

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
DEPARTAMENTO ENGENHARIA MECÂNICA

Maicon Jr. Brum Nágera

**OTIMIZAÇÃO PARAMÉTRICA DE ESTRUTURAS RETICULADAS DO
TIPO PÓRTICOS TRIDIMENSIONAIS UTILIZANDO ALGORITMO
TLBO MODIFICADO**

Santa Maria, RS
2020

Maicon Jr. Brum Nágera

**OTIMIZAÇÃO PARAMÉTRICA DE ESTRUTURAS RETICULADAS DO TIPO
PÓRTICOS TRIDIMENSIONAIS UTILIZANDO ALGORITMO TLBO MODIFICADO**

Monografia de conclusão apresentada ao Curso de Graduação em Engenharia Mecânica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Mecânica.**

Orientador: Prof. Dr. René Quispe Rodriguez

Santa Maria, RS
2020

Maicon Jr. Brum Nágera

**OTIMIZAÇÃO PARAMÉTRICA DE ESTRUTURAS RETICULADAS DO TIPO
PÓRTICOS TRIDIMENSIONAIS UTILIZANDO ALGORITMO TLBO MODIFICADO**

Monografia de conclusão apresentada ao Curso de Graduação em Engenharia Mecânica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Mecânica.**

Aprovado em 26 de agosto de 2020

René Quispe Rodriguez, Dr. (UFSM)
(Presidente / Orientador)

Tiago dos Santos, Dr. (UFSM)

Marcos Daniel de F. Awruch, Dr. (UFSM)

Santa Maria, RS

2020

RESUMO

OTIMIZAÇÃO PARAMÉTRICA DE ESTRUTURAS RETICULADAS DO TIPO PÓRTICOS TRIDIMENSIONAIS

AUTOR: Maicon Jr. Brum Nágera

ORIENTADOR: René Quispe Rodriguez

Com o intenso desenvolvimento tecnocientífico vivido nas últimas décadas, atualmente conta-se com inúmeras técnicas e métodos para se otimizar estruturas. Uma delas é a otimização paramétrica, que apresenta, entre outras vantagens, a facilidade de implementação e a adaptabilidade a projetos específicos. A otimização estrutural é uma ferramenta que auxilia o projetista na confiabilidade e garantia de bons resultados do projeto, tornando-se portanto uma importante aliada do engenheiro no processo de análise e tomada de decisão. No presente trabalho, objetiva-se otimizar o peso de estruturas reticuladas do tipo pórticos tridimensionais, que são muito utilizadas em estruturas de componentes mecânicos, como chassis, torres, guindastes, etc. Outro ponto importante tratando-se de otimização estrutural é quanto à rotina computacional utilizada. No presente trabalho, foi utilizado o *TLBO* Modificado, um algoritmo meta-heurístico que baseia-se na interação entre alunos e professores para a transmissão de conhecimento e melhorar assim a aptidão de todo o grupo de projeto. A implementação computacional foi feita através do *software MatLab*[®], e sua verificação foi feita contrastando os resultados obtidos com os disponíveis na literatura. É importante ressaltar que o trabalho, embora aborde estruturas reticuladas - que são bem difundidas na literatura, as modela como pórtico, e não treliça, como a maioria das vezes é feito. Isso abre a possibilidade de se contrapor as distinções de resultados em ora modelar uma estrutura como um, ora como outro. Os resultados obtidos evidenciam que, por considerarem flexão e torção nos elementos - esforços até então desconsideradas enquanto treliça, pórticos são mais propensos a falhas, necessitando portanto um correto dimensionamento e, na maioria dos casos, resultando em estruturas com maior peso. Por fim, todos os resultados, de verificação e autorais, são mostrados e discutidos, e as devidas conclusões e considerações são feitas, provando a coerência não só dos resultados, mas de toda a metodologia adotada. Além disso, faz-se sugestões para trabalhos futuros seguindo a mesma linha deste.

Palavras-chave: Otimização Estrutural. *TLBO* Modificado. Pórticos Tridimensionais.

ABSTRACT

PARAMETRIC OPTIMIZATION OF TRIDIMENSIONAL FRAME STRUCTURES

AUTHOR: Maicon Jr. Brum Nágera

ADVISOR: René Quispe Rodriguez

With the intense techno-scientific development experienced in the last decades, currently there are countless techniques and methods to optimize structures. One of them is parametric optimization, which has, among other advantages, ease of implementation and adaptability to specific projects. Structural optimization is a tool that assists the designer in the confiability and guarantee of good project results, thus becoming an important ally of the engineer in the analysis and decision-making process. In the present work, the objective is to optimize the weight of tridimensional frames reticulated structures, which are widely used in structures of mechanical components, such as chassis, towers, cranes, etc. Another important point when it comes to structural optimization is regarding the computational routine used. In the present work, the Modified TLBO was used, a meta-heuristic algorithm that is based on the interaction between students and teachers for the transmission of knowledge and thus improving the aptitude of the entire project group. The computational implementation was performed using MatLab textsuperscript textregistered software, and its verification was done by contrasting the results obtained with those available in the literature. It is important to emphasize that the work, although it deals with reticulated structures - which are well disseminated in the literature, models them as a frame, and not a truss, as most of the time it is done. This opens up the possibility of opposing the distinctions of results, at times modeling a structure as one, now as another. The results obtained show that, because they consider flexion and torsion in the elements - efforts hitherto disregarded as a truss, frames are more prone to failure, therefore requiring a correct dimensioning and, in most cases, resulting in structures with greater weight. Finally, all the results of verification and authorship are shown and discussed, and the appropriate conclusions and considerations are made, proving the consistency not only of the results, but of the entire methodology adopted. In addition, suggestions for future work are made along the same lines.

Palavras-chave: Structural Optimization. Modified TLBO. Tridimensional Frames.

Lista de Figuras

1	Exemplo de estruturas reticuladas.	2
2	Exemplo de um pórtico tridimensional. Fonte: Ray (2012).	10
3	Composição de um elemento de pórtico bidimensional a partir dos elementos de barra e de viga. Fonte: autor.	11
4	Representação das Otimizações Paramétrica (a), de Forma (b), e Topológica (c). Fonte: Sigmund (2013)	16
5	Classificação dos principais Métodos de Otimização. Fonte: Yang (2010). Traduzido pelo Autor	16
6	Distribuição de notas em uma turma para dois professores diferentes. Fonte: Farshchin (2014). Traduzido pelo Autor	18
7	Ilustração da Fase Aluno, quando (a) aluno q é melhor que p e (b) aluno p melhor que q Fonte: Farshchin (2014). Traduzido pelo Autor	19
8	Organograma com a estrutura do algoritmo. Fonte: autor.	22
9	Organograma com a estrutura do algoritmo. Fonte: autor.	24
10	Primeiro caso para verificação do MEF para pórticos. Fonte: Ferreira (2009).	27
11	Segundo caso para verificação do MEF para pórticos. Fonte: Ferreira (2009).	28
12	Somatório de forças (esq.) e de momentos (dir.) para nó 1. Fonte: autor.	31
13	Modelagem da estrutura no <i>software CREO</i> (esq.) e resultado da simulação estática realizada por este (dir.). Fonte: autor.	32
14	Estrutura do Exemplo 1, com o número de nós indicados (esq.) e após a aplicação das cargas externas (dir). Fonte: autor.	33
15	Análise de convergência do Exemplo 1. Fonte: autor.	34
16	Estrutura do Exemplo 2, com o número de nós indicados (esq.) e após a aplicação das cargas externas (dir). Fonte: autor.	36
17	Análise de convergência do Exemplo 2. Fonte: autor.	38
18	Análise de convergência do Exemplo 3. Fonte: autor.	41
19	Análise de convergência do Exemplo 4. Fonte: autor.	42

Lista de Tabelas

1	Comparação de resultados obtidos por Ferreira (2009) e o presente trabalho para o caso 1. Fonte: autor.	27
2	Comparação de resultados obtidos por Ferreira (2009) e o presente trabalho para o caso 2. Fonte: autor.	29
3	Comparação entre tensões obtidas pelo <i>CREO</i> e pelo trabalho para os três elementos da estrutura. Fonte: autor.	32
4	Dados de entrada do algoritmo. Fonte: autor.	33
5	Características do exemplo 1. Fonte: autor.	34
6	Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura em diferentes trabalhos com o do presente para o Exemplo 1. Fonte: autor.	35
7	Características do exemplo 2. Fonte: autor.	36
8	Duas condições de carregamentos do Exemplo 2. Fonte: autor.	37
9	Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura em diferentes trabalhos com o do presente para o Exemplo 2. Fonte: autor.	37
10	Características do exemplo 3. Fonte: autor.	40
11	Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura ora modelado com treliça, ora como pórtico para o Exemplo 3. Fonte: autor.	40
12	Características do exemplo 4. Fonte: autor.	41
13	Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura ora modelado com treliça, ora como pórtico para o Exemplo 4. Fonte: autor.	43

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Objetivo do Trabalho	3
1.3	Organização do Trabalho	4
2	Revisão Bibliográfica	5
3	Fundamentação Teórica	9
3.1	Método dos Elementos Finitos	9
3.1.1	Formulação Local do Elemento de Pórtico	10
3.1.2	Formulação Global do Elemento de Pórtico	12
3.2	Otimização Estrutural	14
3.2.1	Tipos de Otimização	15
3.2.2	Métodos de Otimização	15
3.3	Algoritmo TLBO Modificado	17
3.3.1	Fase Professor	18
3.3.2	Fase Aluno	19
4	Implementação Computacional	21
4.1	Organização do Algoritmo	21
4.2	Funcionamento do Algoritmo	23
4.3	Modificações no Algoritmo	23
5	Resultados e Discussões	26
5.1	Verificação MEF para Pórticos	26
5.2	Verificação TLBO para Treliças	32
5.2.1	Exemplo 1 : Torre de 25 barras	32
5.2.2	Exemplo 2: Torre de 72 barras	35
5.3	TLBO para Pórticos	38
5.3.1	Exemplo 3 : Torre de 25 barras	39
5.3.2	Exemplo 4 : Torre de 72 barras	40
6	Conclusão	44
6.1	Sugestão para Trabalhos Futuros	44

1 Introdução

1.1 Contextualização

A análise de estruturas sempre se destacou como um dos ramos mais importantes da engenharia, em especial da mecânica, civil, naval, aeronáutica, aeroespacial, entre outras. Com o expressivo progresso do desenvolvimento tecnológico sofrido pela humanidade nas últimas décadas, a área sofreu também uma relevante evolução. A cada dia mais, projetos de estruturas mecânicas necessitam ser o mais ótimas possíveis, uma vez que se tenha definido qual grandeza deseja-se otimizar (peso, custo, etc.). Destaca-se, dentre as razões para esta necessidade de otimização a diminuição de custos com produção e matéria prima, o melhor aproveitamento de material e a minimização do impacto ambiental causado; garantindo primordialmente, é claro, segurança ao projeto.

É nesse contexto que se insere a ideia de otimização estrutural. da Fonseca (2007) destaca que, convencionalmente, o projeto estrutural dependia unicamente da experiência do engenheiro projetista, sendo essa obtida pelo método de tentativa e erro. A ideia de otimização estrutural vem de forma a desvincular o processo de tomada de decisão da experiência prévia necessária, e garantindo assim que determinada estrutura mecânica seja projetada e desenvolvida da melhor forma possível para cada situação. É evidente que a experiência do engenheiro, através de seu senso crítico, está intrinsecamente ligado ao sucesso da otimização; no entanto, esta é uma ferramenta que, quando devidamente desenvolvida e utilizada, é de grande valia ao engenheiro estrutural no processo de tomada de decisões.

É notável que todo esse incrível desenvolvimento tecnológico está atrelado ao desenvolvimento computacional. O que no passado despendia de muito recurso - intelectual, material e temporal, atualmente é processado em poucos segundos. Isso permite a implementação de métodos numéricos e computacionais robustos. A análise e otimização estrutural não teriam se desenvolvido da forma que se desenvolveram sem essa ferramenta. Faz-se então um destaque fastuoso e incisivo ao Método dos Elementos Finitos - MEF. Esse método, sem muita modéstia, revolucionou a forma com que interpretamos e modelamos o comportamento de estruturas físicas. É claro que o alicerce da análise estrutural está nas equações diferenciais do método analítico descrito desde o século XVIII. Porém a aplicabilidade desses conceitos é limitada para problemas reais de estruturas complexas, impasse esse contornado pelo MEF. O MEF tem por função a simulação do comportamento estruturas sujeitas a carregamentos através da subdivisão do domínio contínuo - que é a estrutura real, em um número limitado de subdomínios - elementos. Isso é feito modelando funções de forma para cada um destes elementos. Atualmente, o MEF é extremamente difundido em vários âmbitos da engenharia, mas sua importância para a análise e otimização estrutural é ímpar.

A análise e projeto de estruturas mecânicas são duas atribuições fundamentais do engenheiro. É importante esclarecer que estas podem ser modeladas de diversas formas diferentes, de acordo

com o problema físico imposto. No caso do trabalho, será abordado estruturas ditas reticuladas do tipo pórticos tridimensionais. Por reticulada, entende-se toda estrutura composta por barras esbeltas, interligadas entre si em algum arranjo geométrico por juntas (conexões, soldas, ligações parafusadas, ..). A figura 1 trás alguns exemplos de estruturas reticuladas. O conceito de pórticos tridimensionais diz respeito à direção dos esforços considerados, como será visto adiante.

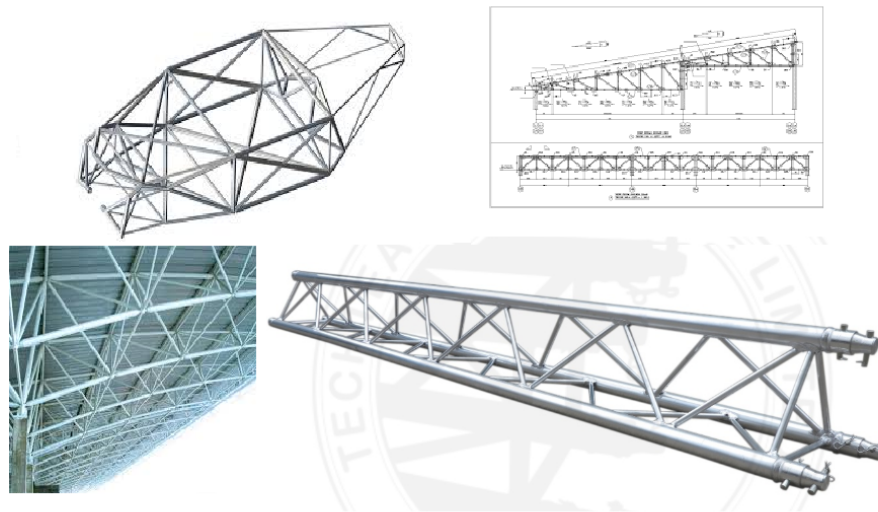


Figura 1: Exemplo de estruturas reticuladas.

Outro ponto importante refere-se à otimização, em seu sentido amplo. Corriqueiramente, atribui-se o sentido semântico de "otimizar" à "melhorar". De fato, otimizar busca obviamente melhorar, porém o ponto é que otimização, no contexto de engenharia, é uma melhora orientada por parâmetros. É uma melhora não só provada, mas desenvolvida quantitativamente. Otimização é, portanto, o processo de se obter o melhor resultado sob certas restrições e condições, (Rao, 2019). Chandrupatla (2019) exemplificam otimização com sua manifestação em fenômenos naturais, como: em metais e suas ligas, átomos sempre tomam a posição de menor energia para formar estruturas cristalinas; ou um líquido que, deixado sob gravidade zero, assume a forma de menor relação volume-superfície possível - a de esfera.

A otimização estrutural é apenas um ramo desse universo de possibilidades, mas não por isso é única. Há ainda diversas formas de como se otimizar uma estrutura. Tratar-se-á sobre isso mais a frente, porém uma otimização paramétrica pode ser considerada uma das mais simples, visto que ela apenas altera os parâmetros que influenciam diretamente o comportamento da estrutura. Parâmetros estes, vale lembrar, que são definidos pelo engenheiro em sua análise crítica do projeto. Por outro lado, nem por isso ela não é eficiente, e pode em determinados casos, ser preferencial.

Um outro ponto importante no que diz respeito à otimização estrutural é como, de forma objetiva e operacional, ela é feita. A resposta são os algoritmos, chamados algoritmos de otimização. Entende-se por algoritmo uma rotina computacional escrita e desenvolvida para armazenar, processar e manipular os dados de entrada do problema, e a partir deles encontrar uma solução. No presente trabalho, por exemplo, toda essa rotina é desenvolvida através do *software Matlab*[®]. Este é uma poderosa ferramenta computacional que permite uma fácil, ágil e precisa manipulação de dados na forma de arranjos matriciais, o que vem muito a calhar com a proposta tanto do MEF quanto da otimização estrutural.

1.2 Objetivo do Trabalho

Os objetivos do trabalho serão divididos em

I Objetivo Geral

II Objetivos Específicos

sendo os objetivos específicos aqueles que conduzem o trabalho ao objetivo geral

Objetivo Geral

Desenvolvimento de uma rotina computacional, escrita em linguagem compatível com o *software Matlab*[®] e utilizado o algoritmo *Teach Learn Based Optimization - TLBO*, para realização de uma otimização estrutural paramétrica visando minimização de peso de estruturas do tipo pórticos tridimensionais, dado um conjunto de informações de entrada (*input*). É também objetivo que essa rotina forneça de volta ao usuário, de forma clara e elucidativa, seus resultados encontrados (*output*). As estruturas utilizadas nas otimizações são casos de *benchmark* da literatura e com isso é possível confrontar os resultados obtidos por outros autores com o do presente trabalho.

Objetivos Específicos

- Realização de uma revisão na bibliografia pertinente ao assunto a fim de avaliar o estado da arte no qual o tema se encontra, como sua evolução ao longo do tempo;
- Implementação de uma rotina computacional capaz de aplicar a metodologia prevista, integrando o algoritmo de otimização com o MEF;
- Definição e organização de uma interface de saída simples e clara para informar ao usuário os resultados obtidos;

- Validação das rotinas (tanto do MEF quanto da otimização) a partir de exemplos presentes e difundidos na literatura com resultado já conhecidos (*benchmark*) a fim de garantir a sua eficácia e eficiência;
- Aplicação do algoritmo nos mesmos casos de *benchmark*, finalmente modelados como pórticos tridimensionais, a fim do confronto dos resultados para estrutura ora modelada como treliça, ora como pórtico;

1.3 Organização do Trabalho

O presente trabalho está organizado na forma de capítulos, ou secções, dividindo-o assim em seis partes. No primeiro capítulo, é feita uma introdução ao tema, apresentando sua contextualização, seus objetivos e também descrevendo como este está organizado. No segundo capítulo, uma revisão da literatura de forma cronológica com os principais autores do tema e suas contribuições ao estado da arte.

No terceiro capítulo encontra-se a base da formulação teórica pertinente ao trabalho, subdividido em Método dos Elementos Finitos, Otimização Estrutural e Algoritmo TLBO Modificado (*Modified Teach Learning Based Optimization*). No quarto capítulo, é descrito em detalhes como é estruturado e como funciona o algoritmo responsável pela otimização, seus parâmetros de entrada e de saída.

No quinto capítulo, primeiramente é feita a validação dos algoritmos do MEF para pórticos tridimensionais e também do algoritmo TLBO para treliças, com exemplos clássicos da literatura onde a solução ótima já é conhecida. A partir daí então é proposto um código TLBO para pórticos, com os mesmos exemplos já validados como treliça. Por fim, no sexto e último capítulo é feita a conclusão do trabalho, apresentando seus pontos principais e a discussão dos resultados obtidos, além de sugestões de trabalhos futuros a partir deste.

2 Revisão Bibliográfica

A ideia de otimização de processos é praticamente inerente à espécie humana. Talvez as primeiras técnicas de otimização tenham vindo da pré história, quando uma ferramenta melhor em relação a outra garantia a sobrevivência de uma tribo. Yang (2010) diz não ser exagero que encontrar a solução de problemas de otimização, intencional ou inconscientemente, seja tão antigo quanto a própria história humana. Ainda nesse contexto, Sant'Anna (2002) destaca que "as técnicas melhor sucedidas eram obtidas por acaso, através da tentativa e erro"; e cita outros exemplo, como a roda ou a roldana.

Porém, quando se fala em otimização como se conhece hoje, em especial a estrutural, o embasamento desse conhecimento foi feito somente a partir do século XIX e consolidado no século passado; embora o primeiro registro do uso de conceitos que viriam posteriormente a ser chamados de otimização remetam ao trabalho de Galileu Galilei, no ano de 1638; (da Fonseca, 2007).

De acordo com Chandrupatla (2019), já em 1847 o uso do método de gradiente - que requer um função objetivo contínua e derivável em relação à(s) sua(s) variável(is) de projeto, foi empregado por Cauchy para minimizações. O trabalho que pode ser considerado o precursor em otimização estrutural é de James C. Maxwell, em 1870, no qual ele utiliza técnicas matemáticas para minimizar o volume de estruturas submetidas a esforços axiais. (Maxwell, 1870).

Já no início século XX, Michell deu continuidade ao trabalho iniciado por Maxwell e chega a resultados expressivos com a minimização do peso de estruturas treliçadas observando a tensão em cada barra. Embora relevante, Michell não obteve o devido reconhecimento por seu trabalho por parte de seus contemporâneos. Ele só veio a ser referenciado e creditado em trabalhos futuros, como os de Cox (1958), Schmit (1960) e Fox (1965), (Coutinho, 2006).

Esses últimos trabalhos se caracterizaram pela disseminação e popularização de técnicas numérico-computacionais no auxílio de problemas de engenharia, inclusive com performances não-lineares de variáveis. Contudo, na prática, eles eram intuitivos e careciam de uma solidez matemática, (Coutinho, 2006). Problema esse que só viria a ser contornado devido à revolução tecnológica e computacional desencadeada a partir do final do século XX. Silva (2018a) também comenta que, na década de 60, devido ao surgimento de computadores e do MEF, "problemas práticos de otimização estrutural passam a ser estudados usando otimização paramétrica, ou seja, alterando-se apenas as dimensões - ou as razões de dimensões". É a partir da década de 80 então que aparecem *softwares* comerciais de otimização estrutural, assim como os já existentes de Elementos Finitos começam a introduzir versões com otimização em seus pacotes, (Silva, 2018a).

Destaca-se alguns trabalhos relevantes na época, como: Fox (1965), foi pioneiro ao conciliar otimização numérica com o MEF, mesmo esse sendo ainda uma tecnologia emergente. Com esse trabalho, ele conseguiu esclarecer o uso de técnicas de programação matemáticas para resolver

problemas não-lineares de projetos de estruturas sujeitas a múltiplos carregamentos e restrições. (Schimit, 1960 *apud* Vanderplaats, 1999). Chan (1965) expande o trabalho de Michell para análise de trabalhos e deformações virtuais em treliças e também pórticos, e trás exemplos de aplicações voltadas para o segmento aeronáutico. Prager (1977) também dá continuação ao trabalho de Michell e considera as juntas e conexões entre barras no projeto, otimizando-as junto à estrutura.

Durante a década de 80, importantes contribuições à literatura foram produzidas. Destaca-se os livros de Vanderplaats (1984), Gürdal (2012) e Arora (2004), onde é apresentado tanto um embasamento teórico robusto como exemplos didáticos e práticos, sendo por isso e entre outros, utilizados até hoje no estudo da otimização estrutural. Muitos artigos interessantes também foram publicados no período. Destaca-se, entre todos, alguns como Johnson (1986) que realiza uma otimização paramétrica do chassi de um caminhão através do recém adicionado "*optimization module*" ao *software Ansys*[®] com o objetivo de redução de massa. Para isso, a estrutura é modelada como pórticos tridimensionais de aço. Ramamurthy (1986) realiza uma otimização paramétrica de estruturas do tipo pórticos tridimensionais e desenvolve um algoritmo para otimiza-las que inclui tanto análise linear como não-linear. A função objetivo considerada foi igualmente a redução de peso. Lógó (1988) mostra uma otimização multicritérios para pórticos reforçados de concreto, com aplicação voltada à construção civil. Rao (1987) e Juang (1989) trazem em seus trabalhos abordagens interessantes de otimização de pórticos metálicos sujeitos não apenas a carregamentos estáticos, mas também dinâmicos como a força de arrasto do vento e as solicitações devido a abalos sísmicos.

Na década de 1990, houve uma grande disseminação de métodos de otimização em vários âmbitos da engenharia. Destacam-se os trabalhos referentes à otimização estrutural neste período de importante contribuição trazida pelo livro de Steven (1997) onde ele desenvolve o método ESO (*Evolutionary Structural Optimization*). Tomasz Arciszewski (1991) desenvolveu uma abordagem no âmbito de otimizações estruturais "baseada no caso", uma estratégia evolutiva que pode ser considerada predecessora ao "*Teaching-learning-Based Optimization*". Jenkins (1992) estudou um caso de projeto ótimo de pórticos planares através da técnica de Algoritmos Genéticos. Petruska (1992) desenvolveu técnicas de otimização paramétrica para estruturas de aço considerando esforços estáticos e dinâmicos. Vanderplaats (1999) discutiu o uso de técnicas de otimização para diferentes projetos de estruturas, mostrando o progresso do estado da arte no assunto anterior a ele, exemplos que demonstram suas técnicas e projeções futuras na área.

Já então no século XXI, pode-se citar uma grande variedade de obras. Zhang (2008), através de uma modelagem não-linear via elementos finitos, realiza uma otimização paramétrica em molas feitas de borracha para serem usadas na suspensão de veículos. Guang (2005) traz uma otimização simultânea - paramétrica e de forma, para treliças considerando condições não-lineares de esforços dinâmicos via uma variação de Algoritmos Genéticos. Mantena (2007) estuda otimização para-

métrica de painéis do tipo "*grid-stiffened*" de material compósito com objetivo de maximizar sua rigidez.

É importante destacar também que, entre as décadas de 60 e 70, um importante trabalho na área de algoritmos de otimização é criado por John Holland, no qual ele desenvolve o chamado Algoritmo Genético (AG). Esse algoritmo foi baseado no fenômeno da adaptação natural, estudado mais de cem anos anos pelo renomado naturalista inglês Charles Darwin. De fato, Holland tinha por objetivo o desenvolvimento de um algoritmo robusto para estudar formalmente os fenômenos de adaptação que ocorrem na natureza, quiçá até prevê-la. Embora não tão exitoso nessa tarefa, o algoritmo foi feito, e logo percebeu-se sua aplicabilidade e adaptabilidade em diversos campos da engenharia, inclusive na estrutural. Sua obra, *Adaptation in Natural and Artificial Systems*, datada de 1970, traz um importante progresso nos chamados algoritmos meta-heurísticos, (Colherinhas, 2016).

Foi inspirado nesse algoritmo que, em 2011, Rao, Savsani e Vakharia (2011) desenvolveram o chamado algoritmo TLBO - *Teach-Learning Based Optimization*; ou numa livre tradução, "Otimização Baseada no Ensino - Aprendizagem". Como sugerido, esse algoritmo tem como ideia central tanto a relação entre aluno e professor como entre alunos como forma de se obter conhecimento. Já em 2014, Farshchin (2014) propôs singelas porém eficazes mudanças no algoritmo de Rao *et al* (2011) e, sob a alcunha de "TLBO Modificado", publicou um artigo no qual descreve seu algoritmo e mostra as modificações feitas em relação ao original - que são basicamente o uso de uma média ponderada para calcular a *fitness* (capacidade, em uma livre tradução) do conhecimento de uma turma. Maiores explicações serão fornecidas posteriormente no capítulo 3.3, destinado a esse assunto.

Finalmente, nos últimos anos, pode-se destacar alguns artigos relevantes produzidos na área. Patel (2013) utiliza o Método de Taguchi, no qual segundo o autor é um conjunto de técnicas matemáticas e estatísticas muito úteis à otimizações paramétricas, para otimizar o chassi de um caminhão modelo Taguchi Eicher 11.10. Rao (2013) realiza uma otimização paramétrica utilizando o método *Teach-Learning Based Optimization*, como se falará em seguida e do qual ele próprio propôs alguns anos antes, referentes à operação de três máquinas, sendo duas mais convencionais (esmerilhadeira e fresadora) e outra nem tanto (máquina de jateamento de água abrasivo). Destaca-se deste trabalho as conclusões dos autores quanto ao algoritmo, citando sua aplicabilidade à problemas reais e sua efetividade, principalmente quando comparados a outros algoritmos conhecidos, como PSO (*Particle Swarm Optimization*), HS (*Harmony Search*) ou ABC (*Artificial Bee Colony*). Pezeshk (2018) realiza a otimização do projeto de pórticos metálicos considerando a minimização de peso das estruturas. Para isso, os autores aplicaram uma generalização do TLBO, o "*School Based Optimization*". O autor utiliza três casos de *benchmark* para demonstrar o algoritmo e também compara-lo com outros em relação ao resultado final, número de análises, etc. Outro ponto

interessante do trabalho que merece ser destacado é que as variáveis de projeto - secção de área transversal, são discretas e baseadas em normas de padronização de perfil, dessa forma o *output* do algoritmo é o perfil a ser utilizado, e não um valor específico.

3 Fundamentação Teórica

3.1 Método dos Elementos Finitos

Os conceitos e a abordagem do Método dos Elementos Finitos - MEF, remetem à década de 40, ainda na segunda guerra, onde originalmente foram usados para o projeto de aeronaves. Porém o termo "*finite elements*" só foi usado em 1960, (Ray, 2012). Devido a revolução tecnológica experimentada no final do século XX e, principalmente da incrível e veloz disseminação computacional, foi possível que o método se tornasse robusto e aplicável em diversas áreas da engenharia, como análise de tensões e deformações, dinâmica de fluidos, transferência de calor, entre outras. Atualmente, problemas de modelagem complexa e com soluções analíticas complicadas têm no MEF uma solução viável.

Esse tema é de tal importância que um trabalho dedicado exclusivamente a ele demandaria muito esforço e várias páginas, o que não é o objetivo do presente trabalho. Aqui deseja-se mostrar apenas uma pequena porção, que é a relação entre cargas externas aplicadas a um corpo deformável e as tensões e deformações gerados nele. Mais especificamente, esse corpo deformável é uma estrutura reticulada do tipo pórtico. Uma estrutura dita reticulada pode ser descrita como aquela composta por barras esbeltas ligadas em nós através de algum tipo de junta. Uma composta por elementos ligados através de nós, (Martha, 2010). Para o caso de pórticos, considera-se esforços (forças e momentos) nos três graus de liberdade lineares e nos três angulares, para cada nó. A Figura (2) traz uma representação de um pórtico tridimensional genérico, no entanto essas estruturas podem ser corriqueiramente encontradas em equipamentos e componentes mecânicos, torres e estruturas metálicas, pontes, construção civil, etc.

A metodologia de abordagem do MEF é padrão e consiste em, primeiramente, discretizar um corpo contínuo em um certo número de elementos. A partir desse ponto, desenvolve-se a formulação para cada elemento de forma isolada, como se ele não fizesse parte do conjunto, sendo essa etapa denominada de **formulação local**. Encontrado o equacionamento que rege o comportamento de um elemento, então se analisa a interação de cada destes com o conjunto, chamando essa etapa de **formulação global**, portanto. Essa segunda parte nada mais é do que uma transformação algébrica entre dois sistemas de coordenadas. A formulação que, inicialmente estava escrita em relação a um sistema de coordenada do próprio elemento, deve ser escrita em relação a um sistema de coordenadas global, ou seja única para todos os elementos, para que assim se possa efetivamente somar as contribuições de cada elemento para a estrutura. Essa última parte é de suma importância à metodologia MEF. A montagem de uma formulação global a partir de cada contribuição local é uma interpretação do problema físico, e qualquer equívoco nesse ponto pode levar a resultados incorretos.

De forma direta, será aqui apresentado, para pórticos tridimensionais, inicialmente a formu-

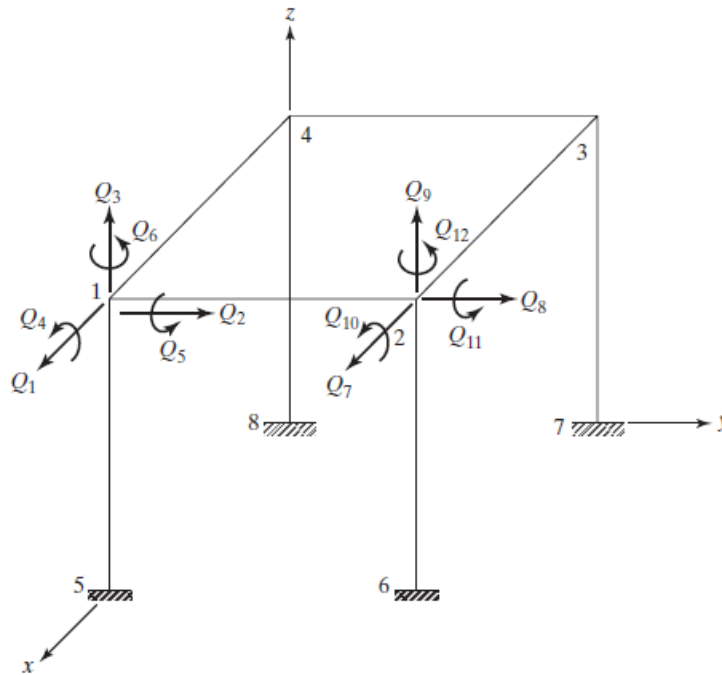


Figura 2: Exemplo de um pórtico tridimensional.
Fonte: Ray (2012).

lação local e posteriormente expandida para a global.

3.1.1 Formulação Local do Elemento de Pórtico

É sabido, através da mecânica dos sólidos, que o comportamento de um corpo deformável sob ação de esforços externos é regido por equações diferenciais que muitas vezes se tornam demasiadas complexas. O grande *insight* do MEF, além da divisão de um domínio contínuo em uma quantidade finita de elementos, é conseguir escrever essas equações diferenciais como um conjunto de equações algébricas - e costumeiramente lineares, que têm uma solução simples. De forma geral, o que se deseja ao utilizar o método é encontrar os deslocamentos dos nós, pois a partir daí encontram-se deformações e tensões. Para isso, busca-se a solução da seguinte equação matricial

$$k^e \cdot q^e = f^e \quad (1)$$

onde q^e é o vetor deslocamento do elemento, f^e o vetor força do elemento e k^e a matriz de rigidez do elemento. Antes de descrever essas matrizes, é preciso deixar claro que um elemento de pórtico é uma super-posição de elementos de barra - que suportam carregamentos axiais ou longitudinais ao eixo do elemento; e elementos de viga - que suportam carregamentos transversais.

Além disso, há os carregamentos axiais angulares, que causam um momento torsor. A Figura (3) ilustra essa super-posição para o caso bidimensional.

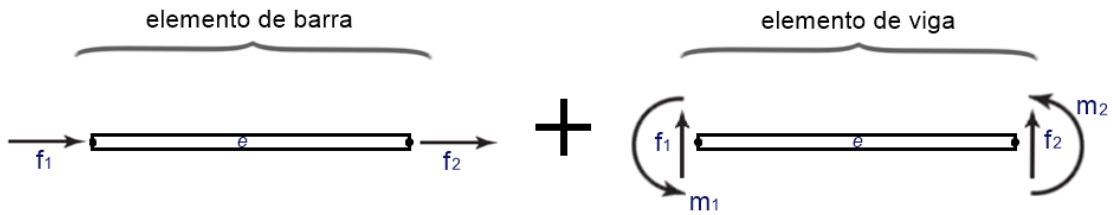


Figura 3: Composição de um elemento de pórtico bidimensional a partir dos elementos de barra e de viga.

Fonte: autor.

• Vetor Deslocamento q^e do Elemento

O vetor deslocamento para a formulação local do elemento de pórtico tridimensional é uma matriz coluna de dimensão 12 e tem, portanto, a seguinte forma:

$$q^e = [\overbrace{q_1, q_2, q_3}^{\text{Lineares do nó 1}}, \overbrace{q_4, q_5, q_6}^{\text{Angulares do nó 1}}, \overbrace{q_7, q_8, q_9}^{\text{Lineares do nó 2}}, \overbrace{q_{10}, q_{11}, q_{12}}^{\text{Angulares do nó 2}}]^T \quad (2)$$

onde q_i é o deslocamento no grau de liberdade i . Para este caso e todos os outros que seguirão, os deslocamentos lineares são, respectivamente, em x , y e z ; e os angulares são respectivamente em torno dos eixos x , y e z .

• Vetor de Forças f^e do Elemento

Analogamente, o vetor força para a formulação local também é uma matriz coluna de dimensão 12 e tem a mesma forma do vetor deslocamento. Assim como anteriormente, a ordem segue, respectivamente, forças lineares em x , y e z ; e momentos em torno dos eixos x , y e z .

$$f^e = [\overbrace{f_1, f_2, f_3}^{\text{Lineares do nó 1}}, \overbrace{f_4, f_5, f_6}^{\text{Momentos do nó 1}}, \overbrace{f_7, f_8, f_9}^{\text{Lineares do nó 2}}, \overbrace{f_{10}, f_{11}, f_{12}}^{\text{Momentos do nó 2}}]^T \quad (3)$$

• Matriz de Rigidez k^e do Elemento

A matriz de rigidez do elemento, portanto, deverá ter dimensão 12×12 . É através dela que todas as equações diferenciais são postas de forma linear, e é através dela que se atribui características geométricas e de material ao elemento. Essa é escrita conforme a formulação a seguir. Nela, E é o módulo de elasticidade longitudinal, G é o módulo de elasticidade transversal, J é o momento

polar de inércia, I_i é o momento de inércia em relação ao eixo i e A é secção de área transversal. Já l_e é o comprimento do elemento e é calculado a partir das coordenadas dos nós 1 - (x_1, y_1, z_1) , e 2 - (x_2, y_2, z_2) , sendo $l_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. Como já mencionado antes, a montagem da matriz de rigidez global não se resume a um simples processo de transformação de coordenadas; mas também uma leitura física e geométrica do problema. Certa contribuição local já devidamente rotacionada, se não alocada em seu devido local onde o problema físico impõe, pode ser crítica em alguns projetos. Uma má alocação implica em uma outra distribuição geométrica e, portanto, outro problema físico. A matriz de Rigidez Local é, segundo Ferreira (2009), escrita matematicamente como:

$$k_i^e = \begin{bmatrix} AS & 0 & 0 & 0 & 0 & 0 & -AS & 0 & 0 & 0 & 0 & 0 \\ & a_{z'} & 0 & 0 & 0 & b_{z'} & 0 & -a_{z'} & 0 & 0 & 0 & b_{z'} \\ & & a_{y'} & 0 & -b_{y'} & 0 & 0 & 0 & -a_{y'} & 0 & -b_{y'} & 0 \\ & & & TS & 0 & 0 & 0 & 0 & 0 & -TS & 0 & 0 \\ & & & & c_{y'} & 0 & 0 & 0 & b_{y'} & 0 & d_{y'} & 0 \\ & & & & & c_{z'} & 0 & -b_{z'} & 0 & 0 & 0 & d_{z'} \\ & & \vdots & & & & AS & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & a_{z'} & 0 & 0 & 0 & -b_{z'} \\ \dots & & \text{sim.} & & \dots & & & & c_{y'} & 0 & b_{y'} & 0 \\ & & & & & & & & & TS & 0 & 0 \\ & & & & & & & & & & c_{y'} & 0 \\ & & & & & & & & & & & c_{z'} \end{bmatrix}$$

onde:

$$\begin{aligned} AS &= \frac{1}{l_e} E \cdot A & ; & & TS &= \frac{1}{l_e} G \cdot J \\ a_{z'} &= \frac{12}{l_e^3} E \cdot I_z & ; & b_{z'} &= \frac{6}{l_e^3} E \cdot I_z & ; & c_{z'} &= \frac{4}{l_e} E \cdot I_z & ; & d_{z'} &= \frac{2}{l_e} E \cdot I_z & (4) \\ a_{y'} &= \frac{12}{l_e^3} E \cdot I_y & ; & b_{y'} &= \frac{6}{l_e^3} E \cdot I_y & ; & c_{y'} &= \frac{4}{l_e} E \cdot I_y & ; & d_{y'} &= \frac{2}{l_e} E \cdot I_y \end{aligned}$$

3.1.2 Formulação Global do Elemento de Pórtico

A transformação da formulação local para a global nada mais é do que um processo algébrico de mudança de base. Para isso, portanto, deve-se encontrar uma matriz, chamada Matriz de Transformação L , que quando multiplicada pelo vetor em sua formulação local, resulta no vetor em sua formulação global. Matematicamente o processo funciona da seguinte forma

$$Q^e = L \cdot q^e \quad ; \quad F^e = L \cdot f^e \quad (5)$$

Pode-se então reescrever a Equação (1). Esta, reescrita em sua formulação global, tem a seguinte forma

$$K^e \cdot Q^e = F^e \quad , \text{ onde } K^e = L^T \cdot k^e \cdot L \quad (6)$$

Essa Matriz de Transformação L , para o caso de pórticos tridimensionais, é, conforme Ferreira (2009):

$$L_{(12 \times 12)} = \begin{bmatrix} \lambda_{(3 \times 3)} & & & 0 \\ & \lambda_{(3 \times 3)} & & \\ & & \lambda_{(3 \times 3)} & \\ 0 & & & \lambda_{(3 \times 3)} \end{bmatrix} \quad , \text{ onde } \lambda_{(3 \times 3)} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \quad (7)$$

A matriz λ é chamada Matriz de Cossenos Diretores. Sua primeira linha relaciona o eixo x da coordenada local com os eixos x , y e z globais, através da primeira, da segunda e da terceira coluna respectivamente. Dessa forma, os elementos l_1 , m_1 e n_1 são escritos como:

$$l_1 = \frac{1}{l_e} (x_2 - x_1) \quad ; \quad l_2 = \frac{1}{l_e} (y_2 - y_1) \quad ; \quad l_3 = \frac{1}{l_e} (z_2 - z_1) \quad (8)$$

As demais linhas serão escritas em função desses elementos da primeira linha e também do parâmetro D .

$$\begin{aligned} D &= \sqrt{l_1^2 + m_1^2} \quad ; \\ l_2 &= -\frac{1}{D} m_1 \quad ; \quad m_2 = \frac{1}{D} l_1 \quad ; \quad n_2 = 0 \\ l_3 &= -\frac{1}{D} l_1 \cdot n_1 \quad ; \quad m_3 = -\frac{1}{D} m_1 \cdot n_1 \quad ; \quad n_3 = D \end{aligned} \quad (9)$$

Dessa forma, os eixos y e z locais também são relacionados com X , Y e Z globais. Há ainda um caso específico que essa formulação apresentada não é válida, que é o caso em que $x_1 = x_2$ e $y_1 = y_2$, ou seja, elemento coincidente com o eixo z . Para esse caso, a matriz λ é

$$\text{se } z_2 > z_1, \text{ então } \lambda = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\text{senão, } \lambda = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (10)$$

Portanto, o vetor deslocamento global é encontrado através da solução do sistema linear envolvendo o vetor força global e a matriz de rigidez global.

$$[K^e] \cdot \{Q^e\} = \{F^e\} \quad (11)$$

3.2 Otimização Estrutural

A partir desse momento, onde o conceito intuitivo de otimização já foi devidamente abordado, é interessante passar a lidar com problemas de otimização do ponto de vista matemático. Nessa nova abordagem, partindo do problema mais genérico possível, pode-se definir o problema de otimização, de acordo com Yang (2010), como:

$$\begin{aligned} &\text{minimize } \rightarrow f(x) ; \\ &\text{sujeito à:} \\ &g_j(x) \leq 0, \quad j = 1, 2, \dots, p \\ &h_k(x) = 0, \quad k = 1, 2, \dots, p \\ &x_i^L \leq x_i \leq x_i^H \end{aligned} \quad (12)$$

onde x é o vetor de projeto, formado pelas de variáveis de projeto; $f(x)$ é a função objetivo; $g(x)$ e $h(x)$ são os vetores com as restrições de desigualdade e igualdade, respectivamente; e x^L e x^H os limites - inferiores e superiores, respectivamente. Estes três (variáveis de projeto, função objetivo e restrições) formam, os Elementos de Formulação do problema de otimização, (Gürdal, 2012).

- O vetor de projeto $x = x_1, x_2, \dots, x_n$, é composto por n parâmetros de projeto. Esse vetor serve como dados de entrada da função objetivo, que por sua vez retorna um determinado valor. No problema de otimização, o que se busca é justamente o vetor de projeto que retorna o menor valor da função objetivo possível. Costumeiramente em otimização estrutural o vetor de projeto carrega consigo características geométricas (dimensões) e físicas do material;
- A função objetivo $f(x)$, conforme visto, é aquela que carrega o problema do otimização em si, descritos em termos matemáticos. Ela é escrita necessariamente em função do(s) parâmetro(s) contido(s) no vetor de projeto. É sobre esta função que aplicam-se todas as

técnicas numerico-matemáticas a fim de minimizá-la. Em otimização estrutural, a função objetivo pode ser peso, rigidez ou flexibilidade, custo, etc;

- Já as restrições, conforme visto podem ser de igualdade ou desigualdade. Restrições de igualdade geralmente estão vinculadas às equações de equilíbrio, em problemas estáticos; enquanto de desigualdade impõem limites em quantidades. Há ainda restrições que impõem limites diretamente na(s) variável(is) de projeto e são por natureza mais simples que as demais. Essas recebem o nome de *side constraints* ("restrições de lado", numa tradução literal);

3.2.1 Tipos de Otimização

Não há um consenso ou uma classificação universal para otimizações estruturais da qual todos seguem fielmente. Na verdade, cada autor tem a liberdade de fazê-la conforme entender, podendo ter uma abordagem mais didática ou mais prática. No presente trabalho, adotar-se-á uma classificação relativamente difundida, que foi adotada por Sigmund (2013) e divide em três as formas de se otimizar estruturas:

- **Paramétrica ou Dimensional** - Neste tipo, dada uma configuração inicial ao problema, varia-se apenas propriedades construtivas, tanto geométricas (secção de área, momento de inércia, espessura, largura, ...) ou do material (tensão de escoamento, módulo de elasticidade, ...). Não há, portanto, nenhuma alteração na forma ou topologia da estrutura.
- **de Forma** - Mantendo-se a topologia fixa, este tipo altera a forma da estrutura, ou seja, seus contornos. Há, portanto, modificações na fronteira entre material e vazios, de forma a mudar posições e contornos de furos, por exemplo, porém sem criar novos.
- **Topológica** - A mais abrangente entre todas, altera-se a topologia da estrutura. Em outras palavras, ao ser determinado o domínio (espaço), encontra-se a melhor distribuição de material possível para o problema. Faz-se isso analisando quais regiões da estrutura são menos solicitadas e remove-as, de acordo com alguns parâmetros pré-determinados.

3.2.2 Métodos de Otimização

Ao se falar em métodos de otimização, refere-se basicamente ao algoritmo responsável por de fato fazer a busca pelo ótimo. Para um mesmo tipo de otimização, com a mesma função objetivo, variáveis de projeto e restrições, é possível que se utilize ainda diferentes métodos de otimização. Yang (2010) dedica uma seção de seu livro à classificação desses métodos, e os divide principalmente em dois grupos: métodos determinísticos e estocásticos.

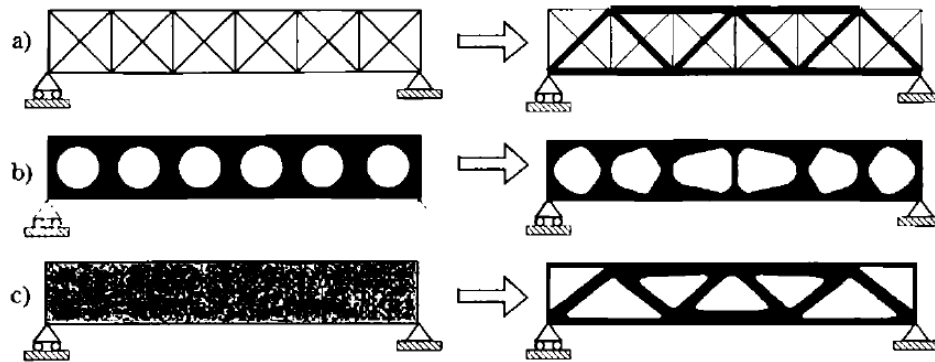


Figura 4: Representação das Otimizações Paramétrica (a), de Forma (b), e Topológica (c).
Fonte: Sigmund (2013)

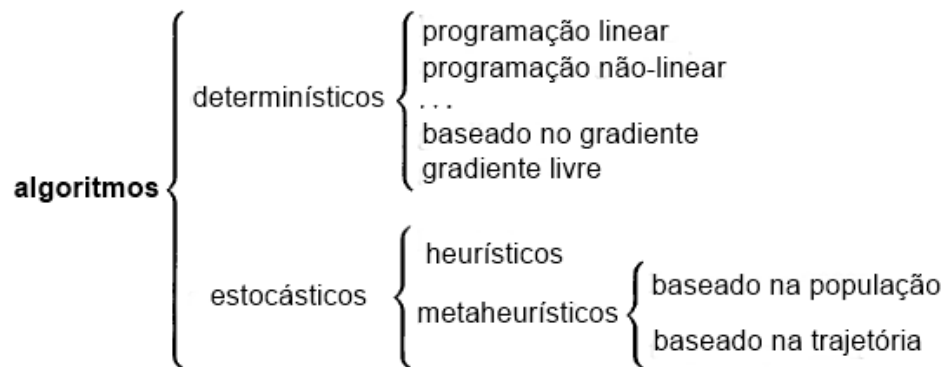


Figura 5: Classificação dos principais Métodos de Otimização.
Fonte: Yang (2010). Traduzido pelo Autor

Métodos determinísticos seguem procedimentos rigorosos baseados em teoremas matemáticos de forma a garantir sua convergência a uma solução ótima. Também chamados de métodos clássicos, englobam métodos de programação linear (como o Simplex) e não-linear, do gradiente, entre outros. Como desvantagens, apresentam a necessidade de ter uma função objetivo e sua(s) derivada(s) contínua(s) no domínio, assim como a dependência do ponto de partida e um mau desempenho em funções multiobjetivos, (Colherinhas, 2016).

Já os métodos estocásticos, ou probabilísticos, segundo Silva (2018b), "possuem seu embasamento em análises probabilísticas das soluções viáveis do problema". Estes introduzem parâmetros gerados aleatoriamente para avaliação da função objetivo, e seguindo por técnicas puramente lógicas conseguem chegar ao seu objetivo, (Colherinhas, 2016). Em geral, métodos estocásticos se subdividem em heurísticos e metaheurísticos. Heurístico refere-se a um conhecimento que não pode ser provado matematicamente. Para esses métodos, soluções ótimas para um problema complexo podem ser encontradas em um período de tempo razoável, mas não há garantia de que assim

sempre será. Além disso, eles são recomendáveis quando se deseja boas soluções para o problema, e não necessariamente a melhor.

O desenvolvimento de métodos heurísticos levam aos metaheurísticos (*meta* = "além de"). Estes são algoritmos inspirados basicamente em processos lógicos e/ou naturais. De alguma forma e em algum nível de complexidade, eles de fato tentam simular fenômenos da natureza. Por exemplo, Algoritmos Genéticos, como já foi citado, baseiam-se na seleção natural. Otimização por colônia de formigas (*Ant Colony Optimization* ou *ACO*) baseia-se no comportamento de formigas ao traçarem seus caminhos. Redes Neurais Artificiais baseia-se no funcionamento de neurônios e do sistema nervoso central; entre diversos outros.

Pezeshk (2018) destacam que a abordagem comum entre métodos metaheurísticos é a geração aleatória da população inicial de possíveis soluções e a partir dela própria melhorar gradativamente a performance em um processo sistemático. Em otimização estrutural, devido à complexidade do espaço de busca e das restrições de projeto, aplicações de métodos metaheurísticos são preferíveis aos baseado no método gradiente, (Farshchin, 2014).

3.3 Algoritmo TLBO Modificado

O *Modified Teaching-Learning Based Optimization*, ou simplesmente TLBO Modificado é um algoritmo de otimização metaheurístico que, como já mencionado no Capítulo 2, foi proposto inicialmente em 2014 - sendo, portanto, relativamente recente; por Charles V. Camp e Mohammad Farshchin. Embora as modificações realizadas pelos autores tenham sido de grande valia, o real mérito pelo desenvolvimento do algoritmo é de Venkata Rao, Vimal Savsani e Dipakkumar Vakharia; que em 2011 publicaram o artigo intitulado "*Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems*".

Nesse trabalho, os autores apresentam o novo algoritmo desenvolvido, explicando-o tanto qualitativa quanto quantitativamente. Eles citam que a principal motivação para desenvolverem um algoritmo baseado na natureza é sua capacidade de resolver diversos problemas de otimização de forma eficiente e eficaz, (Vakharia, 2011). A ideia de seu funcionamento é modelada a partir das relações de aprendizado entre alunos e professores, de modo que o algoritmo tenta simular como o conhecimento é gerado. O objetivo primordial do TLBO Modificado é melhorar, em média, a performance (desempenho) dos indivíduos duma população - classe. Ele parte da premissa de que a distribuição de notas em uma turma segue uma distribuição normal. Primeiramente, assim como é costumeiro em algoritmos metaheurísticos, é gerada uma população inicial. A partir desse ponto, inicia-se um processo iterativo que consiste em duas partes principais: a fase professor e a fase aluno, as quais serão detalhadas a seguir.

3.3.1 Fase Professor

Nesta fase, é simulado a influência de um professor sobre o conhecimento adquirido pelo aluno. entende-se por professor o aluno mais apto da população. Conforme a Figura (6), um professor mais apto (T_B) tende a elevar a classe como um todo, medida através da média (M_B). Para isso, a Fase Professor deve ser aplicada em toda a faixa de variáveis de projeto, de forma a todos os indivíduos da população sofrerem influencia do professor.

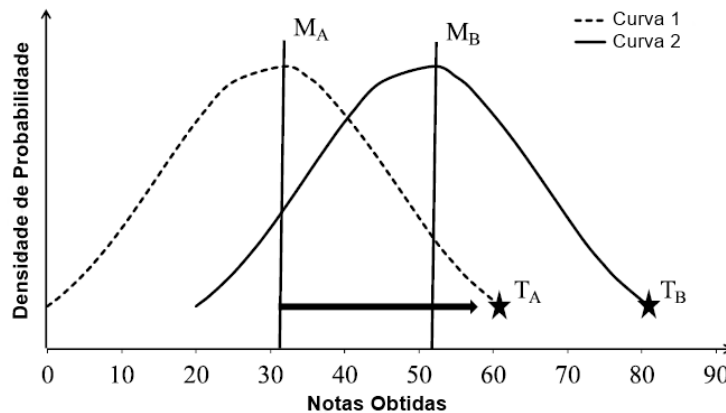


Figura 6: Distribuição de notas em uma turma para dois professores diferentes.
Fonte: Farshchin (2014). Traduzido pelo Autor

Matematicamente falando, o aprendizado dos alunos durante a Fase Professor é definido por:

$$\begin{aligned} x_{new}^k(j) &= x_{old}^k(j) \pm \Delta(j) \\ \Delta(j) &= T_F \cdot r \cdot |M_{(j)} - T_{(j)}| \end{aligned} \quad (13)$$

onde $x^k(j)$ denota a j -ésima variável de projeto do k -ésimo vetor de projeto, T_F é o fator de ensino, r é um número aleatório distribuído uniforme entre $[0, 1]$, $M_{(j)}$ é a média da classe e $T_{(j)}$ o estado do professor. Nota-se que $\Delta(j)$ indica a diferença entre o professor e a média da classe. O fator de ensino T_F é um parâmetro de entrada do algoritmo, e embora seja escolhido arbitrariamente, Vakharia (2011) indicam que um valor de $T_F = 2$ é satisfatório para estruturas reticuladas. Já o parâmetro $M_{(j)}$, que na formulação original de Vakharia (2011) é calculado a partir de uma média aritmética simples, aqui no algoritmo modificado recebe uma ponderação com base no valor da *fitness* do indivíduo, onde *fitness* pode ser entendido como aptidão ou capacidade. O parâmetro é definido matematicamente como:

$$M_{(j)} = \frac{\sum_{k=1}^N (X^k_{(j)}) / (F^k)}{\sum_{k=1}^N 1 / (F^k)} \quad (14)$$

onde F^k é a *fitness* (ou aptidão) do k -ésimo aluno.

3.3.2 Fase Aluno

Outro mecanismo de aprendizado considerado pelo algoritmo, denominado Fase Aluno, advém da interação entre alunos, no qual alunos individualmente aumentam seu conhecimento e melhoram não só sua performance como a da classe em geral. O procedimento nesta fase segue os seguintes passos:

1. Selecionar aleatoriamente da classe o aluno p ;
2. Selecionar aleatoriamente outro aluno q , tal que q seja diferente de p ;
3. Avaliar a *fitness* de cada estudante;
4. Se $F^p < F^q$ (indivíduo p melhor que q), então

$$X_{new}^p(j) = X_{old}^p(j) + r.[X_{old}^p(j) - X^q(j)] \quad (15)$$

se não,

$$X_{new}^p(j) = X_{old}^p(j) + r.[X^q(j) - X_{old}^p(j)] \quad (16)$$

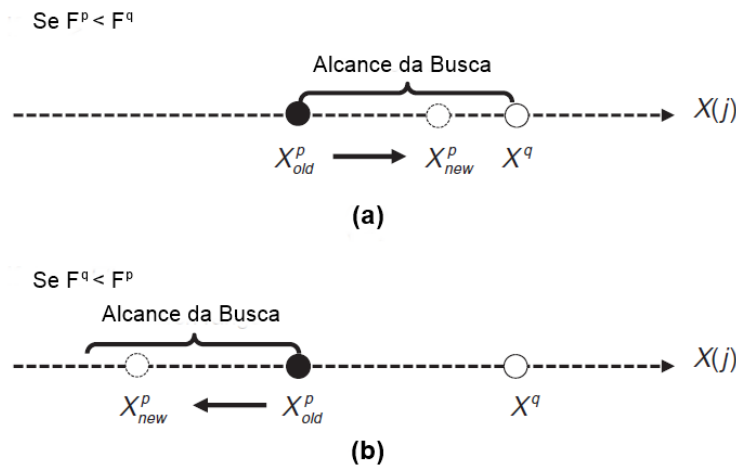


Figura 7: Ilustração da Fase Aluno, quando (a) aluno q é melhor que p e (b) aluno p melhor que q
 Fonte: Farshchin (2014). Traduzido pelo Autor

Novamente, nas equações (15) e (16), r é um número aleatoriamente distribuído entre $[0, 1]$. A ideia de seus autores foi desenvolver um algoritmo que, embora utilize aleatoriedades e arbitrariedades no processo, não o tornem dependente desses valores. É provado que os autores foram bem sucedidos nesse ponto, pois o principal parâmetro de entrada capaz de alterar sensivelmente o resultado é o tamanho da população. Segundo Farshchin (2014), experimentos numéricos indicam que uma população de 75 alunos é adequada para prover bons resultados.

4 Implementação Computacional

Esta seção dedica-se exclusivamente ao detalhamento do algoritmo utilizado no trabalho. Primeiramente, vale ressaltar que tanto o TLBO quanto TLBO Modificado foram inicialmente desenvolvidos para treliças, e não pórticos. Portanto, para esse trabalho, algumas adaptações pontuais foram feitas, mas sem alterar a essência do algoritmo.

O código foi desenvolvido em *Matlab*[®], conforme já dito, e tem sua linguagem portanto compatível com o *software*. É importante também esclarecer que o algoritmo não foi inteiramente desenvolvido pelo autor, mas foi modificado a partir do trabalho de outros autores, como Rodrigues *et al* (2019).

Para o entendimento do algoritmo de maneira geral, o capítulo está dividido em três partes: a primeira referente a sua organização, a forma na qual ele está estruturado; e a segunda diz respeito ao seu funcionamento. É claro que ambas as partes são complementares e se mesclam em muitos pontos, mas acredita-se que essa seja a forma mais didática para a compreensão. Soma-se a essas duas uma terceira parte, complementar, dedicada a descrever e justificar as modificações realizadas em relação à formulação original para treliças.

4.1 Organização do Algoritmo

O organograma trazido pelo Figura 8 é de grande valia para esta seção. Nele, são apresentados, em letras pretas, uma descrição geral de quais partes o algoritmo é composto e sua função; e em letras azuis quais os arquivos *.m* que de fato a realizam.

Na sequência do capítulo, serão explicados com mais detalhes os principais arquivos que compõe o algoritmo.

I *TLBO-MainFile.m* Arquivo principal que chama, direta ou indiretamente, todos os demais arquivos; é o que deve ser compilado. A organização mostrada na Figura 8 é basicamente a estrutura desse arquivo.

II *DataFile-3DFrame.m* Fornece todos os dados de entrada necessários ao algoritmo. É subdividido em duas partes: a primeira sobre as características físicas e geométricas da estrutura, competências então do MEF; e a segunda com a definição do espaço de busca do(s) vetor(es) de projeto e das restrições necessárias impostas ao problema de otimização.

III *Fkfun.m* A função mais utilizada ao longo de todo o processo. É basicamente um *loop* que percorre por toda a população (classe); e a avalia. Faz isso através de duas outras funções: a de penalidade (*PenaltyFunction3D.m*) e a de aptidão (*FitnessFunc3D*). A penalidade utiliza,

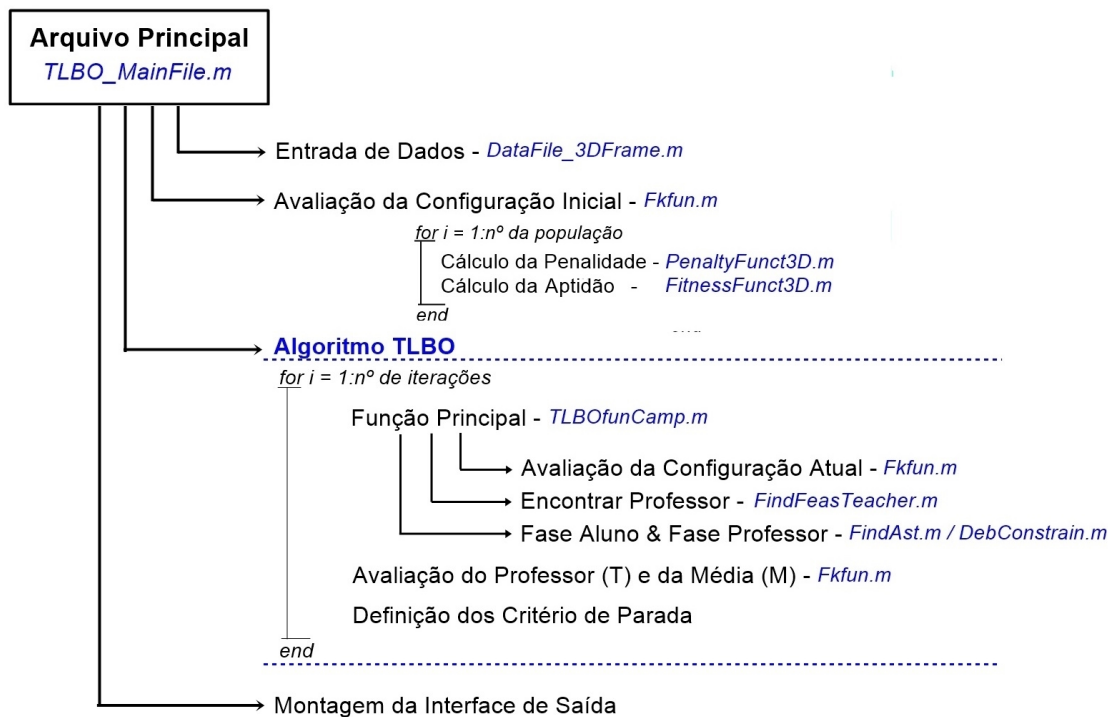


Figura 8: Organograma com a estrutura do algoritmo.

Fonte: autor.

a cada iteração, a análise estrutural através do MEF e, juntamente com as restrições definidas pelos dados de entrada, penaliza aqueles indivíduos que excedem as restrições de deslocamentos e/ou de tensões. Já a função de aptidão calcula de fato a função objetivo para cada indivíduo, que no presente caso é o somatório do produto entre a área de secção transversal, a densidade e o comprimento do elemento. A função *Fkfun.m* portanto, calcula a função objetivo (peso) de cada indivíduo e, quando necessário, aplica uma penalidade que maximiza o peso. Vale ressaltar que a penalidade é calculada como uma generalização daquela detalhada por Camp e Farschin (2014), onde os autores descrevem matematicamente esse processo;

IV *TLBOfunCamp.m* Função que de fato realiza o processo do TLBO. É chamada a cada nova iteração e, para cada uma, encontra o professor da população por meio das funções de aptidão e penalidade (*FindFeasTeacher.m*), e então realiza a fase professor e aluno, tal qual descrito por Camp e Farschin. Com os resultados obtidos por essa função, o próprio arquivo principal aplica os critérios de parada, seja ele não haver variação sensível na função objetivo a um determinado número de iterações, ou se não um número máximos de iterações, definidas pelo usuário.

V *Plot3DFrame.m* Embora não exerça nenhuma influência sobre os resultados do algoritmo, é interessante do ponto de vista da interface de saída que ela apresenta ao usuário, pois pode-

se não apenas conferir a estrutura que foi informada do arquivo de entrada, mas também visualizar como a estrutura se comporta dado o carregamento externo. Ela utiliza apenas os dados referentes ao MEF - através da função *AnaliseMEF.m* para montar suas duas imagens - a estrutura original e a deformada. Ela também utiliza outras "subfunções", como *ncross.m* e *gplot3.m*.

4.2 Funcionamento do Algoritmo

O algoritmo, como qualquer outro, segue uma ordem lógica, desenvolvida com um propósito. Para isso, é necessário que dados de entrada sejam a ele fornecidos. A partir desse ponto, inicia-se uma rotina automatizada e programada e ao fim, não menos importante, é preciso que o algoritmo monte alguma interface de saída para que ele possa informar ao usuário seu(s) resultado(s) obtido(s).

A Figura 9 ilustra essa rotina. Ao iniciar o algoritmo, é gerada uma população inicial com base nos dados de entrada fornecidos e essa população é então avaliada através da *Fkfun.m*. A partir disso entra-se no algoritmo iterativo em si. O primeiro laço (*loop*) é a Fase Professor, onde para toda a população ($k = 1 : n^\circ$ de indivíduos), conforme já descrito anteriormente.

O processo se repete até que toda a população seja influenciada pelo professor. Após esse ponto, entra-se na Fase Aluno, onde basicamente o mesmo processo é repetido, porém agora aplicando a metodologia da Fase Aluno, também já descrita do capítulo anterior.

Os termos F_{new}^k e F_{old}^k referem-se justamente à aptidão do indivíduo (*fitness*), calculado pela *Fkfun.m*. Esse parâmetro nada mais é que o produto entre o valor da função objetivo do indivíduo (peso) e o valor da penalidade total do indivíduo.

4.3 Modificações no Algoritmo

Conforme já dito, tanto a formulação do TLBO quanto o código em si foram desenvolvidos para a otimização de treliças. É sabido que estas são, assim como pórticos, estruturas reticuladas. No entanto, para treliças, há apenas carregamentos e deslocamentos axiais, ou seja, na mesma direção que o eixo dos elementos. Isso implica numa série de ajustes pontuais a serem feitos no código para adaptá-lo ao caso de pórtico.

Logo de início, é preciso adicionar alguns parâmetros de entrada além, que no caso de treliças são desnecessários. É o caso, por exemplo, dos momentos de inércia I_{yy} e I_{zz} ; e dos valores de coordenada máxima, y_{max} e z_{max} . Ambos são necessários para o cálculo da tensão de flexão, $\sigma_{flexão} = (M \cdot c)/I$. Essa tensão, no caso de pórtico, soma-se à tensão normal, $\sigma_{axial} = F/A$, devido ao carregamento axial, que para treliças é a única presente. Essa superposições foram feitas a partir da análise que, uma tensão fletora em torno de um eixo y, por exemplo, é da mesma natureza

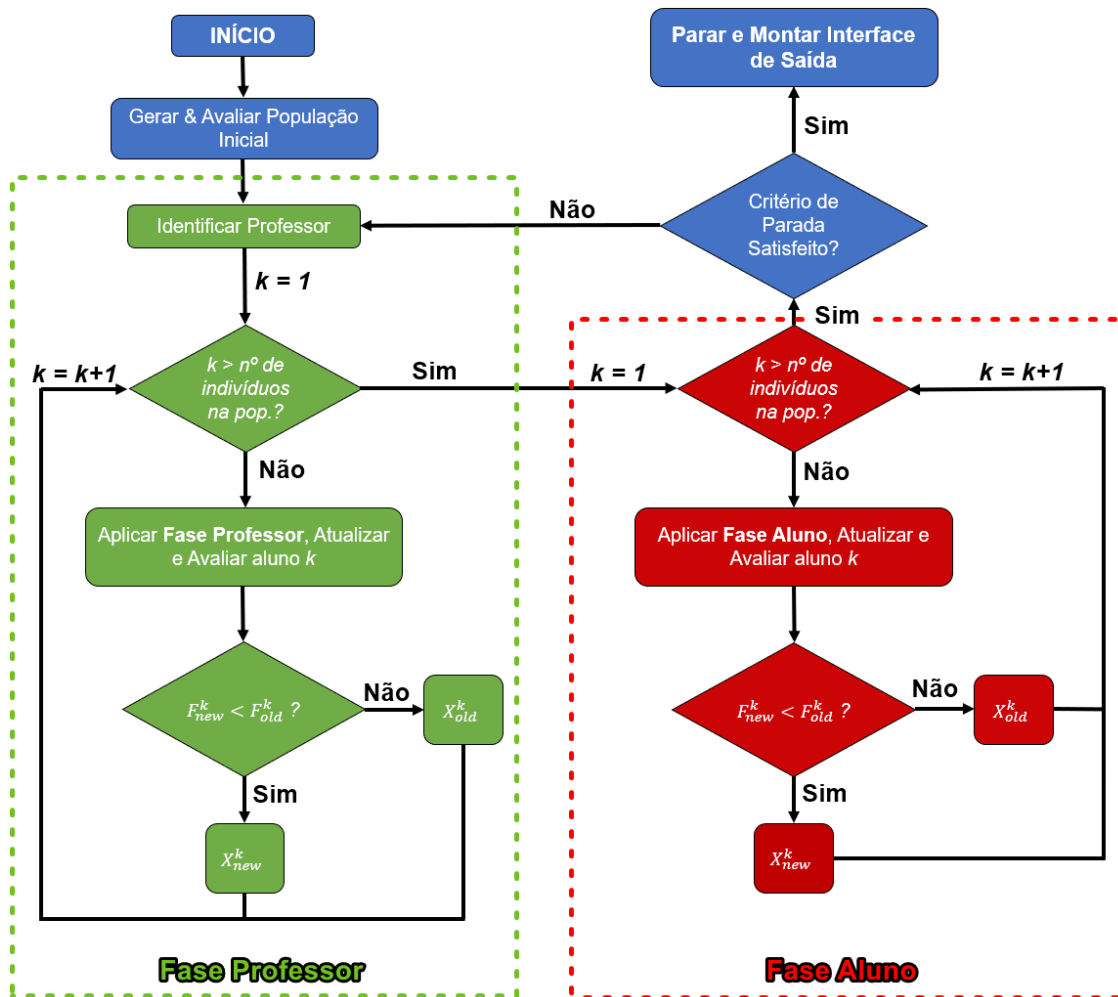


Figura 9: Organograma com a estrutura do algoritmo.

Fonte: autor.

que a tensão normal axial, gerando uma compressão em uma direção e tração na outra, divididas pelo ponto neutro. Para completar, além das características geométricas anteriormente descritas, é adicionado também um valor de Módulo de Elasticidade Transversal G além do Longitudinal E .

Com isso, é de se esperar que a análise por elementos finitos sofra alterações mais sensíveis. Como é sabido, os vetores de deslocamento e de força e a matriz de rigidez tem um formato mais complexo na formulação local de pórticos do que te treliça. A matriz de transformação Local-Global também é escrita de outra forma. Todas as devidas alterações nesse sentido foram feitas, principalmente no arquivo de entrada e no cálculo da penalidade, que é onde a análise por MEF é feita.

Há também as mudanças no TLBO implicadas por essa alteração treliça-pórtico. Originalmente, haviam apenas uma restrição para deslocamentos e outra para tensão, pois só havia um deslocamento - axial, e uma tensão - normal axial. Com a implementação para pórticos, além des-

tes, é preciso impor também uma restrição de deslocamento angular máximo, e outra de tensão de flexão máxima.

No mais, algumas pequenas e quase insignificantes mudanças foram feitas, acrescentando por exemplo o *plot* tridimensional da estrutura - original e deformada. Porém o ponto mais crucial refere-se ao já comentado momento de inércia.

Para o caso de pórticos, é possível que uma mesma área de secção transversal assuma uma infinidade de valores de momento de inércia. E isso pode alterar significativamente o resultado de deslocamentos e tensões geradas. É possível portanto, apenas variando o momento de inércia, alterar a configuração final da solução. Em outras palavras, pode-se por exemplo, maximizar a rigidez da estrutura utilizando como parâmetro de projeto o momento de inércia, e não a área. A forma mais simples de contornar o empecilho é, por exemplo, atribuindo um valor de momento de inércia associado a cada valor de área, ou escrever aquele como uma função deste. Para o presente trabalho foi adotado que todas as secções transversais eram circulares cheias, facilitando assim a atribuição de momentos de inércia, por exemplo, a partir da área. No próximo capítulo, para cada caso - exemplo, será comentado sobre como foi procedido com essa relação de área - momento de inércia.

5 Resultados e Discussões

Com as devidas explicações sobre a fundamentação teórica do tema e do funcionamento do algoritmo, é possível agora mostrar os resultados obtidos pela rotina computacional mostrada. Uma metodologia fundamental quando se trata de algoritmos de otimização, de um modo geral, é a validação dos resultados. Em termos simples, validar significa comparar, confrontar os resultados obtidos com outros já existentes na literatura e, portanto, confiáveis, assim como também em resultados experimentais. Embora seja simples, é parte crucial da implementação, pois confere credibilidade não só ao algoritmo, mas ao trabalho como um todo. Uma vez que, no presente trabalho não foi realizado nenhum experimento e sim apenas a consulta à literatura, será chamado de verificação.

Dito isso, este capítulo é dedicado à discussão de resultados não só de problemas trazidos pelo autor, mas também dos de verificação. Para isso, o capítulo está dividido em três seções, onde a primeira aborda a verificação do código de elementos finitos para pórticos, sem tratar de otimização; o segundo traz da implementação do TLBO para treliças, uma vez que a maioria dos trabalhos na literatura trata desse tipo de estrutura; e o terceiro enfim trás a implementação do código de TLBO para pórticos, com os mesmos exemplos da seção dois, a fim de poder confrontá-los.

5.1 Verificação MEF para Pórticos

Antes de entrar na parte do TLBO propriamente dita, é preciso garantir antes que o código para o cálculo de elementos finitos para pórticos tridimensionais esteja devidamente ajustada. Em outras palavras, é necessário validar o código de elementos finitos para pórticos. Isso será feito confrontando apenas valores de deslocamentos nodais de estruturas presentes na literatura do tema, com resultado portanto também conhecido.

Foram escolhidos dois exemplos de pórticos tridimensionais presentes no livro "*MATLAB Codes for Finite Element Analysis*", de Ferreira (2009), com estruturas relativamente simples. A Figura 10 trás o primeiro caso, uma estrutura de três barras e quatro nós. Vale destacar que embora a figura apresente um carregamento de 15 kN na direção z , esse valor é na verdade de 20 kN, conforme o autor mostra mais adiante no seu código, havendo uma incongruência entre a imagem e a real condição de carregamento. A Tabela 1 compara então os resultados obtidos por Ferreira (2009) e o código utilizado neste trabalho.

Observa-se uma compatibilidade muito grande dos resultados, com erros praticamente desprezíveis, excetuando um resultado que é um valor muito pequeno e provavelmente arredondado com uma precisão menor que o do presente trabalho. O segundo exemplo, uma estrutura de oito nós e 8 barras é trazido pela Figura 11. A Tabela 2 compara os resultados obtidos por Ferreira

(2009) com os do presente trabalho.

Embora esse segundo caso seja minimamente mais complexo que o primeiro, observa-se que os erros foram ainda menores. Embora pudesse ser considerada já satisfatória a verificação do código analisando e comparando apenas os deslocamentos, preferiu-se fazer isso também para reações (de forças e momentos) e tensões. Não pretende-se gastar muito esforço para isso, mas sim apenas demonstrar brevemente que o código é válido nesses pontos também, e por isso será feita apenas para o primeiro caso aqui mostrado - a estrutura de três barras.

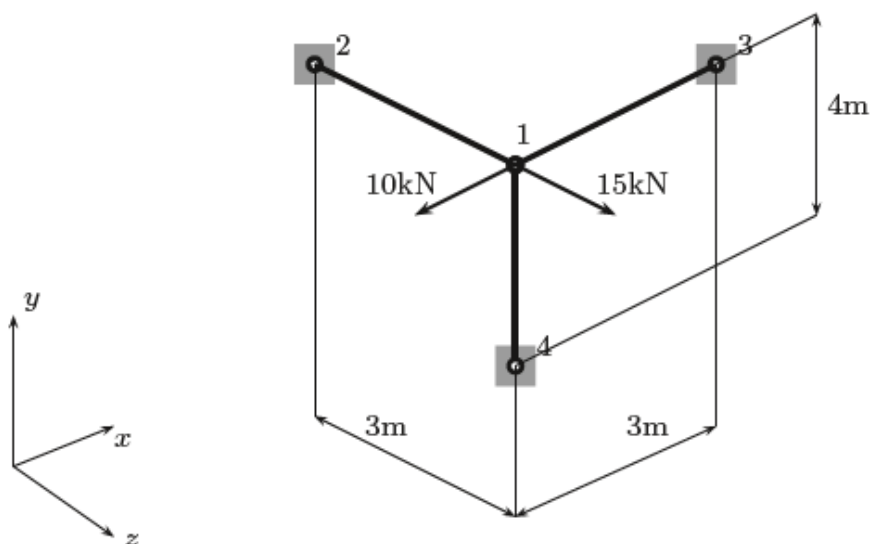


Figura 10: Primeiro caso para verificação do MEF para pórticos.
Fonte: Ferreira (2009).

Verificação MEF Pórtico - Caso 1					
Nó	Direção	#	Ferreira	Presente	Erro [%]
1	x	linear	$-7,05 E - 6$	$-7,05 E - 6$	0,021
	y	linear	$-7,00 E - 8$	$-6,65 E - 8$	4,948
	z	linear	$1,42 E - 5$	$1,42 E - 5$	0,021
	x	angular	$1,45 E - 6$	$1,45 E - 6$	0,153
	y	angular	$1,75 E - 6$	$1,75 E - 6$	0,081
	z	angular	$1,14 E - 6$	$1,14 E - 6$	0,346

Tabela 1: Comparação de resultados obtidos por Ferreira (2009) e o presente trabalho para o caso 1.
Fonte: autor.

Como este não traz resultados para reações ou tensões, seguiu-se a seguinte metodologia: para as reações, realizou-se o somatório de forças e momentos no nó 1 (de aplicação da carga) a fim de verificar que eles são igual a zero (ou igual em módulo às cargas externas, no caso de força). Já para as tensões, o problema foi implementado no *software CREO*, e aí então pode-se comparar os resultados obtidos pelo *software* comercial de análise de elementos finitos com os aqui encontrados.

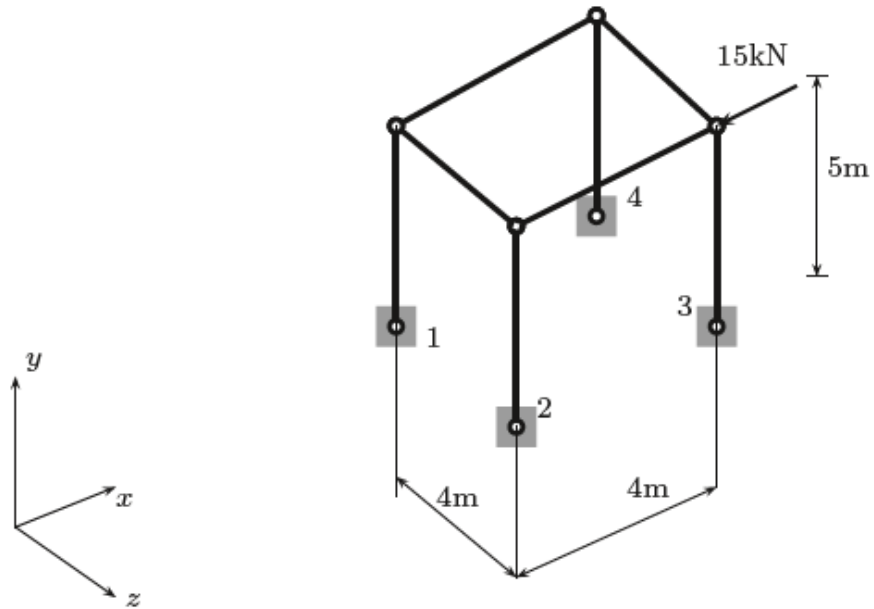


Figura 11: Segundo caso para verificação do MEF para pórticos.
Fonte: Ferreira (2009).

Verificação MEF Pórtico - Caso 2					
nó	direção	#	Ferreira	Presente	Erro [%]
5	x	linear	-0,00227293	$-2,27 E - 3$	0,000
	y	linear	-0,00000393	$-3,93 E - 6$	0,117
	z	linear	0,00115845	$1,16 E - 3$	0,000
	x	angular	0,00007082	$7,08 E - 5$	0,006
	y	angular	0,00013772	$1,38 E - 4$	0,001
	z	angular	0,00023748	$2,37 E - 4$	0,002
6	x	linear	-0,00106188	$-1,06 E - 3$	0,000
	y	linear	-0,00000616	$-6,16 E - 6$	0,021
	z	linear	0,00115835	$1,16 E - 3$	0,000
	x	angular	0,00007201	$7,20 E - 5$	0,005
	y	angular	0,00064608	$6,46 E - 4$	0,001
	z	angular	0,00011324	$1,13 E - 4$	0,002
7	x	linear	-0,00106724	$-1,07 E - 3$	0,000
	y	linear	0,00000581	$5,81 E - 6$	0,058
	z	linear	-0,00116612	$-1,17 E - 3$	0,000
	x	angular	-0,00007274	$-7,27 E - 5$	0,006
	y	angular	0,00075248	$7,52 E - 4$	0,000
	z	angular	0,00011504	$1,15 E - 4$	0,002
8	x	linear	-0,00227804	$-2,28 E - 3$	0,000
	y	linear	0,00000679	$6,79 E - 6$	0,066
	z	linear	-0,00116139	$-1,16 E - 3$	0,000
	x	angular	-0,00007078	$-7,08 E - 5$	0,004
	y	angular	0,00064378	$6,44 E - 4$	0,000
	z	angular	0,00023789	$2,38 E - 4$	0,001

Tabela 2: Comparação de resultados obtidos por Ferreira (2009) e o presente trabalho para o caso 2.

Fonte: autor.

Para a verificação das reações de forças e momentos, é preciso analisar a matriz $[R]$, que tem por dimensões uma linha para cada elemento; e em cada linha as seis reações do primeiro nó seguida das seis do segundo nó, sendo portanto uma matriz (3×12) . Vale ressaltar que é dessa matriz $[R]$ onde se retiram os valores de momentos para o cálculo da tensão de flexão, e para isso ela é escrita em relação a um sistema de coordenadas local. Entretanto, para analisar de forma consistente o somatório das componentes de forças e momentos, é preciso pô-la em relação a um sistema global, se não não haverá sentido em somar suas componentes. A matriz $[R]$, escrita em sua formulação global, tem portanto a seguinte forma:

$$[R]_{(n \text{ elem} \times 2 * n \text{ elem})}^T = [R]_{(12 \times 3)} = \begin{bmatrix} F_{1x} & F_{1x} & F_{1x} \\ F_{1y} & F_{1y} & F_{1y} \\ F_{1z} & F_{1z} & F_{1z} \\ M_{1x} & M_{1x} & M_{1x} \\ M_{1y} & M_{1y} & M_{1y} \\ M_{1z} & M_{1z} & M_{1z} \\ F_{2x} & F_{3x} & F_{4x} \\ F_{2y} & F_{3y} & F_{4y} \\ F_{2z} & F_{3z} & F_{4z} \\ M_{2x} & M_{3x} & M_{4x} \\ M_{2y} & M_{3y} & M_{4y} \\ M_{2z} & M_{3z} & M_{4z} \end{bmatrix}^T \quad (17)$$

Vale ressaltar que a matriz foi mostrada em sua forma transposta por motivo único de facilitar sua formatação no documento. Observa-se, portanto, que as somatórias dos seis graus de liberdade do nó 1 são:

$$\begin{aligned} \Sigma F_x &= \text{soma}(1, n_{col}) & ; & & \Sigma F_y &= \text{soma}(2, n_{col}) & ; & & \Sigma F_z &= \text{soma}(3, n_{col}) \\ \Sigma M_x &= \text{soma}(4, n_{col}) & ; & & \Sigma M_y &= \text{soma}(5, n_{col}) & ; & & \Sigma M_z &= \text{soma}(6, n_{col}) \end{aligned} \quad (18)$$

Essa manipulação, uma vez que a matriz $[R]$ esteja devidamente montada, é relativamente fácil de ser feita. A Figura 12 mostra essas somas e, conforme pode ser visto, apresenta os resultados esperados: para forças, é igual em módulo às cargas externas aplicadas (que não foram contabilizadas) e para momentos é praticamente zero.

Para a verificação das tensões, foi então comparado os resultados obtidos com os do *software* comercial de análise por elementos finitos, *CREO*, e os do presente trabalho. A Figura 13 mostra a

modelagem feita nesse *software* juntamente com os resultados da simulação estática realizada. Os valores de deslocamentos lineares encontrados pelo *CREO* foram muito próximos daqueles trazidos por Ferreira e também dos encontrados no aqui. Mas como esses valores já foram verificados, não há necessidade de mostrá-los novamente.

Tratando então das tensões, o presente trabalho considera duas: a normal axial (que já vem da treliça) e a normal de flexão, calculada como $\sigma_j = \frac{1}{I_j} M \cdot c$, onde j é o elemento em questão, I_j momento de inércia da seção, M o momento fletor e c a coordenada da seção transversal onde deseja-se descobrir a tensão, que no caso máximo, $c = 0,5 \cdot D$, para um o perfil circular cheio. Já no *CREO*, não há a possibilidade de "isolar" apenas estas, então a mostrada é a tensão calculada *Beam Tensile*, que considera os esforços axiais, fletores e também torsões.

A fim de comparar quantitativamente os valores de tensão obtidos pelo *CREO* com os do presente trabalho, a Tabela 3 confronta os resultados obtidos por ambos, com o erro relativo em relação ao valor do *software*. Para constar, a primeira linha da Tabela 3 refere-se à barra orientada no sentido de $-z$, a segunda linha à barra orientada em x , e a terceira à orientada em $-y$. Conforme pode-se observar, os valores são bem similares aos encontrado pelo *CREO*, com um erro relativo um pouco maior na barra orientada segundo $-y$, apenas.

Com isso, acredita-se ficar validado o código de elementos finitos para pórticos tridimensionais a ser utilizado neste trabalho.

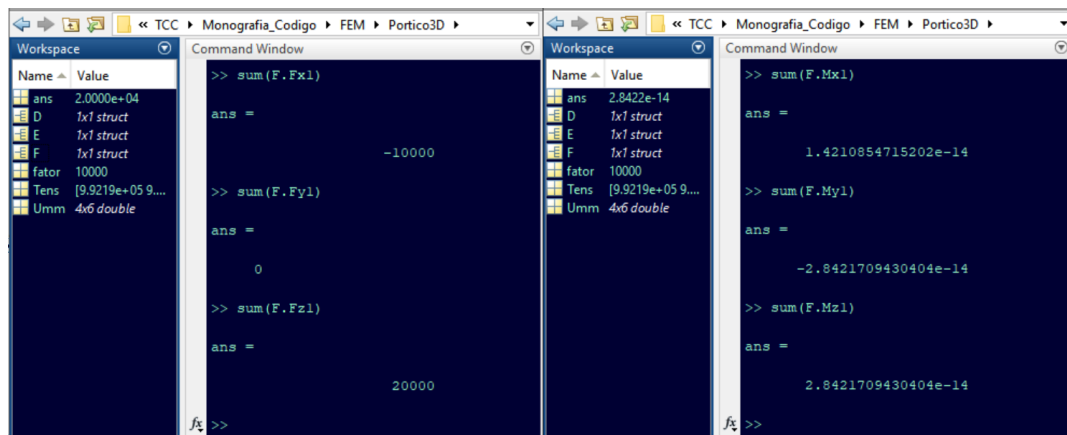


Figura 12: Somatório de forças (esq.) e de momentos (dir.) para nó 1. Fonte: autor.

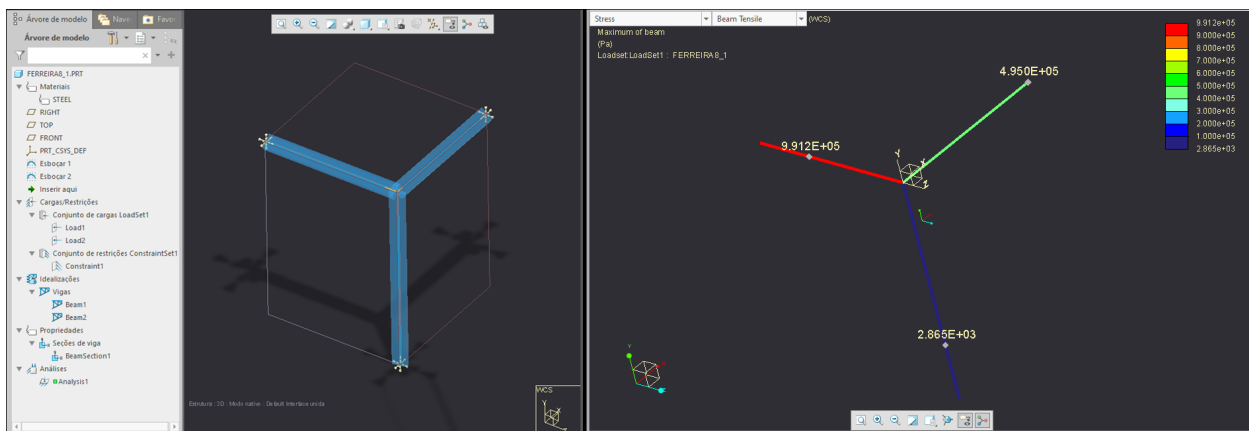


Figura 13: Modelagem da estrutura no *software* *CREO* (esq.) e resultado da simulação estática realizada por este (dir.).

Fonte: autor.

Elemento	Tensão [Pa]		Erro [%]
	<i>CREO Beam Tensile</i>	Tens 1	
1	991212,00	992193,70	0,10
2	495011,00	495343,40	0,07
3	2864,53	3176,44	10,89

Tabela 3: Comparação entre tensões obtidas pelo *CREO* e pelo trabalho para os três elementos da estrutura. Fonte: autor.

5.2 Verificação TLBO para Treliças

5.2.1 Exemplo 1 : Torre de 25 barras

Como primeiro exemplo de implementação do código, é trazido uma conhecida estrutura, e que é um conhecido caso de *benchmark* na literatura. Os dados de entrada em relação ao algoritmo foram considerados os mesmos, e a Tabela 4 trás esses valores. O parâmetro N° de Iterações diz respeito a um critério de parada, assim como o Erro Mínimo. Se o erro relativo de peso de uma iteração para outra for menor que o valor definido como Erro Mínimo, a rotina se encerra. Senão, isso só ocorre quando atingido o n° de iterações. A tabela é de suma importância, pois serve não só para esse primeiro exemplo, mas também de todas as demais otimizações que virão a seguir.

A estrutura pode ser visualizada através da Figura 14, onde à esquerda está representada a estrutura original, com os nós todos indicados e enumerados; e à direita uma representação - qualitativa e com um fator de escala adequado para a visualização; da deformação sofrida por ela após a aplicação das cargas externas. Os dados referentes a esse problema, ou seja, à estrutura, são trazidos pela Tabela 5.

Dados de Entrada do Problema	
Expoente de Penalidade ε	2,0
Fator Professor	2
Nº da População	75
Nº de Iterações	200
Erro Mínimo	$1 E - 7$
Nº de Testes	100

Tabela 4: Dados de entrada do algoritmo.

Fonte: autor.

Nem sempre é interessante que otimize de fato todos os elementos, ou seja, atribuir valores de áreas (ou mesmo outras propriedades geométricas e construtivas) para cada um individualmente. Ao invés disso, pode-se atribuir as mesmas propriedades para diferentes conjuntos de áreas, sendo essas chamadas de grupo de projetos (*desing group*). As grupos de projeto, portanto, são um conjunto de elementos - barras, da estrutura com as mesmas propriedades de secção geométrica. Comumente se encontra para esse problema em específico oito grupos de projetos; alguns com quatro barras, outros com apenas uma ou duas. É interessante destacar que os elementos, ou barras, são, na nomenclatura do algoritmo, alunos. Ao conjunto de alunos, especificamente 75 no caso, chama-se população, ou classe. O resultado (solução ótima), conforme será apresentado a seguir, é o professor (melhor aluno) da classe ao final aplicação da metodologia do TLBO cem vezes.

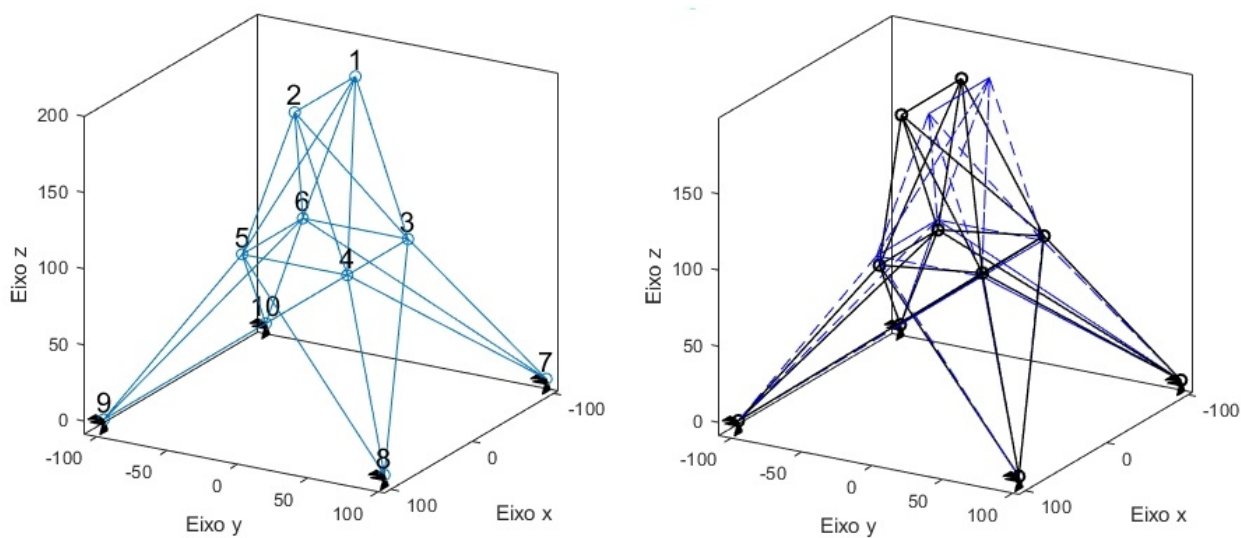


Figura 14: Estrutura do Exemplo 1, com o número de nós indicados (esq.) e após a aplicação das cargas externas (dir).

Fonte: autor.

Problema de Otimização 1	
minimizar $W = \sum_{i=1}^{n_{elem.}} \rho_i \cdot l_i \cdot A_i$, sujeito à:	
$\delta_{(x,y,z)} \leq 0,35$ [in.]	
$\sigma_{(x,y,z)} \leq 40.000$ [ksi]	
Características Construtivas	
Módulo de Elasticidade Longitudinal E	10.000.000 [ksi]
Coefficiente de Poisson ν	0,30 [-]
Densidade ρ	0,1 [lb/in ³]
Áreas Disponíveis	0,1 : 0,1 : 3,4 [in ²]

Tabela 5: Características do exemplo 1.

Fonte: autor.

Resultados

A Tabela 6 traz a comparação entre resultados obtidos por (Farshchin, 2014), (Krishnamoorthy, 1992), (Cao, 1997), (Picelli, 2019) e, obviamente, entre o presente trabalho.

Observa-se um alinhamento de resultados obtidos pelo presente trabalho com os demais autores, com erros praticamente insignificantes, com exceção do trabalho de (Krishnamoorthy, 1992). A Figura 15 mostra a análise de convergência do exemplo 1.

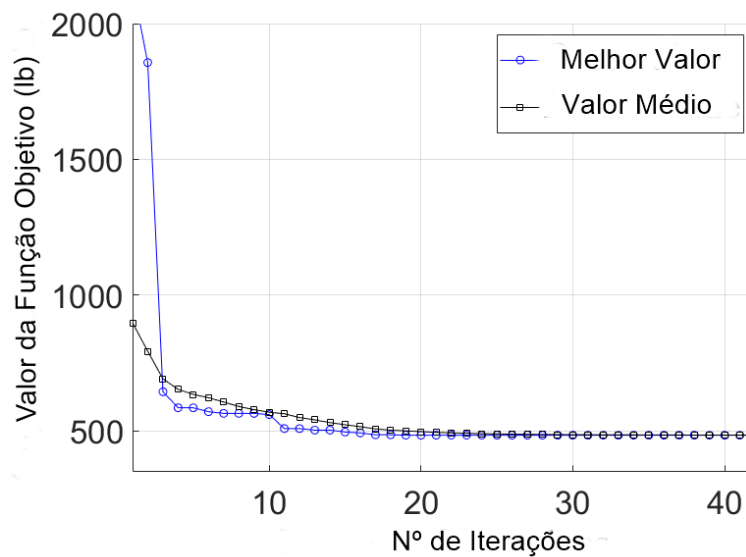


Figura 15: Análise de convergência do Exemplo 1.

Fonte: autor.

Variáveis		Área de Secção Transversal [in ²]				
GP	Elem.	Camp (2014)	Rajeev (1992)	Cao (1997)	Rodriguez (2019)	Presente
1	1	0,10	0,10	0,10	0,10	0,10
2	2 - 5	0,30	1,80	0,50	0,30	0,30
3	6 - 9	3,40	2,30	3,40	3,40	3,40
4	10, 11	0,10	0,20	0,10	0,10	0,10
5	12, 13	2,10	0,10	1,90	2,10	2,10
6	14 - 17	1,00	0,80	0,90	1,00	1,00
7	18 - 21	0,50	1,80	0,50	0,50	0,50
8	22 - 25	3,40	3,00	3,40	3,40	3,40
Peso [lb]		484,85	546,01	485,05	484,85	484,85
Erro Relativo [%]		< 0,01	11,2	0,04	< 0,01	-
Nº de Análises		4910	800	15000	-	6915

Tabela 6: Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura em diferentes trabalhos com o do presente para o Exemplo 1.

Fonte: autor.

5.2.2 Exemplo 2: Torre de 72 barras

Como segundo exemplo, outro conhecido caso da literatura de otimizações estruturais, uma estrutura de 72 barras. Os parâmetros de entradas seguem os mesmos do anterior. A estrutura desse exemplo pode ser visualizada através da Figura 16, onde, novamente, à esquerda está representada a estrutura original, com os nós todos indicados e enumerados; e à direita uma representação da deformação sofrida após a aplicação das cargas externas. Os dados referentes a esse problema são trazidos pela Tabela 7.

É de se destacar que, embora seja uma estrutura conhecida e já difundida, suas restrições e condições não são precisamente conhecidas, ou pode se dizer que não há uma unanimidade de condições, como há para o primeiro exemplo. O que é consenso é que há duas cargas independentes atuando na estrutura, conforme Tabela 8; porém nos artigos revisados não se fica claro para qual condição é, ou se elas sobrepõem-se. Além disso, há variação também na imposição da restrição de deslocamento linear. Alguns autores trazem que esta é aplicada para todos os nós, exceto os de cima (onde a carga é aplicada), outros trazem que essa restrição é válida para todos os nós, mas apenas nas direções x e y . De qualquer forma, no presente trabalho, é considerada a condição em que a restrição de deslocamento é aplicada para todos os nós nas direções x e y ; e que as cargas independentes atuam simultaneamente na estrutura.

Para esse exemplo são designados 16 grupos de projetos, contendo de duas a oito barras por

grupo.

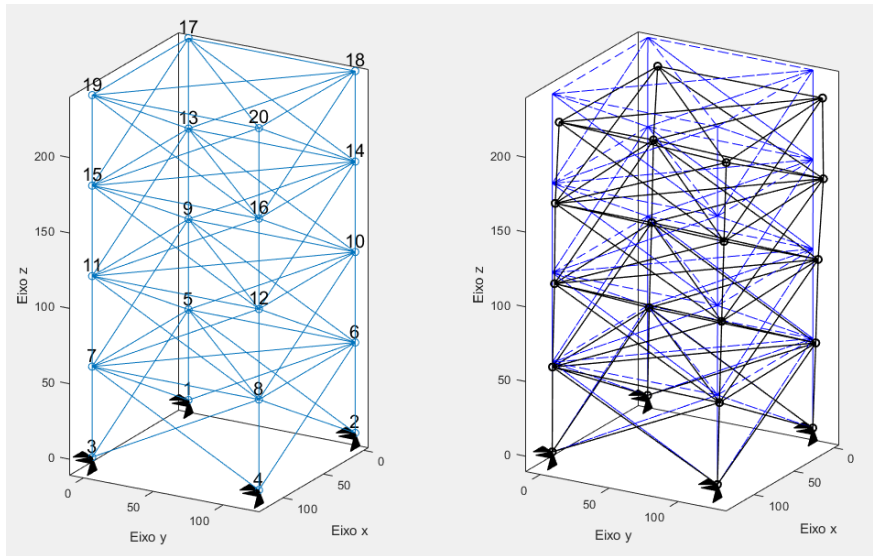


Figura 16: Estrutura do Exemplo 2, com o número de nós indicados (esq.) e após a aplicação das cargas externas (dir).

Fonte: autor.

Problema de Otimização 2	
$\text{minimizar } W = \sum_{i=1}^{n_{elem.}} \rho_i \cdot l_i \cdot A_i$	
, sujeito à:	
$\delta_{(x,y,z)} \leq 0,25 \text{ [in.]}$	
$\sigma_{(x,y,z)} \leq 25.000 \text{ [ksi]}$	
Características Construtivas	
Módulo de Elasticidade Longitudinal E	10.000.000 [ksi]
Coeficiente de Poisson ν	0,30 [-]
Densidade ρ	0,1 [lb/in ³]
Áreas Disponíveis	0,1 : 0,0001 : 3,0 [in ²]

Tabela 7: Características do exemplo 2.

Fonte: autor.

Resultados

A Tabela 9 traz a comparação com os resultados encontrados pelo algoritmo com alguns valores da literatura. Verifica-se, portanto, uma concordância entre os resultados do trabalho com os da literatura, indicando que a rotina utilizada no trabalho é satisfatória, mesma para um exemplo

Caso	Nó	P _x [kips]	P _y [kips]	P _z [kips]
1	17	0,0	0,0	-5,0
	18	0,0	0,0	-5,0
	19	0,0	0,0	-5,0
	20	0,0	0,0	-5,0
2	17	5,0	5,0	-5,0

Tabela 8: Duas condições de carregamentos do Exemplo 2.

Fonte: autor.

mais complexo. Com isso, pode-se partir para a implementação do algoritmo TLBO para o caso de pórticos. A Figura 17 mostra a análise de convergência do exemplo 2.

Variáveis		Área de Secção Transversal [in ²]				
GP	Membros	Camp (2014)	Camp (2007)	Khan (1979)	Schmit (1976)	Presente
1	1-4	1,8807	1,8577	0,1494	0,1565	1,8793
2	5-12	0,5142	0,5059	0,5698	0,5458	0,5135
3	13-16	0,1000	0,1000	0,4434	0,4105	0,1000
4	17,18	0,1000	0,1000	0,5192	0,5699	0,1000
5	19-22	1,2711	1,2476	0,6234	0,5233	1,2797
6	23-30	0,5151	0,5269	0,5234	0,5173	0,5229
7	31-34	0,1000	0,1000	0,1000	0,1000	0,1000
8	35,36	0,1000	0,1012	0,1963	0,1000	0,1000
9	37-40	0,5317	0,5209	1,2076	1,2670	0,5516
10	41-48	0,5134	0,5172	0,5208	0,5118	0,5233
11	49-52	0,1000	0,1004	0,1000	0,1000	0,1000
12	53-54	0,1000	0,1005	0,1000	0,1000	0,1000
13	55-58	0,1565	0,1565	1,7927	1,8850	0,2841
14	59-66	0,5429	0,5507	0,5223	0,5125	0,5679
15	67-70	0,4081	0,3922	0,1000	0,1000	0,3672
16	71,72	0,5733	0,5922	0,1000	0,1000	0,5274
Peso [lb]		379,632	379,850	386,718	379,640	384,339
Erro Relativo [%]		1,24	1,18	0,69	1,24	-
Nº de Análises		21542	6942	13	9	20367

Tabela 9: Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura em diferentes trabalhos com o do presente para o Exemplo 2.

Fonte: autor.

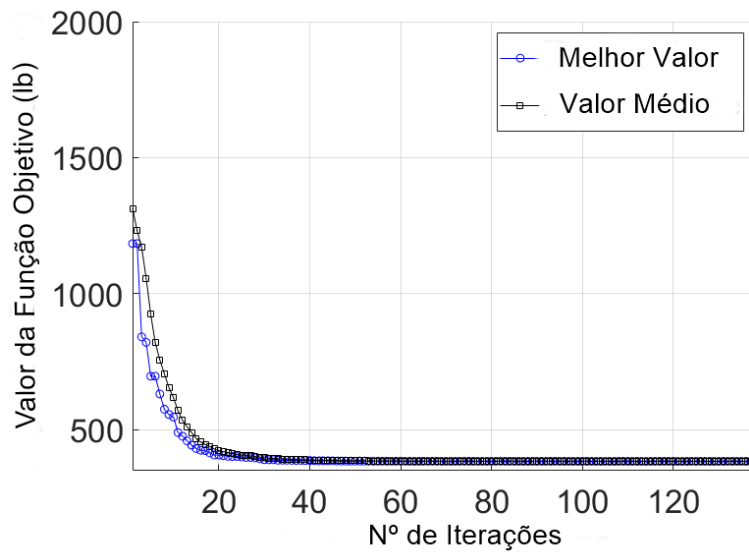


Figura 17: Análise de convergência do Exemplo 2.

Fonte: autor.

5.3 TLBO para Pórticos

Uma vez feita a validação do algoritmo TLBO para treliças e também do código de análise por elementos finitos para pórticos, pode-se finalmente unir ambas, implementando um algoritmo de otimização estrutural paramétrica para pórticos tridimensionais. Embora a modelagem ora como treliça, ora como pórtico implique numa certa variação nos equacionamentos, é relativamente simples alterar o formato da estrutura para pórtico numa rotina já implementada para treliça, principalmente porque a otimização será a mesma: paramétrica, geométrica e objetivando a redução de peso.

Logo de início, observa-se que, para treliça, a única característica geométrica de interesse era a área da seção transversal do elemento. Já para pórtico, embora esse parâmetro seja igualmente importante, não é único. Parâmetros como momento de inércia em relação aos eixos x e y , ou momento polar de inércia, por exemplo, se tornam igualmente relevantes. Além disso, devido à modelagem como pórticos oferecer não só deslocamentos lineares como angulares; e não só tensão normal axial como de flexão, deve-se adicionar restrições não só àquelas como a essas. Ou seja, para treliças era preciso apenas informar ao algoritmo um valor máximo para deslocamento linear e uma tensão axial máxima admissível. Como pórtico, além dessas restrições, é preciso adicionar igualmente valores de deslocamento angular máximo e uma tensão de flexão máxima admissível. Destaca-se que, embora isso não seja necessário de fato ao algoritmo e ele consiga otimizar pórticos apenas com restrições de deslocamento linear e tensão axial impostas, a forma aqui proposta garante maior robustez e confiabilidade ao resultado encontrado.

Para esse trabalho, o valor de máxima tensão admissível foi considerado o mesmo do de nor-

mal axial para treliça. Já o máximo deslocamento angular foi considerado de $0,1 \text{ rad}$ ($\approx 6^\circ$) para todos os casos, e foi escolhido de forma arbitrária, uma vez verificado que valores maiores ou menores que esse não exerciam influência sensível ao resultado. Já os valores de momentos de inércia e coordenada (utilizado pro cálculo da tensão de flexão) foram calculados a partir dos valores de área, que foram mantidas dentro da mesma faixa. Para isso, é necessário antes supôr uma secção transversal, caso contrário não há como chegar em um valor. Para os dois exemplos que seguem, foi considerada uma secção circular cheia, primeiramente por sua maior simplicidade nessas relações, e também por ser viável e factível. Contudo, é sugerido pelo autor que realize outras otimizações dentro das mesmas condições para outras secções, como circular vazada, retangular, perfil *I*, etc.

Conforme será visto a seguir, obteve-se, para as mesmas estrutura de 25 e 72 barras implementadas como treliça anteriormente, resultados próximos porém maiores para massa total final da estrutura. Isso era esperado e indica que a rotina foi satisfatoriamente implementada, uma vez que a estrutura como pórtico necessita ser mais rígida para suportar as mesmas condições, já que considera esforços até então desconsiderados pela treliça. Antes de entrar nos exemplos, é razoável esclarecer que, como objetiva-se comparar os resultados a serem obtidos de pórticos com os de treliça, os parâmetros de entrada do algoritmo, trazidos pela Tabela 4 e que foi usado nos exemplos 1 e 2, será igualmente usada nos 3 e 4.

5.3.1 Exemplo 3 : Torre de 25 barras

Mesmo caso utilizada no Exemplo 1, ilustrada pela Figura 14, a estrutura de 25 barras é modelada agora como pórtico. Isso implica em algumas adições nas características construtivas, que são trazidas pelo Tabela 10. Os grupos de projeto foram considerados os mesmos do Exemplo 1.

Resultados

A Tabela 11 traz a comparação de resultados obtidos entre o já mostrado trabalho de Farshchin (2014) com o presente, modelado como treliça e, finalmente, o presente trabalho modelado como pórtico. Observa-se uma compatibilidade de resultados, e uma estrutura 4,23% mais pesada em relação à treliça. Conforme já comentado, era esperado que esse valor fosse maior para modelagem como pórtico. A Figura 18 mostra a análise de convergência do exemplo 3.

Problema de Otimização 3	
minimizar $W = \sum_{i=1}^{n_{elem.}} \rho_i \cdot l_i \cdot A_i$	
, sujeito à:	
$\delta_{(x,y,z)} \leq 0,35$ [in.]	
$\theta_{(x,y,z)} \leq 0,10$ [rad]	
$\sigma_{(x,y,z)} \leq 40.000$ [ksi]	
Características Construtivas	
Módulo de Elasticidade Longitudinal E	10.000.000 [ksi]
Módulo de Elasticidade Transversal G	5.000.000 [ksi]
Coefficiente de Poisson ν	0,30 [-]
Densidade ρ	0,1 [lb/in ³]
Áreas Disponíveis	0,1 : 0,1 : 3,4 [in ²]

Tabela 10: Características do exemplo 3.

Fonte: autor.

Grupo de Projeto	Variáveis Membros	Área de Secção Transversal		
		Camp (2014)	Presente (Treliza)	Presente (Pórtico)
1	1	0,10	0,10	0,10
2	2-5	0,30	0,30	0,10
3	6-9	3,40	3,40	3,20
4	10,11	0,10	0,10	0,10
5	12,13	2,10	2,10	0,20
6	14-17	1,00	1,00	0,90
7	18-21	0,50	0,50	2,10
8	22-25	3,40	3,40	3,00
Peso [lb]		484,85	484,85	505,83
Nº de Análises		4910	6915	5243

Tabela 11: Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura ora modelado com treliça, ora como pórtico para o Exemplo 3.

Fonte: autor.

5.3.2 Exemplo 4 : Torre de 72 barras

Analogamente como feito para o Exemplo 3 em relação ao 1, o Exemplo 4 traz a a mesma estrutura de 72 barras do Exemplo 2, modelada agora como pórtico. Os grupos de projeto fo-

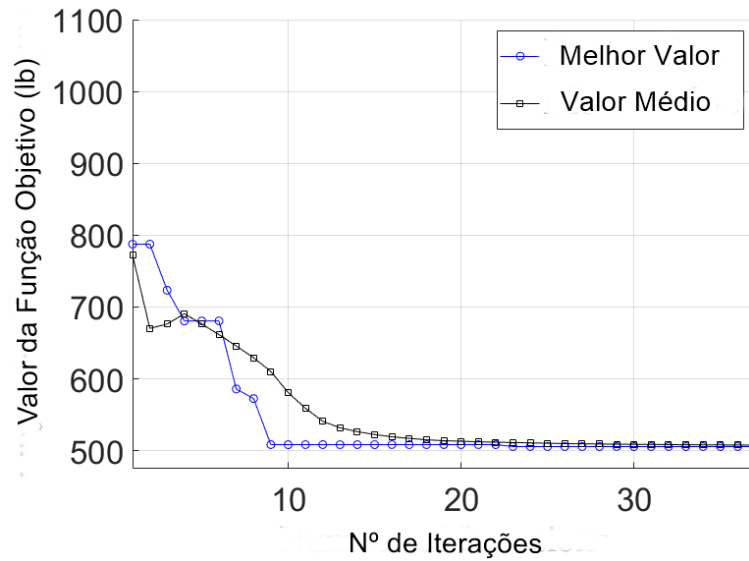


Figura 18: Análise de convergência do Exemplo 3.
Fonte: autor.

ram mantidos e novamente algumas adições pontuais foram feitas nas características construtivas, conforme Tabela 12.

Problema de Otimização 4	
minimizar $W = \sum_{i=1}^{n_{elem.}} \rho_i \cdot l_i \cdot A_i$, sujeito à:	
$\delta_{(x,y,z)} \leq 0,25$ [in.]	
$\theta_{(x,y,z)} \leq 0,10$ [rad]	
$\sigma_{(x,y,z)} \leq 25.000$ [ksi]	
Características Construtivas	
Módulo de Elasticidade Longitudinal E	10.000.000 [ksi]
Módulo de Elasticidade Transversal G	5.000.000 [ksi]
Coefficiente de Poisson ν	0,30 [-]
Densidade ρ	0,1 [lb/in ³]
Áreas Disponíveis 0,1 : 0,001 : 3,0 [in ²]	

Tabela 12: Características do exemplo 4.
Fonte: autor.

Nota-se que o incremento de área para o Exemplo 4 aumentou em relação ao Exemplo 2 (de 0,0001 no Exemplo 2 para 0,001 no Exemplo 4). Isso ocorre porque, modelando como pórtico, cada seção de área diferente necessita de uma linha na matriz de entrada de dados geométricos,

contendo nesta as informações de momentos de inércia e coordenadas associados àquela área, o que não é necessário enquanto treliça. Para o incremento utilizado aqui já foi necessária uma matriz de entrada de dados geométricos de mais de 3 mil linhas, e acredita-se não ser preciso refinar ainda mais esses valores.

Resultados

Os resultados são mostrados na Tabela 13, que confronta os obtidos por (Farshchin, 2014) e do presente trabalho, modelado como treliça, com a modelagem como pórtico. Novamente, a estrutura precisou ser mais rígida para suportar os mesmos carregamentos e, por conseguinte, mais pesada também. Mas o que mais chama atenção é a diferença encontrada entre os pesos: um incremento de expressivos 59,01 % em relação à modelagem da mesma estrutura enquanto treliça.

Esse resultado denota a importância da consideração não só de esforços axiais no projeto e dimensionamento de estruturas reticuladas, sob um risco de haver uma grande discrepância entre os resultados encontrados e a real performance da estrutura. Não que a modelagem aqui feita como pórtico prescreva exatamente o comportamento real da estrutura, visto que ainda são feitas muitas hipóteses ideais ao modelo, mas consegue se aproximar cada vez mais deste. A Figura 19 mostra a análise de convergência do exemplo 4.

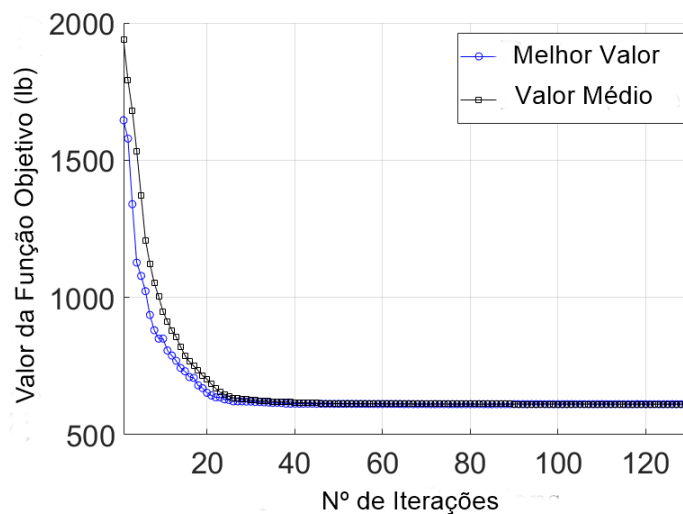


Figura 19: Análise de convergência do Exemplo 4.

Fonte: autor.

Variáveis		Área de Secção Transversal		
GP	Membros	Camp (2014)	Presente (Treliça)	Presente (Pórtico)
1	1-4	1,8807	1,8793	1,4140
2	5-12	0,5142	0,5135	0,5780
3	13-16	0,1000	0,1000	0,1000
4	17,18	0,1000	0,1000	0,1000
5	19-22	1,2711	1,2797	0,9520
6	23-30	0,5151	0,5229	0,5700
7	31-34	0,1000	0,1000	0,1000
8	35,36	0,1000	0,1000	0,1000
9	37-40	0,5317	0,5516	0,4060
10	41-48	0,5134	0,5233	0,5810
11	49-52	0,1000	0,1000	0,1000
12	53-54	0,1000	0,1000	0,2380
13	55-58	0,1565	0,2841	0,2730
14	59-66	0,5429	0,5679	0,6510
15	67-70	0,4081	0,3672	0,3570
16	71,72	0,5733	0,5274	0,6440
Peso [lb]		379,632	384,339	611,144
Nº de Análises		21542	20367	18382

Tabela 13: Comparação de resultados encontrados para secção de área transversal de cada grupo de projeto e peso total da estrutura ora modelado com treliça, ora como pórtico para o Exemplo 4. Fonte: autor.

6 Conclusão

Ao finalizar os exemplos - de verificação e autorais, o autor acredita que a metodologia proposta mostra-se satisfatória para a otimização de estruturas de pórticos tridimensionais. Acredita também que o proposto para esse trabalho, descrito no capítulo introdutório, foi de fato feito, levando o leitor à compreensão tanto do tema de otimizações estruturais em geral (com enfoque em paramétrica) quanto de algoritmos de otimização metaheurísticos (com enfoque no *Teach-Learn Based Optimization*).

Para este trabalho, houve uma preocupação especial com a revisão da literatura; e buscou-se aqui trazer-la cronologicamente de forma bem elucidativa e didática. Da mesma forma, pretendeu-se fundamentar, de maneira simplificada e objetiva, toda a teoria que regem o fenômenos físicos. A partir da revisão da literatura e da fundamentação teórica foi feita uma descrição mais detalhada da rotina computacional desenvolvida para o objetivo proposto.

Conforme visto, tentou-se também inovar ao modelar as estruturas a serem otimizadas como pórticos ao invés de treliça, garantindo, pelo menos em teoria, um caráter mais próximo da realidade. Na prática existem diversos outros fatores e fenômenos envolvidos a serem considerados no projeto de estruturas; mas dentro da proposta do trabalho, acredita-se que essa modelagem como pórtico é de grande valia, uma vez que pode-se mostrar a diferença entre resultados para uma mesma estrutura, ora considerada como treliça, ora como pórtico

Os resultados obtidos primeiramente para validação comprovam a eficácia do algoritmo desenvolvido e permitem então modelagem das estruturas como pórticos. Os resultados para essa segunda foram igualmente satisfatórios e mostram que a modelagem com pórtico implica numa consideração de esforços e deslocamentos até então desconsiderados, o que pode acarretar num projeto inadequado ou impreciso. Conforme esperado, os resultados como pórticos necessitam de estruturas mais rígidas para as mesmas condições, e portanto mais pesadas também. Para alguns determinados casos, viu-se que esse incremento na rigidez é substancial, mesmo que para estruturas idênticas sujeitas aos mesmos carregamentos.

6.1 Sugestão para Trabalhos Futuros

O presente trabalho abre um precedente para diversos outros a serem feitos a partir dele. Nota-se que a otimização paramétrica realizada foi exclusivamente geométrica, não alterando características de material ou construtivas. Sugere-se a qualquer interessado em estudar o assunto portanto, que se realize estudos comparativos entre outras formas de otimização, como de forma ou topológica. Da mesma forma, a implementação de outros métodos de otimização, não necessariamente heurísticos ou metaheurísticos, a fim de confrontar não só resultados, mas também desempenho computacional.

Aconselha-se também realizar estudos diversos sobre a influência dos parâmetros do algoritmo no resultado final, e o quão sensivelmente este se altera ao alterar-se determinado valor atribuído, como o fator professor, expoente de penalidade, número de indivíduos na população, erro mínimo, entre outros.

Por final, e também desviando mais do assunto original do trabalho, é sugerido para quem se interessar em otimizações estruturais que as faça considerando (concomitantemente ou não) esforços dinâmicos, como vibrações, impactos, etc; o que necessitaria evidentemente outras revisão literária e fundamentação teórica, mas acrescentaria muito ao tema geral de otimização estrutural.

Referências

- Jasbir S. Arora. *Introduction to optimum design*. New York: McGraw-Hill, 2004.
- Guozhong Cao. Optimized design of framed structures using a genetic algorithm. 1997.
- W. S. Hemp; H. S. Y. Chan. Optimum structures. Technical report, The College of Aeronautics, Department of Aircraft Design, 1965.
- Ashok D. Belegundu; Tirupathi R. Chandrupatla. *Optimization concepts and applications in engineering*. Cambridge University Press, 2019.
- Gino B. Colherinhas. Ferramenta de otimização via algoritmos genéticos com aplicações em engenharia. Master's thesis, 2016.
- Karilany D. Coutinho. Método de otimização topológica em estruturas tridimensionais. Master's thesis, Universidade Federal do Rio Grande do Norte, 2006.
- Marcelo da Fonseca. Otimização de estruturas treliçadas planas e espaciais sob carregamentos estáticos e dinâmicos, usando algoritmos genéticos e redes neurais. 2007. 212p. Master's thesis, Universidade Federal de Ouro Preto, 2007.
- Charles V. Camp; Mohammad Farshchin. Design of space trusses using modified teaching–learning based optimization. *Engineering Structures*, 62:87–97, 2014.
- Antonio J. M. Ferreira. *MATLAB Codes for Finite Element Analysis: Solids and Structures*. Springer, New York, 2009.
- Lucien A. Schmit Jr; Richard L. Fox. An integrated approach to structural synthesis and analysis. *AIAA Journal*, 3(6):1104–1112, 1965.
- Wei Lingyun; Zhao Mei; Wu Guangming; Meng Guang. Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. *Computational Mechanics*, 35:361–368, 2005.
- Raphael T. Haftka; Zafer Gürdal. *Elements of structural optimization*, volume 11. Springer Science & Business Media, 2012.
- W. M. Jenkins. Plane frame optimum design environment based on genetic algorithm. *Journal of Structural Engineering*, 118(11):3103–3112, 1992.
- David H. Johnson. Finite element design optimization of the wabco 170 ton haulpak truck frame. *SAE Earthmoving Industry Conference*, 1986.

- Franklin Y. Cheng; D. S. Juang. Recursive optimization for seismic steel frames. *Journal of Structural Engineering*, 115:445–466, 1989.
- Srijith Rajeev; C. S. Krishnamoorthy. Discrete optimization of structures using genetic algorithms. *Journal of structural engineering*, pages 1233–1250, 1992.
- A. Vásárhelyi; J. Lógó. Pareto optima of reinforced concrete frames. *Periodica Polytechnica Civil Engineering*, 32:87–96, 1988.
- Prakash Jadhav; P. Raju Mantena. Parametric optimization of grid-stiffened composite panels for maximizing their performance under transverse loading. *Composite structures*, 77:353–363, 2007.
- Luiz Fernando Martha. Métodos básicos da análise de estruturas. Master's thesis, Rio de Janeiro: Campus, 2010.
- James C. Maxwell. I.—on reciprocal figures, frames, and diagrams of forces. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, 26(1):1–40, 1870.
- Tushar M. Patel; Mangal G. Bhatt; Harshad K. Patel. Parametric optimization of eicher 11.10 chassis frame for weight reduction using fea-doe hybrid modeling. *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 6(2):92–100, 2013.
- Kevin Z. Truman; David J. Petruska. Parametric optimal design of steel structures. *Proc. Tenth World Conf. on Earthquake Engineering, Balkema, Rotterdam*, pages 4453–4458, 1992.
- Mohammad Farshchin; Mohsen Maniat; Charles V. Camp; Shahram Pezeshk. School based optimization algorithm for design of steel frames. *Engineering Structures*, 171:326–335, 2018.
- René Q. Rodríguez; Emanuely U. Cardoso; Patrick S. Santos; Alana P. C. Quispe; Renato S. Picelli. Structural optimization of 3d trusses considering the dynamic effect of the wind. *XL Ibero-Latin American Congress on Computational Methods in Engineering*, 2019.
- William Prager. Optimal layout of cantilever trusses. *Journal of Optimization Theory and Applications*, 23(1):111–117, 1977.
- B. S. Sarma; V. Kalyanaraman; L. N. Ramamurthy. Optimum design of lattice portal frames. *Engineering optimization*, 9:273–284, 1986.
- M. P. Kapoor; Venkata Rao. Optimum aseismic design of multistorey steel frames. *Engineering optimization*, 11:339–353, 1987.

- Piyush J. Pawar; Venkata Rao. Parameter optimization of machining processes using teaching–learning-based optimization algorithm. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):995–1006, 2013.
- Singiresu S. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.
- Tirupathi R. Chandrupatla; Ashok D. Belegundu; T. Ramesh; Chaitali Ray. *Introduction to finite elements in engineering*, volume 2. Pearson, 4 edition, 2012.
- Hervandil M. Sant’Anna. Otimização topológica de estruturas bidimensionais contínuas submetidas a restrições de flexibilidade e tensão. Master’s thesis, Universidade Federal do Rio Grande do Sul, 2002.
- Martin Philip Bendsøe; Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.
- Emílio C. N. Silva. *Otimização aplicada ao projeto de sistemas mecânicos*. Escola Politécnica da USP, Departamento de Engenharia Mecatrônica e Sistemas Mecânicos, Universidade de São Paulo, São Paulo, São Paulo, Brasil, 2018a.
- Kamilla E. S. Silva. Otimização topológica evolucionária de estruturas. Master’s thesis, Universidade do Estado de Mato Grosso, 2018b.
- Yi Min Xie; Grant P Steven. Basic evolutionary structural optimization. In *Evolutionary structural optimization*, pages 12–29. Springer, 1997.
- Wojciech Ziarko Tomasz Arciszewski. Structural optimization: Case-based approach. *Journal of computing in civil engineering*, 5:159–174, 1991.
- Venkata Rao; Vimal J. Savsani; D. P. Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3):303–315, 2011.
- Garret N. Vanderplaats. *Numerical optimization techniques for engineering design: with applications*. McGraw-Hill Book Company, 1984.
- Garret N. Vanderplaats. Structural design optimization status and direction. *Journal of Aircraft*, 36(1):11–20, 1999.
- Xin-She Yang. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.

Beibei Sun; Zhihua Xu; Xiaoyang Zhang. Parametric optimization of rubber spring of construction vehicle suspension. In *Global Design to Gain a Competitive Edge*, pages 571–580. Springer, 2008.