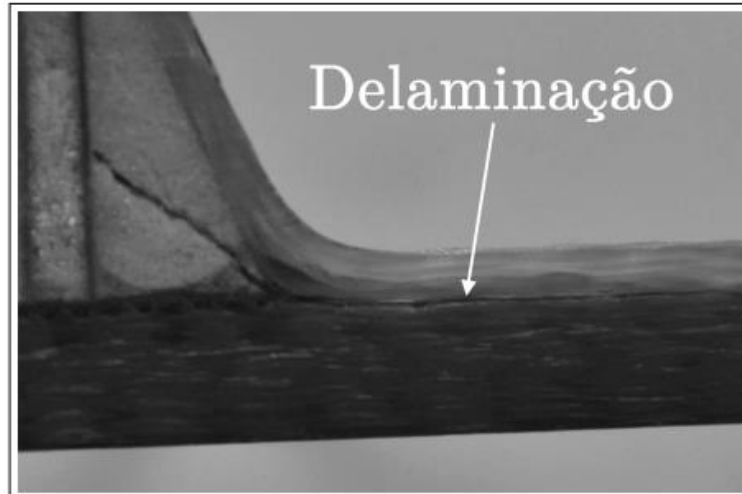
Como visto, é difícil a modelagem de um tipo de dano em seus detalhes. A equação da variação do modo de vibração é analiticamente e manualmente complicada de se solucionar. Por isto o uso de MEF se torna essencial para o desenvolvimento da identificação de danos. No caso de delaminação, a matriz de variação  $\Delta K$  será apresentada na próxima seção.

## 2.2.5 Delaminação

Segundo Kaw (2005), Material Compósito é quando dois ou mais materiais são combinados visando formar um material com funcionalidades melhoradas. Compósitos modernos usam fibras de alta resistência em uma matriz polimérica formando uma única camada. Os compósitos laminados são formados por empilhamento de camadas de diferentes orientações de uma maneira predeterminada. Entre estas camadas pode haver um modo de falha, a delaminação.

Segundo Raju & O'brien (2008), delaminação é a perda de rigidez no material devido a espaços que se formam entre as camadas adjacentes de um laminado, onde as camadas de ambos os lados podem ter orientações diferentes de fibras. A delaminação pode ser vista como uma fissura de interface entre duas laminas de um material anisotrópico.

Delaminações podem surgir não só de discontinuidades nos materiais e estruturas que dão origem a tensões interlaminares, mas também pode resultar de fatores de serviço como impacto de baixa velocidade por objetos estranhos. Embora tal dano ao impacto possa causar uma série de delaminações, ele pode não deixar qualquer indicação externa e é muitas vezes chamado de “danos de impacto meramente-visível” ou simplesmente *Barely Visible Impact Damage* (BVID) e podem gerar redução significativa de rigidez (LEE, 2000). A Figura 2.4 mostra uma delaminação.



**Figura 2.4** – Delaminação em material compósito (adaptado de HESLEHURST, 2014)

Garcia *et al.* (2015) frisam que a delaminação é um modo de falha perigoso, pois ocorre e cresce sob a superfície sem ser visível exteriormente, causando quebra da matriz e levando a alterações nas propriedades físicas de todo o laminado, tais como redução de rigidez e resistência. As técnicas SHM utilizam procedimentos não destrutivos para avaliar a saúde da estrutura e baseando-se em vibração, torna-se um método muito atraente pra este tipo de falha e não requerem nenhuma informação *a priori* para o local.

A Equação (7) que expressa a variação de rigidez numa matriz devido à delaminação pode ser escrita como (GOMES *et al.*, 2018a):

$$\Delta K = \sum_{k=1}^N \alpha_k K_k \quad (7)$$

Sendo que  $\alpha$  é um multiplicador escalar (entre 0 e 1) que modifica a rigidez  $K$  original local no elemento  $k$  e  $N$  é o número total de elementos da estrutura. Este determina a severidade do dano e é uma das variáveis do método de identificação de danos deste trabalho.

## 2.2.6 Método da identificação do dano

Definidos o uso do MEF para simulação dos dados modais simulando uma estrutura real, a melhor forma e número de sensores para adquirir sinais nesta estrutura, o tipo de dano a ser estudado e o parâmetro modal de vibração a ser analisado, falta definir o modelo de processamento inteligente destas informações.



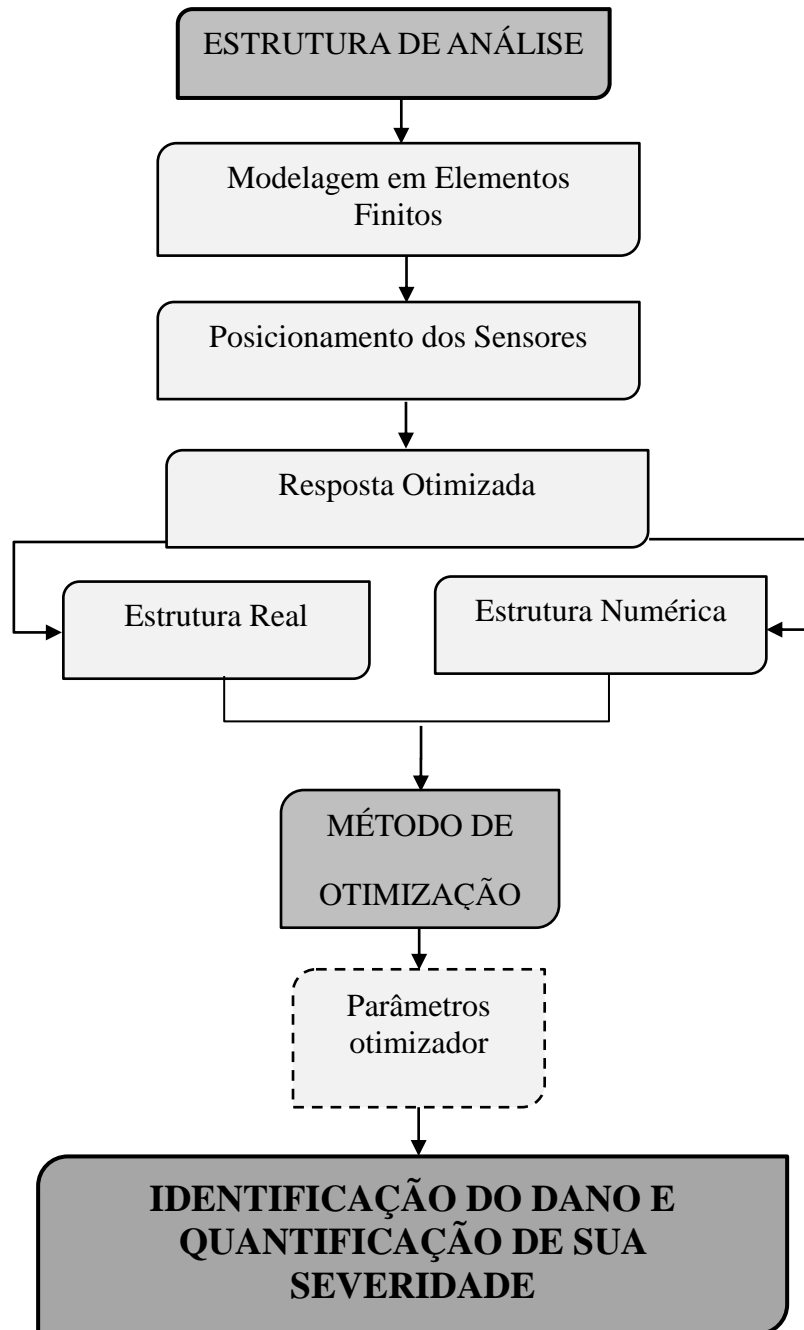
A avaliação do dano estrutural pode ser realizada por meio de comparação entre os dados medidos (reais) e os simulados. Para fornecer os dados simulados, um código numérico é necessário no qual um modelo do problema é usado por um algoritmo de problema inverso. Para o problema direto, um modelo é necessário para se obter informações sobre a resposta de interesse da estrutura com dano, para isto, as condições de contorno e posicionamento do dano são considerados. Para o problema inverso, um modelo é requerido para o procedimento de localizar o dano na estrutura, fornecendo algumas informações (parciais) sobre a quantidade de interesse em algumas localizações específicas (LOPES *et al.*, 2010).

Um modelo muito usado para a identificação são as metaheurísticas. Sabendo-se que o dano altera as respostas dinâmicas de uma estrutura que se difere de uma estrutura intacta, ou seja, sem a presença do dano. Segundo Worden *et al.* (2009), o objetivo do método inverso é empenhar-se na busca pela resposta de uma pergunta ao algoritmo de otimização, ou seja, a partir das respostas adquiridas de uma estrutura, espera-se que o otimizador apresente a possível localização do dano. O otimizador responde minimizando uma função objetivo apropriada, que se devidamente construída é capaz de encontrar a solução global, neste caso a identificação do dano. Gomes (2017) propõe a função objetivo representada na Equação (8):

$$J_{\Phi} = \sum_{i=1}^n \sqrt{\left(1 - \frac{\Phi_{i,s}^{\text{calculado}}}{\Phi_{i,s}^{\text{real}}}\right)^2} \quad (8)$$

Onde  $\Phi_{i,s}^{\text{calculado}}$  são os deslocamentos nodais obtidos em cada ponto de cada iteração pelo algoritmo de otimização referente ao modo de vibração  $i$  e  $\Phi_{i,s}^{\text{real}}$  os deslocamentos conhecidos da estrutura que possivelmente possui algum dano estrutural, obtidos nos pontos nodais ótimos ( $s$ ).

Gomes (2017) utilizou o Algoritmo Genético (AG) devido à sua grande consolidação e o *SunFlower Optimization* (SFO)(GOMES, 2018d) em seu trabalho. Foi fixado em  $10 \times d$  o tamanho da população, em que  $d$  é o número de variáveis para ambos os otimizadores. No modelo do autor, existem duas variáveis de projeto envolvidas,  $\vec{x} = (N_e, \alpha)$ , em que  $N_e$  é o número do elemento (sua posição) e possui uma adaptação no algoritmo para esta variável ser sempre um número inteiro e  $\alpha$  é a taxa do dano, uma medida de sua intensidade. Portanto é definida uma população de 20 indivíduos a cada iteração. E para ambos os algoritmos, foi adotado um número total de iterações (ou gerações) de 100. A Figura 2.5 traz um resumo deste procedimento.



**Figura 2.5** – Fluxograma resumido da identificação de danos por otimização  
(ADAPTADO de GOMES, 2017)

## Capítulo 3

# ALGORITMO DE LICHTENBERG

Como dito anteriormente, a natureza é repleta de fontes de inspiração para a elaboração de novos algoritmos de otimização. Qualquer pessoa em dias chuvosos já teve a experiência de contemplar um relâmpago e observar a sua velocidade de propagação, seu alcance e principalmente sua capacidade de se ramificar repetindo sua macroestrutura em microestruturas enquanto se desloca entre seus objetivos, o que traz uma propriedade importantíssima que será discutida adiante. Idealizar trazer esta velocidade, alcance e ramificações para atacar o espaço de busca em procura de pontos ótimos num problema de otimização balanceando exploração e refinamento pode resultar numa metaheurística de alto desempenho. Tem-se a proposta de uma nova metaheurística inspirada no fenômeno físico das tempestades.

### 3.1 Relâmpagos

Uma nuvem de tempestade é produzida a partir de vapor d'água que pode estar até 15 km de altitude. À medida que este vapor sobe por convecção devido à maior temperatura próxima da terra, transforma-se em líquido ou sólido e pode adquirir formas como gotas de água, gotículas de água super-resfriadas, cristais de neve, granizo leve, pedras de granizo e cristais de gelo em

função da altitude, temperatura e umidade relativa da atmosfera. Evoluem até se tornarem uma célula de tempestade e ter então várias células eletrificadas devido às colisões mútuas e fricções de água em seus diversos estados causadas pelos fortes movimentos ascendentes e descendentes dentro da nuvem. Surgem então centros de carga positivos e negativos. Por fim, o campo elétrico se intensifica e dá origem aos relâmpagos ou cargas atmosféricas (PINTO, 1987).

Para Pinto (1987), relâmpago é o fenômeno completo. Trovão é o som e o raio é o efeito luminoso. Ainda, podem existir relâmpagos de céu e de solo. Dentro dos de céu, existem os relâmpagos intra-nuvem (IN – mesma nuvem), entre-nuvens (EN – nuvens diferentes), no ar (AR – começa na nuvem e termina na atmosfera) e nos de solo, existem o nuvem – solo (NS) e o solo – nuvem (SN). Os IN e NS são os mais comuns.

Sabendo dos tipos de relâmpagos e adequando ao objetivo deste trabalho, percebe-se que os relâmpagos do tipo IN são os que podem ter propagação radialmente, principalmente quando há um tipo de carga no centro da nuvem e outra oposta nas periferias desta, como se pode ver na Figura 3.1. Ao contrário dos de tipo solo, que são unidirecionais majoritariamente. Segundo Naccarato (2001), descargas que se manifestam horizontalmente costumam ser mais ramificadas.



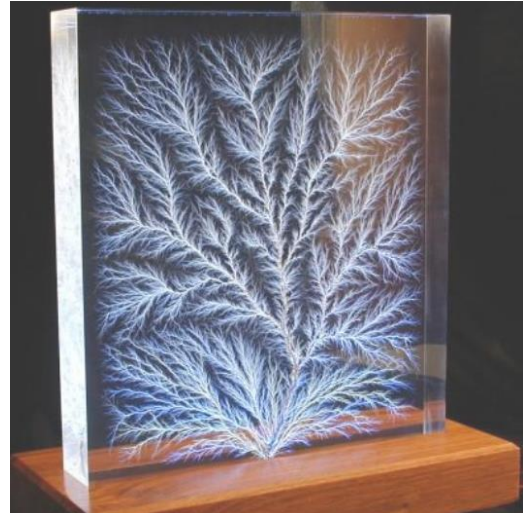
**Figura 3.1** – Relâmpago intra-nuvem (IN) (INPE, 2019)

Apesar da fonte de inspiração ser relâmpagos de uma nuvem de tempestade, a propagação radial e horizontal de descargas elétricas pode ocorrer em diversos meios. No próprio corpo humano, em casos de pessoas que se acidentaram com raios, há estas manchas gravadas sobre a pele, como pode ser vista na Figura 3.2 (a) (R7 NOTÍCIAS, 2018). Ainda, estas inscrições são muito comuns no artesanato, em madeira ou vidro, como pode ser visto na Figura 3.2 (b) (HYPESCIENCE, 2012). Portanto, temos a proposta de construção de uma nova meta-

heurística inspirada em fenômenos físicos – relâmpago – e as figuras geradas pela quebra da resistência do meio de propagação destas descargas (dielétricos). Modelos para auxiliar neste trabalho serão discutidos a seguir.



(a)

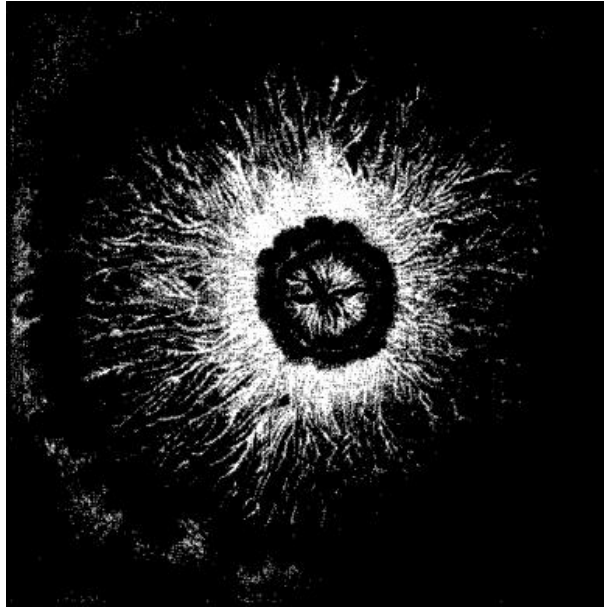


(b)

**Figura 3.2** – Exemplos de figuras oriundas de descargas elétricas em meios resistentes diferentes, sendo (a) em humanos (R7 NOTÍCIAS, 2018) e (b) em acrílicos (HYPESCIENCE, 2012).

## 3.2 Figuras de Lichtenberg (FL)

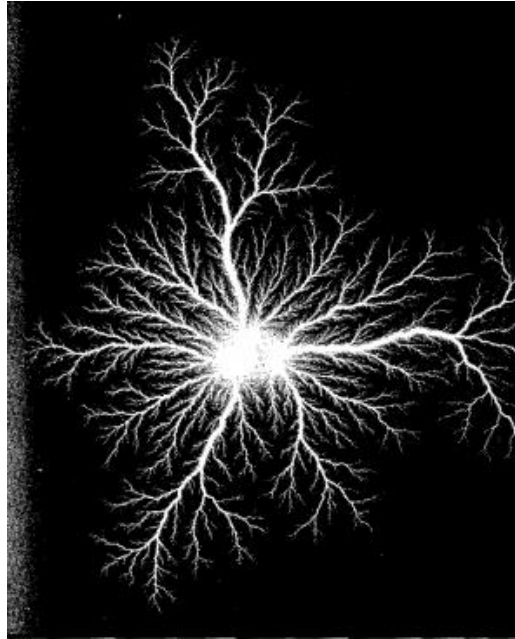
Georg Christoph Lichtenberg (1777) foi o primeiro a estudar este fenômeno de propagação de descargas elétricas em meios dielétricos (resistentes) – o que leva a figura a ter estes aspectos ramificados e tortuosos. Com instrumentos de medições incomuns devido à tecnologia da época, Lichtenberg fez um bolo de resina para experimentos eletrostáticos e deduziu que a polaridade e intensidade das faíscas geradas determinavam a forma da figura criada, que levou seu nome, Figuras de Lichtenberg (FL). A Figura 3.3 mostra a da primeira figura criada e publicada por Lichtenberg.



**Figura 3.3** – FL criada sobre placa de resina (LICHTENBERG, 1777)

Segundo Merrill (1939), a extensão e uma forma geral das FL's podem ser previstas pela magnitude e polaridade da tensão ou diferença de potencial aplicada. A impossibilidade de determinação das resistências em cada ponto, tendo-se as heterogeneidades do meio, determina o crescimento aleatório da figura para cada caso, ainda que sobre mesmo material e condições eletrostáticas. É um crescimento estocástico, portanto, aleatório.

Merrill continuou os estudos de Lichtenberg, mas aplicando em gases. Diversos experimentos positivos foram feitos, todos analisando o aspecto da figura como saída e o tipo de gás, parâmetros eletrostáticos e condições ambientais como parâmetros de entrada. A Figura 3.4 mostra um dos resultados utilizando-se descargas elétricas em gases dielétricos, a fim de se gerar uma figura tridimensional. É observável a representação da figura maior de maneira semelhante em estruturas cada vez menores. É a formação de um fractal, propriedade que será muito bem vinda à confecção do otimizador.



**Figura 3.4** – FL em gás tetra fluoreto sob tensão de 30kV e pressão 30,3atm. (MERRILL, 1939)

De maneira geral, Merrill (1939) identificou que quanto maior a pressão, maior a resistência do dielétrico gasoso e menor é o alcance e menos ramificações tem a FL. Também, quanto maior a tensão elétrica aplicada, maior o alcance e menos ramificações tem a figura. A polaridade também influencia muito, sendo a positiva gerando maior alcance e mais ramificações.

Estes autores e muitos cientistas que vieram posteriormente não trataram sobre um modelo matemático para construção da FL. Um algoritmo que pudesse gerá-la aleatoriamente. Esses trataram de condições físicas para criá-la em meios existentes e não analiticamente. Turner (2019) sugere que a FL pode ser construída por meio de um processo de crescimento aleatório com muitas partículas, formando um aglomerado ou *cluster*. Por ser um modelo estocástico, cada execução do algoritmo pode gerar figuras diferentes e a modelagem do raio pode ser semelhante, dada as várias variáveis que determinam o caminho de menor resistência que uma descarga elétrica percorre.

### 3.3 Métodos para Construir uma Figura de Lichtenberg

Para o desenvolvimento desta dissertação, uma busca por modelos matemáticos para que uma FL fosse modelada foi feita. Funções, algoritmos estocásticos, processos aleatórios, entre outros foram encontrados. Entretanto, os modelos mais fiéis para este escopo certamente foram os algoritmos baseados em crescimento de aglomerados. Estes processos são totalmente estocásticos, únicos e os dois tipos encontrados na literatura se comportam muito bem como fractais, como num relâmpago.

#### 3.3.1 Difusão limitada por agregação

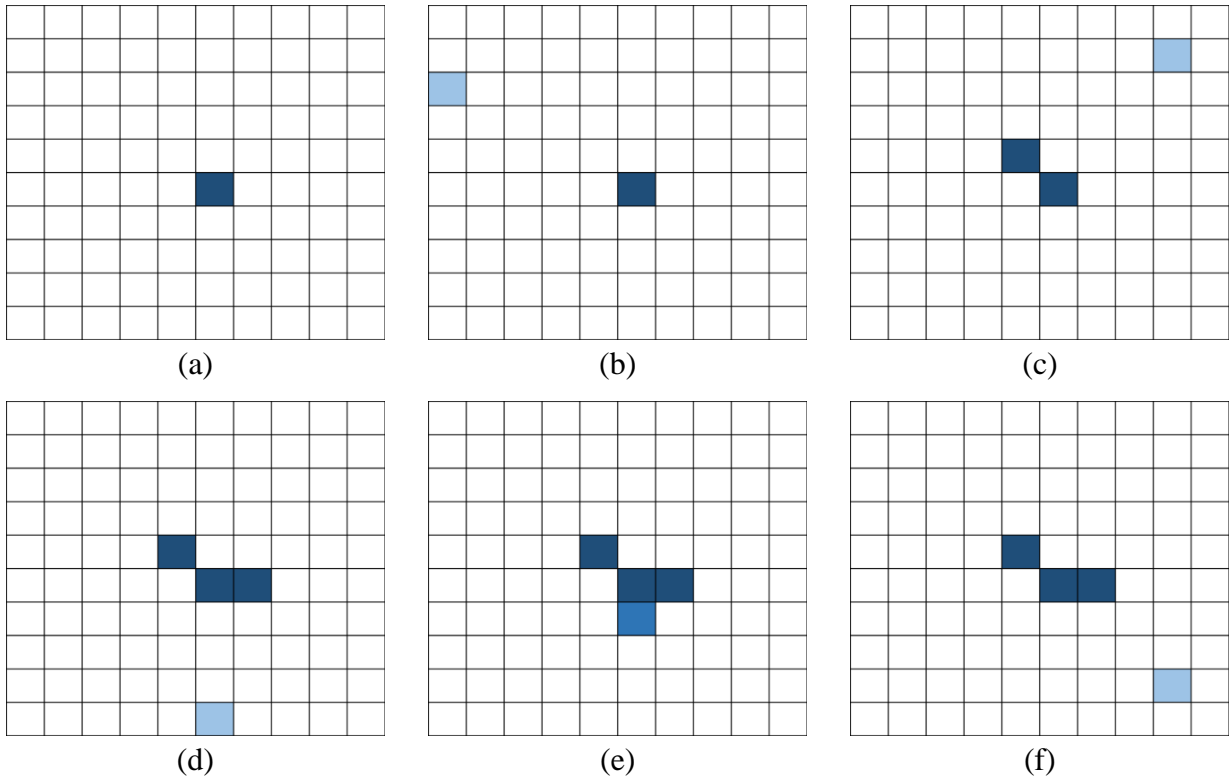
A Difusão Limitada por Agregação (DLA) foi apresentada por Witten & Sander (1981) como um modelo de simulação numérica para crescimento de aglomerados (*clusters*) partindo-se do princípio de partículas sólidas. Um modelo de crescimento de estruturas aleatório utilizado para o estudo de formação de figuras na quebra de resistência de um dielétrico. Um modelo de propagação de descargas elétricas.

Witten & Sander (1981) já previram o potencial do modelo como um fractal desde a sua origem, O modelo de construção se baseia em pegar uma região do espaço, podendo ser bidimensional e fixando-se uma partícula no centro. Uma segunda partícula é adicionada randomicamente a uma distância desta partícula fixa inicial e caminha randomicamente até encontrar com esta partícula central, se aderindo e virando parte do aglomerado fixo. Assim sucessivamente com  $n$  partículas até que o aglomerado seja construído.

Se a partícula ao caminhar randomicamente atingir as bordas do espaço pré-definido para construção do aglomerado, ela é removida e outra entra a caminhar no lugar. Se o aglomerado atinge as bordas, o algoritmo é finalizado. O tamanho do aglomerado e o número de partículas são definidos ao iniciar o algoritmo, mas a forma e tamanho são totalmente indefinidos e únicos após cada execução.

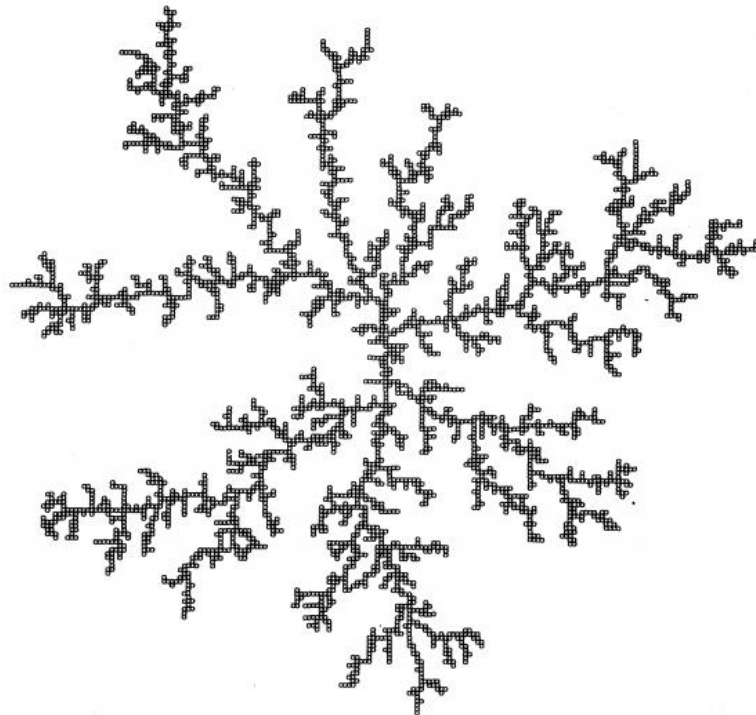
A Figura 3.5 mostra o modelo de crescimento do aglomerado nos primeiros segundos de simulação. Uma partícula fixa no centro (azul escuro), outra solta a uma distância caminhando randomicamente (azul claro). Ao encontrar o aglomerado pode ou não fixar-se por uma checagem de vizinhança (azul médio na indecisão). Se não se fixa, é eliminada e outra começa a caminhar. Se sim, faz parte do aglomerado e se torna azul escuro.





**Figura 3.5** – Iterações iniciais da formação do aglomerado segundo o método do DLA.

A Figura 3.6 traz o primeiro aglomerado simulado na história pelo modelo DLA.

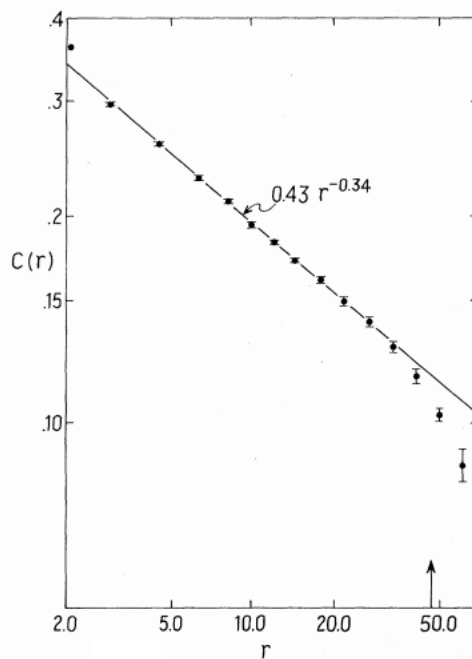


**Figura 3.6** – Aglomerada gerada pelo modelo da agregação limitada por difusão para 3600 partículas (WITTEN & SANDER, 1981)

Segundo Witten & Sander (1983), existe uma correlação para determinar a densidade do aglomerado que pode ser expressa como na Equação (9):

$$C(r) = N^{-1} \sum_r \rho(r')(r+r') \quad (9)$$

Em que  $N$  é número de partículas,  $\rho(r')$  é a densidade,  $r'$  é o raio do aglomerado,  $r$  é o raio do espaço em que o aglomerado está sendo construído. Witten & Sander (1981) levantaram um gráfico onde o valor de  $C(r)$  pode ser retirado em função do raio de construção do aglomerado e então, a densidade do aglomerado pode ser determinada. Veja a Figura 3.7.



**Figura 3.7** – Função de Correlação de Densidade do aglomerado em função do seu raio geométrico de criação (WITTEN & SANDER, 1981)

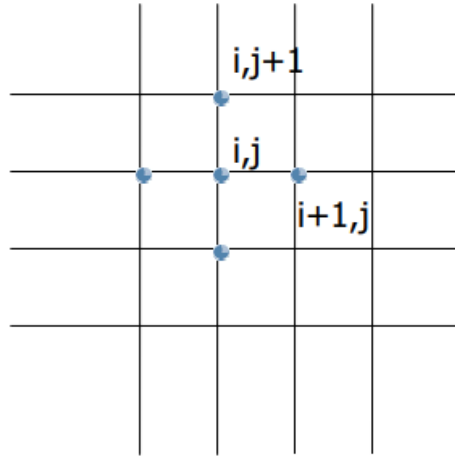
A probabilidade de uma partícula atingir um local  $\vec{x}$  no passo  $k$  é dado por  $u(\vec{x}, k)$  e pode ser descrita pela Equação (10):

$$u(\vec{x}, k+1) = \frac{1}{c} \sum_{\vec{1}} u(\vec{x} + \vec{1}, k) \quad (10)$$

Onde  $\vec{1}$  passa sobre os  $c$  vizinhos de  $\vec{x}$ . O número de vizinhos geralmente é quatro. Esta é uma versão discreta da equação contínua da difusividade, que pode ser escrita como originalmente como na Equação (11):

$$\frac{du}{dt} = \eta \nabla^2 u \quad (11)$$

Sendo  $\eta$  é a constante de difusão e  $\nabla^2$  é o laplaciano. O perfil futuro do aglomerado pode ser encontrado solucionando esta equação diferencial. Para exemplificar melhor o que foi abordado acima, a Figura 3.8 mostra esses processos de difusão e as soluções numéricas para o proposto.



**Figura 3.8** – Exemplificação Numérica da Equação diferencial, sendo  $i$  a direção horizontal e  $j$  a vertical

Deste modo, sendo “ $a$ ” a distância entre os pontos, temos que as equações numéricas de ordem um e dois são podem ser resumidas na Equação (12):

$$\frac{du}{dt} = \frac{u_{i+1,j,t} - u_{i,j,t}}{a} \quad \text{e} \quad \frac{d^2u}{dt^2} = \frac{u_{i+1,j,t} - 2u_{i,j,t} - u_{i-1,j,t}}{a} \quad (12)$$

Ainda, Witten & Sander (1983) adicionaram um coeficiente de aderência ( $S$  - *stick coefficient*) que varia entre 0 e 1. Este parâmetro determina se uma partícula com caminhada randômica ao encontrar o aglomerado se fixa ou não. Um valor aleatório entre zero e um é gerado neste encontro e se é menor que  $S$ , a partícula se fixa e se é maior, ela escapa. Desta maneira, quanto mais próximo de 1 é  $S$ , mais as partículas aderem ao aglomerado e mais rapidamente ele cresce atingindo a parada.

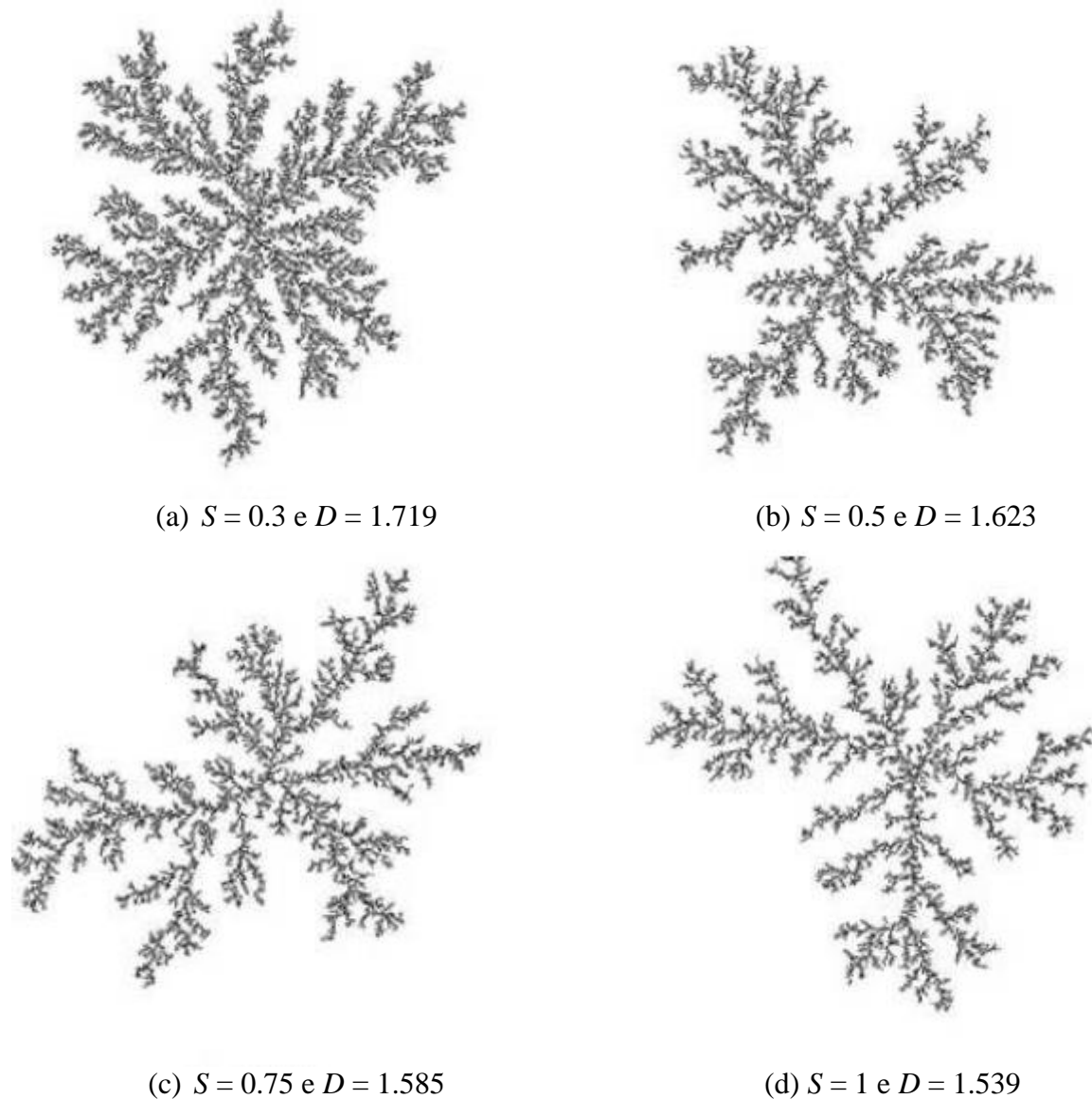
Segundo Meakin (1989), este modelo de crescimento de aglomerado pode levar a comportamentos complexos e caóticos que frequentemente podem ser descritos em termos de geometria fractal. Sem computadores seria impossível. E realmente, diversos experimentos na

literatura provam que as FL correspondem bem a parâmetros matemáticos que são utilizados em fractais.

Para Di Rocco (2009), fractais são formas irregulares que apresentam os mesmos padrões em qualquer escala na mesma figura. Fractais tem um parâmetro denominado Dimensão do Fractal ( $D$ ), que ao contrário do que pensam, não é sobre o tamanho do mesmo. É um parâmetro que mede estatisticamente o quanto um fractal preenche um espaço. Pode ser calculado pela Equação (13), em que  $N_{cluster}$  é o número de partículas e  $R_{cluster}$  é o raio médio que o aglomerado ocupa (pode ser calculado com Pitágoras e contagem das partículas numa direção)

$$D = \frac{\ln(N_{cluster})}{\ln(R_{cluster})} \quad (13)$$

Na grande maioria dos casos, para todas as simulações, a dimensão do fractal para as FL geradas pelo DLA fica entre um e dois. Na maior parte das vezes fica próximo de 1,7. A Figura 3.9 mostra propriedades interessantes. Ela aponta FL's geradas com coeficientes de aderência diferentes e o cálculo da Dimensão do Fractal, provando uma identidade para essas figuras. Note que quanto menor o  $S$ , maior a densidade física da figura e também maior a dimensão do fractal.



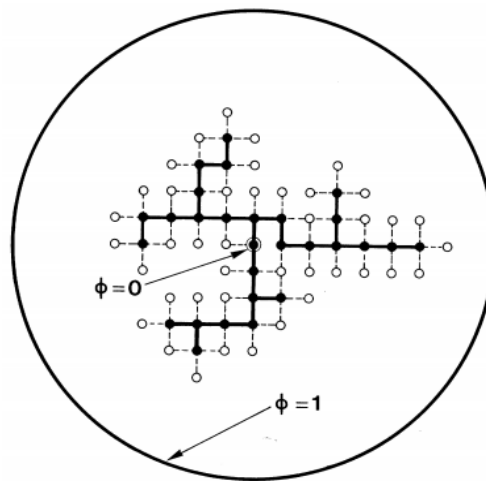
**Figura 3.9** – Influência do Coeficiente de Aderência ( $S$ ) na densidade do aglomerado e, portanto na Dimensão do Fractal ( $D$ ) (adaptado de DI ROCCO, 2009)

### 3.3.2 Modelo de quebra do dielétrico (DBM)

Um segundo e último modelo válido encontrado na literatura foi o modelo de quebra do dielétrico (ou DBM - *Dielectric Break Model*), que nada mais é que uma derivação do DLA incluindo potenciais elétricos. O DBM foi proposto por Niemeyer *et al.* (1984). Neste modelo, um ponto fixo no centro é dito como equipotencial e tem campo com valor nulo adotado. Distante, e pensando-se em gerar uma FL radial, faz-se uma circunferência com todos pontos

contidos nela com potencial igual a um. A partir do centro, regiões adjacentes são candidatas, para cada uma delas é calculada uma probabilidade em função da diferença de potencial.

Como resultado deste cálculo e escolhido randomicamente, um dos pontos é conectado ao central e passa a ser o ponto de partida do próximo cálculo. Este cálculo é feito ponto a ponto em toda a estrutura. A cada ponto esta probabilidade é calculada conforme a equação de Laplace para um caso equipotencial, ou seja, o laplaciano deve ser nulo. Desta forma, como no centro é nulo e nas bordas é um, calculando ponto a ponto, a probabilidade é que o aglomerado sempre cresça em direção as bordas e ao atingi-la, como esta tem potencial igual a um, o crescimento para. A Figura 3.10 ilustra este processo.



**Figura 3.10** – Evolução do aglomerado a partir do Modelo de Quebra do Dielétrico  
(NIEMEYER *et al.*, 1984)

No entanto, este modelo adiciona uma linha de cálculo ponto a ponto que consome mais a memória do computador que já estará sobrecarregada com as tarefas de um otimizador. Ainda, Irurzun *et al* (2002) fizeram uma série de estudos acerca da capacidade de construção de uma estrutura elétrica em árvore utilizando-se o DBM e a DLA e concluíram que não existe diferenças estatísticas, tendo o DLA dispersão ligeiramente melhor com Dimensão de Fractal de 1,66 enquanto o DBM com 1,75.

### 3.3.3 Busca na Literatura por otimizadores baseados em FL

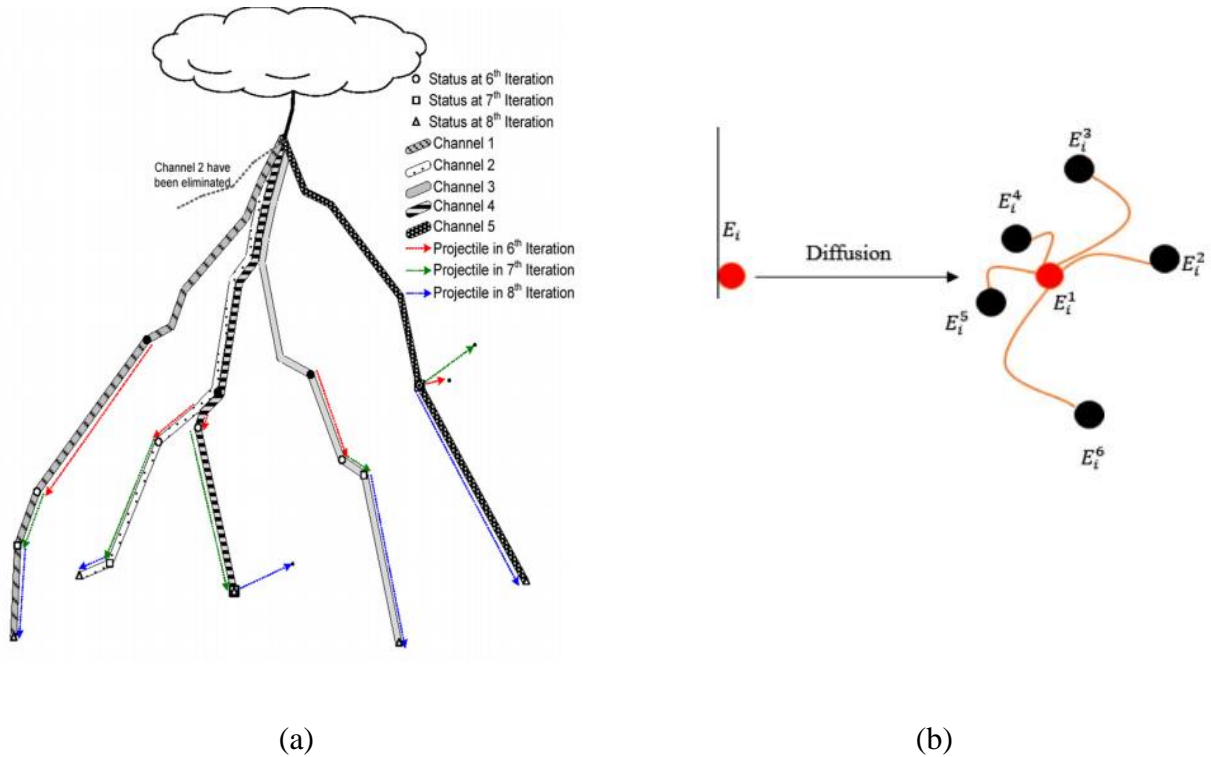
Diversos algoritmos metaheurísticos inspirados na natureza vêm sendo criados a cada ano. No entanto, ao revisar sistematicamente a literatura, nenhum deles foi criado unindo-se fractais e figuras de lichtenberg e principalmente, sendo um modelo híbrido entre trajetória e população. Dois algoritmos que compartilham de ideias semelhantes foram encontrados: *Stochastic Fractal Search* (SFS), criado por Salimi (2014) e *Lightning Search Algorithm* (LSA), criado por Shareef *et al.* (2015).

Salimi (2014) criou o SFS e o comparou com diversos outros algoritmos tendo-se excelentes resultados. O SFS se inspira no modelo de crescimento em fractal do DLA e também se apoia no DBM. Uma figura não é criada, mas é levada em conta a caminhada randômica das partículas e suas energias potenciais. É um algoritmo de população. Diversas partículas com uma energia potencial inicial são espalhadas e tendem a se difundir fixamente dividindo esta energia que também é avaliada conforme o valor do ponto na função objetivo. Estas partículas podem ter grandes ou pequenos saltos conforme a sua aptidão para saírem daquele local e continuarem a caminhar randomicamente. Lugares mais energéticos (conforme função objetivo) tendem a serem concentradores destas partículas com caminhada randômica e o aglomerado a crescer ali. A uma taxa, as partículas ou pequenos aglomerados menos aptos vão sendo eliminados a cada iteração.

Já Shareef *et al.* (2015) criaram o LSA se inspirando no comportamento do relâmpago, mas não usou modelos de crescimento browniano como o DLA e o DBM. É um algoritmo que se inspira nos canais térmicos e energéticos criados por uma descarga elétrica. São formados um canal principal e outros secundários, gerando uma pequena árvore entre a nuvem e solo. Este é o ponto de partida do algoritmo. Vários disparos de trajetória são feitos e canais que levam a *fitness* melhores tendem a se tornar principais e os que levam a piores tendem a ser eliminados. Esta direção de crescimento é definida pelos *fitness* de pontos aleatórios.

A Figura 3.11 traz um esboço do modo de funcionamento destas duas metaheurísticas.

No entanto, nenhuns destes algoritmos são compostos por métodos de busca baseados em trajetória e população ao mesmo tempo. A ideia principal da metaheurística criada é unir estes dois métodos de busca visando distribuir pontos a serem avaliados na função objetivo em regiões ora ocupando todo o espaço de busca e ora regiões muito menores, gerando excelente balanceamento entre exploração e refinamento.



**Figura 3.11** – Algoritmos com inspirações semelhantes às do LA: (a) LSA (SALIMI, 2014) e (b) FSA (SHAREEF, 2015)

### 3.4 Algoritmo de Lichtenberg (LA)

O Algoritmo de Lichtenberg é uma nova metaheurística inspirada no fenômeno físico da propagação radial de um relâmpago do tipo Intra-Nuvem. Desta forma, pode-se usar como modelo as Figuras de Lichtenberg, pois trazem uma propriedade importante para o modelo: o poder dos fractais. Desta forma, desenvolveu-se um algoritmo que usa dos fractais como forma de ramificações das trajetórias que ele forma, mas também se tendo estas FL geradas em escalas diferentes, melhorando a exploração por novas soluções e refinando o resultado das já encontradas tendo-se as populações sendo distribuídas em espaços diferentes a cada iteração. Desta forma ele é híbrido, pois tem trajetórias bem definidas e de tamanhos diferentes a cada iteração e é composto por muitos pontos. Alguns destes pontos são selecionados a cada iteração para serem avaliados na função objetiva.

Dado as explicações do item anterior, o DLA acabou sendo o modelo escolhido para a construção da FL para o Algoritmo de Lichtenberg (LA), ou *Lichtenberg Algorithm*. A



construção ocorre como escrito no modelo. Uma matriz binária (0 e 1) é construída como um mapa e no centro é fixada uma partícula, representada pelo número um. Toda estrutura física do aglomerado é um elemento matricial de valor um e os espaços vazios possuem valor nulo. Esta forma matricial pode ser representada por uma figura *Bitmap* (preto e branco). O espaço para construção da figura é definido pelo raio de criação ( $R_c$ ) e a partir dele é construída a matriz com números de linha e coluna iguais a duas vezes o  $R_c$  (diâmetro) mais 2, já que há uma partícula no centro e uma de limite exterior. Algumas vezes, para aumentar o número de partículas numa mesma direção e diminuir sua distância discreta, o  $R_c$  pode ser acrescido por um valor qualquer.

Partículas são randomicamente soltas e se atingem o aglomerado, possuem uma probabilidade  $S$  de se fixarem. A partícula “caminha” sendo plotada randomicamente, radialmente e como se estivesse num plano cartesiano a partir do centro e se estabelecendo em qualquer local do mapa arredondando-se a posição para um elemento de matriz com linha e coluna. Neste ponto ela pode se adicionar em 4 posições (cima, baixo, direita ou esquerda) se, e somente se houver outra partícula ao seu lado confirmada por uma checagem lateral. Se ela atinge um raio um pouco maior que o  $R_c$ , é exterminada e outra inicia novamente a caminhada randômica. Isto acontece até que todas as partículas determinadas na entrada  $N_p$  estejam contidas no aglomerado ou que este atinja o seu limite de construção. Este processo é muito lento e é um dos inconvenientes do algoritmo, mesmo que figuras não sejam plotadas, a matriz binária tem que ser construída e este processo têm que ocorrer para que o aglomerado cresça pelo DLA.

Cada execução da construção da FL gera figuras totalmente diferentes e o parâmetro  $S$  influencia consideravelmente na densidade da figura. Após a finalização da FL, é contado o número de partículas do aglomerado ( $N_{cluster}$  - que pode ser inferior ao requisitado) e é calculado o seu raio final ( $R_{cluster}$  - que é um raio médio calculado pelo Teorema de Pitágoras contando-se o número de partículas contidas no aglomerado em cada direção), variáveis utilizadas para calcular a Dimensão de Fractal ( $D$ ).

Os resultados desta programação são muito fidedignos aos encontrados na literatura, com dimensão de Fractal próximo de 1.7, como comentado no capítulo anterior. Até o momento, três importantes parâmetros foram apresentados e são exclusivamente para a construção da FL:  $N_p$ ,  $S$  e  $R_c$ . A Figura 3.12 mostra as FL em *Bitmap* com diferentes valores de  $S$  e mesmos  $N_p$  e  $R_c$ . Note a diferença de densidade (e Dimensão do Fractal) devido ao coeficiente de aderência. A Tabela 3.1 mostra os resultados.

O Anexo A traz o código completo e pode-se ver a programação acima descrita.

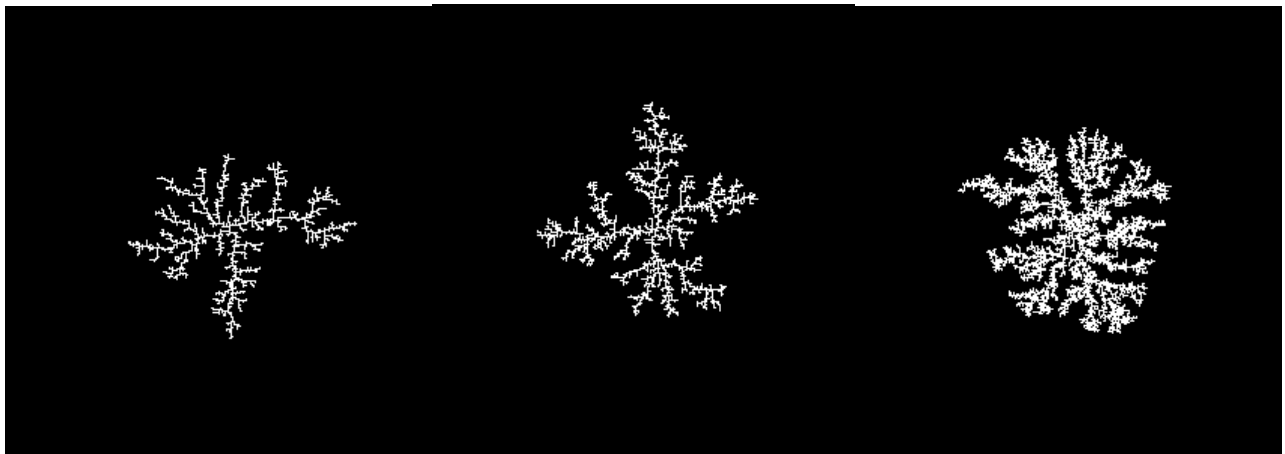
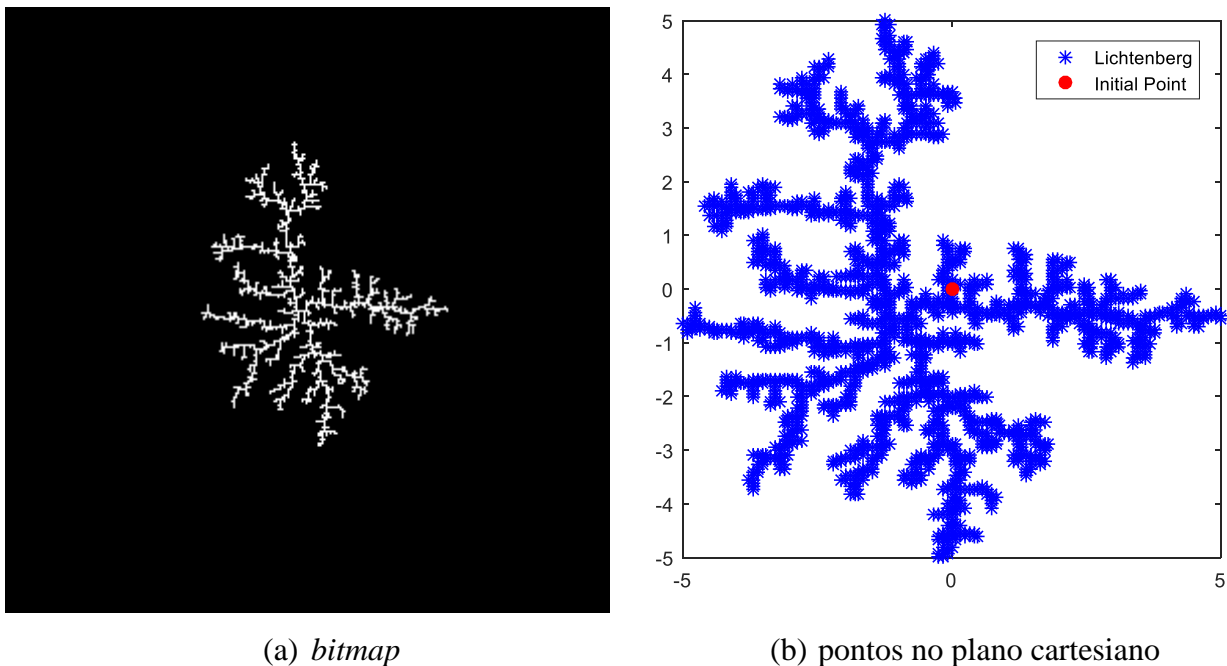
(a)  $R_c = 100$  e  $S = 1$ (b)  $R_c = 100$  e  $S = 0.5$ (c)  $R_c = 100$  e  $S = 0.1$ **Figura 3.12** – Figuras de Lichtenberg em *bitmap* geradas no início da simulação do LA**Tabela 3.1** - Resultados Figuras *Bitmap*

Figura	$R_c$ (entrada)	$N_p$ (entrada)	$S$ (entrada)	$N_{cluster}$	$R_{cluster}$	$D$
a	100	10000	1	1851	4.4198	1.6995
b	100	10000	0.5	2771	3.8834	1.7974
c	100	10000	0.1	2125	4.5870	1.849

Como dito anteriormente, quanto menor o  $S$ , menor a probabilidade das partículas se fixarem ao aglomerado e a densidade deste aumenta. Isto também influencia muito no tempo de construção da figura de lichtenberg: para a simulação da figura (a) o programa levou cerca de 140 segundos, já para o caso (c), cerca de 400 segundos. Quase o triplo do tempo, mesmo tendo o mesmo número de partículas e raio de criação.

Possuindo uma matriz binária ou figura *bitmap*, o MATLAB<sup>®</sup> por meio de da função *ind2sub* consegue encontrar os valores correspondentes a um na matriz, cortar todas as linhas ou colunas inteiramente nulas e extrair pontos para um plano cartesiano transformando o elemento pela sua localização (linha e coluna) em coordenadas. Então, num primeiro momento, a figura extraída é plotada no tamanho exato do espaço de busca e o seu centro do *bitmap* no centro do mesmo utilizando um fator de escala calculado apropriadamente para cada eixo, tocando todos os limites em todas as direções do espaço de busca. Podendo também ser maior que ele, se desejar. A Figura 3.13 ilustra isto.

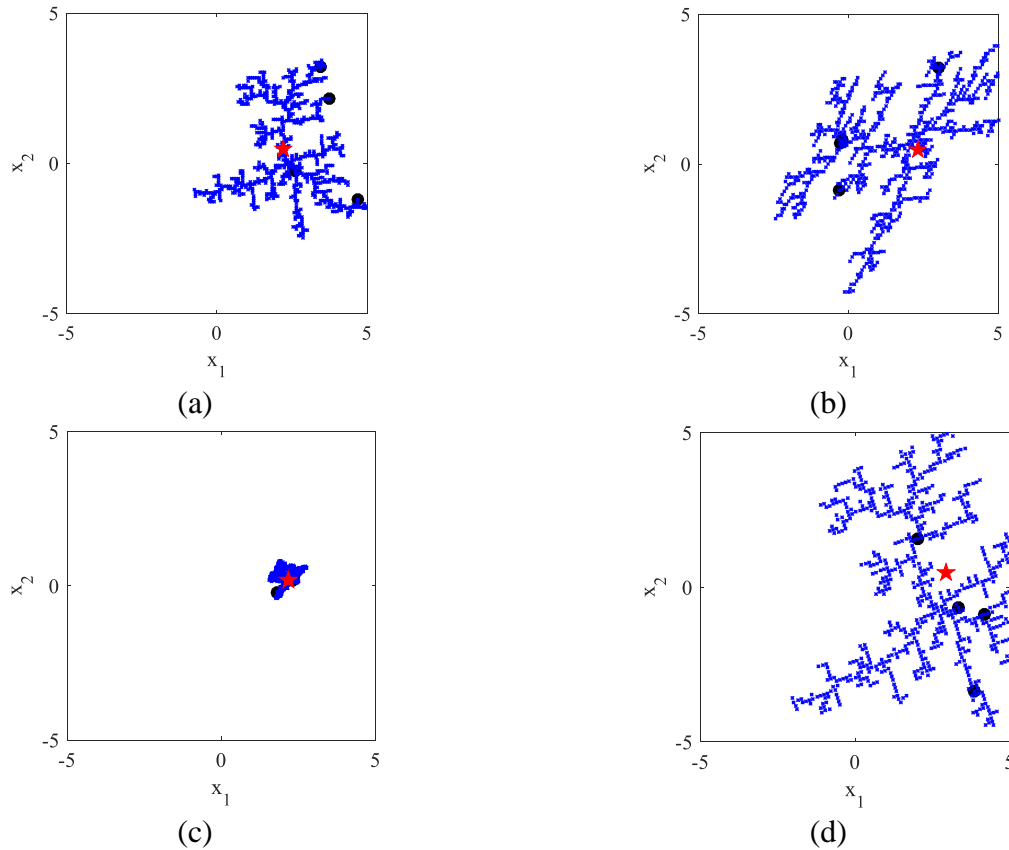


**Figura 3.13** – Transformação das figuras *bitmap* em coordenadas plotadas no intervalo  $[-5, 5]$

Se a FL ficasse desta maneira, é nítido notar que os pontos representados por ela no espaço seriam sempre discretos. Há sempre uma distância entre um ponto e outro e ao usá-la num otimizador, lacunas existiriam e a precisão do otimizador seria comprometida. Para isto, quatro medidas foram tomadas.

A primeira é que um fator de escala variável e randômico, um *rand* - sempre menor ou igual a um (dado que a primeira figura é do tamanho do espaço de busca), atua sempre em cada iteração da execução do programa. Ou seja, a cada iteração a figura é plotada em um tamanho diferente, podendo ter desde o tamanho do espaço de busca até menos que 1% do tamanho dele. Isto diminui muito a distância entre pontos a cada iteração e melhora a eficiência do LA.

Uma segunda medida é a adição de rotação randômica para a FL a cada iteração. Um ângulo é randomicamente dado (novamente pelo uso de um *rand* para cada um dos ângulos necessários) e toda a estrutura é rotacionada conforme as equações de rotação de pontos em equações cartesianas. Isto ocorre sempre para as figuras 2D e 3D e quando é mais de 3 variáveis, o espelhamento destas figuras é dado tendo-se ainda rotações diferentes para as figuras dos planos anteriores. Isto melhora bastante os resultados, pois evita a simples repetição dos pontos. A Figura 3.14 ilustra algumas iterações mostrando a primeira e a segunda medida.



**Figura 3.14** – Exemplificação das escalas e rotações randômicas para uma mesma figura de lichtenberg

As Equações de Rotação para Plano Cartesiano estão nas Equações (14), em que  $\alpha$  é o ângulo de rotação:

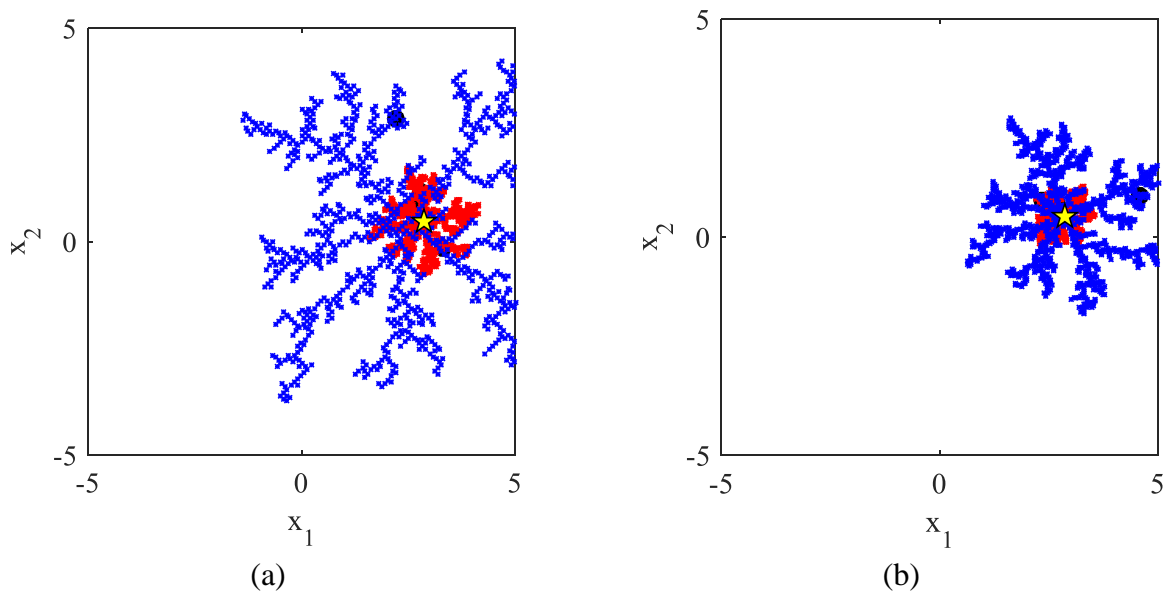
$$\begin{aligned}x' &= x \cdot \cos \alpha - y \cdot \operatorname{sen} \alpha \\y' &= x \cdot \operatorname{sen} \alpha + y \cdot \cos \alpha\end{aligned}\quad (14)$$

Já as equações de Rotação para Espaço Cartesiano estão nas Equações (15), em que  $\gamma$ ,  $\alpha$  e  $\beta$  são as rotações nos eixos  $x$ ,  $y$  e  $z$  respectivamente:

$$\begin{aligned}x' &= x \cdot \cos \gamma \cdot \cos \beta + y(\cos \alpha \cdot \operatorname{sen} \beta + \operatorname{sen} \alpha \cdot \operatorname{sen} \gamma \cdot \cos \beta) + \\&\quad z \cdot (\operatorname{sen} \alpha \cdot \operatorname{sen} \beta - \cos \alpha \cdot \operatorname{sen} \gamma \cdot \cos \beta) \\y' &= x \cdot (-\cos \gamma \cdot \operatorname{sen} \beta) + y(\cos \alpha \cdot \cos \beta - \operatorname{sen} \alpha \cdot \operatorname{sen} \gamma \cdot \operatorname{sen} \beta) + \\&\quad z \cdot (\operatorname{sen} \alpha \cdot \cos \beta + \operatorname{sen} \gamma \cdot \operatorname{sen} \beta) \\z' &= x \cdot \operatorname{sen} \gamma + y(-\operatorname{sen} \alpha \cdot \cos \beta) + z(\cos \alpha \cdot \cos \gamma)\end{aligned}\quad (15)$$

A terceira foi a adição de um refinamento (*ref*), um parâmetro de entrada que pode ser de 0 a 1 e é um criador de uma segunda FL (vermelha) com mesmo ponto de disparo, rotação e fator de escala da FL principal (azul), no entanto, com a escala menor para melhorar a busca

local, reutilizando as vantagens de um fractal. Se o  $ref = 0$ , apenas uma FL atua no otimizador, a global. Antes de se entender a quantidade de pontos da FL que serão distribuídos para a FL global e local durante as iterações, é preciso entender o conceito de população. A Figura 3.15 mostra dois exemplos de iterações com  $ref = 0.3$ . A estrela amarela é o ponto ótimo da iteração anterior e o ponto de disparo da FL da iteração atual.

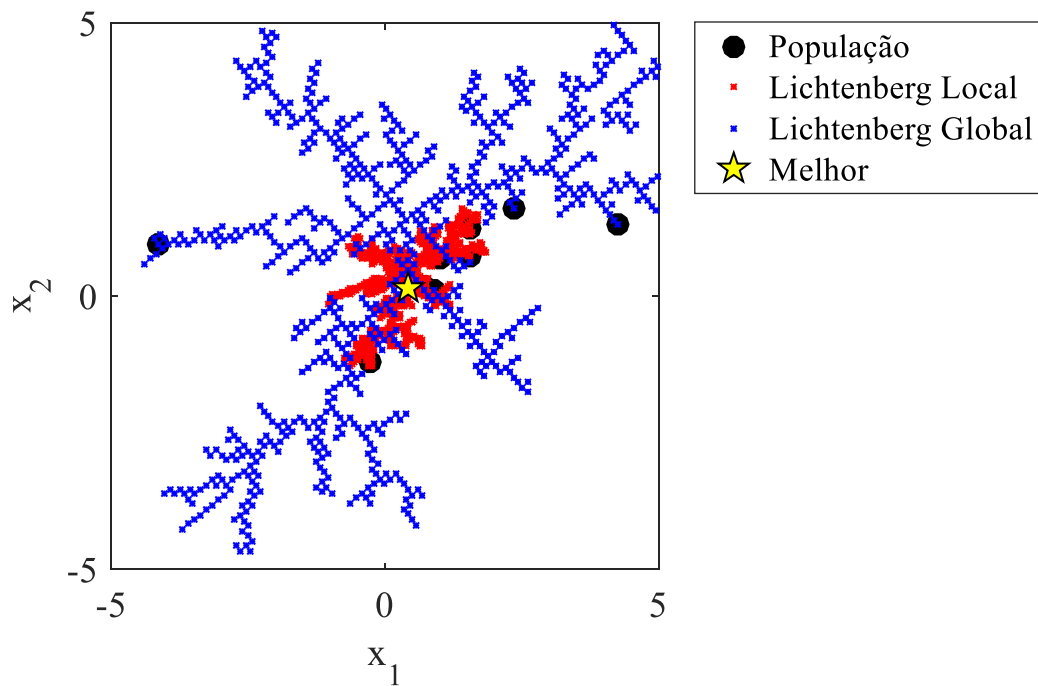


**Figura 3.15** – Figura Local com 30% do tamanho da Global e algumas Iterações

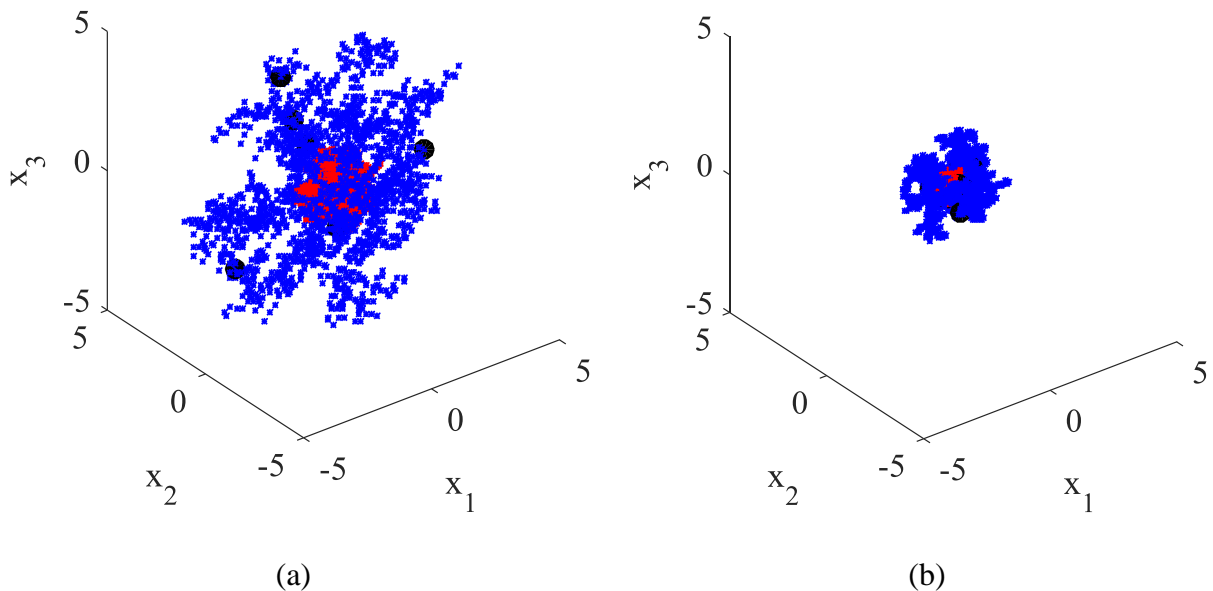
Não são todos os pontos da FL que são utilizados para o cálculo na função objetivo, o número de pontos utilizados para este fim ou população ( $pop$ ) é definido na entrada e geralmente é dez vezes o número de variáveis da função objetivo, podendo ser mais. Os pontos da FL que representarão a população são escolhidos ao longo de toda estrutura da FL (que é modificada a cada iteração) e são representados graficamente por pontos pretos, sempre todos estão dentro do espaço de busca por meio de uma checagem. Esta forma faz do LA um algoritmo híbrido já que funde dois tipos de algoritmos encontrados na literatura: os de população e os de trajetória.

Isto reforça muito a eficiência do otimizador, já que em alguns momentos, todos os pontos da iteração são obrigados a varrer uma região muito inferior ao espaço de busca inteiro. Ao passo que em outros momentos, a figura se torna maior e um espaço muito maior volta a ser varrido. Tendo-se ainda a FL secundária que é inferior por natureza à primária, a busca local fica ainda mais eficiente.

Vários critérios para a distribuição da população entre as figuras primárias e secundárias foram adotados, no entanto, geralmente a distribuição que mostra melhor resultado é 40% da  $pop$  total na FL Global e o restante na Local, como mostra a Figura 3.16 para 2D e a Figura 3.17 para 3D.



**Figura 3.16** – Distribuição da população nas Figuras de Lichtenberg ( $pop = 10$  e  $ref = 0.3$ )



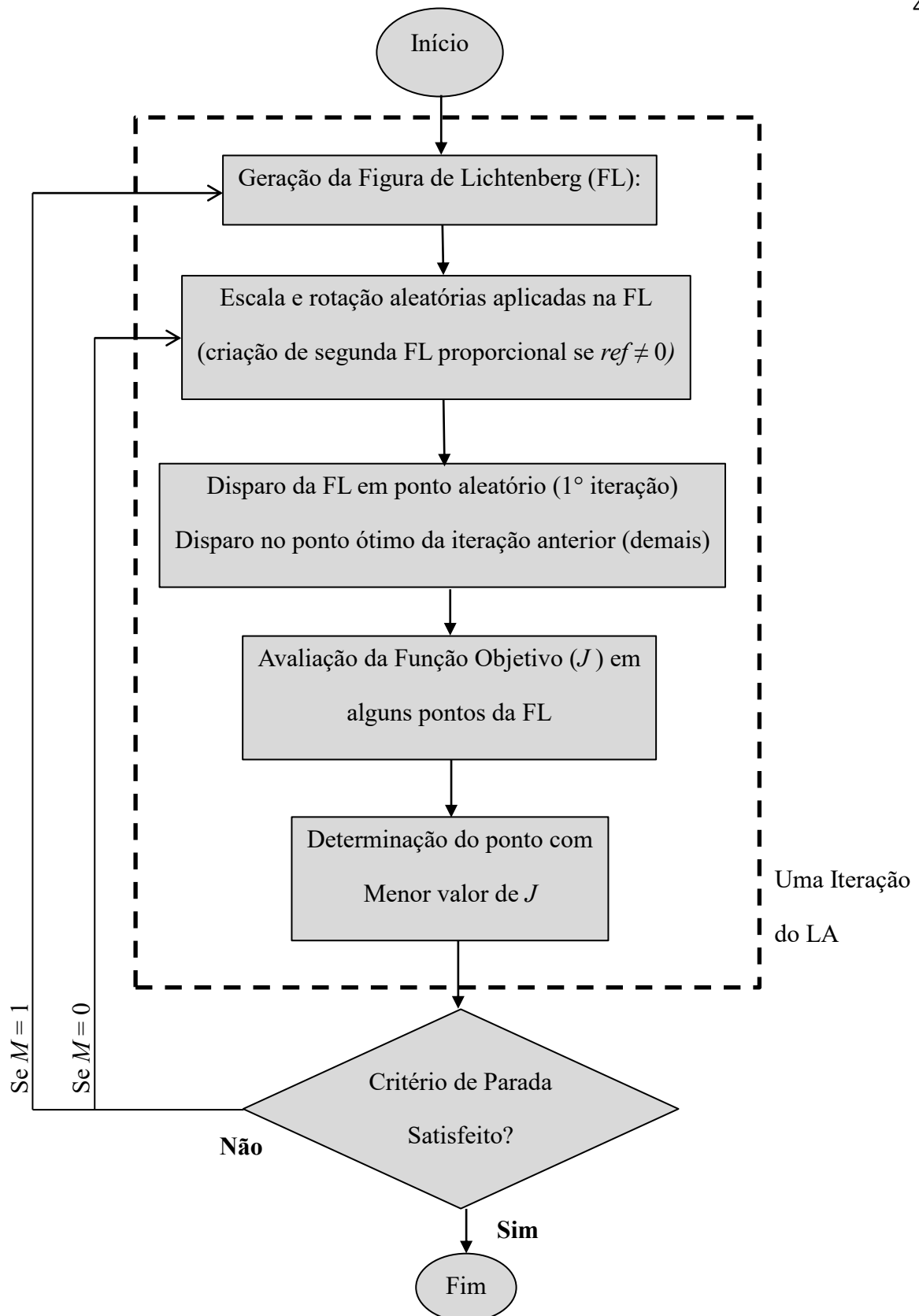
**Figura 3.17** – Distribuição da população no LA para três variáveis e dois exemplos de iterações ( $pop = 10$  e  $ref = 0.3$ )

A quarta medida tomada para não discretizar o espaço de busca é a inserção de um parâmetro  $M$  de troca de figura nos dados de entrada do otimizador. Este parâmetro pode valer zero ou um. Se zero, a mesma figura é usada em todas as iterações da execução e se for 1, uma

nova figura é gerada a cada iteração e ainda sim é multiplicada por um fator de escala randômico, rotacionada e recebe uma FL secundária e menor se desejável. Vale ressaltar que o procedimento que tem maior custo computacional do LA é a geração da FL, então adotar  $M = 1$  o torna exageradamente lento, apesar de poder promover uma convergência para o ótimo com menor número de iterações. Nos testes, não utilizar o parâmetro de troca de figuras não prejudicou os resultados.

Finalizando, o número de iterações ( $N_{iter}$ ) também é definido na entrada do LA, sendo geralmente igual a 100 iterações. A cada iteração o ponto inicial do disparo da FL é o ponto ótimo da iteração anterior e na primeira iteração, é um dos pontos da FL randomicamente escolhido. O poder dos fractais é utilizado com a própria estrutura da FL e ainda com as diferentes escalas que a FL pode ser plotada.

Este algoritmo é utilizado para a construção de figuras planas (2D) e espaciais (3D) com excelentes resultados, embora não haja uma figura *bitmap* para 3D. Como o algoritmo precisa de um espaço físico para ser construído, ao tratar de mais variáveis é feita uma projeção (ou espelhamento) destas figuras para  $n$  variáveis. A Figura 3.18 traz um fluxograma do LA, enquanto que a Tabela 3.2 traz o resumo e recomendações dos seus parâmetros.



**Figura 3.18** – Fluxograma do LA



Observação: Problemas específicos exigem parâmetros específicos. “ $d$ ” é o número de variáveis e o tamanho do espaço de busca, definidos pelo *Upper Bound* (UP) e o *Lower Bound* (LB) dependem do problema. As faixas de intervalos recomendadas abaixo surgiram das aplicações do algoritmo em muitas funções e problemas, onde o algoritmo gera péssimos resultados se fora do intervalo ou ainda, gera custo computacional que não altera significativamente os resultados. Ainda, em alguns casos, pode nem funcionar devido a restrições de memória do software utilizado. Não foi feito um estudo de otimização destes parâmetros.

**Tabela 3.2** - Parâmetros Recomendados do LA

Parâmetros	2D e mais que 3D	3D
$R_c$	50 a 200	50 a 80
$N_p$	$>10^3$ e $<10^6$	$>10^5$ e $<10^6$
$S$	0 a 1	0 a 1
$Pop$	$(10 \text{ a } 40) \times d$	$(10 \text{ a } 40) \times d$
$ref$	0 a 1	0 a 1
$M$	0 ou 1	0 ou 1
$N_{iter}$	$>10^2$ e $<10^3$	$>10^2$ e $<10^3$

Para organização no MATLAB<sup>®</sup>, os parâmetros de entrada, escolha das funções e dados modificáveis ficam no programa principal *MAIN*. Este chama o *LA\_optimization* que executa o otimizador selecionando pontos da estrutura, testando na função objetivo e a levando para o ótimo a cada iteração, além de checar os limites do espaço de busca. O *LA\_optimization* chama o *LA\_points* que é a transformação e centralização da FL de *bitmap* para coordenadas cartesianas. O *LA\_figure* cria a FL em termos matriciais (ou *bitmap*).

## Capítulo 4

# METODOLOGIA - APLICAÇÕES

O LA em sua primeira versão passará por várias rotinas de otimização. A primeira delas consiste na sua validação, a metaheurística enfrentará dez funções de teste usuais na literatura para este fim. Neste momento as funções podem ser unimodais e até mesmo multimodais, mas com restrições delimitando apenas o espaço de busca (lineares). Em seguida, o LA será aplicado no problema do Projeto de Viga Soldada, onde enfrentará restrições não lineares que devem ser obedecidas, a função é complexa e multimodal elevando o nível de dificuldade do algoritmo.

Por fim, o LA será aplicado num problema complexo de identificação de danos. Embora as restrições sejam lineares, as soluções não são simples e exigem bastante do otimizador. Neste contexto, o LA será testado para a identificação tendo-se níveis de ruídos nas leituras dos sinais de aquisição dos dados e ainda com intensidade menor dos danos dificultando seu diagnóstico.

### 4.1 Funções de Teste com Restrições Lineares

A Tabela 4.1 traz as funções utilizadas na validação do LA neste trabalho contendo a função utilizada, para quais números de variáveis, o intervalo que determina o espaço de busca em cada eixo e o valor da função objetivo avaliada nos pontos ótimos alvos desta função.

**Tabela 4.1** – Funções de Teste para Validação do LA

Função	Nº de Variáveis	Intervalo	$f_{min}$
$f_1 = \sum_{i=1}^n (x_i^2)$	1, 2, 3, 4, 5	[-100,100]	0
$f_2 = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	2, 3, 4, 5	[-10,10]	0
$f_3 = \sum_{i=1}^n  x_i  + \prod_{i=1}^n x_i$	2	[-10,10]	0
$f_4 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	2	[-10,10]	0
$f_5 = \sum_{i=1}^n ( x_i + 0.5 )^2$	2	[-10,10]	0
$f_6 = \sum_{i=1}^n ix_i^4 + random[0,1]$	2	[-1.28, 1.28]	0
$f_7 = 20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) = 20 + e$	2	[-32, 32]	0
$f_8 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	2	[-5.12, 5.12]	0
$f_9 = 4x_1^2 - 2.1x_1^4 + 0.3333x_1^2 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{10} = (x_2 - \frac{5,1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	[-5, 5]	0.398

Estas funções e outras podem ser encontradas em Mirjalili & Lewis (2016). As funções de  $f_1$  a  $f_6$  são unimodais e as demais multimodais. Estas funções foram rodadas 10 vezes para cada caso, com exceção das funções bidimensionais (duas variáveis) que foram simuladas 30 vezes. As funções mais testadas em outras variáveis foram as de Rosenbrock ( $f_1$ ) e a de Schwefel ( $f_2$ ). Os resultados estão no próximo capítulo e o programa do LA utilizado nestas funções está no Anexo A. Os resultados com as médias, desvio padrão e  $f_{min}$  dos pontos ótimos serão expostos no Capítulo 5.

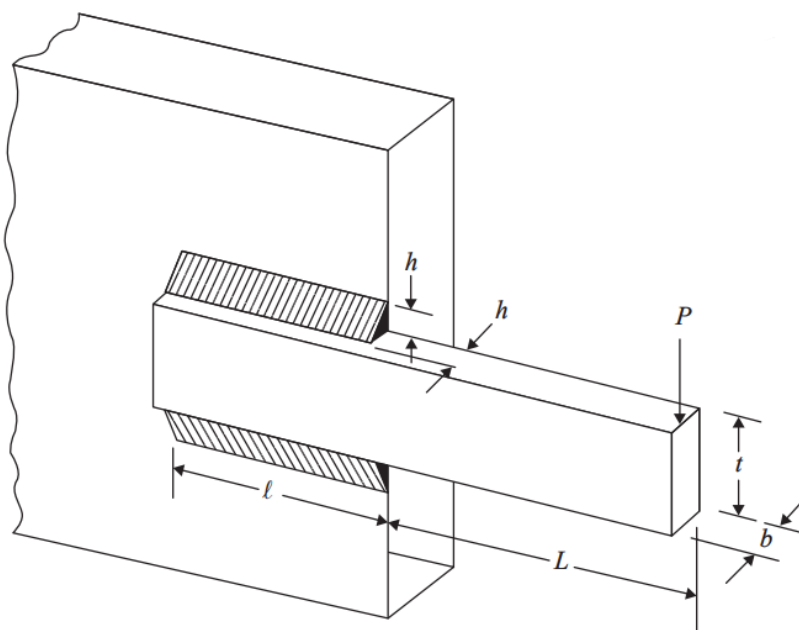
## 4.2 Problema Teste com Restrições não lineares

O Lichtenberg Algoritmo (LA) em sua primeira versão enfrentará um problema teste muito utilizado na literatura e debatido entre muitos autores. É o problema do Projeto De Viga Soldada. Neste caso, além dos limites do espaço de busca determinados pelo projeto, existem ainda algumas equações de igualdades e/ou desigualdades que devem ser checadas pelo algoritmo (restrições não lineares) de otimização antes da avaliação dos pontos  $x_i$  na função objetivo, afim de que sejam absolutamente respeitadas.

### 4.2.1 Projeto de Viga Soldada

O projeto de Viga Soldada envolve quatro variáveis: largura ( $x_1 = h$ ) e comprimento ( $x_2 = l$ ) da área soldada, a profundidade ( $x_3 = t$ ) e a espessura ( $x_4 = b$ ) da viga. O objetivo é minimizar o custo total de fabricação sob as restrições apropriadas de tensão de cisalhamento ( $\tau$ ), tensão de flexão ( $\sigma$ ), carga de flambagem ( $P$ ) e deflexão máxima ( $\delta$ ). O problema pode ser encontrado em Rao (2009, pp. 470).

O esboço do problema está na Figura 4.1 e o modelamento matemático logo a seguir. Os resultados encontrados pelo LA será posto à prova contra estes e uma pequena análise estatística será feita no Capítulo 5.



**Figura 4.1** – Variáveis do Projeto da Viga Soldada (Rao, 2009)

Para a solução, devemos minimizar a Equação (16):

$$\min f(x) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \quad (16)$$

Sujeito às seguintes desigualdades (17):

$$g_1(X) = \tau(X) - \tau_{m\acute{a}x} \leq 0 \quad (17.a)$$

$$g_2(X) = \sigma(X) - \sigma_{m\acute{a}x} \leq 0 \quad (17.b)$$

$$g_3(X) = x_1 - x_4 \leq 0 \quad (17.c)$$

$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5.0 \leq 0 \quad (17.d)$$

$$g_5(X) = 0.125 - x_1 \leq 0 \quad (17.e)$$

$$g_6(X) = \delta(X) - \delta_{m\acute{a}x} \leq 0 \quad (17.f)$$

$$g_7(X) = P - P_C(X) \leq 0 \quad (17.g)$$

E aos seguintes limites de espaço de busca (18):

$$0.1 \leq x_i \leq 2.0, i = 1, 4 \quad (18.a)$$

$$0.1 \leq x_i \leq 10.0, i = 2, 3 \quad (18.b)$$

Sendo (19):

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau\tau'\frac{x_2}{2R} + (\tau'')^2} \quad (19.a)$$

$$\tau'(X) = \frac{P}{x_1x_2\sqrt{2}} \quad (19.b)$$

$$\tau''(X) = \frac{MR}{J} M \quad (19.c)$$

$$M = P\left(L + \frac{X_2}{2}\right) \quad (19.d)$$

$$R = \sqrt{\frac{X_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (19.f)$$

$$J = 2 \left\{ \frac{x_1x_2}{\sqrt{2}} \left[ \frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\} \quad (19.g)$$

$$\sigma(X) = \frac{6PL}{X_4X_3^2} \quad (19.i)$$

$$\delta(X) = \frac{4PL^3}{EX_3^3X_4} \quad (19.j)$$

$$P_C = \frac{4.013}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right) \sqrt{EG \left( \frac{X_3^2X_4^6}{36} \right)} \quad (19.k)$$

Os parâmetros usados de entrada são: Carga ( $P$ ) = 6000lb; Comprimento Viga ( $L$ ) = 14 in; Constante Elástica Longitudinal ( $E$ ) =  $30 \times 10^6$  psi; Constante Elástica Torcional ( $G$ ) =  $12 \times 10^6$  psi; Tensão de Cisalhamento Máxima ( $\tau_{\max}$ ) = 30000 psi e Deflexão Máxima ( $\delta_{\max}$ ) = 0.25 in;

Após a otimização e obtenção das variáveis geométricas ótimas ( $h$ ,  $l$ ,  $t$ ,  $b$ ), pode-se obter os novos custos de viga e soldagem a partir da quantidade de material a ser utilizada para suportar as condições dadas e as restrições.

Os resultados de cada iteração do LA, a média, o desvio padrão das variáveis e do  $f_{\min}$ , e os valores calculados das restrições para checar sua validade encontram-se no Capítulo 5. Também, estes resultados serão comparados com os outros algoritmos encontrados na literatura. Haverá um estudo estatístico comparativo em relação ao  $f_{\min}$  de cada um dos algoritmos.

## 4.3 Metodologia na Identificação de danos

Nesta Seção é discutida a formulação numérica do problema de Identificação de Danos para a aplicação do Lichtenberg Algoritmo na identificação de delaminações. A Figura 4.5 no final deste capítulo resume a metodologia da identificação de danos utilizada neste trabalho.

### 4.3.1 Modelagem do Problema Direto

Sabe-se que o MEF neste trabalho será usado para calcular parâmetros modais (Os seis primeiros modos de vibração da placa em questão) de cada ponto utilizando um *software* CAE (*Computer Aided Engineering*). A geometria modelada é uma placa quadrada com espessura uniforme composta laminada com ou sem delaminação em regime linear elástico. Os deslocamentos de um ponto ( $x,y,z$ ) ao longo dos eixos  $x$ ,  $y$ ,  $z$  são  $u$ ,  $v$  e  $w$ , respectivamente. O elemento finito utilizado é o elemento de casca retangular de 8 nós. Para cada nó existem 6 graus de liberdade, rotações e translações dos eixos.

Assumindo um modelo com movimento harmônico e frequência angular  $\omega$  e usando o princípio de Lagrange, a equação do movimento para vibração livre (Equação 4) é reduzida a um problema padrão de autovalor como segue na Equação (20):

$$([K] - \omega^2[M])\{\delta\} = 0 \quad (20)$$

Sendo  $[M]$  é a matriz de massa global,  $[K]$  é a matriz rigidez,  $\omega^2$  a frequência natural elevada ao quadrado (autovalor),  $\{\delta\}$  o autovetor que define os modos de vibração e  $[M]$  a matriz de massa do sistema. Os parâmetros modais da placa são obtidos pelo MEF a partir da resolução desta equação.

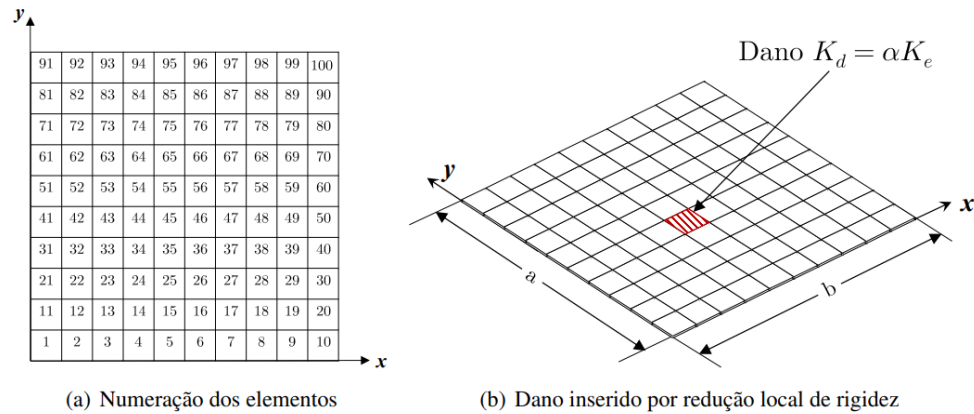
A malha adotada é estruturada, pois a estrutura em questão é uniforme, bidimensional e desta forma há economia computacional. A estrutura ainda é discretizada em 10 x 10 elementos. A condição de contorno é livre (FFFF – *Free- Free- Free- Free*) nos quatro lados da placa. As propriedades do material considerado no estudo são as contidas na Tabela 4.2

**Tabela 4.2** – Propriedades do Material  
(GOMES, 2019)

Propriedade	Valor
$E_1$	82.02 GPa
$E_2$	5.13 GPa
$G_{12}$	8.37 GPa
$\nu_{12}$	0.32
$\rho$	1408.1 kg/m <sup>3</sup>

A placa quadrada tem lado de 30 cm e é um laminado simétrico de material compósito constituído de 12 camadas de diferentes orientações dispostas da forma  $[0/90]_{3S}$ . O amortecimento não é considerado na modelagem numérica.

A redução de rigidez devido ao dano de delaminação é um parâmetro percentual que atua sobre a matriz de rigidez e conserva a sua massa. Segundo Santos *et al.* (2000), para uma estrutura discreta a perturbação da estrutura danificada é dada por  $\tilde{K} = K - \delta K$  em que a perturbação na matriz no elemento "e" é dada por  $\delta K_e = \alpha K_e$ , sendo  $\alpha \in [0, 1]$  um multiplicador de rigidez local que traduz a severidade do dano estrutural, quanto menor, maior o dano. O parâmetro de redução de rigidez local em termos percentuais é dado por  $\beta = (1 - \alpha) * 100$ . A Figura 4.2 traz um esboço do modelo desenvolvido no MEF e indica uma possível posição de dano.



**Figura 4.2** – Modelo utilizado no MEF e identificação dos elementos (GOMES, 2019)

### 4.3.2 Modelagem do Problema Inverso

Este trabalho utilizará como otimizador para o processamento inteligente dos sinais o Lichtenberg Algoritmo para minimizar a função objetivo (8) para a identificação de delaminações. A identificação tratará do diagnóstico de danos em placas com um e dois danos.

Ainda, outros cenários com ruídos e taxas de danos menos intensas serão utilizados para avaliar a capacidade do LA de diagnosticar o dano em condições onde a resposta dinâmica do sistema é menos afetada e onde a aquisição de sinais é mais confusa. A Tabela 4.3 resume os cenários em que o LA será testado. “ $d$ ” é o número de variáveis,  $\vec{x}$  são as variáveis que para maior parte dos cenários serão  $\vec{x} = (N_e, \alpha)$  com ou sem ruídos e no fim terá um cenário com dois danos e mesma taxa de dano,  $\vec{x} = (N_{e1}, N_{e2}, \alpha)$ . Para cada um dos cenários o método foi executado 10 vezes e os resultados estarão no Capítulo 5. A Figura 4.3 traz as localizações dos danos para estes cenários.

**Tabela 4.3** - Cenários de Identificação de danos propostos ao LA

Cenário	$d$	$\vec{x}$	Ruído (%)
I	2	(19, 0.2)	-
II	2	(19, 0.5)	-
III	2	(19, 0.9)	-
IV	2	(19, 0.2)	1
V	2	(19, 0.2)	2
VI	2	(19, 0.2)	5
VII	2	(19, 0.2)	10
VIII	3	(19, 65, 0.2)	-



91	92	93	94	95	96	97	98	99	
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

(a) Um dano

91	92	93	94	95	96	97	98	99	
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

(b) Dois danos

**Figura 4.3** – Localização dos Danos a serem encontrados pelo LA

Os Cenários I, IV, VI e VII serão comparados com os resultados obtidos por Gomes (2017) utilizando o AG e o SFO. Todas as simulações foram executadas pelo software MATLAB<sup>®</sup> onde contém o código do LA. Este aciona internamente e automaticamente o ANSYS<sup>®</sup> para análise numérica. As respostas dos seis primeiros modos de vibração com o dano (única) é dada pelo CAE ao otimizador que desconhece a localização e a taxa de dano (modo de vibração real,  $\Phi_{i,s}^{real}$ ). A localização e taxa deste dano é conhecida pelo operador e dada como entrada no programa, mas pode ser adquirida de uma estrutura real.

A partir disto, o otimizador avalia estocasticamente cada um dos pontos da população de cada iteração numericamente e encontra os modos de vibração calculados ( $\Phi_{i,s}^{calculado}$ ) que são comparados com os reais na função objetivo (8) para os seis primeiros modos de vibração. Quanto mais próximos, menor a função objetivo e por isto este é um problema de minimização.

### 4.3.3 Modelagem de ruídos

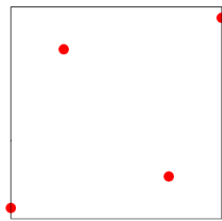
Na prática é impossível adquirir sinais sem ruídos. O desempenho da predição de resposta e identificação do dano sem ruído com certeza será melhor do que com ruído. Para Ganguli (2001), no caso de frequências naturais, dado um delta de medição de frequência  $\Delta\phi$ , um número aleatório  $u$  no intervalo  $[-1, 1]$  e um parâmetro de nível de ruído  $a$ , para todos os modos de vibração a inclusão de incertezas é dada pelo delta de medição ruidoso:

$$\Delta\phi_{ruído} = \Delta\phi(1 + ua) \quad (21)$$

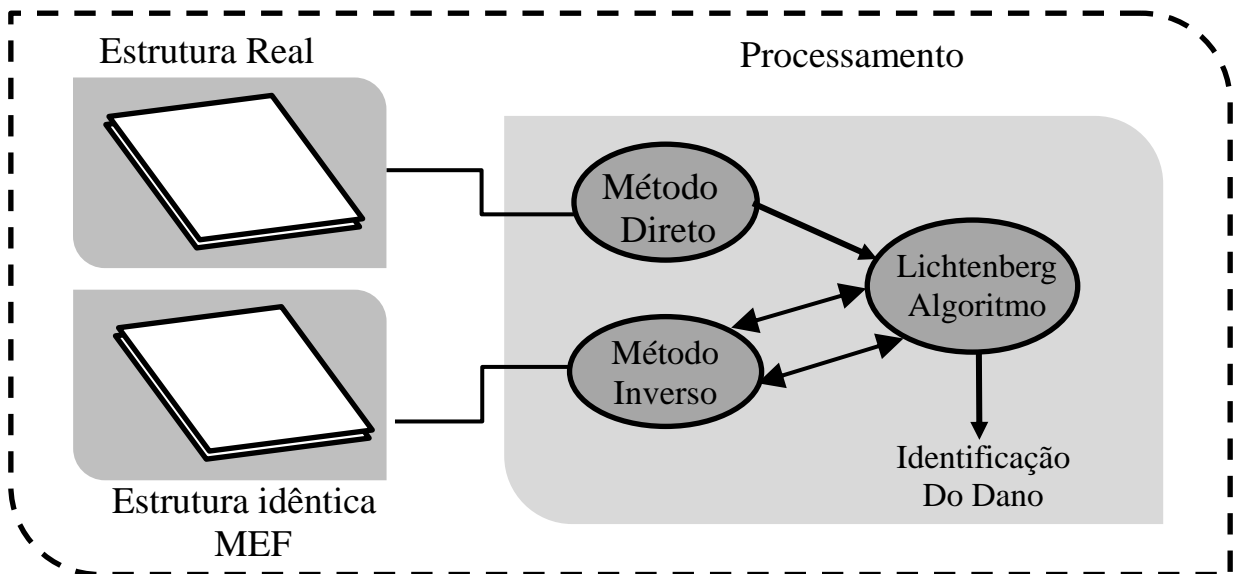
Sendo  $u$  um número de distribuição normal conhecido como ruído Gaussiano branco  $N(0,1)$ . Pode ser considerado como um rand. Neste trabalho serão feitas simulações com ruídos de 1, 2, 5 e 10%.

### 4.3.4 Posição dos Sensores

Os sensores são um canal entre a estrutura (ou MEF) e o algoritmo de otimização para processamento inteligente dos dados e tem grande importância. Dentro das soluções encontradas por Gomes *et al.*(2018.c) a solução ótima possível é a utilização de quatro sensores nas posições de nós 41, 80, 144 e 278 em uma placa quadrada de 100 elementos (10x10). A Figura 4.4 mostra esta solução ótima. Destes quatro sensores, os seis primeiros modos de vibração são calculados para cada caso e são utilizados na identificação dos danos mimizando a Equação (8).



**Figura 4.4** – Localização dos sensores (GOMES, 2017)



**Figura 4.5** – Resumo da Metodologia da Identificação de Danos

## Capítulo 5

# RESULTADOS E DISCUSSÕES

Após a explanação de todos os problemas de otimização que a primeira versão do LA deve enfrentar, neste capítulo serão apresentados os resultados e as discussões do desempenho da metaheurística. Os problemas enfrentados pelo LA são: Funções de Teste sem Restrições, Otimização de Projeto Mecânico com restrições não lineares – Projeto da Viga Soldada e finalmente e mais importante, identificação de danos do tipo delaminação.

### 5.1 Desempenho nas Funções de Teste

As funções de teste são a base de validação de qualquer otimizador. Os parâmetros para todas as simulações foram os mesmos para o mesmo número de variáveis. Número de variáveis diferentes e funções diferentes podem pedir parâmetros específicos para obterem-se bons resultados.

Os parâmetros foram obtidos a partir de tentativas. Os resultados das funções de teste estão no Apêndice B, com exceção daqueles em que foram executados 30 vezes (todas funções exceto  $f_1$  e  $f_2$ , pois o programa foi automatizado para entregar o resultado final). Aqui constarão

apenas os resultados resumidos: parâmetros usados, a média e o desvio padrão dos resultados a partir do número de simulações de execuções do programa.

### 5.1.1 Duas Variáveis

As funções escolhidas para validação do LA para duas variáveis foram todas as dez Funções da Tabela 4.1 e foram simulados 30 vezes cada. O tempo médio das simulações foi de 35 segundos. Os parâmetros estão na Tabela 5.1, o os resultados estão na Tabela 5.2.

**Tabela 5.1** – Parâmetros do LA para otimização de funções bidimensionais

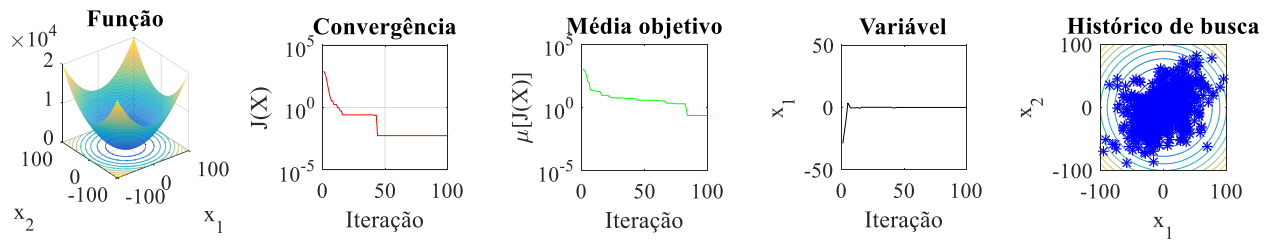
$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$
20	100	100000	1	150	0.2	0
20	100	100000	1	150	0.2	0

**Tabela 5.2** – Resultados de otimização de funções de teste bidimensionais

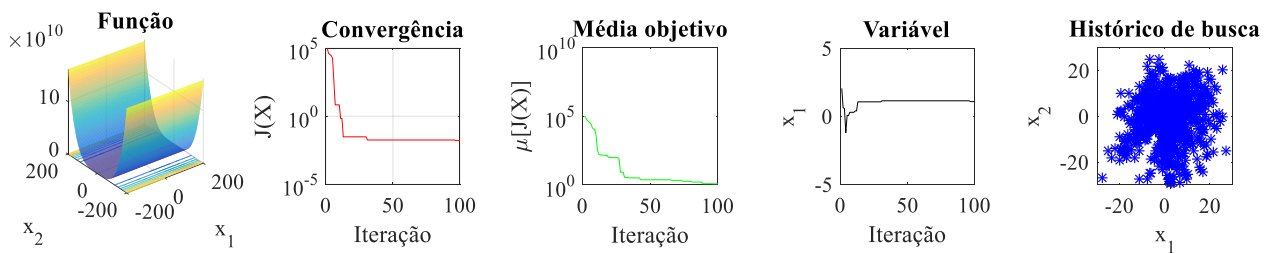
Função	$x_1$		$x_2$		$f_{min}$		Alvo
	Média	Desvio	Média	Desvio	Média	Desvio	
$f_1$	-0,0015	0.0277	0.0117	0.0457	<b>0.0029</b>	0.0044	<b>0</b>
$f_2$	1.0327	0.2090	1.1091	0.6400	<b>0.0454</b>	0.2267	<b>0</b>
$f_3$	-0,0001	0.0032	0,0001	0.0037	<b>0.0047</b>	0.0038	<b>0</b>
$f_4$	-0.0020	0.0033	0.0024	0.0049	<b>0.0000</b>	0.0000	<b>0</b>
$f_5$	-0.5001	0.0032	0.5000	0.0046	<b>0.0000</b>	0.0000	<b>0</b>
$f_6$	0.0109	0.1348	-0.0216	0.1022	<b>0.0020</b>	0.0016	<b>0</b>
$f_7$	-0.0018	0.0084	0.0024	0.0113	<b>0.0377</b>	0.0327	<b>0</b>
$f_8$	0.0003	0.5537	0.0028	0.2519	<b>0.0016</b>	0.4850	<b>0</b>
$f_9$	-0.01782	0.0891	0.1416	0.7097	<b>-1.0318</b>	1.0897	<b>-1.0316</b>
$f_{10}$	3.1418	0.0011	2.2742	0.0028	<b>0.3979</b>	0.0000	<b>0.398</b>

Pode-se observar que o LA teve resultados muito próximos a seus alvos, mostrando-se eficaz na sua validação para otimização bidimensional, tendo-se resultados muito bons para as funções unimodais e ligeira dificuldade para as multimodais. A Tabela 5.3 traz algumas figuras importantes para a análise do desempenho do LA. Da esquerda pra direita, a primeira figura traz o gráfico da função minimizada; a segunda traz a convergência de  $f_{min}$  ao longo das iterações; a terceira mostra a convergência, mas tendo-se a média de todos os pontos de cada iteração; a quarta mostra a convergência da primeira variável e por último, o histórico de busca ao longo de todas as iterações.

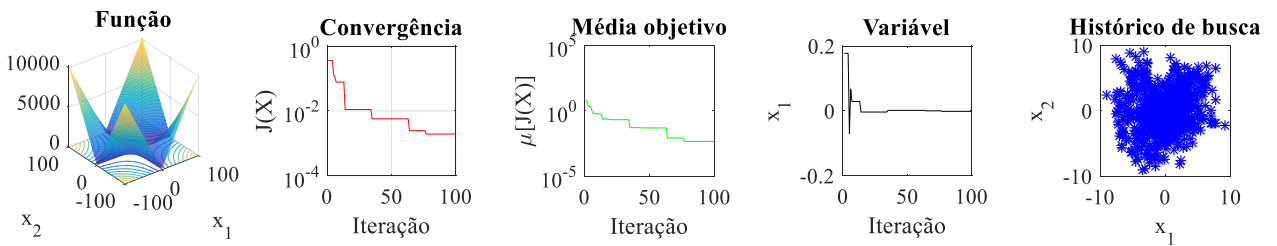
**Tabela 5.3** – Análise dos resultados de busca para funções bidimensionais



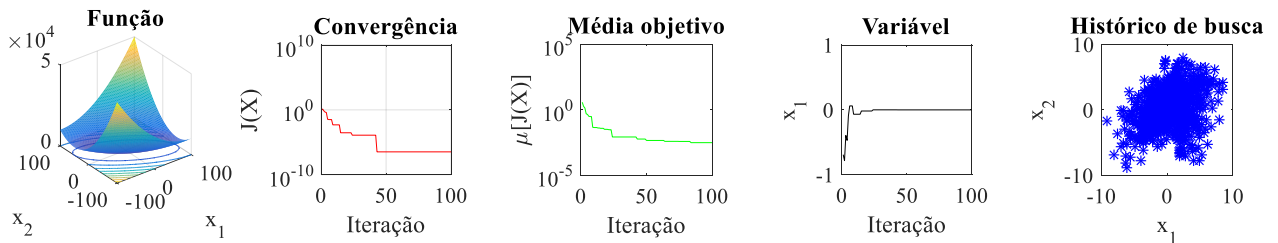
(a) função  $f_1$



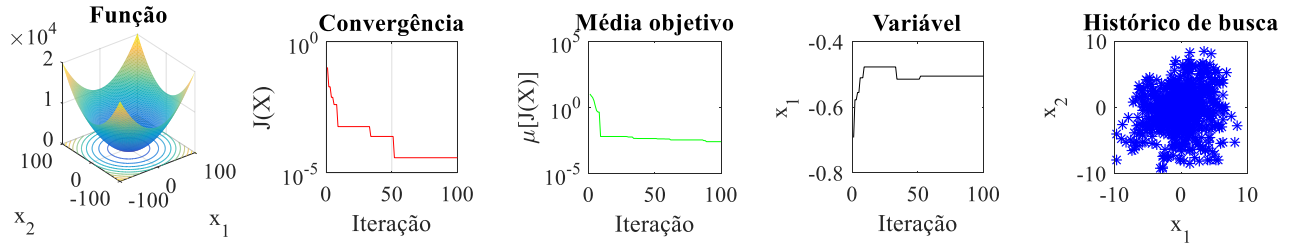
(b) função  $f_2$



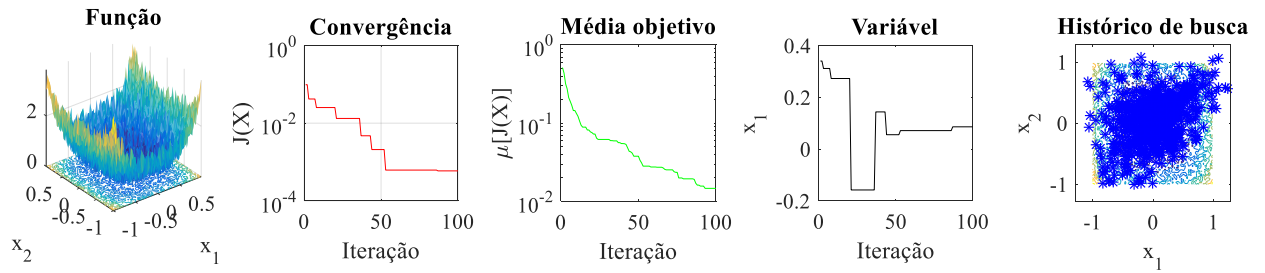
(c) função  $f_3$



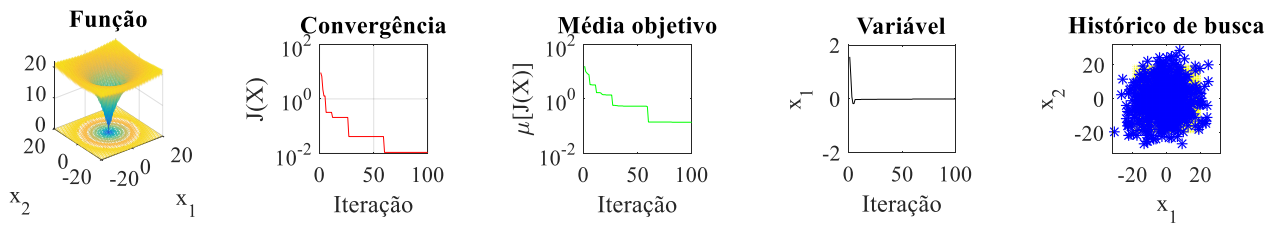
(d) função  $f_4$



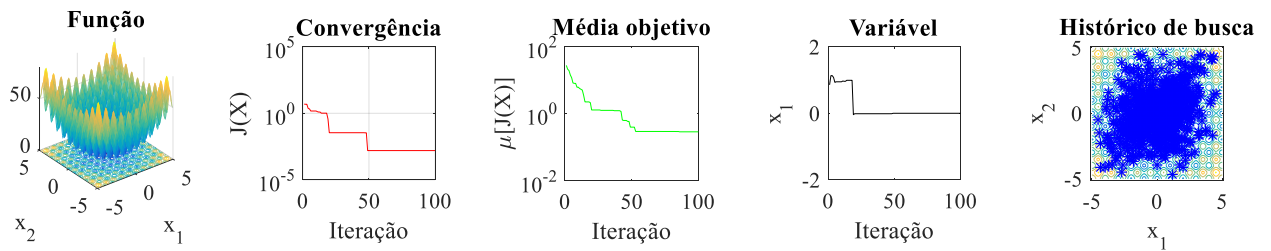
(e) função  $f_5$



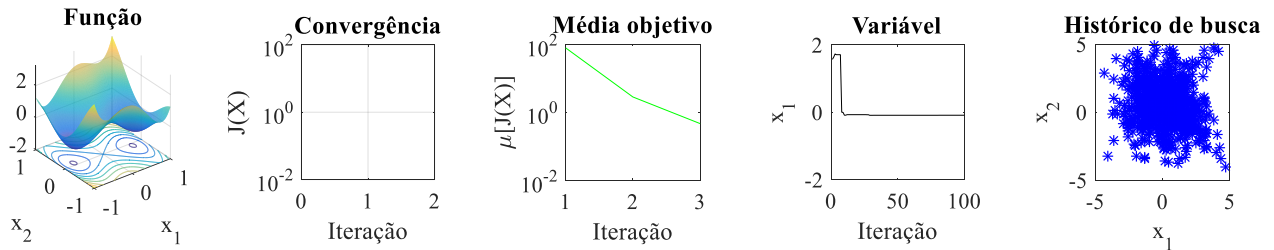
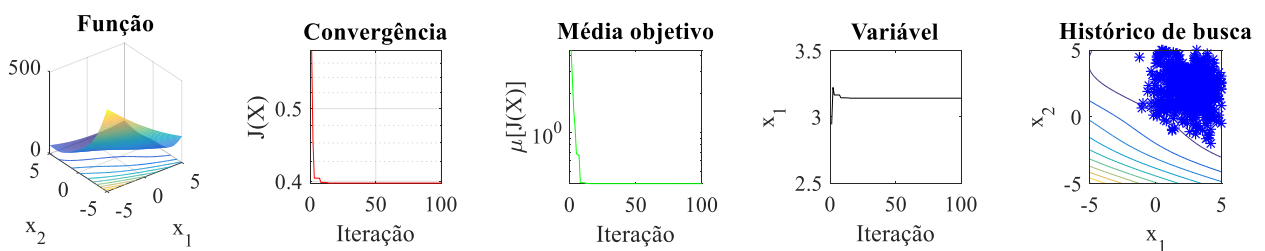
(f) função  $f_6$



(g) função  $f_7$



(h) função  $f_8$

(i) função  $f_9$ (j) função  $f_{10}$ 

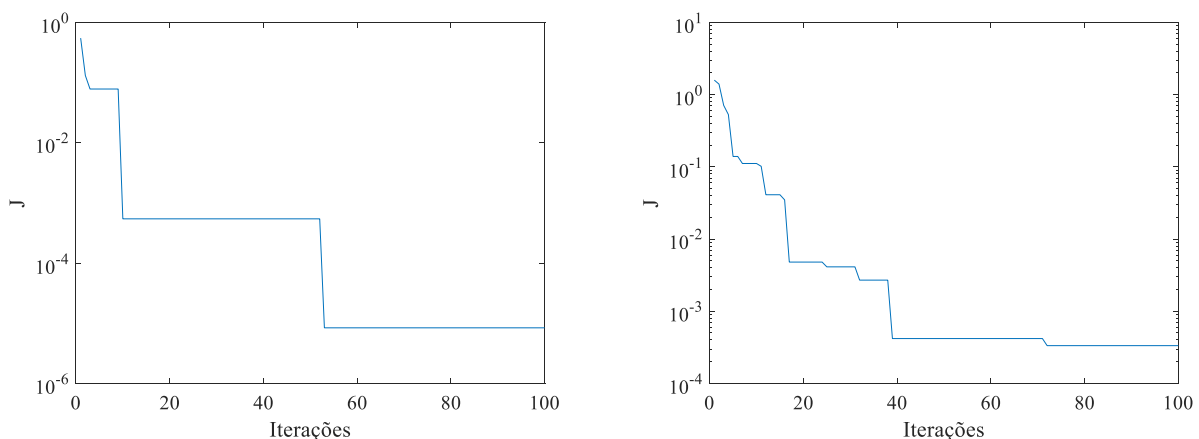
Pode-se observar uma excelente exploração do espaço de busca como um todo para todas as otimizações nas funções de teste, vistas nas figuras do Histórico de Busca. Isto mostra um bom potencial do otimizador para não se isolar em mínimos locais. Ainda, dada à propriedade de ter-se figuras extremamente pequenas em algumas iterações, pode-se observar um poder de convergência bom para o algoritmo para a função objetivo e as variáveis, mesmo em funções graficamente mais complicadas.

A otimização de uma função objetivo de duas variáveis é a que traz a melhor oportunidade para visualizar-se a Figura de Lichtenberg e analisar seu comportamento, já que esta e para três variáveis são as únicas que têm imagens dimensionalmente visualizáveis. Ainda, para qualquer número de variáveis menor e maior que três (exceto ele) é uma FL 2D que é gerada e rotacionalmente expandida para as demais dimensões e portanto, variáveis. Logo, as características são praticamente as mesmas de comportamento para qualquer número de variáveis, exceto para três.

O tempo de simulação é muito influenciável por diversos mecanismos: capacidade de processamento do computador, parâmetros do otimizador, solicitar apresentação na tela das figuras e até mesmo o modo estocástico em que a FL está sendo gerada. Como já dito, o processo com maior custo computacional do LA é a geração da FL, se o parâmetro  $M = 0$ , a

mesma figura é usada em todas as iterações e a execução é mais rápida. No entanto, se  $M = 1$ , uma figura é gerada a cada iteração e o tempo de simulação se estende demasiadamente. A fim de comparação dos tempos, utilizando-se os mesmos parâmetros da Tabela 5.1 (com  $M = 0$ ) e computador, as simulações para duas variáveis que não se plotam as figuras tem média de 35 segundos. Se a figura é solicitada para apresentação na tela, mesmo sendo a mesma FL para cada iteração e não tendo pausa entre as figuras plotadas ( $M = 0$ ), as simulações se estendem para uma média de 43 segundos.

O parâmetro  $M$  não altera significativamente os resultados das simulações, como pode ser visto na Figura 5.1, onde são comparadas duas execuções para a mesma função, parâmetros, computador, etc. Neste caso, não gerar novas FL's culminou em resultados até melhores, tendo-se atingido um valor de função objetivo de  $10^{-3}$  em um menor número de iterações. Por isto nas aplicações deste trabalho manteve-se  $M=0$  e ainda ganha-se economia de tempo devido ao menor custo computacional.



(a) Mesma figura em todas as Iterações ( $M=0$ )      (b) Figuras diferentes a cada iteração ( $M=1$ )

**Figura 5.1** – Gráficos de Convergência da função objetivo avaliando o desempenho do LA para geração ou não de FL diferentes a cada iteração

Os outros parâmetros influenciam de maneira menos significativa nos resultados do LA, mas para ter-se uma otimização precisa é necessário buscar-se a melhor combinação destes parâmetros. O coeficiente de aderência ( $S$ ) pode aumentar demais a densidade do aglomerado, o que pode deixar pobre a distribuição de pontos no espaço de busca pela seleção randômica de pontos da FL para avaliação na função objetivo (muitos pontos próximos na seleção ao acaso, podem ser próximos), o que pode prejudicar a busca global. No entanto, ele pode melhorar a busca local significativamente. O tipo de função objetivo a ser otimizada pode requerer valores



de  $S$  muito diferentes mesmo tendo-se o mesmo número de variáveis. O refinamento (*ref*) funciona de maneira análoga quando atua. Se muito pequeno melhora a busca local e se muito grande, melhora a global. Funções multimodais podem pedir *ref* e  $S$  mais elevados para se evitar mínimos locais, a compensação da busca local vem quando a FL é plotada pequena devido ao fator de escala randômico.

$S$ ,  $R_c$  e  $N_p$  trabalham coletivamente para determinar o número de pontos da FL. Um  $S$  muito baixo faz terem-se muitas partículas próximas e o  $R_c$  pode nem ser atingido. Muitos pontos densos atrapalham o desempenho do algoritmo. Por outro lado, um  $R_c$  elevado com “partículas à vontade” e um  $S$  alto formam as FL com maior alcance e menos pontos muito próximos, o que na hora de seleção randômica de pontos na estrutura pode ser bem vantajoso. Os pontos podem ser selecionados mais distantes, o que fomenta a busca global e na local é compensada pela escala randômica. Porém, em funções multimodais complexas o resultado pode ser não ótimo. Determinar os melhores valores destes parâmetros é uma tarefa que depende de cada função objetivo e número de variáveis. Porém, devido à forma de atuação do algoritmo e como já dito, grandes variações destes três parâmetros influenciam pouco em termos percentuais nos ótimos encontrados.

Por último, o número de iterações ( $N_{iter}$ ) e a População (*pop*) possuem relação quase que direta: quanto maiores, maior a precisão e a busca do ponto ótimo pelo algoritmo. No entanto, grande população e grandes iterações faz com que as metaheurísticas percam a sua proposta e denigram a questão da eficiência. Um dos principais índices de desempenho de metaheurísticas é a medida de sua precisão de encontro do ponto ótimo em menos tempo ou iterações, com uma população e número de iterações menor, por isto se inspiram em comportamentos ótimos da natureza. Caso contrário, uma simples randomização de muitos pontos e iterações ao acaso no espaço de busca já seria o suficiente para localização do ponto ótimo.

### 5.1.2 Três Variáveis

A otimização de três variáveis é um dos modelos que o LA cria uma FL visualmente existente. As funções escolhidas foram as  $f_1$  e  $f_2$  e foram simuladas 10 vezes cada. O tempo médio de simulação é de 1050 segundos para a configuração de parâmetros utilizada, o mais altos das “n” variáveis devido à espera da criação de uma FL em 3D utilizando o DLA. O tempo pode se tornar muito menor se diminuir o  $R_c$  e o  $N_p$ , assim como a precisão nos resultados. Os parâmetros de entrada estão na Tabela 5.4 e o resultado resumido na Tabela 5.5.

**Tabela 5.4** – Parâmetros do LA para funções tridimensionais

Função	Ótimo Global	$Pop$	$(N_{iter})$	$N_p$	$S$	$R_c$	$ref$	$M$
$f_1$	(0, 0, 0)	30	100	1000000	1	150	0.4	0
$f_2$	(1, 1, 1)	30	100	1000000	1	150	0.4	0

**Tabela 5.5** - Resultados das funções tridimensionais

Função		Ótimos Calculados	$f_{min}$
$f_1$	Média	(0.0010, 0.0004, 0.0003)	0
	Desvio Padrão	(0.0071, 0.0061, 0.0146)	0.0009
$f_2$	Média	(0.9952, 0.9898, 0.9796)	0.0039
	Desvio Padrão	(0.0170, 0.0315, 0.0609)	0.0176

Criar a FL em 2D já tinha um custo computacional, em 3D se torna maior. Isto ocorre devido ao modelo de crescimento do DLA em forma matricial ou *bitmap*. Ainda, o MATLAB<sup>®</sup> possui uma limitação de memória que não permite a criação de aglomerados grandes nesta condição. Isto significa que o número de pontos num mesmo eixo fica cada vez menor quanto maior o número de dimensões matriciais que o *software* tem que criar. Isto prejudica o desempenho do LA em 3D se comparado com o 2D e esta diferença é mais nítida quanto mais complexa for a função a ser otimizada.

Nesta parte do desenvolvimento do algoritmo que se percebeu que se tivesse modelos matriciais para confecção da FL para  $n$  variáveis, a qualidade da otimização decairia rapidamente com o aumento do número de variáveis, além do aumento significativo dos tempos de simulações. Por isso se optou pela propagação da figura em 2D nos demais planos para  $n$  variáveis para manter a eficácia do LA. O fator que mais influencia o tempo de simulação aqui é o  $R_c$ : se maior que 80 a criação da FL é mais lenta ainda.

### 5.1.3 Demais Variáveis

Para as demais variáveis foram-se utilizadas novamente as funções de Rosenbrock e a de Schwefel que foram simuladas dez vezes. Sabe-se que para ambas as funções o  $f_{min}$  alvo é nulo, sendo os ótimos globais os pontos (0, ..., 0) para a primeira e (1, ..., 1) para a segunda tendo a quantidade de pontos a depender do número  $d$  de variáveis da função. O tempo médio das simulações para uma variável foi de 38 segundos, de quatro variáveis 72 segundos e para cinco variáveis foi de 67 segundos. Os parâmetros utilizados estão na Tabela 5.6 e o resultado resumido na Tabela 5.7.

**Tabela 5.6** – Parâmetros do LA para demais variáveis

$Pop$	$N_{iter}$	$Np$	$S$	$Rc$	$ref$	$M$
$10*d$	100	100000	1	150	0.2	0
$10*d$	100	100000	1	150	0.2	0

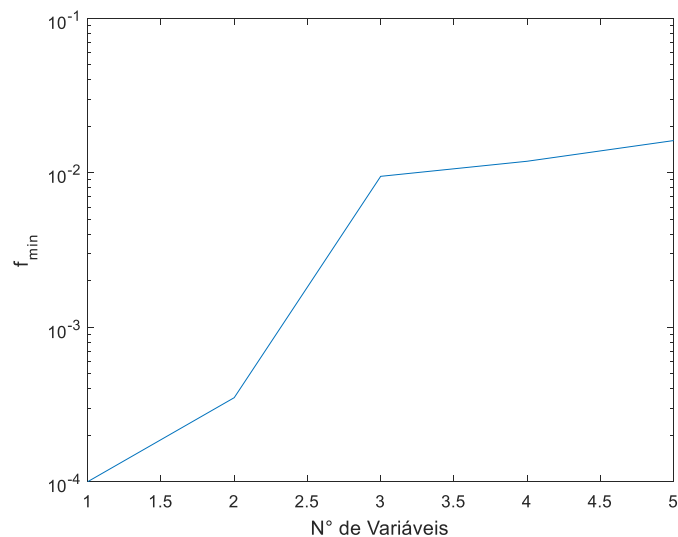
**Tabela 5.7** - Resultados da otimização para demais variáveis

Função		Ótimo Calculado	$f_{min}$
$f_1(d=1)$	Média	0.0001	0.0002
	Desvio Padrão	0.2649	0,0003
$f_2(d=1)$	Média	1.0000	0
	Desvio Padrão	0.0002	0.0003
$f_1(d=4)$	Média	(0.0089, 0.0240, 0.0022, -0.0509)	0.0128
	Desvio Padrão	(0.0657, 0.0932, 0.0777, 0.0801)	0.0233
$f_2(d=4)$	Média	(0.9888, 0.9828, 0.9685, 1.0133)	0.0110
	Desvio Padrão	(0.0270, 0.0525, 0.1025, 0.1839)	0.2683
$f_1(d=5)$	Média	(-0.0022, -0.0010, 0.0030, 0.0018, 0.0022)	0.0013
	Desvio Padrão	(0.0206, 0.0244, 0.0170, 0.0279, 0.0240)	0.0030
$f_2(d=5)$	Média	(1.031, 1.0027, 1.0030, 1.0116, 1.0225)	0.0311
	Desvio Padrão	(0.0165, 0.0335, 0.0651, 0.1231, 0.2226)	0.0425

Para o número de variáveis ( $d$ ) acima de quatro há a espelhamento rotacionado da primeira figura (2D), neste caso é como se fossem  $d/2$  figuras de FL buscando o ótimo. Se  $d$  é ímpar, uma linha é cortada de parte de uma das FL 2D.

### 5.1.4 Comparação dos Menores Valores da Função Objetivo

Finalizadas as simulações de validação para as 8 primeiras funções para as 5 variáveis em que o menor valor da função objetivo deveria ser nulo, foi feito um gráfico para avaliar como a média dos melhores valores calculados evoluiu ao longo do aumento do número de variáveis, representado na Figura 5.2. Como cada avaliação para um número de variável tinha ao menos duas funções, o  $f_{min}$  considerado foi a média destes resultados, apenas para título de comparação e análise de divergência do menor valor da função objetivo encontrado com o aumento do número de variáveis.



**Figura 5.2** – Influência negativa do aumento do número de variáveis na precisão do LA

Como se pode observar pelo gráfico, o  $f_{min}$  se afasta do valor de referência na medida em que o número de variáveis ( $d$ ) aumenta. Sendo o maior valor atingido de 0,0162 para cinco variáveis, mostrando ligeira perda de eficácia quanto maior o número de dimensões para a otimização.

## 5.2 Desempenho no Problema Teste

O Projeto de Viga Soldada é o primeiro problema projeto de otimização com restrições aplicadas em Engenharia Mecânica que o LA enfrenta em sua primeira versão e a solução obtida será comparada com outras da literatura. Para manter-se um padrão de comparação, alguns

critérios foram adotados padronizados: Número de Iterações igual a 200; população de 100; 30 execuções do programa para análise de média e desvio padrão, mesmo espaço de busca e os demais parâmetros sujeitos às especificidades de cada otimizador. Os parâmetros do LA estão na Tabela 5.8.

**Tabela 5.8** - Parâmetros utilizados no LA para o Projeto de Viga Soldada

Função	Variáveis	$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$
<i>Projeto da Viga Soldada</i>	$(h, l, t, b)$	100	200	100000	1	150	0.2	0

O problema consiste em otimizar as variáveis  $h$ ,  $l$ ,  $t$  e  $b$  através da minimização da Equação (16), sujeita às restrições (17) e ao espaço de busca (18) para se ter o menor uso de materiais (e custo) que suporte os esforços e tensões solicitadas. O tempo médio das simulações foi de 72s. O resultado resumido com as variáveis ótimas encontradas em polegadas (*in*) está na Tabela 5.9 e o geral está no Apêndice C.

**Tabela 5.9** - Variáveis ótimas de Projeto (LA)

	$h$	$l$	$t$	$b$	$f_{min}$
Média	0.2213	3.2818	8.7579	0.2216	1.8446
Desvio Padrão	0.0603	0.9986	0.7591	0.0487	0.1597

Diversos pesquisadores analisaram este problema: Deb (1991), Coello (2000), Coello & Montes (2002) aplicaram AG; He & Wang (2007) utilizaram uma versão do Enxame de Partículas, o CPSO; Kaveh & Talatahari (2010) utilizaram o ACO e Xin-She Yang (2010) utilizou uma versão acelerada do PSO, o APSO e Salimi (2014) executou este problema com seu SFS, entre outros. A Tabela 5.10 traz os resultados obtidos por estes autores juntamente com o LA, inclusive os valores das restrições que devem ser menor ou igual a zero.

**Tabela 5.10** - Comparação dos Resultados do Projeto de Viga Soldada entre diferentes algoritmos

Saídas	Deb (1991)	Coello (2000)	Montes (2002)	He (2007)	Kaveh (2010)	Xin (2010)	Salimi (2014)	LA
$h$	0.2489	0.2088	0.2060	0.2024	0.2057	0,1942	0.2057	<b>0.2213</b>
$l$	6.1730	3.4205	3.4713	3.5442	3.4711	3,7409	3.4705	<b>3.2818</b>
$t$	8.1789	8.9975	9.0202	9.0482	9.0367	9,0401	9.0366	<b>8.7579</b>
$b$	0.2533	0.2100	0.2065	0.2057	0.2057	0,2056	0.2057	<b>0.2216</b>
$g_1$	-0.0068	-0.3378	-0.1030	-13.6555	-0.0846	-	0.0000	<b>-60.8008</b>
$g_2$	-255.5769	-353.9026	-0.2317	-75.8141	-0.5907	-	0.0000	<b>-347.5340</b>
$g_3$	-0.0044	-0.0012	0.0005	-0.0034	0.0000	-	0.0000	<b>-0.0003</b>
$g_4$	-2.9829	-3.4119	-3.4300	-3.4246	-3.4329	-	-3.4330	<b>-3.3813</b>
$g_5$	-0.1239	-0.0838	-0.0810	-0.0774	-0.0807	-	-0.0807	<b>-0.0963</b>
$g_6$	-0.2342	-0.2356	-0.2355	-0.2356	-0.2355	-	-0.2355	<b>-0.2353</b>
$g_7$	-0.0045	-363.2324	-58.646	-4.4729	-0.1448	-	0.0000	<b>-1343.922</b>
$f_{min}$	2.4332	1.7483	1.7282	1.7280	1.7280	1,7514	1.7249	<b>1.8446</b>

Para ficar mais clara a comparação, a Tabela 5.11 traz as diferenças percentuais entre os valores encontrados pelo LA e a média dos demais para cada uma das variáveis de projeto.

**Tabela 5.11** - Relação do LA com a média dos outros resultados

Variáveis	$h$	$l$	$t$	$b$	$f_{min}$
Média	0.2059	3.4712	9.0284	0.2061	1.7514
LA	0.2213	3.2818	8.7579	0.2216	1.8446
<b>Diferença (%)</b>	<b>7.5015</b>	<b>5.4571</b>	<b>2.9964</b>	<b>7.5178</b>	<b>5.3221</b>

A Tabela 5.12 traz a média, o desvio padrão, o melhor e o pior valor dos  $f_{min}$  do LA e dos demais algoritmos. Note que o LA chegou a apresentar um valor de  $f_{min}$  muito baixo em pelo menos uma das suas iterações.

**Tabela 5.12** - Comparação Estatística dos menores valores de função objetivo do Projeto de Viga Soldada

Alvo	Deb	Coello	Montes	He	Kaveh	SFS	APSO	LA
Melhor	2.4331	1.7483	1.7282	1.7280	1.7249	1.7249	1.7465	<b>1.7351</b>
Média	NA	1.7720	1.7927	1.7488	1.7298	1.7249	1.7514	<b>1.8446</b>
Pior	NA	1.7858	1.9934	1.7821	1.7760	1.7249	2.2128	<b>2.1480</b>
Desvio	NA	0.0112	0.0747	0.0129	0.0092	0.0000	0.1171	<b>0.1697</b>

Como visto, os valores encontrados pelo LA não se distanciam dos encontrados por otimizadores muito bem consolidados e melhorados, como AG, APSO, etc. Inclusive estando melhor que o encontrado por Deb em uma das primeiras versões do AG (menor  $f_{min}$ ). A diferença de resultado entre o LA e os demais no mundo real seria menos significativa, já que a precisão destas medidas em medições em campo ou sujeitas a variações de temperatura não supririam as diferenças entre as soluções ótimas encontradas por estes algoritmos.

## 5.3 Desempenho na Identificação da Delaminação

O grande propósito do desenvolvimento do LA e do tipo de aplicação neste trabalho é a identificação de delaminações. Respeitadas as condições expostas no Capítulo 4 e os oito cenários, o espaço de busca foi determinado para as variáveis de elemento entre zero e o número máximo de elementos (100), sendo sempre números inteiros. Para a taxa de dano, o espaço de busca variou entre 1 (integridade) e 0 (dano total). A fim de comparação com Gomes (2017) utilizando-se o SFO e o AG, o mesmo número de população ( $10*d$ ) e o mesmo número de iterações (100). O programa foi executado 10 vezes para todos os Cenários, com exceção daqueles cenários que havia ruído nos sensores, que foram executados cinco vezes.

Como dito no Capítulo 4, o modo de vibração é o parâmetro calculado pelo MEF para fornecer ao otimizador os sinais para localizar o dano a partir da busca estocástica, tendo-se como referência os modos de vibração com intensidade e localização conhecidas pelo operador deste método. Isto, pois neste trabalho não se utilizou dados experimentais reais. Durante o processo de otimização, o otimizador não sabe nada sobre esta localização e a intensidade do dano.

### 5.3.1 Sinais sem Ruídos

Os Cenários I, II, e III são constituídos de danos em um único elemento, o elemento ( $N_e$ ) 19. Para cada um dos cenários foram aplicadas uma taxa de dano ( $\alpha$ ) de 0.2, 0.5 e 0.9, respectivamente. Sendo a taxa de dano de 0.2 a que apresenta a maior redução de rigidez (dano mais severo) e, portanto espera-se maior facilidade do método de identificação neste dano. Os parâmetros utilizados nas buscas destes três cenários estão na Tabela 5.13 e o tempo médio de

cada simulação foi de 4996 segundos já que para cada ponto da população e de cada iteração, o MEF atua calculando os modos de vibração.

**Tabela 5.13** – Parâmetros do LA para identificação de um elemento

$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$
20	100	1000000	1	150	0.2	0

Os resultados totais estão no Apêndice D e os resumos com a média e desvio padrão estão na Tabela 5.14. A Tabela 5.15 traz figuras comparativas entre expectativa e identificação para estes três cenários.

**Tabela 5.14** – Resultados da busca pelo dano localizado no elemento número 19 com diferentes taxas de severidade

$N_e$	$\alpha$	-	$N_e^*$	$\alpha^*$	$J_{min}$
19	0.2	Média	19	0.2000	0.0001
		Desvio Padrão	0	0.0002	0.2372
19	0.5	Média	19	0.5000	0.0000
		Desvio Padrão	0	0.0001	0.0017
19	0.9	Média	19	0.9000	0.0000
		Desvio Padrão	0	0.0041	0.0005

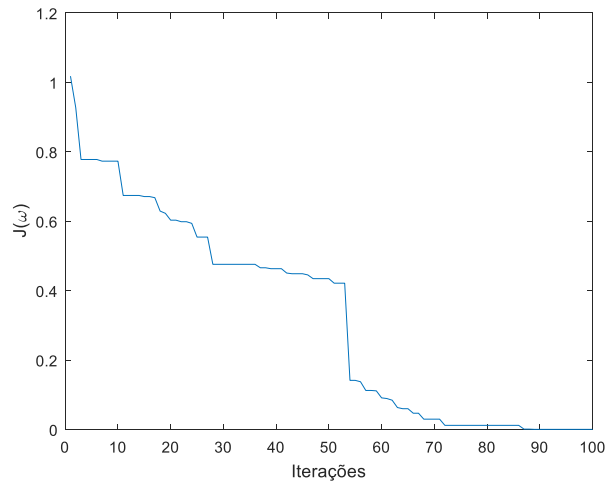
**Tabela 5.15** - Resultados visuais da busca do dano para todos os cenários da tabela anterior

$\alpha$	0.2	0.5	0.9																																																																																																																																																																																																																																																																																																												
Objetivo	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10
	91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																						
	81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																					
	71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																					
	61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																					
	51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																					
	41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																					
	31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																					
	21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																					
	11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																					
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
Identificado	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"> <tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr> <tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr> <tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr> <tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr> <tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr> <tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr> <tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10
	91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																						
	81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																					
	71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																					
	61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																					
	51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																					
	41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																					
	31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																					
	21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																					
	11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																					
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						

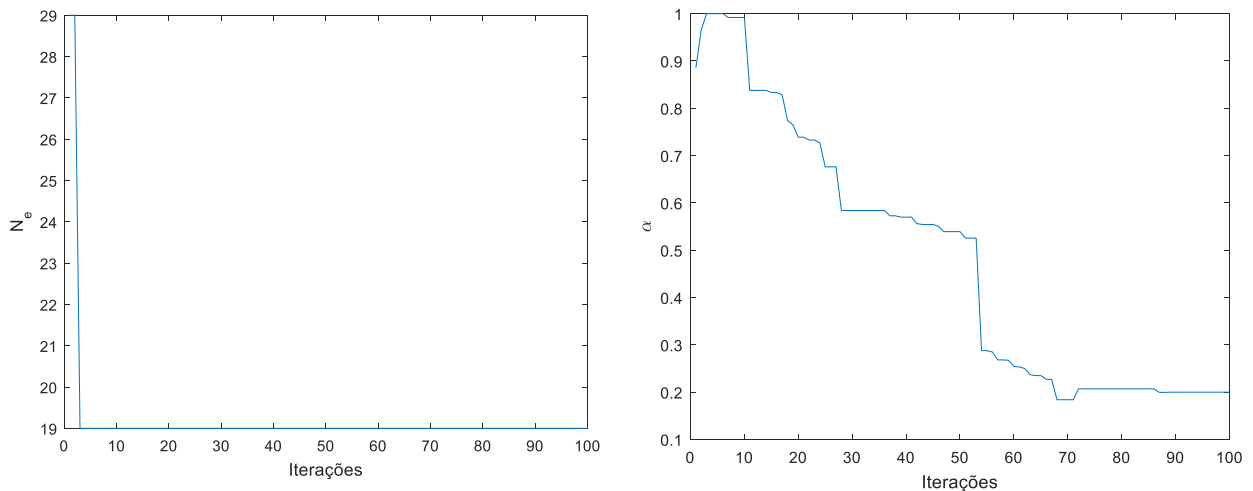


Como pôde ser demonstrado, mesmo considerando um dano com somente uma redução de rigidez de 10% (cenário III), o LA conseguiu localizar com muita precisão a localização do dano e também a intensidade do mesmo sem diferenças na quarta casa decimal na média. Os desvios padrões foram extremamente baixos e crescentes para a intensidade do dano na medida em que este diminuía, revelando realmente uma dificuldade ligeiramente maior para detectá-lo devido às diferenças nos modos de vibração cada vez menores.

Para demonstrar o desempenho do LA, as Figuras 5.3 e 5.4 mostram os gráficos de convergência da função objetivo e das variáveis para o cenário I pelo número total de iterações, sendo a primeira variável a localização do dano ( $N_e$ ) e a segunda a taxa de dano ( $\alpha$ ). Quanto mais próxima de zero é a função objetiva ( $J$ ), mais próxima a identificação do dano se aproxima da realidade.



**Figura 5.3** – Convergência da função objetivo para o Cenário I

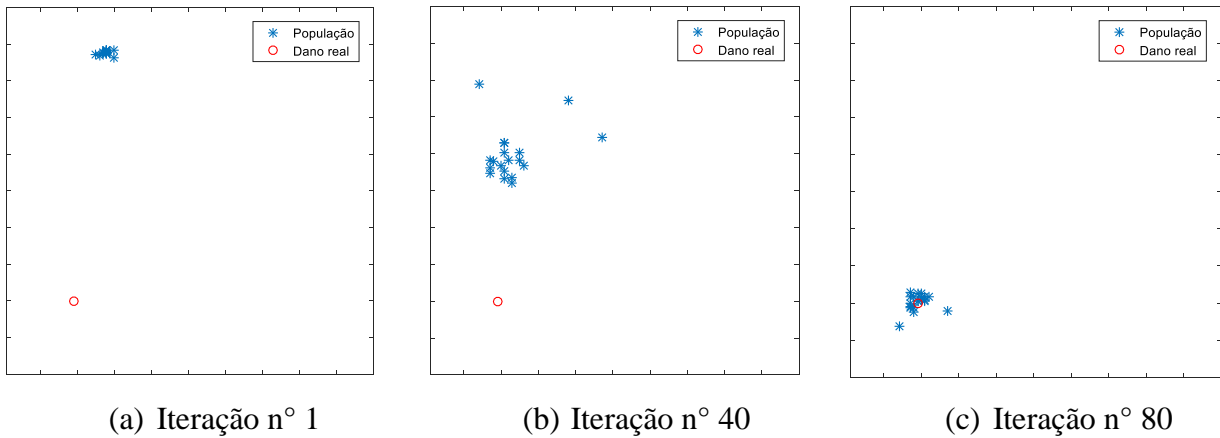


(a) Convergência da localização do dano

(b) Convergência da taxa do dano

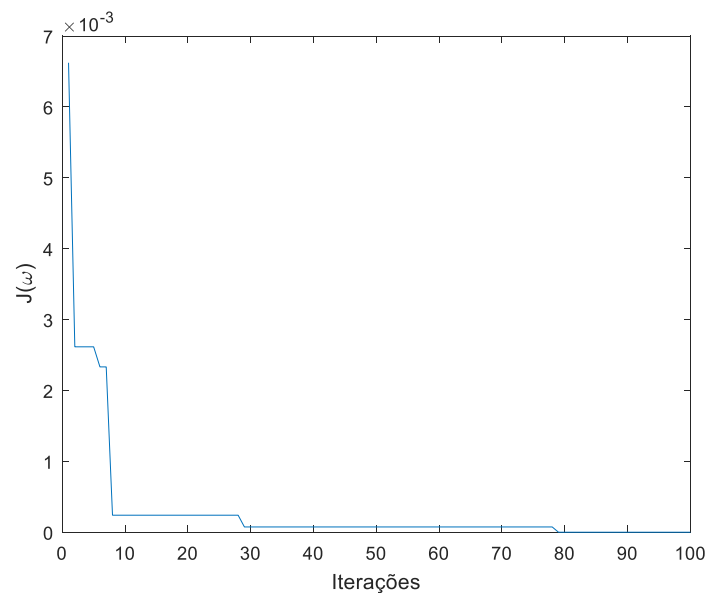
**Figura 5.4** – Gráficos de convergência da identificação do dano para o Cenário I

Ainda, para o cenário I foi feita uma análise da aproximação do ponto ótimo de determinadas iterações para o ponto ótimo global. Uma comparação dos melhores pontos (menores valores de função objetivo) da iteração dada e sua distância para o dano real, representadas na Figura 5.5.

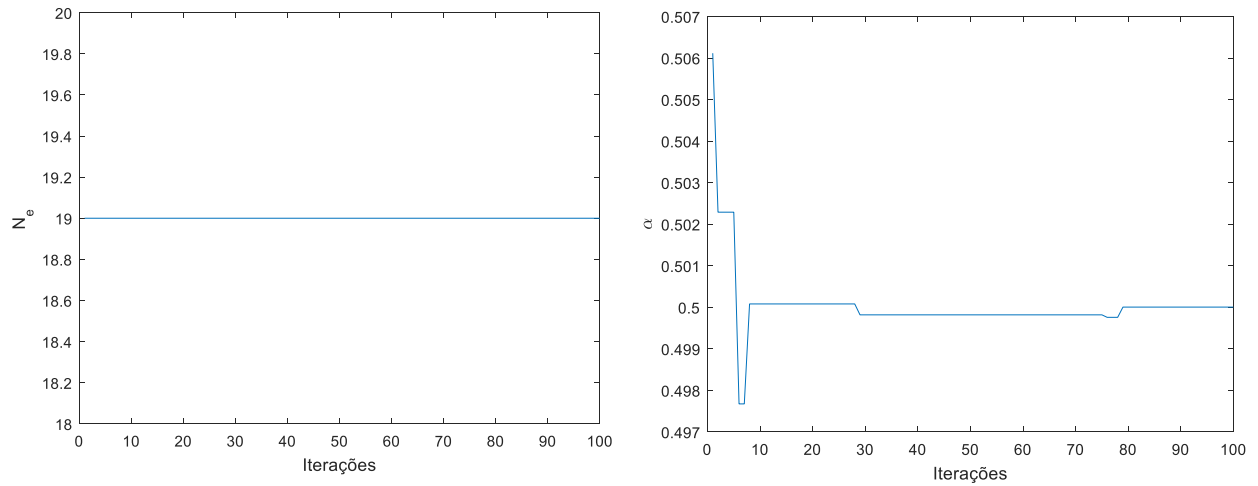


**Figura 5.5** – Distância entre o dano calculado pelo LA e o dano real para algumas iterações

Para os cenários II e III, têm-se os gráficos de convergência da função objetivo e das variáveis em função do número total de iterações. Estão representados nas Figuras 5.6 a 5.9.



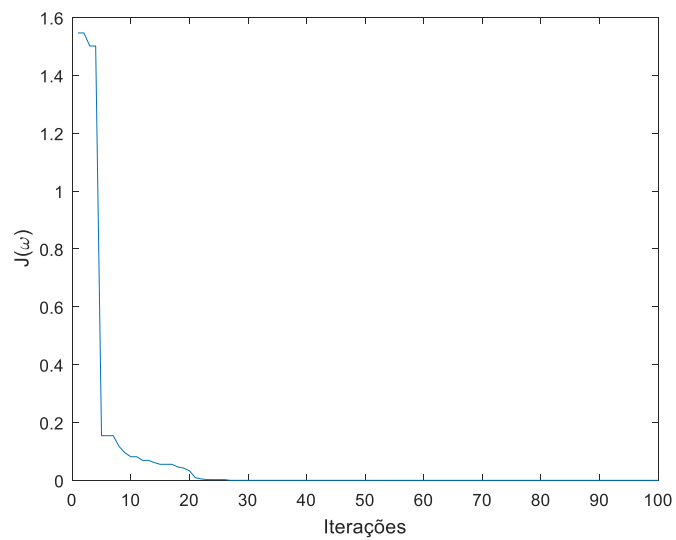
**Figura 5.6** – Convergência da Função Objetivo para o Cenário II



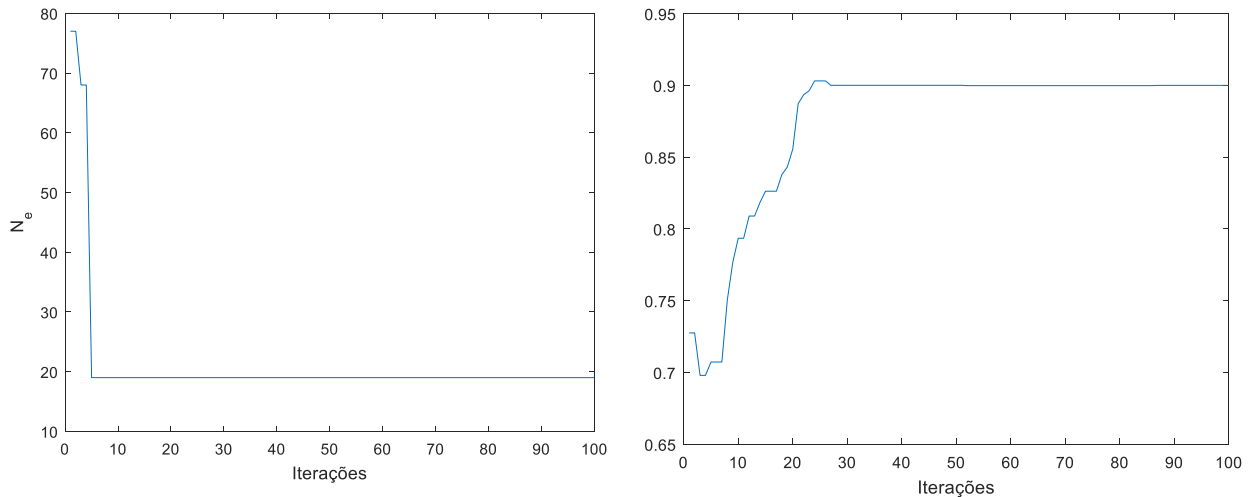
(a) Localização do dano na primeira iteração

(b) Convergência da taxa de dano

**Figura 5.7** – Gráficos de convergência da identificação do dano para o Cenário II



**Figura 5.8** – Convergência da Função Objetivo para o Cenário III



(a) Convergência da Localização do dano

(b) Convergência da taxa de dano

**Figura 5.9** – Gráficos de convergência da identificação do dano para o Cenário III

Como pôde ser demonstrado, mesmo considerando um dano com somente uma redução de rigidez de 10% (Cenário III), o LA conseguiu localizar com muita precisão a localização do dano e também a intensidade do mesmo sem diferenças na quarta casa decimal na média. Os desvios padrões foram extremamente baixos e crescentes para a intensidade do dano na medida em que este diminuía, revelando realmente uma dificuldade ligeiramente maior para detectá-lo devido às diferenças nos modos de vibração cada vez menores. Isto também pode ser observado pelas convergências da função objetivo e das variáveis expostas nas figuras acima. Observe que a localização do dano foi localizada antes da décima iteração para todos os casos sem ruídos.

O Cenário I deste trabalho já teve o dano identificado com os otimizadores AG e SFO utilizando-se o mesmo número de população e iterações para fins de comparação e os resultados estão na Tabela 5.16.

**Tabela 5.16** - Resultados da identificação do dano do Cenário I

Alvo	Algoritmo	AG (GOMES, 2019)	SFO (GOMES, 2017)	LA
$N_e = 19$	Média	19	19	<b>19</b>
	Desvio Padrão	0	0	<b>0</b>
$\alpha = 20\%$	Média	20.0011	19.0774	<b>20.0015</b>
	Desvio Padrão	0.0023	1.5841	<b>0.0163</b>

Todos os algoritmos acertaram com precisão a localização do dano, o que em estruturas reais se torna extremamente vantajoso, já que indica uma região limitada para inspeção mais

rigorosa e até substituição de itens. Também, tendo-se como taxa de dano uma redução de 80% da rigidez, todos os algoritmos identificaram bem a intensidade do dano.

Outro tipo de identificação de dano sem ruídos utilizada neste trabalho foi a inserção de um segundo elemento com mesma taxa de dano (dado a restrições de implementação no MEF). Este é o Cenário VIII, onde há danos nos elementos 19 e 65 com taxa de dano de 0,2. Assim, têm-se três variáveis,  $\vec{x} = (N_{e1}, N_{e2}, \alpha)$  e a população é ligeiramente maior para o caso de três variáveis. Os parâmetros do LA utilizados estão na Tabela 5.17.

**Tabela 5.17** – Parâmetros do LA para identificação de danos em dois elementos

$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$
60	100	1000000	1	150	0.4	0

O tempo médio de cada simulação foi de 22475 segundos e o resultado resumido segue na Tabela 5.18. Os resultados completos estão no Apêndice D.

**Tabela 5.18** - Resultados da identificação de dois danos e mesma severidade

$N_{e1}$	$N_{e2}$	$\alpha$	-	$N_{e1}^*$	$N_{e2}^*$	$\alpha^*$	$J_{min}$
19	65	0.2	Média	19	65	0.2732	0.3890
			Desvio Padrão	0	0.6325	0.0745	0.3238

Como pode ser observado pela Tabela 5.18 e pela Figura 5.10, o LA determinou com exatidão a localização dos dois danos e teve resultado muito próximo da severidade do mesmo.

11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	

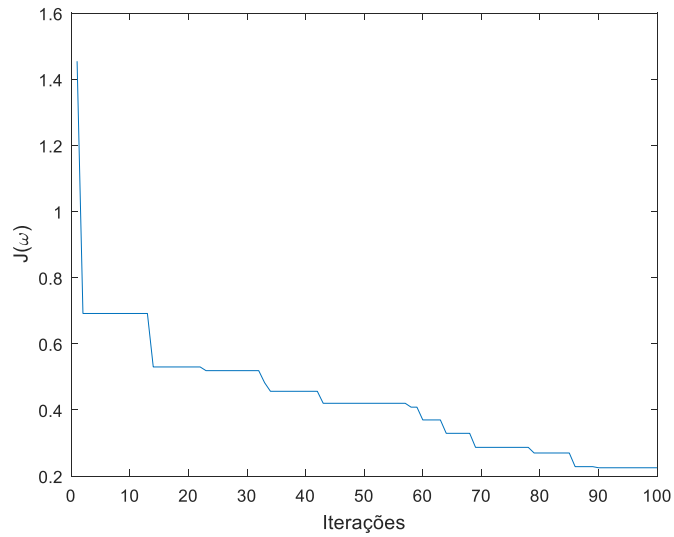
(a) Objetivo

11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	

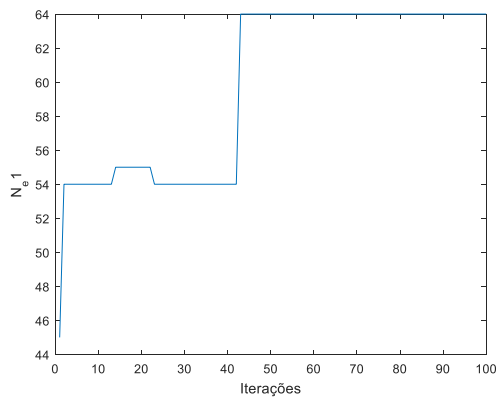
(b) Identificado

**Figura 5.10** – Resultados visuais da identificação de dois danos

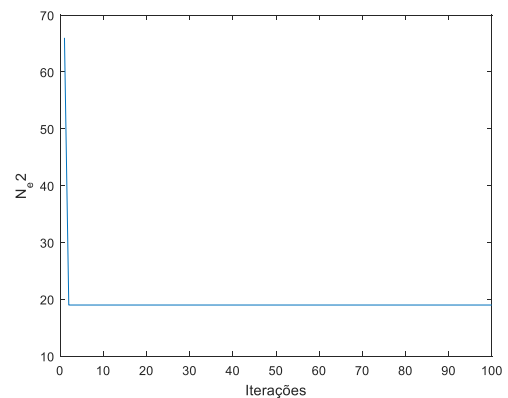
As Figuras 5.11 e 5.12 mostram os gráficos de convergência para a função objetivo e para as três variáveis da identificação dos dois anos deste cenário.



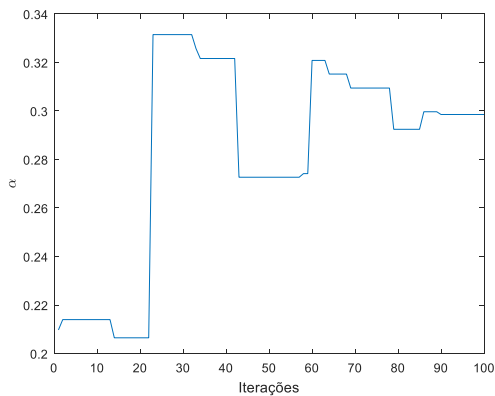
**Figura 5.11** – Convergência da Função Objetivo para o Cenário VIII



(a) Convergência da localização do primeiro dano



(b) Convergência da localização do segundo dano



(c) Convergência  
Da taxa de dano

**Figura 5.12** – Gráficos de convergência da identificação do dano para o Cenário VIII

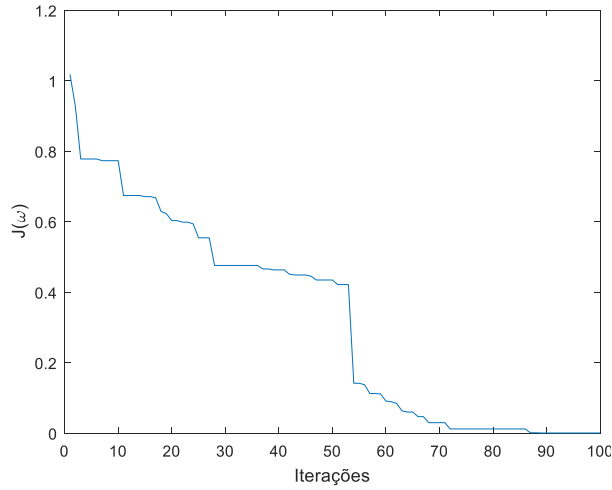
### 5.3.1 Sinais com Ruídos

O Cenário I foi simulado novamente com diferentes intensidades de ruídos: 1, 2, 5 e 10%, dando origem aos Cenários IV, V, VI e VII deste trabalho. A formulação do ruído inclusa no MATLAB<sup>®</sup> foi conforme a Equação (21). Todos os parâmetros são os mesmos do Cenário I, descritos acima na Tabela 5.13. Os resultados completos estão no Apêndice D e na Tabela 5.19 estão os resultados de média e desvio para estes quatro cenários com ruídos e os do Cenário I, sem ruídos, a fim de comparação.

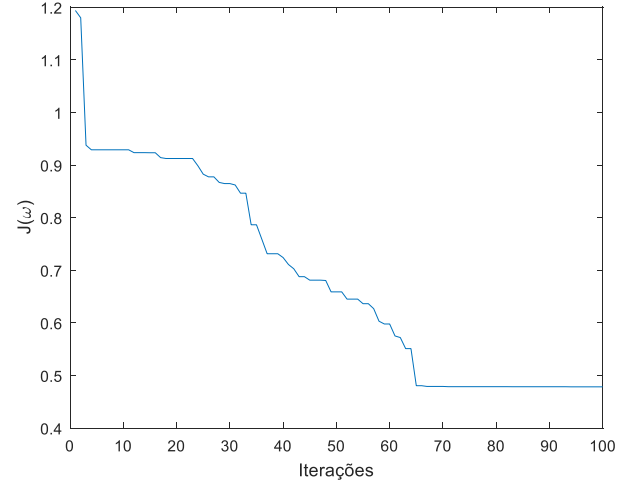
**Tabela 5.19** - Influência do Ruído na média dos resultados

Nível de Ruído (%)		$N_e^*$	$\alpha^*$	$J_{min}$
0	Média	19	0.2000	0.0001
	Desvio	0	0.0002	0.2372
1	Média	19	0.1967	0.2284
	Desvio	0	0.0117	0.0413
2	Média	19	0.1937	0.3493
	Desvio	0	0.0231	0.0582
5	Média	19	0.2278	0.8391
	Desvio	0	0.0417	0.1189
10	Média	19	0.1776	1.6197
	Desvio	0	0.0402	0.1949

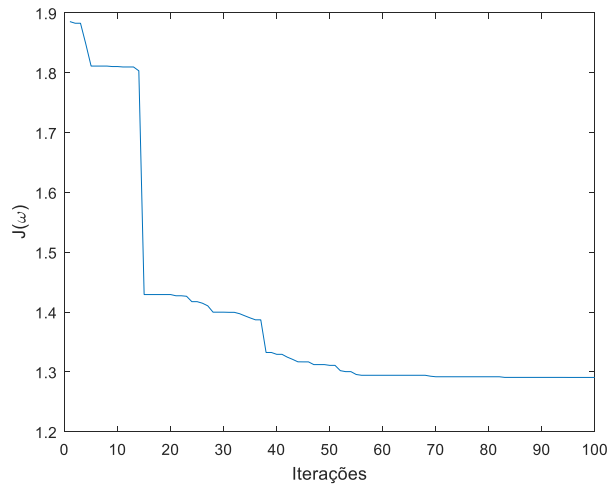
As Figuras 5.13 a 5.15 trazem os resultados gráficos para a identificação de danos com ruídos de 0, 2, 5 e 10% de maneira a poder compará-los.



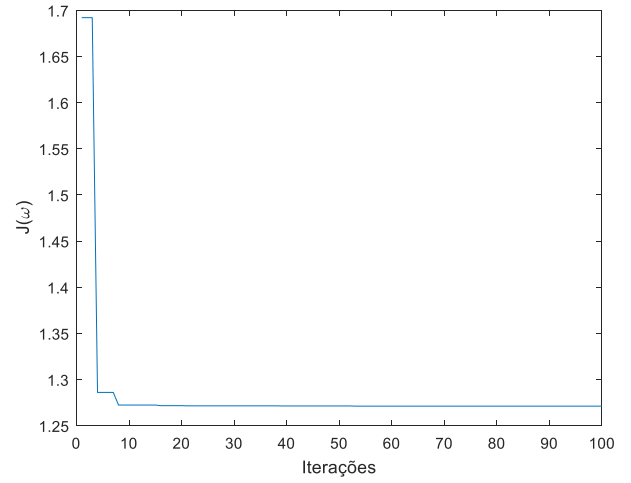
(a) 0%



(b) 2%



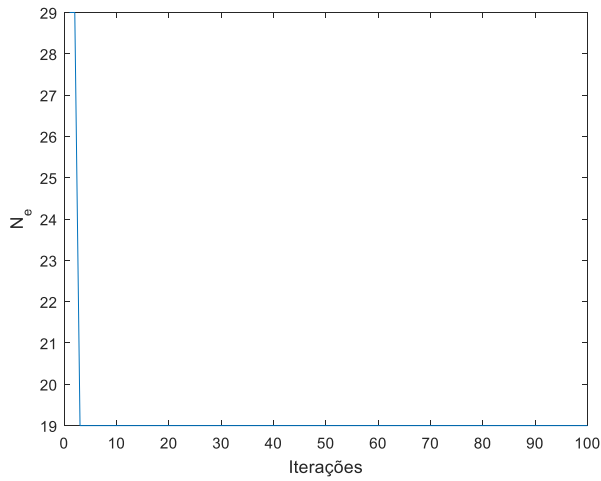
(c) 5%



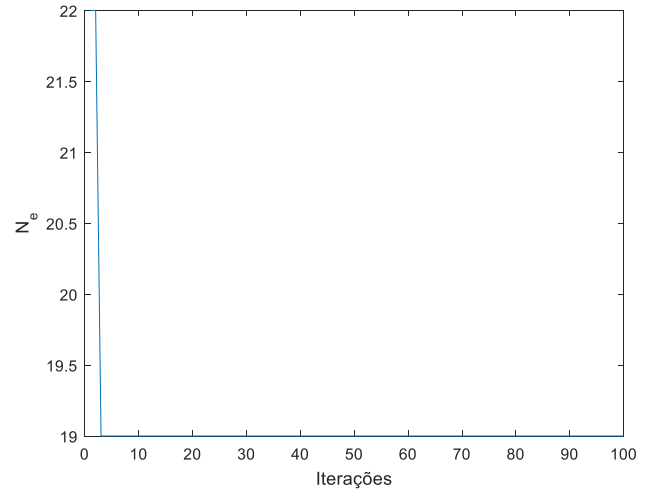
(d) 10%

**Figura 5.13** – Convergências das funções objetivos para o Cenário I sem e com adição de ruídos

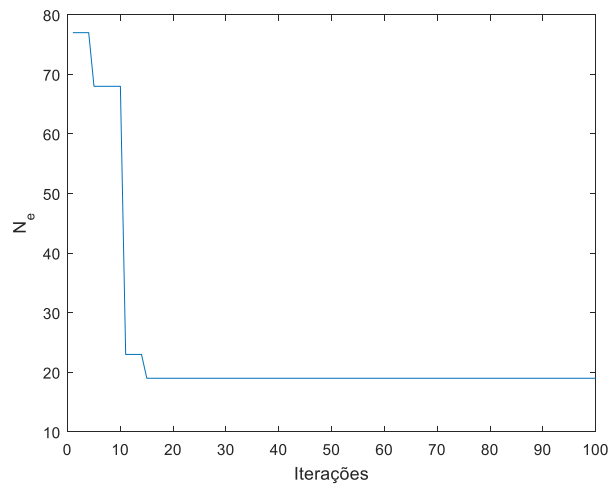




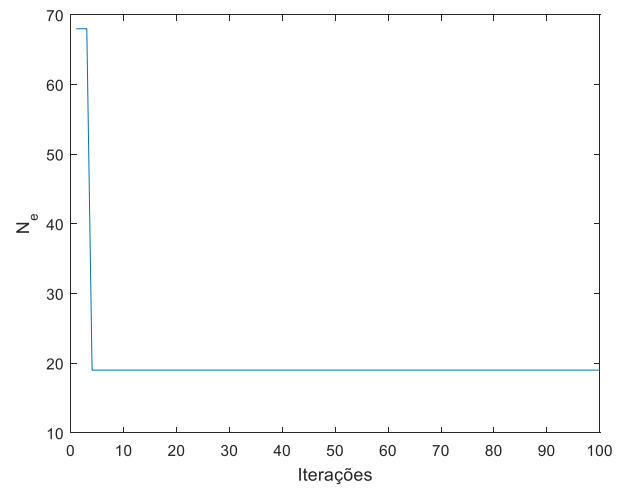
(a) 0%



(b) 2%

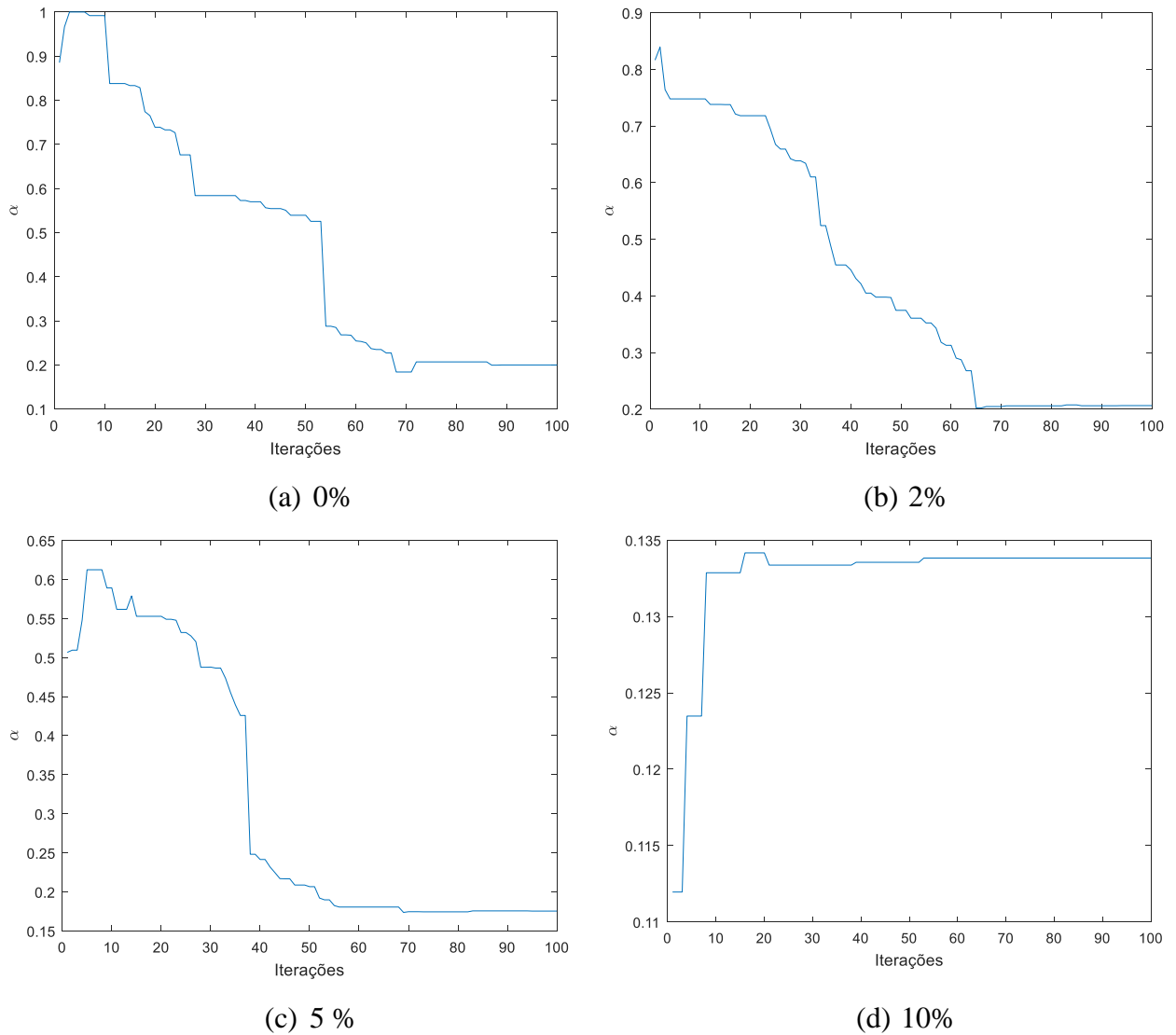


(c) 5 %

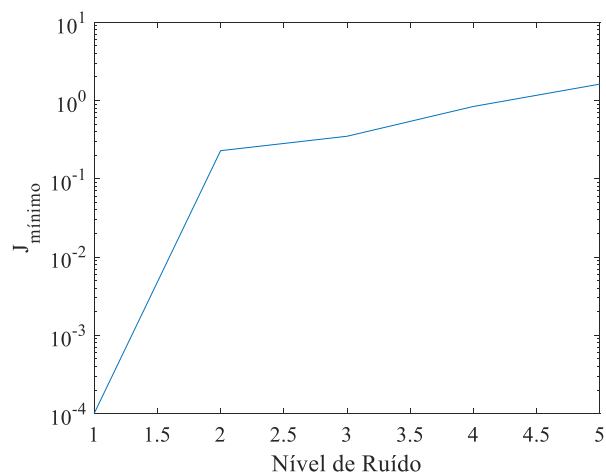


(d) 10%

**Figura 5.14** – Convergências das localizações do dano para o Cenário I sem e com adição de ruídos



**Figura 5.15** – Convergências das taxas de dano para o Cenário I sem e com adição de ruídos

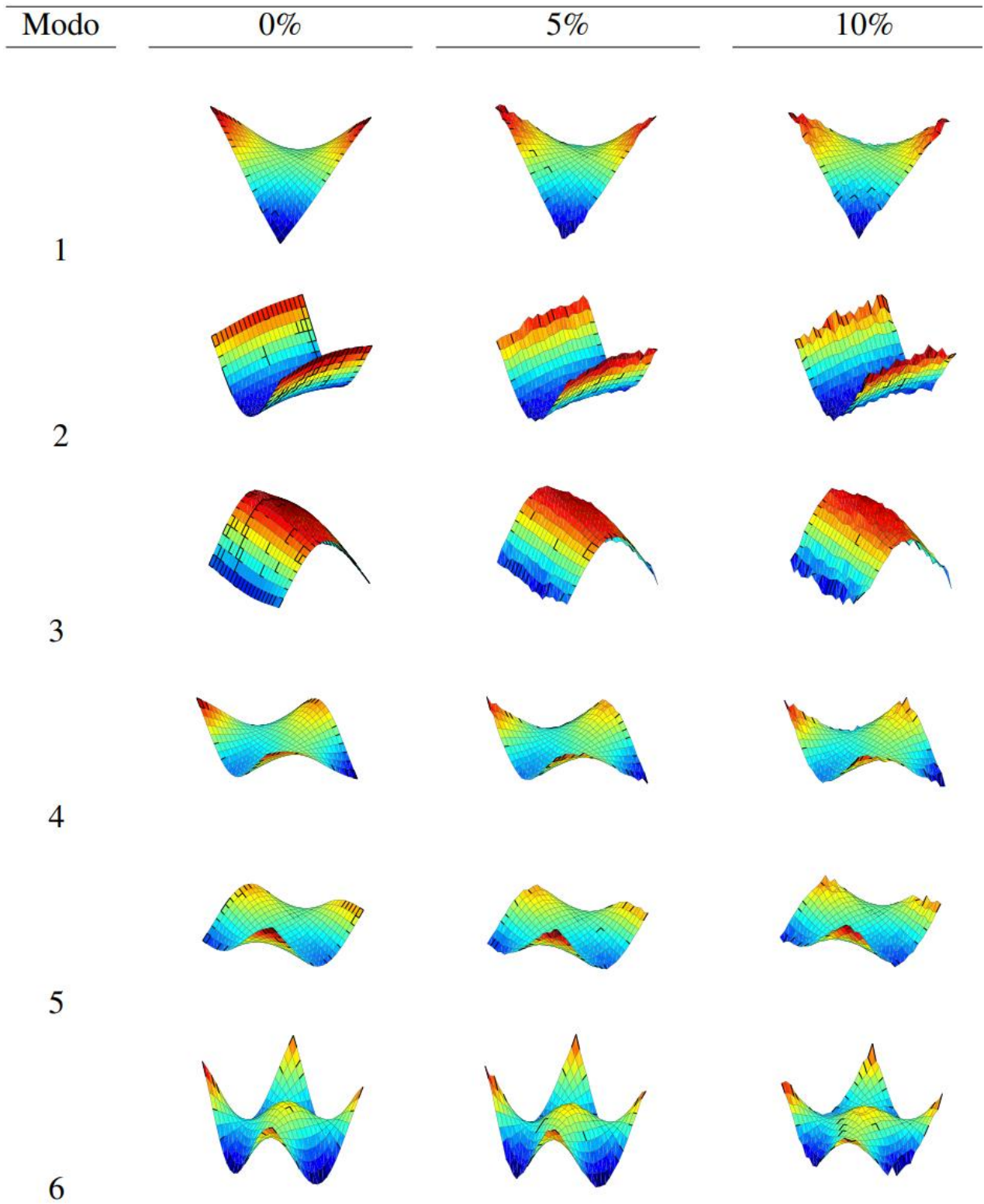


**Figura 5.16** – Influência negativa do acréscimo de ruídos na precisão da identificação do dano

Pôde-se observar que o aumento do ruído deteriora a precisão do método e o melhor parâmetro para avaliar isto é o valor médio do  $J_{min}$ , como pode ser visto na Tabela 5.18 e no gráfico da Figura 5.16.

O LA mesmo com muito ruído (10%) previu a localização exata do dano. O fato de esta variável estar programada pra ser um número inteiro pode contribuir pra este resultado. Já em relação à taxa de dano, a variação máxima do erro percentual para o caso de maior ruído foi de 12,6% em relação ao alvo.

A Figura 5.17 traz graficamente como a superfície de resposta pode se tornar poluída com os níveis de ruídos para os seis primeiros modos de vibração na placa de compósito deste trabalho para a identificação de danos.



**Figura 5.17** – Poluição ruidosa nos seis primeiros modos de vibração (Adaptado de GOMES, 2017)

Gomes (2017) obteve resultados utilizando 1, 5 e 10% de ruído utilizando os algoritmos SFO e AG. Na Tabela 5.20 está a comparação do desempenho sob-ruídos destes dois algoritmos e o LA para a identificação do dano do Cenário I, isto é,  $N_e = 19$  e  $\alpha = 20\%$ .

**Tabela 5.20** - Desempenho da identificação de dano do Cenário I com níveis diferentes de ruídos

Nível de Ruído (%)	Algoritmo	AG (GOMES, 2017)		SFO (GOMES, 2017)		LA	
1	Média	19	19.6168	19	19.7982	<b>19</b>	<b>19.6650</b>
	Desvio Padrão	0	0.0706	0	0.4469	<b>0</b>	<b>1.1742</b>
5	Média	19	22.5694	19	19.7982	<b>19</b>	<b>22.7810</b>
	Desvio Padrão	0	14.7195	0	0.4469	<b>0</b>	<b>4.1750</b>
10	Média	62	41.4346	19	17.1615	<b>19</b>	<b>17.7580</b>
	Desvio Padrão	17,9000	47.8311	0	2.2176	<b>0</b>	<b>4.0161</b>

Para 1 e 5% de ruídos todos os algoritmos localizaram com exatidão a localização do dano. Porém, com 10% o LA e o SFO apresentaram resultados melhores. A Tabela 5.21 mostra em desenhos o objetivo e os resultados para os três algoritmos com adição de 10% de ruídos nos sinais.

**Tabela 5.21** – Resultados pela busca do dano em condição muito ruidosa

Ruído (%)	10	5	1																																																																																																																																																																																																																																																																																																												
Objetivo	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10
	91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																						
	81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																					
	71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																					
	61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																					
	51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																					
	41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																					
	31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																					
	21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																					
	11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																					
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
Identificado	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	<table border="1"><tr><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td><td></td></tr><tr><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td></tr><tr><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr><tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td></tr><tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td></tr><tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr></table>	91	92	93	94	95	96	97	98	99		81	82	83	84	85	86	87	88	89	90	71	72	73	74	75	76	77	78	79	80	61	62	63	64	65	66	67	68	69	70	51	52	53	54	55	56	57	58	59	60	41	42	43	44	45	46	47	48	49	50	31	32	33	34	35	36	37	38	39	40	21	22	23	24	25	26	27	28	29	30	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10
	91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																						
	81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																					
	71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																					
	61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																					
	51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																					
	41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																					
	31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																					
	21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																					
	11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																					
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						
91	92	93	94	95	96	97	98	99																																																																																																																																																																																																																																																																																																							
81	82	83	84	85	86	87	88	89	90																																																																																																																																																																																																																																																																																																						
71	72	73	74	75	76	77	78	79	80																																																																																																																																																																																																																																																																																																						
61	62	63	64	65	66	67	68	69	70																																																																																																																																																																																																																																																																																																						
51	52	53	54	55	56	57	58	59	60																																																																																																																																																																																																																																																																																																						
41	42	43	44	45	46	47	48	49	50																																																																																																																																																																																																																																																																																																						
31	32	33	34	35	36	37	38	39	40																																																																																																																																																																																																																																																																																																						
21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																						
11	12	13	14	15	16	17	18	19	20																																																																																																																																																																																																																																																																																																						
1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																						

Além da localização, o SFO e o LA conseguiram na pior condição de ruídos deste trabalho terem uma diferença percentual de erro de 14,2% para o primeiro e 11,21% para o último na identificação da taxa do dano do dano.

# Capítulo 6

## CONCLUSÃO

### 6.1 Conclusões

Como tema principal desta dissertação, teve-se o desenvolvimento de uma nova meta-heurística inspirada no fenômeno físico do relâmpago que hibridizou as classificações de trajetória e população. Inicialmente parecia distante um modelo matemático que conseguisse estocasticamente gerar as famosas Figuras de Lichtenberg, no entanto, o DLA surgiu como ferramenta imprescindível para a construção do novo algoritmo e permitiu sua origem. Depois de ajustes e apropriações, o Lichtenberg Algoritmo (LA) foi construído e apresentou resultados surpreendentes em sua primeira versão utilizando-se de umas das propriedades mais importantes da sua natureza: os fractais.

Como primeira barreira de criação, o LA enfrentou as Funções de Teste e as superou com resultados muito positivos, tendo-se diferenças mínimas entre o resultado encontrado e o ideal, gerando desvios padrões muito baixos. É correto afirmar que neste momento, as funções de teste não eram muito complexas e as únicas restrições do problema eram referentes ao espaço de busca.

Apesar dos bons resultados, notou-se que o algoritmo devido à dependência ao DLA para a geração da FL mostrou-se lento em relação a muitos encontrados na literatura. Porém, faz parte da sua natureza ao menos nesta versão. Também se concluiu que devido a esta questão do DLA,

a criação de matrizes cada vez maiores para a geração de figuras para  $d$  variáveis se tornaria muito crítica. O tempo de geração das figuras aumentaria bastante e ainda, teriam-se poucos pontos para serem utilizados em cada eixo na avaliação das funções objetivos pelo otimizador, já que na tentativa de criar esta condição, o *software* utilizado apresenta limitações em relação à memória para criar matrizes longas e multidimensionais.

Isto corrobora para uma sutil ineficiência do algoritmo para otimização de funções de três variáveis, dada à precarização da precisão e pela demora ainda maior para criação de uma DLA tridimensional. Isto se comparado a todos os outros casos, mas ainda assim possui bons resultados.

Validado pelas Funções de Teste, o LA enfrentou um problema não linear, o Problema da Viga Soldada, onde procurou pontos ótimos de projeto em regiões multimodais e complexas com limitações além da do espaço de busca. Encontrou resultados positivos se comparados aos encontrados na literatura, onde seu maior erro foi de aproximadamente 7,5% em uma das variáveis de projeto. Obteve resultado melhor que o próprio AG em versões mais antigas e obteve parâmetros ótimos geométricos para o problema em questão.

Por fim, o LA foi utilizado para a identificação de delaminações onde não errou em nenhum caso a localização do dano. Com taxas diferentes de danos, que variaram de 10 a 80% da redução de rigidez local, as diferenças na busca da severidade de dano não tiveram diferenças até a quarta casa decimal. Um resultado muito positivo. Com a adição de ruídos, que variaram de 1 a 10%, também não houve erro de localização e o erro percentual na identificação da severidade foi de aproximadamente 10%, mais um resultado muito positivo demonstrando um potencial grande do LA na identificação de danos em ambientes com leituras de sinais bastante ruidosas e com severidades de dano baixas. Este resultado não foi conquistado pelo AG e dentro dos algoritmos comparados, o LA foi o que teve menor erro percentual em relação à média encontrada e o alvo que era buscado. Ainda assim, foi perceptível que a baixa severidade do dano e o aumento de ruídos nos sinais deterioram a precisão do algoritmo na identificação do dano. No caso do LA, sutilmente para os casos estudados.

Dada à satisfação pelo resultado do LA frente aos problemas propostos, como resultado teve-se o registro do LA no Instituto Nacional de Propriedade Intelectual (INPI) que foi aprovado às vésperas da conclusão da escrita desta Dissertação, cujo número do processo é BR512019002567-9 e o título é *Lichtenberg Algoritmo Optimization* (LA). O Certificado encontra-se no Anexo E.



## 6.2 Sugestões Futuras

Algumas reflexões são deixadas como propostas futuras:

- Elaboração de uma nova estratégia que diminuiria o tempo de simulações dentro do código do LA para todas as variáveis.
- Nova proposta para melhorar o tempo de simulação e precisão para o LA tratando-se de três variáveis. Este e o caso acima já consideram não gerar a figura na tela, mas nada pode ser feito em relação à matriz de formação do DLA.
- Um trabalho estatístico de otimização dos parâmetros do LA para funções objetivos e número de variáveis específicas, já que se percebeu a sensibilidade do algoritmo frente a modificações destas. Uma sugestão é a otimização por superfície de resposta. Neste contexto pode-se ter duas frentes: considerando-se apenas o melhor resultado (ou menor  $J$ ) e considerando-se o melhor resultado com o menor tempo de simulação.
- Pode-se também realizar simulações para observar a partir de qual nível de ruído o LA não consegue mais detectar com precisão ao menos a localização do dano, para validar níveis de eficiência do mesmo.
- Verificar o desempenho do LA na identificação de danos em estruturas não compósitas, principalmente metálicas, com valores de rigidez e massas diferentes.
- Aplicar a metodologia usada em mais problemas e aprimorá-la para estruturas mais complexas.

# REFERÊNCIAS BIBLIOGRÁFICAS

- Baril, C.; Yacout, S.; Clément, B. **Design for Six Sigma through collaborative multiobjective optimization.** *Computers and Industrial Engineering*, v. 60, p. 43–55, 2011.
- Barthorpe & Worden (2009) Robert J Barthorpe e Keith Worden. **Sensor placement optimization.** *Encyclopedia of structural health monitoring*.
- Braun *et al.* (2015) Carlos E Braun, Leonardo D Chiwiacowsky, e Arthur T Gomez. **Variations of ant colony optimization for the solution of the structural damage identification problem.** *Procedia Computer Science*, 51:875–884.
- Deb, K. **Optimal design of a welded beam via genetic algorithms,** *AIAA J.* 29 (11) (1991) 2013–2015.
- Coello Coello C.A., **Use of a self-adaptive penalty approach for engineering optimization problems,** *Comput. Ind.* 41 (2) (2000) 113–127.
- Coello Coello C.A., E. Mezura Montes, **Constraint-handling in genetic algorithms through the use of dominance-based tournament selection,** *Adv. Eng. Inf.* 16 (3) (2002) 193–203.
- Cha & Buyukozturk (2015) Young-Jin Cha e Oral Buyukozturk. **Structural damage detection using modal strain energy and hybrid multiobjective optimization.** *Computer-Aided Civil and Infrastructure Engineering*, 30(5):347–358.
- Chakraborty (2005) D Chakraborty. **Artificial neural network based delamination prediction in laminated composites.** *Materials & design*, 26(1):1–7.
- Cheng R, Li M, Tian Y, Xiang X, Zhang X, Yang S, Jin Y, Yao X (2018) **Benchmark functions for the cec'2018 competition on many-objective optimization.**
- Chiandussi, G.; Codegone, M.; Ferrero, S.; Varesio, F.E. **Comparison of multi-objective optimization methodologies for engineering applications.** [S.l.]:Eslsevier Ltd, 2012. 912-942p.

- Chopard B, Tomassini M (2018). **Problems, Algorithms, and Computational Complexity. An Introduction to Metaheuristics for Optimization.** Springer, pp 1-14
- Di Rocco, E. D. **DLA Model.** Seminar Univerza V Ljubljani, 2009.
- Dorigo M. **Optimization, learning and natural algorithms.** Ph.D. thesis, Politecnico di Milano, Italy; 1992.
- Elaziz MA, Mirjalili S (2019). **A hyper-heuristic for improving the initial population of whale optimization algorithm.** *Knowledge-Based Systems* 172:42-63
- Fister Jr, I., Yang, X. S., Fister, I., Brest, J., Fister, D. **A Brief Review of Nature Inspired Algorithms for Optimization.** University of Maribor, 2013.
- Ganguli (2001) Ranjan Ganguli. **A fuzzy logic system for ground based structural health monitoring of a helicopter rotor using modal data.** *Journal of Intelligent Material Systems and Structures*, 12(6):397–407.
- Garcia *et al.* (2015) D Garcia, R Palazzetti, I Trendafilova, C Fiorini, e A Zucchelli. **Vibration based delamination diagnosis and modelling for composite laminate plates.** *Composite Structures*, 130:155–162.
- Garg (1988) Amar C Garg. **Delamination a damage mode in composite structures.** *Engineering Fracture Mechanics*, 29(5):557–584.
- Gomes, G. F. **Otimização da Identificação de Danos Estruturais por meio de Inteligência Computacional e Dados Modais**, 188 p. Tese (Doutorado em Engenharia Mecânica) – Instituto de Engenharia Mecânica, Universidade Federal de Itajubá, Itajubá, 2017.
- Gomes, G. F., Mendéz, Y. A. D., Alexandrino, P. S. L., CUNHA Jr, S. S., ANCELOTTI, A. C. (2018a). **The use of intelligent computational tools for damage detection and identification with an emphasis on composites— a review.** *Compos Struct* 196:44–54.
- Gomes, G. F., da Cunha, S. S., & Ancelotti, A. C. (2018b). **A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates.** *Engineering with Computers*.

- Gomes, G. F., de Almeida, F. A., da Silva Lopes Alexandrino, P., da Cunha, S. S., de Sousa, B. S., & Ancelotti, A. C. (2018c). **A multiobjective sensor placement optimization for SHM systems considering Fisher information matrix and mode shape interpolation.** *Engineering with Computers.*
- Gomes, G. F., da Cunha, S. S., & Ancelotti, A. C. (2018d). **A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates.** *Engineering with Computers.*
- Gomes, G. F., de Almeida, F. A., Junqueira, D. M., da Cunha, S. S., & Ancelotti, A. C. (2019). **Optimized damage identification in CFRP plates by reduced mode shapes and GA-ANN methods.** *Engineering Structures, 181, 111-123.*
- Gomes, J **Método dos polinômios canônicos de misturas para otimização multi-objetivo.** Itajubá, Minas Gerais, Brasil: Tese de Doutorado – Programa de Pós Graduação em Engenharia de Produção – Universidade Federal de Itajubá, 2013.
- Gomes, J. H. F.; Paiva, a. P.; Costa, S. C.; Balestrassi, P. P.; Paiva, E. J. **Weighted Multivariate Mean Square Error for processes optimization: A case study on flux-cored arc welding for stainless steel claddings.** *European Journal of Operational Research, Elsevier B. V., v226, n.3, p. 522-535, 2013.*
- Gopalakrishnan *et al.*(2011a) Srinivasan Gopalakrishnan, Massimo Ruzzene, e Sathyanarayana Hanagud. **Computational techniques for damage detection, classification and quantification.** Em *Computational Techniques for Structural Health Monitoring*, páginas 407–461. Springer.
- Gu *et al.* (2016) Guimei Gu, Yu Zhao, e Xin Zhang. **Optimal layout of sensors on wind turbine blade based on combinational algorithm.** *International Journal of Distributed Sensor Networks.*
- Guan *et al.* (2017) Xuefei Guan, Yongxiang Wang, e Jingjing He. **A probabilistic damage identification method for shear structure components based on cross-entropy optimizations.** *Entropy, 19(1):27.*

- Guo *et al.* (2004) HY Guo, L Zhang, LL Zhang, e JX Zhou. **Optimal placement of sensors for structural health monitoring using improved genetic algorithms.** *Smart Materials and Structures*, 13(3):528.
- Salimi, H. **Stochastic Fractal Search: A powerful metaheuristic algorithm**, *Knowl. Based Syst.* (2014), <http://dx.doi.org/10.1016/j.knosys.2014.07.025>.
- Hatamlou A (2013). **Black hole: A new heuristic optimization approach for data clustering.** *Information sciences* 222:175-184.
- Hearn & Testa (1991) George Hearn e Rene B Testa. **Modal analysis for damage detection in structures.** *Journal of Structural Engineering*, 117(10):3042–3063.
- Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. **Harris hawks optimization: Algorithm and applications.** 2019. *Future Generation Computer Systems* 97:849-872
- Heslehurst (2014) Rikard Benton Heslehurst. **Defects and damage in composite materials and structures.** CRC Press.
- Holland J. **Adaptation in natural and artificial systems.** Ann Arbor, MI, USA: *University of Michigan Press*; 1975.
- Hypescience. **Artista capta raios em acrílico transformando fenômeno elétrico em obra de arte.** Acesso em Outubro de 2019. Disponível em < <https://hypescience.com/artista-capta-relampagos-em-acrilico-transformando-fenomeno-eletrico-em-obra-de-arte/>>.
- Illusions and Science. **Resonance Experiment.** Acesso em Outubro de 2019. Disponível em <<https://www.youtube.com/channel/UCeQEKFH31vvD-InkTGSvCrA>>
- INPE – Instituto Nacional de Pesquisas Espaciais. **Fotos do Mês.** Acesso em Outubro de 2019. Disponível em < <http://www.inpe.br/webelat/homepage/menu/noticias/foto.do.mes.php>>
- Irurzun, I. M. P. Bergero, V. Mola, M. C. Cordero, J. L. Vicente, E. E. Mola. **Dielectric breakdown in solids modeled by BDM and DLA.** *Chaos, Solitons and Fractals*, 2002.
- Jafarkhani e Masri (2011) Reza Jafarkhani e Sami F Masri. **Finite element model updating using evolutionary strategy for damage detection.** *Computer-Aided Civil and Infrastructure Engineering*, 26(3):207–224.

- Jaimes et al.(2009) Antonio López Jaimes, Saúl Zapotecas Martínez, e Carlos A Coello Coello. **An introduction to multiobjective optimization techniques.** *Optimization in Polymer Processing*, páginas 29–57.
- Kaveh, A. S. Talatahari, **An improved ant colony optimization for constrained engineering design problems**, *Eng. Comput.* 27 (1) (2010) 155–182.
- Kammer & Tinker (2004) Daniel C Kammer e Michael L Tinker. **Optimal placement of triaxial accelerometers for modal vibration tests.** *Mechanical systems and signal processing*, 18(1): 29–41.
- Kaw (2005) Autar K Kaw. **Mechanics of composite materials.** CRC press.
- Kennedy J & Eberhart RC. **Particle swarm optimization.** *In: Proceedings of the IEEE international conference on neural networks*, Piscataway, NJ, USA; 1995. p. 1942–48.
- Kirkpatrick S, Gelatt CD, Vecchi MP. **Optimization by simulated annealing.** *Science* 1983; 220 (4598): 671–80.
- Lee (2000) Jaehong Lee. **Free vibration analysis of delaminated composite beams.** *Computers & Structures*, 74(2): 121–129.
- Lichtenberg, G. C., **Novi. Comment. Gött.** Vol. 8, 168, 1777.
- Lopes, P.S. **Detecção de danos em estruturas por meio de técnicas de redes neurais artificiais e de algoritmos genéticos**, 106p. Dissertação (Mestrado em Projeto e Fabricação) – Instituto de Engenharia Mecânica, Universidade Federal de Itajubá, Itajubá, 2007.
- Lopes *et al.* (2010) Patricia da Silva Lopes, Ariosto Bretanha Jorge, e Sebastiao Simoes Cunha Jr. **Detection of holes in a plate using global optimization and parameter identification techniques.** *Inverse Problems in Science and Engineering*, 18(4):439–463.
- Meakin, P., TOLMAN, S., BLUMEN, A. **Diffusion-Limited Aggregation.** The Royal Society, 1989.
- Merrill, F. H., and Von Hippel, A. (1939). **The atomphysical Interpretation os Lichtenberg Figures and Their Application to the study of Gas Discharge.** *Journal of Applied Physics*, 10(12), 873-887.

- Mirjalili S, Lewis A (2016). **The whale optimization algorithm**. *Advances in engineering software*. 95:51-67
- Mohan *et al.*(2013) SC Mohan, Dipak Kumar Maiti, e Damodar Maity. **Structural damage assessment using frf employing particle swarm optimization**. *Applied Mathematics and Computation*, 219 (20): 10387–10400.
- Montalvão *et al.* (2006) Diogo Montalvão, Nuno Manuel Mendes Maia, e António Manuel Relógio Ribeiro. **A review of vibration-based structural health monitoring with special emphasis on composite materials**. *Shock and Vibration Digest*, 38(4):295–324.
- NACCARATO, K. P. **Estudo de relâmpagos no Brasil com base na análise de desempenho do sistema de localização de tempestades** / K. P. Naccarato – São José dos Campos: INPE, 2001. 165p. – (INPE-8380-TDI/770).
- Nabil E. **A modified flower pollination algorithm for global optimization**. *Expert Systems with Applications*. 2016. 57:192-203
- Nanda *et al.*(2012) Bharadwaj Nanda, Damodar Maity, e Dipak Kumar Maiti. **Vibration based structural damage detection technique using particle swarm optimization with incremental swarm size**. *International Journal Aeronautical and Space Sciences*, 13(3):323–331.
- Niemeyer L, Pietronero L, Wiesmann HJ. *Pys Ver Lett* 1984;52:1033.
- Olorunda O, Engelbrecht AP. **Measuring exploration/exploitation in particle swarms using swarm diversity**. (2008) *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008. IEEE, pp 1128- 1134.
- Ooijevaar (2014) Theodorus Hendricus Ooijevaar. **Vibration based structural health monitoring of composite skin-stiffener structures**. Universiteit Twente.
- Osyczka, A.; Kundu, S. **A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm**. *Structural Optimization*, v. 10, n. Goldberg 1989, p. 94–99, 1995.
- Patel VK, Savsani VJ . **Heat transfer search (HTS): a novel optimization algorithm**. *Information Sciences* 324:217-246 32. (2015)

- Pinto, I. R. C. A. **Estudos sobre campos elétricos e condutividade associados a nuvens eletrificadas na região da América do Sul**. São José dos Campos. 129p. (INPE-4487-TDL/325). Tese (Doutoramento em Geofísica Espacial) - Instituto Nacional de Pesquisas Espaciais, 1987.
- Q. He, L. Wang, **An effective co-evolutionary particle swarm optimization for constrained engineering design problems**, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
- Raju & O'brien (2008) IS Raju e TK O'brien. **Fracture mechanics concepts, stress fields, strain energy release rates, delamination initiation and growth criteria**. *Delamination behaviour of composites*, páginas 3–27.
- Rao & Anandakumar (2007) A Rama Mohan Rao e Ganesh Anandakumar. **Optimal placement of sensors for structural system identification and health monitoring using a hybrid swarm intelligence technique**. *Smart materials and Structures*, 16(6):2658.
- Rao & Rao (2009). Singiresu S Rao e Singiresu S Rao. **Engineering optimization: theory and practice**. John Wiley & Sons.
- Rao & Yap (2011) Singiresu S Rao e Fook Fah Yap. **Mechanical vibrations**, volume 4. Prentice Hall Upper Saddle River.
- Rashedi E, Nezamabadi-Pour H, Saryazdi S. **GSA: a gravitational search algorithm**. *Information sciences*. (2009) 179 (13):2232-2248 27. Zhao W, Wang L, Zhang
- R7 Notícias. **As “tatuagens” impressionantes na pele de atingidos por raios**. Acesso em Outubro de 2019. Disponível em < <https://noticias.r7.com/hora-7/fotos/as-tatuagens-impressionantes-na-pele-de-atingidos-por-raios-19082018#!/foto/11>>.
- Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). **Lightning search algorithm**. *Applied Soft Computing*, 36, 315–333. doi:10.1016/j.asoc.2015.07.028
- Storn R, Price K. **Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces**. *J Global Optimization* 1997; 11:341–59.
- Turner, A. **From Lichtenberg to Lightning: Understanding Random Growth**. *Newsletter of the London Mathematical Society*, 2019.



Vo-Duy *et al.*(2016) Trung Vo-Duy, Vinh Ho-Huu, Hau Dang-Trung, Du Dinh-Cong, e Trung Nguyen-Thoi. **Damage detection in laminated composite plates using modal strain energy and improved differential evolution algorithm.** *Procedia Engineering*, 142:182–189.

WIKIPEDIA. **Relâmpago.** Disponível em: <https://pt.wikipedia.org/wiki/Rel%C3%A2mpago>;  
Acesso em 3 de Novembro de 2019.

Witten, T. A., Sander, L. M. **Diffusion-limited aggregation: a kinetic critical phenomenon,** *Phys. Rev. Lett.* 47, 1981. p. 1400-1403.

Witten, T. A., Sander, L. M. **Diffusion-limited aggregation,** *Phys. Rev. Lett.* 27, 1983. p. 5687-5697.

Worden *et al.*(2007) Keith Worden, Charles R Farrar, Graeme Manson, e Gyuhae Park. **The fundamental axioms of structural health monitoring.** Em *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 463, páginas 1639–1664. The Royal Society.

Worden & Friswell (2009) Keith Worden e Michael I Friswell. **Modal–vibration-based damage identification.** *Encyclopedia of Structural Health Monitoring.*

Yam, L. H., Wei, Z., & Cheng, L. (2004). **Nondestructive Detection of Internal Delamination by Vibration-based Method for Composite Plates.** *Journal of Composite Materials*, 38(24), 2183–2198.

Yang, Xin-She. **Nature-inspired Metaheuristic Algorithms,** Second Edition, Luniver Press, (2010)

Yang, Xin-She. **Nature-inspired optimization algorithms.** Elsevier, 2014.

# **Apêndice A**

## **Programa LA**

Programa completo do LA para aplicação simples em funções de teste, sem restrições e, portanto inapto para problemas reais. Para aplicações específicas entrar em contato com o autor.

-----MAIN-----

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ALGORITMO DE LICHTENBERG (LA) - ALGORITMO PARA OTIMIZAÇÃO NÃO-LINEAR E
%   SEM RESTRIÇÕES
%
%
% AUTORES: João Luiz Junho Pereira, Guilherme Ferreira Gomes and Sebastião
% Simões;
%
% Algoritmo de otimização inspirado em Difusão Limitada por Agregação
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc
format long
tic

set(0,'DefaultAxesFontName', 'Times New Roman')
set(0,'DefaultAxesFontSize', 14)
set(0,'DefaultTextFontname', 'Times New Roman')
set(0,'DefaultTextFontSize', 26)

UB = [5 5 5];           %Limite Superior
LB = [-5 -5 -5];       %limite Inferior
d = length(LB);        %dimensão do problema
pop = 10*d;            %População

% Parâmetros do Otimizador
n_iter = 100;          %Número máximo de iterações
Np = 10000;           %Número de Partículas (se 3D, melhor mais que 10000)
S_c = 1;              %Coeficiente de aderência: entre 0 e 1.
                          Próximo de 0 tem mais agregação e a densidade é maior.
Rc = 50;              %Raio de Criação
refinement = 0.3;     %Escala da 2ª figura em relação à primeira
M = 0;                %M = 0 ou 1. 0 é mesma figura a cada iteração;
x0 = (LB+UB)/2;       %Ponto Inicial

%FUNÇÕES DE TESTE

%Uma variável
Fun1 = @(x) (x(1)-1)^2; % (1)
Fun2 = @(x) x(1)^2;    % (0)

%Duas variáveis (Apenas f1 E f2, as demais não estão neste programa)
Fun3 = @(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
% rosenbrok (1,1)
Fun4 = @(x) x(1)^2 +x(2)^2;

```

```

%Três Variáveis
Fun5 = @(x) x(1)^2 +x(2)^2 +x(3)^2;
%Schwefel (0,0,0)
Fun6 = @(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2 + 100*(x(3)-x(2)^2)^2+(1-x(2))^2;
% rosenbrok (1,1,1)

%Quatro Variáveis
Fun7 = @(x) x(1)^2 +x(2)^2 +x(3)^2 +x(4)^2;
%Schwefel (0,0,0,0)
Fun8 = @(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2 + 100*(x(3)-x(2)^2)^2+(1-x(2))^2
+ 100*(x(4)-x(3)^2)^2+(1-x(3))^2; % rosenbrok (1,1,1,1)

%Cinco variáveis
Fun9 = @(x) x(1)^2 +x(2)^2 +x(3)^2 +x(4)^2 +x(5)^2;
%Schwefel (0,0,0,0,0)
Fun10 = @(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2 + 100*(x(3)-x(2)^2)^2+(1-x(2))^2
+ 100*(x(4)-x(3)^2)^2+(1-x(3))^2 + 100*(x(5)-x(4)^2)^2+(1-x(4))^2; %
rosenbrok (1,1,1,1,1)

Fun = Fun3; %Escolha a Função a ser otimizada, ajuste UP e LB!

[x,fval,iter,state,population]=LA_optimization(Fun,d,pop,x0,Np,S_c,Rc,M,LB,UB
,refinement,n_iter);

toc

```

### -----LA\_optimization-----

```

function
[best,fmin,n_iter,state,population]=LA_optimization(Fun,d,n,x0,Np,S_c,Rc,M,LB
,UB,refinement,n_iter)

for i=1:n
    Individuals(i,:)=LB+(UB-LB).*rand(1,d);
    Fitness(i)=Fun(Individuals(i,:));
end

[fmin,I]=min(Fitness);
best=Individuals(I,:);
S=Individuals;

if M==0
LA_data = LA_figure(x0,Np,S_c,Rc,d);
end

if d == 2 % Figura 2D - 2 variáveis
for t=1:n_iter

    x_start = [best(1,1) best(1,2)];
    X=[];

    % Chamada para criação da LF
    if M == 1
        LA_data = LA_figure(x_start,Np,S_c,Rc, d);
    end
end

```

```

end

scale_factor = rand;
X = LA_points(LA_data, LB, UB, Rc, x_start, scale_factor, d);

if refinement ~=0
    X_local =
LA_points(LA_data, refinement*LB, refinement*UB, Rc, x_start, scale_factor, d);

end

for i=1:n

    if refinement ~=0
        pop1 = round((0.4)*n); %População Local
        pop2 = n-pop1;        %População Global
        S_global = X(randperm(length(X), pop2), :);
        S_ref = X_local(randperm(length(X), pop1), :);
        S = [S_global; S_ref];
    else
        S = X(randperm(length(X), n), :);
    end

    S(i, :) = bound_check(S(i, :), LB, UB);

    Fnew = Fun(S(i, :));
    if (Fnew <= Fitness(i))
        Individuals(i, :) = S(i, :);
        Fitness(i) = Fnew;
    end
    if Fnew <= fmin
        best = S(i, :);
        fmin = Fnew;
    end
end

state(t, :) = [n_iter best fmin];
population{t} = S;

if refinement ~=0
    figure(1) %Mostra passo a passo da otimização
    set(gcf, 'color', 'w');
    h(1)=plot(best(1,1), best(1,2), 'MarkerFaceColor', [1 1
0], 'MarkerSize', 12, 'Marker', 'pentagram', 'LineWidth', 0.2,
'LineStyle', 'none', 'Color', [0 0 0]);
    hold on
    h(2)=plot(S(2:end,1), S(2:end,2), 'MarkerFaceColor', [0 0
0], 'MarkerSize', 8, 'Marker', 'o', 'LineStyle', 'none', 'Color', [0 0 0]);
    h(3)=plot(X_local(:,1), X_local(:,2), 'MarkerFaceColor', [1 0
0], 'MarkerSize', 2, 'Marker', '*', 'LineStyle', 'none', 'Color', [1 0 0]);
    h(4)=plot(X(:,1), X(:,2), 'MarkerFaceColor', [0 0
1], 'MarkerSize', 2, 'Marker', '*', 'LineStyle', 'none', 'Color', [0 0 1]);
    axis([LB(1) UB(1) LB(2) UB(2)]);
    xlabel('x_1')
    ylabel('x_2')
    axis square
    uistack(h(1), 'top')
    legend('Population', 'Lichtenberg Local', 'Lichtenberg
Global', 'Best', 'Location', 'NorthEastOutside')
    hold off
end

```

```

else

    figure(1)          %mostra passo a passo da otimização se ref = 0;
    set(gcf, 'color', 'w');
    h(1)=plot(best(1,1),best(1,2), 'MarkerFaceColor', [1 0
0], 'MarkerSize', 12, 'Marker', 'pentagram', 'LineStyle', 'none', 'Color', [1 0 0]);
    hold on
    h(2)=plot(S(2:end,1),S(2:end,2), 'MarkerFaceColor', [0 0
0], 'MarkerSize', 8, 'Marker', 'o', 'LineStyle', 'none', 'Color', [0 0 0]);
    h(3)=plot(X(:,1),X(:,2), 'MarkerFaceColor', [0 0
1], 'MarkerSize', 2, 'Marker', '*', 'LineStyle', 'none', 'Color', [0 0 1]);
    axis([LB(1) UB(1) LB(2) UB(2)]);
    xlabel('x_1')
    ylabel('x_2')
    axis square
    uistack(h(1), 'top')
    legend('Population', 'Lichtenberg Local', 'Lichtenberg
Global', 'Location', 'NorthEastOutside')
    hold off

end

pause(0.001)

X = sprintf('Iteration %0.f  x1 = %.4f x2 = %.4f
function=%.4f\n', t, best(1), best(2), fmin);
disp(X);

if round(t/100) == t/100
    best
    fmin
end
end
disp(['Total number of function evaluation: ', num2str(n_iter)]);
disp(['Var 1 : ', num2str(best(1))]);
disp(['Var 2 : ', num2str(best(2))]);
disp(['Best Fitness : ', num2str(fmin)]);
end

if d == 3 %para 3D - 3 variáveis
    for t=1:n_iter

        x_start = [best(1,1) best(1,2) best(1,3)];
        X=[];

        if M == 1 %Uma figura para cada iteração
            LA_data = LA_figure(x_start, Np, S_c, Rc, d);
        end

        scale_factor = rand;
        X = LA_points(LA_data, LB, UB, Rc, x_start, scale_factor, d);

        if refinement ~=0
            X_local =
            LA_points(LA_data, refinement*LB, refinement*UB, Rc, x_start, scale_factor, d);
        end
    end
end

```

```

for i=1:n

    if refinement ~=0
        pop1 = round((0.4)*n); %População Local
        pop2 = n-pop1;        %População Global
        S_global = X(randperm(length(X),pop2),:);
        S_ref = X_local(randperm(length(X),pop1),:);
        S = [S_global;S_ref];
    else
        S = X(randperm(length(X),n),:);
    end

    S(i,:)=bound_check(S(i,:),LB,UB);

    Fnew = Fun(S(i,:));
    if (Fnew <= Fitness(i))
        Individuals(i,:) = S(i,:);
        Fitness(i) = Fnew;
    end
    if Fnew <= fmin
        best = S(i,:);
        fmin = Fnew;
    end
end

state(t,:) = [n_iter best fmin];
population{t} = S;

if refinement ~=0
figure(1)
set(gcf,'color','w');
h(1)=plot3(best(1,1),best(1,2),best(1,3),'MarkerFaceColor',[1 1
0], 'MarkerSize',12, 'Marker', 'pentagram', 'LineWidth', 0.2,
'LineStyle', 'none', 'Color',[0 0 0]);
hold on
h(2)=plot3(S(2:end,1),S(2:end,2),S(2:end,3), 'MarkerFaceColor',[0 0
0], 'MarkerSize',8, 'Marker', 'o', 'LineStyle', 'none', 'Color',[0 0 0]);

h(3)=plot3(X_local(:,1),X_local(:,2),X_local(:,3), 'MarkerFaceColor',[1 0
0], 'MarkerSize',2, 'Marker', '*', 'LineStyle', 'none', 'Color',[1 0 0]);
h(4)=plot3(X(:,1),X(:,2),X(:,3), 'MarkerFaceColor',[0 0
1], 'MarkerSize',2, 'Marker', '*', 'LineStyle', 'none', 'Color',[0 0 1]);
axis([LB(1) UB(1) LB(2) UB(2) LB(3) UB(3)]);
xlabel('x_1')
ylabel('x_2')
zlabel('x_3')
axis square
uistack(h(1), 'top')
legend('Population', 'Lichtenberg Local', 'Lichtenberg
Global', 'Best', 'Location', 'NorthEastOutside')
hold off

else

figure(1)
set(gcf,'color','w');
h(1)=plot3(best(1,1),best(1,2),best(1,3), 'MarkerFaceColor',[1 0
0], 'MarkerSize',12, 'Marker', 'pentagram', 'LineStyle', 'none', 'Color',[1 0 0]);
hold on

```

```

        h(2)=plot3(S(2:end,1),S(2:end,2),S(2:end,3), 'MarkerFaceColor',[0 0
0], 'MarkerSize',8, 'Marker','o', 'LineStyle','none', 'Color',[0 0 0]);
        h(3)=plot3(X(:,1),X(:,2),X(:,3), 'MarkerFaceColor',[0 0
1], 'MarkerSize',2, 'Marker','*', 'LineStyle','none', 'Color',[0 0 1]);
        axis([LB(1) UB(1) LB(2) UB(2) LB(3) UB(3)]);
        xlabel('x_1')
        ylabel('x_2')
        zlabel('x_3')
        axis square
        uistack(h(1), 'top')
        legend('Population', 'Lichtenberg Local', 'Lichtenberg
Global', 'Location', 'NorthEastOutside')
        hold off

    end

    pause(0.001)

    X = sprintf('Iteration %0.f  x1 = %.4f x2 = %.4f x3 = %.4f
function=%.4f\n',t,best(1),best(2),best(3),fmin);
    disp(X);

    if round(t/100) == t/100
        best
        fmin
    end
end
disp(['Total number of function evaluation: ',num2str(n_iter)]);
disp(['Var 1 : ',num2str(best(1))]);
disp(['Var 2 : ',num2str(best(2))]);
disp(['Var 3 : ',num2str(best(3))]);
disp(['Best Fitness : ',num2str(fmin)]);
end

if d == 1 | d > 3 %Generalização para n variáveis
for t=1:n_iter

    x_start = best;
    X=[];

    if M == 1
        LA_data = LA_figure(x_start,Np,S_c,Rc, d);
    end

    scale_factor = rand;
    X = LA_points(LA_data,LB,UB,Rc,x_start,scale_factor,d);

    if refinement ~=0
        X_local =
LA_points(LA_data,refinement*LB,refinement*UB,Rc,x_start,scale_factor,d);
    end

    for i=1:n

        if refinement ~=0
            pop1 = round((0.4)*n);
            pop2 = n-pop1;
            S_global = X(randperm(length(X),pop2),:);
            S_ref = X_local(randperm(length(X),pop1),:);

```



```

        S = [S_global;S_ref];
    else
        S = X(randperm(length(X),n),:);
    end

    S(i,:)=bound_check(S(i,:),LB,UB);

    Fnew = Fun(S(i,:));
    if (Fnew <= Fitness(i))
        Individuals(i,:) = S(i,:);
        Fitness(i) = Fnew;
    end
    if Fnew <= fmin
        best = S(i,:);
        fmin = Fnew;
    end
end

state(t,:) = [n_iter best fmin];
population{t} = S;

X = sprintf('Iteration %0.f function=%0.4f\n',t,fmin);
disp(X);
disp(['Best Solution at each iteration: ', num2str(best)]);

if round(t/100) == t/100
    best;
    fmin;
end
end
disp(['Total number of function evaluation: ',num2str(n_iter)]);
disp(['Best solution (X) = ',num2str(best)]);
disp(['Best Fitness : ',num2str(fmin)]);
end

function s = bound_check(s,LB,UB)
ns_tmp=s;
I=ns_tmp<LB;
ns_tmp(I)=LB(I);

J=ns_tmp>UB;
ns_tmp(J)=UB(J);
s=ns_tmp;

-----LA_points-----

function [X]=LA_points(map,LB,UB,Rc,x0,scale_factor,d)
Rk = Rc*1.5;
add = Rk+2;

%***** Resultados do Fractal*****
if d == 2 %2D
    %Número de Patículas no Aglomerado
    Ncluster = sum(sum(map));
    %Raio do Aglomerado
    rcluster=0;
    i=0;

```

```

j=0;
for i=1:((Rk+2)*2)
    for j=1:((Rk+2)*2)
        if map(i,j)==1
            if (hypot(i-add,j-add)>rcluster)
                rcluster=hypot(i-add,j-add);
            end
        end
    end
end

%Dimensão do Fractal
Dfractal = log(Ncluster)/log(rcluster);
end

if d == 3 %3D
    %Número de Partículas no Aglomerado
    Ncluster = sum(sum(sum(map)));
    %Raio do cluster
    rcluster=0;
    i=0;
    j=0;
    k=0;
    for i=1:(Rk+2)*2
        for j=1:(Rk+2)*2
            for k=1:(Rk+2)*2
                if map(i,j,k)==1
                    if (sqrt((i-add)^2+(j-add)^2+(k-add)^2)>rcluster)
                        rcluster=sqrt((i-add)^2+(j-add)^2+(k-add)^2);
                    end
                end
            end
        end
    end
end
rcluster;

%Dimensão do Fractal
Dfractal = log(Ncluster)/log(rcluster);
end

%*****Bitmap para cartesianas*****
A=map;
if d < 3 | d > 3 %todos menos 3D
    [xp,yp] = ind2sub(size(map),find(map == 1));
    K = [xp yp];
    teta = rand; %rotação
    for u =1:length(K)
        K(u,1)= K(u,1)*cos(teta)-K(u,2)*sin(teta);
        K(u,2)= K(u,1)*sin(teta)+K(u,2)*cos(teta);
    end
end

if d == 3 %3D
    [xp,yp,zp] = ind2sub(size(map),find(map == 1));
    K = [xp yp zp];
    teta = rand; %rotação
    alfa = rand;
    beta = rand;
    for u =1:length(K)

```

```

    K(u,1)=
K(u,1)*cos(teta)*cos(beta)+K(u,2)*(cos(alfa)*sin(beta)+sin(alfa)*sin(teta)*co
s(beta))+K(u,3)*(sin(alfa)*sin(beta)-cos(alfa)*sin(teta)*cos(beta));
    K(u,2)= K(u,1)*(-cos(teta))*sin(beta)+K(u,2)*(cos(alfa)*cos(beta)-
sin(alfa)*sin(teta)*sin(beta))+K(u,3)*(sin(alfa)*cos(beta)+cos(alfa)*sin(teta
)*sin(beta));
    K(u,3)= K(u,1)*sin(teta)+K(u,2)*(-
sin(alfa)*cos(teta))+K(u,3)*cos(alfa)*cos(teta);
    end
end

T = length(K);

Ki = zeros(T,d);
if d==1
    Ki(:,1) = K(:,1);
end

if d==2 | d ==3
    for i=1:d
        Ki(:,i) = K(:,i);
    end
end

if d > 3 %rotacionar projeções da figura de lichtenberg
    if mod(d, 2) == 0 %se d é par, de 4 a infinito
        for i=1:2:d
            gama = rand;
            Ki(:,i) = K(:,1)*cos(gama)-K(:,2)*sin(gama);
            Ki(:,i+1) = K(:,1)*sin(gama)+K(:,2)*cos(teta);
        end
    else %se d é impar, de 4 a infinito.
        for i=1:2:d
            gama = rand;
            Ki(:,i) = K(:,1)*cos(gama)-K(:,2)*sin(gama);
            Ki(:,i+1) = K(:,1)*sin(gama)+K(:,2)*cos(teta);
        end
        Ki=Ki(:,1:(end-1)); %exclui coluna excedente caso aconteça
    end
end

%Escala Variável e Randômica
for i = 1:d
    scale(i) = scale_factor*(UB(i)-LB(i))/(max(Ki(:,i))-min(Ki(:,i)));
    Ki(:,i) = scale(i)*Ki(:,i);
end

%Fixando o aglomerado no centro
for i =1:d
    %Center cluster position and delta!
    Pcc(i)=(max(Ki(:,i))-min(Ki(:,i)))/2 + min(Ki(:,i));
    Ki(round(T/2),i)= Pcc(i);
    delta(i)=Pcc(i)-x0(i);
end

X=zeros(size(Ki));
for i=1:d
    X(:,i) = Ki(:,i) - delta(i);
end

```

```

%Raio do aglomerado com a escala aplicada, caso queira.
if d == 2 | d==3
    Rc=0;
for i=1:d
    Rc=Rc+(scale(i))^2;
end
Rcluster=(sqrt(Rc))/2*rcluster;
end
return

```

### -----LA\_figure-----

```

function [map]=LA_figure(x0,Np,S,Rc,d)
%Radius Kill = Radius from which the partile is killed
Rk = Rc*1.1;
%Last region to build the cluster (1+Rk+1)
add = Rk+2;

%***** Interactions *****
particle = 0;
%Ação da partícula: Agregar ou escapar; Caminhar ou morrer;
stuck = 0;
escape = 0;
die = 0;
walk = 0;

if d<3 | d>3
%mapa = se 1 é partícula do aglomerado.
map = zeros(((Rk+2)*2));
%Partícula fixada no centro
map(add,add)=1;
    %Outras variáveis
    X = 0; x = 0;
    Y = 0; y = 0;

    %Execução das Iterações
    while (( particle >= Np) + (escape))==0
        particle=particle+1;
        % randomização
        phi=rand*2*3.14159265359;
        X=Rc*cos(phi);
        Y=Rc*sin(phi);
        % Arredondamento para inteiros
        x=round(X);
        y=round(Y);
        stuck = 0;
        die = 0;
        while ((stuck+die+escape) == 0)
            % Partícula está caminhando
            walk=rand;
            if walk<.25          %Cima
                if map(add+x,add+y+1)==0
                    y=y+1;
                end
            elseif walk<.5      %Direita

```

```

        if map(add+x+1,add+y)==0
            x=x+1;
        end
elseif walk<.75    %Baixo
    if map(add+x,add+y-1)==0
        y=y-1;
    end
else                %Esquerda
    if map(add+x-1,add+y)==0
        x=x-1;
    end
end
%Checando se ultrapassou os limites e eliminá-la se sim
if (hypot(x,y)>=Rk)
    die=1;
else
    % Se não a partícula pode se agregar ao aglomerado a partir de uma
%checagem adjacente
    stuck=0;
    if (map(add+x+1,add+y) + map(add+x-1,add+y) + map(add+x,add+y-1)
+ map(add+x,add+y+1))~=0
        % Se sim, ela pode aglomerar. Mas há o coeficiente de
%aderência ainda!
        if (rand<S)
            stuck=1;
        end
    end
end
end
%A partícula se fixou no mapa
if stuck
    map(add+x,add+y)=1;
    stuck=0;
    clf
    %    figure(1)
    %    imshow(imresize(map,1));
    % Se o aglomerado atingiu o limite, a LF para de ser criada
    if ((hypot(x,y)*1.2)>=Rc)
        escape = 1;
    end
end %finalização para cada partícula

end %Iterações estão finalizadas
%aviso se a lf parou de ser criado por ter atingido o raio de criação
if (escape==1)
    disp('The simulation ended before all particle could be tried because
boundaries were exceeded');
end
%fecha tudo
return
end

% mesmas coisas para 3D
if d==3
map = zeros((2*Rk+4),(2*Rk+4),(2*Rk+4));
map(add,add,add)=1;
X = 0; x = 0;
Y = 0; y = 0;
Z = 0; z= 0;

while ((particle >= Np) + (escape))==0

```

```

particle=particle+1;
alfa=rand*2*pi;
beta=rand*2*pi;
X=Rc*sin(alfa)*cos(beta);
Y=Rc*sin(alfa)*sin(beta);
Z=Rc*cos(alfa);
x=round(X);
y=round(Y);
z=round(Z);
stuck = 0;
die = 0;
while ((stuck+die+escape) == 0)
walk = 1.5*rand;

if walk<.25
    if map(add+x,add+y+1, add+z)==0
        y=y+1;
    end

elseif walk<.5
    if map(add+x+1,add+y, add+z)==0
        x=x+1;
    end

elseif walk<.75
    if map(add+x,add+y-1, add+z)==0
        y=y-1;
    end

elseif walk<1
    if map(add+x-1,add+y, add+z)==0
        x=x-1;
    end

elseif walk<1.25
    if map(add+x,add+y, add+z+1)==0
        z=z+1;
    end

else
    if map(add+x, add+y, add+z-1)==0
        z=z-1;
    end
end
if (sqrt(abs(x).^2+abs(y).^2+abs(z).^2)>=Rk)
die=1;
else
stuck=0;
    if (map(add+x+1,add+y, add+z) + map(add+x-1,add+y, add+z) +
map(add+x,add+y+1, add+z) + map(add+x,add+y-1,add+z) +
map(add+x,add+y,add+z+1) + map(add+x,add+y,add+z-1))~=0
        if (rand<S)
            stuck=1;
        end
    end
end
end
if stuck
    map(add+x,add+y,add+z)=1;
    stuck=0;
end

```

```
clf
%imshow(imresize(map,1));
if (sqrt(abs(x).^2+abs(y).^2+abs(z).^2)*1.2>=Rc)
    escape = 1;
end
end
end
if (escape==1)
    disp('The simulation ended before all particle could be tried because
boundaries were exceeded');
end
end
```

## **Apêndice B**

### **Resultados LA sem Restrição**



Resultados para 10 simulações para as funções de Rosenbrock e Schwefel para uma, três, quatro e cinco variáveis. Para todas as bidimensionais, o resultado da média e o desvio padrão foram obtidos diretamente e automaticamente pelo LA considerando 30 simulações.

**Tabela B-1** - Simulações de validação para uma variável com  $f_1$  e  $f_2$

Alvo	Pop	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$	Ponto Otimizado	$f_{min}$
0	10	100	100000	1	150	0.2	0	<b>0.0000</b>	0.0000
0	10	100	100000	1	150	0.2	0	<b>0.0001</b>	0.0002
0	10	100	100000	1	150	0.2	0	<b>-0.0002</b>	0.0000
0	10	100	100000	1	150	0.2	0	<b>-0.0004</b>	0.0000
0	10	100	100000	1	150	0.2	0	<b>0.0002</b>	0.0002
0	10	100	100000	1	150	0.2	0	<b>-0.0003</b>	0.0008
0	10	100	100000	1	150	0.2	0	<b>0.8378</b>	0.0001
0	10	100	100000	1	150	0.2	0	<b>-0.0003</b>	0.0007
0	10	100	100000	1	150	0.2	0	<b>0.0002</b>	0.0005
0	10	100	100000	1	150	0.2	0	<b>0.0002</b>	0.0005
Média								<b>0.0001</b>	<b>0.0002</b>
Des. Pad								<b>0,2649347</b>	<b>0,000299</b>
1	10	100	100000	1	150	0.2	0	<b>0.9997</b>	0.0008
1	10	100	100000	1	150	0.2	0	<b>1.0002</b>	0.0006
1	10	100	100000	1	150	0.2	0	<b>1.0000</b>	0.0000
1	10	100	100000	1	150	0.2	0	<b>1.0003</b>	0.0000
1	10	100	100000	1	150	0.2	0	<b>1.0001</b>	0.0000
1	10	100	100000	1	150	0.2	0	<b>1.0002</b>	0.0004
1	10	100	100000	1	150	0.2	0	<b>1.0000</b>	0.0000
1	10	100	100000	1	150	0.2	0	<b>0.9997</b>	0.0007
1	10	100	100000	1	150	0.2	0	<b>0.9999</b>	0.0000
1	10	100	100000	1	150	0.2	0	<b>1.0000</b>	0.0000
Média								<b>1.0000</b>	<b>0.0000</b>
Des. Pad								<b>0.0001931</b>	<b>0.0003385</b>

**Tabela B-2** - Simulações de validação para três variáveis com  $f_1$  e  $f_2$ 

Alvos	Pop	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$	Pontos Otimos			$f_{min}$
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>- 0.0020</b>	<b>- 0.0025</b>	<b>0.0003</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0007</b>	<b>- 0.0005</b>	<b>0.0001</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0101</b>	<b>0.0032</b>	<b>0.0002</b>	0.0001
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0198</b>	<b>0.0185</b>	<b>0.0453</b>	0.0028
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>- 0.0020</b>	<b>- 0.0025</b>	<b>0.0003</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0029</b>	<b>0.0022</b>	<b>0.0025</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>- 0.0046</b>	<b>0.0018</b>	<b>0.0014</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0019</b>	<b>0.0013</b>	<b>0.0021</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0013</b>	<b>- 0.0005</b>	<b>0.0006</b>	0.0000
(0, 0, 0)	60	100	1000000	1	150	0.4	0	<b>0.0007</b>	<b>- 0.0005</b>	<b>0.0001</b>	0.0000
Média								<b>0.0010</b>	<b>0.0004</b>	<b>0.0003</b>	<b>0.0000</b>
Des. Pad								<b>0.0071</b>	<b>0.0061</b>	<b>0.0146</b>	<b>0.0009</b>
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>1.0184</b>	<b>1.0138</b>	<b>1.0293</b>	0.0552
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9946</b>	<b>0.9899</b>	<b>0.9806</b>	0.0002
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9996</b>	<b>1.0002</b>	<b>1.0001</b>	0.0001
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9991</b>	<b>1.0013</b>	<b>1.0002</b>	0.0017
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9958</b>	<b>0.9896</b>	<b>0.9786</b>	0.0002
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9670</b>	<b>0.9340</b>	<b>0.8740</b>	0.0059
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>1.0062</b>	<b>1.0132</b>	<b>1.0206</b>	0.0039
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9807</b>	<b>0.9578</b>	<b>0.9187</b>	0.0039
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9637</b>	<b>0.9250</b>	<b>0.8514</b>	0.0101
(1, 1, 1)	60	100	1000000	1	150	0.4	0	<b>0.9894</b>	<b>0.9777</b>	<b>0.9592</b>	0.0018
Média								<b>0.9952</b>	<b>0.9898</b>	<b>0.9796</b>	<b>0.0039</b>
Des. Pad								<b>0.0170</b>	<b>0.0315</b>	<b>0.0609</b>	<b>0.0176</b>

**Tabela B-3** - Simulações de Validação para quatro variáveis com  $f_1$  e  $f_2$

Alvos	Pop	$N_{iter}$	$N_p$	$S$	$R_c$	ref	$M$	Pontos	Otimos	$f_{min}$		
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>-0,0850</b>	<b>0,0785</b>	<b>0,1695</b>	<b>-0,0886</b>	0,0500
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>0,0213</b>	<b>0,0414</b>	<b>-0,0102</b>	<b>-0,0394</b>	0,0038
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>0,0828</b>	<b>-0,1924</b>	<b>-0,0366</b>	<b>0,1277</b>	0,0615
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>-0,0198</b>	<b>0,0065</b>	<b>0,0350</b>	<b>-0,0190</b>	0,0020
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>0,1488</b>	<b>-0,0391</b>	<b>-0,1148</b>	<b>-0,0773</b>	0,0428
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>-0,0397</b>	<b>0,1388</b>	<b>0,0768</b>	<b>-0,1450</b>	0,0478
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>0,0222</b>	<b>-0,0707</b>	<b>-0,0587</b>	<b>0,0587</b>	0,0124
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>-0,0034</b>	<b>0,0700</b>	<b>0,0245</b>	<b>-0,0625</b>	0,0094
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>-0,0268</b>	<b>-0,0008</b>	<b>0,0146</b>	<b>-0,0021</b>	0,0009
(0, 0, 0, 0)	40	100	100000	1	150	0.2	0	<b>0,0243</b>	<b>0,0586</b>	<b>-0,0173</b>	<b>-0,0947</b>	0,0133
							Média	<b>0,0089</b>	<b>0,0240</b>	<b>0,0022</b>	<b>-0,0509</b>	<b>0,0128</b>
							Des. Pad	<b>0,0657</b>	<b>0,0932</b>	<b>0,0777</b>	<b>0,0801</b>	<b>0,0233</b>
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>1,0016</b>	<b>1,0033</b>	<b>1,0066</b>	<b>1,0133</b>	0,0001
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>1,0148</b>	<b>1,0294</b>	<b>1,0602</b>	<b>1,1247</b>	0,0048
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9533</b>	<b>0,9188</b>	<b>0,8510</b>	<b>0,7279</b>	0,8602
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>1,0244</b>	<b>1,0470</b>	<b>1,0944</b>	<b>1,2010</b>	0,0137
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9798</b>	<b>0,9596</b>	<b>0,9206</b>	<b>0,8467</b>	0,0084
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9826</b>	<b>0,9646</b>	<b>0,9303</b>	<b>0,8656</b>	0,0065
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9663</b>	<b>0,9350</b>	<b>0,8723</b>	<b>0,7580</b>	0,0230
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>1,0232</b>	<b>1,0492</b>	<b>1,1021</b>	<b>1,2142</b>	0,0141
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9951</b>	<b>1,0010</b>	<b>1,0075</b>	<b>1,0172</b>	0,0001
(1, 1, 1, 1)	40	100	100000	1	150	0.2	0	<b>0,9522</b>	<b>0,9048</b>	<b>0,8177</b>	<b>0,66602</b>	0,0457
							Média	<b>0,9888</b>	<b>0,9828</b>	<b>0,9685</b>	<b>1,0133</b>	<b>0,0110</b>
							Des. Pad	<b>0,0270</b>	<b>0,0525</b>	<b>0,1025</b>	<b>0,1839</b>	<b>0,2683</b>

**Tabela B-4** - Simulações de Validação para cinco variáveis com  $f_1$  e  $f_2$ 

Alvos	Pop	$N_{iter}$	$N_p$	$S$	$R_c$	ref	$M$	Pontos Otimos					$f_{min}$
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0240</b>	<b>0.0015</b>	<b>0.0115</b>	<b>-0.0007</b>	<b>0.0151</b>	0.0009
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0043</b>	<b>-0.0018</b>	<b>0.0004</b>	<b>-0.0007</b>	<b>0.0031</b>	0.0000
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>0.0063</b>	<b>-0.0064</b>	<b>0.0085</b>	<b>0.0002</b>	<b>-0.0123</b>	0.0003
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0002</b>	<b>-0.0001</b>	<b>0.0050</b>	<b>0.0034</b>	<b>-0.0045</b>	0.0001
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>0.0200</b>	<b>-0.0269</b>	<b>-0.0210</b>	<b>0.0162</b>	<b>0.0100</b>	0.0019
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0219</b>	<b>0.0048</b>	<b>0.0163</b>	<b>0.0197</b>	<b>-0.0036</b>	0.0012
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>0.0393</b>	<b>-0.0401</b>	<b>0.0191</b>	<b>0.0322</b>	<b>-0.0317</b>	0.0056
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>0.0236</b>	<b>-0.0575</b>	<b>-0.0226</b>	<b>0.0639</b>	<b>0.0014</b>	0.0085
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0146</b>	<b>0.0253</b>	<b>0.0010</b>	<b>-0.0176</b>	<b>0.0185</b>	0.0015
(0, 0, 0, 0, 0)	50	100	100000	1	150	0.2	0	<b>-0.0056</b>	<b>0.0019</b>	<b>-0.0281</b>	<b>-0.0392</b>	<b>0.0604</b>	0.0060
							Média	<b>-0.0022</b>	<b>-0.0010</b>	<b>0.0030</b>	<b>0.0018</b>	<b>0.0022</b>	<b>0.0013</b>
							Des. Pad	<b>0.0206</b>	<b>0.0244</b>	<b>0.0170</b>	<b>0.0279</b>	<b>0.0240</b>	<b>0.0030</b>
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0132</b>	<b>1.0286</b>	<b>1.0634</b>	<b>1.1280</b>	<b>1.2699</b>	0.0262
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0025</b>	<b>1.0048</b>	<b>1.0088</b>	<b>1.0181</b>	<b>1.0374</b>	0.0006
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>0.9936</b>	<b>0.9923</b>	<b>0.9999</b>	<b>1.0029</b>	<b>1.0009</b>	0.0291
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0046</b>	<b>1.0070</b>	<b>1.0037</b>	<b>1.0114</b>	<b>1.0315</b>	0.0207
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0062</b>	<b>1.0012</b>	<b>1.0130</b>	<b>1.0118</b>	<b>1.0133</b>	0.0559
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>0.9586</b>	<b>0.9126</b>	<b>0.8329</b>	<b>0.6901</b>	<b>0.4758</b>	0.1387
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0037</b>	<b>1.0136</b>	<b>1.0244</b>	<b>1.0482</b>	<b>1.1073</b>	0.0311
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>0.9811</b>	<b>0.9563</b>	<b>0.9157</b>	<b>0.8398</b>	<b>0.6985</b>	0.0437
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0016</b>	<b>1.0029</b>	<b>1.0023</b>	<b>1.0047</b>	<b>1.0134</b>	0.0028
(1, 1, 1, 1, 1)	50	100	100000	1	150	0.2	0	<b>1.0115</b>	<b>1.0024</b>	<b>1.0013</b>	<b>1.0207</b>	<b>1.0485</b>	0.0714
							Média	<b>1.0031</b>	<b>1.0027</b>	<b>1.0030</b>	<b>1.0116</b>	<b>1.0225</b>	<b>0.0311</b>
							Des. Pad	<b>0.0165</b>	<b>0.0335</b>	<b>0.0651</b>	<b>0.1231</b>	<b>0.2226</b>	<b>0.0425</b>

## **Apêndice C**

### **Resultados LA Projeto - de Viga Soldada**

Resultados de todas as simulações do LA para as variáveis do problema teste

**Tabela C.1** - Problema Teste - Projeto Viga Soldada

Simulação	$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$h$	$l$	$t$	$b$	$f_{min}$
1	100	200	100000	1	150	0.2	<b>0.1480</b>	<b>5.3714</b>	<b>9.0359</b>	<b>0.2058</b>	1.8628
2	100	200	100000	1	150	0.2	<b>0.3058</b>	<b>2.5677</b>	<b>7.4111</b>	<b>0.3059</b>	2.0722
3	100	200	100000	1	150	0.2	<b>0.2945</b>	<b>2.6444</b>	<b>7.5406</b>	<b>0.2955</b>	2.0379
4	100	200	100000	1	150	0.2	<b>0.2061</b>	<b>3.4388</b>	<b>9.1015</b>	<b>0.2061</b>	<b>1.7351</b>
5	100	200	100000	1	150	0.2	<b>0.2068</b>	<b>3.5090</b>	<b>8.9253</b>	<b>0.2109</b>	1.7514
6	100	200	100000	1	150	0.2	<b>0.2669</b>	<b>2.8508</b>	<b>7.9387</b>	<b>0.2670</b>	1.9428
7	100	200	100000	1	150	0.2	<b>0.1994</b>	<b>3.3505</b>	<b>9.7602</b>	<b>0.2024</b>	1.7960
8	100	200	100000	1	150	0.2	<b>0.2689</b>	<b>2.8322</b>	<b>7.9021</b>	<b>0.2691</b>	1.9480
9	100	200	100000	1	150	0.2	<b>0.2852</b>	<b>2.7041</b>	<b>7.6757</b>	<b>0.2855</b>	2.0040
10	100	200	100000	1	150	0.2	<b>0.2114</b>	<b>3.4066</b>	<b>8.9104</b>	<b>0.2116</b>	1.7471
11	100	200	100000	1	150	0.2	<b>0.2297</b>	<b>3.2194</b>	<b>8.4936</b>	<b>0.2329</b>	1.8263
12	100	200	100000	1	150	0.2	<b>0.1896</b>	<b>3.8779</b>	<b>9.0065</b>	<b>0.2071</b>	1.7584
13	100	200	100000	1	150	0.2	<b>0.1777</b>	<b>4.2254</b>	<b>9.0195</b>	<b>0.2065</b>	1.7806
14	100	200	100000	1	150	0.2	<b>0.1897</b>	<b>3.8362</b>	<b>9.1030</b>	<b>0.2054</b>	1.7570
15	100	200	100000	1	150	0.2	<b>0.1415</b>	<b>6.1298</b>	<b>8.8766</b>	<b>0.2133</b>	1.9692
16	100	200	100000	1	150	0.2	<b>0.2616</b>	<b>2.4727</b>	<b>9.4493</b>	<b>0.2619</b>	2.1480
17	100	200	100000	1	150	0.2	<b>0.3390</b>	<b>2.3753</b>	<b>7.0317</b>	<b>0.3398</b>	2.1839
18	100	200	100000	1	150	0.2	<b>0.2549</b>	<b>2.9645</b>	<b>8.1012</b>	<b>0.2560</b>	1.9054
19	100	200	100000	1	150	0.2	<b>0.2665</b>	<b>2.8526</b>	<b>7.9287</b>	<b>0.2672</b>	1.9418
20	100	200	100000	1	150	0.2	<b>0.2206</b>	<b>3.2985</b>	<b>8.7118</b>	<b>0.2214</b>	1.7824
21	100	200	100000	1	150	0.2	<b>0.4154</b>	<b>2.0345</b>	<b>6.3597</b>	<b>0.4155</b>	2.4262
22	100	200	100000	1	150	0.2	<b>0.2603</b>	<b>2.9011</b>	<b>8.0337</b>	<b>0.2605</b>	1.9186
23	100	200	100000	1	150	0.2	<b>0.2428</b>	<b>3.0163</b>	<b>8.4255</b>	<b>0.2429</b>	1.8717
24	100	200	100000	1	150	0.2	<b>0.2070</b>	<b>3.4860</b>	<b>8.9308</b>	<b>0.2106</b>	1.7475
25	100	200	100000	1	150	0.2	<b>0.2168</b>	<b>3.3370</b>	<b>8.7898</b>	<b>0.2176</b>	1.7684
26	100	200	100000	1	150	0.2	<b>0.1888</b>	<b>3.8765</b>	<b>9.0381</b>	<b>0.2065</b>	1.7576
27	100	200	100000	1	150	0.2	<b>0.2219</b>	<b>3.2651</b>	<b>8.7259</b>	<b>0.2219</b>	1.7859
28	100	200	100000	1	150	0.2	<b>0.2260</b>	<b>3.2379</b>	<b>8.5980</b>	<b>0.2273</b>	1.8032
29	100	200	100000	1	150	0.2	<b>0.1336</b>	<b>6.1460</b>	<b>9.0726</b>	<b>0.2056</b>	1.9287
30	100	200	100000	1	150	0.2	<b>0.1631</b>	<b>4.7005</b>	<b>9.0413</b>	<b>0.2057</b>	1.8115
Média							<b>0.2213</b>	<b>3.2818</b>	<b>8.7579</b>	<b>0.2216</b>	<b>1.8446</b>
Des.Pad							<b>0.0603</b>	<b>0.9986</b>	<b>0.7591</b>	<b>0.0487</b>	<b>0.1597</b>

## **Apêndice D**

### **Resultados LA – Identificação de Danos**

## Resultados na Identificação de danos pelo LA

**Tabela D-1** – Simulações de Identificação de Delaminação – Dano induzido no elemento 19 e taxas de dano de 0.2, 0.5 e 0.9

$N_e$	$\alpha$	Sensores	$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$	Resultados	$J_{min}$
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.5178
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.0000
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.0000
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2001</b>	0.5178
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2002</b>	0.0002
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.0000
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2005</b>	0.0009
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.0000
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2000</b>	0.0000
19	0,2	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.2001</b>	0.5178
Média										<b>19 0.2000</b>	<b>0.0001</b>
Desvio										<b>0 0.0002</b>	<b>0.2372</b>
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.4997</b>	0.0002
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5000</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5000</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5000</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5000</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5000</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5001</b>	0.0001
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.4999</b>	0.0000
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.4997</b>	0.0002
19	0,5	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.5051</b>	0.0056
Média										<b>19 0.5000</b>	<b>0.0000</b>
Desvio										<b>0 0.0001</b>	<b>0.0017</b>
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9000</b>	0.0000
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.8900</b>	0.0002
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9077</b>	0.0007
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9000</b>	0.0000
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.8990</b>	0.0007
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9021</b>	0.0015
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9000</b>	0.0000
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9001</b>	0.0000
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9000</b>	0.0000
19	0,9	4(41,80,144,278)	20	100	1000000	1	150	0.2	0	<b>19 0.9000</b>	0.0000
Média										<b>19 0.9000</b>	<b>0.0000</b>
Desvio										<b>0 0.0041</b>	<b>0.0005</b>



**Tabela D-2** – Simulações de Identificação de Delaminação – Dano induzido nos elementos 19 e 65 com severidade de 0.2

$N_{e1}$	$N_{e2}$	$\alpha$	$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$	Resultados			$J_{min}$
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.2000</b>	0.0000
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.20009</b>	0.0014
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.2004</b>	0.0054
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.19999</b>	0.0000
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>66</b>	<b>0.3944</b>	0.7633
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>64</b>	<b>0.3683</b>	0.7001
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.2214</b>	0.5513
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>65</b>	<b>0.2482</b>	0.5634
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>64</b>	<b>0.3019</b>	0.6041
19	65	0,2	60	100	1000000	1	150	0.4	0	<b>19</b>	<b>64</b>	<b>0.2982</b>	0.2268
										<b>19</b>	<b>65</b>	<b>0.2732</b>	<b>0.3890</b>
										<b>0</b>	<b>0.6325</b>	<b>0.0745</b>	<b>0.3238</b>

**Tabela D-3** – Simulações de Identificação de Delaminação – Dano induzido no elemento 19 com severidade de 0.2 e adição de ruídos nos sinais

$N_e$	$\alpha$	Ruído (%)	$Pop$	$N_{iter}$	$N_p$	$S$	$R_c$	$ref$	$M$	Resultados	$J_{min}$
19	0,2	1	20	100	1000000	1	150	0.2	0	<b>19 0.1967</b>	0.2517
19	0,2	1	20	100	1000000	1	150	0.2	0	<b>19 0.1940</b>	0.1729
19	0,2	1	20	100	1000000	1	150	0.2	0	<b>19 0.2140</b>	0.1684
19	0,2	1	20	100	1000000	1	150	0.2	0	<b>19 0.2144</b>	0.2284
19	0,2	1	20	100	1000000	1	150	0.2	0	<b>19 0.1844</b>	0.2708
Média										<b>19 0.1967</b>	<b>0.2284</b>
Desvio										<b>0 0.0117</b>	<b>0.0413</b>
19	0,2	2	20	100	1000000	1	150	0.2	0	<b>19 0.1712</b>	0.2535
19	0,2	2	20	100	1000000	1	150	0.2	0	<b>19 0.1937</b>	0.3033
19	0,2	2	20	100	1000000	1	150	0.2	0	<b>19 0.1746</b>	0.3493
19	0,2	2	20	100	1000000	1	150	0.2	0	<b>19 0.2243</b>	0.4035
19	0,2	2	20	100	1000000	1	150	0.2	0	<b>19 0.2241</b>	0.4029
Média										<b>19 0.1937</b>	<b>0.3493</b>
Desvio										<b>0 0.0231</b>	<b>0.0582</b>
19	0,2	5	20	100	1000000	1	150	0.2	0	<b>19 0.2244</b>	0.8391
19	0,2	5	20	100	1000000	1	150	0.2	0	<b>19 0.1371</b>	0.6520
19	0,2	5	20	100	1000000	1	150	0.2	0	<b>19 0.2527</b>	0.6510
19	0,2	5	20	100	1000000	1	150	0.2	0	<b>19 0.2278</b>	0.8906
19	0,2	5	20	100	1000000	1	150	0.2	0	<b>19 0.2468</b>	0.9308
Média										<b>19 0.2278</b>	<b>0.8391</b>
Desvio										<b>0 0.0417</b>	<b>0.1189</b>
19	0,2	10	20	100	1000000	1	150	0.2	0	<b>19 0.1972</b>	1.3803
19	0,2	10	20	100	1000000	1	150	0.2	0	<b>19 0.1504</b>	1.9281
19	0,2	10	20	100	1000000	1	150	0.2	0	<b>19 0.2422</b>	1.6197
19	0,2	10	20	100	1000000	1	150	0.2	0	<b>19 0.1250</b>	1.5146
19	0,2	10	20	100	1000000	1	150	0.2	0	<b>19 0.1776</b>	01.7938
Média										<b>19 0.1776</b>	<b>1.6197</b>
Desvio										<b>0 0.0402</b>	<b>0.1949</b>

## **Anexo A**

### **Resultados LA – Registro no INPI**





**REPÚBLICA FEDERATIVA DO BRASIL**  
MINISTÉRIO DA ECONOMIA  
**INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL**  
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

## Certificado de Registro de Programa de Computador

**Processo Nº: BR512019002567-9**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 11/11/2019, em conformidade com o 52º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

**Título:** Lichtenberg Algorithm Optimization (LA)

**Data de publicação:** 11/11/2019

**Data de criação:** 11/10/2019

**Titular(es):** UNIFEI - UNIVERSIDADE FEDERAL DE ITAJUBÁ

**Autor(es):** GUILHERME FERREIRA GOMES; JOÃO LUIZ JUNHO PEREIRA; SEBASTIÃO SIMÕES DA CUNHA JUNIOR

**Linguagem:** MATLAB

**Campo de aplicação:** CC-05; IN-03; MT-01; MT-05

**Tipo de programa:** TC-01

**Algoritmo hash:** SHA-512

**Resumo digital hash:**  
cb6ac987024bf655c58221798abd7e106bcd02ba0ac9dd745fb2ce18f9227a8213d9639427e05f32d1410ea1cc569bbb310a40239634e77abd7e7ce5d604e184

**Expedido em:** 19/11/2019

**Aprovado por:**  
Helmar Alvares  
Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019

**Figura A-1** – Certificado de Registro do LA ao INPI