Adriel Vinícius Morais Araújo
Otimização Multiobjetivo e Simulação Computacional
de Problema de Mecânica Estrutural
Timóteo
2022

#### Adriel Vinícius Morais Araújo

# Otimização Multiobjetivo e Simulação Computacional de Problema de Mecânica Estrutural

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Centro Federal de Educação Tecnológica de Minas Gerais

Campus Timóteo

Graduação em Engenharia de Computação

Orientador: João Batista Queiroz Zuliani

Coorientador: Almir Silva Neto

Timóteo

2022

#### Adriel Vinícius Morais Araújo

#### Otimização Multiobjetivo e Simulação Computacional de Problema de Mecânica Estrutural

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 15 de fevereiro de 2022:

Prof. Dr. João Batista Queiroz Zuliani Orientador

> Prof. Dr. Almir Silva Neto Coorientador

Prof. Dr. Dênis Emanuel da Costa Vargas Professor Convidado

Prof. Me. Douglas Nunes de Oliveira Professor Convidado

> Timóteo 2022

#### MINISTÉRIO DA EDUCAÇÃO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS SISTEMA INTEGRADO DE PATRIMÔNIO, **CEFET-MG** ADMINISTRAÇÃO E CONTRATOS

#### FOLHA DE ASSINATURAS

Emitido em 15/02/2022

#### FOLHA DE ROSTO (PLATAFORMA BRASIL) Nº 1/2022 - DFGTM (11.63.03)

 $(N^o$  do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 15/02/2022 22:14) ALMIR SILVA NETO

PROFESSOR ENS BASICO TECN TECNOLOGICO DMQTM (11.63.04) Matrícula: 1036181

(Assinado digitalmente em 16/02/2022 15:56 ) DOUGLAS NUNES DE OLIVEIRA

PROFESSOR ENS BASICO TECN TECNOLOGICO DCCTM (11.63.05) Matrícula: 2921288

(Assinado digitalmente em 16/02/2022 21:34)

Adriel Vinicius Morais Araújo DISCENTEMatrícula: 201717060056

(Assinado digitalmente em 15/02/2022 21:55) DENIS EMANUEL DA COSTA VARGAS

PROFESSOR ENS BASICO TECN TECNOLOGICO DM (11.56.11) Matrícula: 1551720

(Assinado digitalmente em 15/02/2022 21:54) JOAO BATISTA QUEIROZ ZULIANI PROFESSOR ENS BASICO TECN TECNOLOGICO

DFGTM (11.63.03) Matrícula: 1324186

Para verificar a autenticidade deste documento entre em <a href="https://sig.cefetmg.br/documentos/">https://sig.cefetmg.br/documentos/</a> informando seu número: 1, ano: 2022, tipo: FOLHA DE ROSTO (PLATAFORMA BRASIL), data de emissão: 15/02/2022 e o código de verificação: 6fb26c0f74

.

.

## Agradecimentos

Agradeço primeiramente a Deus por tudo o que tenho, pois acredito firmemente que sem Ele nada poderia fazer, à Ele a glória eternamente e amém. Em segundo lugar, gostaria de agradecer aos meus pais, que me criaram, ensinaram e sempre investiram mais do que poderiam na minha educação, além de agradecer a minha irmã pela alegria e carinho. Em terceiro lugar, agradeço aos professores João Batista e Almir Silva pela disposição, tempo e paciência em ensinar, orientar e ajudar ao longo desta pesquisa. Agradeço aos meus parceiros de curso que cuja paixão e competência naquilo que fazem me impulsionavam a procurar ser melhor a cada dia. Em seguida, agradeço a FAPEMIG pelo período de bolsa PIBIT que culminou neste TCC. Por fim, agradeço a todos os professores do CEFET que trabalham para fazer desse campus um ambiente de busca e produção de conhecimento.



## Resumo

Otimização Estrutural é uma técnica que busca melhorar as características de uma determinada estrutura submetida a determinadas condições de contorno. Neste trabalho, objetivou-se avaliar o desempenho do algoritmo evolucionário multiobjetivo NSGA-II para executar uma otimização estrutural de forma sobre uma mão francesa cuja modelagem envolveu a utilização de NURBS. Para isso, buscou-se comunicar a biblioteca de otimização multiobjetivo evolucionária Pymoo com o software de simulação ANSYS MAPDL. Com essa abordagem, executou-se experimentos relacionados à quantidade de pontos de controle e valores dos pesos que são utilizados na definição de NURBS que modelam a fronteira da peça. Também comparou-se alguns resultados com os de uma otimização estrutural realizada pelo software Ansys Mechanical.

Palavras-chave: Otimização Estrutural, Algoritmos Genéticos e NSGA-II.

### **Abstract**

Structural Optimization is a technique to improve the characteristics of a certain structure submitted a certains boundary conditions. In this work, the objective is to evaluate the performance of the evolutionary multiobjective algorithm NSGA-II to execute a shape structural optimization of a corner bracket using NURBS. To do this, the evolutionary multiobjetive optimization framework Pymoo was prepared to comunicate with the ANSYS MAPDL software. With this approach, experiments were performed related to the number of control points and weight values that are used in the definition of NURBS that model the boundary of the device. Furthermore, some results was compared with a structural optimization developed by Ansys Mechanical.

Keywords: Structural Optimization, Genetic Algorithm and NSGA-II.

# Lista de ilustrações

Figura 1 – Tipos de Otimização Estrutural	15
Figura 2 – Passos em um AG	19
Figura 3 – Recombinação por ponto de corte	20
Figura 4 – Operador de mutação binária	20
Figura 5 – Fluxograma do NSGA-II	22
Figura 6 – Esquema de fronteiras Pareto	22
Figura 7 — Calculo do Cuboid	23
Figura 8 - Método Geral para análise de Sistemas discretos - Sistema Discreto Padrão.	25
Figura 9 – Estrutura reticulada - Esquema de uma treliça de 10 barras	26
Figura 10 – Estrutura Tridimensional - Malha de elementos finitos de um martelo	26
Figura 11 – B-splines racionais cúbicas com peso $w_3$ variando	28
Figura 12 – Curvas racionais quadrática com peso $w_1$ variando	28
Figura 13 – Comunicação Python APDL	31
Figura 14 – Comunicação Python APDL	32
Figura 15 – Modelo final de comunicação Python APDL	32
Figura 16 – Peça de referencia da Mão francesa	34
Figura 17 – Representação das peças nos limites minimo e máximo para os pontos de	
controle	35
Figura 18 – Exemplos de peças geradas durante o processo de otimização.	35
Figura 19 - Figura representando as condições de contorno aplicadas sobre as áreas	
dos furos do suporte de esquina	36
Figura 20 – Box Plot dos tempos de execução	40
Figura 21 – Box Plot dos hipervolumes calculados a partir das Fronteiras Pareto Estimadas	. 41
Figura 22 – Exemplos da evolução dos valores de hipervolume para o maior e o menor	
valor de tempo total de execução.	42
Figura 23 – Exemplo da convergência das fronteiras estimadas para o caso de maior	
tempo de execução. Tensão em Pascal $(Pa)$ e volume em metros cúbicos	
$(m^3)$	42
Figura 24 – Convergência da Otimização para o caso de menor tempo de execução.	
Tensão em Pascal $(Pa)$ e volume em metros cúbicos $(m^3)$	43
Figura 25 – Evolução dos valores de Hipervolume para o maior e o menor valor de Hi-	
pervolume em todos os experimentos	43
Figura 26 – Convergência da Otimização com maior valor de Hipervolume. Tensão em	
Pascal $(Pa)$ e volume em metros cúbicos $(m^3)$	44
Figura 27 – Convergência da Otimização de menor valor de Hipervolume. Tensão em	
Pascal $(Pa)$ e Volume em metros cúbicos	44
Figura 28 - Fronteira Pareto Estimada do valor médio dos experimentos, peça base e	
Otimizações Automáticas. Tensão em Pascal $(Pa)$ e Volume em metros cú-	
bicos $(m^3)$	47

# Lista de tabelas

abela 1 – Especificações do Processador	30
abela 2 – Posição dos Furos na peça	34
abela 3 — Experimentos realizados	37
abela 4 – Tempos de execução (em horas) das rodadas da otimização	39
abela 5 – Hipervolume das Fronteiras Pareto Estimadas	40
abela 6 - Otimizações automáticas promovidas pelo Ansys Mechanical. Tensão em	
Pascal $(Pa)$ e volume em metros cúbicos $(m^3)$	45

# Sumário

1	INTRODUÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Otimização	17
2.1.1	Dominância e soluções eficientes	18
2.2	Algoritmos Evolucionários	18
2.2.1	Algoritmo Genético Multiobjetivo baseado em não dominância: NSGA-II	21
2.2.2	Ordenação rápida com base em não dominância	21
2.2.3	Atribuidor de distância de aglomeração	22
2.2.4	Operador de Distância de Aglomeração	23
2.2.5	Rotina principal do Algoritmo NSGA-II	23
2.3	Método dos Elementos Finitos	24
2.4	Representação geométrica por B-Splines racionais não uniformes (NURBS	s) 27
3	TRABALHOS CORRELATOS	29
4	MÉTODO DE PESQUISA	30
4.1	Classificação da Pesquisa	30
4.2	Recursos Utilizados	30
4.3	Procedimentos de pesquisa	30
4.3.1	Comunicação Python APDL	31
4.3.1.1	Da modelagem das peças	33
4.4	O Modelo	33
4.4.1	A parametrização	34
4.4.2	A Malha e as condições de contorno	35
4.5	A otimização	36
4.6	Dos Experimentos	37
4.7	Otimização Automática	38
5	RESULTADOS	39
5.1	Comparação	44
6	CONSIDERAÇÕES FINAIS	48
	REFERÊNCIAS	49

## 1 Introdução

"O temor do Senhor é o princípio da sabedoria; bom entendimento têm todos os que cumprem os seus mandamentos; o seu louvor permanece para sempre".

Salmos 111:10

A busca por produtos de melhor qualidade geralmente leva em consideração a redução do custo (final e de produção), do tempo de produção e a melhoria dos critérios de qualidade. Assim, a otimização de projeto de engenharia, que é a identificação da combinação de parâmetros de um produto, em geral tem que lidar com critérios em conflito.

Durante um longo período, para realizar essa busca, a maneira de se obter soluções mais adequadas esteve relacionada a testes físicos, que visava, por meio da experiência dos profissionais envolvidos, escolher, produzir e avaliar uma dada solução. Essa maneira de proceder era cara, lenta e limitante. Por outro lado, como alternativa aos testes físicos, o cálculo das equações analíticas dos fenômenos que modelam o problema em questão podem ser utilizadas. Contudo, prover avaliação de soluções por meio desta abordagem exige cálculos extensos e complexos, além de simplificações que tornam os resultados encontrados menos acurados. Para superar essas desvantagens, a utilização de ferramentas de simulação computacional permite resolver cálculos complexos obtendo resultados tão precisos quanto necessário, apesar de computacionalmente custosa.

Embutidos nestas ferramentas de simulação computacional é comum a análise pelo Método dos Elementos Finitos (MEF) - ou em inglês: Finite Element Method (FEM) - que é um dos métodos numéricos aplicados em problemas de engenharia mais conhecidos. Trabalhos relacionados ao MEF começaram a surgir já no início dos anos sessenta, com foco principalmente no contexto da Mecânica Estrutural (BATHE, 2014). Contudo, rapidamente percebeu-se que este método poderia ter ampla aplicação em diversas outras classes de problemas. Atualmente, tal método é aplicado em temas que variam desde Mecânica Estrutural até Dinâmica dos Fluídos ou Transferência de Calor. Outra característica do método que o tornou bem sucedido desde o início é a produção de equações simultâneas geradas e resolvidas por meio de computadores digitais (FILHO, 2018). Desta maneira, por sua versatilidade e adequação computacional, é ampla a variedade de bibliotecas, softwares e empresas que oferecem produtos e serviços relacionados a ela.

É comum o processo de otimização de um produto, para a determinação de um conjunto apropriado de parâmetros deste, ser frequentemente realizado de forma manual, numa abordagem passo a passo lenta e que pode levar a resultados que não são ótimos globais (CASTORANI et al., 2017). Assim, é desejável a implementação de métodos que conduzam de forma automática este processo. Os chamados Algoritmos de Otimização, amplamente encontrados na literatura e disponíveis em uma ampla variedade de softwares (comerciais ou livres), geram e escolhem automaticamente as soluções potenciais a serem avaliadas direcionando o resultado para ótimos.

Capítulo 1. Introdução

Os métodos de otimização dividem-se em dois tipos principais: os analíticos e os estocásticos. O primeiro tipo, encontra soluções em menos iterações, o que exige menos avaliações, porém não são capazes de obter bons resultados em todos os casos. Isso se deve ao fato de exigirem cálculos dependentes de conhecimento aprofundado do problema em questão. Por sua vez, os Métodos Estocásticos, representados neste trabalho pelos Algoritmos Evolucionários, são capazes de lidar com problemas de otimização sem exigir os valores do gradiente das funções, suposições relacionadas à continuidade ou diferenciabilidade (LI et al., 2015) associados, com vários fatores a serem otimizados, amplo espaço de pesquisa e na presença de ruído. Contudo podem exigir um considerável número de iterações, consequentemente de avaliações sendo, portanto, computacionalmente custosos.

Para realização deste trabalho, dentre os Algoritmos Evolucionários, os Algoritmos Genéticos foram escolhidos levando em consideração as características citadas em (MULLER; KLASHORST, 2017): sua capacidade de encontrar soluções promissoras em problemas complexos de otimização, de possuírem ampla representatividade em bibliotecas de diferentes linguagens de programação, por serem facilmente implementáveis e versáteis o bastante para serem empregados em uma grande quantidade de problemas e áreas com pouco esforço.

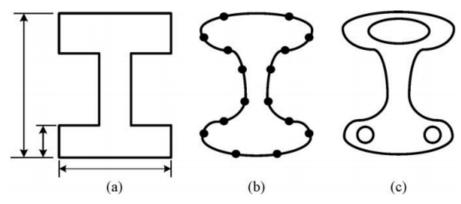
No contexto da mecânica estrutural, uma otimização pode ser empregada com o objetivo de reduzir os custos associados a fabricação de uma determinada peça, ao mesmo tempo que garante ou mesmo melhora sua performance quando aplicada a determinadas condições de contorno. Em outras palavras, procura-se minimizar o volume da peça ao mesmo tempo que minimiza, ou ao menos mantém, a tensão máxima e os deslocamentos da estrutura. Dessa forma garante-se, por exemplo, o ganho em economia de material da estrutura sem comprometer seu comportamento, sua durabilidade e sua segurança. À área da otimização que trata de problemas como este chama-se de Otimização Estrutural.

Otimização estrutural é uma técnica que procura encontrar as melhores estruturas mecânicas que suportem determinados carregamentos e restrições (GUIRGUIS et al., 2020). Neste processo pode ser de interesse: (1) fixada a forma, descobrir a(s) dimensão(ões) que conduz(em) aos melhores resultados; (2) fixada a topologia, descobrir a(s) forma(s) (e aí incluídas as dimensões) que conduzem aos melhores resultados; e (3) descobrir a(s) topologia(s) (com sua forma e dimensões) que conduz(em) aos melhores resultados (MORTAZAVI; TOğAN, 2016). Tais problemas podem ser formulados como problemas de otimização e a literatura possui diversos métodos extensamente discutidos (veja a figura 1).

Problemas de mundo real normalmente exigem a consideração de múltiplos critérios para avaliar soluções (LI et al., 2015). É comum tais critérios serem conflitantes, ou seja, a melhora em um deles piora o resultado em outro. Como já declarado, geralmente deseja-se otimizar um ou mais critérios relacionados ao custo da estrutura enquanto otimiza um ou mais critérios associados à performance dela. Para isso, a abordagem escolhida geralmente irá: considerar um único fator a ser otimizado mantendo os outros limitados a um valor, gerar um valor de objetivo que seja a soma ponderada dos restantes ou considerar dois ou mais critérios como objetivos da otimização. No primeiro caso, é intuitivo pensar que apenas um critério será efetivamente otimizado. Por sua vez, no segundo caso, nem sempre é possível ao deci-

Capítulo 1. Introdução 15

Figura 1 – Tipos de Otimização Estrutural



Otimização Estrutural: a) Paramétrica, b) de Forma e c) Topológica (LUH; LIN, 2009).

sor mensurar com certeza os valores corretos para os pesos (NG; SIEGMUND; DEB, 2018). Neste trabalho, serão escolhidos um critério relacionado ao custo da estrutura e um critério relacionado à performance da mesma, considerados separadamente. Para isso, deve-se utilizar algoritmos de otimização que consigam lidar com problemas de dois objetivos. Os algoritmos capazes de lidar com problemas desta natureza são chamados Algoritmos de Otimização Multiobjetivo. Portanto, uma vez definido que os algoritmos genéticos serão a classe de algoritmos de otimização escolhida, busca-se por meio deste trabalho utilizar métodos pertencentes à classe dos algoritmos genéticos de otimização multiobjetivo.

Para se propor uma otimização do tipo estrutural é necessário primeiro um método de otimização e em seguida um método de avaliação das soluções encontradas ao longo das iterações. Busca-se neste cenário que as soluções encontradas pelo método de otimização sejam submetidas à análise pelo software de simulação, para que, após terem sua performance medida, os valores obtidos re-alimentem a rotina de otimização. Contudo, não existe um meio facilitado de se comunicar qualquer algoritmo de otimização ao método de avaliação das soluções, fazendo com que o projetista se limite aos métodos presentes nos softwares comerciais ou desenvolva sua própria abordagem de comunicação. Realizar manualmente a chamada para a simulação de cada solução gerada aumenta consideravelmente o tempo de execução da simulação, limita o número de soluções a serem consideradas e impede os profissionais de se concentrarem em outras tarefas importantes. Por conta desses problemas, se faz necessário o desenvolvimento de uma comunicação automática entre os métodos de otimização e o software de avaliação via elementos finitos. Se por um lado o uso de softwares comerciais pode facilitar a aceitação em setores industriais, por outro lado os sistemas de otimização proprietários embutidos em softwares como o ANSYS® "carecem de sofisticação, permitindo que apenas algumas restrições simples sejam usadas na otimização, e funciona como uma caixa preta, não permitindo que o usuário modifique a otimização como achar melhor. O usuário ou pesquisador também fica à mercê do ANSYS® para quaisquer atualizações que tragam funcionalidades adicionais."(JANKOVICS et al., 2018).

Tendo em vista os problemas propostos e as soluções disponíveis, busca-se por meio deste trabalho promover uma comunicação entre um algoritmo genético de otimização multiob-

Capítulo 1. Introdução

jetivo e o software de simulação por elementos finitos a fim de responder a seguinte pergunta de pesquisa: A aplicação de um algoritmo evolucionário para otimização multiobjetivo acoplado à simulação por elementos finitos resulta em boas soluções para problemas de mecânica estrutural?

A estratégia de comunicação será realizada utilizando linguagem de programação Python e comunicará os algoritmos genéticos de otimização multiobjetivo ao software de Simulação Ansys Mechanical APDL (ANSYS®, Release 21.1c).

O objetivo geral é:

 Avaliar o desempenho de um algoritmo genético de otimização multiobjetivo na execução de uma otimização de forma utilizando simulação via elementos finitos em um problema aplicado.

Para alcançar este objetivo, pretende-se:

- Adaptar o código para que a biblioteca Pymoo comunique-se de forma eficiente com o ANSYS® de forma a realizar as simulações e fornecer de volta os valores objetivo de interesse;
- Escrever os códigos em Python que definem um problema de mão francesa de modo adequado para utilização pela biblioteca Pymoo;
- Comparar os resultados obtidos por meio da otimização proposta neste trabalho com os resultados obtidos por um dos algoritmos de otimização do ANSYS®.

# 2 Fundamentação teórica

"Por exemplo, poucos de nós seguiram o raciocínio no qual estão baseadas dez por cento das verdades nas quais acreditamos. Nós a seguimos com base na autoridade dos especialistas e agimos com sabedoria ao fazê-lo, pois embora sejamos às vezes enganados, ainda assim viveríamos como selvagens se não o fizéssemos".

C.S.Lewis

Nessa seção são definidos de forma breve os principais termos relacionados ao trabalho desenvolvido e apresentada uma descrição do algoritmo *Nondominated sorting genetic algorithm* (NSGA-II) escolhido para obter as soluções do problema de otimização.

### 2.1 Otimização

De acordo com (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012), o termo otimização é definido como "o campo de conhecimentos cujas técnicas visam determinar os extremos (máximos ou mínimos) de funções, em domínios determinados". Seguindo a abordagem deste mesmo autor, a formulação genérica de um problema de otimização pode ser efetuada da seguinte maneira:

$$\mathbf{x}^* = arg\,min \quad \mathbf{f}(\mathbf{x})$$

$$sujeito \ a: \begin{cases} g_1(\mathbf{x}) \le 0 \\ g_2(\mathbf{x}) \le 0 \\ \vdots \\ g_m(\mathbf{x}) \le 0 \end{cases}$$
 (2.1)

Em que  $\mathbf{x}^*$  é o vetor solução de minimiza um conjunto de funções  $f(\mathbf{x})$  (chamadas funções objetivo) sujeitas a um conjunto de restrições  $g(\mathbf{x})$ .

Um problema de otimização pode ser classificado quanto a seu número de objetivos em três categorias: Mono Objetivo, Multiobjetivo e Muitos Objetivos. A primeira categoria se refere a problemas de objetivo único, a segunda categoria é geralmente atribuída a problemas de dois ou três objetivos, enquanto a última categoria é comumente atribuída aos problemas com ao menos quatro objetivos (LI et al., 2015) <sup>1</sup>.

Para problemas com mais de um objetivo, define-se a componente f(x) como sendo um vetor de funções objetivo como se vê a seguir:

$$f(x) = (f_1(x), f_2(x), ..., f_m(x))$$
(2.2)

<sup>&</sup>lt;sup>1</sup> Esse autor classifica de maneira que a última categoria seja um caso especial da segunda.

Sendo m o número de objetivos do problema, compreende-se  $2 \le m \le 3$  para problemas multiobjetivo e  $m \ge 4$  para problemas com muitos objetivos (LI et al., 2015).

#### 2.1.1 Dominância e soluções eficientes

Apesar de muitos problemas serem tratados por meio de abordagens de objetivo único, grande parte deles lidam naturalmente com mais de um fator de avaliação. Assim, não há mais como definir se uma solução é "melhor" que outra por meio de uma comparação direta de um valor de objetivo único. Na maioria dos casos, os objetivos estão em conflito entre si, de modo que, exceto em casos muito particulares, não existe mais uma única solução capaz de otimizar todos os objetivos (ZHOU et al., 2011). Desta forma, o conceito de solução ótima é substituído pelo conceito de solução eficiente. Para isto, define-se (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012) a relação de dominância:

**Definição 1** (Dominância) Sejam  $x_a$  e  $x_b$  vetores factíveis do do problema de otimização multiobjetivo 2.1. O vetor factível  $x_a$  domina o vetor factível  $x_b$  se, e somente se,  $f_a(x) \neq f_b(x)$ ,  $f_a(x) \leq f_b(x)$  e exista índice i tal que  $f_{ai}(x) < f_{bi}(x)$ . Para indicar dominância utiliza-se o símbolo  $\leq$ , de modo que, se  $f_a$  domina  $f_b$ , escreve-se  $f_a \leq f_b$ .

E daí, o conceito de solução ótima é substituído pelo conceito de solução eficiente, cuja definição é (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012):

**Definição 2** (Solução eficiente) Um vetor de variáveis de decisão x\* factível é uma solução eficiente do problema de otimização multiobjetivo definido na equação 2.1 se não existir nenhuma outra solução factível desse problema que domine x\*.

A partir destes conceitos, definem-se o conjunto e a fronteira Pareto-Ótima (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012):

**Definição 3** O conjunto  $\mathcal{P}$  de todas as soluções eficientes de um problema de otimização multiobjetivo chama-se Conjunto Pareto-Ótimo (ou simplesmente Conjunto Pareto) e sua imagem  $\mathcal{F}$  no espaço objetivo recebe o nome de Fronteira Pareto-Ótima (ou simplesmente Fronteira Pareto).

Assim, "o alvo da otimização multiobjetivo é obter uma aproximação da Fronteira Pareto no espaço de objetivo de tal maneira que nenhuma melhoria posterior em qualquer objetivo é alcançada sem piorar o resto dos objetivos"(LI et al., 2015).

### 2.2 Algoritmos Evolucionários

De acordo com (KICINGER; ARCISZEWSKI; JONG, 2005), "algoritmos evolucionários são uma família de algoritmos de pesquisa baseados em população que simulam a evolução de estruturas individuais por meio de processos inter-relacionados de seleção, reprodução e

variação". Por possuírem forte semelhança e muitas características herdadas das áreas da genética, biologia celular e da teoria da evolução, os algoritmos evolucionários emprestam termos como população, geração, indivíduo, descendente ou filho, progenitores, evolução, gene, alelo, genoma, cromossomo e reprodução.

Dentre os métodos evolucionários mais conhecidos, encontram-se os algoritmos genéticos (AG), uma técnica que tem ganhado especial notoriedade nos últimos anos, devido sua ampla versatilidade e capacidade de gerar bons resultados em problemas de otimização, mesmo sem garantias de proximidade com o ótimo, o que a caracteriza como heurística. Tal técnica, possui etapas de execução bem definidas, o que exige, para sua implementação, escolher como representar uma solução, como avaliar as soluções, o tipo de reprodução e mutação, o método de seleção e o esquema de substituição (LIM, 2014). Essas escolhas deverão compor um fluxo semelhante ao presente na figura 2.

Próxima Geração

Não

Não

Critério de Terminação Satisfeito?

Sim

Figura 2 – Passos em um AG.

Fonte: Autor. Baseado em (JOSHI; GYANCHANDANI; WADHVANI, 2021).

Inicialmente, um algoritmo genético inicia sua execução por meio da geração de uma população inicial de indivíduos. Essa população inicial pode ser formada de maneira aleatória, por um conjunto de soluções já definidas previamente ou arbitrariamente escolhidas (KICINGER; ARCISZEWSKI; JONG, 2005). Essa população deverá passar por uma etapa de avaliação, na qual cada solução receberá um valor de *Fitness*, também chamado de aptidão. Basicamente, esta etapa irá calcular o valor da(s) função(ões) objetivo(s) para cada indivíduo. Logo em seguida, com base no valor de aptidão, os indivíduos passarão por uma etapa de seleção. Nesta etapa, uma técnica deverá ser aplicada para escolher aqueles que irão assumir o papel de progenitores. Diversas técnicas podem ser escolhidas: Seleção por Roleta ou "Roulette-wheel" - na qual a probabilidade de um indivíduo ser escolhido é baseada na relação entre a sua aptidão em comparação com a aptidão de toda a população, Seleção por torneio ou "tournament selection" - em que um indivíduo é escolhido dentre uma parte da população, Seleção por Ranque ou "rank selection" - em que a seleção dos indivíduos é feita com base no ranque de cada um deles, além de outros (LIM, 2014).

Após realizada a etapa de seleção dos progenitores, os indivíduos escolhidos devem realizar a etapa de reprodução, na qual, por meio deles, novos indivíduos serão gerados. Neste ponto, dois tipos de operadores de reprodução são possíveis: a recombinação e a mutação. Geralmente a recombinação é executada para gerar um ou mais indivíduos através da utiliza-

ção de genes dos cromossomos de dois indivíduos progenitores. Um tipo conhecido de recombinação é aquela realizada por meio de um ponto de corte, na qual um ponto do cromossomo é escolhido para "dividir" em partes o conjunto de genes, de modo que as partes separadas sejam trocadas para criação dos descendentes. Desta maneira, cria-se novos indivíduos sem introduzir variação genética na população (LIM, 2014). A figura 3 ilustra as recombinações por meio de um ponto de corte e por dois pontos de corte <sup>2</sup>.

Figura 3 – Recombinação por ponto de corte.

```
Example 1: Single-point crossover

Parent 1: 111|111 → Child 1: 111|001
Parent 2: 011|001 → Child 2: 011|111

Example 2: Two-point crossover

Parent 1: 11|11|11 → Child 1: 11|10|11
Parent 2: 01|10|01 → Child 2: 01|11|01

Fonte: (LIM, 2014)
```

Enquanto o operador de recombinação não introduz variação genética, o operador de mutação o faz alterando uma pequena parte do cromossomo do indivíduo por meio de uma mudança aleatória. Segundo (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012), o objetivo da mutação é "manter um nível de diversidade adequado na população, garantindo que alelos que eventualmente desapareçam tenham a possibilidade de reaparecer". O tipo de mutação mais simples encontrado é o operador de mutação binário, que pode ser observado pela figura 3. Seu funcionamento básico é alterar o valor do bit escolhido <sup>3</sup>.

Figura 4 – Operador de mutação binária.

```
Example 3: 1-bit mutation operator

Chromosome: 111111 → 111101

Fonte: (LIM, 2014)
```

Uma vez finalizada a etapa de reprodução no esquema dos algoritmos genéticos, uma nova população deve ser escolhida com base na anterior e na população de descendentes recém gerada. Neste ponto, o método pode substituir toda a população anterior por novos indivíduos ou pode substituir apenas parte dela. Aos algoritmos que se encaixam no primeiro caso, chama-se "Algoritmo Geracional", enquanto àqueles que aplicam o segundo caso, denominase "steady-state" (KICINGER; ARCISZEWSKI; JONG, 2005). Em seguida, um teste de terminação deve ser executado. Caso esse teste seja satisfeito, retorna-se a(s) solução (ões) obtidas e finaliza, ou recomeça o ciclo a partir da etapa de seleção. O critério de terminação pode

<sup>&</sup>lt;sup>2</sup> A literatura apresenta vários outros métodos de recombinação. Nesta revisão, apresentou-se um método simples de recombinação de indivíduos compostos de variáveis binárias apenas a título de exemplo.

O comentário anterior é igualmente válido para a mutação.

ser encontrar um valor ótimo, ter realizado um número definido de gerações, ter avaliado um número definido de indivíduos entre outros.

O processo descrito é ilustrado pelo algoritmo 1, que esquematiza a estrutura básica de um algoritmo genético genérico.

```
Algoritmo 1: Estrutura genérica de um AG. Fonte: (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012)
```

```
1 t \leftarrow 0;
2 Gerar população inicial P(t);
3 Avaliar os indivíduos de P(t);
4 Avaliar os indivíduos de P(t);
5 while (critério de terminação não atingido) do
6 Selecionar os progenitores P'(t) a partir de P(t);
7 Aplicar Operadores genéticos a P'(t) obtendo a nova população P(t + 1);
8 Avaliar P(t + 1);
9 t \leftarrow t + 1;
end
10 Output: Resultado final da otimização
```

### 2.2.1 Algoritmo Genético Multiobjetivo baseado em não dominância: NSGA-II

Uma estratégia conhecida de otimização multiobjetivo é a segunda versão do algoritmo Nondominated sorting genetic algorithm, conhecida como NSGA-II. Essa versão surge da necessidade de superar três problemas encontrados na versão inicial do método: Alta complexidade na ordenação das soluções, falta de elitismo e necessidade de especificar um parâmetro para diversidade  $\sigma_{share}$  (DEB et al., 2002). A superação dos desafios encontrados ocorre por meio de três principais mudanças: criação de um novo método de ordenação de soluções - o Fast Nondominated Sorting, a criação de um método para atribuir um valor de distância de aglomeração (Crowding Distance Assignment) para cada solução e um novo comparador de soluções (Crowding Distance Operator). Cada uma dessas mudanças está descrita nas subseções abaixo.

#### 2.2.2 Ordenação rápida com base em não dominância

Para estabelecer um critério mais eficiente de ordenação das soluções e induzir um critério melhor que garanta elitismo, a versão dois do NSGA implementa um método cujo objetivo é ordenar as soluções com base no critério de dominância definido anteriormente. Esse ordenador separa as soluções em grupos chamados de "fronts" ou fronteiras, dando proeminência as soluções que não são dominadas por nenhuma outra: na primeira fronteira permanecem aquelas que não são dominadas por nenhuma outra solução, na segunda permanecem aquelas dominadas por apenas uma outra solução e assim por diante. O esquema de segregação das soluções é representado na imagem 6. Mais detalhes sobre o processo e implementação podem ser encontrados em (DEB et al., 2002).

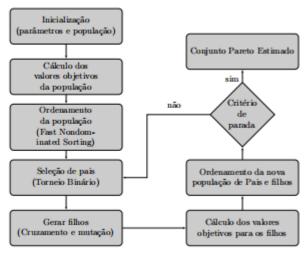
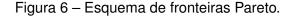
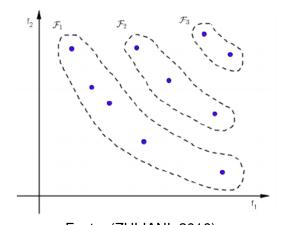


Figura 5 - Fluxograma do NSGA-II.

Fonte: (ZULIANI, 2016)





Fonte: (ZULIANI, 2016)

#### 2.2.3 Atribuidor de distância de aglomeração

Para estabelecer um critério de comparação de diversidade entre as soluções, um algoritmo de atribuição de distância de aglomeração foi desenvolvido. Esse algoritmo ordena cada valor de objetivo e atribui uma distância infinita para os indivíduos com valores extremos. Para os outros indivíduos que possuem valores intermediários, os valores de objetivo imediatamente maior e menor ao seu serão utilizados para calcular a distância por meio do cálculo do cuboide, como a imagem 7 e a equação 2.3:

$$DA = DA + \frac{obj[i+1] + obj[i-1]}{obj_{maior} - obj_{menor}}$$
(2.3)

Sendo,

• DA: distância de aglomeração.

- obj[i+1] e obj[i-1]: respectivamente o valor imediatamente maior e menor no vetor de indivíduos ordenados levando em consideração um dado valor da função objetivo.
- obj<sub>maior</sub> e obj<sub>menor</sub>: respectivamente o maior e o menor valor do vetor de indivíduos ordenados levando em consideração um dado valor da função objetivo.

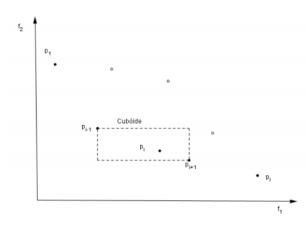


Figura 7 – Calculo do Cuboid.

Fonte: (DEB et al., 2002).

O valor calculado é atribuído para cada uma das soluções e será usado como medida de seleção para progenitores. A interpretação dos valores de aglomeração diz que quanto maior for a distância final de aglomeração, menos populada é aquela região e, portanto, a diversidade desta solução é superior e é desejado investigar a região em sua volta. Por outro lado, quanto menor o valor, mais comum é a solução e seu entorno já foi investigado. Logo, como uma adequada exploração do espaço de busca é um fator fundamental para uma correta convergência de um AG, busca-se por soluções com distância de aglomeração superior. Para mais informações a respeito do método de atribuição de distância de aglomeração, consultar (DEB et al., 2002).

#### 2.2.4 Operador de Distância de Aglomeração

Como explicado, um AG deve promover elitismo e diversidade em suas populações. Para isso, após separar as soluções por meio de fronteiras e atribuir a cada um valor de distância de aglomeração, é necessário definir um critério de comparação que leve em consideração essas duas características. O operador escolhido para atribuir este critério é o operador de distância de aglomeração. Esse operador seleciona a solução que está em uma fronteira inferior e, quando ambas estão na mesma fronteira, escolhe a com maior distância de aglomeração. Para mais detalhes consultar (DEB et al., 2002).

#### 2.2.5 Rotina principal do Algoritmo NSGA-II

Após gerar de maneira aleatória os valores da população inicial, selecionar os progenitores  $P_t$  e gerar uma população de descendentes  $Q_t$ , uma população  $R_t$  é gerada unindo  $P_t$  e  $Q_t$ . Essa nova população passará pelo método de separação de fronteiras que atribuirá um

valor de classificação com base nos seus valores de objetivo. Após esse processo, uma nova população será montada unindo indivíduos de fronteiras em ordem crescente, até que seu tamanho atinja o número esperado de indivíduos da próxima população  $P_{t+1}$ . Para cada um dos indivíduos do conjunto unido, o valor da distância de aglomeração é atribuído. Após isso, os indivíduos serão novamente ordenados por meio do comparador de distância de aglomeração explicado acima. Após ordenados, os valores da posição N em diante, que superam o tamanho da esperado da população, serão descartados. Essa população deverá passar pelo processo de geração de novos indivíduos e voltar ao passo de união com os progenitores em  $R_t$ . Esses passos serão repetidos até que o critério de terminação seja satisfeito. A rotina principal de repetição do NSGA-II está representada pelo algoritmo 2.

**Algoritmo 2:** Procedimento, a cada geração, de formação da população do NSGA-II. Fonte: (DEB et al., 2002).

```
\begin{array}{l} \mathbf{1} \ R_t = P_t \cup Q_t; \\ \mathbf{2} \ \mathsf{F} = \mathsf{fast\_non\_dominated\_sort}(R_t); \\ \mathbf{3} \ P_{t+1} = \emptyset \ \mathsf{e} \ i = 1; \\ \mathbf{4} \ \ \mathsf{while} \ | P_{t+1}| + |F_i| \leq N \ \mathsf{do} \\ \mathbf{5} \ | \ \mathsf{crowding\_distance\_assigment}(F_i); \\ \mathbf{6} \ | \ P_{t+1} = P_{t+1} \cup F_i \ ; \\ \mathbf{7} \ | \ \mathsf{i} = \mathsf{i} + 1; \\ \ \ \mathsf{end} \\ \mathbf{8} \ \mathsf{Sort}(F_i, \succ_n); \\ \mathbf{9} \ P_{t+1} = P_{t+1} \cup F_i[1:(N-|P_{t+1}|)]; \\ \mathbf{10} \ Q_{t+1} = make\_new\_pop(P_{t+1}); \\ \mathbf{11} \ \ \mathsf{t} = \mathsf{t} + \mathsf{1}; \\ \end{array}
```

#### 2.3 Método dos Elementos Finitos

Apoiando-se no método geral para análise de sistemas discretos (observar figura 8), o método dos elementos finitos (MEF) é um método numérico capaz de realizar análises de engenharia em estruturas complexas, cuja a abordagem por técnicas clássicas seria muito difícil ou mesmo inexistentes (FILHO, 2018). Seu funcionamento se baseia em dividir o modelo em partes de tamanho finito, ou seja, em elementos finitos. Por meio desta discretização, é possível analisar o todo de uma estrutura complexa a partir da compreensão de suas várias subdivisões.

O MEF divide o modelo contínuo em um número finito de partes de tamanho não infinitesimal. Cada uma dessas partes, chamadas de elementos são conectadas umas as outras por meio de pontos discretos, chamados de nós (FILHO, 2018). Os cálculos necessários à analise são efetuados inicialmente sobre esses nós e, em seguida, por meio do conhecimento do comportamento desses pontos, passados aos elementos. Dessa maneira, ao invés de ser exigida uma equação que consiga descrever todos os infinitos pontos da estrutura, os cálculos são realizados inicialmente para os pontos discretos. Por conta disso, é necessário definir cuidadosamente, o quão acurado deve ser o resultado, a fim de que a quantidade de nós seja suficiente para representar o fenômeno. Além disso, uma vez que os elementos representarão

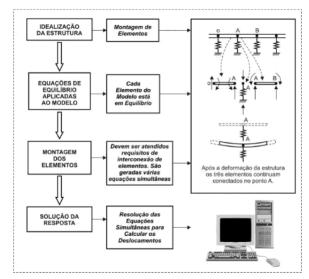


Figura 8 – Método Geral para análise de Sistemas discretos - Sistema Discreto Padrão.

Fonte: (FILHO, 2018)

partes da estrutura e terão como parâmetros iniciais os valores calculados para seus nós, o tipo de elemento utilizado e a definição do mesmo também é importante, pois quanto mais bem definido e adequado o elemento para o contexto em análise, mais próximo será o resultado do sistema real.

Ao realizar uma análise via elementos finitos, geralmente é possível se deparar com os seguintes tipos de estruturas: Vigas ou estruturas reticuladas, análise de cascas, chapas e analise de estruturas tridimensionais. O primeiro tipo, geralmente possui uma subdivisão mais intuitiva, sendo possível, por exemplo, dividir visualmente a estrutura em vários componentes por meio de suas ligações (ou soldas) naturais. Este tipo de estrutura geralmente é representada por um modelo inicial formado por pontos (soldas e ligações) e retas (barras ou vigas). Desta maneira, ainda que na discretização seja executada uma divisão mais refinada da estrutura - dividindo cada reta em várias partes finitas, os elementos permanecerão conectados por apenas dois pontos de contato. Para ilustrar esse conceito, observe a figura 9. Nesta figura é possível observar a estrutura por meio de várias arestas conectadas por meio de pontos numerados de 1 a 6. Neste tipo de análise, quando esta estrutura for discretizada, a representação das partes da estrutura será feita por meio elementos conectados apenas por meio de 2 nós.

Por outro lado, as estruturas do tipo casca ou tridimensionais, são estruturas contínuas, em que seus elementos estão conectados por infinitos pontos na estrutura original, mas que serão conectados, após a discretização, apenas por meio dos nós dos elementos finitos. Portanto, para estabelecer uma análise adequada das mesmas é necessário uma teoria (e portanto equações) mais robusta, que garanta a compatibilidade dos elementos conectados (FILHO, 2018). Para ilustrar essa situação, observe a figura 10, veja que, ainda que a teoria do MEF ordene a conexão dos elementos por meio de nós, os vários elementos estariam conectados por infinitos pontos na estrutura original.

9.144 m 9.144 m 9.144 m 9.144 m 9.144 m 445 kN 445 kN

Figura 9 – Estrutura reticulada - Esquema de uma treliça de 10 barras.

Fonte: (MULLER; KLASHORST, 2017)

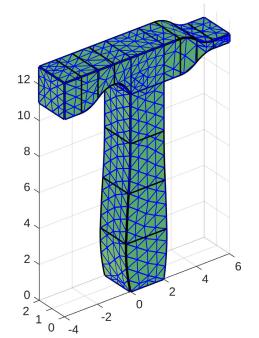


Figura 10 – Estrutura Tridimensional - Malha de elementos finitos de um martelo.

Fonte: (LóPEZ; ANITESCU; RABCZUK, 2020)

Por ser um método numérico capaz de fazer boas aproximações de fenômenos por meio de discretização, é de suma importância compreender alguns aspectos deste tipo de análise. Tais aspectos são pontuados por (BATHE, 2014): Para uma boa análise é fundamental um bom modelo. É de suma importância que o projetista consiga extrair do mundo real os componentes necessários para que a análise seja efetuada de maneira coerente. Paralelo a esse aspecto inicial, está o fato de que nenhum modelo simulado será capaz de extrair com precisão todos os detalhes de um fenômeno, pois, independentemente do quão bem feita é, toda análise será por fim reducionista. Desta forma, é importante que o projetista seja capaz de filtrar do fenômeno em análise fatores que não possuam significativo impacto naquilo que deseja simular. Tendo em vista que se trata de uma aproximação, é fundamental que os resultados sejam julgados a luz da experiencia, do conhecimento teórico e até mesmo a luz de um modelo mais abrangente, para que incoerências na análise sejam corrigidas.

### Representação geométrica por B-Splines racionais não uniformes (NURBS)

Uma curva NURBS (B-Splines racionais não uniformes) é um ferramenta matemática capaz de representar formas livres com grande capacidade de movimentação localizada. Uma curva deste tipo é definida pela seguinte fórmula (PIEGL; TILLER, 1995):

$$C(t) = \frac{\sum_{i=0}^{n} N_{i,p}(u)w_i P_i}{\sum_{i=0}^{n} N_{i,p}(t)w_i}$$
 (2.4)

Sendo que,

- w<sub>i</sub>: Pesos
- P<sub>i</sub>: Pontos de controle
- $N_{i,p}(u)$ : Função base B-Spline do i-ésimo grau definida em um vetor de nós (knot vector) não periódico (e não uniforme) :

$$U = \{a, ..., a, U_{p+1}, ..., U_{m-p-1}, b, ...b\}$$

Os pontos de controle definem um polígono de controle, cujo corpo é geralmente seguido pela curva e cuja envoltória convexa geralmente contém a curva, enquanto os pesos determinam o quão próximo do ponto de controle a curva passará (ROGERS, 2001) apud (KROHE, 2020). A figura 11 ilustra o fato de que esta curva permanece dento da envoltória convexa do polígono criado pelos pontos de controle. Além disso, é visível que quanto maior é o valor do peso variável, mais próximo do ponto de controle relacionado a ele a curva fica.

Uma característica fundamental que torna esse tipo de abordagem geométrica notável é a capacidade de movimentações localizadas. Observe na figura 12 que apesar da movimentação dos pesos alterar o comportamento da curva, é possível notar que a parte final desta curva não sofre alteração. Isso ocorre devido a uma propriedade denominada Aproximação

 $P_1$   $P_2$   $P_5$   $P_6$   $P_0$   $P_0$ 

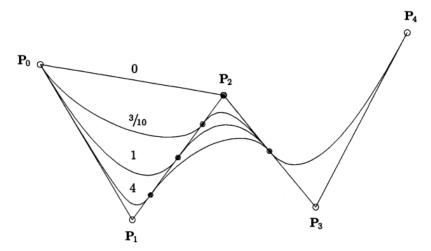
Figura 11 – B-splines racionais cúbicas com peso  $w_3$  variando.

Fonte: (PIEGL; TILLER, 1995).

 $P_4$ 

local, que diz: "se o ponto de controle  $P_i$  é movido, ou o peso  $w_i$  é alterado, isso afeta apenas aquela porção da curva no intervalo  $[u_i,u_{i+k1})$ "(PIEGL; TILLER, 1995).

Figura 12 – Curvas racionais quadrática com peso  $w_1$  variando.



Fonte: (PIEGL; TILLER, 1995).

Assim, as NURBS representam o estado da arte em representação paramétrica de curvas. Também podem ser definidas superfícies utilizando-se dos mesmos princípios aplicados às curvas. Neste trabalho, as curvas serão utilizadas para definir a forma da estrutura e por isto restringe-se aqui à uma breve descrição.

## 3 Trabalhos correlatos

"Confie no Senhor de todo o seu coração e não se apóie em seu próprio entendimento; reconheça o Senhor em todos os seus caminhos, e ele endireitará as suas veredas.".

Provérbios 3:5,6

(CASTORANI et al., 2017) utiliza um algoritmo genético de otimização multiobjetivo para propor uma otimização que leve em consideração tanto as propriedades mecânicas, quanto também o custo de produção conectando ferramentas de "Projeto Assistido por Computador"(CAD - Computer-aided Design), "Engenharia apoiada por computador"(CAE - Computer-aided Engineering) e "Custo de Design"(DFC - Design for Cost). Para reduzir o custo computacional do método de simulação por elementos finitos, os autores utilizam técnicas para gerar uma superfície de resposta que aproxima o modelo analisado. Essa superfície de resposta é gerada por meio da variação propositada dos parâmetros, visando compreender suas influencias nos resultados final. Em seguida o Algoritmo Genético de Otimização Multiobjetivo MOGA é utilizado para resolver o problema de otimização.

(SU et al., 2018) utiliza o método de otimização multiobjetivo enxame de partículas (MOPSO), para propor a otimização multiobjetivo de *GFRP elastic gridshells* (que é um tipo de estrutura treliçada). Para esse processo, o autor utiliza análise via elementos finitos para gerar um arquivo contendo avaliações de 360 estruturas. Em seguida, utiliza esse arquivo para treinar um algoritmo de aprendizado de máquina, que será utilizado durante a otimização, a fim de reduzir o custo computacional inerente das otimizações utilizando Algoritmos evolucionários e Simulação via Elementos finitos.

(LIU et al., 2016) utiliza o método de enxame de partículas para otimização de nanopartículas tri-metálicas Pt-Pd-Rh. O autor buscará avaliar a estabilidade do átomo, minimizando a energia potencial, tendo como variável a posição dos átomos na estrutura.

(HSIAO et al., 2020) utiliza o Algoritmo de otimização MOEA/D-DE para otimização de design do braço de um robô, tendo como variáveis cinco parâmetros da estrutura e como objetivos peso, momento de inércia e deslocamento. Neste trabalho, um modelo de regressão linear foi utilizado para substituir as simulações por elementos finitos durante a otimização, a fim de reduzir custo computacional.

## 4 Método de Pesquisa

"O profundidade das riquezas, tanto da sabedoria, como da ciência de Deus! Quão insondáveis são os seus juízos, e quão inescrutáveis os seus caminhos!".

Romanos 11:33

### 4.1 Classificação da Pesquisa

Sob a ótica das classificações de (WAZLAWICK, 2014), o trabalho aqui apresentado é quanto à natureza original, quanto ao objetivo exploratório e quanto aos procedimentos técnicos experimental. Diz-se original pois procura encontrar e gerar informações novas relacionadas ao tema. Define-se exploratório, pois objetiva examinar e estudar fenômenos inerentes aos problemas de otimização estrutural apoiada em simulação por elementos finitos na área de mecânica estrutural. Classifica-se experimental, pois este trabalho visa introduzir e testar heurísticas bioinspiradas no contexto citado, propor experimentos e julgar os resultados obtidos por meio técnicas de comparação consolidadas.

#### 4.2 Recursos Utilizados

Para realização das simulações, foi escolhido o software Ansys Mechanical Apdl versão "Student 2021 R1". Para o desenvolvimento do programa, as bibliotecas disponíveis na linguagem python utilizadas foram:

- pymoo (Blank; Deb, 2020), para reuso de algoritmos de otimização.
- NURBS-Python (BINGOL; KRISHNAMURTHY, 2019), para geração de curvas B-Spline Racionais não Uniformes (NURBS).

Para realização dos experimentos, contou-se com um equipamento com as seguintes especificações de *hardware*:

Tabela 1 – Especificações do Processador

Processador	Intel Core i5-6200U
Frequência	2.30 GHz - 2.40 GHz
Memória RAM	8 Gigas

Fonte: o Autor.

### 4.3 Procedimentos de pesquisa

Para executar este trabalho, escolheu-se o problema de otimização da mão francesa, que é um exemplo de problema de mecânica estrutural em condições estáticas, e foi realizada

a codificação da geometria do problema deste utilizando linguagem APDL. Em seguida, a forma geométrica escolhida foi preparada para otimização: escolheu-se parte de seu contorno e utilizou-se NURBS para geração de formas livres. Após ter executado a parametrização, implementou-se um programa que executava comunicação entre os algoritmos genéticos de otimização contidos na biblioteca *Pymoo* e o software da Ansys. Além disso, fez-se a descrição dos experimentos a serem aplicados e foram executadas as otimizações.

Definir a Implementação Implementar a Escolher o do código APDL arametrização abordagem de problema para correspondente da peça comunicação correspondente otimização. ao Problema para o modelo. escolhido. ao Problema. Definir os experimentos. Analisar os Executar os resultados. experimentos Fonte: o Autor.

Figura 13 - Comunicação Python APDL.

Para o problema acima, duas classes foram implementadas em *python*: uma para encapsular os atributos do problema e outra para realizar a tradução e mediar a comunicação entre o algoritmo de otimização e o software de simulação.

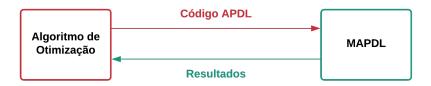
#### 4.3.1 Comunicação Python APDL

A comunicação proposta para a realização deste trabalho busca permitir um processo de retroalimentação entre o algoritmo de otimização e o software de simulação (figura 14).

Por meio desta comunicação, as potenciais soluções geradas pelos algoritmos de otimização em *python* eram traduzidas em código APDL e enviadas ao software de simulação computacional *Mechanical APDL (MAPDL)*. O código traduzido era executado pelo MAPDL, que executava os cálculos e devolvia os valores objetivo à rotina, realimentando a otimização. Esta abordagem é semelhante à abordagem de comunicação proposta em (DITTMER et al., 2006).

A cada nova população, todos os indivíduos eram traduzidos em código APDL e escritos em um arquivo do tipo *.mac*. Este arquivo continha o código da simulação para todos os indivíduos daquela população, separados por um comando \CLEAR, que é um comando APDL

Figura 14 - Comunicação Python APDL.



Fonte: o Autor. Inspirado em (DITTMER et al., 2006).

responsável por remover os dados da simulação anterior. Cada um destes códigos APDL ordenava a criação de arquivos que conteriam os valores objetivo selecionados e eram enumerados de modo a separar as simulações. Após gerados os arquivos, a rotina de otimização era executada para extrair os dados desejados e envia-los ao método de otimização. Desta forma, o processo genérico mostrado pela figura 14, quando exposto, se torna como o ilustrado em 15.

Pontos de Controle Vetores de Solução e Pesos Ag Pymoo Problema NURBS-Python Valores de Curva NURBS Objetivo Coordenadas da Curva Valores Resultados Tradução e Comunicação Chamada de terminal Rotina de extração Geração de MAPDL resultados Resultados

Figura 15 – Modelo final de comunicação Python APDL.

Fonte: o Autor.

O modelo de otimização é encapsulado por meio de um componente Problema. Este componente será definido por meio de uma classe que conterá os atributos do modelo, assim como os métodos responsáveis por gerar os limites de criação dos valores de solução e or-

denar a tradução. Esse componente implementa a interface *Problem* de (Blank; Deb, 2020) e define o método de avaliação das soluções (\_evaluate). Este método recebe um vetor de vetores contendo os pontos de controle e pesos para geração das curvas de todos os indivíduos de uma população. Os pontos de controle são separados dos pesos e ambos são enviados pelo componente Problema para o componente NURBS que se comunica com a biblioteca (BINGOL; KRISHNAMURTHY, 2019). Esta biblioteca gerará a curva e retornará à classe Problema os pontos gerados. Após todos os indivíduos terem sido traduzidos, o vetor de curvas NURBS geradas é enviado ao componente de tradução, que irá traduzir o código, gerar o arquivo .mac e chamar o software de simulação por meio de uma chamada utilizando código de terminal. Os valores gerados na simulação serão colocados em arquivos, extraídos pelo componente de Tradução e retornados ao componente Problema, que irá enviá-los ao algoritmo de otimização.

#### 4.3.1.1 Da modelagem das peças

O modelo escolhido é proveniente de uma peça comumente presente no contexto da mecânica: Mão francesa. Para torná-la suscetível à otimização, modelou-se uma parte de seu contorno por meio de uma curva NURBS. Essa curva é gerada por meio de pontos de controle e pesos variáveis pelo AG, dentro de limites estabelecidos no componente Problema. Desta maneira, apenas uma região limitada e específica do contorno da peça era parametrizado. Para mensurar a performance destas peças no contexto escolhido, definiu-se que a simulação deveria retornar 2 valores: um relacionado ao custo da peça, o volume, e a tensão máxima sofrida pela peça quando submetida a determinados carregamentos dentro de determinadas restrições.

#### 4.4 O Modelo

O modelo escolhido foi o de uma mão francesa. A geometria dessa peça pode ser visualizada por meio da figura 16. Esse geometria foi construída tendo 200.0 milímetros de largura, 300.0 milímetros de altura e 20 milímetros de espessura. Cada furo possui 10 milímetros de raio com centro em um par de coordenadas (X,Y). Essas coordenadas podem ser conferidas por meio da tabela 2.

Com o software de simulação obtém-se o estado de tensões internas desenvolvidas no interior da peça devido à solicitação (esforço/força) feita em um dos furos e a condição de travamento dos outros dois furos. Dessas condições de contorno temos o que chamamos na mecânica dos sólidos de condição estática. Tal condição é comum nas estruturas de máquinas e equipamentos civis e industriais. Existe um limite no qual o material que é fabricado a estrutura (i.e. a mão francesa de aço) resiste ao aumento da tensão interna. Um dos critérios mais utilizados para se definir esse limite de resistência é pelo cálculo da tensão resultante de von Mises. E a tensão de von Mises é calculada pelo ANSYS para cada elemento. Tal tensão que é usada nas considerações do problema de otimização.

Furo	Coordenada X	Coordenada Y
Superior esquerdo	20.0 mm	$-20.0 \; {\rm mm}$
Superior direito	162.6 mm	$-20.0 \; {\sf mm}$

Tabela 2 – Posição dos Furos na peça

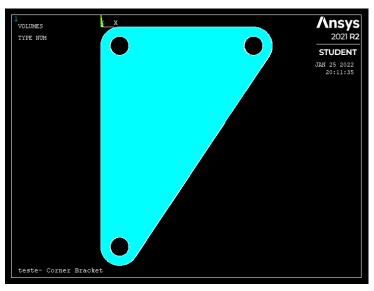
Fonte: o Autor.

 $20.0~\mathrm{mm}$ 

-233.9 mm

Inferior

Figura 16 – Peça de referencia da Mão francesa.



Modelo da peça referência com material em toda a região de projeto.

#### 4.4.1 A parametrização

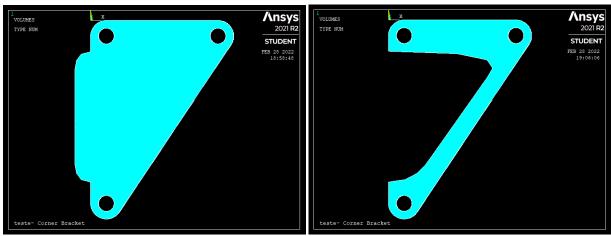
Para promover a parametrização da peça, modelou-se um trecho da lateral esquerda da peça utilizando uma curva NURBS. A princípio, os pesos e os pontos de controle da curva foram considerados como variáveis para a otimização.

Algumas restrições quanto ao domínio dos valores das variáveis foram impostas à curva que define o perfil da lateral esquerda da peça para que fossem geradas peças factíveis como solução. Assim, definiu-se que o ponto inicial do trecho variável dessa curva seria o ponto (0,-40) e seu ponto final estaria em (0,-223.3333). As ordenadas dos pontos de controle da curva foram deixadas fixadas, de maneira que cada ponto tinha sua altura definida, e a abscissa do i-ésimo ponto de controle poderia variar no intervalo  $[-20,K_i]$ , em que o valor de  $K_i$  foi definido por:

$$K_i = (i * distance) - security\_v * (1 + (i/10))$$
 (4.1)

em que distance é o valor obtido dividindo a largura da peça pela quantidade de pontos de controle escolhida e  $security\_v$  um valor adequado para que a lateral esquerda da peça não chegue à lateral direita (em diagonal) gerando peças factíveis. Para ilustrar aproximadamente a região em que a curva estaria limitada, as figuras 17a e 17b mostram respectivamente as

Figura 17 – Representação das peças nos limites minimo e máximo para os pontos de controle.

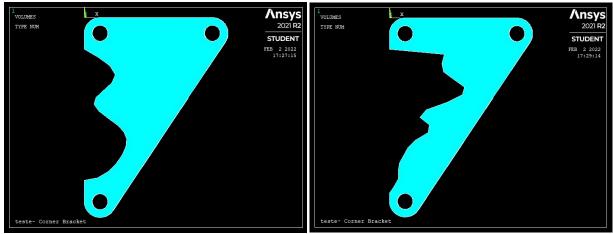


(a) Peça com pontos de controle igual ao limite in- (b) Peça com pontos de controle igual ao limites obferior -20 e pesos iguais a 1. tidos pela equação 4.1 e pesos iguais a 1.

Fonte: o autor.

peças geradas com os pontos de controle nos valores mínimo e máximo. Com relação aos valores dos pesos, definiu-se que poderiam variar entre 0 e 15. Por meio dessa abordagem foi possível gerar peças como nas figuras 18a e 18b.

Figura 18 – Exemplos de peças geradas durante o processo de otimização.



(a) Peça gerada utilizando-se 6 pontos de controle (b) Peça gerada utilizando-se 12 pontos de controle em trecho da lateral esquerda.

Fonte: o autor.

#### 4.4.2 A Malha e as condições de contorno

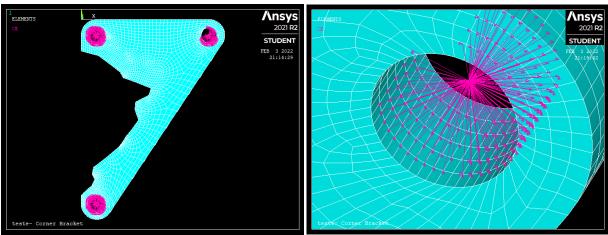
Como exposto anteriormente, uma etapa importante do método de Elementos Finitos é a geração da Malha de Elemento Finitos. Para gerar essa malha, primeiramente utilizou-se a opção livre. Por meio dessa abordagem o próprio software promoveria a discretização da área principal da peça. Em seguida, fez-se um refino desta malha inicial. Para esta etapa de discre-

tização bidimensional, utilizou-se o elemento com características bidimensionais PLANE182. Logo em seguida, a malha refinada era transferida para a peça inteira por meio do comando apdl VSWEEP utilizando o elemento SOLID185. Por fim, a fim de evitar problemas durante os cálculos, a malha 2D utilizada como base era excluída.

As condições de contorno utilizadas foram Suportes Cilíndricos para os furos da lateral esquerda da peça e um *Bearing Load* para o furo a direita da peça. Um suporte cilíndrico é um tipo de restrição que impede que as áreas cilíndricas nas quais é aplicado sofram deformação radial, axial ou tangencial. O *Bearing Load*, por sua vez, é um tipo de carregamento definido em regiões cilíndricas que atua sobre uma determinada parte da superfície, simulando forças apenas radiais. Em ambos os casos, uma vez que durante a geração do modelo não foram encontrados esses tipos de condições de contorno suportados pelo próprio software de elementos finitos, procurou-se via código um meio de simular ambos.

Para simular esses carregamentos, um ponto era definido no centro de cada furo, um elemento de nó único era atribuído para cada um desses pontos, o carregamento ou a restrição era colocado sobre esse ponto e então transferido para todos os nós das 4 áreas que formam cada um dos furos para o Suporte Cilíndrico ou para as duas áreas onde a força radial do Bearing load seria aplicada. Neste modelo, serão aplicadas 2 componentes de força através do Bearing Load formando um carregamento de (50, -100) pascal. Uma peça contendo esses carregamentos aplicados pode ser visualizada a por meio da figura 19.

Figura 19 – Figura representando as condições de contorno aplicadas sobre as áreas dos furos do suporte de esquina.



(a) Modelo de peça inteira, cor rosa representando (b) Visão do Bearing Load localizada, mostrando as as condições de contorno. três áreas onde ocorre a distribuição.

Fonte: o autor.

### 4.5 A otimização

Uma vez apresentado a peça de base, sua modelagem geométrica e parametrização, define-se um problema de otimização bi objetivo, considerando como fatores a serem otimiza-

dos: Tensão e Stress. Este problema de otimização pode ser matematicamente descrito por meio da equação 4.2.

$$\begin{cases} \text{ Minimize } & \mathbf{F}(\mathbf{x}) = \left(\mathbf{T}(\mathbf{x}), \mathbf{V}(\mathbf{x})\right) \in \mathbb{R}^2 \\ \text{ sujeito a.: } & \mathbf{x} \in \Omega \end{cases}$$
 (4.2)

Em que x é o vetor solução, V(x) é o volume da peça, T(x) é o valor da Tensão máxima de Von Misses sofrida pela peça e  $\Omega$  é o conjunto de todos os valores possíveis para as variáveis que definem a peça.

Um vetor de solução x é chamado no contexto da otimização evolucionária de indivíduo e possui sua estrutura geral formada de acordo com:

$$[P_1, P_2, ..., P_n, W_1, W_2, ..., W_n]$$

Sendo  $P_i$  o i-ésimo ponto de controle,  $W_i$  o i-ésimo peso da curva NURBS e n o número de pontos de controle, de modo que seja igual o número de pesos. Os pontos de controle possuíam como limitação inferior -20 e superior de acordo com a equação 4.1. Por sua vez, os pesos eram limitados inferiormente pelo valor 0 e superiormente pelo valor 15.

#### 4.6 Dos Experimentos

Além de avaliar as soluções apresentadas pelo método aqui proposto, pretende-se buscar também indícios sobre a influência da quantidade de pontos de controle e dos pesos nos resultados da otimização. Assim, realizou-se um experimento no qual a quantidade de pontos de controle podiam assumir duas possibilidades (6 ou 12) e os pesos também foram considerados de duas maneiras (pesos variáveis em um intervalo ou fixos iguais a 1). A escolha da quantidade de pontos foi realizada pela experiência do autor com a simulação do problema e as dimensões da peça adotada.

Desta maneira, executou-se 4 variações intercaladas dos experimentos conforme a tabela abaixo demonstra:

Furo Nº Pontos de Controle Comportamento dos Pesos Experimento 01 6 Fixos iguais a 1

Experimento 02 6 Variáveis [0, 15]

Experimento 03 12 Fixos iguais a 1

Experimento 04 12 Variáveis em [0, 15]

Tabela 3 – Experimentos realizados

Fonte: o Autor.

Cada otimização foi executada utilizando-se o algoritmo NSGA-II com as seguintes características:

Algoritmo de seleção: Torneio binário.

Algoritmos para geração de novos indivíduos:

Recombinação: Crossover Binário Simulado

- Mutação: Mutação polinomial

Critério de terminação: Máximo de 70 Gerações

Tamanho da população: 16 indivíduos

Utilizou-se uma curva NURBS de grau 3 e um delta de 0.05. O número total de gerações foi escolhido considerando pré-testes das execuções e a observação da evolução do valores de hipervolume e o tempo disponível para a execução dos experimentos.

Cada um dos casos foi executado 10 vezes nos quais os valores objetivo extraídos via ANSYS® foram normalizados ao serem repassados ao NSGA-II, de modo a ser um valor entre 0 e 1, a fim de que a consideração deles pelo algoritmo não fosse discrepante e que a visualização fosse mais adequada.

Os resultados obtidos serão apresentados no próximo capítulo.

### 4.7 Otimização Automática

Para estabelecer uma comparação com o método proposto por este documento, uma geometria com características semelhantes à peça de referência foi desenhada utilizando-se a plataforma Ansys Worckbench (ANSYS®, Release 21.1b), e em seguida importada e analisada pelo software Ansys Mechanical (ANSYS®, Release 21.1a). A malha de elementos finitos foi gerada automaticamente pelo software e refinada considerando curvatura e proximidade. Em seguida, avaliou-se o a Tensão Equivalente de Von Misses da peça inicial. Tendo finalizada a análise inicial da peça de referência, buscou-se promover otimizações topológicas utilizando-se o módulo "Structural Optimization". Neste módulo, definiu-se como restrição determinados valores para o volume: 40%, 50%, 60% e 70% para a peça ao final da otimização e uma tensão máxima limitada a 6.5e+5 Pa. Ao final deste processo, extraiu-se o stress equivalente da peça obtida, o volume e o tempo total da otimização.

### 5 Resultados

"E, na verdade, tenho também por perda todas as coisas, pela excelência do conhecimento de Cristo Jesus, meu Senhor; pelo qual sofri a perda de todas estas coisas, e as considero como escória, para que possa ganhar a Cristo, E seja achado nele, não tendo a minha justiça que vem da lei, mas a que vem pela fé em Cristo, a saber, a justiça que vem de Deus pela fé;".

Filipenses 3:8,9

Tendo executado os experimentos propostos e descritos no capítulo anterior, obtevese os tempos de execução e foram calculados os valores de hipervolume da Fronteira Pareto estimada que são mostrados nas tabelas 4 e 5.

	Experimento 01	Experimento 02	Experimento 03	Experimento 04
	•	•	•	-
1	1,288628	1,495184	1,566909	1,37772
2	1,458678	1,593233	2,029988	1,436612
3	1,319121	1,597885	1,507216	1,235713
4	1,394808	1,562474	1,867744	1,128182
5	1,345839	1,412682	1,497481	1,272803
6	1,312315	1,568396	1,664245	1,354151
7	1,355272	1,668351	1,350301	1,309355
8	1,322198	1,513369	1,393019	1,31979
9	1,362943	1,613074	1,574735	1,323164
10	1,546272	1,524419	1,511824	1,360962
Média	1,370607	1,554907	1,596346	1,311845
Desvio	0,07431	0,068013	0,198592	0,080845

Tabela 4 – Tempos de execução (em horas) das rodadas da otimização

Fonte: o Autor.

Com relação a tabela de tempos de otimização, observa-se o maior valor na coluna do experimento 03 e o menor valor na coluna do experimento 04. Vê-se que o maior desvio padrão é aquele contido na coluna do experimento 03 e o menor na coluna do experimento 02. Enquanto a maior e menor média são respectivamente encontradas nos experimentos 03 e 04. Destaca-se que o experimento 4, embora o espaço de busca esteja em dimensão maior que os demais (pois neste caso são 24 variáveis livres - 12 pesos e 12 abscissas dos pontos) o tempo médio (e também a mediana) de seus tempos está significativamente abaixo dos experimentos 2 e 3 (ambos com 12 variáveis livres).

A figura 20 apresenta graficamente os valores da tabela anterior. Veja que no primeiro experimento aproximadamente 75% dos valores estão no intervalo próximo de 1,3 horas e abaixo de 1,4 horas e que apenas um único valor se afasta consideravelmente do outros. No segundo caso, vê-se que não há valor que se afaste muito dos outros, porém, observa-se um conjunto um pouco mais abrangente em relação ao anterior partindo das proximidades de 1,4 horas e chegando perto de 1,7 horas. No terceiro caso, observa-se o conjunto mais

Box Plot - Tempos de Execução

2.0

(second)

2.0

0

1.8

0

1.4

1.2

1.2

Experimentos

Figura 20 – Box Plot dos tempos de execução.

Fonte: o Autor.

abrangente e a presença de dois valores que consideravelmente se afastam dos demais. Por fim, no último caso, observa-se um conjunto compacto, mas com um único valor afastado dos demais, sendo muito inferior.

Na tabela 5 apresentam-se os hipervolumes de cada um dos experimentos em cada uma das rodadas. E os dados desta tabela estão sintetizados graficamente na figura 21.

	Rodadas	Experimento 01	Experimento 02	Experimento 03	Experimento 04
Ì	1	0,294964	0,294965	0,28036	0,270764
	2	0,293661	0,290511	0,274136	0,283544
	3	0,283367	0,283691	0,274684	0,269922
	4	0,282943	0,269095	0,270602	0,272279
	5	0,287523	0,294406	0,270576	0,257468
	6	0,293986	0,283207	0,268975	0,271753
	7	0,282534	0,271983	0,28468	0,271205
	8	0,291674	0,259898	0,257738	0,280796
	9	0,278303	0,298837	0,269589	0,277228

Tabela 5 – Hipervolume das Fronteiras Pareto Estimadas

Fonte: o Autor.

0,275463

0,27268

0,006871

0,269658

0,272462

0,006776

0,276787

0,282338

0,012098

10

Média

Desvio

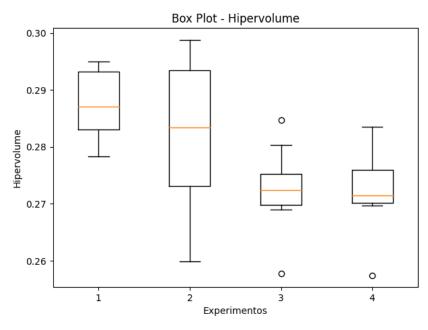
0,286648

0,28756

0,005486

Observa-se na figura 21 que os valores de hipervolume encontrados por meio do experimento 01 são em média mais elevados em comparação com os outros experimentos. Observa-se também que o experimento 02 é o que apresentou maior dispersão nos resultados.

Figura 21 – Box Plot dos hipervolumes calculados a partir das Fronteiras Pareto Estimadas.



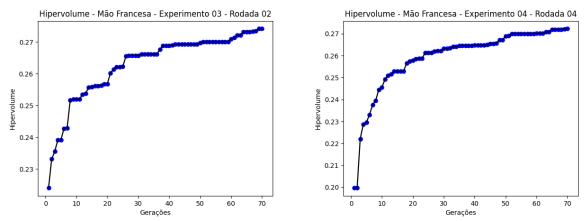
Fonte: o Autor.

Para exemplificar, a figura 22a apresenta a evolução do valor de Hipervolume da rodada 2 do experimento 3 (que foi a de maior tempo global de execução nos experimentos) e a figura 22b a evolução da rodada 4 do experimento 4 (que foi a de menor tempo global de execução nos experimentos). Observa-se para o maior tempo de execução: um rápido crescimento do valor de Hipervolume, atingindo 0,25 já na  $8^{\circ}$  geração. Em seguida, apesar do crescimento ser mais lento, vê-se que os valores se aproximam de 0,27 já próximo da geração 40. No caso da otimização de menor tempo, o gráfico atinge 0,25 próximo da  $12^{\circ}$  geração e se aproximando de 0,27 próximo da geração 50. No primeiro caso, os valores ultrapassam 0,27 após a geração 50 e depois de um período de lento crescimento reinicia uma curva um pouco mais acentuada ao final. Enquanto no segundo caso, os valores crescem lentamente após geração 50 e finalizam um pouco acima de 0,27.

A figura 23a apresenta seis gerações distribuídas ao longo de toda a otimização da rodada 2 do terceiro experimento. Coerente com aquilo que fora observado na análise do Hipervolume, há um grande salto entre a geração 1 e a 15, que começa inclusive a desenhar corretamente uma forma de fronteira. Em seguida, observa-se que a distância entre os pontos das gerações posteriores não cresce muito. A figura 23b apresenta a Fronteira Pareto estimada (pontos em vermelho) junto com o ponto (em azul) que representa a peça de referência.

Com relação aos resultados obtidos pelo menor tempo de execução, as figuras 24a e 24b ilustram respectivamente a evolução de 6 populações distribuídas ao longo da otimização e a Fonteira Pareto Estimada realçando o valor do caso base. Observando a evolução das populações, percebe-se uma população inicial espalhada nas regiões superiores do gráfico. Em seguida, é possível ver que as fronteiras vão se aproximando da Pareto Esti-

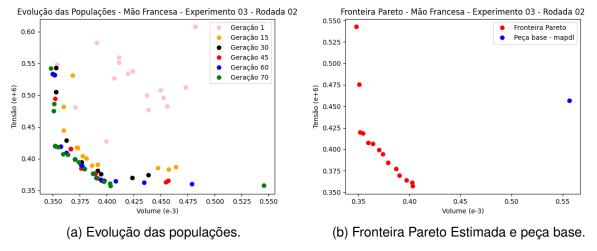
Figura 22 – Exemplos da evolução dos valores de hipervolume para o maior e o menor valor de tempo total de execução.



(a) Evolução do valor de hipervolume da rodada 2 (b) Evolução do valor de hipervolume da rodada 4 do experimento 3. do experimento 4.

Fonte: o autor.

Figura 23 – Exemplo da convergência das fronteiras estimadas para o caso de maior tempo de execução. Tensão em Pascal (Pa) e volume em metros cúbicos  $(m^3)$ .

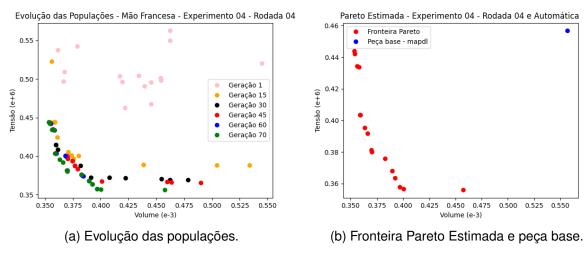


Fonte: o autor.

mada final ao mesmo tempo que há um pequena concentração de indivíduos no intervalo de  $[0,450;0,500]m^3$ . No final de todo o processo, a Fronteira Estimada permanece tendo uma solução próxima de 0,450, mas se concentra em valores de volume inferiores a 0,400.

A figura 25a apresenta as curvas de Hipervolume para o maior e menor valor entre aqueles contidos na tabela 5. Por meio da figura 25b, observa-se para o maior valor: um rápido crescimento do valor de Hipervolume, atingindo valores superiores a 0,26 já na  $8^{\circ}$  geração. Em seguida, apesar de inciar um período de pouco crescimento a partir da  $14^{\circ}$  geração, vê-se que os valores ultrapassam 0,28 próximo da geração 40. No caso da otimização de menor valor, observa-se um crescimento mais lento, atingindo 0,24 próximo da  $12^{\circ}$  geração e ultrapassando

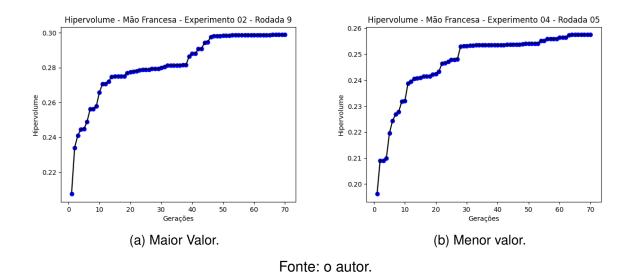
Figura 24 – Convergência da Otimização para o caso de menor tempo de execução. Tensão em Pascal (Pa) e volume em metros cúbicos  $(m^3)$ .



Fonte: o autor.

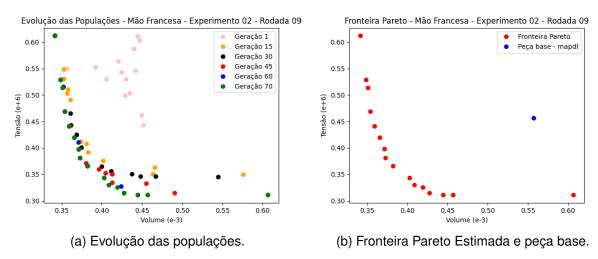
0,25 próximo da geração 30. No primeiro caso, os valores se aproximam de 0,30 ao final, enquanto no segundo caso, os valores crescem lentamente desde meados da  $30^{\circ}$  geração sem atingir 0,26.

Figura 25 – Evolução dos valores de Hipervolume para o maior e o menor valor de Hipervolume em todos os experimentos.



O comportamento observado nos gráficos de hipervolume podem ser rastreados por meio das figuras 26 e 27 . Na sub figura 26a mostra-se a evolução de 6 populações distribuídas ao longo da otimização. Por meio dessa imagem é possível identificar o período de rápida convergência até a 15 geração e uma evolução lenta porém contínua até a última geração. A figura 26b apresenta uma Fronteira Pareto Estimada bem espalhada apresentando um longo intervalo de valores tanto para a tensão quanto para o volume.

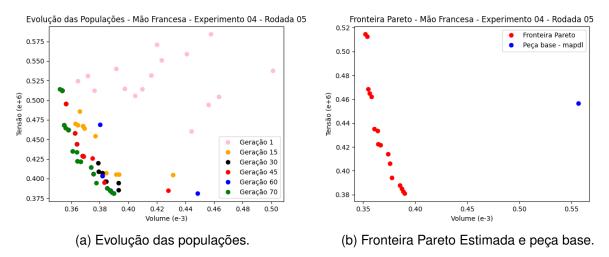
Figura 26 – Convergência da Otimização com maior valor de Hipervolume. Tensão em Pascal (Pa) e volume em metros cúbicos  $(m^3)$ .



Fonte: o autor.

Com relação ao segundo caso, observa-se rápida convergência para uma região compacta do gráfico até atingir a Fronteira Pareto Estimada. Como observado pelo valor de hipervolume obtido, a fronteira Pareto se insere em um pequeno espaço contido pelo intervalo a [0,40;0,52] no eixo das tensões.

Figura 27 – Convergência da Otimização de menor valor de Hipervolume. Tensão em Pascal (Pa) e Volume em metros cúbicos.



Fonte: o autor.

### 5.1 Comparação

Conforme definido no capítulo anterior, promoveu-se também a otimização topológica proprietária embutida no ANSYS para estabelecer uma comparação entre a metodologia automática aqui descrita e os resultados obtidos por aquela. Analisando a peça de base por

meio do Ansys Mechanical, obteve-se os valores de tensão e volume respectivamente iguais a  $0.55691(e-3)m^{-3}$  e 0.86166(e+6)Pa. Ao final do processo de otimização executada pela ferramenta embutida no ANSYS, extraiu-se os valores de tensão, volume e tempo total de execução. Estes valores podem ser visualizados por meio da tabela 6.

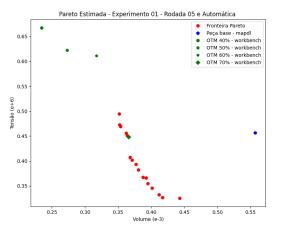
Tabela 6 – Otimizações automáticas promovidas pelo Ansys Mechanical. Tensão em Pascal (Pa) e volume em metros cúbicos  $(m^3)$ 

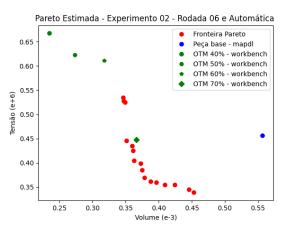
	Execução 03 (40%)	Execução 01 (50%)	Execução 02 (60%)	Execução 04 (70%)
Volume	0.23443(e-3)	0.2732(e-3)	0.31744(e-3)	0.36599(e-3)
Tensão	0.66748(e+6)	0.62253(e+6)	0.61127(e+6)	0.44792(e+6)
Tempo	3 min 14 s	2min54s	2 min 15 s	1 min 35 s

Fonte: o Autor.

Os resultados gráficos obtidos para cada uma dessas fronteiras podem ser vistos nos gráficos da figura 28d. Observa-se em todos os casos valores elevados de tensão para a otimização executada por meio do próprio Ansys em relação aos encontrados pela metodologia proposta utilizando python. Isso se deve, possivelmente devido a diferença de malha, assim como a natureza simulada das condições de contorno na otimização python-apdl. Contudo, observa-se que a otimização biobjetivo proposta pelo acoplamento python-apdl consegue explorar regiões do gráfico que não são exploradas pelo algoritmo de otimização proprietário do ansys, ainda que este atinja valores de volume inferiores aos encontrados na Frente de pareto estimada. Além disso, a frente de pareto possibilita ao projetista escolher entre varias possíveis soluções eficientes, enquanto o algoritmo de otimização topológica retorna apenas uma solução ao final. Desta maneira é possível afirmar que a otimização de forma utilizando a comunicação python-apdl tem como vantagem uma melhor exploração do espaço de pesquisa, apesar do longo tempo tomado para a otimização. Por sua vez, a abordagem topológica do Ansys retorna resultados em tempos inferiores, apesar da limitada busca no espaço de solução.

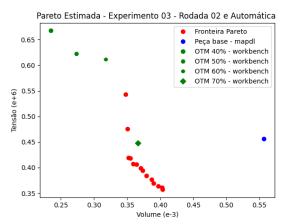
Figura 28 – Fronteira Pareto Estimada do valor médio dos experimentos, peça base e Otimizações Automáticas. Tensão em Pascal (Pa) e Volume em metros cúbicos ( $m^3$ ).

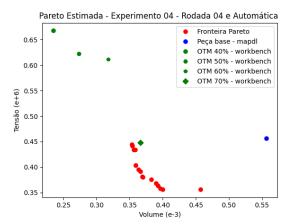




Experimento 01.

(a) Comparação do valor médio de Hipervolume do (b) Comparação do valor médio de Hipervolume do





(c) Comparação do valor médio de Hipervolume do (d) Comparação do valor médio de Hipervolume do Experimento 03. Experimento 04.

Fonte: o autor.

# 6 Considerações finais

"You deserve the greater glory Overcome, I lift my voice To the King in need of nothing Empty handed I rejoice. You deserve the greater glory Overcome with joy I sing By Your love I am accepted You're a good and gracious King"

CityAlight

Foi executado por meio deste trabalho um procedimento de comunicação automática entre o algoritmo de otimização multiobjetivo contido pela biblioteca pymoo e o software de Simulação Ansys Mechanical APDL. Por meio da implementação padronizada da classe Problema, é facilitada a otimização do problema aqui proposto por meio dos outros algoritmos ali contidos. Portanto, pode-se afirmar que é possível, por meio da metodologia aqui implementada, aumentar o escopo de métodos e estratégias de otimização para além dos limites do software de simulação.

Por meio dos resultados obtidos observamos o potencial dos métodos evolucionários de explorar o espaço de solução e promover múltiplas possibilidades de soluções promissoras. Por meio dos resultados encontrados, viu-se que tal otimização promoveu melhorias em ambos os critérios de objetivos escolhidos para otimização em relação ao caso base.

Por meio da comparação das soluções, observou-se que, diferentemente dos resultados obtidos pela metodologia python-apdl, os resultados do algoritmo de otimização do software Ansys Mechanical ofereceram uma única solução ao final. Por outro lado, os tempos de execução das soluções fornecidas pelo software proprietário da Ansys eram significativamente inferiores.

A partir do que foi executado neste trabalho sugere-se realização de mais experimentos e mais execuções para cada, a fim de aumentar o entendimento do impacto da quantidade dos pontos e pesos e até da própria presença dos pesos. Justifica-se esta proposta, pois vêse que a média de tempo de execução superior está justamente entre os experimentos 2 e 3, que possuem 12 variáveis livres. No primeiro caso os pesos são livres dentro do intervalo [0, 15], enquanto no segundo caso são fixos em 1, porém com o dobro de pontos de controle. Ao mesmo tempo, observou-se que o menor tempo de execução esteve relacionado ao experimento 04, cujo espaço de pesquisa era o maior (24 variáveis livres) e cuja média de valores de Hipervolume não não foi a menor entre as outras 3. Apesar disso, observa-se valores superiores de hipervolume para os experimentos com 6 pontos de controle com e sem variação de pesos. Logo, sugere-se investigar mais a fundo qual o impacto da presença dos pesos a medida que o número de pontos de controle aumentam.

Por fim, tendo em vista a ampla variedade de Métodos de Otimização Multiobjetivo no contexto da Computação Evolucionária, sugere-se a comparação de outros da mesma categoria e até mesmo de outros que lidam com muitos objetivos. Tal sugestão visa a exploração do tema da computação evolucionária multiobjetivo no contexto da otimização estrutural.

## Referências

- ANSYS®. Ansys Mechanical. Release 21.1. Citado na página 38.
- ANSYS®. Ansys Workbench. Release 21.1. Citado na página 38.
- ANSYS®. Mechanical APDL. Release 21.1. Citado na página 16.
- BATHE, K.-J. (Ed.). *Finite element procedures*. 2nd edition. ed. Watertown, MA: K.J. Bathe, 2014. ISBN 9780979004957. Citado nas páginas 13 e 27.
- BINGOL, O. R.; KRISHNAMURTHY, A. NURBS-Python: An open-source object-oriented NURBS modeling framework in Python. *SoftwareX*, Elsevier, v. 9, p. 85–94, 2019. Citado nas páginas 30 e 33.
- Blank, J.; Deb, K. Pymoo: Multi-objective optimization in python. *IEEE Access*, v. 8, p. 89497–89509, 2020. Citado nas páginas 30 e 33.
- CASTORANI, V. et al. A CAD-based method for multi-objectives optimization of mechanical products. *Computer-Aided Design and Applications*, v. 14, n. 5, p. 563–571, jul. 2017. ISSN 1686-4360. Disponível em: <a href="http://www.cad-journal.net/files/vol\_14/Vol14No5.html">http://www.cad-journal.net/files/vol\_14/Vol14No5.html</a>. Citado nas páginas 13 e 29.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, abr. 2002. ISSN 1089778X. Disponível em: <a href="http://ieeexplore.ieee.org/document/996017/">http://ieeexplore.ieee.org/document/996017/</a>>. Citado nas páginas 21, 23 e 24.
- DITTMER, J. P. et al. Mesh Optimization Using a Genetic Algorithm to Control Mesh Creation Parameters. *Computer-Aided Design and Applications*, v. 3, n. 6, p. 731–740, jan. 2006. ISSN 1686-4360. Disponível em: <a href="http://www.cad-journal.net/files/vol\_3/Vol3No6.html">http://www.cad-journal.net/files/vol\_3/Vol3No6.html</a>. Citado nas páginas 31 e 32.
- FILHO, A. *Elementos Finitos A base da tecnologia CAE*. Saraiva Educação S.A., 2018. ISBN 9788536525495. Disponível em: <a href="https://books.google.com.br/books?id=JrdiDwAAQBAJ">https://books.google.com.br/books?id=JrdiDwAAQBAJ</a>. Citado nas páginas 13, 24 e 25.
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra / Coimbra University Press, 2012. (Ensino). ISBN 9789892601502. Disponível em: <a href="https://books.google.com.br/books?id=9Di5CwAAQBAJ">https://books.google.com.br/books?id=9Di5CwAAQBAJ</a>. Citado nas páginas 17, 18, 20 e 21.
- GUIRGUIS, D. et al. Evolutionary Black-Box Topology Optimization: Challenges and Promises. *IEEE Transactions on Evolutionary Computation*, v. 24, n. 4, p. 613–633, ago. 2020. ISSN 1089-778X, 1089-778X, 1941-0026. Disponível em: <a href="https://ieeexplore.ieee.org/document/8906012/">https://ieeexplore.ieee.org/document/8906012/</a>». Citado na página 14.
- HSIAO, J. C. et al. Shape Design Optimization of a Robot Arm Using a Surrogate-Based Evolutionary Approach. *Applied Sciences*, v. 10, n. 7, p. 2223, mar. 2020. ISSN 2076-3417. Disponível em: <a href="https://www.mdpi.com/2076-3417/10/7/2223">https://www.mdpi.com/2076-3417/10/7/2223</a>. Citado na página 29.
- JANKOVICS, D. et al. Developing topology optimization with additive manufacturing constraints in ansys®. *IFAC-PapersOnLine*, v. 51, n. 11, p. 1359–1364, 2018. ISSN 2405-8963. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018. Disponível em:

Referências 50

<a href="https://www.sciencedirect.com/science/article/pii/S2405896318314642">https://www.sciencedirect.com/science/article/pii/S2405896318314642</a>. Citado na página 15

- JOSHI, M.; GYANCHANDANI, M.; WADHVANI, D. R. Analysis Of Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing On Benchmark Functions. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). Erode, India: IEEE, 2021. p. 1152–1157. ISBN 9781665403603. Disponível em: <a href="https://ieeexplore.ieee.org/document/9418458/">https://ieeexplore.ieee.org/document/9418458/</a>>. Citado na página 19.
- KICINGER, R.; ARCISZEWSKI, T.; JONG, K. D. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers & Structures*, v. 83, n. 23-24, p. 1943–1978, set. 2005. ISSN 00457949. Disponível em: <a href="https://linkinghub.elsevier.com/retrieve/pii/S0045794905001161">https://linkinghub.elsevier.com/retrieve/pii/S0045794905001161</a>. Citado nas páginas 18, 19 e 20.
- KROHE, F. Numerical Surface Reconstruction with Non-uniform Rational B-splines: Computational Solution in Matlab and ANSYS APDL. [S.I.]: Universitätsbibliothek der Technischen Universität Hamburg-Harburg, 2020. Google-Books-ID: 80KWzgEACAAJ. Citado na página 27.
- LI, B. et al. Many-Objective Evolutionary Algorithms: A Survey. *ACM Computing Surveys*, v. 48, n. 1, p. 13:1–13:35, set. 2015. ISSN 0360-0300. Disponível em: <a href="https://doi.org/10.1145/2792984">https://doi.org/10.1145/2792984</a>>. Citado nas páginas 14, 17 e 18.
- LIM, T. Y. Structured population genetic algorithms: a literature survey. *Artificial Intelligence Review*, v. 41, n. 3, p. 385–399, mar. 2014. ISSN 0269-2821, 1573-7462. Disponível em: <a href="http://link.springer.com/10.1007/s10462-012-9314-6">http://link.springer.com/10.1007/s10462-012-9314-6</a>. Citado nas páginas 19 e 20.
- LIU, T.-D. et al. Structural optimization of Pt–Pd–Rh trimetallic nanoparticles using improved genetic algorithm. *Journal of Alloys and Compounds*, v. 663, p. 466–473, abr. 2016. ISSN 09258388. Disponível em: <a href="https://linkinghub.elsevier.com/retrieve/pii/S0925838815319381">https://linkinghub.elsevier.com/retrieve/pii/S0925838815319381</a>. Citado na página 29.
- LUH, G.-C.; LIN, C.-Y. Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, v. 9, n. 4, p. 1343–1353, set. 2009. ISSN 15684946. Disponível em: <a href="https://linkinghub.elsevier.com/retrieve/pii/S1568494609000672">https://linkinghub.elsevier.com/retrieve/pii/S1568494609000672</a>. Citado na página 15.
- LóPEZ, J.; ANITESCU, C.; RABCZUK, T. CAD-compatible structural shape optimization with a movable Bézier tetrahedral mesh. *Computer Methods in Applied Mechanics and Engineering*, v. 367, p. 113066, ago. 2020. ISSN 00457825. Disponível em: <a href="https://linkinghub.elsevier.com/retrieve/pii/S0045782520302504">https://linkinghub.elsevier.com/retrieve/pii/S0045782520302504</a>>. Citado na página 26.
- MORTAZAVI, A.; TOğAN, V. Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. *Structural and Multidisciplinary Optimization*, v. 54, n. 4, p. 715–736, out. 2016. ISSN 1615-1488. Disponível em: <a href="https://doi.org/10.1007/s00158-016-1449-7">https://doi.org/10.1007/s00158-016-1449-7</a>>. Citado na página 14.
- MULLER, T.; KLASHORST, E. A Quantitative Comparison Between Size, Shape, Topology and Simultaneous Optimization for Truss Structures. v. 14, dez. 2017. Citado nas páginas 14 e 26.
- NG, A.; SIEGMUND, F.; DEB, K. Reference point based evolutionary multi-objective optimization with dynamic resampling for production systems improvement. v. 20, nov. 2018. Citado na página 15.

Referências 51

PIEGL, L.; TILLER, W. *The NURBS Book*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. (Monographs in Visual Communications). ISBN 9783642973871 9783642973857. Disponível em: <a href="http://link.springer.com/10.1007/978-3-642-97385-7">http://link.springer.com/10.1007/978-3-642-97385-7</a>>. Citado nas páginas 27 e 28.

ROGERS, D. F. *An introduction to NURBS: with historical perspective.* San Francisco: Morgan Kaufmann Publishers, 2001. ISBN 9781558606692. Citado na página 27.

SU, Y. et al. Shape Generation of Grid Structures by Inverse Hanging Method Coupled with Multiobjective Optimization: Inverse hanging method coupled with multiobjective optimization. *Computer-Aided Civil and Infrastructure Engineering*, v. 33, n. 6, p. 498–509, jun. 2018. ISSN 10939687. Disponível em: <a href="https://onlinelibrary.wiley.com/doi/10.1111/mice.12361">https://onlinelibrary.wiley.com/doi/10.1111/mice.12361</a>. Citado na página 29.

WAZLAWICK, R. S. *Metodologia de pesquisa para ciência da computação*. [s.n.], 2014. OCLC: 902734376. ISBN 9788535277838. Disponível em: <a href="http://www.sciencedirect.com/science/book/9788535277821">http://www.sciencedirect.com/science/book/9788535277821</a>. Citado na página 30.

ZHOU, A. et al. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 32–49, mar. 2011. ISSN 22106502. Disponível em: <a href="https://linkinghub.elsevier.com/retrieve/pii/S2210650211000058">https://linkinghub.elsevier.com/retrieve/pii/S2210650211000058</a>. Citado na página 18.

ZULIANI, J. B. Q. Uma metodologia multiobjetivo para otimização topológica. fev. 2016. Disponível em: <a href="https://repositorio.ufmg.br/handle/1843/33416">https://repositorio.ufmg.br/handle/1843/33416</a>. Citado na página 22.