

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

OTIMIZAÇÃO DE TRELIÇAS COM RESTRIÇÕES DE FALHA COMBINANDO
TÉCNICAS DE PROGRAMAÇÃO DE ALGORITMOS CONTÍNUOS E DISCRETOS

por

Rodrigo Pruença de Souza

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, Outubro de 2009.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

OTIMIZAÇÃO DE TRELIÇAS COM RESTRIÇÕES DE FALHA COMBINANDO
TÉCNICAS DE PROGRAMAÇÃO DE ALGORITMOS CONTÍNUOS E DISCRETOS

por

Rodrigo Pruença de Souza

Engenheiro Mecânico

Dissertação submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Mestre em Engenharia

Área de Concentração: Mecânica dos sólidos

Orientador: Prof. Dr. Jun Sérgio Ono Fonseca

Comissão de Avaliação:

Prof. Dr. Herbert Martins Gomes

Prof. Dr. Ignacio Iturrioz

Prof. Dra. Letícia Fleck Fadel Miguel

Prof. Dr. Rafael Antonio Comparsi Laranja

Prof. Dr. Horácio Antonio Vielmo

Coordenador do PROMEC

Porto Alegre, Outubro de 2009.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Jun Sérgio Ono Fonseca pelo incentivo, amizade e ensinamentos durante esta etapa da minha formação.

Ao Prof. Dr. Ignacio Iturrioz, pela orientação inicial.

Aos amigos Rafael Sommer e Júnior pelas discussões acadêmicas, amizade e companheirismo.

A toda minha família pelo apoio, confiança e estímulo. Sempre me encorajando a galgar novos desafios.

E por fim um agradecimento mais que especial à minha esposa Raquel e a minha filha Manoela por me inspirarem e me darem forças nas batalhas do dia a dia.

“A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original.”

Albert Einstein

RESUMO

O presente trabalho tem por objetivo apresentar uma formulação para a otimização estrutural de treliças planas e espaciais submetidas a restrições de tensão e deslocamento, combinando técnicas que consideram o espaço de busca contínuo e discreto.

É implementado um algoritmo cuja finalidade é incorporar a simplicidade, eficiência e velocidade dos métodos determinísticos com a capacidade de produzir resultados mais realísticos dos métodos estocásticos.

O programa desenvolvido possui módulos de elementos finitos, análise de sensibilidade e otimização. Os problemas são resolvidos em duas etapas: A primeira etapa é baseada em Programação Linear Seqüencial (PLS). Este método de programação matemática necessita que a função objetivo e as restrições sejam sucessivamente linearizadas por expansão em séries de Taylor e a análise de sensibilidade é resolvida utilizando o método analítico. A segunda etapa usa Algoritmos Genéticos (AG) e emprega o método das funções penalizadas, no qual o problema restrito é transformado em irrestrito, associando uma penalidade às restrições violadas. Os resultados encontrados na primeira etapa são utilizados para melhorar a convergência da segunda etapa.

Para ilustrar o desempenho do algoritmo proposto são apresentados exemplos numéricos de problemas clássicos comparando-os com outros métodos encontrados na literatura.

ABSTRACT

The present work has as objective the presentation a formulation for structural optimization of plane or space truss with local stress and displacement constraints, combining techniques that consider the search space continuous and discrete.

An algorithm was implemented with the purpose to incorporate the simplicity, efficiency and rapidity of the deterministic methods with the ability to produce more realistic results of stochastic methods.

The software developed has modules for finite element, sensitivity analysis and optimization. Problems are solved in two steps: The first is based on Sequential Linear Programming (SLP). This method of mathematical programming requires that the objective function and constraints are successively linearized applying a Taylor series expansion and sensitivity analysis is solved using the analytical method. The second step uses genetic algorithms (GA) and Penalty function methods for transform the constrained problem into an unconstrained problem, associating a penalty if a constraint is violated. The results in the first stage are used to improve the convergence of the second stage.

Numerical examples are presented to illustrate the performance of the proposed algorithm comparing with other methods found in literature.

INDÍCE

1. INTRODUÇÃO.....	1
1.1 Motivação.....	1
1.2 Objetivos.....	3
1.3 Organização do trabalho.....	4
2. OTIMIZAÇÃO ESTRUTURAL.....	5
2.1 Histórico otimização estrutural.....	5
2.2 Introdução aos problemas de otimização.....	5
2.3 Definições básicas.....	7
2.3.1 Variáveis de projeto.....	7
2.3.2 Função Objetivo.....	8
2.3.3 Restrições.....	8
2.3.4 Formulação do problema de otimização.....	9
2.3.5 Domínio Viável e Inviável.....	10
3. ALGORITMOS PARA OTIMIZAÇÃO ESTRUTURAL.....	12
3.1 Classificação dos algoritmos.....	12
3.1.1 Programação Linear (LP).....	13
3.1.2 Programação Linear Seqüencial (PLS).....	15
3.2 Análise de Sensibilidade.....	17
3.2.1 Método das Diferenças Finitas (MDF).....	17
3.2.2 Método Analítico (MA).....	18
3.2.3 Método Semi-Analítico (MSA).....	20
4. ALGORTIMOS GENÉTICOS.....	21
4.1 Introdução.....	21
4.2 Histórico.....	22
4.3 Definições.....	23
4.4 Estrutura do algoritmo genético básico.....	24
4.5 Representação e Codificação.....	25
4.6 Geração da população inicial.....	26
4.7 Operadores Genéticos.....	26
4.8 Parâmetros Genéticos.....	28
4.9 Funções de Aptidão (Fitness).....	29
5. MÉTODO DOS ELEMENTOS FINITOS.....	30

5.1	Histórico do Método	30
5.2	Elasticidade Linear	31
5.3	Elasticidade Unidimensional	34
5.4	Treliças	35
5.4.1	Formulação Variacional.....	35
6.	IMPLEMENTAÇÃO COMPUTACIONAL.....	39
6.1	Determinação da derivada da restrição de flexibilidade.....	42
6.2	Determinação da derivada da restrição de tensão.....	44
6.3	Determinação da derivada da restrição de flambagem.....	45
6.4	Determinação da Função de Aptidão (Fitness).....	46
7.	APLICAÇÃO EM PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL.....	47
7.1	Introdução.....	47
7.2	Casos estudados:.....	47
7.3	Treliça de 10 barras	48
7.3.1	Resultados empregando Programação Linear Seqüencial (PLS).....	49
7.3.2	Resultados empregando Algoritmos Genéticos (AG).....	51
7.3.3	Resultados empregando Algoritmos Genéticos com utilização de população inicial	53
7.4	Treliça espacial de 25 barras	55
7.4.1	Resultados empregando Programação Linear Seqüencial (PLS).....	56
7.4.2	Resultados empregando Algoritmos Genéticos (AG).....	59
7.4.3	Resultados empregando Algoritmos Genéticos com utilização de população inicial	61
7.5	Treliça espacial de 72 barras	63
7.5.1	Resultados empregando Programação Linear Seqüencial (PLS).....	65
7.5.2	Resultados empregando Algoritmos Genéticos (AG).....	67
7.5.3	Resultados empregando Algoritmos Genéticos com utilização de população inicial	69
8.	CONCLUSÃO.....	72
9.	SUGESTÕES PARA FUTUROS TRABALHOS.....	73
10.	REFERÊNCIAS BIBLIOGRÁFICAS	75

LISTA DE SÍMBOLOS

A_i	Área da barra i,	[m ²]
A_i^{max}	Área da seção transversal máxima admissível para a barra i	[m ²]
A_i^{min}	Área da seção transversal mínima admissível para a barra i,	[m ²]
C	Coeficiente de penalização imposto para a violação restrição.	
E	Módulo de Young	[N/m ²]
f	Vetor de força externas	
$f(x)$	Função objetivo	
f_i	Vetor forças de corpo	
$f_p(x)$	Função objetivo penalizada,	
G	Módulo de elasticidade transversal	[N/m ²]
$g_j(x)$	Conjunto das restrições de desigualdade	
$h_k(x)$	Conjunto das restrições de igualdade	
I	Matriz identidade	
K	Matriz de Rigidez do MEF	
L_i	Comprimento da barra i	[m]
nap	Número de áreas de projeto	
nel	Número de elementos	
N_i	Função de interpolação do nó i	
n_j	Vetor normal unitário a superfície j	
$q(x)$	Carga distribuída na seção transversal da viga	[N/m]
r_i	Raio externo do tubo i	
t	Vetor de trações superficiais	[N/m ²]
Te	Matriz dos cossenos diretores	
t_i	Espessura do tubo i.	
U	Energia de deformação	
u	Vetor de deslocamentos	
U_i	Deslocamentos nodal do grau de liberdade i	[mm]
U_i^{max}	Deslocamento nodal máximo do grau de liberdade i,	[mm]
U_i^{min}	Deslocamento nodal mínimo do grau de liberdade i,	[mm]
W	Trabalho virtual externo	
x	Vetor das variáveis de projeto	

ε_{ij}	Deformação infinitesimal na direção j perpendicular ao eixo i	
λ_i	Multiplicadores de Lagrange.	
ρ_i	Massa específica do material da barra i,	[kg/m ³]
σ_i	Tensão normal atuante na barra i,	[MPa]
σ_i^C	Tensão admissível de compressão na barra i,	[MPa]
σ_i^F	Tensão admissível de Euler,	[MPa]
σ_{ij}	Tensão na direção de j perpendicular ao eixo i	[N/m ²]
σ_i^T	Tensão admissível de tração na barra i,	[MPa]

ÍNDICE DE FIGURAS

Figura 1.1: Fluxograma para o processo de projeto convencional.....	2
Figura 1.2: Fluxograma para processo de projeto otimizado.....	3
Figura 2.1: Problemas básicos de otimização de treliças.....	6
Figura 2.2: Regiões de domínio viável e inviável.....	10
Figura 2.3: Conceito de mínimo local e global.....	11
Figura 3.1: Processo seqüencial para encontrar o máximo da função.....	16
Figura 3.2: Efeito do tamanho do passo na derivada.....	18
Figura 4.1: Representação de um Algoritmo Genético básico.....	25
Figura 4.2: Operador de cruzamento em um ponto.....	27
Figura 4.3: Operador de mutação.....	28
Figura 5.1: Estado de tensões em torno do ponto P em um corpo tridimensional.....	32
Figura 5.2: Elemento de barra unidimensional.....	34
Figura 5.3: Forças atuando em um elemento de barra.....	36
Figura 5.4: Elemento de barra 2D genérico e suas funções de interpolação.....	37
Figura 6.1: Diagrama representativo do Algoritmo PLS.....	39
Figura 6.2: Diagrama representativo do Algoritmo Genético.....	39
Figura 7.1: Treliça de 10 barras.....	48
Figura 7.2: Gráfico da convergência do PLS para a treliça de 10 barras.....	50
Figura 7.3: Geometria final treliça plana de 10 barras deformada.....	51
Figura 7.4: Gráfico da convergência do AG para treliça de 10 barras.....	52
Figura 7.5: Gráfico da convergência do PLS+AG e AG para a treliça de 10 barras.....	54
Figura 7.6: Treliça de 25 barras.....	55
Figura 7.7: Gráfico da convergência do SLP para treliça de 25 barras.....	57
Figura 7.8: Geometria final da treliça espacial de 25 barras.....	58
Figura 7.9: Gráfico da convergência do AG para a treliça de 25 barras.....	60
Figura 7.10: Gráfico da convergência do PLS+AG e AG para a treliça de 10 barras.....	62
Figura 7.11: Treliça de 72 barras.....	63
Figura 7.12: Gráfico da convergência do PLS para treliça de 72 barras.....	66
Figura 7.13: Geometria final da treliça espacial 72 barras.....	66
Figura 7.14: Gráfico da convergência do AG para a treliça de 72 barras.....	69
Figura 7.15: Gráfico da convergência do PLS+AG e AG para a treliça de 72 barras.....	70

ÍNDICE DE TABELAS

Tabela 4.1: Vantagens dos AG em relação aos métodos de programação matemática	24
Tabela 7.1: Treliça de 10 barras - Carregamentos aplicados	49
Tabela 7.2: Treliça de 10 barras - Propriedades	49
Tabela 7.3: Treliça de 10 barras - Comparativo entre as áreas obtidas com o PLS	49
Tabela 7.4 : Treliça de 10 barras - Deslocamentos obtidos ao final da otimização com SLP	50
Tabela 7.5: Treliça de 10 barras - Comparativo entre as áreas obtidas com AG	52
Tabela 7.6: Treliça de 10 barras - Deslocamentos obtidos no final da otimização com AG	53
Tabela 7.7: Treliça de 10 barras - Comparativo dos resultados obtidos com PLS+AG e AG	54
Tabela 7.8: Treliça de 25 barras - Carregamentos aplicados	55
Tabela 7.9: Treliça de 25 barras - Variáveis de projeto	56
Tabela 7.10: Treliça de 25 barras - Propriedades	56
Tabela 7.11: Treliça de 25 barras - Tensão de flambagem em cada elemento da estrutura.....	56
Tabela 7.12: Treliça de 25 barras - Comparativo entre resultados obtidos com PLS	57
Tabela 7.13: Treliça de 25 barras - Deslocamentos obtidos no final da otimização com SLP	58
Tabela 7.14: Treliça de 25 barras - Comparativo entre resultados obtidos com AG	59
Tabela 7.15: Treliça de 25 barras - Deslocamentos obtidos no final da otimização com AG	61
Tabela 7.16: Treliça de 25 barras - Comparativo resultados obtidos com PLS+AG e AG	62
Tabela 7.17: Treliça de 72 barras - Carregamento aplicados	64
Tabela 7.18: Treliça de 72 barras - Variáveis de projeto	64
Tabela 7.19: Treliça de 72 barras - Propriedades	64
Tabela 7.20: Treliça de 72 barras - Resultados obtidos empregado PLS.....	65
Tabela 7.21: Treliça de 72 barras - Deslocamentos obtidos no final da otimização com PLS	67
Tabela 7.22: Treliça de 72 barras - Resultados obtidos com AG.....	68
Tabela 7.23: Treliça de 72 barras - Resultados obtidos com PLS + AG.....	70

1. INTRODUÇÃO

A utilização de variáveis discretas é obrigatória em muitos problemas de engenharia, pois os elementos estruturais estão disponíveis somente em algumas dimensões comerciais. Devido à dificuldade de aplicar algoritmos discretos, é comum tratar problemas que envolvem variáveis discretas como variáveis contínuas, e ao final do processo de otimização adotar um valor isolado mais próximo da solução. Entretanto, segundo [Fletcher, 1987] não existe garantia que este processo seja correto ou que produza bons resultados quando os valores das variáveis discretas estão muito espaçados.

A maioria dos métodos matemáticos e numéricos destinados ao projeto de treliças é baseada em técnicas que consideram o espaço de busca contínuo, e por isso, tendem a tratar a otimização estrutural como um problema no qual o espaço de busca é linear, quando na verdade o problema é discreto. Estes métodos são geralmente simples, eficientes e rápidos, mas conduzem a dimensões não disponíveis comercialmente.

Os Algoritmos Genéticos (AG), por sua vez, são métodos de otimização estocástica baseados na teoria da evolução. Esta técnica considera um espaço de busca discreto, produzindo resultados mais realísticos que os métodos de Programação Linear. No entanto, possuem a reputação de serem caros computacionalmente e usados somente em situações especiais ou quando todos os outros métodos falham.

1.1 Motivação

Para demonstrar a importância da otimização estrutural é necessário ressaltar as diferenças básicas entre o projeto convencional e o projeto ótimo.

No processo convencional o trabalho do engenheiro é muitas vezes baseado no método da tentativa e erro, e o dimensionamento estrutural depende apenas da habilidade, experiência e intuição do projetista. Esta maneira de elaborar projetos pode levar a resultados perigosos e errôneos, pois o processo depende unicamente do fator humano.

No método convencional não existem garantias de que a solução encontrada seja a melhor do ponto de vista econômico. A figura 1.1 ilustra o fluxograma para o processo de projeto convencional.

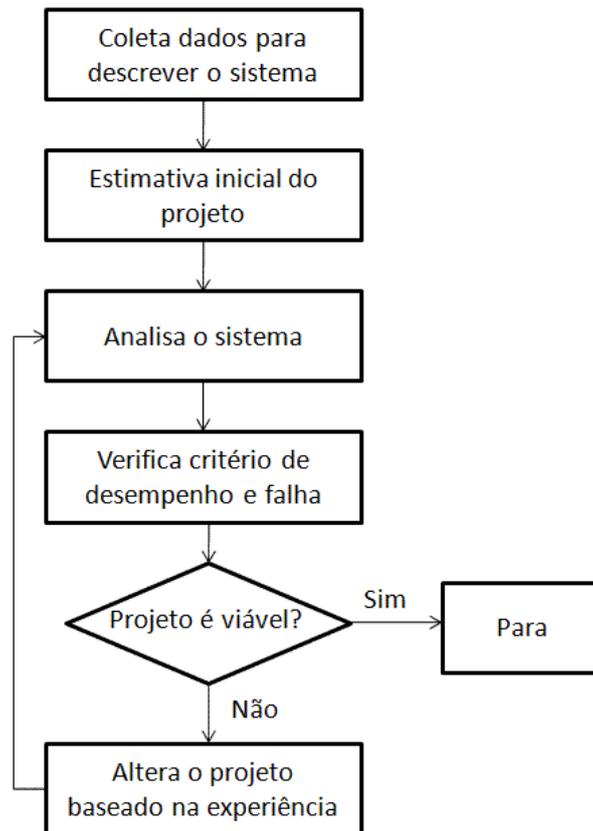


Figura 1.1: Fluxograma para o processo de projeto convencional

A necessidade atual por maior eficiência e competitividade, tem forçado os responsáveis pelo dimensionamento de estruturas a terem grande interesse nos aspectos econômicos de seus projetos. Os avanços na tecnologia computacional tornaram disponíveis aos projetistas várias disciplinas da engenharia que permaneciam intocáveis. O projeto ótimo não é apenas baseado na intuição, mas num processo iterativo que obriga o projetista a identificar explicitamente as variáveis do projeto, a função objetivo a ser minimizada ou maximizada e as restrições impostas ao sistema. Esta formulação permite obter um melhor entendimento do problema. Entretanto o engenheiro, ao optar por esta alternativa, pode obter grandes benefícios usando sua experiência e intuição, fazendo uma estimativa inicial mais correta. A figura 1.2 mostra o fluxograma para o processo de projeto ótimo.

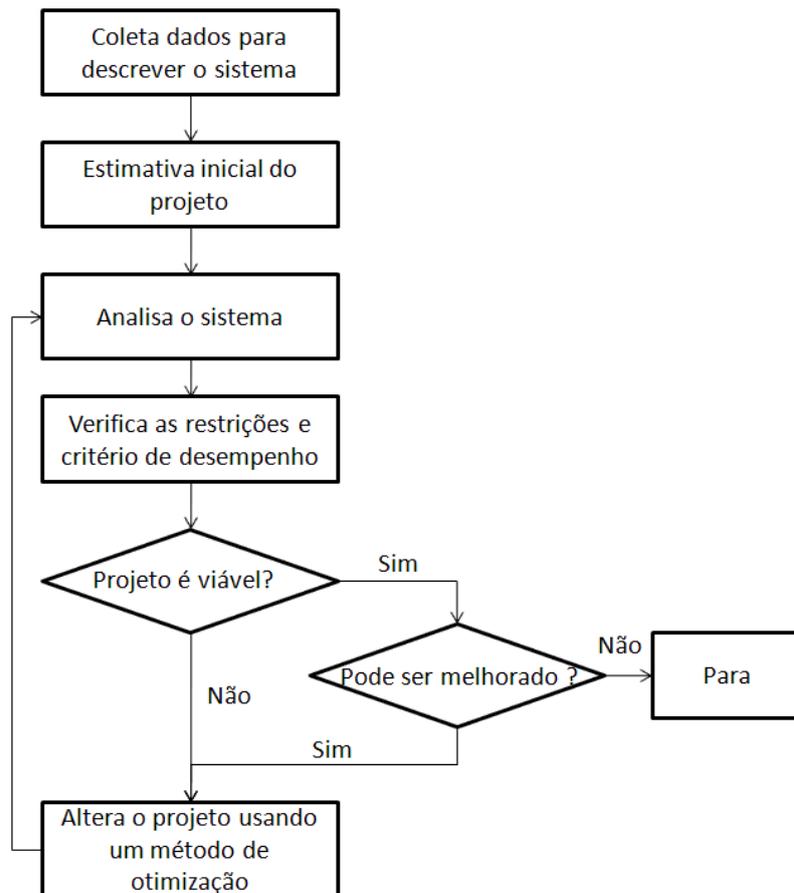


Figura 1.2: Fluxograma para processo de projeto otimizado

1.2 Objetivos

- Apresentar uma formulação matemática para minimização da massa de treliças planas e espaciais, submetidas a restrições de tensão, deslocamentos e áreas mínimas.
- Implementar um programa utilizando o MATLAB para otimização estrutural de treliças empregando dois algoritmos distintos. O primeiro baseado em Programação Linear Sequencial (PLS) utilizando variáveis contínuas para obter as soluções. O segundo método trabalha com Algoritmos Genéticos (AG) e utiliza variáveis discretas.
- Demonstrar que é possível acelerar a convergência do Algoritmo Genético através da inicialização de uma população composta pelas soluções encontradas no PLS, mantendo-se a qualidade dos resultados.

- Comparar os resultados obtidos aos encontrados na literatura, para demonstrar a eficiência dos algoritmos propostos.

1.3 Organização do trabalho

No capítulo 2 é apresentada a introdução aos problemas de otimização estrutural e suas definições básicas para a correta formulação dos problemas. Uma breve revisão bibliográfica sobre o método também é apresentada neste capítulo.

O capítulo 3 mostra alguns dos principais códigos de programação matemática utilizados neste trabalho, dando-se maior ênfase à Programação Linear Seqüencial (PLS). Neste capítulo descreve-se também os métodos de análise de sensibilidade, como o Método das Diferenças Finitas (MDF), Método Analítico (MA) e Método Semi Analítico (MSA).

O capítulo 4 apresenta uma revisão bibliográfica e os principais fundamentos da técnica de Algoritmos Genéticos (GA), suas definições, procedimentos, vantagens, parâmetros de configuração e outros tópicos importantes.

O capítulo 5 traz uma breve introdução do método dos elementos finitos utilizados para análise estrutural, bem como a determinação da matriz de rigidez dos elementos implementada no programa de análise.

O capítulo 6 apresenta a formulação matemática do problema proposto, assim como a descrição detalhada da implementação computacional da Programação Linear Seqüencial (PLS) e dos Algoritmos Genéticos (AG). Apresenta-se também um estudo das expressões das sensibilidades que foram implementadas neste trabalho.

Os resultados da aplicação dos algoritmos propostos em exemplos clássicos encontrados na literatura são apresentados no capítulo 7.

As conclusões deste trabalho são apresentadas no capítulo 8, juntamente com algumas recomendações para possível continuidade da pesquisa.

No Apêndice I é mostrada a composição detalhada dos custos de soldagem.

2. OTIMIZAÇÃO ESTRUTURAL

2.1 Histórico otimização estrutural

Os primeiros passos da otimização estrutural foram dados por [Maxwell, 1872] e posteriormente por [Michell, 1904]. Os problemas consistiam em minimizar o volume de estruturas formadas por barras (treliças) sujeitas a carregamentos aplicados em pontos do domínio e restrições de deslocamentos aplicados em outros pontos. Seus surpreendentes resultados são ainda hoje referências na teoria de otimização e também utilizados para aferir os programas comerciais de otimização estrutural.

Entretanto, após os resultados obtidos por Michell, a otimização estrutural praticamente não evoluiu até a década de 60, quando [Schmit, 1960] publicou o seu importante trabalho. O surgimento dos computadores no final desta década, somado ao desenvolvimento do Método dos Elementos Finitos (MEF) e da programação matemática possibilitou a criação de técnicas para a solução dos problemas de otimização estrutural. Antes desse período eram apenas estudados casos baseados em soluções de derivadas de equações diferenciais resolvidas de forma analítica.

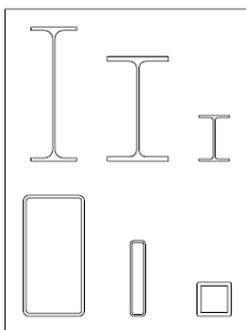
Na década de 70 um grande número de algoritmos de otimização começou a ser implementado para a solução de problemas não lineares. Na década de 80 surgem os primeiros programas comerciais de otimização estrutural e os softwares de elementos finitos passaram a incorporar módulos de otimização. Este período marca o desenvolvimento do Método de Otimização Topológica (MOT), sendo que, este método tornou-se um dos campos mais promissores para o projeto de estruturas na indústria automotiva e aeroespacial.

De 1990 até os dias atuais, o MOT tornou-se disponível em diversos programas comerciais, sendo utilizado em outras áreas da engenharia além da mecânica.

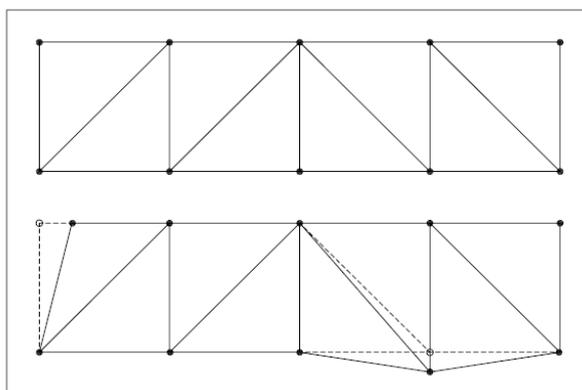
2.2 Introdução aos problemas de otimização

Existem basicamente três abordagens em otimização estrutural: otimização paramétrica, otimização de forma e otimização topológica. Na figura 2.1 são apresentadas as três abordagens aplicadas ao projeto de treliças. Na otimização paramétrica, figura 2.1.a, são otimizados parâmetros discretos como as dimensões (ou razão das dimensões) da estrutura, mantendo a sua

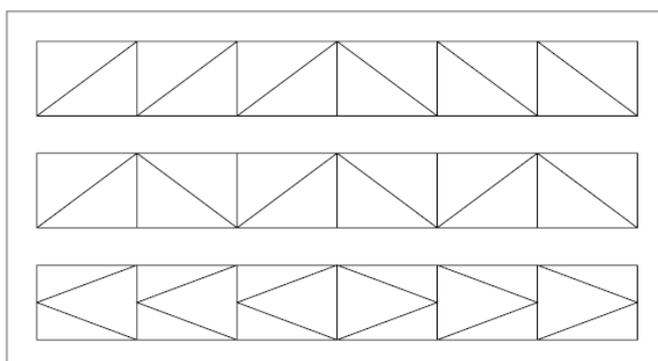
forma pré-definida. Na otimização de forma, figura 2.1.b, são alterados os contornos (internos e externos) da estrutura, ou seja, modificam-se as posições dos nós da treliça. A otimização topológica, figura 2.1.c, consiste na retirada de material do interior da estrutura formando uma nova topologia.



(a) Otimização paramétrica



(b) Otimização de forma



(c) Otimização topológica

Figura 2.1: Problemas básicos de otimização de treliças

2.3 Definições básicas

Em problemas de engenharia existem muitas soluções viáveis e o engenheiro deve selecionar a mais adequada. Basicamente, em um problema de otimização, deseja-se minimizar ou maximizar uma função, a qual é denominada função objetivo. Esta função depende de determinados parâmetros que podem ser alterados durante o processo, chamados de variáveis de projeto. Porém, a maioria dos problemas é submetida a restrições para que o projeto seja considerado viável. Estas restrições são devido às leis da física, requisitos legais, limitações no orçamento e muitos outros fatores.

Os conceitos que serão apresentados neste capítulo podem ser encontrados com maior detalhe nos trabalhos de [Haftka e Gürdal, 1992; Arora,1989].

2.3.1 Variáveis de projeto

As variáveis de projeto são o grupo de parâmetros usados para descrever o sistema [Haftka e Gürdal, 1992] e são expressos pelo vetor:

$$x = (x_1, x_2, \dots, x_n) \quad (2.1)$$

Do ponto de vista da otimização estrutural, estas variáveis podem ser agrupadas em três classes. A primeira classe de variáveis é chamada de parâmetros prescritos que, por exemplo, são os valores das propriedades dos materiais, como por exemplo: módulo de elasticidade, coeficiente de Poisson e tensões admissíveis. A segunda classe de variáveis é conhecida como variáveis de estado. São exemplos de variáveis de estado o campo de deslocamento, estado de tensões geradas por carregamentos externos, as cargas de flambagem e as amplitudes de vibrações da estrutura. A terceira classe de variáveis, chamada de variáveis de projeto, são as que efetivamente o projetista pode modificar e alterar durante a concepção do projeto.

As variáveis de projeto podem representar certa dimensão que será alterada, como área da seção transversal de uma viga, espessura de uma chapa, casca ou membrana e a orientação das fibras de um material compósito.

A escolha das variáveis de projeto é importante para o sucesso da otimização estrutural. O uso de muitas variáveis de projeto.

As variáveis de projeto são classificadas em variáveis de projeto discretas e variáveis de projeto contínuas. As variáveis de projeto discretas estão limitadas a valores isolados, enquanto as variáveis de projeto contínuas podem assumir qualquer valor. Em geral, o processo de otimização estrutural que utiliza variáveis de projeto discretas é mais difícil de ser resolvido do que os que utilizam variáveis de projeto contínuas.

2.3.2 Função Objetivo

A função objetivo é utilizada como medida da eficiência do projeto, sendo função das variáveis de projeto [Haftka e Gürdal, 1992]. A função objetivo pode ser classificada em simples ou multiobjetivo, sendo denotada por $f(x)$ e $F(x) = [f_1(x), f_2(x), \dots, f_p(x)]$, respectivamente.

2.3.3 Restrições

As restrições são limitações impostas para se obter a solução ótima. Podem ser classificadas em três tipos: laterais, igualdade e desigualdade. As restrições laterais impõem limites superiores e inferiores nos valores das variáveis de projetos.

Considerando o conjunto de variáveis de projeto, equação (2.1), uma restrição lateral é do tipo:

$$x_{\min} \leq x_i \leq x_{\max} \quad i = 1, \dots, n \quad (2.2)$$

Uma restrição de desigualdade é uma equação do tipo:

$$g_j(x) \leq 0, \quad j = 1, \dots, n_g \quad (2.3)$$

A restrição de igualdade pode ser escrita da seguinte forma:

$$h_k(x) = 0 \quad k = 1, \dots, n_e \quad (2.4)$$

As restrições de igualdade ou desigualdade são em geral mais difíceis de serem implementadas. Enquanto que, as restrições de igualdade são representadas em problemas de otimização pelas equações de equilíbrio que a estrutura deverá satisfazer em termos das variáveis

de projeto. As restrições de desigualdade são utilizadas para restringir a tensão ou deslocamento em cada ponto da estrutura.

Um aspecto relevante a ser observado é a importância da normalização das restrições. Em problemas de otimização estrutural é comum trabalhar com restrições com ordens de grandeza diferentes. No presente trabalho, por exemplo, enquanto o valor da restrição de tensão é da ordem de MPa, o valor da restrição de deslocamento é da ordem de milímetros. Este fato gera problemas de condicionamento numérico. Deste modo, deve-se normalizar as restrições como mostrado na seguinte expressão:

$$g_j(x) \leq g_{\max_j} \Rightarrow \frac{g_j(x)}{g_{\max_j}} \leq 1 \Rightarrow \bar{g}_j(x) - 1 \leq 0 \quad (2.5)$$

Procura-se evitar, na medida do possível, um grande número de restrições no problema, pois isso encarece o custo computacional do processo. Além disto, as restrições podem ser classificadas em restrições locais e globais. As restrições de tensão e deslocamento num ponto específico da estrutura são exemplos de restrições locais, pois devem ser impostas em cada ponto da estrutura. O volume máximo, frequência natural e cargas de flambagem da estrutura são exemplos de restrições globais. Transformar uma restrição local em uma restrição global pode ser útil para tratamento eficiente do problema.

2.3.4 Formulação do problema de otimização

A notação adotada para representar problemas de otimização estrutural sujeito a restrições é definida da seguinte forma [Haftka e Gürdal, 1992].

$$\begin{array}{ll} \text{Minimizar} & f(x) \\ \text{Sujeito a} & g_j(x) \leq 0, \quad j = 1, \dots, n_g \\ & h_k(x) = 0 \quad k = 1, \dots, n_e \end{array} \quad (2.6)$$

Onde x denota o vetor das variáveis de projeto, $f(x)$ é a função a ser minimizada e as funções $g_j(x)$ e $h_k(x)$ são as funções que contém as restrições de igualdade e desigualdade impostas ao sistema.

2.3.5 Domínio Viável e Inviável

Definido o problema de otimização, precisa-se determinar a região de localização de sua solução. As restrições dividem o espaço de projeto em domínio viável e domínio inviável. A figura 2.2 mostra um espaço de projeto bidimensional, onde cada eixo de coordenada representa uma variável do projeto x_i . Qualquer ponto no espaço é candidato a solução. No espaço de projeto, as constantes $f(x) = c_1$, $g_j(x) = c_2$ e $h_k(x) = c_3$, representam superfícies onde c_i é uma constante dada. A superfície $f(x) = c$ é chamada de contorno objetivo e $g_j(x) = 0$ e $h_k(x) = 0$ são as superfícies de restrição. Na maior parte dos problemas o mínimo da função é encontrado no limite entre os domínios viável e inviável, que é um ponto onde $g_k(x)=0$. No domínio viável todas as restrições são satisfeitas e pelo menos uma das restrições está ativa, enquanto no domínio inviável pelo menos uma das restrições é violada.

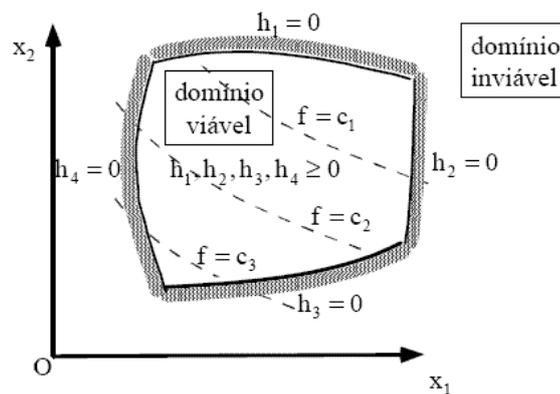


Figura 2.2: Regiões de domínio viável e inviável

A figura 2.3 mostra as curvas de nível da função objetivo f e dependendo da função restrição especificada surge o conceito de mínimo local e mínimo global. O mínimo local aparece numa cavidade “secundária”. Em geral problemas de otimização tem muitos mínimos locais e por muitas vezes os algoritmos de otimização estacionam em um destes pontos.

Em problemas de otimização estrutural é muito difícil encontrar o ótimo global devido à natureza do mesmo. Contudo, existe uma classe especial de problemas denominados convexas, onde a função objetivo é uma função convexa e o domínio viável é um domínio convexo também. Para estes problemas pode-se provar que o mínimo local é também o mínimo global.

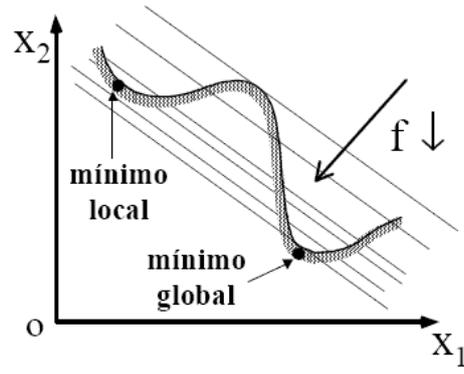


Figura 2.3: Conceito de mínimo local e global

3. ALGORITMOS PARA OTIMIZAÇÃO ESTRUTURAL

A programação matemática foi primeiramente empregada em problemas de otimização estrutural por [Schmit, 1960]. Pode ser usada diretamente em uma gama de problemas práticos nas diversas áreas do conhecimento (engenharia, economia, biologia, etc.). Muitos autores de otimização estrutural [Haftka e Gürdal, 1992; Arora,1989] têm reservado um capítulo ou mais, abordando os métodos e teorias da programação matemática em seus trabalhos

3.1 Classificação dos algoritmos

Para solucionar um problema de otimização estrutural existem diversos métodos de programação matemática, definidos de acordo com as características da função objetivo e das restrições. Os métodos de programação matemática podem ser divididos quanto às funções envolvidas: linear ou não linear.

Na programação linear a função objetivo e as restrições são funções lineares das funções objetivos, enquanto na programação não linear a função objetivo ou pelo menos uma das restrições não são funções lineares das variáveis de projeto.

A programação linear apresenta facilidade de implementação, entretanto sua aplicação a problemas de otimização em engenharia é limitada, pois a maioria dos problemas reais é não linear. Quando é possível empregar programação linear, o método com maior aplicação é o *Simplex*.

Na programação não linear pode ser encontrada uma variedade de métodos para solução de problemas irrestritos, com restrições laterais, com restrições de desigualdade e restrições de igualdade.

Os problemas de otimização não linear irrestritos ocorrem com pouca frequência em aplicações práticas de engenharia. Alguns dos métodos requerem apenas informações sobre o valor da função objetiva, outros, utilizam a primeira ou segunda derivada. Estes métodos são conhecidos como métodos de ordem zero, um e dois, respectivamente.

Os métodos *Pattern Search* e *Powell* são exemplos de métodos de ordem zero. O *Método do Gradiente* e do *Gradiente Conjugado* pertence à classe de métodos de primeira ordem, enquanto o *Método de Newton* é um exemplo típico de segunda ordem. Existem também métodos de ordem intermediária com é o caso do *Método Quasi-Newton*.

Os problemas de programação matemática que envolve funções não lineares com restrições são os de maior dificuldade de solução e podem ser divididos em três categorias: técnicas de transformação, técnicas diretas e técnicas recursivas.

As técnicas de transformação transformam um problema com restrições em um problema com restrições laterais, sendo as *funções de penalidades*, *de barreira* e o *Lagrangeano* as mais empregadas para este fim.

Os métodos de técnica direta atacam diretamente o problema restrito, através de uma seqüência de minimizações unidimensionais, ao longo de direções viáveis. São exemplos típicos desta técnica o *Método das Direções Viáveis*, do *Gradiente Reduzido* e do *Gradiente Projetado*.

A técnica recursiva é simplesmente a aplicação recursiva de um algoritmo, onde a cada iteração tanto a função objetivo quanto as restrições são linearizadas. Pode-se citar como exemplo desta técnica a *Programação Linear Seqüencial (PLS)*.

Nesta pesquisa foram utilizados os algoritmos programação linear seqüencial (SLP) e algoritmos genéticos (GA) com funções de penalidades. A explicação detalhada dos métodos de otimização podem ser encontrada nos trabalhos de [Haftka e Gürdal, 1992; Arora,1989; Cheng, 1992].

3.1.1 Programação Linear (LP)

O termo programação linear descreve uma particular classe de problemas de otimização linear, onde a função objetivo e as restrições são funções lineares em relação às variáveis de projeto. Quando as funções são não-lineares, ainda é possível utilizar este método, expandido as funções em séries de Taylor truncada nos termos lineares. Então a solução é obtida através de várias iterações. Esse método é denominado programação linear seqüencial (PLS) e será visto em detalhes na seção 3.1.2.

A formulação geral para problemas de programação linear é:

$$\begin{array}{ll}
 \text{encontrar} & x = (x_1, x_2, \dots, x_n)^T \\
 \text{min} & f(x) = C^T x \\
 \text{tal que} & Ax = b \\
 & x \geq 0
 \end{array} \tag{3.2}$$

onde n é o número de variáveis de projeto, C é o vetor de dimensão $n \times 1$, A é uma matriz de dimensões $m \times n$, com m restrições, e b é um vetor de dimensões $m \times 1$. A matriz A pode ser composta por restrições de igualdade e de desigualdade.

Em 1947, [Dantzig, 1963] propôs o método chamado “Simplex” que tornou possível a solução de problemas de otimização de vários tipos, como transporte, produção, alocação de recursos e problemas de escalonamento. O desenvolvimento dos computadores permitiu a aplicação do Simplex em problemas de grande porte, enquanto que, o método revelou alguns dos problemas numéricos que podem ocorrer em cálculos feitos através dos computadores.

Dantzig demonstrou em seu trabalho que o chamado *problema primal* da minimização da função linear sobre um conjunto de restrições lineares é equivalente a um problema de maximização de outra função linear sobre outro conjunto de restrições, chamado de *problema dual*.

O *problema primal* é definido da seguinte forma:

$$\begin{aligned} \min \quad & f_p = c_1x_1 + c_2x_2 + \dots + c_nx_n && (n \text{ variáveis}) && (3.3) \\ \text{sujeito a:} \quad & \sum_{j=1}^n a_{ij}x_j \geq b_i, && i=1, \dots, m, && (m \text{ restrições}) \\ & x_j \geq 0 && j=1, \dots, n. \end{aligned}$$

E o *problema dual* é definido:

$$\begin{aligned} \max \quad & f_d = b_1\lambda_1 + b_2\lambda_2 + \dots + b_n\lambda_n && (m \text{ variáveis}) && (3.4) \\ \text{sujeito a:} \quad & \sum_{i=1}^m a_{ij}\lambda_i \leq c_j, && j=1, \dots, m, && (n \text{ restrições}) \\ & \lambda_i \geq 0 && j=1, \dots, m. \end{aligned}$$

onde λ_i são os multiplicadores de Lagrange.

A escolha da formulação do problema *primal* ou *dual* depende do número de variáveis do projeto e do número de restrições. O esforço computacional na solução dos problemas de programação linear aumenta com o número de restrições.

A rotina LINPROG do MATLAB será utilizada para resolver os problemas propostos. Este algoritmo é conhecido como método dos pontos interiores (MPI) e foi desenvolvido por [Karmarkar, 1984]. É significativamente diferente do método simplex, que resolve o problema linear começando com um ponto interno ao longo da fronteira para, finalmente, chegar a um ponto extremo ótimo. O método projetado por Karmarkar raramente visita pontos extremos, antes que um ponto ótimo seja alcançado, ou seja, o algoritmo acha soluções no interior da solução.

3.1.2 Programação Linear Seqüencial (PLS)

O método da Programação Linear Seqüencial utilizado neste trabalho faz parte de um grupo de métodos de solução denominados de aproximação seqüencial. Este método é empregado em problemas de otimização que apresentam alto custo computacional no cálculo da função objetivo e derivadas. O problema não-linear pode ser definido da seguinte forma:

$$\begin{array}{ll}
 \text{Minizar} & f(x) \\
 \text{Sujeito a} & g_j(x) \leq 0, \quad j = 1, \dots, n_g \\
 & h_k(x) = 0 \quad k = 1, \dots, n_e \\
 & x_{\min} \leq x \leq x_{\max}
 \end{array} \tag{3.5}$$

onde x denota o vetor das variáveis de projeto sobre o qual se impõem limites máximos x_{\max} e mínimos x_{\min} , $f(x)$ é a função a ser minimizada, $g_j(x)$ e $h_k(x)$ são as matrizes que contém as restrições de igualdade e desigualdade, respectivamente. Considera-se que estas funções são todas contínuas e diferenciáveis.

A idéia básica do PLS para solucionar problemas não lineares de otimização é aproximá-lo por sucessivos subproblemas lineares de otimização que podem ser resolvidos usando métodos de programação linear. A cada iteração a função objetivo e as restrições são linearizadas usando séries de Taylor.

A formulação para cada subproblema linear é dado por:

$$\begin{array}{ll}
 \text{Minimizar} & f(x_0) + \sum_{i=1}^n (x_i - x_{0i}) \left(\frac{\partial f}{\partial x_i} \right)_{x_0} \\
 \text{sujeito a:} & g_j(x_0) + \sum_{i=1}^n (x_i - x_{0i}) \left(\frac{\partial g_j}{\partial x_i} \right)_{x_0} \geq 0 \quad j=1, \dots, n_g \\
 & h_k(x_0) + \sum_{i=1}^n (x_i - x_{0i}) \left(\frac{\partial h_k}{\partial x_i} \right)_{x_0} = 0 \quad k=1, \dots, n_g \\
 \text{e} & a_{li} \leq x_i - x_{0i} \leq a_{ui}
 \end{array} \tag{3.6}$$

Após encontrar a solução x_i do problema linearizado, substitui-se x_i no lugar de x_0 e lineariza-se novamente o problema. O processo é repetido até obter a convergência do problema não-linear.

A figura 3.1 demonstra o conceito do método. Na figura 3.1 deseja-se encontrar o máximo de uma função unidimensional $f(x)$.

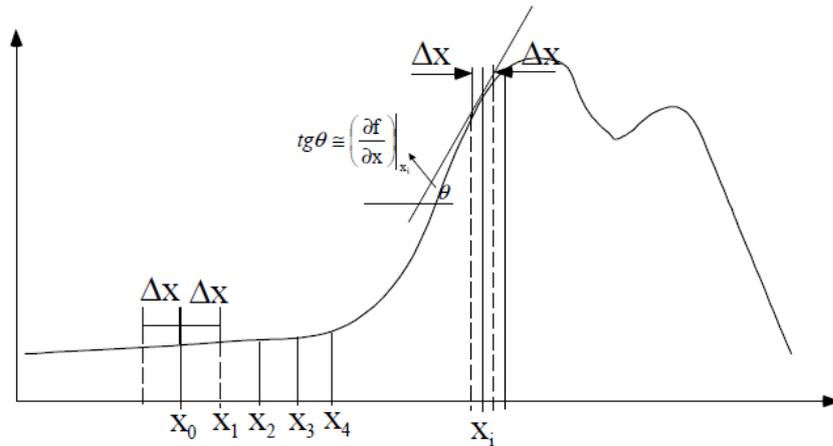


Figura 3.1: Processo sequencial para encontrar o máximo da função

Usando séries de Taylor para linearizar a função $f(x)$ em torno do ponto x_0 , tem-se:

$$f(x) \cong f(x_0) + (x - x_0) \left(\frac{\partial f}{\partial x_i} \right)_{x_0} + \varepsilon(\Delta x^2) \quad (3.7)$$

A aproximação da função $f(x)$ apresentada na equação 3.7 é somente válida próxima a vizinhança do ponto x_0 . Desta forma, a solução deste problema de otimização estará num dos extremos dos limites móveis $x_1 = x_0 + \Delta x$. Na próxima iteração a solução x_1 será o novo ponto de partida, e assim por diante. As iterações procedem com x_i sendo substituído por x_{i+1} , até que ocorre a convergência da solução.

Os limites móveis desempenham papel fundamental no processo de convergência minimizando erros na expansão em série de Taylor. Se o incremento Δx for muito grande, o erro da aproximação será grande e se forem muito pequenos podem aumentar o custo computacional para obter a solução do problema.

Na medida em que o algoritmo se aproxima da solução ótima (derivada zero) deve-se reduzir os limites móveis, caso contrário pode-se “passar” pelo ponto ótimo gerando oscilações no valor da função objetivo. Para evitar erros excessivos e aumentar a conservatividade do problema, isto é, a tendência a gerar uma solução dentro da região admissível, é acrescentada restrições laterais adicionais ao problema. Estas restrições são artificiais e vão sendo atualizadas a cada iteração, sendo por este motivo denominado *limites móveis*. O coeficiente de relação ε que é mostrado na equação (3.7) é usado para fazer a redução dos limites móveis durante cada uma das iterações.

3.2 Análise de Sensibilidade

O objetivo da análise de sensibilidade é determinar o cálculo das derivadas (ou gradientes) da função objetivo e restrições em relação às variáveis de projeto. Ela permite analisar a mudança do comportamento da estrutura em relação a pequenas mudanças nas variáveis do problema, com um custo menor do que realizar novas análises.

Existem três formas de análises básicas para o cálculo da sensibilidade: método das diferenças finitas (MDF), método semi-analítico (MSA) e método analítico (MA). A escolha do método mais adequado irá depender da expressão matemática para função objetivo e restrições.

3.2.1 Método das Diferenças Finitas (MDF)

O método das diferenças finitas é um método de resolução de equações diferenciais que se baseia na aproximação de derivadas por diferenças finitas. A grande vantagem deste método é que ele pode ser usado caso não haja a derivada analítica. A maneira mais simples de calcular os gradientes de uma função pelo método das diferenças finitas é usar a aproximação de primeira ordem, denominada diferença progressiva. Dada uma função $u(x)$ a aproximação por diferença progressiva de $\Delta u/\Delta x$ da derivada du/dx é dada pela equação (3.8).

$$\frac{\Delta u}{\Delta x} = \frac{u(x + \Delta x) - u(x)}{\Delta x} \quad (3.8)$$

Outra aproximação por diferenças finitas muito utilizada é a chamada diferença finita central, que é mostrada na equação (3.9).

$$\frac{\Delta u}{\Delta x} = \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x} \quad (3.9)$$

Também é possível usar diferenças finitas para calcular a derivada de expressões de alta ordem, mas estes casos, são raros em otimização estrutural porque apresentam um elevado custo computacional.

O grande dilema do método esta na escolha do tamanho do passo Δx adequado ao problema. Para tamanhos de passo Δx elevados predominam erros de truncamento, e para baixos

valores de Δx predomina os erros de condicionamento numérico. A figura 3.2 ilustra o efeito do tamanho do passo no cálculo da derivada.

O erro de truncamento é resultado dos termos negligenciados na aproximação du/dx utilizando expansão por séries de Taylor. O erro de condicionamento é a diferença entre o valor numérico da função e o valor exato [Haftka e Gürdal, 1992].

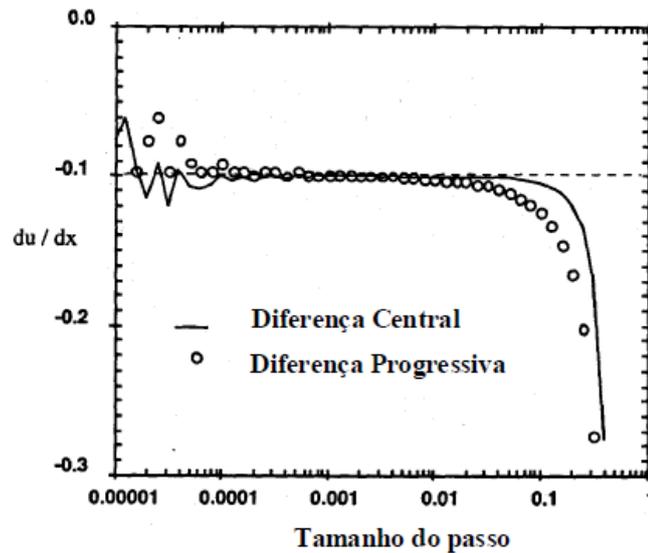


Figura 3.2: Efeito do tamanho do passo na derivada

3.2.2 Método Analítico (MA)

Entre todas as opções disponíveis para cálculo da sensibilidade os métodos analíticos são os mais precisos. Neste método a sensibilidade é determinada analiticamente antes de realizar a avaliação numérica por elementos finitos.

Existem dois métodos analíticos clássicos para o cálculo da sensibilidade: o método direto e o método adjunto. No presente trabalho adotou-se o método analítico direto para o cálculo das sensibilidades, pois para elementos finitos isoparamétricos, as expressões obtidas pelo método analítico são simples de serem implementadas e não apresentam dificuldades adicionais em relação aos outros métodos.

Considerando uma equação do tipo $f(u,x)$ que dependa dos deslocamentos nodais u e das variáveis de projeto x . Usando a regra da cadeia é possível obter a derivada desta função em relação às variáveis de projeto x :

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + z^T \frac{du}{dx} \quad (3.10)$$

onde: $z = \nabla f$ e $\left(z_i = \frac{\partial f}{\partial u_i} \right)$

Para obter o valor final da sensibilidade de f é necessário determinar a sensibilidade dos deslocamentos que dependem implicitamente das variáveis de projeto ($Z^T du/dx$). A parcela explícita (df/dx) é nula porque não depende das variáveis de projeto.

Método Direto

A equação de equilíbrio em termos dos deslocamentos nodais u é dada pela equação (3.11):

$$Ku = f \quad (3.11)$$

onde K é a matriz de rigidez e f é o vetor forças. Aplicando a regra da cadeia para obter a derivada da expressão (3.11) em relação às variáveis de projeto x obtém-se:

$$\frac{du}{dx} = K^{-1} \left(\frac{df}{dx} - \frac{dK}{dx} u \right) \quad (3.12)$$

Calculando a parte implícita da equação (3.12), a derivada total é dada:

$$\frac{df}{dx} = \frac{df}{dx} + Z^T K^{-1} \left(\frac{df}{dx} - \frac{dK}{dx} u \right) \quad (3.13)$$

Método Adjunto

O método adjunto define um vetor adjunto λ que representa a solução do sistema. A equação (3.11) pode ser reescrita da seguinte forma:

$$K\lambda = z \implies \lambda^T = z^T K^{-1} \quad (3.14)$$

Derivando a equação (3.14) em relação às variáveis de projeto e calculando a parte implícita a derivada fica:

$$\frac{df}{dx} = \frac{df}{dx} + \lambda^T \left(\frac{df}{dx} - \frac{dK}{dx} u \right) \quad (3.15)$$

Os resultados gerados pelas equações (3.13) e (3.15) são idênticos, entretanto o custo computacional entre o método direto e o método adjunto é diferente, e depende da relação entre o número de variáveis de projeto e restrições. O método direto é mais eficiente quando o número de variáveis de projeto é menor que o número de restrições.

3.2.3 Método Semi-Analítico (MSA)

A utilização do método direto ou adjunto requer as derivadas da matriz de rigidez com relação às variáveis de projeto. Esta derivada é frequentemente difícil de ser calculada analiticamente. Por essa razão o Método Semi-Analítico calcula a derivada da matriz de rigidez usando diferenças finitas, ou seja:

$$\frac{dK}{dx} \cong \frac{K(x + \Delta x) - K(x)}{\Delta x} \quad (3.16)$$

A derivada total utiliza este termo e o restante da regra da cadeia é analítica. Este método é bastante simples de ser implementado e apresenta resultados satisfatórios. Porém, em algumas estruturas modeladas por elementos finitos de viga, placa ou casca, o método demonstra problemas quando a estrutura apresenta movimentos de corpo rígido e rotação. A discussão detalhada deste problema pode ser encontrada em [Muñoz-Rojas, 2003].

4. ALGORITMOS GENÉTICOS

Os métodos baseados em derivadas necessitam que as funções sejam contínuas e diferenciáveis podendo ficar preso a mínimos locais, fato este, que limita seu domínio de aplicação. Os métodos estocásticos, por sua vez, podem trabalhar tanto com codificação contínua como discreta das variáveis e não necessitam que as funções sejam diferenciáveis. O método de Monte Carlo e os Algoritmos Genéticos são exemplos de métodos estocásticos. No entanto, o Método de Monte Carlo, apesar de possuir prova de convergência, é ineficiente para a maioria dos casos, enquanto os Algoritmos genéticos são razoavelmente eficientes para a maioria dos problemas existentes.

4.1 Introdução

Algoritmos Genéticos (AG) são métodos computacionais de otimização inspirados nos mecanismos da evolução natural e genética, descritos pela primeira vez por Charles Darwin em 1858. Foram inicialmente propostos por [Holland, 1975]. Combinam os conceitos de adaptação seletiva, troca do material genético e sobrevivência dos indivíduos mais capazes.

[Goldberg e Samtani, 1986] foram os primeiros a sugerir uso de Algoritmos Genéticos para otimização estrutural usando variáveis discretas. No artigo publicado em 1986 o peso da treliça clássica de dez barras foi minimizado. A partir deste artigo uma série de outros relevantes trabalhos surgiu na literatura.

Rajeev e seus colaboradores [Rajeev e Krishnamoorthy, 1992] usaram funções de penalização para resolver problemas com variáveis discretas aplicadas com um parâmetro que depende do grau de violação das restrições. [Adeli e Kamal, 1992] apresentaram um algoritmo genético para otimização de grandes estruturas. [Grierson e Park, 1993] resolveram alguns problemas com variáveis discretas envolvendo tamanho, geometria e topologia de estruturas de edifícios. [Cai and Thierief, 1993; Coello, 1994] apresentaram em seus trabalhos versões modificadas dos Algoritmos Genéticos. [Lemonge e Barbosa, 2000] propuseram Algoritmos Genéticos para otimização de treliças espaciais combinando variáveis discretas e contínuas. O trabalho de [Guerra, 2008] apresenta os fundamentos e conceitos necessários para aplicar Algoritmos Genéticos na otimização de treliças planas e espaciais, considerando restrições de tensões, flambagem e deslocamentos.

A idéia básica dos Algoritmos Genéticos é transformar uma população inicial de indivíduos em uma nova geração mais apta, simulando os processos naturais, através da

utilização de operadores genéticos tais como cruzamento e mutação. Cada indivíduo da população representa uma possível solução do problema de otimização, sendo representado por um conjunto de genes denominado “cromossomo”. Os cromossomos, por sua vez, são representados por uma cadeia de bits de tamanho variável, que definem as características do indivíduo.

4.2 Histórico

Nesta seção será apresentado um breve histórico dos Algoritmos Genéticos baseado nos trabalhos de [Castro, 2001; Guerra, 2008].

Durante o século XIX, os naturalistas acreditavam que cada espécie havia sido criada separadamente por um ser supremo ou através da geração espontânea. O trabalho do naturalista Carolus Linnaeus levou a acreditar na existência de certa relação entre as espécies. Por outro lado, Thomas Robert Malthus propôs que fatores ambientais tais como doenças e carências de alimentos, limitavam o crescimento de uma população. No século XX, após anos de observação e experimentos, Charles Darwin apresentou em 1858 sua teoria da evolução através de seleção natural. Esta teoria privilegia os indivíduos mais aptos e com maior probabilidade de reprodução.

Os princípios básicos de genética populacional se baseiam na idéia que a variabilidade entre indivíduos de uma população de organismos que se reproduzem sexualmente se dá pela mutação e pela recombinação genética. Estas idéias foram desenvolvidas durante os anos 30 e 40 por biólogos e matemáticos de importantes centros de pesquisa.

A década de 40 também marca o surgimento do Método de Monte Carlo (MMC). O MMC caracteriza-se por ser um método estatístico utilizado em simulações estocásticas com diversas aplicações em áreas da física, matemática e engenharia. Sua origem data de 1949, com a publicação do artigo “The Monte Carlo Method” [Metropolis, 1949].

Nas décadas de 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos, entretanto, foi John Holland, da universidade de Michigan, quem começou a desenvolver as primeiras pesquisas sobre o tema. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro [Holland, 1975], hoje considerado a principal referência em Algoritmos Genéticos. Nos anos 80, David Goldberg, ex-aluno de Holland, consegue o primeiro sucesso em aplicação industrial de algoritmos genéticos. Os estudos de Goldberg foram publicados [Goldberg, 1989]. Desde então, estes algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização.

4.3 Definições

Segundo [Goldberg, 1989], as principais definições relacionadas com os Algoritmos Genéticos são:

- **Cromossomo** – Cadeia de caracteres representando alguma informação relativa às variáveis do problema. Cada cromossomo representa deste modo uma possível solução do problema.
- **Gen ou Gene** – É a unidade básica do cromossomo. Cada cromossomo tem certo número de gens, cada um descrevendo uma variável do projeto.
- **População** – Conjunto de cromossomos ou soluções.
- **Geração** – O número de iterações que o algoritmo genético executa.
- **Operações genéticas** – Operações que o algoritmo genético realiza sobre cada cromossomo.

Os AG constituem uma classe de ferramentas versátil e robusta que pode ser utilizada na solução de problemas de otimização estrutural [Goldberg, 1989], embora não devam ser considerados estritamente minimizadores de funções. A tabela 4.1 mostra uma comparação entre os métodos de programação matemática, Algoritmos Genéticos e Monte Carlo..

Tabela 4.1: Vantagens dos AG em relação aos métodos de programação matemática

Métodos de Programação Matemática	Algoritmos Genéticos	Método de Monte Carlo
Têm dificuldade em identificar soluções ótimas globais, uma vez que dependem do ponto de partida.	Não apresentam nenhuma restrição quanto ao ponto de partida.	Não apresentam nenhuma restrição quanto ao ponto de partida.
Têm dificuldade em tratar problemas de engenharia com variáveis discretas.	Trabalham tanto com codificação contínua como discreta das variáveis ou ainda uma combinação de ambas.	Trabalham tanto com codificação contínua como discreta das variáveis ou ainda uma combinação de ambas.
Requerem funções diferenciáveis.	Não necessitam que a função objetiva ou restrições sejam contínuas ou diferenciáveis.	Não necessitam que a função objetiva ou restrições sejam contínuas ou diferenciáveis.
Têm domínio de aplicação restrito.	São razoavelmente eficientes para a maioria dos problemas existentes	São ineficientes para a maioria dos problemas existentes.

Os Algoritmos Genéticos não são presos a mínimos locais, como os algoritmos clássicos, podendo por esse motivo, levar a descoberta de soluções não convencionais e inovadoras.

4.4 Estrutura do algoritmo genético básico

A figura 4.1 é apresentada a representação de um AG genérico, o qual funciona da seguinte forma:

- Inicialmente é gerada uma população, na forma de cromossomos, formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema e respectivo cálculo da função objetivo.

- Acontece então o processo evolutivo, onde esta população é avaliada e cada indivíduo recebe uma nota, ou índice, refletindo sua habilidade de adaptação a um determinado ambiente.
- Posteriormente inicia-se o processo iterativo, uma porcentagem dos indivíduos mais adaptados é mantida, enquanto os outros são descartados.
- Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamentos ou recombinação genética gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada. Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são suficientemente complexos para fornecer mecanismos de busca adaptativos poderosos e robustos.

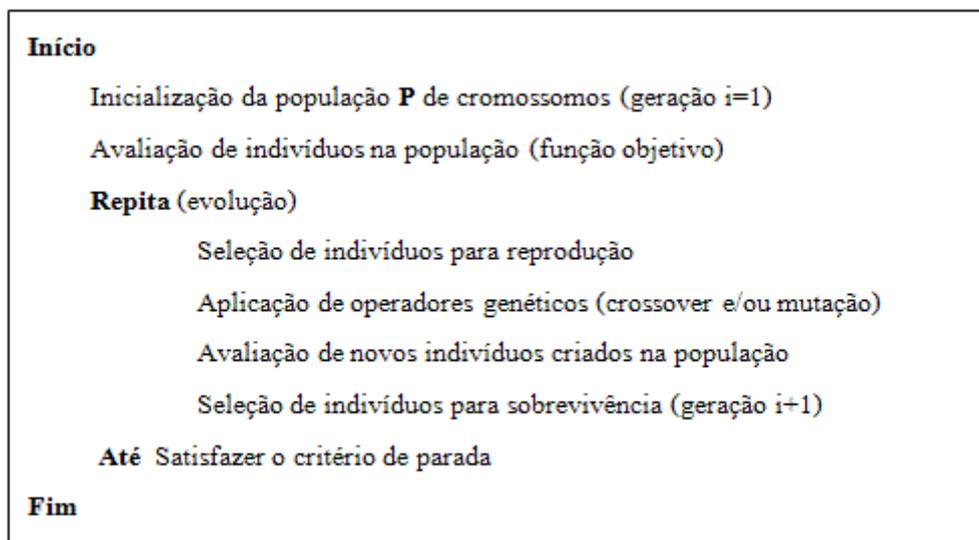


Figura 4.1: Representação de um Algoritmo Genético básico

4.5 Representação e Codificação

Basicamente existem três tipos de codificação dos parâmetros em algoritmos genéticos. A codificação clássica (ou binária), números inteiros e a codificação real. Esta codificação é necessária para transformar as variáveis do problema em um cromossomo para que o algoritmo possa manipular corretamente. A codificação inteira representa uma gama de problemas muito pequena.

A codificação binária, ou clássica foi a primeira a ser construída, sendo utilizada na maioria dos trabalhos desenvolvidos. Na codificação binária cada cromossomo é um vetor

composto por zeros e uns, com cada bit representando um gene do mesmo. Porém, conforme a necessidade de maior precisão numérica, as cadeias de bits tornam-se excessivamente longas, o que acarreta a necessidade de um esforço computacional maior, causando também um consumo maior de tempo até a convergência dos algoritmos.

A codificação real trabalha diretamente com os números reais, o que torna possível cobrir um domínio bastante abrangente, mesmo para domínios desconhecidos, das variáveis. Utiliza-se números de ponto flutuante para representar o valor das variáveis e executar as operações genéticas de cruzamento e mutação. Existe assim, uma grande desvantagem no caso da representação binária, que seriam as constantes conversões entre valores de ponto flutuantes e binários.

4.6 Geração da população inicial

Geralmente a população inicial de indivíduos ou cromossomos é realizada de forma aleatória. Neste trabalho, é introduzida uma população inicial de indivíduos resultantes de um processo de otimização anterior, baseado em Programação Linear Seqüencial na tentativa de acelerar o processo de otimização. No entanto, a literatura demonstra [Goldberg, 1989] que a inicialização usando uma população não é crítica, ou seja, se o número de indivíduos da população for grande suficientemente a utilização de uma população inicial não afeta o processo.

4.7 Operadores Genéticos

Segundo Goldberg o princípio básico dos operadores genéticos é transformar a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Os operadores genéticos são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores. Entre os principais operadores genéticos existentes, podem-se citar dois: cruzamentos e mutações que têm papel significativo nos Algoritmos Genéticos.

Cruzamento

O cruzamento é o operador responsável pela recombinação de características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características. A idéia principal é propagar as características positivas dos indivíduos mais aptos da população através da troca de segmentos de informações entre os mesmos, o que originará novos indivíduos.

As formas mais comuns de troca de segmentos nos AG são:

- um - ponto: O ponto de cruzamento é escolhido de forma aleatória e a partir deste ponto as informações genéticas dos pais serão trocadas. As informações anteriores a este ponto em um dos pais são ligadas às informações posteriores a este ponto no outro pai, como é mostrado na figura 4.2.
- multipontos: é realizada da mesma forma do cruzamento de um ponto, porém a troca de material genético através de um número maior de pontos, onde muitos pontos de cruzamento podem ser utilizados.

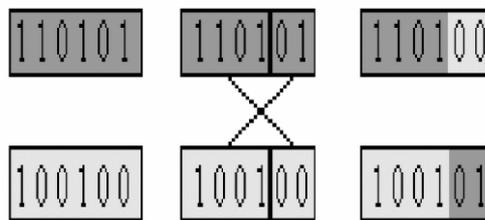


Figura 4.2: Operador de cruzamento em um ponto

Mutação

O operador de mutação é conhecido com um operador “background” sendo responsável pela introdução e manutenção da diversidade genética da população [Holland, 1975], alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, como é ilustrado na figura 4.3. Trata-se de uma modificação aleatória no valor do elo da cadeia. Caso o elo escolhido seja zero ele passa a valer um e vice-versa. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca seja zero, com o intuito de tentar contornar o problema de ótimos locais.

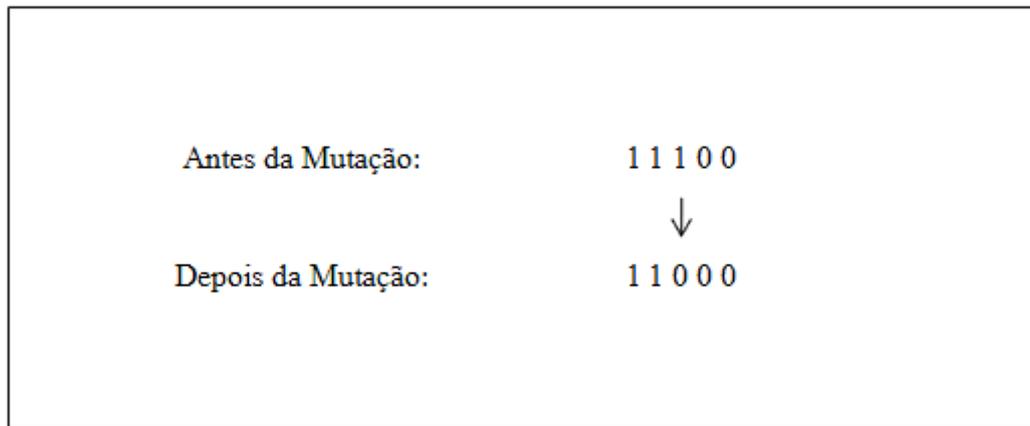


Figura 4.3: Operador de mutação

4.8 Parâmetros Genéticos

Os parâmetros genéticos influenciam o comportamento e desempenho dos algoritmos genéticos, e afetam da mesma forma a codificação real e a codificação clássica.

A eficiência dos Algoritmos Genéticos é altamente dependente dos seguintes parâmetros:

Tamanho da População: O tamanho da população afeta o desempenho global e a eficiência dos AG. Ao trabalhar-se com uma população pequena o desempenho pode se comprometer, pois deste modo o espaço de busca é reduzido. Uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. Porém, ao trabalhar com grandes populações, são necessários maiores recursos computacionais.

Taxa de Cruzamento: Uma alta taxa de cruzamento permite que novas estruturas sejam introduzidas na população. No entanto, se esta taxa for muito elevada, a maior parte da população com boas aptidões serão substituídas. Um valor baixo, o algoritmo pode tornar-se muito lento.

Taxa de Mutação. Uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue a qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória.

Intervalo de Geração: Determina a porcentagem da população que será substituída na geração seguinte. Caso o valor seja alto, a maior parte da população é substituída, aumentando as chances de perder estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.

4.9 Funções de Aptidão (Fitness)

As funções de penalização são utilizadas para considerar as restrições em algoritmos genéticos. Basicamente, essa técnica consiste em utilizar uma combinação linear da função objetivo e uma função penalização para testar os membros de uma determinada geração. A função objetivo é sempre bem definida, pois já está estipulada no problema de otimização a ser resolvido. O problema consiste, pois, em determinar "como" penalizar.

Um método simples para se penalizar soluções não viáveis é aplicar uma penalidade constante. Assim, a função objetivo penalizada assume o valor da função objetivo não penalizada adicionada da penalidade. Existem técnicas alternativas a proposta no presente trabalho e podem ser encontradas em [Deb, 2001; Kripka, 2004]. A função de penalização para o problema de minimização, com muitas restrições pode ser escrito da seguinte forma:

$$f_p(x) = f(x) + \sum C_i \delta_i \quad (4.1)$$

Onde:

$f_p(x)$ é a função penalizada,

$f(x)$ é a função original

$\delta_i = 1$ se a restrição é violada

$\delta_i = 0$ se a restrição não é violada

C_i é o coeficiente de penalização imposto para a violação de i -ésima restrição.

5. MÉTODO DOS ELEMENTOS FINITOS

O Método dos Elementos Finitos (MEF) é uma ferramenta de simulação numérica que pode obter soluções aproximadas de uma abrangente variedade de problemas de engenharia como: análise de tensões, escoamento de fluidos e transferência de calor. Embora originalmente desenvolvido para estudar tensões em estruturas aeronáuticas complexas, vêm sendo amplamente empregado em diversos campos da mecânica do contínuo.

Os métodos analíticos tradicionais permitem o cálculo dos deslocamentos, deformações e tensões na estrutura, entretanto são limitados a casos simples. Desta forma, têm-se procurado desenvolver procedimentos aproximados que pudessem ser aplicados, independente da forma da estrutura e da condição de carregamento. Dentre estes métodos, os que têm sido mais utilizados são aqueles baseados na divisão do meio contínuo em geometrias mais simples.

O Método dos Elementos Finitos é a técnica de discretização de sistemas contínuos mais utilizada como alternativa aos procedimentos analíticos clássicos.

5.1 Histórico do Método

Embora o termo “elementos finitos” tenha sido citado primeiramente por [Clough, 1960], em um artigo sobre problemas de elasticidade plana, as idéias da análise de elementos finitos datam de muito antes. De fato, a questão de quem originou o método dos elementos finitos? e quando começou? há três diferentes respostas, depende se perguntado a um matemático, físico ou engenheiro [Felippa, 1995]. Todos estes especialistas têm suas justificativas para reivindicar o método dos elementos finitos, porque cada um desenvolveu as idéias independentemente e em diferentes momentos e por razões diferentes. O esforço destes três grupos resultou em três conjuntos de artigos com distintos pontos de vista.

Em 1943, o matemático Richard Courant [Courant, 1943] sugeriu a utilização de funções lineares espaciais sobre regiões triangulares e aplicou o método para solução de problemas de torção. Este elemento é conhecido como triângulo de deformação constante (CST), sendo utilizado até hoje para resolver problemas de elasticidade plana.

As Publicações de [Argyris e Kelsey, 1960; Turner et al, 1956] podem ser consideradas um marco no estudo do MEF, pois tais publicações uniram os conceitos de análise estrutural e

análise do contínuo. Assim, as equações de rigidez passaram a ser escritas em notação matricial e resolvidas em computadores digitais.

Em 1963, o método foi reconhecido como rigorosamente correto e tornou-se uma respeitável área de estudos acadêmicos. Este período foi seguido por um intensivo desenvolvimento de programas computacionais, e passou a ser utilizado também na solução de problemas de mecânica dos fluidos, termodinâmica e magnetismo.

5.2 Elasticidade Linear

Os corpos materiais se deformam quando submetidos a ações externas (forças devidas ao contato com outros corpos, ação gravitacional agindo sobre sua massa, etc.), e retornam a sua configuração inicial quando a carga é retirada. Esta afirmação é válida em elasticidade linear até certo limite, onde as tensões aplicadas são proporcionais à deformação.

A teoria da elasticidade fornece os conceitos fundamentais para a solução dos problemas da mecânica clássica e método dos elementos finitos. Na solução de problemas, seja analiticamente ou numericamente é necessário resolver as equações de equilíbrio para determinar a resposta do sistema. Os trabalhos de [Boresi e Chong, 1987; Atkin e Fox, 1980] apresentam de detalhadamente as equações de equilíbrio e a teoria da elasticidade.

A equação de equilíbrio é escrita da seguinte forma:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + f_i = 0 \quad (5.1)$$

O termo σ_{ij} refere-se à tensão na direção do eixo j perpendicular ao eixo i , f é o vetor de força de corpo e representa a força distribuída por unidade de volume. Estas equações são à base de toda a mecânica do contínuo. Aplicando o princípio da conservação da quantidade de movimento angular [Atkin e Fox, 1980], chega-se à conclusão que o tensor de tensão é simétrico, para isso a seguinte relação dever ser satisfeita:

$$\sigma_{ij} = \sigma_{ji} \quad (5.2)$$

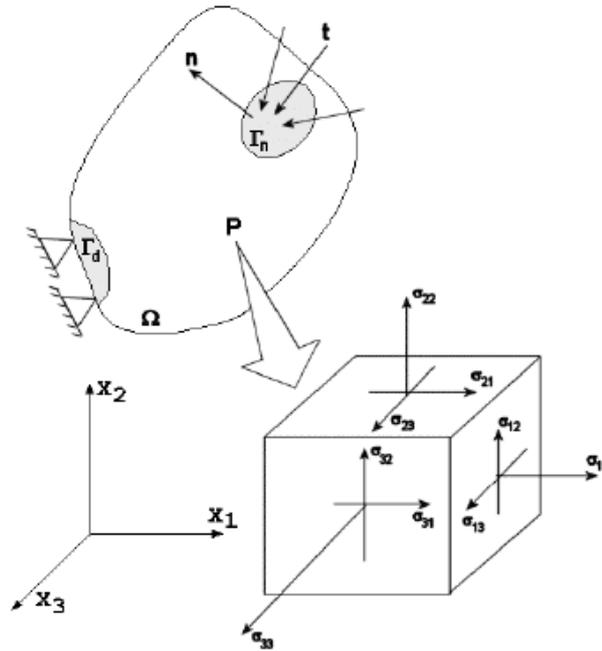


Figura 5.1: Estado de tensões em torno do ponto P em um corpo tridimensional

Conhecendo os componentes de tensão σ_{ij} e as normais n_j de uma superfície em um elemento diferencial como o mostrado na figura 5.1, o vetor de tração atuando em qualquer plano neste elemento diferencial pode ser calculado. Este vetor é conhecido como tensor *tensão de Cauchy* e pode ser calculado pela equação (5.3):

$$t_i = \sigma_{ij}n_j \quad (5.3)$$

Para aplicações em estática, normalmente se requer apenas a posição inicial e final do corpo. O mapeamento da configuração inicial para a final é chamado de deformação do corpo. O movimento do corpo pode ser descrito através da seguinte relação:

$$u = u_0(x_1, x_2, x_3) \quad (5.4)$$

onde u_0 é o vetor que representa a configuração no início do fenômeno.

Desprezando os termos de alta ordem, define-se o tensor de deformações infinitesimais ε como:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (5.5)$$

onde ε_{ij} é a componente da deformação infinitesimal na direção j perpendicular ao eixo i (figura 5.1).

Sabe-se através da lei de Hooke que a tensão é diretamente proporcional à deformação. Esta regra é limitada a pequenas deformações e certos materiais. A forma usual da elasticidade infinitesimal linear pode ser escrita:

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + 2\mu \varepsilon_{ij} \quad (5.6)$$

onde λ e μ são os coeficientes de Lamé. Portanto, um material isotrópico apresenta apenas duas constantes independentes. Os materiais mais comuns em engenharia podem ser analisados como isotrópicos, especialmente os metais policristalinos, cujo tamanho de grão é pequeno em relação à peça. Entretanto, em engenharia, prefere-se trabalhar com constantes diferentes do módulo de Young, o coeficiente de Poisson, ou o coeficiente volumétrico. As equações para a conversão das constantes elásticas isotrópicas podem ser encontradas com detalhe em [Atkin e Fox, 1980].

No estado plano de tensões (EPT) a espessura do corpo é pequena comparada com as outras dimensões e os carregamentos são aplicados no plano que contém a estrutura. Estas considerações permitem desconsiderar as tensões em uma das direções da estrutura ($\sigma_{33}=0$). Com isto, os coeficientes de Lamé podem ser relacionados ao módulo de Young (E), coeficiente de Poisson (ν) e Módulo de elasticidade transversal (G) pelas seguintes expressões:

$$E = \mu \frac{3\lambda + 2\mu}{\lambda + \mu} \quad (5.7)$$

$$\lambda = \frac{\nu E}{1 - \nu^2} \quad (5.8)$$

$$\mu = G = \frac{E}{2(1 + \nu)} \quad (5.9)$$

O estado plano de deformação (EPD) se caracteriza por ter a espessura grande comparada com as outras dimensões, e os carregamentos serem aplicados apenas no plano transversal da estrutura. Pode-se, desta forma, assumir que o deslocamento em uma das direções é desprezível ($\epsilon_{33}=0$). Neste caso:

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad (5.10)$$

5.3 Elasticidade Unidimensional

No presente trabalho deseja-se otimizar estruturas compostas de barras, desta forma, será utilizada a hipótese de que a tensão seja unidimensional para reduzir o problema de mecânica dos sólidos para uma dimensão. Seja o corpo de seção transversal constante orientado com o eixo (1) mostrado na figura 5.2. Supondo que todos os carregamentos t estão aplicados na direção (1) e uniformemente distribuídos na seção transversal.

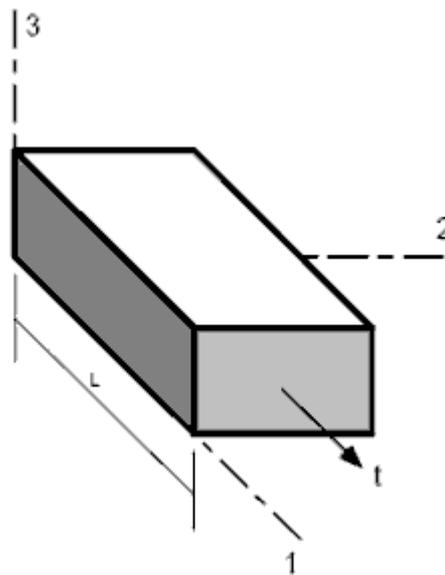


Figura 5.2: Elemento de barra unidimensional

A equação (5.3) pode ser simplificada, pois as trações se anulam nas outras faces ($t_2=t_3=0$).

$$t_1 = \sigma_{1j} n_j \quad (5.11)$$

Se não houver forças de corpo a tensão será constante, e seu valor vai depender apenas das cargas aplicadas no contorno. Para um material isotrópico com deslocamentos infinitesimais o problema de mecânica dos sólidos pode ser reduzido à seguinte forma:

$$\sigma_{11} = E\varepsilon_{11} \quad (5.12)$$

5.4 Treliças

A Treliça consiste em uma série de elementos estruturais retos de comprimento muito maior que as dimensões de sua seção transversal, e que conectados uns aos outros em suas extremidades compõem uma estrutura reticulada [Alves Filho, 2000]. A particularidade importante das treliças é que as juntas estruturais conectadas, ou seja, o encontro discreto de dois ou mais membros do conjunto são articulados.

As forças externas atuantes em uma estrutura em forma de treliças são aplicadas nos nós, excepcionalmente uma barra de treliça poderá estar sujeita a cargas atuando entre os nós. Porém, para propósitos de análise, estas cargas poderão ser substituídas por cargas estaticamente equivalentes atuando no interior nos nós.

As barras das treliças transmitem apenas forças axiais de tração ou compressão, isto é, na direção da barra. Para este elemento não são contabilizados esforços decorrentes da ação de momentos fletores, torçores e forças cortantes.

5.4.1 Formulação Variacional

Nos mecanismos dos materiais é possível demonstrar que a energia interna U em um ponto de um material com comportamento linear-elástico sujeito a um estado de tensão unidimensional σ e uma deformação ε é dada pela seguinte relação:

$$U = \int_0^L \sigma(x)\varepsilon(x)dx \quad (5.13)$$

onde σ é relacionado com o deslocamento u através da lei de Hooke equação (5.12) e a relação deformação-deslocamento é dada pela equação (5.5).

O trabalho virtual externo W pode ser definido como o trabalho feito pelas forças de superfície e de corpo. Entretanto, para este trabalho as barras em estudo estão submetidas apenas a forças discretas, e nenhuma força de corpo atua. A expressão do trabalho fica reduzida a:

$$W = \int_0^L q(x)u(x)dx \quad (5.14)$$

$q(x)$ é a carga distribuída na seção transversal da viga apresentada na figura 5.3

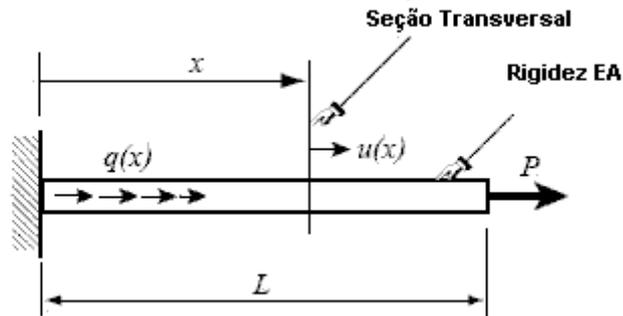


Figura 5.3: Forças atuando em um elemento de barra

A função do deslocamento aproximado $u(x)$ varia linearmente ao longo do elemento e pode ser escrita:

$$u^{(e)}(x) = N_i^{(e)}u_i^{(e)} + N_j^{(e)}u_j^{(e)} \quad (5.15)$$

As funções $N_i^{(e)}$ e $N_j^{(e)}$ que multiplicam os deslocamentos nodais $u_i^{(e)}$ e $u_j^{(e)}$ são chamadas de funções de interpolação. Estas funções interpolam os deslocamentos internos diretamente nos nós. Na figura 5.4 é possível visualizar estas funções que são definidas nas equações (5.16) e (5.17):

$$N_i^{(e)} = 1 - \frac{x}{\ell} = 1 - \zeta \quad (5.16)$$

$$N_j^{(e)} = \frac{x}{\ell} = \zeta \quad (5.17)$$

onde $\ell = L^{(e)}$ denota o comprimento do elemento, e $\zeta = x/\ell$ é a coordenada adimensional, também conhecida como coordenada natural.

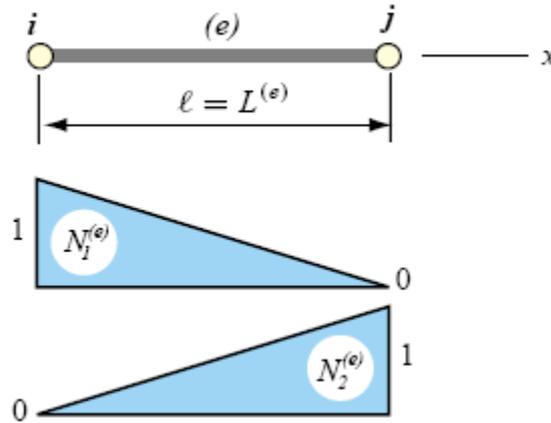


Figura 5.4: Elemento de barra 2D genérico e suas funções de interpolação

A equação (5.5) que relaciona a deformação com deslocamento pode ser reescrita na seguinte forma:

$$\varepsilon = \frac{du^{(e)}}{dx} = \left[\frac{dN_i^{(e)}}{dx} \quad \frac{dN_j^{(e)}}{dx} \right] \begin{bmatrix} u_i^{(e)} \\ u_j^{(e)} \end{bmatrix} = \frac{1}{\ell} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} u_i^{(e)} \\ u_j^{(e)} \end{bmatrix} = \mathbf{B}u^{(e)} \quad (5.18)$$

onde \mathbf{B} é chamada matriz de deformação-deslocamento.

Para o elemento de barra de dois nós, a energia interna U^e é:

$$U^e = \frac{1}{2} \int_{x_1}^{x_2} \varepsilon EA \varepsilon dx = \frac{1}{2} \int_0^1 \varepsilon EA \varepsilon \ell d\zeta \quad (5.19)$$

A deformação ε é relacionada com os deslocamentos nodais através da equação (5.18). Esta forma é simetricamente expandida pela inserção de $\varepsilon = \mathbf{B}u^e$ na segunda deformação e fazendo $\varepsilon = \varepsilon^T$ na primeira tem-se:

$$U^e = \frac{1}{2} \begin{bmatrix} u_1^e & u_2^e \end{bmatrix} \int_0^1 \frac{EA}{\ell^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \ell d\zeta \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \frac{1}{2} (u^e)^T \mathbf{K}^e u^e \quad (5.20)$$

A rigidez do elemento é dada pela equação (5.21):

$$K^e = \int_0^1 \frac{EA}{\ell^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \ell d\xi \quad (5.21)$$

Se a rigidez EA for constante sobre o elemento a matriz de rigidez do elemento pode ser escrita da seguinte maneira:

$$K^e = \frac{EA}{\ell} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.22)$$

A barra em estudo esta submetida apenas a forças discretas, e nenhuma força de corpo atua, a expressão do trabalho para o elemento de barra com dois nós fica reduzida a:

$$W^e = \begin{bmatrix} u_1^e & u_2^e \end{bmatrix} \begin{bmatrix} f_1^e \\ f_2^e \end{bmatrix} = (\mathbf{u}^e)^T \mathbf{f}^e \quad (5.23)$$

No método dos elementos finitos o processo de discretização para funcional TPE pode ser definido na seguinte forma algébrica:

$$\Pi^e = \sum_{e=1}^{nel} (U^e - W^e) \quad (5.24)$$

É importante notar que estas três energias dependem apenas da função de deslocamento nodal u^e . U^e e W^e dependem quadraticamente e linearmente da função deslocamento.

Fazendo o variacional da discretização do TPE da Equação (5.24) com respeito aos deslocamentos nodais dados na equação (5.23):

$$\Pi = \sum_{i=1}^{nel} (\delta u^e)^T \frac{\delta \Pi^e}{\delta u^e} = \sum_{i=1}^{nel} (\delta u^e)^T [K^e u^e - f^e] = 0 \quad (5.25)$$

Como o δu^e das variações pode ser arbitrário, esta quantidade deve desaparecer, produzindo:

$$K^e u^e - f^e = 0 \quad (5.26)$$

6. IMPLEMENTAÇÃO COMPUTACIONAL

Neste capítulo será apresentada a organização das estruturas computacionais que compõem este estudo. A plataforma escolhida para a programação das rotinas matemáticas foi o MATLAB R2008e.

No ambiente MATLAB foram implementados dois algoritmos de otimização, o primeiro baseado em Programação Linear Seqüencial (PLS) e o segundo utilizando Algoritmos Genéticos (AG). Também foi desenvolvido um código em elementos finitos de barras para trabalhar em conjunto com estes dois algoritmos. Os algoritmos podem trabalhar de forma individual ou conjunta e a estrutura de cada um pode ser dividida em pré-processamento, programa principal e pós-processamento. Cada etapa contém funções com finalidades distintas como mostram as figuras 6.1 e 6.2.

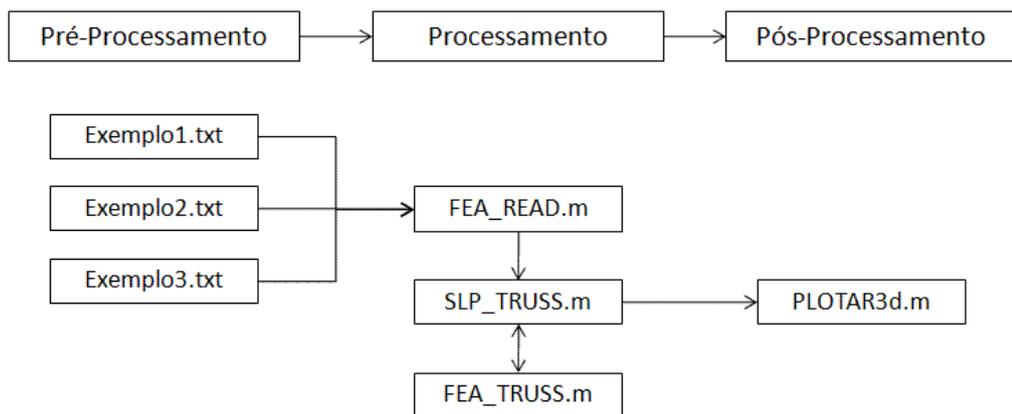


Figura 6.1: Diagrama representativo do Algoritmo PLS

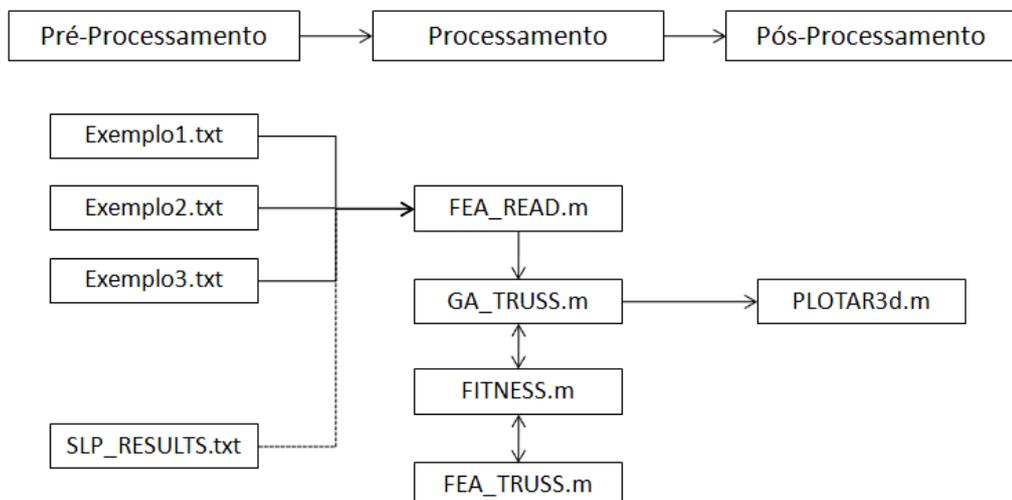


Figura 6.2: Diagrama representativo do Algoritmo Genético

Na fase de pré-processamento é definido em um arquivo do tipo ASCII a geometria do problema, condições de contorno, carregamentos, conectividades e propriedades dos elementos.

Na fase de processamento uma rotina especializada (*FEA_READ.m*) fará a leitura dos arquivos com extensão *.txt* e carregará os dados na rotina de otimização. Existem duas rotinas de otimização estrutural denominadas (*SLP_TRUSS.m*) e (*GA_TRUSS.m*). Ambas as rotinas estão conectadas ao módulo de elementos finitos (*FEA_TRUSS.m*). A diferença entre os dois fluxogramas é a existência de uma rotina responsável pela função de avaliação da aptidão (*FITNESS.m*) nos algoritmos genéticos.

No final das iterações a rotina (*PLOTA3d.m*) apresenta os resultados da convergência dos algoritmos e ainda permite visualizar a estrutura tridimensionalmente plotando as áreas ótimas.

Um fato importante de ser mencionado é a possibilidade de combinar as duas rotinas. O resultado gerado pela primeira pode ser armazenado em um arquivo que será posteriormente lido pela segunda rotina. Neste caso, o Algoritmo Genético usará esta informação para definir uma população inicial não aleatória.

A seguir será apresentada a formulação matemática implementada no MATLAB. A grande dificuldade neste tipo problema deve-se ao fato de as restrições serem funções não-lineares em relação às várias de projeto A_i .

O problema de otimização de treliças pode ser formulado da seguinte maneira [Souza e Fonseca, 2008]:

$$\text{minimizar} \quad W = \sum_{i=1}^{nel} \rho_i L_i A_i \quad (6.1)$$

$$\begin{aligned} \text{sujeito a:} \quad & \sigma_c^i \leq \sigma_i \leq \sigma_T^i & i=1, \dots, nel \\ & U_i^{\min} \leq U_i \leq U_i^{\max} & i=1, \dots, nglr \\ & \sigma_i \geq \sigma_i^F & i=1, \dots, nel \\ & A_i^{\min} \leq A_i \leq A_i^{\max} & i=1, \dots, nap \end{aligned}$$

onde as três primeiras restrições são de desigualdade e a última é lateral.

Na expressão (6.1) têm-se:

A_i = Área da barra i ,

A_i^{max} = Área da seção transversal máxima admissível para a barra i ,

A_i^{min} = Área da seção transversal mínima admissível para a barra i ,

L_i = Comprimento da barra i ,

nap = Número de áreas de projeto,

nel = Número de elementos,

U_i = Deslocamento nodal do grau de liberdade i ,

U_i^{max} = Deslocamento nodal máximo do grau de liberdade i ,

U_i^{min} = Deslocamento nodal mínimo do grau de liberdade i ,

ρ_i = massa específica do material da barra i ,

σ_i = Tensão normal atuante na barra i ,

σ_i^C = Tensão admissível de compressão na barra i ,

σ_i^F = Tensão admissível de Euler,

σ_i^T = Tensão admissível de tração na barra i ,

A função objetivo é linear e por isso não precisa ser expandida em séries de Taylor, entretanto, as equações das restrições são não-lineares e precisam ser linearizadas para sua utilização no algoritmo de programação linear. A restrição da tensão expandida em séries de Taylor pode ser reescrita da seguinte forma:

$$\sigma_C^i \leq \sigma_0^i + \sum_{j=1}^{nap} \frac{\partial \sigma}{\partial A_j} (A_j - A_j^0) \leq \sigma_T^i \quad (6.2)$$

Expandido a restrição de deslocamento obtém-se:

$$U_{\min}^i \leq \sigma_0^i + \sum_{j=1}^{nap} \frac{\partial U_i}{\partial A_j} (A_j - A_j^0) \leq U_{\max}^i \quad (6.3)$$

A restrição de flambagem pode ser reescrita na seguinte forma:

$$\sigma_0^i + \sum_{j=1}^{nap} \frac{\partial \sigma_i}{\partial A_j} (A_j - A_j^0) \geq \sigma_F^i + \sum_{j=1}^{nap} \frac{\partial \sigma_F^i}{\partial A_j} (A_j - A_j^0) \quad (6.4)$$

A próxima etapa consiste em determinar os gradientes das restrições em relação às variáveis de projeto A_j . Esta etapa é conhecida como análise de sensibilidade e estes gradientes indicam a sensibilidade da resposta das funções das restrições a pequenas mudanças.

A análise de sensibilidade é a etapa principal e mais demorada no processo de otimização, consumindo grande esforço computacional na maioria das vezes. Erros na precisão do cálculo do método fatalmente levam a problemas de convergência nos algoritmos de otimização. Portanto, na escolha do método a ser utilizado para o cálculo dos gradientes devem ser considerados dois aspectos básicos: precisão e eficiência.

Adota-se o método de diferenciação direta. Entretanto, o cálculo da derivada em relação às variáveis de projeto não é simples requerendo o uso da regra da cadeia.

6.1 Determinação da derivada da restrição de flexibilidade

A equação de equilíbrio para estruturas com comportamento linear estático (treliças) é obtida através do método dos elementos finitos:

$$u = K^{-1} f \quad (6.5)$$

onde K é a matriz de rigidez, u é vetor dos deslocamentos nodais e f é o vetor das cargas externas.

Derivando implicitamente a equação (6.5) em relação à variável de projeto A_j , têm-se:

$$\frac{du}{dA_j} = \frac{dK^{-1}}{dA_j} f \quad (6.6)$$

Ao aplicar na matriz de rigidez a propriedade básica da álgebra que diz que uma matriz multiplicada pela sua inversa resulta em uma matriz identidade de mesma ordem, obtém-se:

$$KK^{-1} = I \quad (6.7)$$

Derivando ambos os lados da equação (6.7) em relação à A_j , têm-se:

$$\frac{d(KK^{-1})}{dA_j} = \frac{dI}{dA_j} \quad (6.8)$$

Aplicando a regra da cadeia na equação (6.8):

$$K \frac{dK^{-1}}{dA_j} + \frac{dK}{dA_j} K^{-1} = 0 \quad (6.9)$$

Isolando-se o termo $\frac{dK^{-1}}{dA_j}$, obtém-se a derivada desejada:

$$\frac{dK^{-1}}{dA_j} = K^{-1} \frac{dK}{dA_j} K^{-1} \quad (6.10)$$

Substituindo a expressão (6.10) em (6.6), o cálculo da derivada da matriz inversa torna-se desnecessário:

$$\frac{du}{dA_j} = K^{-1} \frac{dK}{dA_j} u \quad (6.11)$$

A derivada da matriz de rigidez é calculada através da soma das contribuições da matriz de rigidez de cada elemento.

$$\frac{dK}{dA_j} = \sum_{i=1}^{nel} [T_i]^T \frac{d[K_i]}{dA_j} [T_i] \quad (6.12)$$

onde $[T_i]$ é a matriz dos cossenos diretores relativos ao i -ésimo elemento da barra e esta matriz têm a seguinte forma:

$$[T_i] = \begin{bmatrix} nx_i & ny_i & nz_i & 0 & 0 & 0 \\ 0 & 0 & 0 & nx_i & ny_i & nz_i \end{bmatrix} \quad (6.13)$$

sendo:

$$nx_i = (X_2 - X_1)_i / L_i$$

$$ny_i = (Y_2 - Y_1)_i / L_i$$

$$nz_i = (Z_2 - Z_1)_i / L_i$$

6.2 Determinação da derivada da restrição de tensão

Esta restrição é fundamental em problemas de otimização estrutural, pois quase todas as estruturas são projetadas para suportar carregamentos sem ultrapassar a tensão admissível em qualquer ponto da estrutura. Uma característica importante deste problema é o fato dele ser um problema “local”, ou seja, cada ponto da estrutura deve ter seu limite de tensão controlado. Por este motivo, os problemas envolvendo tensão geralmente apresentam um grande número de restrições, sendo às vezes difíceis de manusear.

Para o elemento de barra a relação tensão versus deformação é dada pela seguinte relação:

$$\sigma_{xx} = E\varepsilon_{xx} \quad (6.14)$$

O deslocamento pode ser relacionado com a deformação através da expressão:

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} = \frac{\Delta L}{L} = \frac{u_j - u_i}{L} \quad (6.15)$$

onde os índices i e j são os nós finais e iniciais do elemento de barra.

Deste modo, a equação (6.14) pode ser reescrita da seguinte forma:

$$\sigma_{xx} = E \frac{u_j - u_i}{L} \quad (6.16)$$

Expandido a expressão da tensão em série de Taylor e truncando no termo linear, obtém-se:

$$\sigma = \sigma_0 + \frac{\partial \sigma}{\partial A_j} (A_j - A_j^0) \quad (6.17)$$

A expressão $\frac{\partial \sigma}{\partial A_j}$ pode ser encontrada derivando a equação (6.16) em relação à variável de projeto:

$$\frac{\partial \sigma_k}{\partial A_j} = \frac{E}{L_k} \left(\frac{\partial u_j}{\partial A_j} - \frac{\partial u_l}{\partial A_j} \right) \quad (6.18)$$

onde o σ_k é uma medida de tensão equivalente em um elemento relativo ao caso k. Introduzindo a constante B_k a expressão (6.18) pode ser reescrita:

$$\frac{\partial \sigma_k}{\partial A_j} = \frac{E}{L_k} \frac{\partial}{\partial A_j} (B_k u_k) \quad (6.19)$$

$$B_k = [1 \quad -1]$$

$$u_k = \begin{bmatrix} u_j \\ u_l \end{bmatrix}$$

Desta forma, a única derivada que precisa ser determinada é a $\frac{\partial u}{\partial A_j}$ definida anteriormente na equação (6.11).

6.3 Determinação da derivada da restrição de flambagem

A tensão de flambagem local para uma barra é conhecida pela expressão:

$$\sigma_F^i = \frac{\pi^2 E_i I_i}{A_i L_i^2} \quad (6.20)$$

onde I_i é o momento de inércia da seção e E_i é o módulo de elasticidade do material. Pode-se definir o coeficiente K:

$$K = \frac{\pi^2 I_i}{A_i^2} \quad (6.21)$$

A equação (6.20) pode ser reescrita na seguinte forma:

$$\sigma_F^i = \frac{KE_i A_i}{L_i^2} \quad (6.22)$$

Derivando a equação (6.22) em relação a A_j obtém-se:

$$\frac{d\sigma_F^i}{dA_i} = \frac{KE_i}{L_i^2} \frac{\partial A_i}{\partial A_j} = \frac{KE_i}{L_i^2} \delta_{ij} \quad (6.23)$$

6.4 Determinação da Função de Aptidão (Fitness)

A função aptidão (a ser minimizada) implementada no Algoritmo Genético desenvolvido no presente trabalho é a mesma usada no trabalho de [Lemonge e Barbosa, 2000] e é dada pela equação (6.24):

$$f_p(x) = f(x) + C \left(\sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{\max}} - 1 \right]^+ \right)^2 + \sum_j \left(\left[\frac{|u_j|}{u_{\max}} - 1 \right]^+ \right)^2 \right) \quad (6.24)$$

onde C é o coeficiente de penalização da função objetivo, e $[x]^+ = x$ se $x > 0$.

7. APLICAÇÃO EM PROBLEMAS DE OTIMIZAÇÃO ESTRUTURAL

7.1 Introdução

Três problemas de otimização estrutural são apresentados neste capítulo para ilustrar a metodologia e os algoritmos discutidos nos capítulos anteriores. Os exemplos servem como teste padrão e têm sido estudados extensamente por diversos pesquisadores na literatura [Haftka e Gürdal, 1992; Rajeev e Krishnamoorthy; 1992; Kripka 2004]. Os problemas aqui estudados irão permitir demonstrar a viabilidade da metodologia proposta e a eficiência do uso de algoritmos baseados em Programação Linear Seqüencial (PLS) e Algoritmos Genéticos (AG) na solução de problemas de otimização estrutural.

Em todos os casos deseja-se minimizar a massa das treliças, sendo que a tensão axial e a deflexão de cada uma das barras devem ficar dentro de limites estabelecidos. As variáveis de projeto serão as áreas das seções transversais das barras.

Os testes computacionais foram realizados em um microcomputador com processador Intel® Core®2 Duo T7250 2.0 GHz e 2038 MB de RAM.

A eficiência dos métodos de otimização foi avaliada em termos do número de iterações e tempo de que cada algoritmo gastou para atingir o critério de parada.

7.2 Casos estudados:

Cada um dos problemas utilizados no presente trabalho foi resolvido empregando três abordagens diferentes:

1. Caso 1: Os problemas são resolvidos considerando as variáveis de projetos como contínuas e utilizando um algoritmo de Programação Linear Seqüencial (PLS) baseado nos trabalhos de [Haftka e Gürdal,1992; Cheng, 1992].
2. Caso 2: Na segunda abordagem, as variáveis de projeto são tratadas como variáveis discretas, e os problemas são resolvidos utilizando Algoritmos Genéticos (AG) [Goldberg, 1989] com a geração da população inicial de forma aleatória.

3. Caso 3: Roda-se o problema de otimização utilizando novamente a rotina baseada em AG. No entanto, é introduzida uma população inicial composta pelas soluções encontradas no caso 1. Desta forma, espera-se acelerar o tempo de convergência do algoritmo.

7.3 Treliza de 10 barras

A treliça de 10 barras mostrada na figura 7.1, tem sido estudada por diversos pesquisadores, entre eles [Cai e Thierut, 1993; Rajeev e Krishnamoorthy, 1992] para demonstrar a validade de novos algoritmos.

A mínima massa da estrutura é obtida variando as seções transversais das barras. A treliça é submetida a restrições de tensão e deslocamento. Na solução do problema utilizando Programação Linear Seqüencial, as áreas das seções transversais das barras, são representadas por variáveis contínuas e deverão ser limitadas a um valor mínimo de $0,1 \text{ in}^2$ ($64,5 \text{ mm}^2$) para permitir que os elementos não desapareçam durante o processo de otimização. No Algoritmo Genético as variáveis de projeto são valores discretos e foram obtidos a partir do manual do Instituto Americano de Construção em Aço AISC (American Institute of Steel Construction). Os carregamentos aplicados e as propriedades dos materiais são apresentados nas tabelas 7.1 e 7.2, respectivamente.

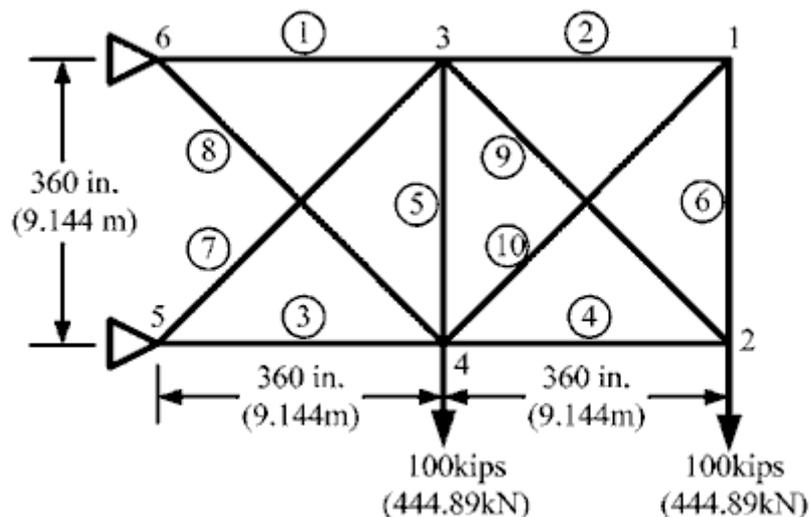


Figura 7.1: Treliza de 10 barras

Tabela 7.1: Treliça de 10 barras - Carregamentos aplicados

Número Nó	Px (Kips)	Py (Kips)
2	0	-100 (-444,89 kN)
4	0	-100 (-444,89 kN)

Tabela 7.2: Treliça de 10 barras - Propriedades

Módulo de elasticidade	$E=1,0 \times 10^7$ psi (68950 MPa)
Densidade	$\rho = 0,10$ lb/in ³ (2767 kg/m ³)
Tensão admissível	± 25000 psi (± 172 MPa)
Deslocamentos	$\pm 2,0$ in (50,8 mm)

7.3.1 Resultados empregando Programação Linear Sequencial (PLS)

Na tabela 7.3 são apresentados os valores das variáveis de projeto encontradas ao final do processo de otimização. A evolução da função objetivo com o número de iterações é mostrada na figura 7.2, enquanto a geometria final da treliça de 10 barras pode ser visualizada na figura 7.3.

A solução detalhada deste problema pode ser encontrada em [Haftka e Gürdal,1992].

Tabela 7.3: Treliça de 10 barras - Comparativo entre as áreas obtidas com o PLS

Método	Massa lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)	A9 in ² (mm ²)	A10 in ² (mm ²)
Área inicial	2900 (1315)	5,0 (3226)									
Área final	5037 (2284)	31,37 (20240)	0,1 (65)	23,24 (14994)	14,34 (9252)	0,1 (65)	0,1 (65)	8,56 (5523)	20,34 (13123)	20,94 (13510)	0,1 (65)
Haftka	5061 (2295)	30,52 (19692)	0,1 (65)	23,2 (14969)	15,22 (9820)	0,1 (65)	0,55 (355)	7,46 (4813)	21,04 (13575)	21,53 (13891)	0,1 (65)

Como se pode observar na tabela 7.3, os valores das variáveis de projeto e da função objetivo encontrados ficaram próximos aos encontrados por Haftka e Gürdal.

Tabela 7.4 : Treliça de 10 barras - Deslocamentos obtidos ao final da otimização com SLP

Número Nó	Deslocamento em x – in (mm)	Deslocamento em y - in (mm)
1	0,26 (6,60)	-2,00 (-50,80)
2	- 0,50 (-12,7)	-2,00 (-50,80)
3	0,25 (6,35)	-0,72 (-18,28)
4	- 0,27 (6,85)	-1,47 (-37,34)
5	0,0	0,0
6	0,0	0,0

Como se pode verificar na tabela 7.4 as restrições de deslocamento no nó número um e dois na direção y encontram-se ativas. Para este exemplo as restrições de tensões não foram ativadas e os valores encontrados para as tensões nas barras ficaram abaixo dos valores admissíveis para este exemplo.

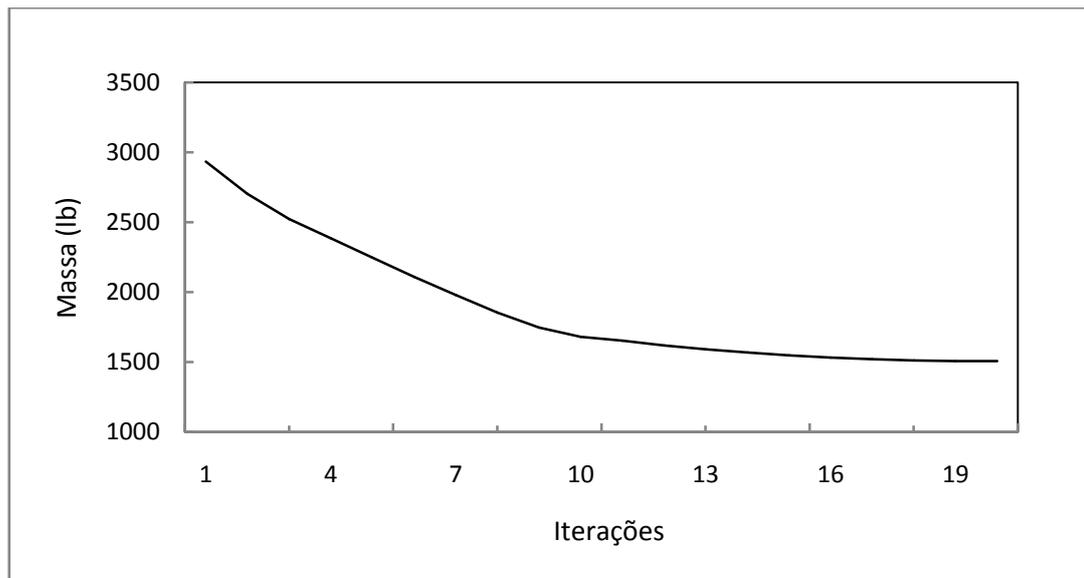


Figura 7.2: Gráfico da convergência do PLS para a treliça de 10 barras

A figura 7.2 mostra o valor final da função objetivo após a realização de 20 iterações em cerca de 5 segundos.

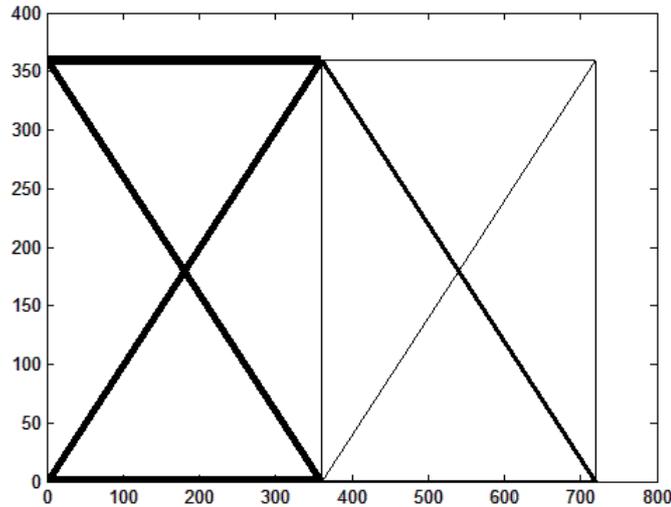


Figura 7.3: Geometria final treliça plana de 10 barras deformada

Na figura 7.3 é apresentada a geometria final da estrutura deformada através da aplicação dos carregamentos propostos.

7.3.2 Resultados empregando Algoritmos Genéticos (AG)

Na tabela 7.5 são apresentados os resultados encontrados para o problema de otimização da treliça de 10 barras usando Algoritmos Genéticos. Os resultados são comparados com os obtidos por diversos pesquisadores. Os valores discretos correspondentes às áreas das seções transversais das barras, que serão atribuídos às variáveis de projeto, podem ser escolhidos a partir do Manual AISC. Foram considerados os seguintes valores: 1,62; 1,80; 1,99; 2,13; 2,38; 2,62; 2,63; 2,88; 2,93; 3,09; 3,13; 3,38; 3,47; 3,55; 3,63; 3,84; 3,87; 3,88; 4,18; 4,22; 4,49; 4,59; 4,80; 4,97; 5,12; 5,74; 7,22; 7,97; 11,50; 13,50; 13,90; 14,20; 15,50; 16,00; 16,90; 18,80; 19,90; 22,00; 22,90; 26,50; 30,00; 33,50 (in²).

Configuração do AG utilizado:

- População inicial: aleatória
- Tamanho da população: 300
- Número máximo de iterações: 100
- Probabilidade de mutação Pm: 0,01
- Coeficiente de penalidade: $1,0 \times 10^8$

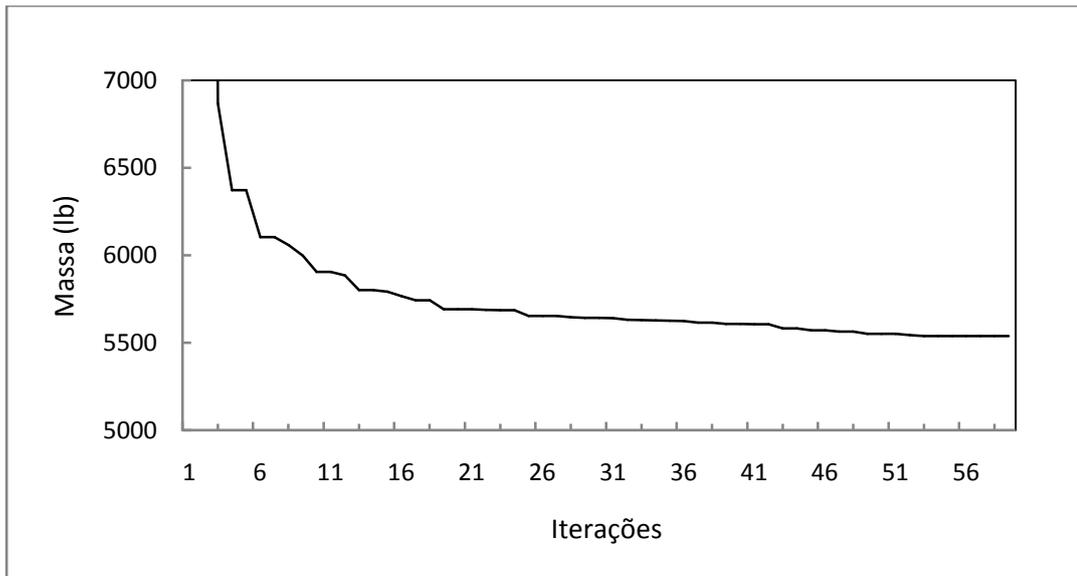


Figura 7.4: Gráfico da convergência do AG para treliça de 10 barras

A figura 7.4 mostra o histórico da convergência da função objetivo, resultando em 59 iterações em cerca de 40 segundos.

Tabela 7.5: Treliça de 10 barras - Comparativo entre as áreas obtidas com AG

Método	Massa lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)	A9 in ² (mm ²)	A10 in ² (mm ²)
1	5491,7 (2491)	33,5 (21613)	1,62 (1045)	22,9 (14774)	15,5 (10000)	1,62 (1045)	1,62 (1045)	22 (14193)	7,97 (5141)	1,62 (1045)	22 (14193)
2	5613,8 (2546)	33,5 (21613)	1,62 (1045)	22 (14193)	15,5 (10000)	1,62 (1045)	1,62 (1045)	19,9 (12839)	14,2 (9161)	2,62 (1690)	19 (12258)
3	5491,7 (2491)	33,5 (21613)	1,62 (1045)	22,9 (14774)	15,5 (10000)	1,62 (1045)	1,62 (1045)	22 (14193)	7,92 (5109)	1,62 (1045)	22 (14193)
4	5490,7 (2490)	33,5 (21613)	1,62 (1045)	22,9 (14774)	14,2 (9161)	1,62 (1045)	1,62 (1045)	22 (14193)	13,9 (8968)	1,62 (1045)	22 (14193)
5	5538,1 (2512)	33 (21290)	1,62 (1045)	22,9 (14774)	13,5 (8710)	1,62 (1045)	1,62 (1045)	22,9 (14774)	11,5 (7419)	1,99 (1284)	22 (14193)

onde:

- 1- Método da função da penalidade melhorada (Cai and Thiereu, 1993)
- 2- Algoritmo Genético (Rajeev and Krishnamoorthy, 1992)
- 3- Algoritmo Genético (Coello, 1994)
- 4- Método do Recozimento Simulado (Kripka, 2004)
- 5- Presente trabalho

Na tabela 7.5, apresenta-se os valores das variáveis de projeto e da função objetivo encontrados ao final do processo de otimização empregando o algoritmo genético proposto. Os resultados obtidos mostraram-se satisfatórios quando comparado com os valores encontrados pelos autores citados acima. Entretanto, ao comparar o custo computacional deste algoritmo com o SLP, exemplo 7.3.1, pode-se concluir que o presente algoritmo apresenta uma clara desvantagem, levando um tempo maior para encontrar o ponto ótimo.

Tabela 7.6: Treliça de 10 barras - Deslocamentos obtidos no final da otimização com AG

Número Nó	Deslocamento em x – in (mm)	Deslocamento em y - in (mm)
1	0,39 (9,90)	-1,92 (-48,76)
2	- 0,51 (-12,25)	-2,00 (-50,80)
3	0,25 (6,35)	-0,82 (-20,82)
4	- 0,26 (6,60)	-1,10 (-27,94)
5	0,0	0,0
6	0,0	0,0

Como se pode verificar na tabela 7.6 a restrição de deslocamento no nó número dois na direção y encontra-se ativa. Para este exemplo as restrições de tensões não foram ativadas e os valores encontrados para as tensões nas barras ficaram abaixo dos valores admissíveis para o exemplo.

7.3.3 Resultados empregando Algoritmos Genéticos com utilização de população inicial

O problema de minimização da massa da treliça de 10 barras apresentado na seção 7.3.2 é novamente resolvido, mas agora, introduz-se uma população inicial no problema.

Os indivíduos usados para formar a população inicial são os valores obtidos na solução do problema empregando o PLS, seção 7.3.1. Entretanto, estes elementos não coincidem com os valores do AISC, e por isso foram selecionados valores próximos aos obtidos para gerar a população inicial do problema.

Os resultados obtidos são comparados aos encontrados na seção 7.3.2 e podem ser visualizados na figura 7.5:

Configuração do AG utilizado:

População inicial: 7,97; 0,01; 7,97; 3,88; 0,01; 0,01; 5,74; 5,74; 3,88; 0,01

Tamanho da população: 300

Número máximo de iterações: 100

Probabilidade de mutação Pm: 0,01

Coefficiente de penalidade: $1,0 \times 10^8$

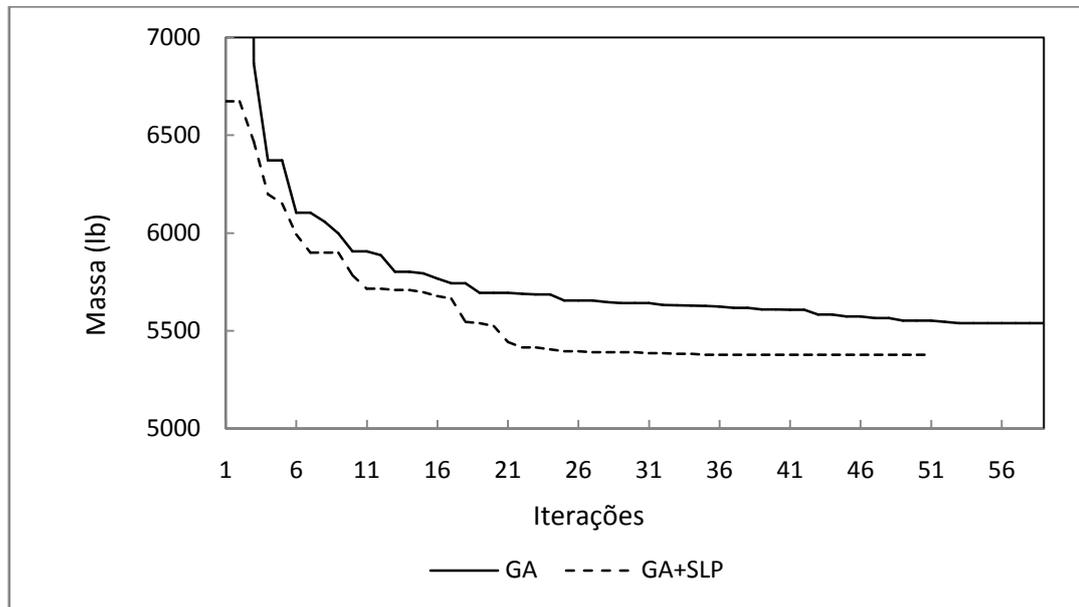


Figura 7.5: Gráfico da convergência do PLS+AG e AG para a treliça de 10 barras

A figura 7.5 mostra uma ligeira redução do valor da função objetivo conseguido com a introdução de uma população inicial não aleatória. Entretanto, o ganho significativo com a metodologia proposta foi no tempo computacional, que é reduzido de 59 segundos para 31 segundos neste exemplo.

Tabela 7.7: Treliça de 10 barras - Comparativo dos resultados obtidos com PLS+AG e AG

Método	Massa lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)	A9 in ² (mm ²)	A10 in ² (mm ²)
AG	5538 (2512)	33,0 (21290)	1,62 (1045)	22,9 (14774)	13,5 (8710)	1,62 (1045)	1,62 (1045)	22,9 (14774)	11,5 (7419)	1,99 (1284)	22,0 (14193)
SLP+ AG	5377 (2439)	30,0 (19355)	2,93 (1890)	22,9 (14774)	13,9 (8968)	1,62 (1045)	1,80 (1161)	19,9 (12839)	11,5 (7419)	3,63 (2342)	18,8 (12129)

Na tabela 7.7, apresenta-se os valores das variáveis de projeto e da função objetivo encontrados ao final do processo de otimização utilizando algoritmo genético com população inicial não aleatória (SLP+AG) e algoritmo genético com população inicial aleatória (AG). A utilização desta metodologia permitiu uma redução de 3% na massa total da estrutura.

7.4 Treliça espacial de 25 barras

A treliça de 25 barras mostrada na figura 7.6, foi projetada para suportar o carregamento apresentado na tabela 7.8. As propriedades do material são dadas na tabela 7.10. A otimização deve ser realizada mantendo-se a relação de simetria da estrutura em relação aos planos y-z e x-z. Por esta razão, oito variáveis de projeto são usadas para dimensionar os 25 elementos da treliça, conforme ilustrado na tabela 7.9. A tabela 7.11 mostra a tensão admissível de flambagem em cada um dos elementos da estrutura.

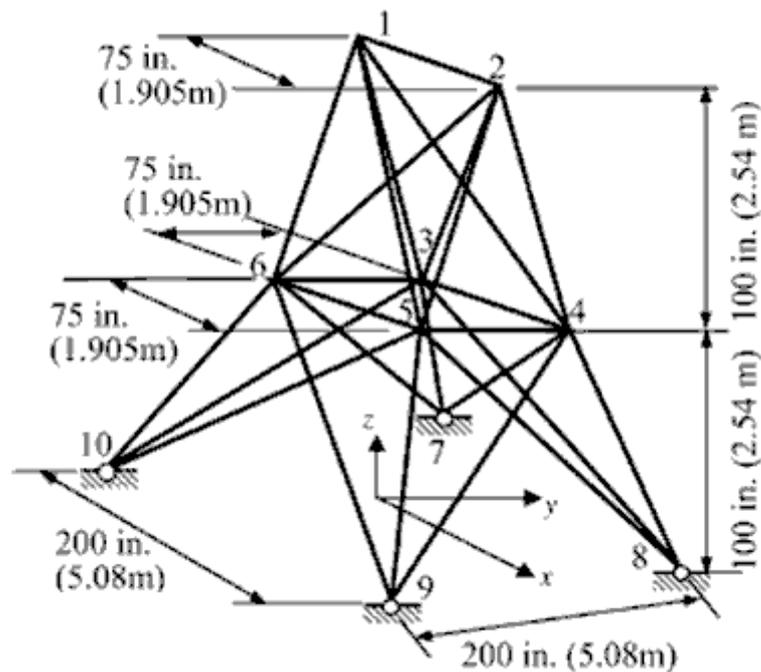


Figura 7.6: Treliça de 25 barras

Tabela 7.8: Treliça de 25 barras - Carregamentos aplicados

Número Nó	Px (kips)	Py (kips)	Pz (kips)
1	1,0 (4,454 kN)	-10 (-44,53 kN)	-10 (-44,53 kN)
2	0	-10 (-44,53 kN)	-10 (-44,53 kN)
3	0,5 (2,227 kN)	0	0
4	0,6 (2.672 kN)	0	0

Tabela 7.9: Treliça de 25 barras - Variáveis de projeto

Variáveis de projeto	Conectividades dos nós
1	(1,2)
2	(1,4),(1,5),(2,3),(2,6)
3	(1,3),(1,6),(2,4),(2,5)
4	(3,6),(4,5)
5	(3,4),(5,6)
6	(3,10),(4,9),(5,8),(6,7)
7	(3,8),(4,7),(5,10),(6,9)
8	(3,7),(4,8),(5,9),(6,10)

Tabela 7.10: Treliça de 25 barras - Propriedades

Módulo de elasticidade	$E=1 \times 10^7$ psi (68950 MPa)
Densidade	$\rho = 0.10$ lb/in ³ (2767 kg/m ³)
Tensão admissível	± 40000 psi (257,8 MPa)
Deslocamento admissível	$\pm 0,35$ in (8,89 mm)

Tabela 7.11: Treliça de 25 barras - Tensão de flambagem em cada elemento da estrutura

Variáveis de projeto	Cargas de flambagem
1	-35092 psi (-226MPa)
2	-11590 psi (-75 MPa)
3	-17305 psi (-111 MPa)
4	-35092 psi (-226MPa)
5	-35092 psi (-226MPa)
6	-6759 psi (-44 MPa)
7	-6959 psi (-45 MPa)
8	-11082 psi (-71 MPa)

7.4.1 Resultados empregando Programação Linear Sequencial (PLS)

Os resultados encontrados para o projeto final da estrutura usando PLS são mostrados na tabela 7.12. Nesta tabela os resultados são comparados aos obtidos por [Haftka e Gürdal,1992]. A geometria final é apresentada na figura 7.8 e o gráfico de convergência da função objetivo é mostrado na figura 7.7.

Tabela 7.12: Treliza de 25 barras - Comparativo entre resultados obtidos com PLS

Método	Massa lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)
Área inicial	3307,20 (1500)	10 (6500)							
Área final	498,78 (226)	0,01 (6,5)	0,192 (123,9)	3,59 (2316,1)	0,01 (6,5)	1,27 (819,4)	0,819 (528,4)	0,846 (545,8)	3,66 (2361,3)
Haftka	542,22 (246)	0,01 (6,5)	1,987 (1281,9)	2,991 (1929,7)	0,01 (6,5)	0,012 (7,7)	0,683 (440,6)	1,679 (1083,2)	2,664 (1718,7)

Pode-se observar na tabela 7.12 que, após a otimização, houve uma redução da função objetivo de 84,9 %. Os resultados encontrados ficaram próximos aos valores encontrados por Haftka e Gürdal.

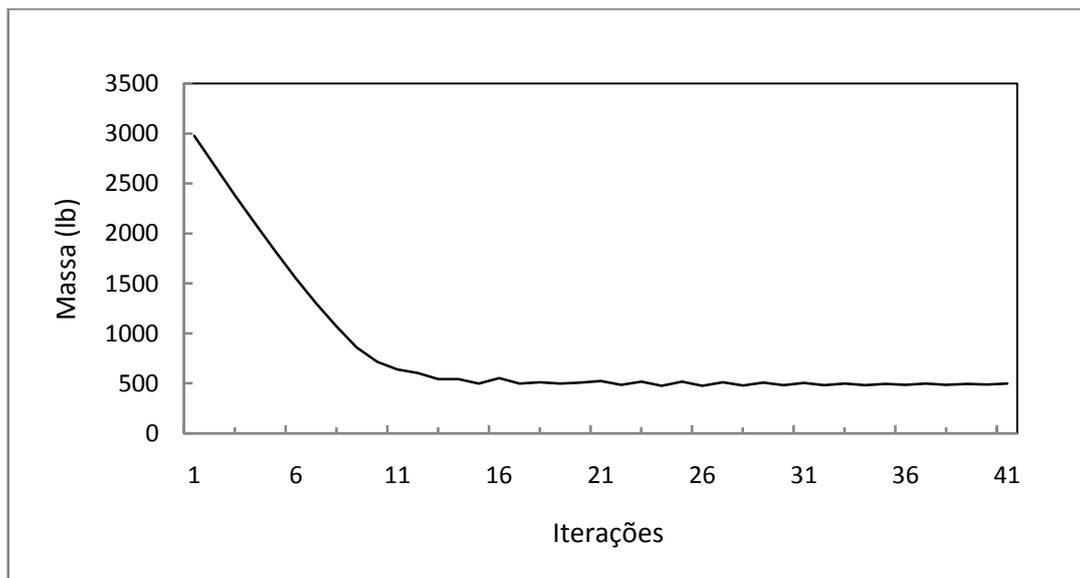


Figura 7.7: Gráfico da convergência do SLP para treliza de 25 barras

A figura 7.7 mostra o histórico da convergência da função objetivo, neste exemplo o algoritmo encontrou a solução ótima após 41 iterações em cerca de 16 segundos.

Tabela 7.13: Treliça de 25 barras - Deslocamentos obtidos no final da otimização com SLP

Número Nó	Desloc. em x (in)	Desloc. em y (in)	Desloc. em z (in)
1	0,09	-0,34	-0,04
2	0,04	-0,35	-0,05
3	0,00	0,02	0,06
4	0,02	0,02	0,06
5	-0,02	0,04	-0,14
6	0,02	0,04	-0,13
7	0,00	0,00	0,00
8	0,00	0,00	0,00
9	0,00	0,00	0,00
10	0,00	0,00	0,00

Observa-se na tabela 7.13 que a restrição de deslocamento máximo no nó número dois na direção y se faz ativa no final da otimização. As restrições de tensões e flambagem não foram ativadas neste exemplo, pois os valores não alcançaram os limites definidos na tabelas 7.10 e 7.11.

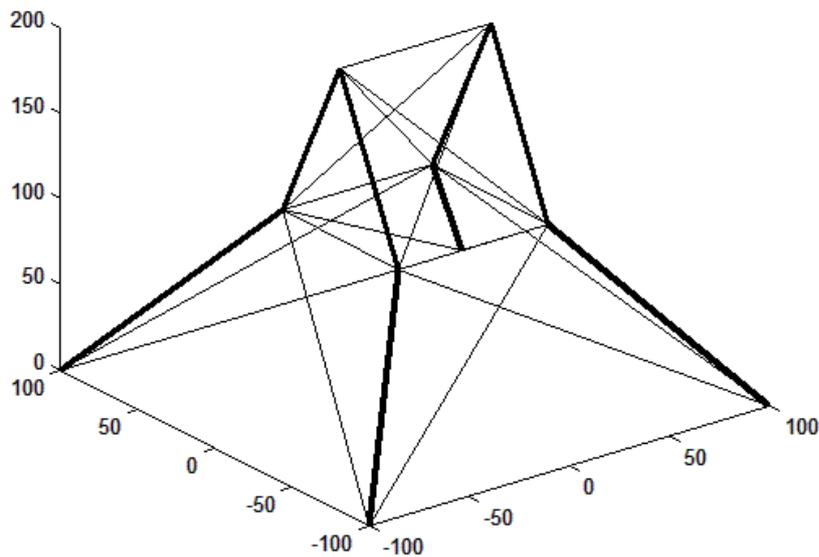


Figura 7.8: Geometria final da treliça espacial de 25 barras

Na figura 7.8 é apresentada a geometria final da estrutura ao final do processo de otimização.

7.4.2 Resultados empregando Algoritmos Genéticos (AG)

Neste exemplo o problema da minimização da massa da treliça de 25 barras é resolvido, utilizando Algoritmos Genéticos. Os valores discretos, correspondentes as áreas das seções transversais das barras, podem ser obtidos a partir da seguinte lista de perfis disponíveis: 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0; 1,1; 1,2; 1,3; 1,4; 1,5; 1,6; 1,7; 1,8; 1,9; 2,0; 2,1; 2,2; 2,3; 2,4; 2,5; 2,6; 2,8; 3,0; 3,2; e 3,4 (in²).

Os resultados encontrados são mostrados na tabela 7.14, bem como as comparações com os de outros autores. A figura 7.9 mostra a curva do processo de convergência do problema de otimização discreta da treliça.

Configuração do AG utilizado:

- População inicial: aleatória
- Tamanho da população: 300
- Número máximo de iterações: 100
- Probabilidade de mutação Pm: 0,01
- Coeficiente de penalidade: $1,0 \times 10^5$

Tabela 7.14: Treliça de 25 barras - Comparativo entre resultados obtidos com AG

Método	Peso lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)
1	487,41 (221,1)	0,1 (64,5)	0,1 (64,5)	3,4 (2193,5)	0,1 (64,5)	2,0 (1290,3)	1,0 (645,2)	0,7 (451,6)	3,4 (2193,5)
2	546,01 (247,7)	0,1 (64,5)	1,8 (1161,3)	2,3 (1483,9)	0,2 (129)	0,1 (64,5)	0,8 (516,1)	1,8 (1161,3)	3,0 (1935,5)
3	539,78 (244,8)	1,5 (967,7)	0,7 (451,6)	3,4 (2193,5)	0,7 (451,6)	0,4 (258,1)	0,7 (451,6)	1,5 (967,7)	3,2 (2064,5)
4	484,33 (219,7)	0,1 (64,5)	0,4 (258,1)	3,4 (2193,5)	0,1 (64,5)	2,2 (1419,4)	1,0 (645,2)	0,4 (258,1)	3,4 (2193,5)
5	495,2 (224,6)	0,1 (64,5)	0,6 (387,1)	3,2 (2064,5)	0,1 (64,5)	0,8 (516,1)	0,9 (580,6)	0,9 (580,6)	3,4 (2193,5)

- 1- Método da função da penalidade melhorada (Cai and Thiereu, 1993)
- 2- Algoritmo Genético (Rajeev and Krishnamoorthy, 1992)
- 3- Algoritmo Genético (Coello, 1994)
- 4- Método do Recozimento Simulado (Kripka, 2004)
- 5- Presente trabalho

Na tabela 7.15, é possível notar que o algoritmo genético empregado nesta dissertação conseguiu resultados compatíveis em relação aos resultados encontrados pelos autores citados acima.

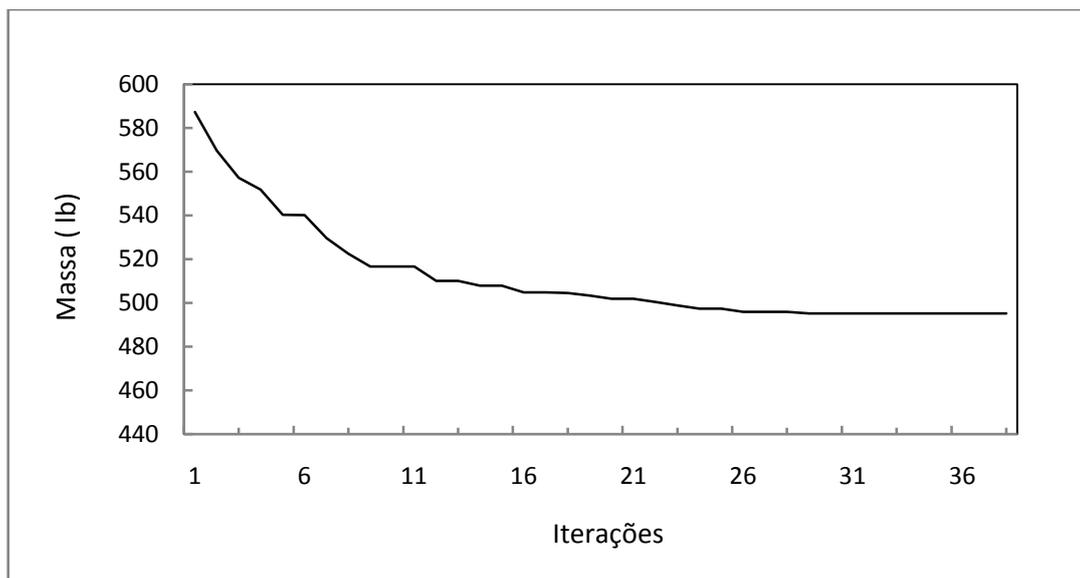


Figura 7.9: Gráfico da convergência do AG para a treliça de 25 barras

A figura 7.9 mostra a curva de convergência da função objetivo para a otimização da treliça de 25 barras empregando algoritmos genéticos. Este algoritmo encontrou a solução ótima após 38 iterações em cerca de 128 segundos. Ao comparar o custo computacional deste algoritmo com o SLP, exemplo 7.4.1, pode-se concluir que o algoritmo necessita de um tempo maior para encontrar o ponto ótimo.

Tabela 7.15: Treliça de 25 barras - Deslocamentos obtidos no final da otimização com AG

Número Nó	Desloc. em x (in)	Desloc. em y (in)	Desloc. em z (in)
1	-0,01	-0,35	-0,03
2	0,01	-0,35	-0,03
3	0,01	0,01	0,07
4	0,01	0,03	0,07
5	-0,02	0,02	-0,12
6	0,01	0,00	-0,10
7	0,00	0,00	0,00
8	0,00	0,00	0,00
9	0,00	0,00	0,00
10	0,00	0,00	0,00

Observa-se na tabela 7.15 que a restrição de deslocamento máximo no nó número um e dois estão ativas na direção y. Da mesma forma do exemplo 7.4.1, as restrições de tensões e flambagem não se encontram ativas.

7.4.3 Resultados empregando Algoritmos Genéticos com utilização de população inicial

O problema descrito na seção 7.4.2 é novamente resolvido. Porém, desta vez é definida uma população inicial empregando os resultados encontrados na seção 7.4.1.

Os resultados obtidos para as variáveis de projeto são mostrados na tabela 7.11. O histórico da convergência da função objetivo é disposto na figura 7.10.

Configuração do AG utilizado:

- População inicial: 0,10; 0,2; 3,4; 0,10; 1,30; 0,80; 0,80; 3,40 (in²)
- Tamanho da população: 300
- Número máximo de iterações: 100
- Probabilidade de mutação P_m: 0,01
- Coeficiente de penalidade: 1,0 x 10⁵

Tabela 7.16: Treliça de 25 barras - Comparativo resultados obtidos com PLS+AG e AG

Método	Peso lb (kg)	A1 in ² (mm ²)	A2 in ² (mm ²)	A3 in ² (mm ²)	A4 in ² (mm ²)	A5 in ² (mm ²)	A6 in ² (mm ²)	A7 in ² (mm ²)	A8 in ² (mm ²)
AG	495,2 (224,6)	0,1 (64,5)	0,6 (387,1)	3,2 (2064,5)	0,1 (64,5)	0,8 (516,1)	0,9 (580,6)	0,9 (580,6)	3,4 (2193,5)
SLP+AG	485,5 (220,2)	0,1 (64,5)	0,2 (129)	3,4 (2193,5)	0,1 (64,5)	1,5 (967,7)	0,9 (580,6)	0,8 (516,1)	3,4 (2193,5)

Na comparação dos resultados utilizando algoritmo genético com população inicial não aleatória (SLP+AG) e algoritmo genético com população inicial aleatória (AG), tabela 7.16, o algoritmo com a população inicial não aleatória obteve o melhor resultado, com 1,95% de diferença a menos na massa.

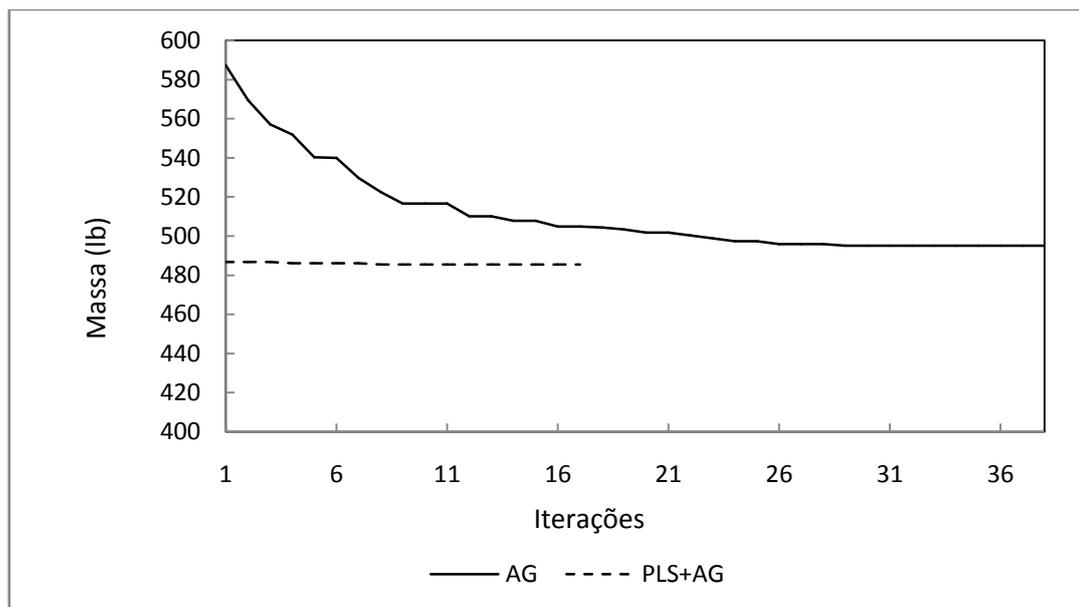


Figura 7.10: Gráfico da convergência do PLS+AG e AG para a treliça de 10 barras

Neste exemplo o Algoritmo Genético com a população inicial aleatória, seção 7.4.2, converge em 38 iterações em cerca de 120 segundos, enquanto o Algoritmo Genético em que foi definida a população inicial, seção 7.4.3, converge em 17 iterações em cerca de 54 segundos. Isto implica em uma redução do custo computacional de 85 % com o emprego de uma população inicial não aleatória.

7.5 Treliça espacial de 72 barras

A treliça tridimensional de 72 barras é ilustrada na figura 7.11. Devido às simetrias existentes na treliça as variáveis de projeto podem ser agrupadas em 16 parâmetros de projeto como listado na tabela 7.18. Os apoios desta estrutura estão localizados na base e os carregamentos externos são mostrados na tabela 7.17. As barras são feitas de alumínio e suas propriedades estão definidas na tabela 7.19.

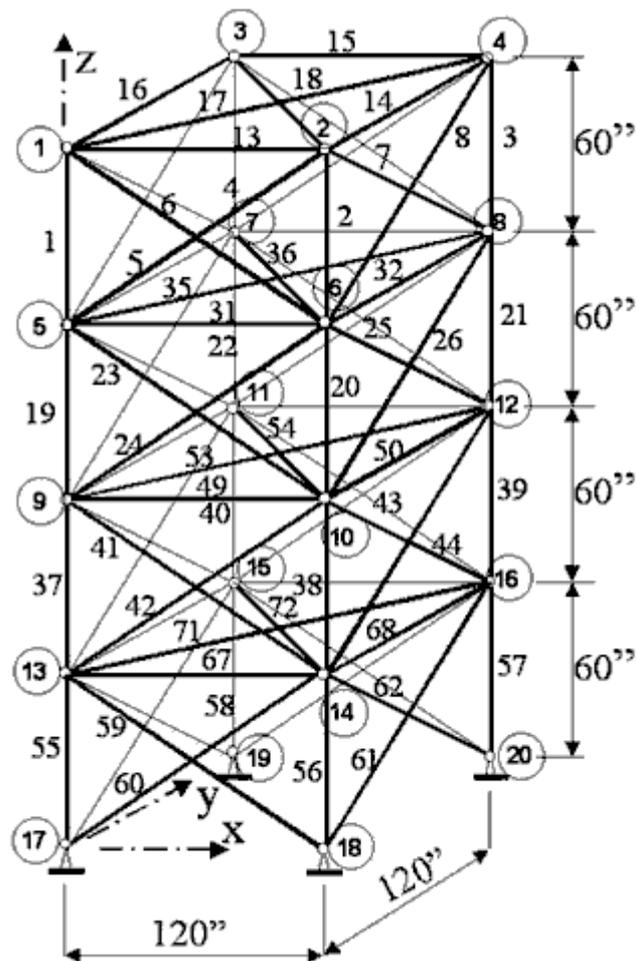


Figura 7.11: Treliça de 72 barras

Tabela 7.17: Treliza de 72 barras - Carregamento aplicados

Caso	Número Nó	Px (Kips)	Py (Kips)	Pz (Kips)
1	1	5.0 (22.25 kN)	5 (22.25 kN)	-5 (-22.25 kN)
2	1	0	0	-5 (-22.25 kN)
	2	0	0	-5 (-22.25 kN)
	3	0	0	-5 (-22.25 kN)
	4	0	0	-5 (-22.25 kN)

Tabela 7.18: Treliza de 72 barras - Variáveis de projeto

Variáveis de projeto	Barras
1	1-4
2	5-12
3	13-16
4	17,18
5	19-22
6	23-30
7	31-34
8	35,36
9	37-40
10	41-48
11	49-52
12	53,54
13	55-58
14	59-66
15	67-70
16	71,72

Tabela 7.19: Treliza de 72 barras - Propriedades

Módulo de elasticidade	$E=10^7$ psi (68950 MPa)
Densidade	$\rho = 0.10$ lb/in ³ (2767 kg/m ³)
Tensão admissível	± 25000 psi (± 172 MPa)
Deslocamento admissível	± 0.25 in ($\pm 0,635$ mm)

7.5.1 Resultados empregando Programação Linear Seqüencial (PLS)

Os resultados encontrados aplicando Programação Linear Seqüencial (PLS) são mostrados na tabela 7.20. Nesta tabela os resultados mostrados são comparados aos obtidos por [Haftka e Gürdal,1992]. As figuras 7.12 e 7.13 mostram a curva de convergência e a geometria final da treliça espacial de 72 barras, respectivamente.

Tabela 7.20: Treliça de 72 barras - Resultados obtidos empregado PLS

Variáveis de projeto	Haftka		Presente Trabalho	
	in ²	(mm ²)	in ²	(mm ²)
1	0,16	(101,4)	0,17	(110,8)
2	0,53	(344,3)	0,76	(491,2)
3	0,41	(264,5)	0,45	(288,4)
4	0,57	(367,1)	0,37	(241)
5	0,51	(326,9)	0,59	(378,3)
6	0,52	(335,5)	0,62	(400,9)
7	0,10	(64,5)	0,10	(64,5)
8	0,10	(64,5)	0,10	(64,5)
9	1,28	(825,8)	1,04	(671)
10	0,51	(332,1)	0,54	(351,5)
11	0,10	(64,5)	0,10	(64,5)
12	0,10	(64,5)	0,10	(64,5)
13	1,90	(1223,9)	1,54	(991,8)
14	0,52	(332,4)	0,53	(345)
15	0,10	(64,5)	0,10	(64,5)
16	0,10	(64,5)	0,10	(64,5)
Massa lb (kg)	379,66(172,2)		403,24(182,9)	

Comparando os valores das variáveis de projeto e da função objetivo encontrados ao final do processo de otimização com os resultados obtidos com Haftka e Gürdal, tabela 7.20, pode-se notar que os valores são satisfatórios Calculando a relação entre as funções objetivo da tabela 7.20 verifica-se uma diferença igual a 5,98%.

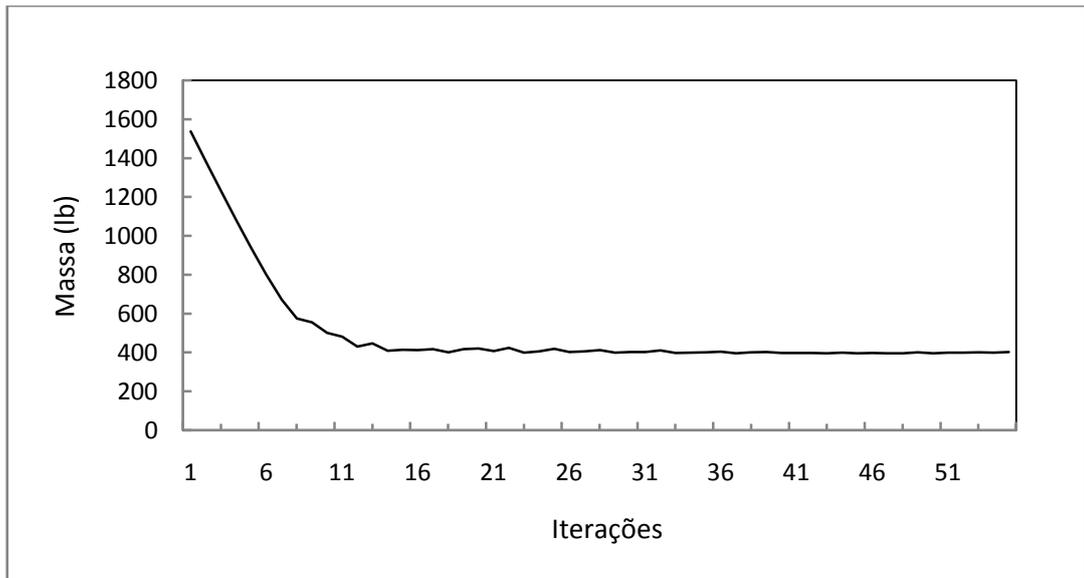


Figura 7.12: Gráfico da convergência do PLS para treliça de 72 barras

Na figura 7.12 é possível visualizar a curva de convergência da função objetivo para o processo de otimização da treliça de 72 barras empregando PLS. Este algoritmo encontrou a solução ótima após 63 iterações em cerca de 1 minuto e 45 segundos.

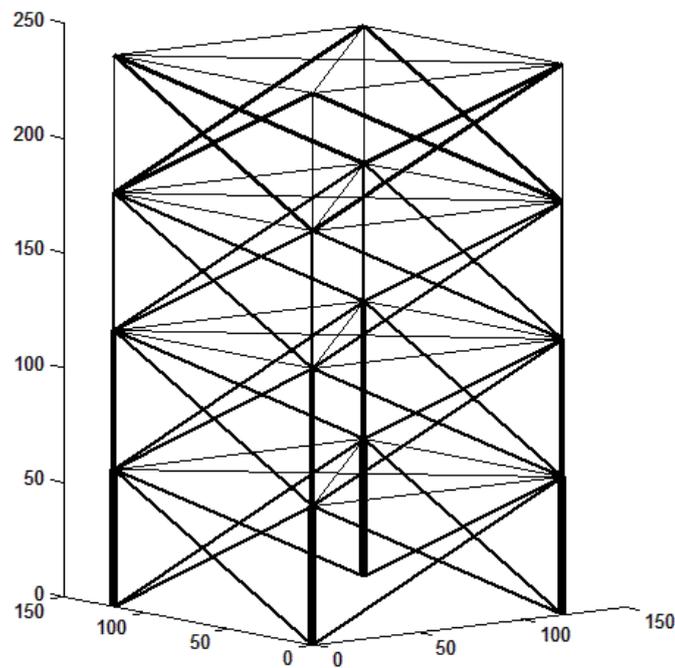


Figura 7.13: Geometria final da treliça espacial 72 barras

Tabela 7.21: Treliça de 72 barras - Deslocamentos obtidos no final da otimização com PLS

Número Nó	Desloc. em x (in)	Desloc. em y (in)	Desloc. em z (in)
1	0,25	0,24	-0,05
2	0,21	0,19	-0,03
3	0,20	0,20	-0,10
4	0,19	0,21	-0,03
5	0,16	0,16	0,03
6	0,20	0,14	-0,03
7	0,14	0,14	-0,08
8	0,14	0,20	-0,03
9	0,12	0,12	0,04
10	0,11	0,09	-0,01
11	0,10	0,10	-0,05
12	0,09	0,10	-0,01
13	0,05	0,05	0,02
14	0,05	0,04	-0,01
15	0,05	0,05	-0,03
16	0,04	0,04	-0,01
17	0,00	0,00	0,00
18	0,00	0,00	0,00
19	0,00	0,00	0,00
20	0,00	0,00	0,00

Na tabela 7.21 pode-se notar que a restrição de deslocamento máximo no nó número um na direção x se faz ativa ao final da otimização. A restrição de tensão não se encontra ativa em nenhum dos elementos.

7.5.2 Resultados empregando Algoritmos Genéticos (AG)

A otimização da treliça de 72 barras utilizando Algoritmos Genéticos é um exemplo numérico clássico e tem sido alvo de estudo por diversos pesquisadores como [Venkayya, 1971; Schmit, 1980]. Diferentemente dos demais exemplos utilizados neste trabalho, a treliça de 72 barras usa variáveis de projeto contínuas combinadas com AG.

As áreas das seções transversal das barras são limitadas em $0,1 \text{ in}^2$ e $36,0 \text{ in}^2$ [$64,5 \text{ mm}^2$ - 23226 mm^2]. A tabela 7.21 mostra os resultados encontrados. A curva de convergência é mostrada na figura 7.14. Nota-se, que o custo computacional foi novamente superior ao encontrado utilizando PLS.

Configuração do AG utilizado:

- População inicial aleatória
- Tamanho da população: 300
- Número máximo de iterações: 200
- Probabilidade de mutação Pm: 0,01
- Coeficiente de penalidade: $1,0 \times 10^1$

Tabela 7.22: Treliça de 72 barras - Resultados obtidos com AG

Variáveis de projeto	Venkayya		Schmit		Presente Trabalho	
	in ²	(mm ²)	in ²	(mm ²)	in ²	(mm ²)
1	0,16	(101)	0,16	101	0,20	(128,6)
2	0,56	(352,1)	0,55	352,1	0,34	(220,8)
3	0,38	(268,1)	0,42	268,1	0,12	(75,9)
4	0,51	(367,7)	0,57	367,7	0,25	(161,7)
5	0,61	(337,6)	0,52	337,6	0,61	(390,8)
6	0,53	(333,7)	0,52	333,7	0,36	(229,5)
7	0,10	(64,5)	0,10	64,5	0,12	(80,1)
8	0,10	(64,5)	0,10	64,5	0,16	(102,3)
9	1,25	(817,4)	1,27	817,4	1,12	(721,2)
10	0,52	(330,2)	0,51	330,2	0,42	(268,8)
11	0,10	(64,5)	0,10	64,5	0,12	(75,9)
12	0,10	(64,5)	0,10	64,5	0,12	(79,3)
13	1,82	(1216,1)	1,89	1216	1,89	(1216,8)
14	0,52	(330,6)	0,51	330,6	0,44	(282,1)
15	0,10	(64,5)	0,10	64,5	0,10	(64,6)
16	0,10	(64,5)	0,10	64,5	0,10	(65)
Massa lb (kg)	381,28(172,2)		379,64 (172,2)		380,80 (172,7)	

Na tabela 7.22, é possível notar que os resultados obtidos utilizando algoritmos genéticos apresentam ampla concordância com os valores utilizados como comparação.

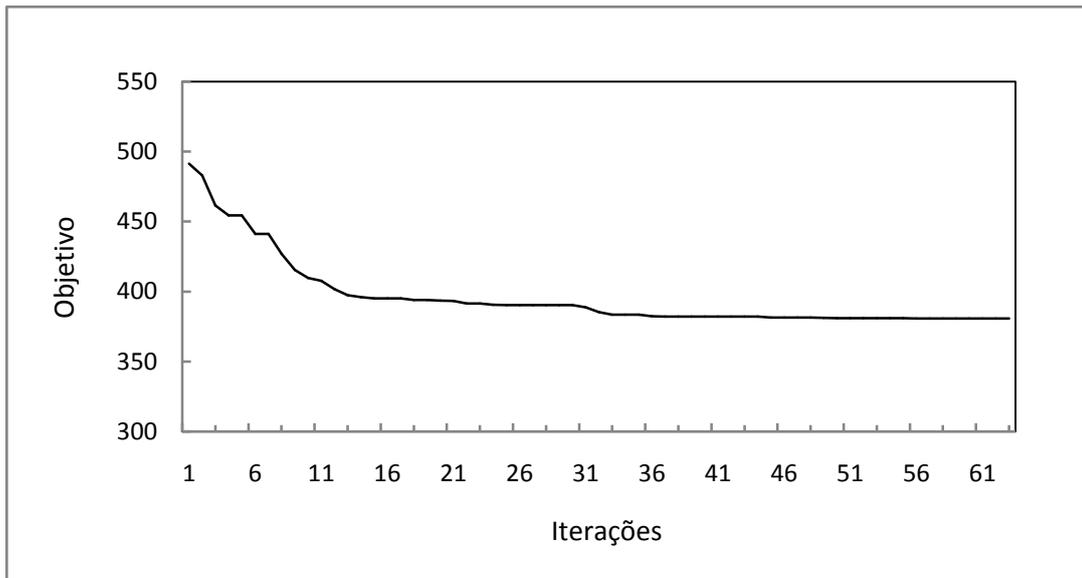


Figura 7.14: Gráfico da convergência do AG para a treliça de 72 barras

Na figura 7.14 é possível visualizar a curva de convergência da função objetivo para o processo de otimização da treliça de 72 barras empregando AG. Este algoritmo encontrou a solução ótima após 63 iterações em cerca de 9 minutos.

7.5.3 Resultados empregando Algoritmos Genéticos com utilização de população inicial

Almejando obter ganhos na eficiência do algoritmo, o problema da seção 7.5.2 é resolvido novamente. Porém, neste exemplo, usam-se os resultados obtidos na seção 7.5.1 para definir a população inicial. Os resultados obtidos para as variáveis de projeto são mostrados na tabela 7.17. O histórico da convergência da função objetivo é disposto na figura 7.23.

Configuração do AG utilizado:

- População inicial: 0,17; 0,76; 0,45; 0,37; 0,59; 0,62; 0,1; 0,1; 1,0; 0,5; 0,1; 0,1; 1,55; 0,53; 0,1; 0,1
- Tamanho da população: 300
- Número máximo de iterações: 100
- Probabilidade de mutação P_m : 0,01
- Coeficiente de penalidade: 1.0×10^1

Tabela 7.23: Treliça de 72 barras - Resultados obtidos com PLS + AG

Variáveis de projeto	in ²	(mm ²)
1	0,21	(137,3)
2	0,33	(213,9)
3	0,11	(70,7)
4	0,27	(176,3)
5	0,62	(398,5)
6	0,37	(237,1)
7	0,10	(64,5)
8	0,10	(64,5)
9	1,17	(751,6)
10	0,41	(263,1)
11	0,10	(64,5)
12	0,12	(76,2)
13	1,76	(1132,6)
14	0,44	(283,3)
15	0,10	(64,5)
16	0,10	(64,5)
Massa lb (kg)	377,90	(171,4)

Nota-se na tabela 7.23, que a redução da massa foi de apenas 0,76% em relação aos valores obtidos na seção 7.5.2 do presente trabalho.

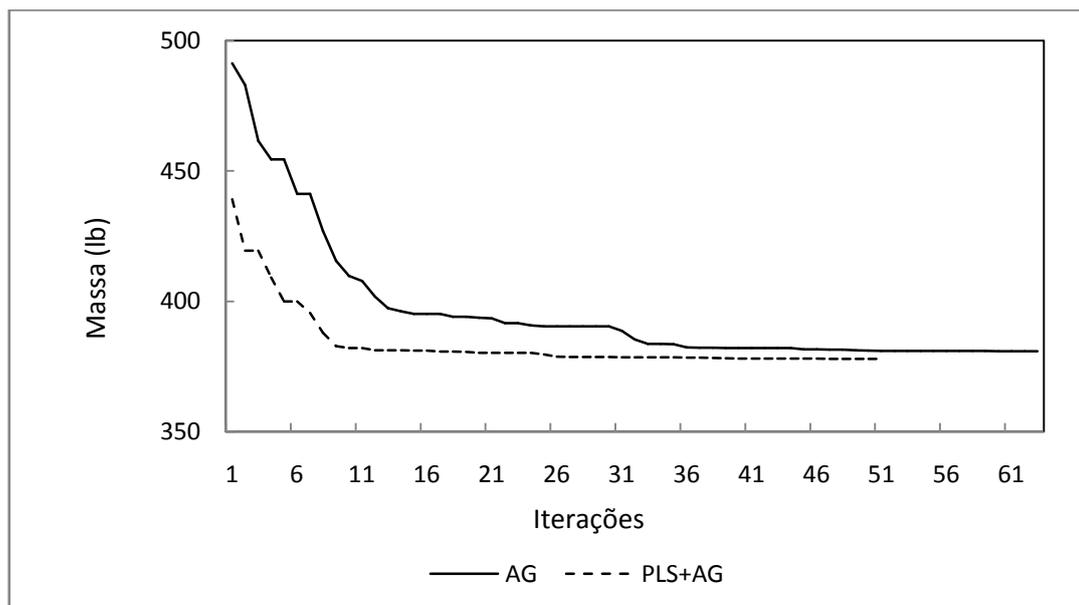


Figura 7.15: Gráfico da convergência do PLS+AG e AG para a treliça de 72 barras

Na figura 7.15 é possível visualizar a curva de convergência da função objetivo para o o Algoritmo Genético com a população inicial aleatória (AG) e Algoritmo Genético em que foi definida a população inicial (PLS+AG). É possível notar uma redução no número de iterações dp PLS+AG que converge em 51 iterações em cerca de 7 minutos e 20 segundos, enquanto AG, seção 7.5.2, converge em 61 iterações em cerca de 9 minutos. Isto implica em uma redução do custo computacional de 18,5 % com o emprego de uma população inicial não aleatória.

8. CONCLUSÃO

O objetivo principal desta dissertação era desenvolver uma formulação matemática para o problema de minimização da massa de treliças bidimensionais e tridimensionais, submetidas a restrições de tensão e deslocamento, utilizando um programa desenvolvido em MATLAB. Este objetivo foi alcançado, tendo em vista a formulação do problema apresentada no capítulo 6, com os respectivos resultados apresentados no capítulo 7. Os resultados obtidos apresentam ampla concordância com os valores fornecidos pela literatura consultada.

A escolha do MATLAB como plataforma de programação facilitou muito a implementação dos códigos fontes, pois além da facilidade de operação, possui diversas funções e permite operar com os mais variados tipos de linguagem.

Em relação ao PLS, o algoritmo mostrou-se eficiente na solução dos problemas propostos no capítulo 7 com um custo computacional competitivo. No entanto, o algoritmo mostrou-se sensível em relação aos limites móveis, e em alguns casos, se não definidos corretamente, podem gerar oscilações próximas à vizinha do mínimo da função. Este problema está descrito em detalhes no trabalho de [Muñoz-Rojas, 2003].

A utilização do SLP implica realizar o cálculo da sensibilidade a cada iteração, que é responsável pelo consumo da maior parte do esforço computacional. Entretanto, devido à utilização de elementos finitos isoparamétricos foi possível utilizar o método analítico direto para o cálculo da sensibilidade. Desta forma, o custo computacional da análise de sensibilidade tornou-se semelhante ao tempo gasto pela análise estrutural em todos os exemplos estudados.

O Algoritmo Genético também se mostrou eficiente na solução dos problemas propostos, tanto com variáveis discretas como contínuas. Porém, o custo computacional foi maior que o PLS em todos os casos estudados, considerando a população inicial totalmente aleatória. A eficiência do AG está fortemente ligada aos parâmetros utilizados (tamanho da população, taxa de mutação, entre outros). A influência destes parâmetros em problemas de otimização de treliças pode ser obtido em detalhes no trabalho de [Guerra, 2008].

O emprego do Algoritmo Genético, com a inicialização de uma população composta pelas soluções encontradas no PLS, mostrou-se com um custo computacional melhor que o algoritmo com uma população inicial aleatória. Através dos resultados apresentados no capítulo 7 pôde-se demonstrar que é possível acelerar a velocidade de convergência dos Algoritmos Genéticos através da inicialização de uma população inicial, mesmo que, a literatura demonstre que a inicialização do problema não seja crítica, se o número de indivíduos da população for grande suficiente.

9. SUGESTÕES PARA FUTUROS TRABALHOS

- Implementar um Algoritmo Genético baseado no trabalho de [Deb, 2001] evitando os problemas associados ao método da penalidade externa utilizado nesta pesquisa. Neste método é fixado um parâmetro de penalidade, P para todo o processo e surgem duas dificuldades associadas a este método:

1. A solução da função objetivo depende do parâmetro de penalidade, P . Diversos autores procuraram encontrar o melhor valor para P , para conduzir a procura dentro da região admissível. Isto requerer uma extensa experimentação para encontrar um valor razoável. Este problema é tão complicado que alguns pesquisadores usam diferentes valores de P , dependendo do nível de violação das restrições, enquanto outros atualizam o parâmetro de penalidade a cada iteração a partir de parâmetros que fixavam o raio de evolução (métodos de penalidade dinâmica).

2. A inclusão do termo de penalidade distorce a função objetivo, para valores pequenos de P , a distorção é pequena, mas o ótimo da função penalizada pode não estar perto do verdadeiro ótimo. Por outro lado, se é usado um P grande, o ótimo da função penalizada é próximo do mínimo, mas a distorção pode ser tão grande que a função penalizada pode ter mínimos fictícios.

- Acrescentar restrições com requisitos de confiabilidade conectados aos possíveis modos de falha das estruturas criando um método de otimização estrutural baseado em confiabilidade e não mais determinístico. Tipicamente estas restrições especificam que a probabilidade de falha seja menores que 10^{-3} a 10^{-5} . Estas pequenas probabilidades implicam no fato de que os métodos usuais de otimização estocástica não podem ser usados efetivamente na otimização baseada em confiabilidade.

- Verificar o comportamento do algoritmo implementado na presente pesquisa na otimização de estruturas submetidas a carregamentos dinâmicos. A presença deste tipo de carregamento torna o processo de otimização extremamente caro, do ponto de vista do esforço computacional e existe um grande interesse no desenvolvimento de novos algoritmos mais robustos e eficientes. Os Algoritmos destinados a problemas dinâmicos com variáveis contínuas estão bem desenvolvidos, enquanto que os para otimização discreta ou mista, estão ainda em fase de desenvolvimento.

- Tornar o algoritmo mais robusto para permitir a otimização dos custos de produção levando em considerações outros processos envolvidos na fabricação, permitindo a criação de uma ferramenta útil para a análise de investimentos em centros de trabalho.

10. REFERÊNCIAS BIBLIOGRÁFICAS

- Adeli, H., & Kamal, O. (1992). Concurrent Optimization of Large Structures Part I and Part II. *Journal of Aerospace Engineering ASCE* , 79-100.
- Alves Filho, A. (2000). *Elementos Finitos: A Base da Tecnologia CAE*. São Paulo: Érica.
- Argyris, J. H., & Kelsey, S. (1960). Energy Theorems and Structural Analysis. *Aircraft Engineering* , 1954-1955.
- Arora, J. S. (1989). *Introduction to Optimum Design*. McGraw-Hill.
- Atkin, R. J., & Fox, N. (1980). *Introduction to Theory of Elasticity*. Longman.
- Boresi, A. P., & Chong, K. P. (1987). *Elasticity in Engineering Mechanics*. Nova York: Elsevier.
- Cai, J., & Thiereut, G. (1993). Discrete Optimization of Structures Using Improved Penalty Function Method. *Engineering Optimization Vol. 21, No.4* , 293-306.
- Castro, R. E. (2001). Otimização de Estruturas com Multi-Objetivos via Algoritmos Genéticos. *Tese de Doutorado* . Rio de Janeiro: Universidade do Rio do Rio de Janeiro.
- Cheng, G. (1992). Introduction to Structural Optimization: Theory, Methods and Solution. *Lecture Notes* . Dalian University of Technology.
- Clough, R. W. (1960). *The Finite Element Method in Plane Stress Analysis*. Proceeding of 2 nd ASCE Conf. in Electronic Computation.
- Coello, C. (1994). Discrete optimization of trusses using genetic algorithms. *paper presented at EXPERSYS-94,I.I.T.T* , 331-336.
- Courant, R. (1943). Variational Methods for solutions of Problems of Equilibrium and Vibration. *Bull. Am. Math. Soc. Vol.49* , 231-236.
- Dantzig, G. (1963). *Linear Programming and Extensions*. New Jersey: 1963.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons.
- Felippa, C. A. (2001). A Historical Outline of Matrix Structural Analysis: a Play in Three Acts. *Computers and Structures* , 1313-1324.
- Fletcher, R. (1987). *Practical Methods of optimization*. Nova York: Wiley.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley Longman Inc.: Nova York.
- Goldberg, D. E., & Samtani, M. P. (1986). Engineering Optimization Via Genetic Algorithms. *In Proc. of the 9th Conf. Eletronic Computation ASCE* , 471-482.

- Grierson, D. E., & Pak, W. H. (1993). Optimal Sizing, Geometrical and Topological Design Using a Genetic Algorithm . *Structural Optimization* , 151-159.
- Guerra, C. (2008). otimização Paramétrica de Estruturas Treliçadas por Algoritmos Genéticos. *Dissertação de Mestrado* . Universidade Federal do Rio Grande do Sul.
- Haftka, R., & Gürdal, Z. (1992). *Elements of Structural Optimization*. Kluwer Academic Publishers.
- Holland, J. H. (1975). *Adaptation in natural and Artificial Systems*. Michigan: Ann Arbor: University of Michigan Press.
- Karmarkar, N. (1984). A new polynomial time algorithm for linear programming. *Combinatorica* , 373 - 395.
- Kripka, M. (2004). Discrete Optimization of Trusses by Simulated Annealing. *J. of the Braz. Soc. of Mech. Sci. & Eng. Vol.XXVI No.2* , 170=173.
- Lemonge, A., & Barbosa, H. (2000). Genetic Algorithm for Shape, Sizing and Configuration Optimization of framed Structures. *Proc. of the 9th Conference in Eletronic Computation* , Rio Janeiro.
- Machado, I. G. (1996). *Soldagem e técnicas conexas: Processos*. Porto Alegre: Editado pelo Autor.
- Maxwell, J. (1872). On Reciprocal Figures, Frames and Diagrams of Force. *Trans. Royal Soc. Edinb. Vol. 26/1* .
- Metropolis, N., & Ulam, S. (1949). The Monte Carlo Method. *Journal of American Statistical Association, Vol.44, No. 247* , 335 - 341.
- Michell, A. (1904). The Limits of Economy of Material in Frame Structures. *Philosophical Maganize 8* , 589-597.
- Muñoz, P. A. (2003). Otimização de Pré- formas e Matrizes em Problemas Bidimensionais de Forjamento. *Tese de Doutorado* . Porto Alegre.
- Rajeev, S., & Krishnamoorthy, C. S. (1992). Discrete Optimization of Structures Using Genetic Algorithms. *J. Struct. Eng* , 1233-1250.
- Sant'Anna, H. M. (2002). Otimização Topológica de Estruturas Bidimensionais Contínuas Submetidas a Restricções de Flexibilidade e Tensão. *Dissertação de Mestrado* . Porto Alegre: Universidade Federal do Rio Grande do Sul.
- Schmit, L., & Fleury, C. (1980). Discrete-Continuous Variables Structures Using Dual Methods. *AIAA J 18* , 1515-1524.
- Souza, R. P., & Fonseca, J. (2008). Optimum Truss Design under Failure Constraints Combining Continuous and Integer Programming. *EngOpt 2008 - International Conference on Engineering Optimization* .
- Turner, M. J., Clough, R. w., Martin, H., & Top, L. J. (1956). Stiffness and Deflection Analysis of Complex Structures. *J. Aeron. Sci.* , 805-823.

Venkayya, V. B. (1971). Design of Optimum structures. *J. Computers and Structures Vol.1, no. 1-2* , 265-309.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)