

UNIVERSIDADE FEDERAL DE UBERLÂNDIA - UFU
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FLORISVALDO CARDOZO BOMFIM JUNIOR

PROPOSTA DE METODOLOGIA PARA OTIMIZAÇÃO DE
DISPOSITIVOS MAGNÉTICOS POR MÉTODO HÍBRIDO APLICANDO
ELEMENTOS FINITOS E ALGORITMOS GENÉTICOS

Uberlândia-MG

2025

FLORISVALDO CARDOZO BOMFIM JUNIOR

**PROPOSTA DE METODOLOGIA PARA OTIMIZAÇÃO DE
DISPOSITIVOS MAGNÉTICOS POR MÉTODO HÍBRIDO
APLICANDO ELEMENTOS FINITOS E ALGORITMOS GENÉTICOS**

Tese apresentada ao Programa de Pós-Graduação
em Engenharia Elétrica da Universidade Federal de
Uberlândia, como requisito para a obtenção do título
de Doutor em Ciências.

Orientador: Prof. Dr. Keiji Yamanaka

Uberlândia-MG

2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

| | |
|--|--|
| B695 | Bomfim Junior, Florisvaldo Cardozo, 1980- |
| 2025 | Proposta de metodologia para otimização de dispositivos magnéticos por método híbrido aplicando elementos finitos e algoritmos genéticos [recurso eletrônico] / Florisvaldo Cardozo Bomfim Junior. - 2025. |
| <p>Orientador: Keiji Yamanaka. Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Elétrica. Modo de acesso: Internet. DOI http://doi.org/10.14393/ufu.te.2025.604 Inclui bibliografia. Inclui ilustrações.</p> | |
| <p>1. Engenharia elétrica. I. Yamanaka, Keiji, 1956-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Elétrica. III. Título.</p> | |

CDU: 621.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091

Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA - UFU
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito para a obtenção do título de Doutor em Ciências.

Uberlândia, Setembro de 2025

Banca Examinadora:

Prof. Dr. Keiji Yamanaka (UFU)

Prof. Dr. Edilberto Pereira Teixeira (UNIUBE)

Prof. Dr. José Roberto Camacho (UFU)

Prof. Dr. David Calhau Jorge (UFTM)

Prof. Dr. Virgílio de Melo Langoni (UFTM)



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Coordenação do Programa de Pós-Graduação em Engenharia
Elétrica

Av. João Naves de Ávila, 2121, Bloco 3N - Bairro Santa Mônica, Uberlândia-MG, CEP
38400-902

Telefone: (34) 3239-4707 - www.posgrad.feelt.ufu.br - cobel@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

| | | | | | |
|------------------------------------|---|-----------------|-------|-----------------------|-------|
| Programa de Pós-Graduação em: | Engenharia Elétrica | | | | |
| Defesa de: | Tese de Doutorado, 327, PPGEELT | | | | |
| Data: | Vinte e seis de Setembro de Dois mil e vinte e cinco | Hora de início: | 14:00 | Hora de encerramento: | 16:30 |
| Matrícula do Discente: | 11813EEL012 | | | | |
| Nome do Discente: | Florisvaldo Cardozo Bomfim Junior | | | | |
| Título do Trabalho: | Proposta de metodologia para otimização de dispositivos magnéticos por método híbrido aplicando elementos finitos e algoritmos genéticos | | | | |
| Área de concentração: | Processamento da Informação | | | | |
| Linha de pesquisa: | Metodologia e Técnicas da Computação | | | | |
| Projeto de Pesquisa de vinculação: | Coordenador do projeto: Keiji Yamanaka. Título do projeto: Aplicações de técnicas de Inteligência Artificial. Vigência do projeto: 01/2008 - atual. | | | | |

Reuniu-se no Anfiteatro do Bloco 1E, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta:

Doutores: José Roberto Camacho (UFU), David Calhau Jorge (UFTM), Virgílio de Melo Langoni (UFTM), Edilberto Pereira Teixeira (UNIUBE) e Keiji Yamanaka, orientador do discente.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Keiji Yamanaka, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir, o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

APROVADO.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor. O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme, foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Keiji Yamanaka, Professor(a) do Magistério Superior**, em 26/09/2025, às 16:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **David Calhau Jorge, Usuário Externo**, em 26/09/2025, às 16:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Edilberto Pereira Teixeira, Usuário Externo**, em 26/09/2025, às 16:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **José Roberto Camacho, Professor(a) do Magistério Superior**, em 26/09/2025, às 16:50, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Virgílio de Melo Langoni, Usuário Externo**, em 29/09/2025, às 11:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4984741** e o código CRC **9F24D1B4**.

Resumo

Este trabalho propõe uma metodologia para a otimização de dispositivos magnéticos, com foco na construção de indutores do tipo EI. A abordagem desenvolvida integra o método dos elementos finitos com algoritmos genéticos, utilizando o *software* FEMM. O objetivo principal é minimizar o volume de material, a quantidade de espiras da bobina e a distância do entreferro, de forma a gerar dispositivos mais eficientes e com menor custo de produção. Os resultados demonstraram a eficácia da metodologia proposta, evidenciando sua aplicabilidade na determinação das proporções ideais para projetos de indutores.

Palavras-chave: **Algoritmo Genético, Elementos Finitos, Indutores, Linguagem Lua.**

Abstract

This work proposes a methodology for optimizing magnetic devices, focusing on the construction of EI inductors. The developed approach integrates the finite element method with genetic algorithms, using FEMM software and the Lua programming language. The main objective is to minimize the material volume, the number of coil turns, and the air gap distance, in order to generate more efficient devices with lower production costs. The results demonstrate the effectiveness of the proposed methodology, demonstrating its applicability in determining the ideal proportions for inductor designs.

Keywords: **Genetic Algorithm, Finite Element Inductors, Lua Language.**

Lista de Figuras

| | | |
|----|---|----|
| 1 | Software FEMM | 11 |
| 2 | Espaço de busca de soluções de um problema. | 18 |
| 3 | Fluxograma do Algoritmo Genético. | 19 |
| 4 | Modelo de bobina adotado | 26 |
| 5 | Densidade de campo do indutor de 200 nH | 27 |
| 6 | Valores encontrados na simulação | 28 |
| 7 | Densidade de campo do indutor de 10 nH | 29 |
| 8 | Valores encontrados na simulação | 29 |
| 9 | Peça E do indutor | 31 |
| 10 | Indutor adotado para a otimização | 34 |
| 11 | Estrutura do Indivíduo | 36 |
| 12 | Indivíduo inicial | 37 |
| 13 | Indivíduo intermediário | 38 |
| 14 | Indivíduo otimizado encontrado | 38 |
| 15 | Gráfico de variação do volume | 39 |
| 16 | Simulação do indutor a ser otimizado | 40 |
| 17 | Tabela de densidade de fluxo | 40 |
| 18 | Simulação do indutor proposto | 41 |
| 19 | Tabela de densidade de fluxo | 42 |

Lista de Tabelas

| | | |
|---|--|----|
| 1 | Dimensionamento de indutores de núcleo de ar | 27 |
| 2 | Dimensionamento de indutores de núcleo de ar profundidade fixa | 28 |
| 3 | Evolução | 33 |
| 4 | Dados do indutor adotado | 35 |
| 5 | Dimensões do melhor indivíduo da primeira época | 37 |
| 6 | Dimensões do melhor indivíduo da sexta época | 38 |
| 7 | Dimensões do melhor indivíduo da décima época | 39 |

Sumário

| | |
|--|-----------|
| 1 INTRODUÇÃO | 3 |
| 2 Revisão de Literatura | 5 |
| 2.1 Campo Magnético | 5 |
| 2.2 Densidade de Fluxo | 5 |
| 2.3 Permeabilidade magnética | 6 |
| 2.4 Relutância | 6 |
| 2.5 Lei de Ampère | 7 |
| 2.6 Força magnetomotriz | 7 |
| 2.7 Indutores | 7 |
| 2.8 Cálculo de indutância | 7 |
| 2.9 Histerese | 9 |
| 2.9.1 Aplicações tecnológicas da histerese magnética | 9 |
| 2.10 Elementos Finitos | 9 |
| 2.10.1 Histórico da Aplicação | 10 |
| 2.10.2 <i>Software</i> FEMM | 11 |
| 2.10.3 Condições de Contorno | 12 |
| 2.10.4 Problemas Magnetostáticos | 12 |
| 3 Linguagem LUA | 14 |
| 3.0.1 Linguagem Lua - Aplicações e Uso | 14 |
| 3.0.2 Linguagem Lua - Vantagens | 14 |
| 3.0.3 Linguagem Lua - Desvantagem | 14 |
| 3.0.4 Funções de comandos da linguagem Lua dentro do software FEMM | 15 |
| 3.0.5 História | 16 |
| 3.1 Algoritmo Genético | 17 |
| 3.1.1 Funcionamento | 18 |
| 3.1.2 População inicial | 19 |
| 3.2 Seleção | 20 |
| 3.2.1 Seleção por Torneio | 20 |

| | | |
|----------|--|-----------|
| 3.3 | Elitismo | 21 |
| 3.4 | Reprodução | 21 |
| 3.4.1 | Operadores Genéticos de Recombinação (<i>Crossover</i>) | 21 |
| 3.4.2 | <i>Crossover</i> Geométrico de Michalewicz | 21 |
| 3.4.3 | Crossover de Blend (BLX) | 22 |
| 3.4.4 | SBX (<i>Simulated Binary Crossover</i>) | 23 |
| 3.5 | Mutação | 23 |
| 3.5.1 | Mutação uniforme proposta por Michalewicz | 24 |
| 3.5.2 | Aptidão | 24 |
| 4 | Resultados e Discussões | 25 |
| 4.1 | Dimensionamento de um indutor de núcleo de ar | 25 |
| 4.2 | Otimização da indutância por meio da distância do entreferro | 28 |
| 4.3 | Otimização da indutância por meio das dimensões | 34 |
| 4.3.1 | Cálculo Analítico do Modelo Proposto | 34 |
| 4.3.2 | Modelo de Estrutura | 35 |
| 4.3.3 | Função Fitness | 36 |
| 4.3.4 | Estrutura do Cromossomo | 36 |
| 4.3.5 | Parâmetros do AG proposto | 37 |
| 4.3.6 | Resultados | 37 |
| 4.3.7 | Variância Volumétrica da Estrutura | 39 |
| 4.3.8 | Validação do modelo | 39 |
| 5 | Conclusão | 43 |
| 6 | Anexo 1 | 47 |
| 6.1 | Comentário | 47 |
| 7 | Anexo 2 | 57 |
| 7.1 | Comentário | 57 |

Lista de Símbolos

\mathfrak{R} : Relutância magnética;

l_g : Entreferro;

μ_0 : Permeabilidade magnética;

n : Número de espiras;

L : Indutância;

NV : Número de Voltas

L : Largura

n_c : número de indivíduos da população. d_{nucleo} : diâmetro do núcleo de ar. DF : Densidade de Fluxo;

NP_n : Nível de prioridade; B : Densidade de fluxo em teslas(T)

ϕ : Fluxo em webers(Wb)

ρ : Resistividade elétrica $\Omega \cdot m$;

l : Comprimento do caminho magnético (em metros, m);

A : Área da seção transversal do caminho (em metros quadrados, m^2) ;

CAPÍTULO 1

1 INTRODUÇÃO

Uma das principais barreiras encontradas na construção de dispositivos magnéticos reside na fase de projeto, devido à discrepância entre os valores calculados analiticamente e aqueles efetivamente aferidos no dispositivo físico. Tais divergências decorrem das aproximações utilizadas nos cálculos tradicionais, que muitas vezes desconsideram efeitos relevantes, como a dispersão do fluxo magnético e os efeitos de borda. Essa imprecisão resulta na necessidade de construção de diversos protótipos até se alcançar uma configuração satisfatória que atenda às especificações do projeto, o que acarreta aumento significativo nos custos de desenvolvimento.

Dante desse cenário, é possível destacar três fatores críticos:

- Divergência entre os valores calculados e os obtidos experimentalmente;
- Aumento do custo do projeto devido à necessidade de múltiplos protótipos;
- Saturação do material do núcleo, resultante da complexidade das variáveis envolvidas, como formato e dimensões do dispositivo.

A partir dessas dificuldades, surgiu a seguinte indagação: “E se fosse possível desenvolver um dispositivo magnético no qual o protótipo já fosse o produto final?”

Essa questão motivou o desenvolvimento de uma metodologia alternativa para o dimensionamento de dispositivos magnéticos, visando determinar as proporções ótimas de projeto sem a necessidade de prototipagem repetida. O dispositivo escolhido como estudo de caso foi um indutor com núcleo do tipo EI.

O projeto de indutores exige a definição cuidadosa de seus componentes, buscando o equilíbrio entre variáveis como tamanho, peso, custo e eficiência. Por exemplo, sabe-se que a relação entre a seção transversal do núcleo e o número de espiras do enrolamento pode ser ajustada: ao se aumentar uma dessas variáveis, é possível reduzir a outra. No entanto, aumentar o número de espiras pode resultar em aquecimento excessivo e dificuldades no

acondicionamento da bobina na janela do núcleo. Por outro lado, o aumento da seção do núcleo acarreta maior massa e, consequentemente, um dispositivo mais caro.

Dante dessas complexidades, foram selecionadas como ferramentas para o desenvolvimento desta proposta o software FEMM (Finite Element Method Magnetics), a linguagem de programação Lua e os Algoritmos Genéticos (AGs). A combinação dessas ferramentas permite integrar a precisão do método dos elementos finitos com a capacidade de busca global e otimização dos algoritmos genéticos.

Os resultados obtidos demonstraram grande potencial, com otimizações significativas na distância do entreferro e nas dimensões construtivas do indutor, levando à redução do consumo de materiais e evitando a saturação do núcleo.

CAPÍTULO 2

2 Revisão de Literatura

Nesta seção, são abordados os principais conceitos teóricos que sustentam a metodologia proposta neste trabalho.

2.1 Campo Magnético

Ao redor de um ímã permanente existe um campo magnético que pode ser representado por linhas de campo. Em uma barra magnetizada, essas linhas se deslocam do polo norte para o polo sul, externamente, e no sentido oposto internamente, sendo igualmente espaçadas dentro da barra. A intensidade do campo é proporcional à quantidade de linhas de campo por unidade de área.

Na presença de corrente elétrica, um campo magnético é induzido ao redor de um fio condutor. Quando o condutor é enrolado em espiras, o campo se intensifica, especialmente no centro da espira. O enrolamento de várias espiras resulta em um campo magnético ainda mais intenso.

2.2 Densidade de Fluxo

O número de linhas de campo por unidade de área é chamado de densidade de fluxo magnético, sendo representado pela letra B e medido em teslas, sendo determinada pela equação 1.

$$B = \frac{\phi}{A} \quad (1)$$

onde:

B : Densidade de fluxo em teslas(T)

ϕ : Fluxo em webers(Wb)

A : Área da seção em metros quadrados(m^2)

2.3 Permeabilidade magnética

Um núcleo de mesmas dimensões, mas confeccionado com materiais diferentes, apresenta permeabilidades magnéticas diferentes, pois cada tipo de material permite uma certa quantidade de linhas de campo atravessarem por ele, e aqueles que possuem um fluxo intenso apresentam uma elevada permeabilidade magnética.

A razão entre a permeabilidade magnética de um material e a permeabilidade do vácuo é chamada de permeabilidade relativa.

$$\mu_r = \frac{\mu}{\mu_0} \quad (2)$$

Materiais magnéticos : $\mu_r \geq 100$

2.4 Relutância

A resistência ao escoamento de carga (R) que um material possui é representada pela equação 3.

$$R = \rho \cdot \frac{l}{A} \quad (3)$$

onde:

ρ : Resistividade elétrica $\Omega \cdot m$;

l : Comprimento do caminho magnético (em metros, m);

A : Área da seção transversal do caminho (em metros quadrados, m^2) ;

Realizando uma comparação, temos que a relutância é a tentativa de estabelecer um fluxo magnético no interior do material e é dada pela equação 4.

$$\Re = \frac{l}{\mu \cdot A} \quad (4)$$

2.5 Lei de Ampère

Fazendo uma analogia a essa lei, podemos dizer que, em um circuito magnético fechado, a soma algébrica das variações de força magnetomotriz é igual a zero.

2.6 Força magnetomotriz

É uma força proporcional ao produto do número de espiras em torno de um núcleo com a intensidade de corrente que atravessa o enrolamento.

$$F = N \cdot I \quad (5)$$

E sua relação com o fluxo magnético é dada por:

$$F = \phi \cdot \mathfrak{R} \quad (6)$$

A força magnetomotriz por unidade de comprimento é chamada de força magnetizante (H).

Assim,

$$H = \frac{F}{l} \quad (7)$$

Substituindo a equação (5) em (7), temos:

$$H = \frac{N \cdot I}{l} \quad (8)$$

2.7 Indutores

O elemento indutor possui uma característica de resposta semelhante, em muitos aspectos, à do capacitor.

2.8 Cálculo de indutância

Para o cálculo da indutância aproximada, adotamos primeiramente que não existem vazamentos e franjas, e que a contribuição das seções de ferro para a relutância são

desprezíveis em relação ao ar. Dessa forma, podemos deduzir que o fluxo flui pelo circuito magnético e é obtido por meio da equação a seguir.

$$\mathfrak{R} \cdot \phi = N \cdot i \quad (9)$$

ϕ : Fluxo;

\mathfrak{R} : Relutância do circuito magnético;

N : Número de espiras;

i : Corrente;

μ_0 : Permeabilidade magnética;

A indutância depende diretamente do número de espiras ao quadrado, dividida pela relutância do circuito, dada pela equação a seguir:

$$L = \frac{N^2}{\mathfrak{R}} \quad (10)$$

Para gerar um comparativo do cálculo da indutância por meio dos elementos finitos e do método convencional, são demonstrados os cálculos analíticos da indutância do projeto proposto, o que vem a apresentar uma divergência considerável quando comparado a outro método, que possui um menor erro quando é comparado ao elemento real analisado.

$$\begin{aligned} \mathfrak{R} &= \frac{l_g}{\mu_0 \cdot a_{pole}} \\ L &= \frac{n^2}{\mathfrak{R}} \end{aligned} \quad (11)$$

$$\begin{aligned} \mathfrak{R} &= \frac{2 \cdot 0.025}{\mu_0 \cdot \mu_m \cdot a_{pole}} = 3.2 \cdot 10^6 \\ L &= \frac{n^2}{\mathfrak{R}} = 1.36mH \end{aligned} \quad (12)$$

Se compararmos com os resultados usando o método de elementos finitos, teremos um valor de indutância igual a 1.78418 mH .

2.9 Histerese

Um material ferromagnético, quando adicionado a um campo magnético externo, em um dado instante, todos os momentos magnéticos do material ficam alinhados ao campo externo, conhecido como magnetização de saturação. Após ter atingido a saturação, um aumento posterior do campo magnético não produz mais nenhum aumento da magnetização. Reduzindo o campo magnético a zero, nota-se uma magnetização residual. Esse efeito é denominado de histerese. A magnetização e a desmagnetização de um material que possui histerese produzem perda de energia (área interna da curva de histerese) e a temperatura do material aumenta durante o processo. A curva de histerese é própria para cada material.

A partir da saturação inicial, o campo H passa a ser reduzido, a curva de magnetização não retorna seguindo seu trajeto original e produz-se um efeito de histerese. O efeito de histerese é gerado pela resistência à movimentação de paredes de domínio causada pela presença de defeitos no material cristalino, sendo que a área compreendida pela curva de histerese representa a perda de energia por unidade de volume, por ciclo de magnetização e desmagnetização.

2.9.1 Aplicações tecnológicas da histerese magnética

É amplamente utilizada em gravações de dados em cartões, discos rígidos e outros. O material apresentará uma maior resistência à desmagnetização quanto maior for a sua coercividade, ou seja, maior deve ser a intensidade do campo magnético externo para anular a magnetização do material. Os materiais que apresentam essa característica são interessantes para aplicações eletrônicas, pois as informações armazenadas não seriam facilmente destruídas por campos magnéticos externos.

2.10 Elementos Finitos

O método dos elementos finitos foi desenvolvido na quinta década do século XX, quando foram lançados os primeiros computadores. Esses métodos matemáticos já eram conhecidos, mas, devido às ferramentas de cálculos pouco desenvolvidas, a sua implementação ficava inviabilizada.

No início, os elementos finitos foram aplicados em análise de problemas da mecânica dos sólidos, logo após se estendendo para outros fenômenos físicos.

Sua utilização é de grande importância para o desenvolvimento de produtos e suas pesquisas, tendo como aplicação diversas áreas da engenharia, e como exemplo o cálculo de distribuição térmica em transformadores e máquinas girantes, sem a necessidade de desenvolver um protótipo para sua avaliação (BASTOS, 2012).

2.10.1 Histórico da Aplicação

A distribuição de campo em qualquer dispositivo eletromagnético obedece às equações diferenciais de segunda ordem, sendo que a solução analítica para essas equações é muitas vezes difícil. Devido a esse problema muitos pesquisadores utilizam métodos empíricos para gerar uma possível solução.

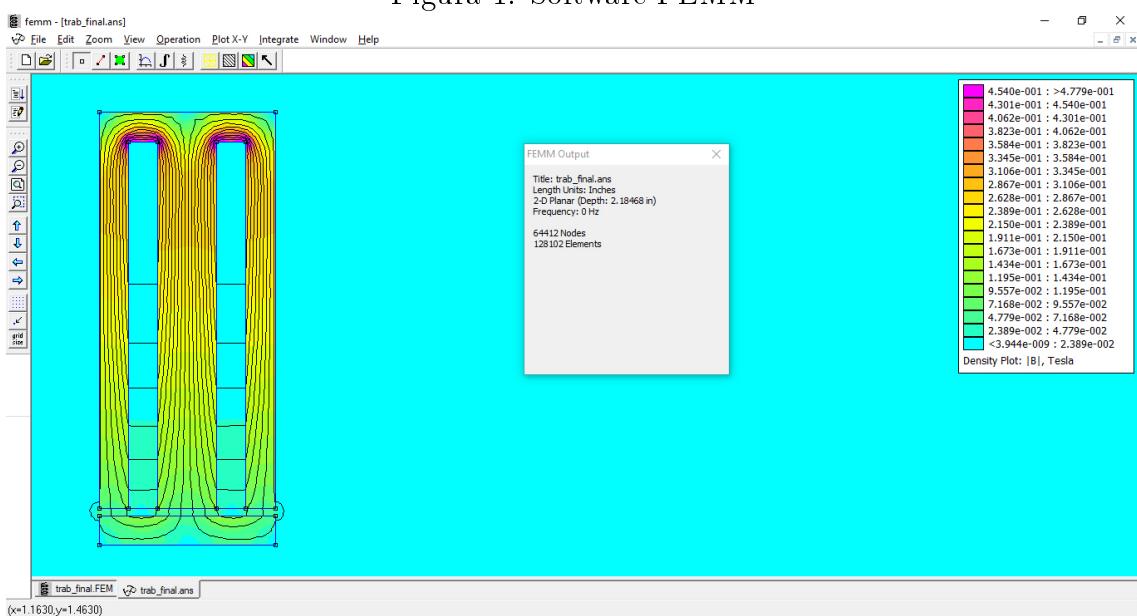
O método apresenta problemas estáticos. Os problemas decorrentes da variação de campo eletromagnético são comuns em diversos dispositivos, como é o caso de capacitores, indutores, ímãs e outros; portanto, o comportamento de dispositivos eletromagnéticos depende da distribuição dos campos em seu interior. Assim, por meio dos cálculos pode-se obter os projetos finais, sem a necessidade da construção de inúmeros protótipos (BASTOS, 2012).

2.10.2 Software FEMM

FEMM, representado na Figura 1, é uma plataforma que apresenta um conjunto de programas para resolver problemas eletromagnéticos de baixa frequência em domínios planares e axissimétricos. Atualmente, ele aborda problemas magnetostáticos lineares/não lineares, problemas magnéticos harmônicos lineares/não lineares no tempo, problemas eletrostáticos lineares e problemas de fluxo de calor em estado estacionário.

Está dividido em três partes:

Figura 1: Software FEMM



Fonte: Autor

- Shell interativo (femm.exe). Este programa é um pré-processador de Interface de Documentos Múltiplos.
- Triangle divide a região da solução em um grande número de triângulos.
- Solvers: fkern para problemas magnéticos e eletrostáticos; hsolv para problemas de fluxo de calor; e csolv para problemas de fluxo de corrente.

A linguagem Lua é a utilizada para a programação e integrada ao shell interativo. Ao contrário das versões anteriores, apenas uma instância de Lua está sendo executada por vez. A instância em Lua pode construir e analisar uma geometria e avaliar os resultados do pós-processamento, simplificando a criação de vários tipos de execuções em "lote".

2.10.3 Condições de Contorno

É necessário garantir um número adequado de condições de contorno com o intuito de garantir uma solução única.

2.10.4 Problemas Magnetostáticos

Problemas magnetostáticos são aqueles nos quais os campos são invariantes no tempo. Intensidade (H) e densidade de fluxo (B) devem obedecer às equações apresentadas a seguir.

$$\nabla \times H = J \quad (13)$$

$$\nabla \cdot B = 0 \quad (14)$$

sujeito a uma relação constitutiva entre B e H para cada material:

$$B = \mu \cdot H \quad (15)$$

Se um material não for linear, a permeabilidade μ fica em função de B :

$$\mu = \frac{B}{H(B)} \quad (16)$$

FEMM procura encontrar um campo que satisfaça (13)-(15) por meio de uma abordagem de potencial vetorial magnético. A densidade de fluxo é escrita em termos do potencial vetorial, A , como:

$$B = \nabla \times A \quad (17)$$

Agora, esta definição de B sempre satisfaz (14). Então, (13) pode ser reescrito como:

$$\nabla \times \left(\frac{1}{\mu(B)} \nabla \times A \right) = J \quad (18)$$

Para um material isotrópico linear (e assumindo que $\nabla \cdot A = 0$), a equação (18) se reduz para:

$$-\frac{1}{\mu} \nabla^2 A = J \quad (19)$$

O *software* FEMM mantém a forma de (18) para a solução de problemas magnetostáticos com relação B-H não lineares. No caso geral tridimensional, A é um vetor com três componentes. Entretanto, nos casos bidimensionais planos ou assimétricos, duas dessas componentes são nulas, restando apenas a direção “para fora do plano” (MEEKER, 2018).

A principal vantagem do uso do potencial vetorial é reunir todas as condições a serem satisfeitas em uma única equação. Ao determinar A, é possível obter B e H por diferenciação. Essa equação diferencial parcial elíptica aparece em diversos fenômenos de engenharia, e ao longo dos anos foram desenvolvidas muitas ferramentas para resolvê-la (MEEKER, 2018).

A seção seguinte apresenta as características da linguagem usada nas simulações.

CAPÍTULO 3

3 Linguagem LUA

A linguagem Lua é uma linguagem de programação interpretada, sendo seus *scripts* em alto nível e tendo uma tipagem dinâmica e multiparadigma. Foi desenvolvida na PUC-Rio, em 1993, (Tecgraf) para expandir aplicações em geral, usada para embarcar em sistemas de software complexos, como jogos e tendo uma semelhança com Python, Ruby e Icon, dentre outras. Devido à sua eficiência, facilidade de aprendizagem e clareza, passou a ser usada em diversos ramos da programação, como desenvolvimento de jogos, robótica e outros.

3.0.1 Linguagem Lua - Aplicações e Uso

- Aplicações industriais: Adobe Lightroom e Adobe Photoshop;
- Jogos: Roblox, Angry Birds, The Sims e muitos outros.

3.0.2 Linguagem Lua - Vantagens

- É uma linguagem leve;
- É portátil;
- Quando comparada com Perl e Python, o seu desempenho é melhor;

3.0.3 Linguagem Lua - Desvantagem

- É mais famosa fora do Brasil do que onde foi criada;
- Quando comparada com a linguagem C, apresenta desempenho inferior;
- Tem alto risco de vulnerabilidade.

3.0.4 Funções de comandos da linguagem Lua dentro do software FEMM

Os comandos selecionados a seguir são os mais utilizados na construção do elemento a ser simulado:

- `newdocument()` : Cria um novo documento de pré-processador e abre uma nova janela de pré-processador. Especifique `doctype` como 0 para um problema magnético, 1 para um problema eletrostático, 2 para um problema de fluxo de calor ou 3 para um problema de fluxo de corrente.
- `mi_probdef()` : Define a frequência desejada em Hertz. O parâmetro de unidades especifica as unidades usadas para medir o comprimento no domínio do problema.
- `mi_getmaterial` : Busca o material especificado na biblioteca de materiais.
- `mi_addcircprop` : Adiciona uma nova propriedade de circuito. O parâmetro *circuittype* é 0 para um circuito conectado em paralelo e 1 para um circuito conectado em série.
- `mi_addnode(x,y)` : Adiciona um novo nó nas coordenadas x e y.
- `mi_addelement(x1,y1,x2,y2)` : Adiciona um novo segmento de linha do nó mais próximo de (x1,y1) para o nó mais próximo de (x2,y2).
- `mi_addblocklabel(x,y)` : Adiciona um novo rótulo de bloco em (x,y).
- `mi_drawrectangle` : Adiciona um retângulo com valores de início e fim.
- `mi_selectlabel(x,y)` : Seleciona o rótulo próximo a (x,y). Retorna as coordenadas do rótulo selecionado.
- `mi_setblockprop` : Define os rótulos dos blocos selecionados para terem as propriedades escolhidas.
- `mi_clearselected()` : Limpa todos os nós, blocos, segmentos e segmentos de arco selecionados.

- `mi_addarc(x1,y1,x2,y2,angle,maxseg)` : Adiciona um novo segmento de arco do nó mais próximo de (x_1, y_1) para o nó mais próximo de (x_2, y_2) com o ângulo dividido em segmentos.
- `mi_selectarcsegment(x,y)` : Seleciona o segmento de arco mais próximo de (x, y) .
- `mi_addboundprop("propname", A0, A1, A2, Phi, Mu, Sig, c0, c1, BdryFormat, eu, oa)` adiciona uma nova propriedade de limite com o nome "`propname`".
- `mi_setarcsegmentprop(maxsegdeg, "propname", hide, group)` : Define o arco selecionado para receber as propriedades de limite.
- `mi_saveas("filename")` : Salva o arquivo com o nome "`filename`".
- `mi_createmesh()` : executa *Triangle* para criar uma malha. Observe que este não é um precursor necessário para realizar uma análise, pois `mi_analyze()` garantirá que a malha esteja atualizada antes de executar uma análise. O número de elementos na malha é colocado de volta na pilha.
- `mi_analyze(flag)` : executa *fkern* para resolver o problema. O parâmetro *flag* controla se a janela *fkern* está visível ou minimizada. Para uma janela visível, não especifique nenhum valor para *flag* ou especifique 0. Para uma janela minimizada, o *flag* deve ser definido como 1.

3.0.5 História

A linguagem Lua foi criada no ano de 1993, por Roberto Ierusalimschy, Luiz Henrique de Figueiredo e Waldemar Celes, membros do *Computer Graphics Technology Group* na PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro) no Brasil. O primeiro projeto desenvolvido usando a linguagem foi em jogos. Na seção seguintes são apresentados os principais conceitos sobre algoritmos genéticos.

3.1 Algoritmo Genético

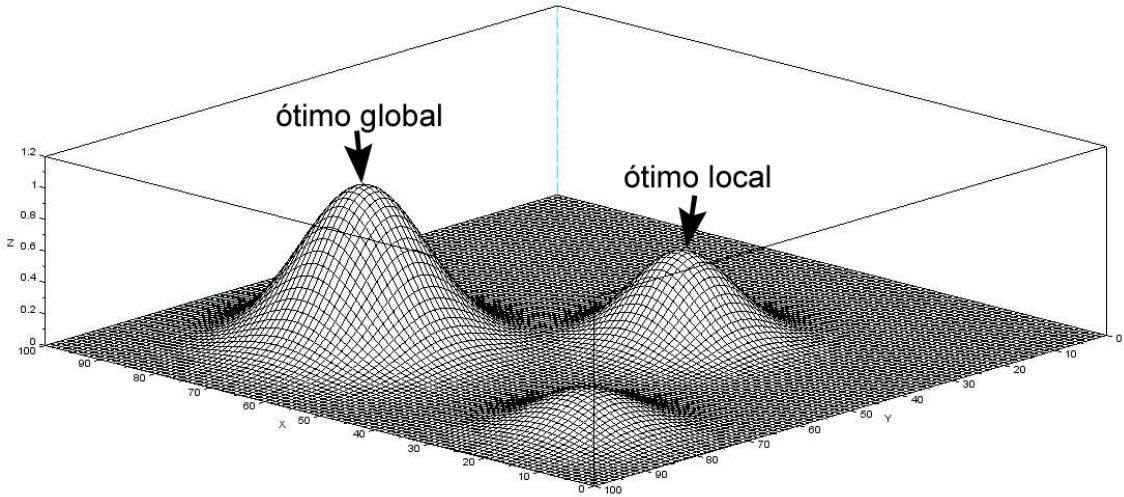
Algoritmos Genéticos são métodos de otimização de simples compreensão e implementação, desenvolvidos por John Holland e por um grupo de pesquisadores da Universidade de Michigan, na década de 1960. O método imita os conceitos consagrados da genética, sendo inspirado na teoria da seleção natural dos indivíduos e da sobrevivência dos mais adaptados ao meio, de Charles Darwin (COLEY, 1999). Dessa forma, indivíduos com aptidões genéticas mais favoráveis às condições do meio têm maiores chances de sobrevivência e de se reproduzir, se comparados a outros com características genéticas menos aptas que tendem a desaparecer no processo evolutivo, devido à sua menor possibilidade de reprodução; mas lembrando que todos os indivíduos possuem o mesmo grau de importância, pois, durante o processo de reprodução, dois indivíduos com menor aptidão podem gerar um que apresente uma característica genética superior à da população vigente. Na busca da solução ótima do problema, interpreta-se cada possível solução do problema como um indivíduo de uma população em processo evolutivo, e, seguindo os princípios da teoria Darwiniana, a cada iteração são selecionadas as melhores soluções para serem progenitoras das soluções que irão compor a nova população de soluções e descartadas aquelas que se encontram distantes da solução ótima do problema, direcionando, assim, a busca pela solução ótima. O Algoritmo Genético inicia sua busca a partir de uma população composta por vários indivíduos espalhados em um espaço de soluções potenciais, sendo esta uma das maiores vantagens deste método de otimização, apresentando as seguintes características para a solução do problema:

- Independência de conhecimentos matemáticos avançados;
- Possibilidade de hibridização com outros métodos de otimização;
- Capacidade de otimizar problemas com grande número de variáveis e restrições;
- Facilidade para escapar dos pontos de ótimos locais existentes no espaço de soluções potenciais.

Entende-se como ótimo global a melhor solução possível para um determinado problema, e, como soluções ótimas locais, aquelas que se despontam em relação à maioria;

porém, não superam a solução ótima global. A Figura 2 ilustra o espaço de busca de soluções de um determinado problema, apontando o ponto de ótimo global e os pontos ótimos locais.

Figura 2: Espaço de busca de soluções de um problema.



Fonte: Autor.

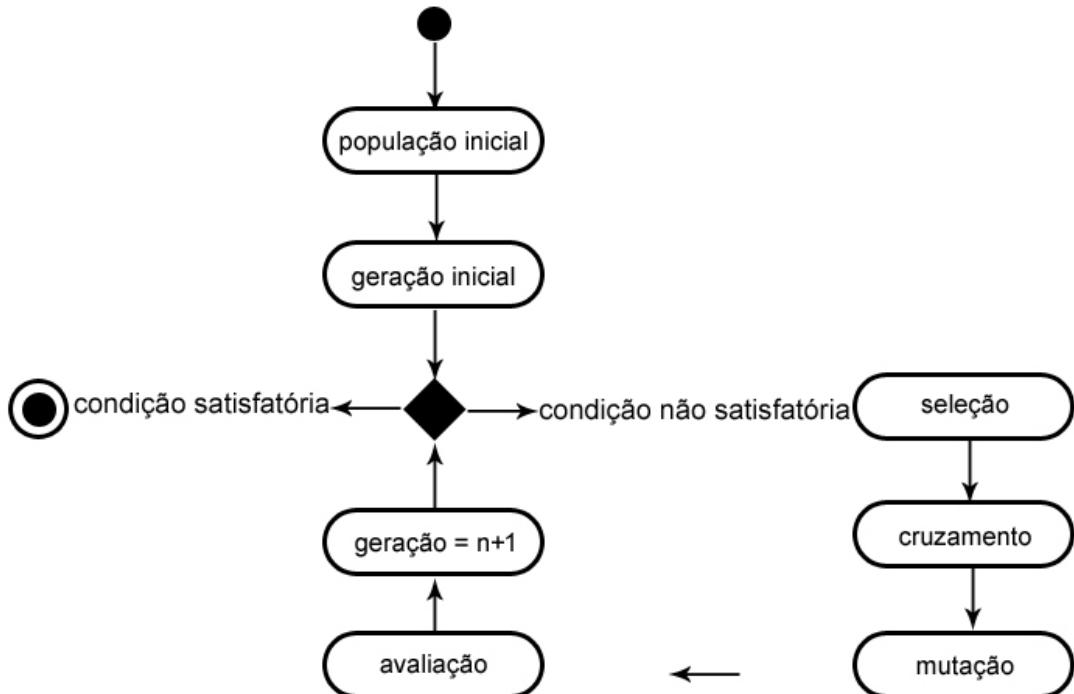
Para uma melhor compreensão do método, apresenta-se, na Figura 3, o fluxograma de um Algoritmo Genético.

3.1.1 Funcionamento

Em vez de partir de um único ponto (ou adivinhar) dentro do espaço de busca, AGs são inicializados com uma população de suposições. Estes são geralmente aleatórios e espalhados por todo o espaço de busca. Um algoritmo típico usa três operadores - seleção, recombinação e mutação (escolhidos em parte por analogia com o mundo natural) - a fim de direcionar a população, ao longo de uma série de etapas de tempo ou gerações, para a convergência no ótimo global. Normalmente, essas suposições iniciais são mantidas como codificações binárias (ou *strings*) das variáveis verdadeiras, embora um número crescente de AGs use "valores reais" (ou seja, base decimal), ou codificações que foram escolhidas para imitar, de alguma maneira, a estrutura de dados naturais do problema (COLEY, David A., 1999).

Quanto aos três principais operadores da população, temos:

Figura 3: Fluxograma do Algoritmo Genético.



Fonte: Autor.

1. A seleção tenta pressionar a população de uma maneira semelhante à da seleção natural encontrada nos sistemas biológicos.
2. A recombinação permite que as soluções troquem informações de forma semelhante à que é usada por um organismo natural submetido à reprodução sexual.
3. A mutação é usada para mudar aleatoriamente (*flip*) o valor de *bits* individuais dentro de *strings* individuais. O uso típico é usado com muita moderação.

Em seguida, os operadores de seleção, recombinação e mutação são aplicados à população inicial. Uma nova população é formada, e é adicionado mais 1 no contador da geração. Este processo de seleção, *crossover* e mutação é continuado até certo número de gerações ter decorrido ou até alguma forma de critério de convergência ser cumprido.

3.1.2 População inicial

Na primeira etapa do AG, com o objetivo de mapear um espaço inicial de busca de possíveis soluções, cria-se uma população com número predeterminado de indivíduos, sendo

cada um destes uma possível solução do problema. A quantidade de indivíduos que irá compor essa população deve ser definida de forma cautelosa, tendo em vista a influência direta desta escolha no tempo de processamento do algoritmo.

Aconselha-se que a definição do número de indivíduos seja pautada na busca de um equilíbrio entre abrangência do espaço de busca da solução, qualidade dos resultados obtidos e tempo de processamento do algoritmo.

Para o nosso estudo, o cromossomo adotado será:

$$cromossomo = [NV, L] \quad (20)$$

em que:

NV : Número de Voltas

L : Largura

3.2 Seleção

A seleção é particularmente simples: os melhores 50% são selecionados para se reproduzir, e o resto é descartado. Este é um método prático, mas não é o mais comum. Um dos motivos para isso é que não faz distinção entre "bom" e "muito bom". Além disso, em vez de apenas permitir soluções pobres avançarem para a próxima geração com uma probabilidade muito menor, ele simplesmente as aniquila (reduzindo a diversidade genética da população). O operador de seleção proporcional mais comum é a roleta. Com isso, a probabilidade de seleção é proporcional à aptidão de um indivíduo.

3.2.1 Seleção por Torneio

No método de seleção por torneio, dois indivíduos ("r" e "c") são escolhidos aleatoriamente na população, sendo que o que apresentar a maior aptidão é selecionado para a próxima etapa.

3.3 Elitismo

A seleção proporcional não garante a seleção do indivíduo mais apto. Embora pareça contraproducente, isso pode ser vantajoso para alguns problemas porque retarda o algoritmo, permitindo que se explore mais o espaço de pesquisa antes da convergência. Elitismo é a técnica que reserva, a cada geração, os melhores indivíduos, para que não se percam durante a evolução.

3.4 Reprodução

A literatura apresenta vários tipos de operadores genéticos usados nesta etapa, tendo como alguns exemplos: *crossover geométrico* de Michalewicz, *crossover Blend*, *crossover SBX* (propostas neste trabalho), mutação heurística, mutação heurística não uniforme, mutação uniforme e não uniforme de Michalewicz. A taxa de recombinação genética foi de 90%. Uma importante observação é que todos os operadores genéticos são moldados de forma a não permitirem que os novos indivíduos sejam gerados fora do seu universo de discurso. Caso o fato aconteça, os algoritmos geram um indivíduo aleatório dentro da faixa adotada.

3.4.1 Operadores Genéticos de Recombinação (*Crossover*)

Nesta seção serão apresentados um dos três operadores genéticos mais utilizados.

3.4.2 *Crossover Geométrico* de Michalewicz

Esse tipo de operador gera filhos por meio de dois indivíduos X e Y (Michalewicz, 1996), podendo ser aplicado a um número maior de indivíduos escolhidos.

O funcionamento deste operador é apresentado a seguir:

Passo 1 : Obter dois indivíduos selecionados aleatoriamente X e Y (pais).

$$X = \{x_1, x_2, \dots, x_{nc}\}, Y = \{y_1, y_2, \dots, y_{nc}\} \quad (21)$$

Passo 2 : Determinação do novo indivíduo W.

$$W = \{\sqrt{x_1 \cdot y_1}, \sqrt{x_2 \cdot y_2}, \dots, \sqrt{x_n \cdot y_n}\} \quad (22)$$

Em que: n_c é o número de indivíduos da população.

Um dos problemas encontrados neste tipo de reprodução é a ocorrência de números negativos, o que comprometeria o funcionamento do algoritmo. Dessa forma, são tomadas medidas para que, caso ocorra, a ocorrência seja tratada antes de iniciar o processo apresentado acima.

3.4.3 Crossover de Blend (BLX)

A proposta deste operador é gerar um novo indivíduo, levando em consideração a diferença genética entre os indivíduos pais selecionados X e Y . Este operador genético cria um novo indivíduo de acordo com uma distribuição de probabilidade uniforme em uma região contendo os dois indivíduos pais, sendo gerado na região mediana entre os pais. Isto é, há maior probabilidade de ser gerado em uma região central.

O funcionamento do método é apresentado a seguir:

Passo 1: Obter dois indivíduos X e Y .

$$X = \{x_1, x_2, \dots, x_{n_c}\}, Y = \{y_1, y_2, \dots, y_{n_c}\} \quad (23)$$

Passo 2: Gerar um número aleatório entre 0 e 1.

Passo 3: Calcular o valor do gama para a geração W_i .

$$\gamma = u_1 \cdot (1 + 2 \cdot \alpha) - \alpha \quad (24)$$

Em que: $\alpha = 0.5$, conforme (Deb e Beyer, 2001)

Passo 4: Para $i = 1, \dots, n_c$, determinar W .

$$\begin{aligned} W_i &= (1 - \gamma) \cdot x_i + \gamma \cdot y_i, & (x_i < y_i) \\ W_i &= (1 - \gamma) \cdot y_i + \gamma \cdot x_i, & (x_i \geq y_i) \end{aligned} \quad (25)$$

3.4.4 SBX (*Simulated Binary Crossover*)

O *crossover* binário simulado (SBX – *Simulated Binary Crossover*) trabalha com dois indivíduos selecionados, P_1 e P_2 , e cria dois novos indivíduos C_1 e C_2 . O operador SBX e o *crossover* Blend são considerados operadores autoadaptativos, pois se adaptam automaticamente durante a execução de um algoritmo genético. O *crossover* SBX utiliza uma distribuição de probabilidade não uniforme bimodal. O indivíduo criado é uma aproximação de centro nos dois indivíduos pais, isto é, tem maior probabilidade de ser gerado em duas regiões, cujos centros são os dois indivíduos pais. (Deb, Joshi e Anand, 2001).

$$u_i = \text{rand}(1) \quad (26)$$

$$B_{qi} = (2u_i)^{\frac{1}{\eta+1}} \quad u_i \leq \frac{1}{2} \quad (27)$$

$$B_{qi} = \left(\frac{1}{2 \cdot (1 - u_i)} \right)^{\frac{1}{\eta+1}} \quad u_i > \frac{1}{2} \quad (28)$$

$$C_1 = \frac{1}{2}((1 - B_{qi}) \cdot P_1 + (1 + B_{qi}) \cdot P_2) \quad (29)$$

$$C_2 = \frac{1}{2}((1 - B_{qi}) \cdot P_2 + (1 + B_{qi}) \cdot P_1) \quad (30)$$

onde o índice i é uma referência ao i -ésimo alelo; B_{qi} é definido pelas Equações 27 e 28; u_i , uma variável aleatória com distribuição uniforme entre 0 e 1. Note que, como não podem existir coeficientes negativos, existe um teste que define como zero os filhos que porventura se tornem negativos ao final da recombinação.

3.5 Mutação

No mundo natural, vários processos podem causar mutação. As taxas de bactérias são aproximadamente 2×10^{-3} por genoma por geração (FU90, BA96, p. 19). Por meio de uma representação binária, a mutação é particularmente fácil de implementar. A cada nova geração, toda a população é varrida, com cada posição de *bit* em cada *string* visitada, e muito ocasionalmente, 1 é invertido para 0, ou vice-versa. A probabilidade de mutação P é tipicamente da ordem de 0,001, ou seja, um *bit* em cada mil será mutado. No entanto,

assim como em todo o resto sobre os AGs, a correta configuração para P dependerá do problema. Alguns autores realizam a mutação, visitando cada posição de *bit*, jogando aleatoriamente 0 ou 1 e substituindo o *bit* existente por esse novo valor. Como há uma probabilidade de 50% de o *bit* pré-existente e o substituto serem idênticos, a mutação será realizada apenas pela metade da taxa.

3.5.1 Mutação uniforme proposta por Michalewicz

O método consiste em gerar um novo indivíduo a partir de um indivíduo pai, alterando o valor de um dos componentes do indivíduo doador, por um valor aleatório pertencente ao universo de discurso deste componente (Michalewicz, 1996).

O procedimento deste operador é apresentado a seguir:

Passo 1: Obter o indivíduo doador.

$$X = \{x_1, x_2, \dots, x_{n_c}\} \quad (31)$$

Passo 2: Escolher aleatoriamente um componente p do indivíduo doador.

Passo 3: Calcular um novo indivíduo, gerando um novo valor pertencente ao universo de discurso.

3.5.2 Aptidão

Durante cada iteração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Através da seleção, são determinados quais indivíduos conseguirão se reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pelo seu índice de aptidão. Em outras palavras, os indivíduos com maior adaptação relativa têm maiores chances de se reproduzir.

CAPÍTULO 4

4 Resultados e Discussões

4.1 Dimensionamento de um indutor de núcleo de ar

O primeiro passo para a geração do método foi a implementação de um algoritmo que inicialmente encontrasse o número de espiras e suas dimensões para um valor de indutância desejada, tendo como objetivo primário o valor da indutância, e, como secundário, o número de espiras, pois uma das metas é reduzir o valor de produção do elemento por meio da quantidade de cobre usado em sua confecção. Para isso, inicialmente, foi adotado o condutor 12 AWG e uma corrente de 1 A que será aplicada nas espiras, conforme o Algoritmo 1 apresentado a seguir:

Algoritmo 1: Configurações para a simulação

```
newdocument(0)
mi_probdef(0, 'millimeters', 'planar', 1E-8, dnucleo, 30, 0)
mi_getmaterial('Air')
mi_getmaterial('12 AWG')
mi_getmaterial('US Steel Type 2-S 0.018 inch thickness')
mi_addcircprop('C12AWG', 1, 1)
```

Em que: dnucleo = diâmetro do núcleo de ar.

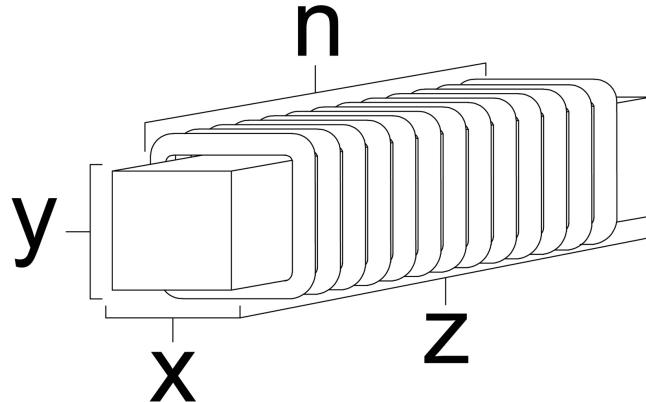
Para a simulação, fez-se necessário que o *software* fosse capaz de montar a nova estrutura para cada indivíduo da população. Dessa forma, optou-se pelo modelo que está apresentado na Figura 4, e a sua lógica de montagem da estrutura é representada pelo Algoritmo 2.

O AG foi configurado com os seguintes valores padrões:

- População: 200
- Gerações: 30
- Número de indivíduos de torneio: 3¹

¹Três indivíduos são selecionados aleatoriamente e, logo após, é observado aquele que possui a melhor aptidão. O vencedor é enviado para o próximo processo.

Figura 4: Modelo de bobina adotado



Fonte: Autor

Algoritmo 2: Montagem da estrutura

```

for loops=1, medvolta do
    mi_addnode(coordenadax, coordenaday)
    mi_addnode(coordenadax+rcond, coordenaday+dcond+dnucleo)
    if loops<medvolta then
        mi_addblocklabel(coordenadax+rcond, coordenaday)
        mi_addblocklabel(coordenadax+dcond, coordenaday+dcond+dnucleo)
        mi_selectlabel(coordenadax+rcond, coordenaday)
        mi_setblockprop('12 AWG', 1, 0 , 'C12AWG', 0, 0, 1)
        mi_selectlabel(coordenadax+dcond, coordenaday+dcond+dnucleo)
        mi_setblockprop('12 AWG', 1, 0 , 'C12AWG', 0, 0, -1)
        mi_clearselected()
    end

```

- Probabilidade de recombinação: 0,95 ou 95% (na falha, os pais selecionados são copiados para a próxima geração)
- Probabilidade de mutação: 0,15 ou 15%
- Atualização da geração com troca dos pais, pelos filhos.

Foram realizadas seis simulações, sendo, cada uma delas, com um valor de indutância alvo diferente, sendo eles os valores de 10 nH, 20 nH, 30 nH, 40 nH, 100 nH e 200 nH. Para isso, a função *fitness* é apresentada a seguir:

corrente, Voltagedrop, fluxx = mo_getcircuitproperties('C12AWG')

$$\text{funfit} = \text{abs}(3 \times 10^{-7} - \text{fluxx}/\text{corrente}) \quad (32)$$

$$\text{custo[varredura]} = 10^{10} \cdot \text{funfit} + \text{volta}$$

A função *fitness* tem como objetivo encontrar o valor em Henry desejado e também forçar a redução do número de espiras.

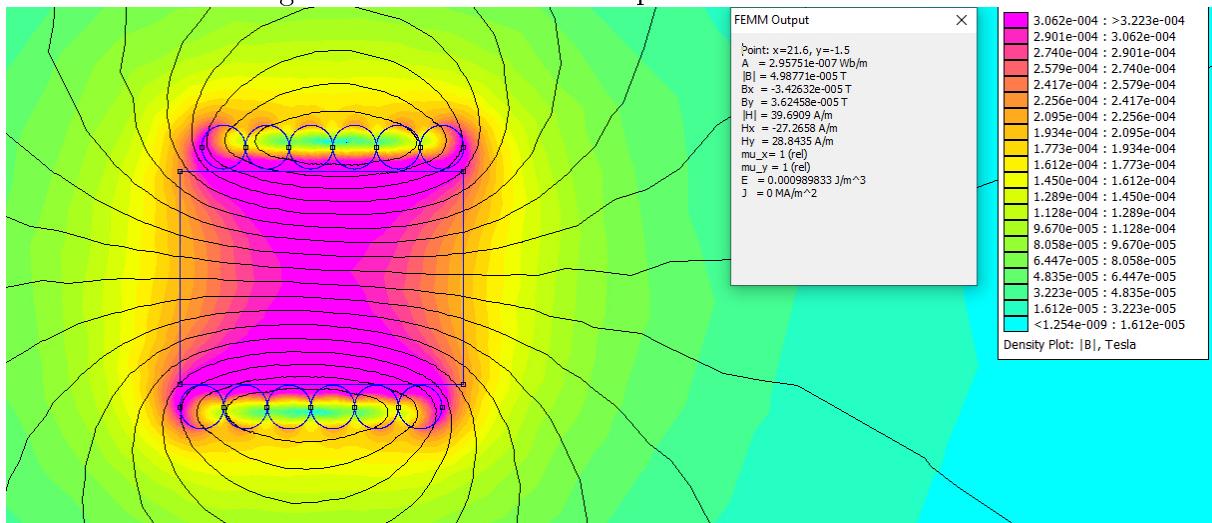
Após realizadas as simulações, foram encontradas as seguintes dimensões para os valores de indutância desejado.

Tabela 1: Dimensionamento de indutores de núcleo de ar

| Indutância Alvo | Indutância Encontrada | Número de Voltas | Dimensões ($x \times y \times z$) |
|-----------------|-----------------------|------------------|-------------------------------------|
| 10 nH | 10,16519 nH | 5 | 10 × 1,597 × 1,597 mm |
| 20 nH | 19,9882 nH | 2 | 4 × 6,587 × 6,587 mm |
| 30 nH | 29,9635 nH | 2 | 4 × 8,85 × 8,85 mm |
| 40 nH | 40,2067 nH | 2 | 4 × 10,99 × 10,99 mm |
| 100 nH | 100,603 nH | 7 | 14 × 5,47 × 5,47 mm |
| 200 nH | 200,241 nH | 6 | 12 × 9,87 × 9,87 mm |

Os resultados das simulações são apresentados a seguir, demonstrando as densidades de campo em cada ponto do indutor:

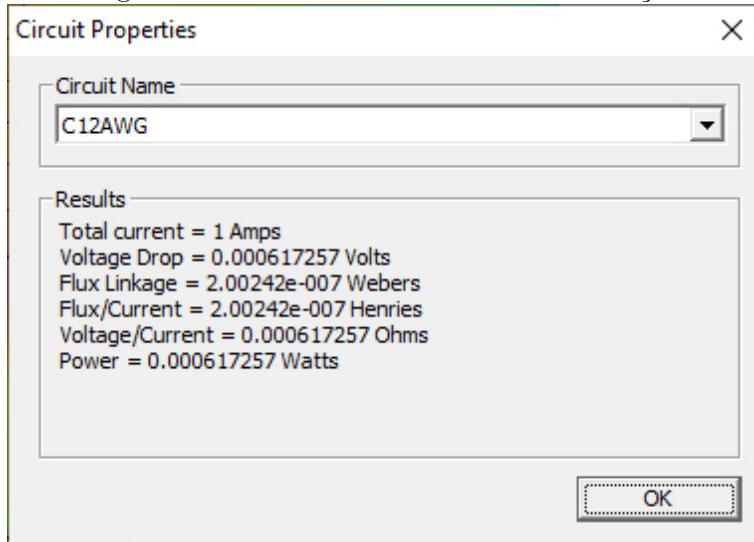
Figura 5: Densidade de campo do indutor de 200 nH



Fonte: Autor

As propriedades do circuito são apresentadas na Figura 6, onde podemos encontrar o valor de 200 nH, que foi um dos objetivos da simulação.

Figura 6: Valores encontrados na simulação



Fonte: Autor

Um outro ponto de vista explorado foi quando uma das dimensões, por uma necessidade do projeto, fosse fixada. Devido a esse fato, o *software* foi alterado para que a dimensão da profundidade permanecesse em 2 mm (conforme apresentado no Algoritmo 3), e o algoritmo ajustado para encontrar os valores de 10 nH, 20 nH, 30 nH e 40 nH.

Algoritmo 3: Fixando a profundidade na simulação

```
mi_probdef(0,millimeters,planar,1E-8,2,30,0)
```

Tabela 2: Dimensionamento de indutores de núcleo de ar profundidade fixa

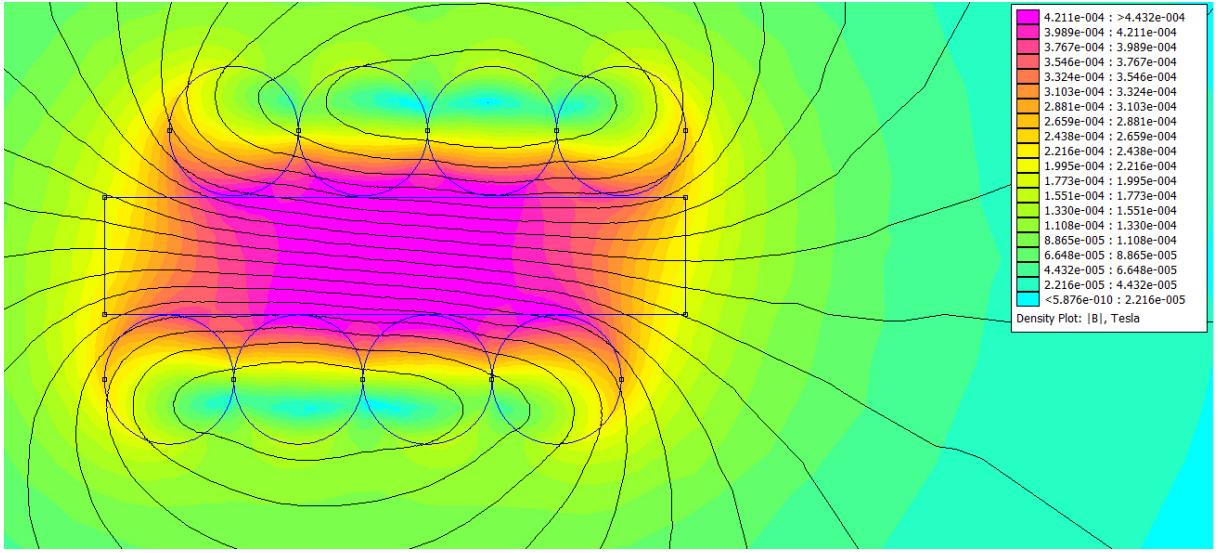
| Indutância Alvo | Indutância Encontrada | Número de Voltas | Dimensões ($x \times y \times z$) |
|-----------------|-----------------------|------------------|-------------------------------------|
| 10 nH | 10,0768 nH | 4 | $2 \times 1,8455 \times 8$ mm |
| 20 nH | 20,2718 nH | 6 | $2 \times 2,6447 \times 12$ mm |
| 30 nH | 30,0719 nH | 5 | $2 \times 9,3735 \times 10$ mm |
| 40 nH | 40,4769 nH | 5 | $2 \times 18,1 \times 10$ mm |

Os resultados da simulação são apresentados a seguir:

4.2 Otimização da indutância por meio da distância do entreferro

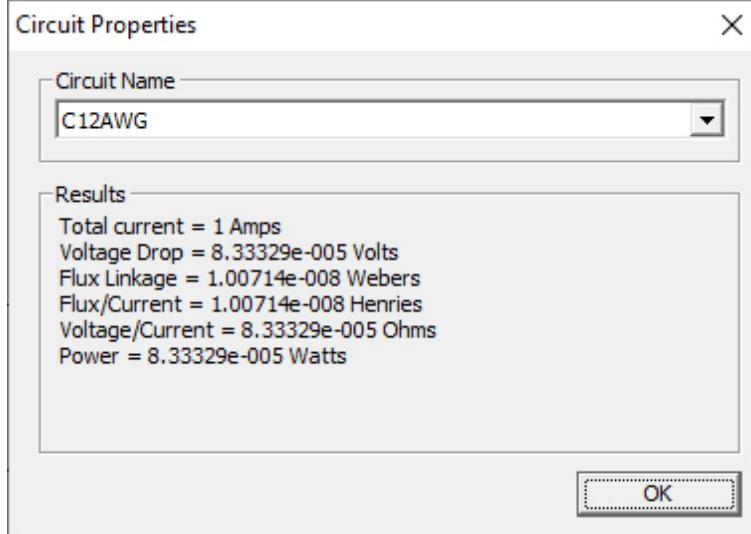
Para realizar a otimização da indutância, tendo como variável a distância do entreferro, foram realizados os cálculos de forma analítica e confrontados com o encontrado por meio do *software* desenvolvido. O exemplo adotado será um indutor que apresenta uma

Figura 7: Densidade de campo do indutor de 10 nH



Fonte: Autor

Figura 8: Valores encontrados na simulação



Fonte: Autor

geometria composta por um núcleo laminado E-I. O núcleo E apresenta um polo central de 0.5" e dois polos externos de 0.25" por 1" de profundidade, tendo uma distância g que será calculada para que o mesmo apresente uma indutância de 0,6 mH. A permeabilidade relativa adotada será de 2500 e ele vai conter um total de 49 espiras usando um condutor 18 AWG. Determinando a relutância desejada:

$$\mathfrak{R} = \frac{n^2}{L} = \frac{49^2}{0,6 \text{ mH}} = 4001667 \text{ A/Wb} \quad (33)$$

$$g = \frac{\Re \cdot \mu_0 \cdot a_{\text{pole}}}{2} = 0,0319'' \quad (34)$$

Dessa forma, conseguimos determinar qual a distância do entreferro que esse indutor deve ser aferida para obter a indutância desejada. Mas, para realizar a montagem, muitos projetistas se deparam com uma discrepância elevada entre o valor calculado e o valor real encontrado.

O *software* desenvolvido foi para determinar o valor otimizado do entreferro. Ele consiste em deixar fixo o E do indutor e movimentar a posição do I para encontrar a indutância desejada.

Os valores adotados para as configurações do projeto, considerando:

- Frequência: 0 Hz
- Unidade: inches
- Tipo do Problema: Planar
- Precisão: 1E-8
- Profundidade: 1
- Menor ângulo: 30°

Os materiais adotados para a construção do indutor foram:

- Ar
- 18 AWG
- M-15 Steel

O circuito adotado foi o C18 AWG, tendo uma corrente de 1 A. O Algoritmo 4 representa como as configurações foram realizadas usando a linguagem Lua.

Para o desenvolvimento da estrutura E (Figura 9) usamos somente os comandos `addnode()` e `addsegment()`, em sua grande maioria, gerando a seguinte estrutura.

A construção da estrutura I (Algoritmo 6), por ser móvel, recebeu uma variável chamada *variância* que realiza o deslocamento da peça para cada valor dos indivíduos da população.

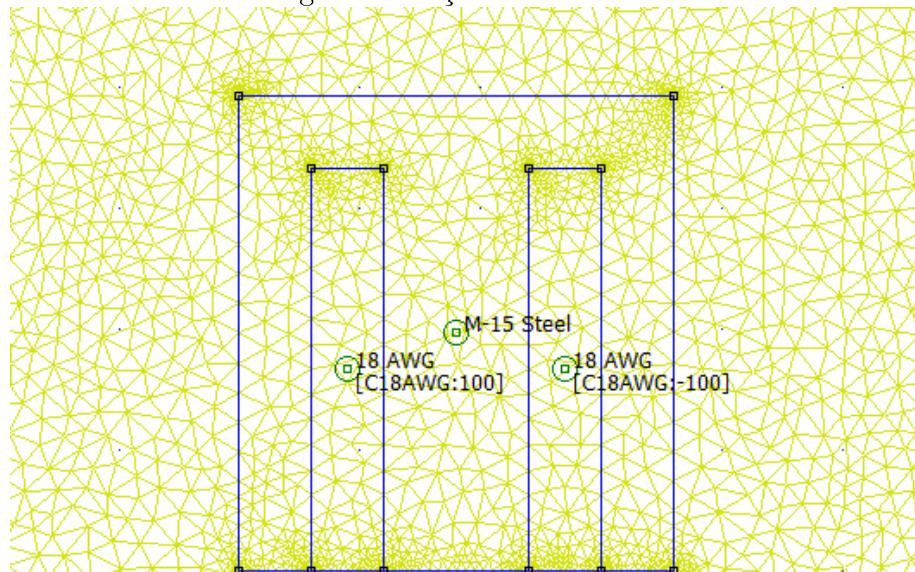
Algoritmo 4: Configurações de elementos para a simulação

```
mi_probdef(0,'inches','planar',1E-8,0.5,30,0)
mi_getmaterial('Air')
mi_getmaterial('18 AWG')
mi_getmaterial('aço')
mi_addcircprop('C18AWG', 1, 1)
```

Algoritmo 5: Algoritmo de montagem do E

```
mi_addnode(0,0)
mi_addnode(0.25,0)
mi_addsegment(0,0,0.25,0)
...
mi_addsegment(0,0.75,0,0)
mi_addsegment(0.25,0,0.5,0)
mi_addsegment(1,0,1.25,0)
```

Figura 9: Peça E do indutor



Fonte: Autor

Algoritmo 6: Montagem da estrutura I

```
variancia=-1*pop[varredura]
mi_addblocklabel(0.75,variancia-0.25/2)
mi_drawrectangle(0,variancia,1.5,variancia-0.25)
mi_selectlabel(0.75,variancia-0.25/2)
mi_setblockprop('junior', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()
```

A condição de contorno (Algoritmo 7), assim como o E do indutor, foi deixada fixa por não haver a necessidade de realocamento da posição.

Algoritmo 7: Condição de contorno

```

mi_addnode(3,0)
mi_addnode(-1.5,0)
mi_addarc(3,0.25,-1.5,0,180,1)
mi_addarc(-1.5,0.25,3,0,180,1)
mi_selectarcsegment(0.75,2.25)
mi_selectarcsegment(0.75,-2.25)
mi_addboundprop('zero', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
mi_setarcsegmentprop(1, 'zero', 1, 0)
mi_clearselected()

```

A função *fitness* adotada visa apenas gerar a indutância desejada pela distância do entreferro, apresentada pelo Algoritmo 8.

Algoritmo 8: Função Fitness

```

1 corrente, Voltagedrop, fluxx = mo_getcircuitproperties('C18AWG')
2 funfit=sqrt(abs(0,0006 - fluxx/corrente))

```

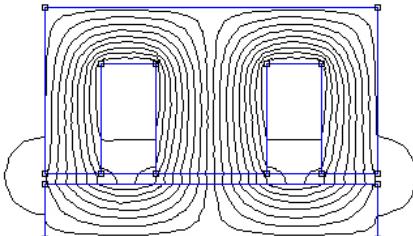
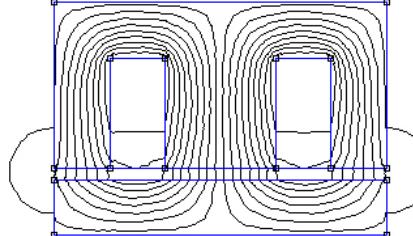
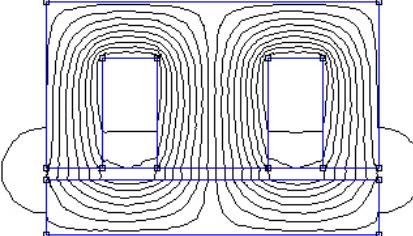
O AG foi configurado com os seguintes valores padrões:

- População: 400
- Gerações: 50
- Número de indivíduos de torneio: 3^2
- Probabilidade de recombinação: 0,95 ou 95% (na falha, os pais selecionados são copiados para a próxima geração)
- Probabilidade de mutação: 0,15 ou 15%
- Atualização da geração com troca dos pais, pelos filhos.

A Tabela 3 apresenta os valores encontrados das gerações durante a execução do *software*.

²Três indivíduos são selecionados aleatoriamente e, logo após, é observado aquele que possui a melhor aptidão. O vencedor é enviado para o próximo processo.

Tabela 3: Evolução

| Época | Esquema | Entreferro (polegada) | Indutância (H) |
|-------|---|-----------------------|----------------|
| 1 |  | 0,043 | 0,000644600 |
| 5 |  | 0,048 | 0,000600039 |
| 14 |  | 0,0475 | 0,000600000 |

Os resultados encontrados após a otimização para o circuito do indutor apresentado foram:

$$\text{Total current} = 1 \text{ Amps}$$

$$\text{Voltage Drop} = 0,052116 \text{ Volts}$$

$$\text{Flux Linkage} = 0,0006 \text{ Webers}$$

$$\text{Flux/Current} = 0,0006 \text{ Henries}$$

$$\text{Voltage/Current} = 0,052116 \text{ Ohms}$$

$$\text{Power} = 0,052116 \text{ Watts}$$

A distância do entreferro adotada para o mesmo cenário do cálculo analítico foi de 0,0475", sendo distante do valor encontrado inicialmente, que foi de 0,0319", gerando um erro de 0,0156".

Com o *software* foi possível também determinar se a configuração para a fabricação do indutor é capaz de atingir a indutância desejada. Para o mesmo cenário, vamos fazer a verificação se o indutor é capaz de atingir os seguintes valores de indutância:

- 0,7 mH
- 0,8 mH

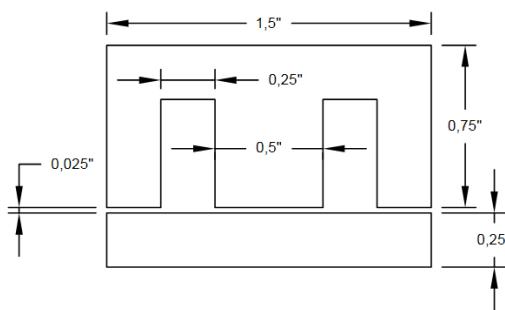
Após análise, verificou-se que para todas as medidas solicitadas o indutor atende, alterando apenas o valor do entreferro. Foram encontrados os seguintes resultados:

- 0,7 mH (0,0386")
- 0,8 mH (0,033")

4.3 Otimização da indutância por meio das dimensões

Para a demonstração do método proposto, foram consideradas as medidas e os materiais usados no projeto do indutor, conforme demonstrado na Figura 10 e Tabela 4 a seguir. Dessa forma, ele foi simulado no *software* FEMM para o levantamento do fluxo magnético e seu valor de impedância.

Figura 10: Indutor adotado para a otimização



Fonte: Autor

4.3.1 Cálculo Analítico do Modelo Proposto

Para demonstrar a precisão dos elementos finitos, agora serão comparados os valores da indutância calculados de forma analítica com os gerados por meio dos elementos finitos. A determinação da indutância é dada por:

Tabela 4: Dados do indutor adotado
Características

| | |
|-----------------------|------------|
| Largura: | 1.5" |
| Altura: | 1" |
| Profundidade: | 1" |
| Número de espiras: | 66 |
| Corrente do Circuito: | 1 A |
| Entreferro: | 0.025" |
| Condutor: | 18 AWG |
| Material Núcleo: | M-15 Steel |

$$\begin{aligned}\mathfrak{R} &= \frac{l_g}{\mu_0 \cdot a_{\text{pole}}} \\ L &= \frac{n^2}{\mathfrak{R}}\end{aligned}\tag{35}$$

em que:

\mathfrak{R} : Relutância magnética;

l_g : Entreferro;

μ_0 : Permeabilidade magnética;

n : Número de espiras;

L : Indutância;

$$\begin{aligned}\mathfrak{R} &= \frac{2 \cdot 0,025}{\mu_0 \cdot \mu_r \cdot a_{\text{pole}}} = 3,2 \cdot 10^6 \\ L &= \frac{n^2}{\mathfrak{R}} = 1,36 \text{ mH}\end{aligned}\tag{36}$$

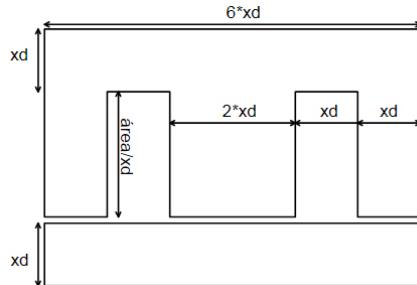
Se for comparado com os resultados, usando o método de elementos finitos, tem-se um valor de indutância igual a 1,78418 mH.

4.3.2 Modelo de Estrutura

Após definidas as dimensões e materiais, foi desenvolvido o algoritmo em linguagem Lua, para realizar a montagem bidimensional do indutor, que segue as condições de me-

didas apresentadas na Figura 11.

Figura 11: Estrutura do Indivíduo



Fonte: Autor

4.3.3 Função Fitness

Para a função *fitness* do algoritmo, foram levados em consideração o valor da indutância em Henry e a densidade de fluxo em Tesla.

$$\text{erro}_{\text{Henry}} = \sqrt{\left(\frac{\text{Henry}_{\text{Alvo}} - \text{Indivíduo}_{\text{AG}}}{\text{Henry}_{\text{Alvo}}} \right)^2} \quad (37)$$

$$\text{erro}_{\text{DF}} = \sqrt{\left(\frac{\text{DF}_{\text{Alvo}} - \text{Indivíduo}_{\text{AG}}}{\text{DF}_{\text{Alvo}}} \right)^2} \quad (38)$$

$$\text{Função}_{\text{fitness}} = NP_1 \cdot \text{erro}_{\text{Henry}} + NP_2 \cdot \text{erro}_{\text{DF}} \quad (39)$$

em que:

DF: Densidade de Fluxo;

NP_n : Nível de prioridade;

4.3.4 Estrutura do Cromossomo

A estrutura do cromossomo desenvolvida possui apenas três elementos, sendo eles: a profundidade, a largura e a distância do entreferro (EF).

$$\text{Estrutura}_{\text{cromossomo}} = [\text{Profundidade}, \text{Largura}, \text{EF}] \quad (40)$$

| Dimensões do indivíduo inicial | |
|--------------------------------|---------------------------------|
| Altura: | 1.01" |
| Largura: | 1.68" |
| Profundidade: | 2.44" |
| Janela | Altura: 0.45" Largura: 0.28" |

Tabela 5: Dimensões do melhor indivíduo da primeira época

4.3.5 Parâmetros do AG proposto

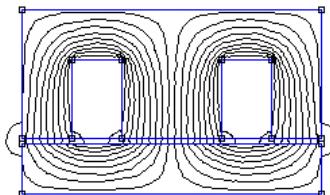
O AG foi configurado com os seguintes valores padrões:

- População: 200
- Gerações: 50
- Probabilidade de recombinação: 0,95 (na falha, os pais selecionados são copiados)
- Probabilidade de mutação: 0,15
- Atualização: substituição dos pais pelos filhos

4.3.6 Resultados

Após a primeira época, foi apresentado o primeiro indivíduo mais apto (Figura 12), o qual apresentou um valor de volume igual a $2,17 \text{ in}^3$, indutância de 4,6853 mH e densidade de fluxo de 0,05961 Tesla. Suas dimensões são apresentadas na Tabela 5.

Figura 12: Indivíduo inicial

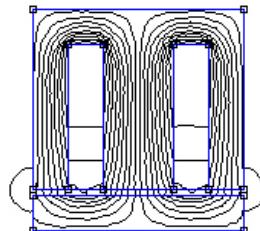


Fonte: Autor

Na quinta geração, foi notada uma grande variação do seu formato juntamente com o seu volume. Ele teve a sua área de base comprimida e sua altura elevada (devido à exigência de aumentos de densidade de fluxo magnético), passando para um volume de

1,09 in³, uma indutância de 1,795 mH e densidade de fluxo igual a 0,1076 Tesla, conforme Figura 13 e Tabela 6.

Figura 13: Indivíduo intermediário



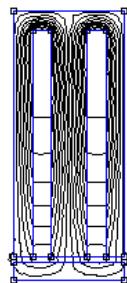
Fonte: Autor

| Dimensões do 6º indivíduo | | |
|---------------------------|---------------------------------|--|
| Altura: | 1.06" | |
| Largura: | 1.04" | |
| Profundidade: | 1.17" | |
| Janela | Altura: 0.72" Largura: 0.17" | |

Tabela 6: Dimensões do melhor indivíduo da sexta época

O algoritmo estabilizou a partir da sexta época, atingindo o valor da indutância desejada. Tendo um volume final de 0,7929 in³, densidade de fluxo igual a 0,2374 Tesla e um valor de indutância de 1,78418 mH, que era justamente o valor alvo desejado. Sua estrutura é apresentada na Figura 14 e Tabela 7.

Figura 14: Indivíduo otimizado encontrado



Fonte: Autor

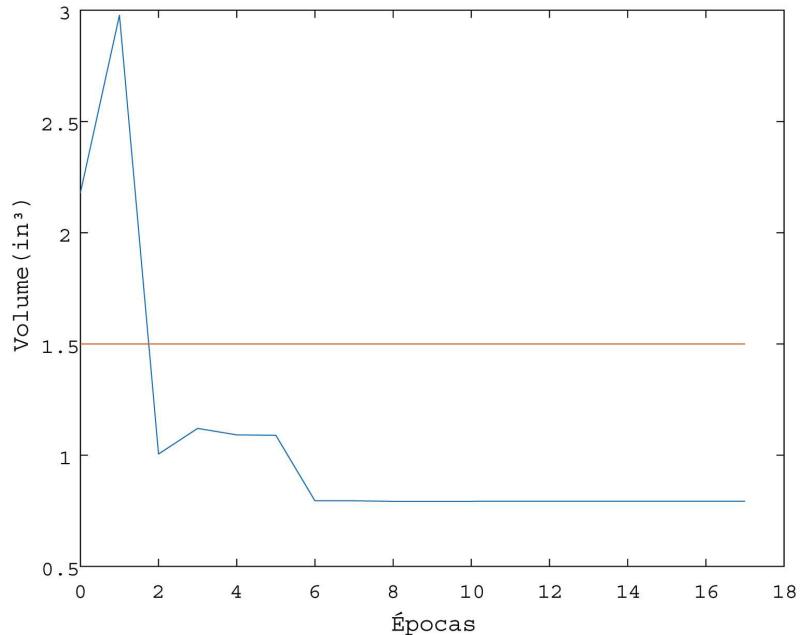
Tabela 7: Dimensões do melhor indivíduo da décima época

| Dimensões do indivíduo otimizado | | |
|----------------------------------|--------------------|--|
| Altura: | 1.45" | |
| Comprimento: | 0.60" | |
| Profundidade: | 0.98" | |
| Janela | Altura: 1.25" | |
| | Comprimento: 0.10" | |

4.3.7 Variância Volumétrica da Estrutura

A Figura 15 a seguir demonstra o gráfico de comportamento do volume dos melhores indivíduos encontrados durante o processo de otimização.

Figura 15: Gráfico de variação do volume



Fonte: Autor

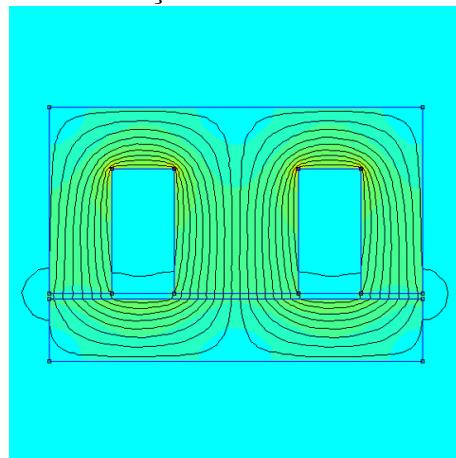
4.3.8 Validação do modelo

Para realizar a validação do modelo encontrado, montou-se a estrutura do indutor escolhido para a otimização, realizando sua simulação no *software* FEMM (Figuras 16 e 17), e levantando os valores apresentados a seguir:

- Total current = 1 Amps

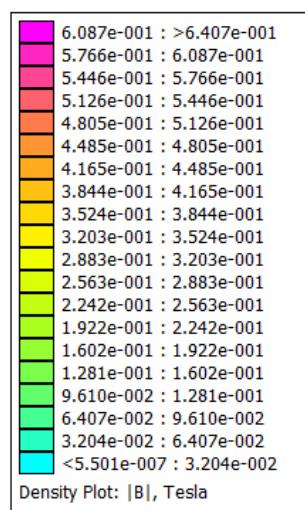
- Voltage Drop = 0,0701971 Volts
- Flux Linkage = 0,00178418 Webers
- Flux/Current = 0,00178418 Henries
- Voltage/Current = 0,0701971 Ohms
- Power = 0,0701971 Watts

Figura 16: Simulação do indutor a ser otimizado



Fonte: Autor

Figura 17: Tabela de densidade de fluxo

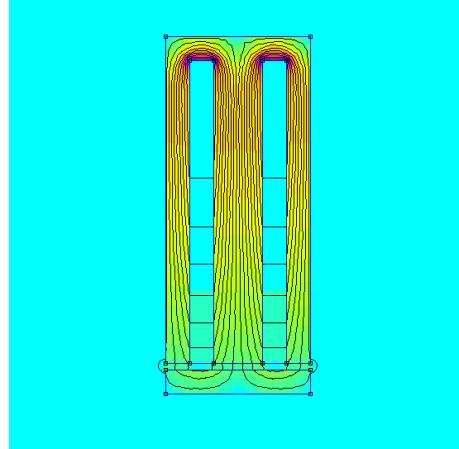


Fonte: Autor

Logo após, levantados os valores do indutor alvo, foi realizada a simulação do indutor idealizado pelo algoritmo genético (Figuras 18 e 19) e seus dados são apresentados a seguir:

- Total current = 1 Amps
- Voltage Drop = 0,0687189 Volts
- Flux Linkage = 0,00178417 Webers
- Flux/Current = 0,00178417 Henries
- Voltage/Current = 0,0687189 Ohms
- Power = 0,0687189 Watts

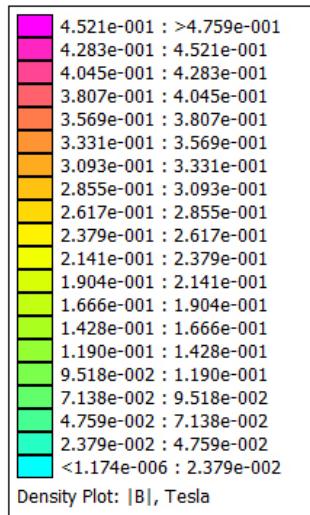
Figura 18: Simulação do indutor proposto



Fonte: Autor

Dessa forma, comprova-se que o valor da indutância foi o mesmo, mas com uma redução, de aproximadamente, 50% do indutor inicial.

Figura 19: Tabela de densidade de fluxo



Fonte: Autor

CAPÍTULO 5

5 Conclusão

Uma das grandes dificuldades encontradas durante a fase de produção de um elemento magnético está na fase de protótipo, pois há uma expressiva discrepância entre os valores calculados de forma analítica e os extraídos do elemento físico, fator que provoca gastos devido à confecção de protótipos com o objetivo de atingir as características desejadas. Isso se deve ao fato de as condições adotadas no cálculo desconsiderarem características que são de grande relevância, como a dispersão de fluxo e os efeitos de borda. Por esse motivo, adotou-se o método dos elementos finitos para adicionar tais características e gerar uma simulação mais fidedigna à realidade. Mas o cenário atual não exige apenas um protótipo bem dimensionado, e sim um que possua a característica desejada juntamente com a otimização do uso de materiais utilizados em sua confecção, o que se tornou a porta de entrada para os algoritmos genéticos, que têm como característica realizar uma otimização combinatória e, dessa forma, encontrar as configurações que atendam ao projeto e reduzam a quantidade de material adotado. Para a comprovação do método desenvolvido elaboraram-se três etapas:

- Dimensões e números de espiras de um indutor de núcleo de ar.
- Determinação da distância ideal do entreferro de um indutor E-I.
- Redução do volume do indutor E-I preservando o valor da indutância.

As simulações apresentadas na seção anterior, apresentaram resultados promissores para a aplicação do método proposto, fato que foi validado por meio dos resultados encontrados, onde apresentaram uma redução no número de espiras e volume de material utilizado em sua manufatura. O algoritmo mostrou-se robusto, uma vez que todas as simulações convergiram praticamente para as mesmas dimensões. Para estudo futuro, o método também pode ser extrapolado para problemas eletrostáticos, de fluxo de corrente e de fluxo de calor.

Referências

- [1] Schmidt, H., Simulation von Elektromagneten mit FEMM und Modelica, in Simulation von Elektromagneten mit FEMM und Modelica. p. 1-10.<https://doi.org/10.3139/9783446460706.fm>
- [2] T. Benamimour, A. Bentounsi and H. Djeghloud, "CAD of electrical machines using coupled FEMM-MATLAB softwares,"2013 3rd International Conference on Electric Power and Energy Conversion Systems, Istanbul, Turkey, 2013, pp. 1-6, doi: 10.1109/EPECS.2013.6712995.
- [3] A. Suhaimi and H. S. A. Halim, "Failure analysis of cable accessories using Finite Element Method Magnetics (FEMM) software,"2012 IEEE International Conference on Condition Monitoring and Diagnosis, Bali, Indonesia, 2012, pp. 995-998, doi: 10.1109/CMD.2012.6416323.
- [4] Z. Zakaria, M. S. B. Mansor, A. H. Jahidin, M. S. Z. Azlan and R. A. Rahim, "Simulation of magnetic flux leakage (MFL) analysis using FEMM software,"2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA), Penang, Malaysia, 2010, pp. 481-486, doi: 10.1109/ISIEA.2010.5679417.
- [5] COLEY, David A. An introduction to genetic algorithms for scientists and engineers. World Scientific Publishing Co Inc, 1999. <https://doi.org/10.1142/3904>
- [6] Deb, Kalyanmoy; Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148. doi: 10.1142/3904
- [7] D.C. Meeker, "Improvised asymptotic boundary conditions for electrostatic finite elements,"*IEEE Transactions on Magnetics*, 50(6):7400609, June 2014. <http://dx.doi.org/10.1109/TMAG.2014.2300196>
- [8] BASTOS, J. P. A. Eletromagnetismo para engenharia – estática e quase estática. 3. ed. Florianópolis: UFSC, 2012. 396 p.
- [9] GRUPO DE CONCEPÇÃO E ANÁLISE DE DISPOSITIVOS ELETROMAGNÉTICOS. Manual do EFCAD. Florianópolis: UFSC, 2002. Disponível em:

<<http://www.grucad.ufsc.br/>>. Acesso em: 02 mar. 2015. IDA, N.; BASTOS, J. P. A. Electromagnetics and calculation of fields. 2. ed. New York: Springer, 1997.

- [10] C. N. Obiozor and M. N. O. Sadiku, "Finite elements in electromagnetics for undergraduate curriculum,"Proceedings of Southeastcon '93, Charlotte, NC, USA, 1993, pp. 4 p.-, doi: 10.1109/SECON.1993.465779.
- [11] MEEKER, David. Finite Element Method Magnetics: Version 4.2 User's Manual. [S. l.]: [s. n.], 2018. Disponível em: <https://www.femm.info/wiki/HomePage> . Acesso em: 31 out. 2025.
- [12] VELOSO, C. H. L. Desenvolvimento de Uma Ferramenta de Ensino da Equação de Laplace Usando Uma Nova Abordagem do Método de Elementos Finitos. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 41., 2013, Maranhão. Anais eletrônicos... Maranhão, 2013. Disponível em: <http://www.fadep.br/engenharia-eletrica/congresso/pdf/117634_1.pdf>. Acesso em: 10 jun. 2021.
- [13] Chubarova, Irina Nikolayevna, "Modeling Cholera Using Engineered Genetic Algorithm"(2015). Theses and Dissertations. 374. <https://ir.library.illinoisstate.edu/etd/374> <http://doi.org/10.30707/ETD2015.Chubarova.I>
- [14] Brooks, Christopher, 'A Genetic Algorithm to Design Optimal Patch Configurations Using Raster Data Structures', Spatial Evolutionary Modeling (New York, 2001; online edn, Oxford Academic, 12 Nov. 2020), <https://doi.org/10.1093/oso/9780195135688.003.0013>, accessed 16 Oct. 2025.
- [15] Deng, Pi-Sheng. "Genetic Algorithm Applications to Optimization Modeling."Encyclopedia of Artificial Intelligence, edited by Juan Ramón Rabanal Dopico, et al., IGI Global Scientific Publishing, 2009, pp. 748-754. <https://doi.org/10.4018/978-1-59904-849-9.ch111>

- [16] Arabas, J.; Kozdrowski, S. (2001). Applying an Evolutionary Algorithm to Telecommunication Network Design. *IEEE Transactions on Evolutionary Computation*, 4(5), 309–322. DOI: 10.1109/4235.942526
- [17] Ballester, P. J.; Carter, J. N. (2004). An Effective Real-Parameter Genetic Algorithm with Parent Centric Normal Crossover for Multimodal Optimisation. In *Proceedings of GECCO 2004* (Lecture Notes in Computer Science), pp. 901–913. DOI: 10.1007/978-3-540-24854-5
- [18] Del Rosso, C. (2006). Reducing Internal Fragmentation in Segregated Free Lists Using Genetic Algorithms. In *Proceedings of the 2006 ACM Workshop on Interdisciplinary Software Engineering Research*. DOI: 10.1145/1137661.1137674
- [19] Pierre, S.; Legault, G. (1998). A Genetic Algorithm for Designing Distributed Computer Network Topologies. *IEEE Transactions on Systems, Man, and Cybernetics. Part B*, 28(2), 249–258. DOI: 10.1109/3477.662766
- [20] Adil Amirjanov, Modeling the Dynamics of a Changing Range Genetic Algorithm, Procedia Computer Science, Volume 102, 2016, Pages 570-577, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2016.09.444>.

6 Anexo 1

6.1 Comentário

O algoritmo implementa um método evolutivo para otimizar o desempenho de um núcleo magnético tipo E-I modelado no FEMM. Ele inicia gerando uma população aleatória de soluções, onde cada indivíduo representa uma variação geométrica da parte móvel do núcleo. Em cada iteração, o algoritmo avalia cada indivíduo, executando uma simulação magnética e calculando uma função de custo baseada na diferença entre o fluxo obtido e o desejado.

[Algorítimo para determinação do entre ferro.]

```
-- Inicialização de variáveis básicas  
menorvalor=1000;          -- Armazena o menor valor de custo encontrado  
contt=0                    -- Contador para salvar figuras resultantes  
pop={}                     -- Vetor que representa a população de soluções  
npop=50                   -- Número de indivíduos na população  
  
-- Geração aleatória da população inicial  
for i=1,npop do  
pop[i]=random(0,500)/1000  -- Gera valores aleatórios entre 0 e 0.5  
end
```

```
loop=20          -- Número de iterações (épocas)
custo={}
-- Vetor que armazenará os custos calculados para cada indivíduo

-- LOOP PRINCIPAL DE EVOLUÇÃO
for epoca=1,loop do
-- Avaliação de cada indivíduo na população
for varredura=1,npop do
newdocument(0)           -- Cria um novo documento FEMM
mi_probdef(0,'inches','planar',1E-8,1,30,0) -- Define o problema magnético: unidade, tipo, precisão e dimensões
8
-- Carrega materiais usados na simulação
mi_getmaterial('Air')      -- Ar
mi_getmaterial('18 AWG')    -- Fio de cobre 18 AWG
mi_getmaterial('junior')     -- Material do núcleo magnético
mi_addcircprop('C18AWG', 1, 1) -- Define uma bobina com 1 A de corrente

-- Desenho da parte "E" do núcleo magnético
mi_addnode(0,0)           -- Adiciona nó inicial
mi_addnode(0.25,0)
mi_addsegment(0,0,0.25,0)
```

```
mi_addnode(0.25,0.5)
mi_addsegment(0.25,0,0.25,0.5)
mi_addnode(0.5,0.5)
mi_addsegment(0.25,0.5,0.5,0.5)
mi_addnode(0.5,0)
mi_addsegment(0.5,0.5,0.5,0)
mi_addnode(1,0)
mi_addsegment(0.5,0,1,0)
mi_addnode(1,0.5)
mi_addsegment(1,0,1,0.5)
mi_addnode(1.25,0.5)
mi_addsegment(1,0.5,1.25,0.5)
mi_addnode(1.25,0)
mi_addsegment(1.25,0.5,1.25,0)
mi_addnode(1.5,0)
mi_addsegment(1.25,0,1.5,0)
mi_addnode(1.5,0.75)
mi_addsegment(1.5,0,1.5,0.75)
mi_addnode(0,0.75)
mi_addsegment(1.5,0.75,0,0.75)
```

```
mi_addsegment(0,0.75,0,0)

-- Inserção de blocos com materiais e bobinas

mi_adddblocklabel(0.75,0.6)           -- Núcleo principal
mi_adddblocklabel(0.36,0.25)          -- Bobina 1
mi_adddblocklabel(1.12,0.25)          -- Bobina 2

-- Define propriedades do núcleo magnético
mi_selectlabel(0.75,0.6)
mi_setblockprop('junior', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()

-- Define propriedades da primeira bobina (corrente positiva)
mi_selectlabel(0.36,0.25)
mi_setblockprop('18 AWG', 0, 0.05, 'C18AWG', 0, 0, 66)
mi_clearselected()

-- Define propriedades da segunda bobina (corrente negativa)
mi_selectlabel(1.12,0.25)
mi_setblockprop('18 AWG', 0, 0.05, 'C18AWG', 0, 0, -66)
```

```
mi_clearselected()

-- Desenho da parte "I" (móvel, variável com o indivíduo)
variancia=-1*pop[varredura]           -- Define deslocamento com base no indivíduo
mi_addblocklabel(0.75,variancia-0.25/2)
mi_drawrectangle(0,variancia,1.5,variancia-0.25)
mi_selectlabel(0.75,variancia-0.25/2)
mi_setblockprop('junior', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()
```

15

```
-- Adição do ar e fronteiras externas
mi_addblocklabel(0.75,1.5)
mi_selectlabel(0.75,1.5)
```

```
mi_setblockprop('Air', 0, 0.05, 0, 0, 0, 0)
```

```
mi_clearselected()
```

```
mi_addnode(2.5,0)
```

```
mi_addnode(-1,0)
```

```
mi_addarc(2.5,0.25,-1,0,180,1)
```

```
mi_addarc(-1,0.25,2.5,0,180,1)
```

```

mi_selectarcsegment(0.75,1.75)
mi_selectarcsegment(0.75,-1.75)
mi_addboundprop('zero', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
mi_setarcsegmentprop(1, 'zero', 1, 0)
mi_clearselected()

-- Execução da análise magnética
mi_saveas('trab_final.FEM')          -- Salva o arquivo do modelo
mi_createmesh()                      -- Gera a malha
mi_analyze(0)                        -- Executa a simulação
mi_loadsolution()                    -- Carrega a solução

-- Cálculo da função de custo
corrente,Voltagedrop,fluxx=mo_getcircuitproperties('C18AWG') -- Obtém propriedades do circuito
funfit=sqrt((0.00173-fluxx/corrente)^2)      -- Função de custo baseada na diferença de fluxo/corrente
mi_close(0)                                -- Fecha o editor de modelo
mo_close(0)                                 -- Fecha o pós-processador
custo[varredura]=funfit                     -- Armazena o custo do indivíduo
end

```

```
-- Busca pelo menor valor (melhor indivíduo da geração)
for varredura=1,npop do
if custo[varredura]<menorvalor then
menorvalor=custo[varredura]
valorpop=pop[varredura]
contt=contt+1
end
end
```

cc -- Seleção por torneio (3 indivíduos aleatórios)

```
popnew={}
for varredura=1,npop do
t1=random(1,npop)
t2=random(1,npop)
t3=random(1,npop)
if custo[t1]<=custo[t2] then
if custo[t1]<=custo[t3] then
popnew[varredura]=pop[t1]
else
popnew[varredura]=pop[t3]
```

```
end
else
if custo[t2]<=custo[t3] then
popnew[varredura]=pop[t2]
else
popnew[varredura]=pop[t3]
end
end
end
```

45

```
-- Cruzamento Simulado Binário (SBX)
eta=5
for i=1,npop,2 do
recon=random(0,100)/100           -- Probabilidade de cruzamento
if recon>0.15 then                -- Executa cruzamento em 85% dos casos
u = random(0,100)/100            -- Número aleatório entre 0 e 1
if u <= 0.5 then
beta=(2*u)^(1/(eta+1))
else
beta=(1/(2*(1-u)))^(1/(eta+1))
```

```

end

filho1 = 0.5*((1+beta)*popnew[i] + (1-beta)*popnew[i+1]) -- Gera primeiro filho
filho2 = 0.5*((1-beta)*popnew[i] + (1+beta)*popnew[i+1]) -- Gera segundo filho
if filho1<=0 then filho1=0 end -- Garante limite inferior
if filho2<=0 then filho2=0 end
popnew[i]=filho1
popnew[i+1]=filho2
end
end

```

CC

```

-- Mutação (ocorre com 5% de probabilidade)

for i=1,npop do
muta = random(0,100)/100
if muta>0.95 then
popnew[i]=random(0,500)/1000 -- Novo valor aleatório (mutação)
end
end

```

```

-- Atualização da população com elitismo (mantém o melhor)

for i=1,npop-1 do

```

```
pop[i]=popnew[i]
end
pop[npop]=valorpop
end

-- Finalização do processo
messagebox('End System')          -- Exibe mensagem de conclusão
messagebox(valorpop)              -- Mostra o melhor valor encontrado
```

7 Anexo 2

7.1 Comentário

O algoritmo implementa uma otimização evolutiva para ajustar as dimensões e densidade de fluxo de um núcleo magnético tipo E-I usando o FEMM. Ele inicia criando uma população inicial aleatória, onde cada indivíduo define deslocamentos e densidades geométricas do núcleo e das bobinas. Para cada indivíduo, o algoritmo executa simulações magnéticas e calcula uma função de custo baseada na densidade de fluxo e indutância desejadas. Os melhores indivíduos são selecionados usando torneio e cruzamento SBX, gerando novos indivíduos para a próxima geração. Uma mutação aleatória ocasional é aplicada para manter diversidade genética. O processo é repetido por várias iterações até convergir. Um elitismo implícito garante que o melhor indivíduo seja preservado. Ao final, o algoritmo retorna o melhor conjunto de parâmetros do núcleo e das bobinas. Assim, combina simulação por elementos finitos com técnicas de algoritmos genéticos para otimização automatizada.

[Algoritmo para otimização da estrutura do indutor.]

```
popd = {}                      -- População de densidade
npop = 100                      -- Tamanho da população

-- Geração inicial de indivíduos
for i = 1, npop do
    pop[i] = random(10, 40) / 100
    popd[i] = random(100, 400) / 100
end

loop = 50                         -- Número de épocas (iterações principais)
custo = {}                         -- Vetor para armazenar o custo de cada indivíduo

-- Início das iterações principais
for epoca = 1, loop do
    for varredura = 1, npop do

        newdocument(0)
        xd = pop[varredura]
        area = 0.125
        variancia = -0.025
```

```
-- Definição do problema no FEMM  
mi_probdef(0, 'inches', 'planar', 1E-8, popd[varredura], 30, 0)  
mi_getmaterial('Air')  
mi_getmaterial('18 AWG')  
mi_getmaterial('M-15 Steel')  
mi_addcircprop('C18AWG', 1, 1)
```

```
-- Desenho geométrico do núcleo tipo E-I  
65  
mi_addnode(0, 0)  
mi_addnode(xd, 0)  
mi_addsegment(0, 0, xd, 0)  
mi_addnode(xd, area / xd)  
mi_addsegment(xd, 0, xd, area / xd)  
-- (continua repetição de segmentos para formar o núcleo...)
```

```
-- Adiciona material e define propriedades de cada região  
mi_addblocklabel(6 * xd / 2, variancia - xd / 2)  
mi_drawrectangle(0, variancia, 6 * xd, variancia - xd)  
mi_selectlabel(6 * xd / 2, variancia - xd / 2)
```

```
mi_setblockprop('M-15 Steel', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()

-- Define bobinas e ar
mi_adddblocklabel(6 * xd / 2, (area / xd + xd) / 2)
mi_setblockprop('M-15 Steel', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()

mi_selectlabel(xd / 2 + xd, area / (2 * xd))
60
mi_setblockprop('18 AWG', 0, 0.05, 'C18AWG', 0, 0, bobinas)
mi_clearselected()

-- Região de ar
mi_adddblocklabel(0.75, 1.5)
mi_selectlabel(0.75, 1.5)
mi_setblockprop('Air', 0, 0.05, 0, 0, 0, 0)
mi_clearselected()

-- Análise no FEMM
mi_saveas('trab_final.FEM')
```

```
mi_createmesh()
mi_analyze(0)
mi_loadsolution()

lx = 6 * xd / 2 -- Cálculo de coordenada X para análise da densidade de fluxo;
ly = (area / xd + xd) / 2 -- Cálculo de coordenada Y para análise da
densidade de fluxo;
V, Bx, By = mo_getpointvalues(lx, ly) --Recuperando Valores

-- Funções de aptidão (avaliação do indivíduo)
funfit1 = sqrt((1.75 - By)^2) --1.75 Densidade de Fluxo Escolhida
corrente, Voltagedrop, fluxx = mo_getcircuitproperties('C18AWG')
funfit2 = sqrt((0.005 - fluxx / corrente)^2) -- 0.005 H Indutância Desejada;
funfit = funfit1 + funfit2

custo[varredura] = funfit

mi_close(0)
mo_close(0)
end
```

```
-- Seleção do melhor indivíduo  
for varredura = 1, npop do  
  if custo[varredura] < menorvalor then  
    menorvalor = custo[varredura]  
    valorpop = pop[varredura]  
    valorpopd = popd[varredura]  
  end  
end
```

62

```
-- Cruzamento e mutação (Simulated Binary Crossover - SBX)  
eta = 4  
for i = 1, npop, 2 do  
  recon = random(0, 100) / 100  
  if recon > 0.15 then  
    u = random(0, 100) / 100  
    if u <= 0.5 then  
      beta = (2 * u)^(1 / (eta + 1))  
    else  
      beta = (1 / (2 * (1 - u)))^(1 / (eta + 1))
```

end

filho1 = 0.5 * ((1 + beta) * pop[i] + (1 - beta) * pop[i + 1])

filho2 = 0.5 * ((1 - beta) * pop[i] + (1 + beta) * pop[i + 1])

filho3 = 0.5 * ((1 + beta) * popd[i] + (1 - beta) * popd[i + 1])

filho4 = 0.5 * ((1 - beta) * popd[i] + (1 + beta) * popd[i + 1])

-- Garantindo limites mínimos

if filho1 <= 0.1 then filho1 = 0.1 end

if filho2 <= 0.1 then filho2 = 0.1 end

if filho3 <= 0.1 then filho3 = 0.1 end

if filho4 <= 0.1 then filho4 = 0.1 end

pop[i] = filho1

pop[i + 1] = filho2

popd[i] = filho3

popd[i + 1] = filho4

end

end

```
-- Mutação aleatória de alguns indivíduos  
for i = 1, npop do  
    muta = random(0, 100) / 100  
    if muta > 0.95 then  
        pop[i] = random(10, 40) / 100  
        popd[i] = random(100, 400) / 100  
    end  
end
```

end -- Fim do loop principal

```
messagebox('End System')  
messagebox(valorpop)
```