



ANDRÉ BATISTA DE ALMEIDA

Otimização estrutural em componentes mecânicos utilizando Algoritmos Genéticos

45/14

**CAMPINAS
2014**



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

ANDRÉ BATISTA DE ALMEIDA

Otimização estrutural em componentes mecânicos utilizando Algoritmos Genéticos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO(A) ALUNO(A).....*André Batista de Almeida* E ORIENTADA PELO(A) PROF(A). DR(A)...*Auteliano Antunes dos Santos Junior*

ASSINATURA DO(A) ORIENTADOR(A)

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Elizangela Aparecida dos Santos Souza - CRB 8/8098

AL64o Almeida, André Batista, 1978-
Otimização estrutural em componentes mecânicos utilizando algoritmos genéticos / André Batista de Almeida. – Campinas, SP : [s.n.], 2014.

Orientador: Auteliano Antunes dos Santos Júnior.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Otimização estrutural. 2. Algoritmos genéticos. 3. Método dos elementos finitos. I. Santos Júnior, Auteliano Antunes dos, 1963-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Structural optimization in mechanics components using genetics algorithms

Palavras-chave em inglês:

Structural optimization

Genetic algorithm

Finite element method

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora:

Auteliano Antunes dos Santos Júnior [Orientador]

Hélio Fiori de Castro

Marli de Freitas Gomes Hernandez

Data da defesa: 29-04-14

Programa de Pós-Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETOS MECÂNICOS

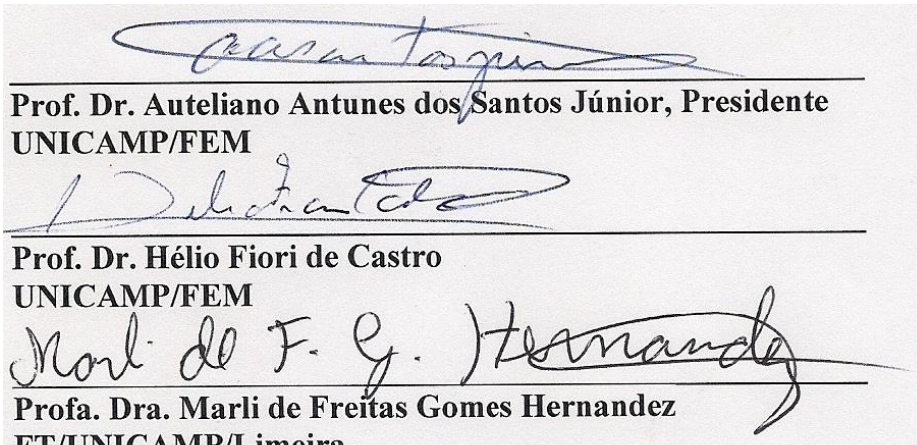
DISSERTAÇÃO DE MESTRADO ACADEMICO

Otimização estrutural em componentes mecânicos utilizando Algoritmos Genéticos

Autor: André Batista de Almeida

Orientador: Prof. Dr. Auteliano Antunes dos Santos Júnior

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. Auteliano Antunes dos Santos Júnior, Presidente
UNICAMP/FEM

Prof. Dr. Hélio Fiori de Castro
UNICAMP/FEM

Profa. Dra. Marli de Freitas Gomes Hernandez
FT/UNICAMP/Limeira

Campinas, 29 de abril de 2014.

Dedicatória

Dedico este trabalho às pessoas que acreditaram que eu podia acreditar. Dedico este trabalho a minha esposa Keila por cooperar, tolerar e suportar todos os desafios que apareceram, ao longo desses anos, na estrada de nossas vidas.

Agradecimentos

Meus sinceros agradecimentos:

Aos companheiros de trabalho Alexandre Perussi, Ednei Lopes e Roberto Oliveira com quem aprendi muito durante os anos que estivemos na Tyco Electronics.

À equipe de CAE da GE Óleo e Gás onde tive a oportunidade de trabalhar com pessoas brilhantes, dos quais destaco o Sr. Yoshio e aos amigos Hugo Lopes, Danilo Oliveira e Yuri Del’Sarto.

Em especial, agradeço às pessoas que me cobraram quase todos os dias, neste último ano, para que finalizasse este trabalho, Sra. Marli Pusinhol e Lucas Menochelli. Ao Professor Francisco Bianchi que, graças à sua sabedoria, deu-me conselhos imprescindíveis para que a comunicação entre os softwares ocorressem, à Professora Vera Gabaldi pela ajuda e toda a equipe da Faculdade de Tecnologia de Itu.

Agradeço ao Professor Auteliano por acreditar em meu trabalho e, principalmente, pela paciência que teve ao longo desses anos.

*“Sábio é aquele que
conhece os limites da própria
ignorância.”*

Sócrates

Resumo

Estudou-se, neste trabalho, o emprego de uma ferramenta de otimização metaheurística, o método dos Algoritmos Genéticos, para a otimização de forma de componentes mecânicos. O trabalho envolveu o estudo das ferramentas usuais e de diversas novas ferramentas para otimização, com foco em AG. Para tanto, foram desenvolvidas rotinas computacionais específicas. Os resultados obtidos com este método foram comparados com os obtidos com o emprego de um programa comercial (Ansys[®]), que utiliza ferramentas tradicionais de otimização (método de aproximação por subproblema). O trabalho buscou ainda comparar os seus resultados com os obtidos em literatura. Os resultados mostraram que é possível utilizar AG para otimização de forma, com desempenho adequado em relação ao método de aproximação por subproblema e aos resultados de literatura. Mostraram ainda que, utilizando parâmetros otimizados, é possível obter resultados mais adequados com o programa desenvolvido nesta dissertação.

Palavras-chave: Otimização de Forma, Algoritmo Genético, Elementos Finitos.

Abstract

This study consisted of using a metaheuristic optimization tool, Genetic Algorithm, for the mechanical components shape optimization. It involved the study of usual and several new tools for optimization, focusing GA. For this, specific computing routines were developed. The obtained results were compared to the ones generated by using a commercial program (Ansys), that uses traditional optimization tools (subproblem approximation method). The study also compared its results with the ones obtained in the literature. The results showed that it is possible to use AG for shape optimization with adequate performance related to the subproblem approximated method and the literature results. It was also showed that it is possible to obtain more adequate results, using optimized parameters, with the program developed in this study.

Keywords: Shape Optimization, Genetic Algorithm, Finite Elements.

Lista de Ilustrações

Figura 2.1 O mínimo de uma função $f(x)$ é o ponto que corresponde ao valor máximo da negativa da função $-f(x)$	5
Figura 2.2 Categorias de Otimização	6
Figura 2.3 Restrições superficiais em um domínio hipotético no plano	10
Figura 2.4 Contorno de uma função objetivo.....	12
Figura 2.5 Comportamento de uma função abstrata, definindo seus mínimos locais e globais ...	17
Figura 2.6 As três categorias de otimização estrutural	20
Figura 2.7 Representação computacional de alguns problemas de engenharia	23
Figura 2.8 Evolução das ferramentas matemáticas para o MEF.....	24
Figura 2.9 Geometria, carga e a malha gerada pelo MEF.....	25
Figura 2.10 Divisão do domínio em subdomínios (elementos)	29
Figura 2.11 Descrição dos principais elementos e seus respectivos números nodais	30
Figura 2.12 Definição do método por topografia.....	34
Figura 2.13 Representação gráfica do algoritmo colônia de formigas.....	38
Figura 2.14 Fluxograma do algoritmo RS.....	42
Figura 3.1 Elemento Plane183	55
Figura 3.2 Elemento Solid186.....	56
Figura 3.3 Estrutura do AG.....	60
Figura 3.4 Esquema de seleção por <i>fitness</i> -proporcional (roleta)	63
Figura 3.5 Fluxograma do software desenvolvido, mostrando a interação entre o programa de AG e o software de elementos finitos Ansys®	66
Figura 3.6 Detalhe parcial do software implementado	67
Figura 3.7 Detalhe do arquivo batch	67
Figura 3.8 Representação da condição de contorno inicial para uma placa bidimensional	70
Figura 3.9 Representação da condição de contorno inicial de um gancho bidimensional.....	72
Figura 3.10 Representação da condição de contorno inicial para um componente tridimensional.....	73
Figura 3.11 Demonstração das variáveis para o terceiro estudo de caso	74

Figura 3.12 Representação da condição de contorno inicial para o quarto estudo de caso	75
Figura 4.1 Influência das alterações dos parâmetros do AG.....	79
Figura 4.2 Representação das geometrias iniciais e finais de uma placa bidimensional, utilizando o AG como ferramenta de otimização.....	80
Figura 4.3 Influência das alterações dos parâmetros do AG.....	82
Figura 4.4 Geometrias iniciais e finais de um gancho bidimensional utilizando o AG e o método por subproblemas como ferramentas de otimização	83
Figura 4.5 Influência das alterações dos parâmetros do AG.....	85
Figura 4.6 Detalhe da evolução para o primeiro estudo de caso.....	85
Figura 4.7 Representação das geometrias iniciais e finais para um estudo tridimensional, utilizando o AG como ferramenta de otimização.....	86
Figura 4.8 Influência da população em função do mínimo global.....	88
Figura 4.9 Tempo computacional em função da variação da população	89
Figura 4.10 Tempo computacional por iteração em função do tipo de processador.....	90
Figura 4.11 Condição de contorno do modelo proposto para o quarto estudo de caso.....	91
Figura 4.12 Geometria final do modelo proposto por Annicchiarico e Cerrolaza, (2001)	92
Figura 4.13 Geometria final do modelo proposto por Annicchiarico e Cerrolaza, (2001), utilizando as configurações da Tabela 4.9	93

Lista de Tabelas

Tabela 2.1	Possibilidade de convergência em função da escolha do método	18
Tabela 2.2	Graus de liberdade e vetores de força em diferentes campos da Engenharia.....	29
Tabela 3.1	Tabela comparativa entre o RS e o AG	59
Tabela 3.2	Valores de máximo e mínimo de cada variável para o primeiro estudo de caso.....	71
Tabela 3.3	Valores de máximo e mínimo de cada variável para o segundo estudo de caso	72
Tabela 3.4	Valores de máximo e mínimo de cada variável para o terceiro estudo de caso	73
Tabela 3.5	Valores de máximo e mínimo de cada variável para o quarto estudo de caso	76
Tabela 4.1	Limites dos parâmetros máximo, mínimos, e os valores finais para cada variável referente ao primeiro estudo de caso	81
Tabela 4.2	Valores de Tensão e Área para as configurações analisadas no primeiro estudo.....	81
Tabela 4.3	Limites dos parâmetros máximo, mínimos, e os valores finais para cada ferramenta de otimização referente ao segundo estudo de caso	83
Tabela 4.4	Valores de Tensão e Área para as configurações analisadas no segundo estudo	84
Tabela 4.5	Limites dos parâmetros máximo, mínimos e os valores finais para cada variável referente ao terceiro estudo de caso	86
Tabela 4.6	Valores de Tensão e Volume para as configurações analisadas no terceiro estudo	87
Tabela 4.7	Resultados obtidos em função do aumento da população	88
Tabela 4.8	Métodos adotados por ambos os autores	91
Tabela 4.9	Métodos adotados por ambos os autores	92

Lista de Abreviaturas e Siglas

Letras Latinas

f(x) – função objetivo / *fitness*

g(x) – restrição

n – número de variáveis

m – número de restrições

x – variável

E – energia

T – temperatura

k – constante de Boltzmann

n – tamanho da população

r – valor da variável em uma determinada iteração

z – valor da *string*

l – comprimento de uma *sub-string*

L – comprimento da *string*

M – número total de incógnitas

N – número de formigas

P_m – Probabilidade de Mutação

P_c – Probabilidade de Cruzamento

R_c – Número Aleatório

C_m – Escalonamento

A – Área

E – Módulo de Elasticidade; Energia

P(E) – Probabilidade em função da energia

Letras Gregas

σ – Tensão [MPa]

ν – Poisson

Abreviações

AG – Algoritmo Genético

VP – Variável de Projeto

VE – Variável de Estado

BT – Busca Tabu

SIMP – Solid Isotropic Material with Penalization

CMLPSA – Corrected Multi-Level & Multi-Point Simulated Annealing

RS – Recozimento Simulado

NHPGA – Niche Hybrid Parallel Genetic Algorithm

DOE – Design of Experiments

APDL – Ansys Parametric Design Language

GUI – Interface Gráfica com o Usuário

HSAPS – Hybrid Simulated Annealing and Pattern Search

PCV – Problema do Caixeiro Viajante

SUMT - Sequential Unconstrained Minimization Technique

CSA – Cloud Simulated Annealing

Siglas

MEF – Método dos Elementos Finitos

CFD – Computational Fluid Dynamics

CAD – Computer Aided Design

CAE – Computer Aided Engineering

CAI – Computer Aided Innovation

CACD – Computer Aided Conceptual Design

TRIZ – Teoria da Solução Inventiva de Problemas

QFD – Desdobramento da Função Qualidade

SUMÁRIO

1. INTRODUÇÃO	1
2. REVISÃO DA LITERATURA E CONCEITOS BÁSICOS	4
2.1 Conceitos sobre otimização.....	4
2.1.1 Definição de um problema de otimização	8
2.1.2 Termos imprescindíveis para um estudo de otimização.....	9
2.1.3 Classificação dos problemas de otimização	12
2.2 Técnicas de otimização	16
2.3 Otimização de forma x otimização topológica.....	19
2.4 Métodos Numéricos voltados à otimização	21
2.4.1. Nós	28
2.4.2. Elementos	30
2.4.3. Funções de Aproximação	31
2.4.4. Métodos de Aproximação	31
2.5 Softwares Comerciais de MEF e seus módulos de otimização	33
2.5.1 Altair®	33
2.5.2 Simulia®	34
2.5.3 Nastran®	35
2.5.4 Ansys®	36
2.6 Métodos não-tradicionais de otimização	37
2.6.1 Colônia de Formiga.....	37
2.6.2 Enxame de Partículas	39
2.6.3 Recozimento Simulado	40
2.6.4 Busca Tabu.....	43
2.6.5 Algoritmos Genéticos.....	43
2.7 Estudos relacionados à otimização.....	45
3. MODELAGEM TEÓRICA E MÉTODOS.....	53
3.1. Descrição do Programa de Elementos Finitos:	53

3.1.1 APDL (Ansys Parametric Design Language)	54
3.1.2 Algoritmos de Otimização disponíveis no Ansys®	56
3.2 Ferramentas de otimização não-tradicionais	58
3.2.1 Comparação entre o Algoritmo Genético e Recozimento Simulado	58
3.2.2 Algoritmo Genético.....	60
3.2.2.1 Precisão do algoritmo.....	60
3.2.2.2 População	62
3.2.2.3 Mutação	62
3.2.2.4 Seleção	63
3.2.2.5 Elitismo	64
3.2.2.6 Cruzamento	64
3.2.2.7 Escalonamento	65
3.3 GAnsys	65
3.4 Métodos	66
3.5 Definição do problema	68
3.5.1 Primeiro estudo de caso.....	69
3.5.2 Segundo estudo de caso.....	71
3.5.3 Terceiro estudo de caso	73
3.5.4 Quarto estudo de caso	75
4. RESULTADOS E DISCUSSÕES	77
4.1 Estudos de otimização de forma.....	77
4.1.1 Estudo de uma placa bidimensional	77
4.1.2 Estudo de um gancho bidimensional.....	81
4.1.3 Estudo de um modelo tridimensional.....	84
4.2 Influências da alteração do parâmetro população	87
4.3. Estudo comparativo em relação a trabalhos realizados anteriormente	90
5 CONCLUSÕES E SUGESTÕES PARA PRÓXIMOS TRABALHOS.....	77
REFERÊNCIAS	96
APÊNDICE A – Influência das alterações dos parâmetros do AG	101
APÊNDICE B – Discretização do domínio em função do número de iteração.....	122

APÊNDICE C – Influência do aumento da população	125
APÊNDICE D – Programa GAnsys.....	126
APÊNDICE E – Programa em APDL de um Gancho bidimensional	144
APÊNDICE F – Programa em APDL para a utilização do método por subproblemas – Estudo referente a um Gancho bidimensional.....	147

1. INTRODUÇÃO

A competitividade do mercado automobilístico e o desenvolvimento de novos materiais e de novas tecnologias tornaram o segmento um campo fértil para que novos conceitos de gerenciamento e novos produtos fossem criados, ditando os avanços tecnológicos para outros segmentos. Como exemplo, novas ferramentas de medição, utilização de materiais poliméricos e compósitos, evolução nos equipamentos, processos de gerenciamento, entre muitas outras atividades, estão entre as inovações recentes na área. Isso fez com que tal segmento do mercado necessitasse de constantes pesquisas nos mais diversos campos, cujo principal objetivo é quase sempre, a minimização, ou a maximização, ou mesmo a convergência.

Até pouco tempo atrás, para dimensionar uma caldeira, um prédio, ou mesmo um motor, havia um profissional com conhecimento profundo sobre determinado produto ou equipamento que, por meio de cálculos analíticos e também de resultados experimentais, chegava a valores de tensão, temperatura, velocidade de fluxo entre outros.

Para as situações e necessidades daquela época, tal forma de desenvolvimento de um novo produto ou equipamento atendia plenamente. Contudo em poucos anos houve uma súbita transformação em todas as áreas. Se o modo de viver das pessoas já não é mais o mesmo que há vinte anos, nas indústrias, a evolução foi ainda mais rápida. Provavelmente, a popularização do computador foi um dos responsáveis por tais avanços.

O desenvolvimento de muitos produtos era realizado, tendo como referência fórmulas e resultados obtidos no passado; engenheiros e projetistas faziam os seus estudos para o desenvolvimento de um novo produto balizado nos históricos dos produtos desenvolvidos. Entretanto, as incertezas eram tantas que a equipe de engenharia era obrigada a superdimensionar seus produtos, quase sempre utilizando fatores de segurança elevados.

Com a disseminação do computador, a partir do início dos anos 60, softwares para simulação dos efeitos dos esforços físicos foram desenvolvidos. A área aeroespacial foi pioneira na utilização de tais ferramentas.

Devido à quantidade de cálculos matemáticos e à velocidade com que o computador consegue trabalhar, modelos matemáticos e formulações, que ficaram no baú da história,

passaram a ser implementados nas últimas décadas. Um dos principais exemplos foi o método de Newton-Raphson cuja finalidade é obter soluções de comportamentos não-lineares, amplamente utilizado em métodos numéricos atualmente. Este método foi desenvolvido em 1687 por Isaac Newton e Joseph Raphson, mas somente no século XX, com a utilização dos computadores, o método foi implementado em cálculos numéricos.

Para o projeto de desenvolvimento de um motor, de componentes ou de um novo design, devem-se levar em consideração suas características estruturais, dinâmicas ou mesmo aerodinâmicas, que são interdependentes. Existem diferentes métodos que se prestam a resolver os problemas de cada área separadamente. Para uma análise estrutural ou dinâmica, o Método dos Elementos Finitos (MEF) é, atualmente, o método mais difundido. O MEF é um método voltado para simulação de fenômenos físicos direcionados aos problemas da engenharia.

Em todas as aplicações, o analista busca uma quantidade de campo de uma determinada grandeza. Na análise de tensão, o objetivo é encontrar o campo de deslocamento ou o campo de tensão; em uma análise térmica, busca-se o campo de temperatura ou o fluxo de calor; em frentes de fluxos, busca-se a função da velocidade potencial; se a finalidade é a de realizar um estudo voltado aos comportamentos aerodinâmicos, existem diversas ferramentas voltadas às simulações computacionais cuja área se concentra em estudos dinâmicos de fluídos, e a sigla em inglês é CFD (*Computational Fluid Dynamics*). Independentemente dos métodos ou da área a serem empregados, os resultados de maior interesse são os picos, que mostram as regiões de maiores/menores esforços. O método dos elementos finitos é um caminho para encontrar uma solução numérica de um problema específico. Com a velocidade de processamento obtida pelos computadores, atualmente é possível obter excelentes resultados para as mais diversas aplicações, em especial com componentes estruturais, em pouco tempo.

Existem diversos métodos de otimização metaheurístico, que são ferramentas de otimização combinatória cuja finalidade é encontrar o ponto ótimo ou um ponto mínimo de uma função arbitrária. Essas ferramentas são utilizadas em problemas em que não se conhece um algoritmo eficiente e direto para um número elevado de equações, pois derivá-las ficaria impraticável.

Nos dias de hoje, esses métodos são amplamente utilizados, para problemas de otimização de rota, para otimização no tempo na troca de dados, para otimizações estruturais entre muitas

outras aplicações. A finalidade deste trabalho é a utilização de ferramentas com estas características dentro do método dos elementos finitos.

O objetivo desse trabalho é estudar a aplicação das técnicas de otimização baseadas em Algoritmos Genéticos (AG) em problemas relacionados a otimização de forma de componentes mecânicos estruturais. Essa técnica será aplicada em conjunto com o programa Ansys[®], software voltado aos estudos de esforços mecânicos pelo método dos elementos finitos, e o desempenho do estudo de otimização será comparado com a técnica implementada no programa para otimização de forma.

O uso AG, ao invés dos métodos tradicionais como o método aproximação de subproblema e método de primeira ordem para estudos de otimização, é desejável, pois se espera que os métodos tradicionais tenham dificuldades de convergência diante de problemas em que há uma quantidade relativamente grande de variáveis de projeto, apesar de que sua adoção poder acarretar em um aumento no custo computacional.

No capítulo 2, será feita uma revisão bibliográfica a respeito dos métodos dos elementos finitos e, principalmente, dos métodos relacionados à otimização pesquisados e empregados atualmente. No capítulo 3, serão abordadas as ferramentas MEF e a comunicação entre um software comercial voltado ao MEF e um algoritmo desenvolvido para estudos de otimização de forma. No capítulo 4, serão mostrados os resultados da comunicação entre o algoritmo genético e o método dos elementos finitos para resolução de problemas relacionados à otimização de forma e, no capítulo 5, serão apresentadas as conclusões deste trabalho e feitas algumas sugestões para os próximos trabalhos.

2. REVISÃO DA LITERATURA E CONCEITOS BÁSICOS

Este capítulo apresenta os conceitos sobre otimização, as técnicas utilizadas, a comparação entre os tipos de otimização de forma normalmente empregados, os métodos numéricos voltados à otimização, os programas comerciais, os métodos não-tradicionais de otimização e uma breve revisão sobre os estudos realizados sobre otimização. Cada um desses assuntos tem relação com o tema dessa dissertação e, por isso, está sendo apresentado.

2.1 Conceitos sobre otimização

Otimização é o ato de obter o melhor resultado sob dadas circunstâncias. Em design, construção, manutenção de qualquer sistema de engenharia, os engenheiros têm de assumir muitas decisões tecnológicas e gerenciais em diversas etapas. O objetivo final de todas essas decisões ou é minimizar o esforço necessário ou maximizar o benefício desejado (RAO, 2009).

O conceito de otimização está intrinsecamente ligado ao desejo de melhorar dia após dia. Embora não se possa reconhecê-lo conscientemente, o processo de otimização assume diferentes formas em diferentes campos de atuação. O trabalho, para fazer melhor do que foi feito, antes consome muita energia, quer para atletas, artistas, empresários, pessoas ou engenheiros. Assim, fazer melhor com a quantidade mínima de energia faz parte do cotidiano (VANDERPLAATS, 2007).

Segundo Haupt et al. (2004), otimização é o processo de fazer alguma coisa melhor. Otimização consiste nas variações dos conceitos iniciais, usando algumas informações e seguindo uma metodologia com o objetivo de melhorar uma determinada ideia ou proposta.

O esforço necessário ou o benefício desejado, em qualquer situação prática, pode ser expresso em função de variáveis de decisão. Assim, a otimização pode ser definida como o processo de encontrar as condições que fornecerão o valor máximo (benefício) ou mínimo (esforço) de uma função que descreve adequadamente o problema.

O conceito pode ser visto na Figura 2.1. Se um ponto x corresponde ao valor mínimo da função $f(x)$, o mesmo ponto também corresponde ao valor máximo da negativa da função $-f(x)$. Desta forma, o inverso da função será o ponto de máximo.

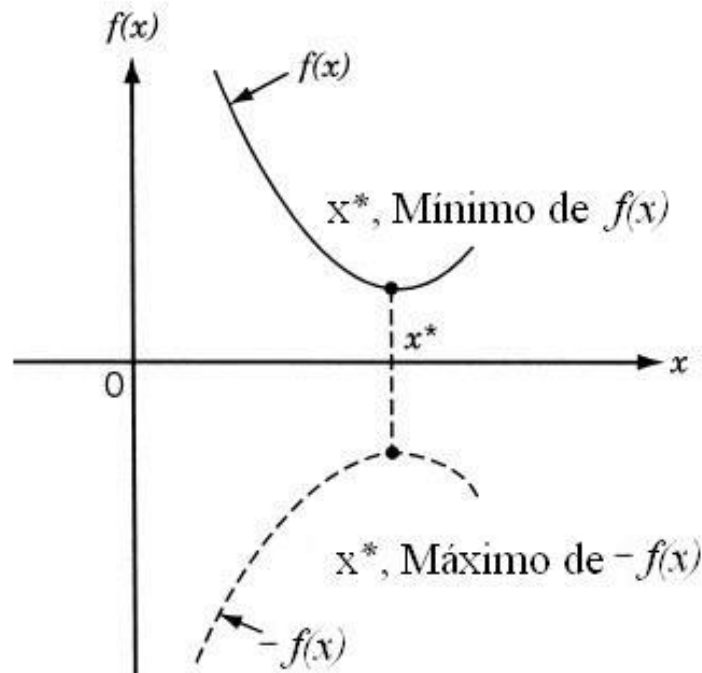


Figura 2.1 O mínimo de uma função $f(x)$ é o ponto que corresponde ao valor máximo da negativa da função $-f(x)$

Fonte Adaptado de Rao (2009 p. 2)

Muitas técnicas de otimização foram desenvolvidas a partir do século XVII, principalmente a partir dos trabalhos desenvolvidos por Euler e Lagrange, porém muitos desses métodos acabaram ficando relegados à história (BROTCHIE, 1997). Somente após a utilização do computador para desenvolvimento de cálculos exaustivos, tais métodos foram amplamente estudados e muitos outros foram criados.

Os primeiros métodos que surgiram foram os métodos clássicos de otimização que são úteis para encontrar as soluções ótimas de uma função contínua e diferenciável. Estes métodos são analíticos e fazem uso de técnicas de cálculos diferenciais para a localização do ponto ótimo. Uma ferramenta útil para este tipo de problema é conhecida como *multiplicadores de Lagrange*,

empregado em situações aonde há igualdade nas restrições. Há também a ferramenta Kuhn-Tucker, empregada quando há desigualdade.

Segundo Simon (2013), a palavra heurística, empregada no primeiro capítulo dessa dissertação, vem do grego, e significa encontrar, descobrir. Não se espera com a utilização de algoritmos heurísticos encontrar uma resposta ideal para determinado problema, e sim encontrar um conjunto de soluções que se aproxima à resposta ideal, permitindo escolher a melhor resposta. O termo metaheurística é usado para descrever uma família de algoritmos heurísticos.

Conforme Haupt et al. (2004), o processo de otimização pode ser dividido em seis categorias. A Figura 2.2 mostra de maneira sucinta as categorias relacionadas à otimização.

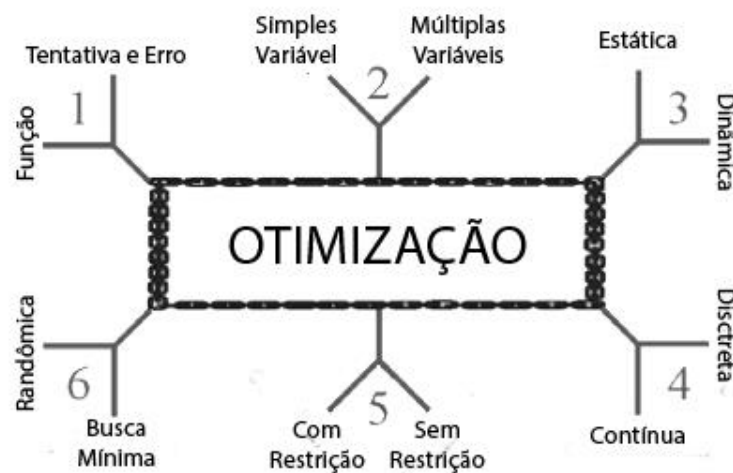


Figura 2.2 Categorias de Otimização

Fonte Traduzido de Haupt et al. (2004 p. 3)

O processo de otimização por tentativa e erro se refere ao ajuste das variáveis que afetam a saída sem conhecer exatamente os processos que produzem os valores de saída. Se o problema a ser estudado possuir somente uma variável, ele é um problema unidimensional, caso contrário, multidimensional. Otimização dinâmica é quando os valores de saída são obtidos em função do tempo e pode ser empregado em estudos relacionados à economia ou na análise da variação de uma área em uma viga prismática ao longo do seu comprimento em problemas estruturais. As variáveis podem ter um número finito de valores aceitáveis, isto é, discretas, ou um número

infinito de valores, no caso, contínuas. Tais variáveis podem ou não ter limites, isto é, restrições.

Alguns algoritmos tentam minimizar o custo, partindo de um ponto inicial, utilizando um conjunto de valores de variáveis admissíveis. Esses métodos são conhecidos como métodos tradicionais de otimização, geralmente baseados no cálculo diferencial, e se enquadram na categoria de busca mínima. Há ainda os métodos de busca aleatória, que utilizam cálculos probabilísticos para encontrar os valores das variáveis.

O problema dos métodos de busca mínima é que eles podem levar a ótimos locais e não globais, são apenas aplicáveis a fenômenos bem comportados e dependem da existência de derivadas e de continuidade da função objetivo. Para domínios onde há muitas descontinuidades, grandes espaços multimodais ou ruídos, eles são pouco robustos.

Não existe um método único disponível para resolver todos os problemas de otimização eficientemente (RAO, 2009). Uma série de métodos de otimização foram desenvolvidos para resolver diferentes tipos de problemas ao longo dos anos. Há, como exemplo, um método chamado pesquisa operacional: um ramo da matemática preocupado com a aplicação de métodos técnicos e científicos para tomada de decisão e cuja finalidade é encontrar as melhores soluções ou as soluções ótimas.

O tema pesquisa operacional surgiu nos anos 40 conforme aponta Ghiani et al. (2014), durante o período da Segunda Guerra Mundial. Durante a guerra, os militares britânicos enfrentaram o problema de alocação de recursos, que eram muito escassos e limitados: a quantidade de aviões de caça, radares e submarinos para diversas atividades eram pequena (devido à quantidade numerosa de objetivos e destinos). Ainda não havia métodos sistemáticos disponíveis para resolver tais problemas. Assim, os militares chamaram uma equipe de matemáticos para desenvolver métodos para resolver o problema de uma forma científica. Os métodos desenvolvidos pela equipe foram fundamentais para o sucesso da Grã-Bretanha durante a guerra. Estes métodos, tais como a programação linear, foram desenvolvidos como resultado de pesquisa (militar) operacional. Posteriormente, o método ficou conhecido como pesquisa operacional.

2.1.1 Definição de um problema de otimização

Simon (2013) define que um problema de otimização pode ser definido como:

$$\text{Encontrar } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ que minimiza ou maximiza } f(\mathbf{X})$$

Sujeito às restrições:

$$\begin{aligned} g_j(\mathbf{X}) &\leq 0, & j &= 1, 2, \dots, m \\ l_j(\mathbf{X}) &= 0, & j &= 1, 2, \dots, p \end{aligned} \tag{2.1}$$

Onde $\mathbf{X}$ é um vetor n -dimensional denominado vetor de projeto, $f(\mathbf{X})$ é um termo da função objetivo, e $g_j(\mathbf{X})$ e $l_j(\mathbf{X})$ são conhecidas como restrições de desigualdade e igualdade, respectivamente. O número de variáveis n e o número de restrições m não necessita estar relacionados. O problema mencionado pela Equação 2.1 é chamado de problema de otimização com restrição. Alguns problemas em que não envolvem restrições podem ser definidos como:

$$\text{Encontrar } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ que minimiza } f(\mathbf{X})$$

2.1.2 Termos imprescindíveis para um estudo de otimização

No estudo da otimização, os termos variáveis de projeto, variáveis de estado e restrição de superfície são imprescindíveis para o entendimento:

As variáveis de projeto (VP) são quantidades independentes que são alteradas com o objetivo de obter um projeto ótimo. Limites superiores e inferiores são especificados para servir como “contorno” para as variáveis de projeto. As variáveis de projeto são representadas como um vetor $X = \{x_1, x_2, \dots, x_n\}^T$. Exemplos são: posição, espessura, orientação, etc. (WIDMAIER, 2005 p. 65).

Já as variáveis de estado (VE) são quantidades que definem as especificações do projeto. Elas também são conhecidas como “variáveis dependentes”, e são tipicamente as quantidades resultantes de uma função das variáveis de projeto. Exemplos: tensão, deformação, critério de falha, etc. (WIDMAIER, 2005 p. 65).

As restrições de superfície estão relacionadas aos limites impostos ao estudo. Um problema de otimização com somente uma restrição de desigualdade $g_j(X) \leq 0$. O conjunto de valores de X que satisfazem a equação $g_j(X) = 0$ forma uma superfície no espaço de projeto e é chamado de restrição de superfície. A superfície é de subespaço dimensional $(n-1)$, onde n é o número de variáveis de projeto. As restrições de superfície dividem o espaço de projeto em duas regiões: uma no qual $g_j(X) < 0$ e outra no qual $g_j(X) > 0$. Assim, os pontos na superfície satisfarão as restrições $g_j(X)$ de modo crítico, os pontos na região aonde $g_j(X) > 0$ são inaceitáveis e os pontos que atendem a $g_j(X) < 0$ são aceitáveis. O conjunto com todas as restrições de superfície $g_j(X) = 0$, $j=1, 2, \dots, m$, que separa as regiões aceitáveis são chamadas de superfícies de restrição composta (RAO, 2009 p. 8).

Os termos apresentados pelo autor op.cit., apesar de ser tratados cada um de acordo com a sua especificidade, corroboram para a eficácia de um processo de otimização.

A Figura 2.3 mostra uma situação hipotética com as variáveis de projeto no plano. As regiões, em que as restrições definidas não apresentam a solução, estão hachuradas. Uma posição em uma ou mais superfícies de restrição é chamada de ponto limite e as restrições associadas são chamadas de restrições ativas. Os pontos onde não há nenhuma restrição são chamados de pontos

livres. Dependendo das condições impostas, um ponto pode ser aceitável ou não. De um modo geral, um ponto pode ser:

1. Livre e aceitável;
2. Livre e inaceitável;
3. Restrito e aceitável;
4. Restrito e inaceitável.

Todos os pontos e condições estão definidos na Figura 2.3:

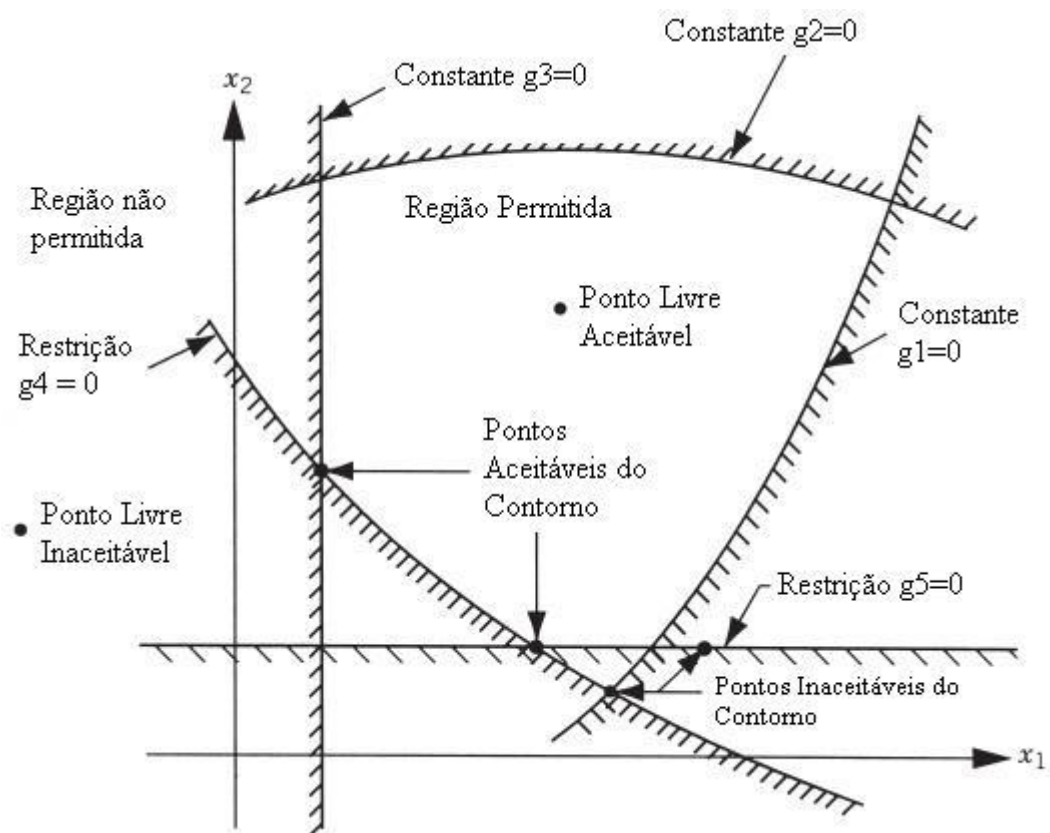


Figura 2.3 Restrições superficiais em um domínio hipotético no plano

Fonte Adaptado de Rao (2009 p. 8)

As restrições de superfície definem a função objetivo, que é tratado de acordo com a natureza do problema.

O procedimento convencional de um projeto é encontrar uma condição aceita ou um projeto adequado que satisfaz as condições definidas previamente. Em geral, serão aceitas mais do que uma condição. O propósito da otimização é a escolha da melhor situação dentre as condições aceitas. Assim, um critério deve escolher, comparando diferentes alternativas aceitas e selecionar a condição ótima. A escolha da função objetivo é governada pela natureza do problema. A função objetivo para minimização é definida como sendo uma função capaz de reduzir (o peso de um avião, ou a estrutura de uma aeronave) dadas condições de projeto (WIDMAIER, 2005). Em projeto de estruturas na área civil, a minimização do custo em função de seu peso (VANDERPLAATS, 2007). A maximização da eficiência mecânica é uma escolha óbvia em um sistema mecânico (SONMEZ, 2007). Por isso, a escolha de uma função objetivo depende dos problemas a serem resolvidos. Por exemplo, em um projeto mecânico, uma caixa de câmbio transmitindo a potência máxima pode não ter o peso mínimo (GOLOGLU; ZEYVELI, 2009). Similarmente em um projeto estrutural, um projeto com o peso mínimo pode não corresponder às tensões mínimas de projeto (NGIM et al., 2006). Assim, a seleção de uma função objetivo pode ser uma das mais importantes decisões para a busca de um projeto ótimo.

Um problema de otimização que envolve múltiplos objetivos é conhecido como problema multiobjetivos. Com múltiplos objetivos, há a possibilidade de que a função entre em conflito (objetivos incompatíveis). Isso pode fazer com que o problema não permita a obtenção de resultados satisfatórios.

Uma vez que as superfícies da função objetivo são desenhadas ao longo das restrições das superfícies, um ponto ótimo pode ser determinado sem muitas dificuldades. Um problema surge quando o número de variáveis de projeto excede duas ou três. Nesse caso, as restrições e as superfícies da função objetivo se tornam complexas para visualização e este tem que ser resolvido puramente através de equações matemáticas. A Figura 2.4 ilustra tal situação. Nela, podem ser vistas as restrições do problema sendo representadas pelas linhas hachuradas. Os valores encontrados em cada iteração até o encontro o ponto ótimo também são mostrados.

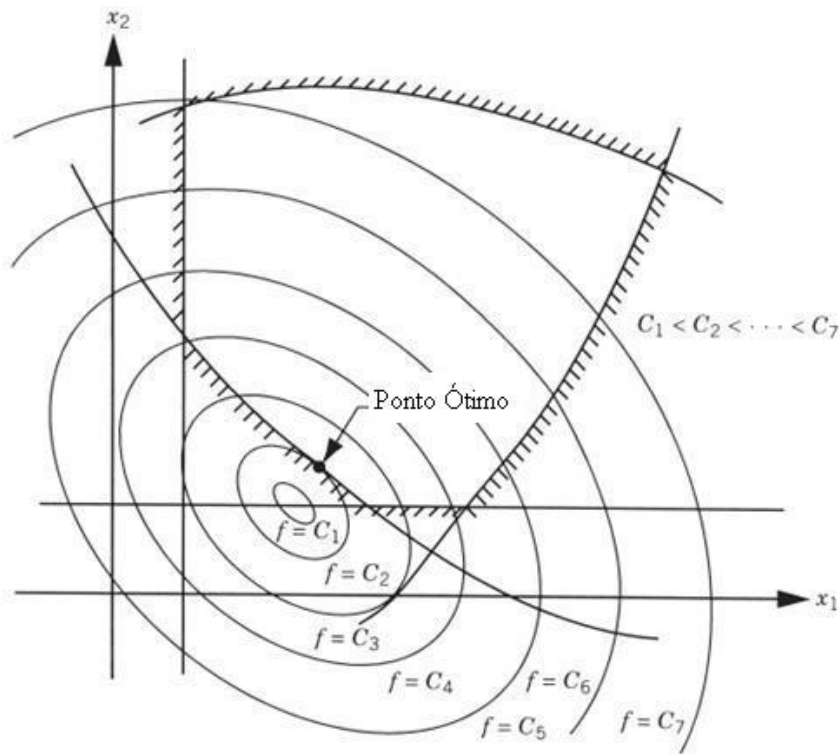


Figura 2.4 Contorno de uma função objetivo

Fonte Adaptado de Rao (2009 p. 10)

Cada valor de C corresponde a um membro da família de superfícies, isto é, corresponde a um conjunto de valores em uma determinada iteração. Tais superfícies são chamadas de superfícies da função objetivo. A cada iteração tais superfícies se alterarão em função de seus valores, podendo ou não estar dentro do domínio. Para o estudo apontado na Figura 2.4, é mostrado um estudo em um espaço bidimensional.

2.1.3 Classificação dos problemas de otimização

Os problemas de otimização podem ser classificados, segundo a existência de restrições, segundo a natureza das variáveis de projeto e segundo a continuidade da função objetivo entre outras características, detalhadas a seguir.

Quanto à classificação baseada na existência de restrições Hapt (2004), Vanderplaats (2007) e Rao (2009) ressaltam que qualquer problema de otimização pode ser classificado como restrito ou irrestrito, dependendo das restrições existentes no problema. Projetar uma coluna tubular engastada sujeita aos esforços de flambagem, sendo que seu diâmetro e sua espessura influenciam diretamente no custo do projeto, é um exemplo de problema de otimização restrito.

Para classificação com base na natureza das variáveis de projeto, os problemas de otimização podem ser classificados em duas categorias baseada na natureza das variáveis encontradas no projeto. Na primeira categoria, o problema é encontrar os valores com relação a um conjunto de parâmetros do projeto que fazem que alguma função destes parâmetros atinja mínimos para determinadas restrições. Por exemplo, o problema do peso mínimo de uma barra prismática sujeita a uma limitação de deflexão máxima. Problemas definidos por situações com estas características são chamados de problemas de otimização estática. Na segunda categoria, o objetivo é encontrar as variáveis de projeto no qual as funções contínuas dependem de outros parâmetros para minimizar a função objetivo, sujeito a um conjunto de restrições. Por exemplo, otimizar uma barra engastada em que a largura e o comprimento são função da espessura, de modo que a seção transversal pode variar ao longo do seu comprimento. Problemas com essas características são conhecidos como problemas de otimização dinâmica (VANDERPLAATS, 2007).

Já para a classificação com base na estrutura física do estudo, problemas de otimização podem ser classificados como de controle ótimo e de controle não ótimo (DENG et al., 2013).

Problema de controle ótimo: Um problema de controle ótimo é um problema que envolve uma série de etapas, onde cada fase se desenvolve a partir das anteriores de uma forma sequencial.

Geralmente, é descrito por dois tipos de variáveis:

As de controle (projeto) e as de estado.

As variáveis de projeto definem o sistema que governa a evolução de um estágio para o outro e as variáveis de estado descrevem o comportamento ou o estado do sistema para qualquer estágio. O problema é encontrar um conjunto de controle ou variáveis de projeto que atenda a função objetivo em cada etapa. A minimização está sujeita a um conjunto de restrições sobre o controle e as variáveis de estado. Exemplo: Um foguete é projetado para viajar uma distância de

12s na direção vertical. O empuxo do foguete somente poderá ser alterado em pontos discretos localizados na distância correspondente ao tempo de 0s, 2s, 3s,... 12s, o objetivo é minimizar o tempo, adotando as seguintes restrições:

1. O foguete viaja contra a força gravitacional.
2. A massa do foguete reduz-se proporcionalmente à distância percorrida.
3. A resistência do ar é proporcional à velocidade do foguete.

Problema de controle não-ótimo: Um problema de controle não-ótimo é um problema em que não envolve etapas sequenciais para a resolução do problema a ser estudado.

Quanto a classificação dos problemas relacionados a otimização, esta é baseada na natureza das equações envolvidas, nas expressões para a definição da função objetivo e o tipo de restrição. De acordo com essa classificação, problemas de otimização podem ser classificados como lineares, não-lineares, geométricos e problemas de programação quadrática. Esta classificação é extremamente útil do ponto de vista da computação, pois existem muitos métodos disponíveis para uma solução eficiente para uma série de tipos de problemas classificados dessa forma. Assim, a primeira tarefa será investigar a classe do problema encontrado. Isto, em muitos casos, define os tipos de procedimentos da solução a ser adotada para o problema (RAO, 2009).

Problemas de otimização linear. Se a função objetivo e todas as restrições são funções lineares das variáveis de projeto, o problema é chamado de problema de otimização linear.

O comportamento do fenômeno ou as restrições podem ter características não lineares. Este é o problema mais geral de otimização e todos os outros podem ser considerados como casos especiais deste problema (HAFKA, 1992). Para problemas não lineares, existem diversos métodos e técnicas para sua resolução. Tais métodos são conhecidos como métodos numéricos e podem ser divididos em métodos de eliminação e métodos de interpolação (ANTONIOU; LU, 2007). O método de eliminação pode ser usado para a minimização de uma função descontínua, por exemplo, método de Fibonacci e o método de busca irrestrita são ferramentas para a resolução deste tipo de problema. O método de interpolação consiste em uma aproximação da função objetivo de modo polinomial. Esta pode ser quadrática ou cúbica (ou de outra ordem) e

pode ou não necessitar de derivadas, dependendo do método adotado. O tipo do problema é o que vai definir qual é o melhor algoritmo a ser adotado.

Em problemas mais complexos, que exigem uma quantidade maior de variáveis, existem outras ferramentas que podem ser classificadas como método direto, nos quais as restrições são manuseadas de forma explícita, e método indireto, para os quais um problema com restrição é resolvido como uma sequência de problemas de minimização sem restrição. O software que será empregado nesta dissertação, o Ansys[®], o software adota o método indireto em uma de suas ferramentas voltadas a estudos de otimização. O método é conhecido como aproximação por subproblema.

Problemas de programação geométrica é a classificação dada a problemas que utilizam um determinado método para resolver problemas não-lineares (RAO, 2009). A diferença em relação ao método apresentado anteriormente é que este método foca na solução da função objetivo, enquanto os outros focam na solução das variáveis da função objetivo.

Outro método é conhecido como programação quadrática. Este método, também é voltado para problemas não-lineares necessita das derivadas da função objetivo para a resolução do problema.

Classificação com base nos valores admissíveis das variáveis de projeto, Iqbal (2013) menciona que dependendo dos valores permitidos para as variáveis de projeto, os problemas de otimização podem ser classificados como de valores inteiros e ou reais.

Problema de otimização inteiro. Se algumas ou todas as variáveis de projeto $x_1, x_2, \dots, x_n$ de um problema de otimização devem assumir apenas valores inteiros (ou discretos), o problema é chamado de um problema de otimização inteiro. Por outro lado, se todas as variáveis do projeto podem assumir qualquer valor real, o problema de otimização é chamado um problema de programação de valor real.

A classificação com base na natureza determinística das variáveis, Norris (2009) afirma que com base na natureza determinista das variáveis envolvidas e problemas de otimização podem ser classificados como problemas de programação determinísticos ou estocásticos.

Problema de otimização estocástica: É um problema de otimização no qual todas as variáveis ou alguns parâmetros (variáveis de projeto e/ou parâmetros pré-atribuídos) são probabilísticos, isto é, não determinísticos ou estocásticos.

Já, para os problemas determinísticos, não há variáveis com características probabilísticas, isto é, é suposto que uma propriedade (ex. o módulo de elasticidade) seja constante em diversas amostras. Na realidade, o módulo de elasticidade possui características estocásticas, entretanto para estudos com simulação numérica normalmente é considerado como determinístico.

De acordo com Vanderplaats (2007), a classificação também pode ser baseada na separação das funções. Os problemas de otimização podem ser classificados como de programação separável e não separável baseado na separação das funções objetivo e das restrições. Uma função $f(X)$ é dita separável se ela pode ser expressa pela somatória de funções com uma variável, $f_1(X_1)$, $f_2(X_2)$,..., $f_n(X_n)$. Por exemplo: Uma loja armazena e vende três tipos de televisores, onde cada televisor tem um custo para compra e uma taxa de demanda, se cada televisor possui uma função que representa seu custo e seu retorno. Qual é a quantidade de televisores para que haja o mínimo de estoque?

A classificação pode ser com base no número de funções objetivo a serem minimizadas, problemas de otimização podem ser classificados como problemas de otimização simples ou multiobjetivo.

O problema sugerido para o trabalho dessa dissertação se enquadra como multiobjetivo, pois, a função a ser otimizada deverá atender a um conjunto de funções e restrições. O problema multiobjetivo pode ser descrito como:

Encontrar um $\mathbf{X}$ que minimize $f_1(\mathbf{X})$, $f_2(\mathbf{X})$,..., $f_k(\mathbf{X})$ sujeito à:

$$g_j(\mathbf{X}) \leq 0, \quad j=1, 2, \dots, m$$

Onde $f_1, f_2, \dots, f_k$ denotam as funções objetivo a serem minimizadas.

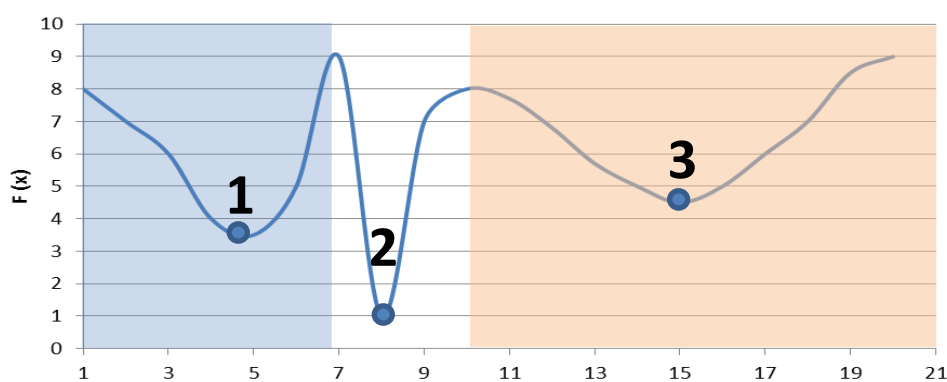
2.2 Técnicas de otimização

Widmaier (2005) menciona que o processo de localização do ponto ótimo global é normalmente, um processo árduo, requerendo procedimentos sofisticados. Em casos típicos de problemas de otimização estrutural, pode haver várias configurações mínimas locais. Utilizando o algoritmo denominado *downhill*, no qual o valor da função objetivo diminui de forma monotônica, pode ocorrer de o algoritmo encontrar um novo mínimo local e não o mínimo

global, já que os valores são interativamente criados pelo software a partir de um dado de entrada inicial do usuário. Desta forma, o sucesso depende da escolha correta dos parâmetros de entrada. O procedimento usual para resolver a questão é empregar o algoritmo, muitas vezes, iniciando de configurações diferentes com a expectativa de que a posição inicial seja suficientemente perto da configuração mínima global. Como muitos valores locais, pode-se escolher o menor valor, que poderá ser a solução mínima global. Uma desvantagem é que, dependendo do método de busca utilizado, se o ponto de partida inicial escolhido estiver fora da região permitida para os parâmetros a serem otimizados, o algoritmo pode convergir para um resultado fora do domínio estabelecido. A Figura 2.5 mostra uma função com os seus mínimos locais e globais. A Tabela 2.1 mostra a possibilidade de resposta em função do método adotado.

Embora exista diversas técnicas de otimização, como foi dito anteriormente, ainda não há uma única ferramenta que atenda plenamente todos os tipos de problemas. Nos últimos anos, diversos novos métodos de otimização foram implementados, que são conceitualmente diferentes das técnicas de programação matemática até então desenvolvidas. Estes métodos são chamados de métodos modernos ou métodos não-tradicionais de otimização.

Os métodos acima mencionados estão se tornando cada vez mais populares para a solução de problemas de engenharia. A principal característica destes métodos é que eles requerem somente os valores da função objetivo, isto é, não depende da derivada das variáveis a serem estudadas, características de um método tradicional de otimização.



Pontos 1 e 3 – Mínimo Local

Ponto 2 – Mínimo Global

Figura 2.5 Comportamento de uma função abstrata, definindo seus mínimos locais e globais

Tabela 2.1 Possibilidade de convergência em função da escolha do método

Tentativa Inicial	Métodos Tradicionais	Métodos não-tradicionais (RS, AG entre outros)
Região Azul	Converge em 1	Converge em 2
Região Rosa	Converge em 3	Converge em 2
Região Branco	Converge em 2	Converge em 2

Os métodos clássicos de cálculo diferencial podem ser usados para encontrar os máximos e os mínimos de uma função com diversas variáveis irrestritas. Tais métodos presumem que a função é diferenciável com relação às variáveis de projeto e as derivadas são contínuas (HAPT, 2004). Para problemas com restrições de igualdade, o método dos multiplicadores de Lagrange pode ser usado. Se o problema tem restrições de desigualdade, as condições de Kuhn-Tucker podem ser usadas para identificar o ponto ideal, porém estes métodos levam a um conjunto de equações simultâneas não-lineares que podem ser difíceis de resolver (RAO, 2009).

As técnicas de programação não-linear, linear geométrica, quadrática, ou inteira podem ser usadas para a solução de uma determinada classe de problemas, conforme indicado pelo nome da técnica. Entretanto, a maioria destes métodos leva a uma solução aproximada, que é procurada pelo processo de forma iterativa, começando a partir de uma solução inicial. Entre os métodos modernos de otimização incluem-se o recozimento simulado, a otimização por enxame de partículas, o método chamado de Busca Tabu, a otimização de colônia de formigas, a otimização baseada em redes neurais, a otimização fuzzy e a otimização estrutural evolucionária. Nesta última, podem-se destacar os algoritmos genéticos. Atualmente, esses são os métodos mais estudados (SIMON, 2013). Uma das principais características, destes métodos, é que eles não são sensíveis ao ponto de início.

A utilização de ferramentas de otimização não-tradicionais para resolução de problemas de otimização de forma em conjunto com o método dos elementos finitos é um problema muito estudado atualmente (SONMEZ, 2007).

2.3 Otimização de forma x otimização topológica

Em problemas de otimização de forma o objetivo é encontrar a forma ideal dado um determinado componente. A forma do problema é definida em um domínio no qual os vértices são as variáveis de projeto. Deste modo, os vértices poderão deslocar-se a fim de se obter a solução ótima, dada uma determinada condição de contorno.

Já a otimização topológica de uma estrutura sólida envolve a determinação de características tal como o número e a localização de furos ou as conectividades do domínio a fim de encontrar a melhor geometria (topologia). A otimização topológica tem como objetivo encontrar uma geometria ótima onde os elementos são as variáveis de projeto.

Em outras palavras, o objetivo da otimização de forma é encontrar a geometria ideal para o problema a ser estudado utilizando os vértices como variáveis. Já a otimização topológica envolve a adição ou remoção de elementos a serem estudados. Desta forma, a estratégia consiste na remoção de material não útil até o perfil de a estrutura chegar a um produto ótimo (DAS, 2005). Assim, não somente o contorno é otimizado, mas também são permitidas alterações em sua região interna conforme Figura 2.6.

O método conhecido como Material Isotrópico Sólido com Penalização foi um marco para estudos de otimização topológica e é um dos mais utilizados atualmente nesta área. O método foi desenvolvido por Bendsøe e Sigmund no ano de 2003 (LIN et al., 2009). Neste método, os elementos da malha desenvolvida pelo método dos elementos finitos são as variáveis do estudo, do modo que, a cada iteração alguns elementos são removidos com o intuito de se obter o menor volume.

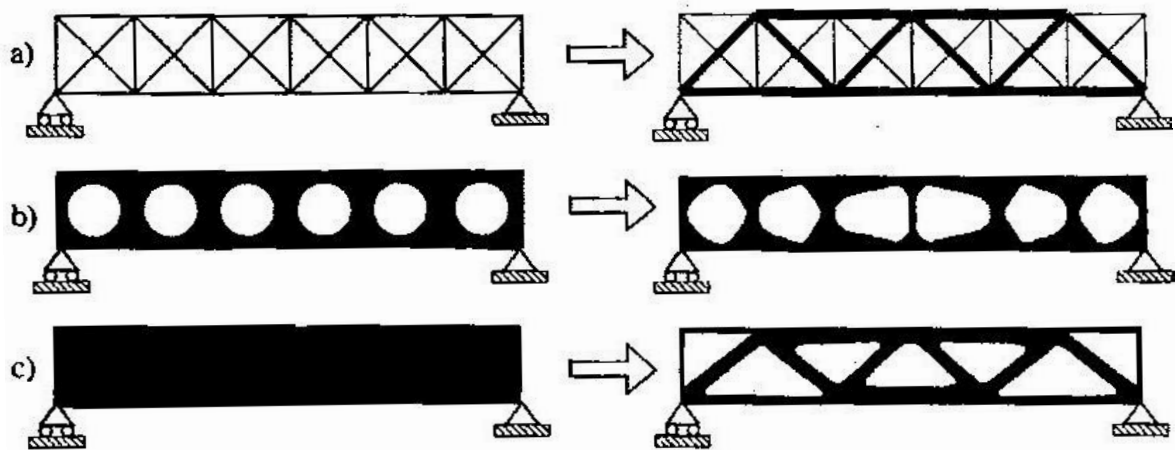


Figura 2.6 As três categorias de otimização estrutural

Fonte Bendsøe e Sigmund (2003 p. 2)

Na Figura 2.6, os problemas iniciais são definidos, ao lado esquerdo; já, ao lado direito, é a solução ótima para cada tipo de problema em cada categoria de otimização onde a) Otimização convencional de uma estrutura treliçada b) Otimização de Forma e c) Otimização Topológica.

Em um estudo de otimização estrutural convencional, a estrutura é desenvolvida em função das adições ou remoções de barras, assim as barras são as variáveis. Em um estudo de otimização de forma, a perfil externo não poderá variar analisando o problema da Figura 2.6. Entretanto, o perfil do alívio poderá sofrer alterações, pois as variáveis são os vértices que formam o perfil interno. Na otimização topológica, os elementos (em um estudo por elementos finitos) são adicionados ou removidos em função da solicitação do esforço em que cada elemento sofre. Desta forma, os elementos que não estão sofrendo esforços, poderão ser removidos.

Em se tratando de otimização de forma, que está relacionado ao desenvolvimento deste trabalho de mestrado, o que se busca é encontrar o mínimo absoluto de uma função objetivo, sem depender da definição de um valor inicial. Para isso, uma ferramenta de otimização que tem sido empregada em problemas de otimização estrutural é através do emprego de técnicas estocásticas.

Nesse método algumas variáveis, ou todas, são descritas como estocásticas ou aleatórias. Desta forma, definido os limites dos valores de cada variável, seus valores são definidos a cada

iteração de modo aleatório. Este método se comporta muito bem para problemas estruturais, tendo como principais vantagens:

- Não ser sensível à definição de um ponto de início;
- Sua facilidade para levar à solução da função objetivo em um grande domínio, devido à sua característica de movimentos, isto é, versatilidade para escapar dos mínimos locais ao longo da função objetivo;
- Não haver necessidade de derivação;
- Facilidade de resolver problemas com grandes descontinuidades e com elevados valores nos limites das variáveis.

2.4 Métodos dos Elementos Finitos voltados à otimização

As limitações da mente humana são tais que não se pode compreender o comportamento de uma operação complexa de forma ampla. Com efeito, o processo de subdividir todo o sistema em componentes menores (elementos), cujo comportamento é de fácil compreensão, e, em seguida, reconstruir o sistema original com tais componentes para estudar o seu comportamento, é a estratégia que o engenheiro, o cientista e mesmo o economista usam (ZIENKIEWICZ, 2005).

Em muitas situações, um modelo adequado é obtido utilizando um número finito de componentes, isto é, discretizando um domínio (COOK, 2007). Em outros, a subdivisão do problema só pode ser definida usando o conceito da matemática infinitesimal. Isto leva a equações diferenciais, que implicariam em um número infinito de elementos. Este problema se enquadra como um sistema contínuo. Com os novos computadores, problemas discretos podem ser resolvidos facilmente, mesmo se o número de elementos for muito grande. Como a capacidade de todos os computadores é finita, problemas contínuos só podem ser resolvidos através de manipulações matemáticas. Assim, técnicas matemáticas para soluções exatas geralmente são limitadas a problemas simples ou análises muito simplificadas de problemas complexos.

Para superar a dificuldade com problemas contínuos, vários métodos de discretização foram desenvolvidos ao longo do tempo. Todos envolvem uma aproximação que, espera-se, através de variáveis discretas, represente adequadamente um problema contínuo, dentro de um limite de erro aceitável (LI; MULAY, 2013).

A discretização de problemas contínuos tem sido abordada de forma diferente por matemáticos e engenheiros (ZIENKIEWICZ, 2005). Matemáticos desenvolveram técnicas gerais aplicáveis diretamente às equações diferenciais que regem o problema, como aproximações por diferenças finitas, ou técnicas para determinar um conjunto de funções a partir de uma função, cuja área da matemática é conhecida como cálculo variacional, cujos termos das funções são definidos como *funcionais*.

O engenheiro, por outro lado, muitas vezes, aborda o problema de uma forma mais intuitiva, criando uma analogia entre elementos discretos de modo real em regiões finitas de um domínio contínuo. Por exemplo, no campo da mecânica dos sólidos McHenry et al. (1940) apud (ZIENKIEWICZ, 2005) mostraram que soluções adequadas para um problema contínuo de modo elástico podem ser obtidas através da substituição de pequenas regiões do contínuo por um conjunto de barras elásticas simples. Mais tarde, Turner et al. (1956) apud (BELYTSHKO, 2009) mostraram que a subdivisão de pequenos componentes, para a realização de um estudo e seus esforços inseridas em uma matriz, pode ser muito eficaz.

É a partir deste ponto de vista que nasceu o termo *elementos finitos*. Clough (1960) parece ser o primeiro a usar este termo (LOGAN, 2007).

Após o início dos anos 60, o progresso do MEF ocorreu de forma vertiginosa. Hoje, têm-se métodos e conceitos muito eficazes para os mais diversos segmentos (BELYTSHKO, 2009). De um modo geral, o método dos elementos finitos pode ser definido como um procedimento de discretização de um problema contínuo definido por declarações matemáticas que regem um determinado problema. Existem outros métodos de discretização como, por exemplo, o método de elementos de contorno, Bubnov-Galerkin, e àqueles em que não há necessidade de discretização como, por exemplo, o método chamado Free-Galerkin, o método dos elementos difusos, entre outros (LI; MULAY, 2013). Porém, o método dos elementos finitos ainda é o método mais utilizado atualmente, tanto no meio acadêmico como no meio industrial.

O engenheiro, quando lida com estruturas, primeiro calcula as forças e os deslocamentos

para cada elemento da estrutura e então começa a montar o todo, seguindo um procedimento bem definido, que consiste em estabelecer o equilíbrio local em cada *nó* ou ponto de conexão da estrutura. As equações resultantes podem ser resolvidas para os deslocamentos desconhecidos. Da mesma forma, os engenheiros elétricos ou hidráulicos, lidando com uma rede de componentes elétricos (resistores, capacitâncias, etc) ou dutos hidráulicos, primeiro estabelecem uma relação entre correntes (fluxos) e potenciais de elementos individuais e então começam a montar o sistema, garantindo a continuidade dos fluxos conforme Figura 2.7.

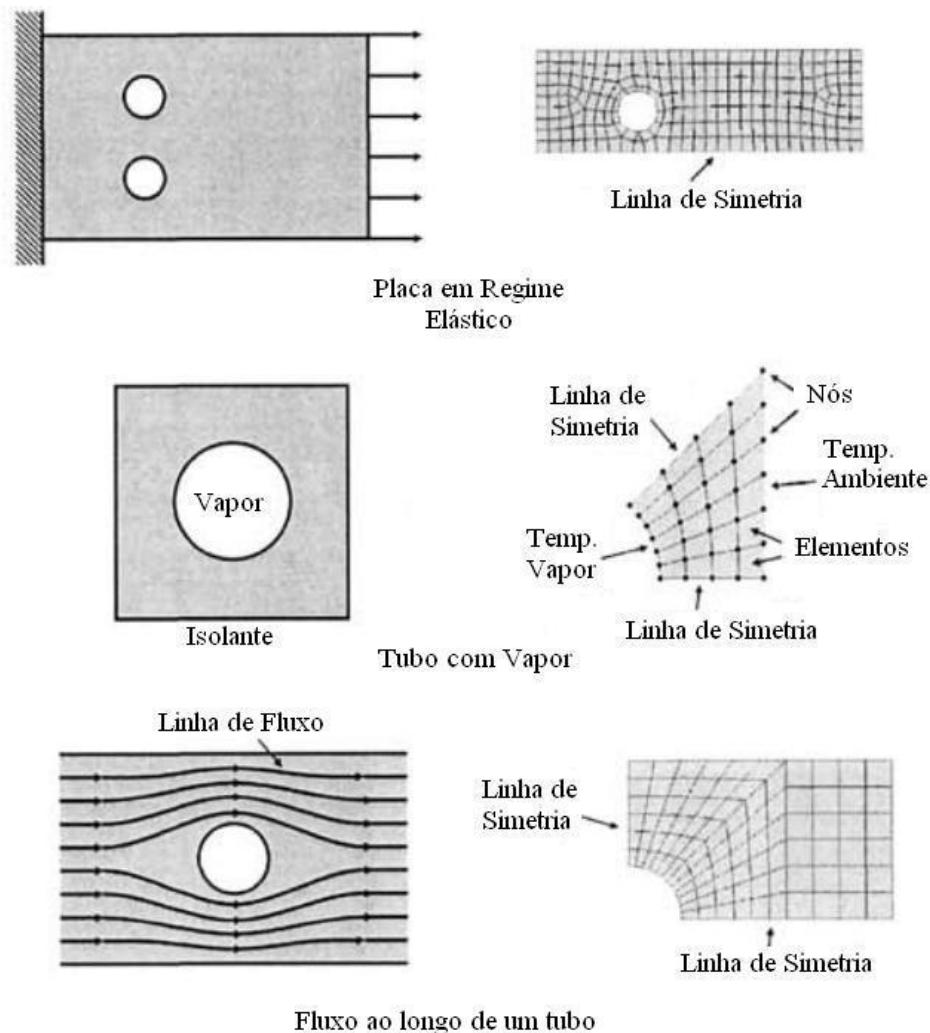


Figura 2.7 Representação computacional de alguns problemas de engenharia

Fonte Madenci (2006 p. 2)

A Figura 2.8 mostra como ocorreu evolução dos métodos de aproximação. Como apontado anteriormente, esta dissertação utilizará o MEF em seu desenvolvimento, embora os conceitos de otimização propostos possam ser empregados com outros métodos, com adaptações.

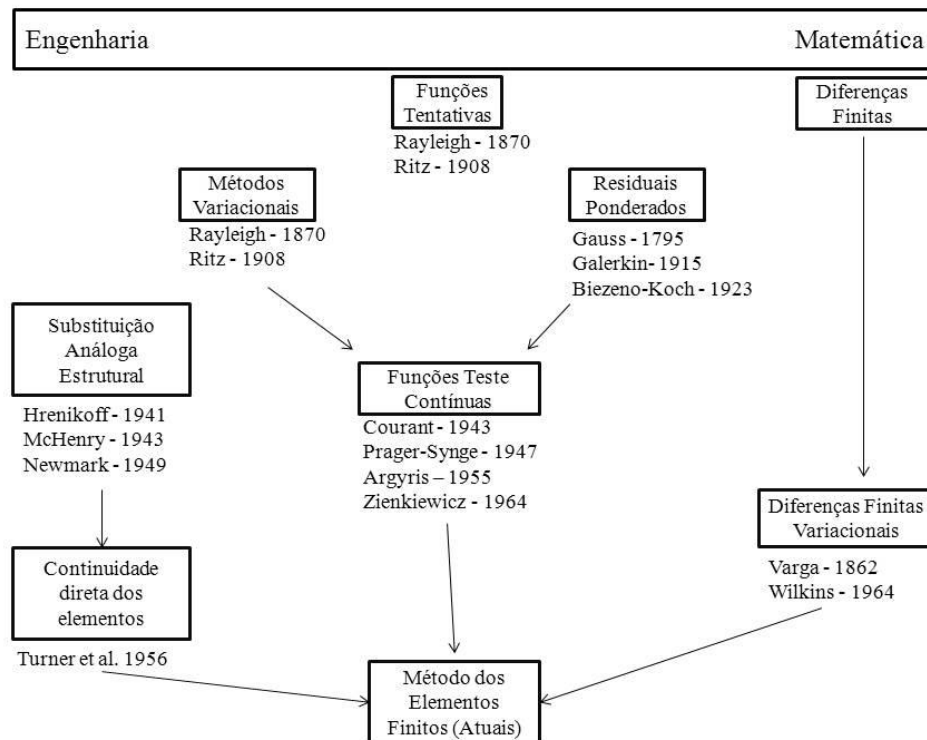


Figura 2.8 Evolução das ferramentas matemáticas para o MEF

Fonte Zienkiewicz (2005 p. 3)

A ideia básica do MEF é dividir o domínio em elementos finitos, ou apenas *elementos*, ligado por nós, e obter uma solução aproximada como mostrada na Figura 2.9. O conjunto de nós e elementos são chamados de malha de elementos finitos e o processo de fazer a malha é chamado de geração de malha ou discretização.

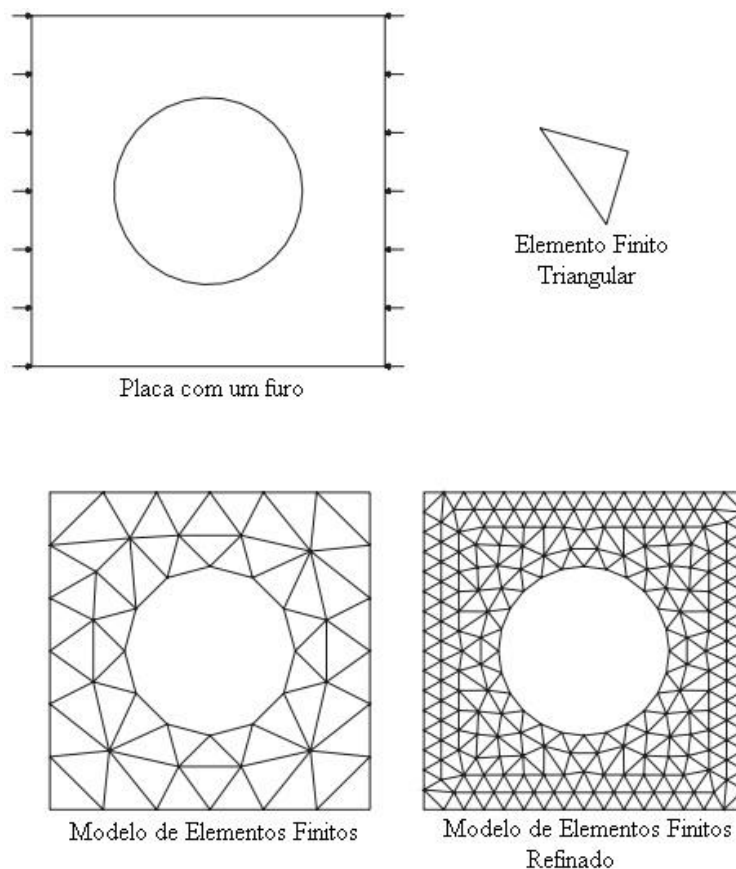


Figura 2.9 Geometria, carga e a malha gerada pelo MEF

Fonte Adaptado de Belytshko (2009 p. 2)

O MEF fornece uma metodologia sistemática pela qual as soluções que, no caso da Figura 2.9, é uma distribuição de temperaturas, podem ser determinadas por um programa de computador. Para problemas lineares, a solução é determinada pela resolução de um sistema de equações lineares. O número de incógnitas (que são as temperaturas nodais neste exemplo) é igual ao número de nós. Para obter uma solução razoavelmente precisa, milhares de nós são geralmente necessários. Quase sempre, a precisão da solução melhora à medida que o número de elementos (e nós) aumenta, porém o tempo computacional, portanto o custo, também aumenta.

O programa de elementos finitos determina a temperatura em cada nó e o calor que flui através de cada elemento. Os resultados podem ser apresentados de forma gráfica e as regiões do domínio apresentam cores diferentes representando uma determinada escala de magnitudes. Com

isso, o analista consegue visualizar a distribuição de temperatura em cada nó.

A mesma abordagem é utilizada em outros tipos de problemas. Na análise de tensão, a variável é o campo de deslocamentos; em sistemas químicos, a variável de campo é a concentração de material; em eletromagnetismo, o campo potencial. O mesmo tipo de malha é usado para representar a geometria da estrutura ou componente e para desenvolver as equações de elementos finitos. Para sistemas lineares, os valores nodais são obtidos, resolvendo um grande número de equações algébricas de 10^3 a 10^6 são valores comuns, podendo em algumas situações chegar a 10^9 (BELYTSHKO, 2009).

A disponibilidade de uma ampla gama de capacidades de análise em um programa torna possível a simulação de muitos problemas complexos da vida real. Por exemplo, os fluxos de ar em torno de um carro e através do compartimento do motor podem ser obtidos por um programa direcionado a fluídos, cujo campo é chamado de Dinâmica Computacional dos Fluídos (CFD). Isto permite aos engenheiros e projetistas preverem o fator de arrasto ou o fluxo no compartimento do motor. O fluxo no compartimento do motor é então usado como base para cálculos de transferência de calor no bloco do motor e radiador.

As distribuições de temperatura e de rendimento podem ser combinadas com as cargas mecânicas, para obter uma análise de tensão no motor. Da mesma forma, na concepção de um computador ou um micro dispositivo, as temperaturas nos componentes podem ser determinadas por meio de uma combinação de análise de fluídos (para o ar que flui ao redor dos componentes) e o calor. As temperaturas resultantes podem então serem usadas para determinar as tensões nos componentes, como nas juntas de solda, que são cruciais para a vida do componente. O mesmo modelo em elementos finitos, com algumas modificações, pode ser usado para determinar os campos eletromagnéticos (BASKHARONE, 2014).

Widmaier (2005) aponta que no projeto de aeronaves, cargas a partir de cálculos CFD e testes de túnel de vento são usadas para prever as cargas mecânicas sobre a estrutura. Um modelo de elementos finitos é usado com milhares de situações de carga, que incluem cargas de vários tipos de operações como, pouso, decolagem e, assim por diante, para determinar as tensões na estrutura. Quase todas essas análises são lineares, apenas a determinação da capacidade de carga de ruptura de uma estrutura exige uma análise não-linear.

Processos de fabricação também são simulados por elementos finitos. Assim, a solidificação de fundidos é simulada para garantir a qualidade adequada do produto. No desenvolvimento de chapas e no processo de estampagem de produtos para carros e máquinas de lavar, o processo é simulado para garantir que a peça possa ser estampada e para verificar se, após retorno elástico, a peça ainda está em conformidade com as especificações dadas pelo projeto (NETO et al., 2008).

Procedimentos semelhantes se aplicam na maioria dos outros setores. De fato, é incrível como o MEF vem transformado o ambiente de trabalho da engenharia nos últimos 40 anos. Na década de 60, a maioria dos departamentos de projeto de engenharia consistia de uma sala com mesas de 1,5m x 3m em que os engenheiros desenvolviam o seu projeto com suas réguas T e uma série de outros instrumentos. O projeto era então estimado por fórmulas e conceitos simples, visto em mecânica dos sólidos, como taxa de alongamento, flexão e torção. Estas fórmulas ainda são úteis, especialmente para verificar as soluções em elementos finitos, porque se o elemento finito difere dessas fórmulas por uma ordem de magnitude, a solução de elementos finitos normalmente está errada. Para verificar a solidez de um projeto, protótipos eram desenvolvidos e testados. Os protótipos são usados, ainda hoje, porém somente nos últimos estágios de um projeto. Desta forma, o custo do protótipo foi reduzido significativamente. Assim, o MEF tem levado a expressivas reduções no tempo de ciclo de um novo produto e o uso eficaz dessas ferramentas é crucial para a competitividade do produto durante sua fase comercial. O texto apresentado, nesta dissertação, é limitado à análise linear pelo MEF.

Kim e Sankar (2009) mencionam que, em muitos projetos industriais, os dados de elementos finitos são um componente chave no desenvolvimento do produto porque são usados para um grande número de análises diferentes, embora, em muitos casos, a malha tenha que ser adaptada para aplicações específicas.

A interface do banco de dados dos elementos finitos gerados a partir do banco de dados CAD (*Computer Aided Design*) é uma estratégia adotada por muitas empresas fabricantes de programas especializados devido a sua versatilidade. Como exemplo, tem-se os programas Creo[®] e o Pro/Mechanica[®] ou Catia[®] e o Simulia[®]. Portanto, os sistemas de elementos finitos podem conter interpretadores, que geram malhas de elementos finitos a partir dos bancos de dados CAD. A necessidade de duas bases de dados é um dos principais gargalos nas análises

computadorizadas atualmente, porque os dois conjuntos de dados, muitas vezes, não são compatíveis (BELYTSHKO, 2009).

Segundo Madenci (2006), o método dos elementos finitos consiste dos seguintes passos para seu desenvolvimento:

- Discretização do domínio em um número finito de subdomínios (elementos);
- Seleção das funções de interpolação;
- Desenvolvimento da matriz de rigidez do elemento para o subdomínio definido;
- Montagem em uma única matriz, todas as matrizes dos elementos criadas anteriormente, esta definida como matriz de rigidez global;
- Imposição das condições de contorno;
- Solução das equações;
- Computações adicionais (gráficos, tabelas e curvas), caso necessário.

Segundo Cook (2007), os seguintes termos são imprescindíveis para o estudo em elementos finitos.

2.4.1. Nós

A transformação de problemas práticos de engenharia em representações matemáticas é desenvolvida quando o domínio de interesse é discretizado, isto é, o domínio é dividido em subdomínios (elementos). Estes elementos são conectados uns aos outros através de nós “comuns”. Um nó, especifica uma localização no espaço, onde os graus de liberdade e as ações dos problemas físicos existem.

Os nós em comum, conforme Figura 2.10, representam a continuidade das variáveis nodais (graus de liberdade). Os graus de liberdade do nó definem a natureza física do problema e o tipo de elemento. A Tabela 2.2 apresenta os graus de liberdade correspondente às *forças* usadas com o MEF para os mais variados tipos de problemas.

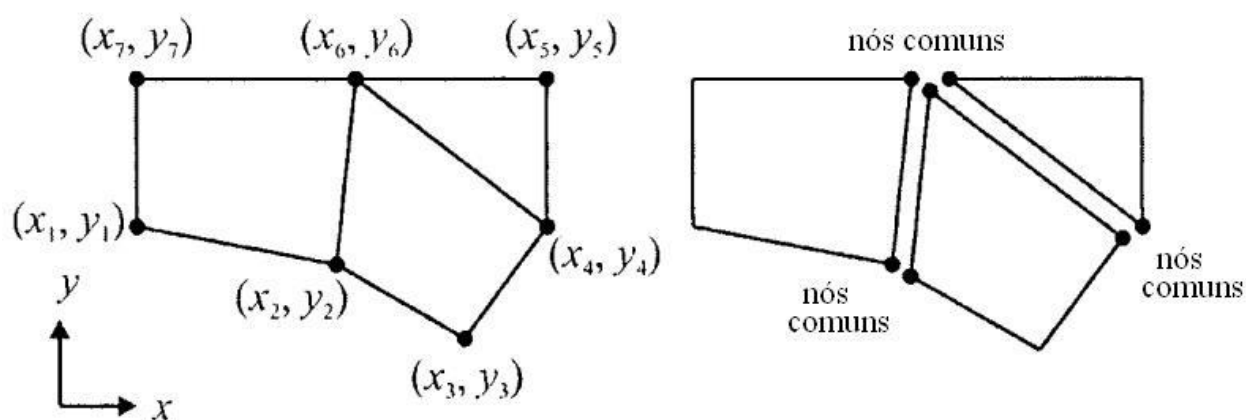


Figura 2.10 Divisão do domínio em subdomínios (elementos)

Fonte Traduzido de Madenci (2006 p. 3)

Tabela 2.2 Graus de liberdade e vetores de força em diferentes campos da Engenharia

Fonte Madenci (2006 p. 4)

Disciplina	Graus de Liberdade	Vetor de Forças
Mecânica dos Sólidos	Deslocamentos	Forças Mecânicas
Condução de Calor	Temperatura	Fluxo de Calor
Fluído Acústica	Deslocamento Potencial	Velocidade das Partículas
Fluxo Potencial	Pressão	Velocidade das Partículas
Fluxos Gerais	Velocidade	Fluxos
Eletroestática	Potencial Elétrico	Densidade de Carga
Magnetoestática	Potencial Magnético	Intensidade Magnética

2.4.2. Elementos

Dependendo da geometria e da natureza física do problema, o domínio de interesse pode ser discretizado, empregando linhas, áreas ou volumes. Cada elemento é identificado pelo número do elemento e é definido por uma sequência específica em relação aos números dos nós globais. Desta forma, cada nó e cada elemento são numerados a fim de que haja uma organização na montagem dos elementos na matriz global, permitindo o cálculo de todas as incógnitas necessárias para a solução do problema a Figura 2.11 demonstra os principais elementos utilizados.

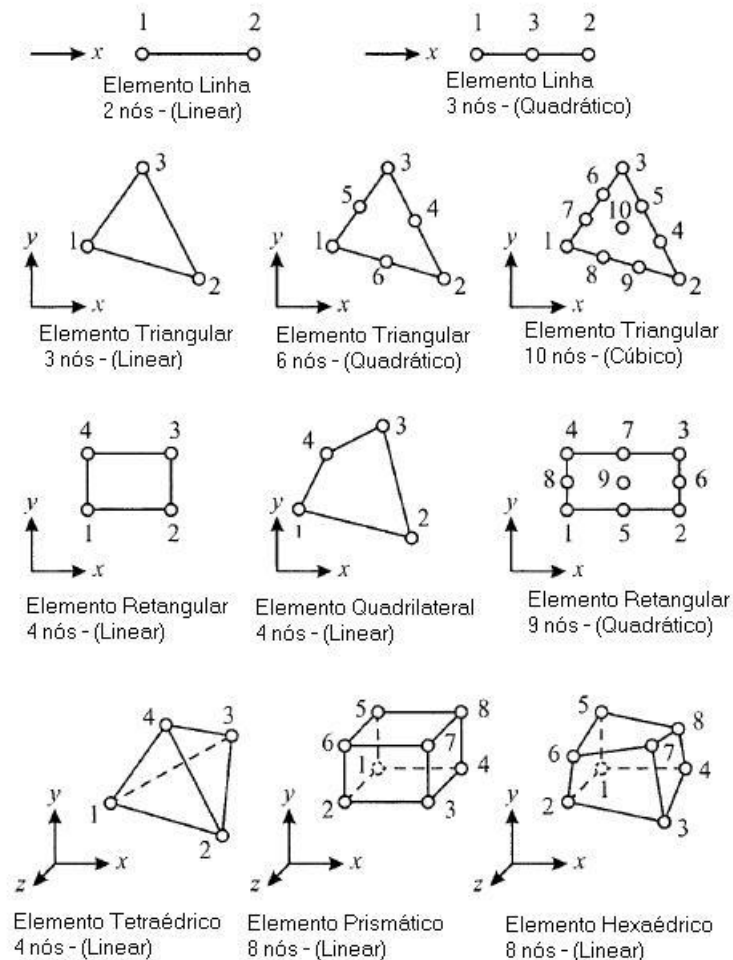


Figura 2.11 Descrição dos principais elementos e seus respectivos números nodais

Fonte Madenci (2006, p. 38)

Em um estudo de MEF, há a possibilidade de inserir elementos de baixa e alta ordem, ou mesmo elementos unidimensionais com tridimensionais. O conceito de MEF resume-se em preencher um domínio a ser estudado com elementos que melhor represente o comportamento físico do problema.

2.4.3. Funções de Aproximação

As funções de aproximação estão diretamente relacionadas à qualidade do resultado, pois cada elemento fará parte de um conjunto que corresponderá ao domínio a ser estudado. Dependendo da complexidade da geometria, do tamanho do elemento ou mesmo da ordem polinomial do elemento a serem adotados, os resultados podem ser muito diferentes, comprometendo o estudo.

Matematicamente, existem diversas ferramentas voltadas à aproximação de um domínio que podem ser aplicadas, utilizando o conceito do método dos elementos finitos. Os métodos de Galerkin e o Rayleigh-Ritz são os mais populares. Porém, o método de Galerkin é o mais utilizado devido a não necessidade de derivação das funcionais. Desta forma, tornando mais fácil a formulação do problema (NAKAZONE, 2006; KIM e SANKAR, 2009; BASKHARONE, 2014).

Elementos de baixa ordem (lineares) têm uma menor quantidade de incógnitas, sendo assim, o custo computacional tende a ser menor. Entretanto, elementos quadráticos ou cúbicos apesar de serem computacionalmente mais caros, tendem a levar a resultados mais precisos, pois a função que representa o deslocamento de um elemento possui um grau maior (uma ordem maior).

2.4.4. Métodos de Aproximação

De um modo geral, existem diversas ferramentas para o modelamento matemático

computacional de problemas físicos com o MEF (BASKHARONE, 2014). Os principais métodos são:

- **Aproximação Direta** – Esta ferramenta é usada para problemas relativamente mais simples, ou cujo objetivo é explicar os conceitos básicos do MEF. Cada elemento é tratado como se fosse uma mola linear e a rigidez da mola é definida pelas propriedades dos materiais. Desta forma, cada elemento é tratado da mesma maneira e todas as ações e reações encontradas são montadas em uma única matriz, definida como matriz global. A resolução desta matriz, através da fórmula $\mathbf{K} \cdot \mathbf{u} = \mathbf{F}$ mostrará todos os deslocamentos do domínio analisado (BASKHARONE, 2014);
- **Método dos Resíduos Ponderados (formulação fraca)** - O método envolve a aproximação de funcionais e variáveis dependentes na equação diferencial que rege o problema (LOGAN, 2007). Quando substituído nas equações diferenciais, a forma aproximada das variáveis dependentes conduz a um erro chamado de residual. O erro residual é empregado para se obter a média de erro no domínio, que deve ser minimizado. Este é um método versátil, permitindo a aplicação do MEF nos mais diversos campos. Existem dois métodos relacionados à resíduos ponderados, o método de Galerkin e o método de subdomínio ou método dos volumes finitos;
- **Método de Rayleigh-Ritz - Princípio de mínima energia potencial (variacional)** – O método de Galerkin não é apropriado para todos os problemas estruturais devido a algumas dificuldades matemáticas em relação às geometrias e condições de contorno. Uma alternativa ao método é o princípio de mínima energia potencial (KIM; SANKAR, 2009). O método da energia envolve a determinação de valores estacionários da energia global. Isto requer a aproximação de funcionais e variáveis dependentes até a energia global tornar-se estacionária. O valor estacionário pode ser máximo, mínimo ou em um ponto neutro.

2.5 Softwares Comerciais de MEF e seus módulos de otimização

Existem diversos softwares comerciais voltados ao segmento. Os softwares mais conhecidos são:

- Altair[®];
- Simulia[®];
- Nastran[®];
- Ansys[®].

Abaixo, serão demonstradas suas características e particularidades:

2.5.1 Altair[®]

A Altair Engineering foi fundada em 1985. O Altair é um pacote de programas voltado às simulações físicas de engenharia, capaz de resolver problemas lineares e não-lineares, simulação de processos de extrusão, CFD entre outros.

Dentre os vários pacotes que a empresa oferece, existe um pacote chamado OptiStruct[®]. Este é um pacote que trabalha em conjunto com o software de CAE para estudos voltados à otimização. A cada iteração realizada, é feita uma avaliação considerando as configurações do problema naquele instante. Após a obtenção dos resultados, que podem ser tensão, deslocamento, velocidade de fluxo entre outros, tais informações servirão como base de entrada para que a próxima iteração seja realizada buscando a minimização ou a maximização das variáveis de interesse.

O seu principal algoritmo voltado à otimização baseia-se no método do gradiente, ou seja, realiza o estudo de topologia baseado nos critérios desenvolvidos por Bendsøe e Sigmund (2003) apud (ALTAIR, 2014). Os documentos de ajuda do software op.cit. menciona um tipo de método

de otimização chamado de otimização por topografia. A otimização por topografia é similar ao método de otimização topológica, isto é, os elementos finitos do estudo são as variáveis de projeto, porém somente os elementos relacionados à espessura são as variáveis a serem estudadas neste tipo de otimização. A Figura 2.12 demonstra o conceito relacionado a este tipo de otimização.

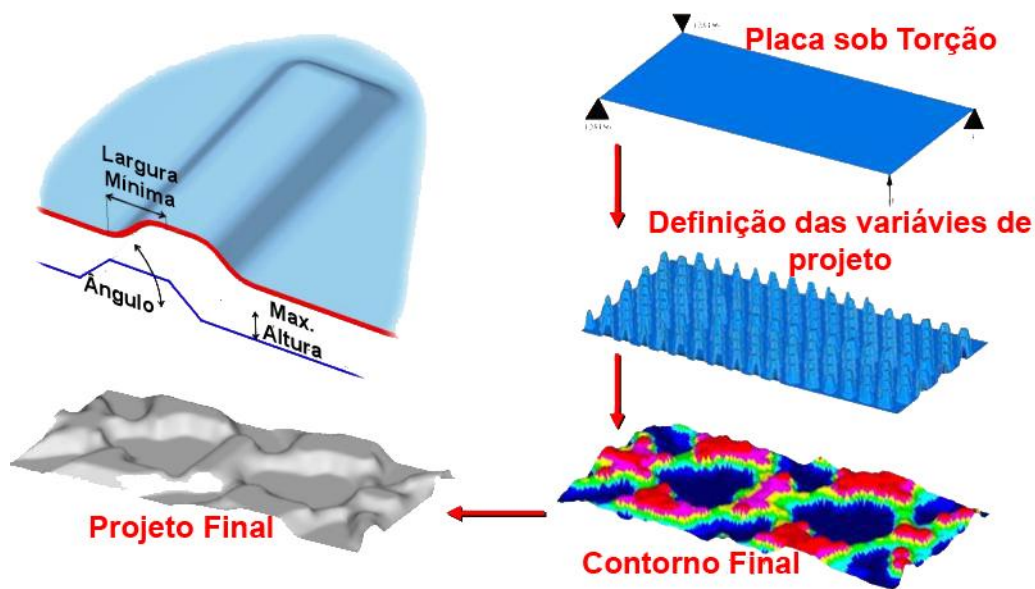


Figura 2.12 Definição do método por topografia
Fonte Altair (2014)

2.5.2 Simulia®

Popularmente conhecido como Abaqus, devido ao seu programa, a companhia fabricante foi fundada em 1978 com nome de HKS, letras que representavam seus fundadores. Em 2005, a Dassault Systèmes Simulia Corp adquiriu o software Abaqus e o aprimorou, integrando-o com o seu software de CAD, conhecido como CATIA®. Após essa adaptação, o software passou-se a ser chamado de Simulia®.

O software é dividido em módulos, claros e objetivos. Basicamente, existem o

Abaqus/Standard, cujo software é voltado para estudos de MEF, podendo realizar estudos lineares e não-lineares nas condições estáticas ou dinâmicas, o Abaqus/Explicit, voltado a estudos altamente não-lineares ou transientes e o Abaqus/CFD, voltado para o estudo de fluídos.

Existem vários pacotes adicionais que podem ser inseridos aos módulos mencionados anteriormente, entretanto, devido ao foco desta pesquisa, o interesse se resume no módulo Abaqus/CAE Topology Optimization.

No emprego das ferramentas voltadas aos estudos topológicos que o software apresenta, o algoritmo ajusta a densidade e o preenchimento do domínio. Desta forma, a algoritmo vai removendo os elementos que estão sendo menos solicitados, enquanto tenta satisfazer a função objetivo e suas restrições (BENDSØE e SIGMUND, 2003). Outro método aplicado a este tipo de problema de otimização é o algoritmo chamado Otimização de Rigidez Topológica, que se baseia na energia da deformação e nas tensões nos nós, como dados de entrada para calcular a rigidez local de suas variáveis de projeto. O algoritmo foi desenvolvido por Bakhtiary (1996) apud (SIMULIA, 2012) e segundo a empresa que o vende, tem como virtude sua eficiência, entretanto com capacidades limitadas.

Na ferramenta apontada, há a possibilidade para a realização de estudos de otimização de forma. O algoritmo, para esse fim matematicamente, é similar ao Método de Otimização de Rigidez Topológica, isto é, a energia da deformação e as tensões nos nós são dados de entrada para calcular os esforços que estão agindo na estrutura, alterando assim, as regiões em que o nível de energia é menor. Desta forma, através dos valores dos esforços a geometria irá se alterar.

2.5.3 Nastran®

É um programa de elementos finitos que foi originalmente desenvolvido pela NASA nos meados dos anos 60. Seu desenvolvimento foi patrocinado pelo governo norte-americano. O objetivo inicial do programa era tornar os veículos espaciais mais eficazes e este foi empregado, por exemplo, no programa de desenvolvimento dos ônibus espaciais que surgiram no final dos anos 60 e, início dos anos 70, segundo (NASTRAN, 2013).

No ano de 1971, o programa se tornou público com o nome de MSC.Nastran[®]. Atualmente, existem diversos módulos e o programa completo hoje supera um milhão de linhas (NASTRAN, 2013).

Em relação aos softwares estudados, Nastran[®] é o software que mais possui ferramentas voltados à otimização, para problemas relacionados à otimização de forma. Para estudos topológicos, o programa possui diversos algoritmos, entretanto, os algoritmos analisados são derivados dos conceitos desenvolvidos por Bendsøe e Sigmund em 2003.

Similar ao software da Altair[®], o Nastran[®] também possui um módulo voltado ao estudo de topografia.

2.5.4 Ansys[®]

A companhia que dá o nome ao software foi fundada em 1970 pelo Dr. John A. Swanson. O propósito inicial era o de desenvolver softwares voltados à engenharia para resolver problemas relacionados a estudos estáticos, térmicos e dinâmicos. A empresa cresceu muito, nas últimas décadas, realizando diversas aquisições.

Em se tratando de otimização, o programa apresenta um módulo voltado à otimização estocástica DOE (*Design of Experiments*), um voltado à otimização topológica, baseado nas metodologias de Bendsøe, e um módulo, baseado no gradiente, cujo domínio é desenvolvido a partir de uma interpolação realizada através do método dos mínimos quadrados.

Os métodos baseados no gradiente disponíveis no Ansys[®] são os métodos de aproximação por subproblema e o método de primeira ordem. Esses assuntos serão explorados mais adiante porque esses algoritmos serão utilizados neste trabalho de mestrado.

Existem outros softwares relacionados a elementos finitos, alguns até com o código-fonte aberto para modificações. Para este trabalho, entretanto, foram pesquisados somente os principais softwares comerciais disponíveis atualmente.

2.6 Métodos não-tradicionais de otimização

Nos últimos anos, surgiram diversos métodos de otimização conceitualmente diferente das técnicas de programação matemática até então desenvolvidas. Tais métodos estão se tornando cada vez mais populares para a resolução de problemas de engenharia. As principais características destes métodos são que estes requerem somente os valores da função objetivo e que independem no ponto de início.

2.6.1 Colônia de Formiga

O conceito do algoritmo colônia de formiga, desenvolvido por Dorigo (1992) apud (VANDERPLAATS, 2007; RAO, 2009 e SIMON, 2013), baseia-se no comportamento real das formigas, que são capazes de encontrar o caminho mais curto do seu ninho até a comida.

O processo colônia de formiga pode ser explicado, conforme o gráfico multicamadas mostrado na Figura 2.14. Nesse gráfico, o número de camadas é a quantidade de variáveis de projeto e o número de nós na camada correspondente são os números discretos admissíveis. Assim, cada nó é associado a um valor discreto de uma variável de projeto. A Figura 2.14 mostra um problema com seis variáveis de projeto e com oito valores admissíveis para cada variável.

O método colônia de formigas pode ser explicado como se segue. Uma determinada colônia é composta por N formigas.

As formigas partem de uma origem e viaja através das diversas camadas a partir da primeira até a última, conforme Figura 2.14; o ciclo termina, quando a formiga chega ao destino. Assim, o algoritmo finaliza um ciclo ou iteração.

Cada formiga pode selecionar somente um nó por camada. Os nós selecionados, ao longo do caminho visitado por uma formiga, representam uma solução. Por exemplo, o caminho percorrido pela formiga é mostrado na Figura 2.14 representa a solução (x12, x23, x31, x45, x56, x64).

Similar aos fenômenos da natureza, uma vez que o caminho é completo, as formigas depositam feromônio ao longo do caminho que percorre. Quando todas as formigas completam seus caminhos, os melhores caminhos são atualizados.

Os valores das variáveis de projeto, denotados pelos nós do caminho com a maior quantidade de feromônio, são considerados como o vetor de solução ótimo. Em geral, quando a solução ótima é encontrada, todas as formigas percorrem o mesmo caminho.

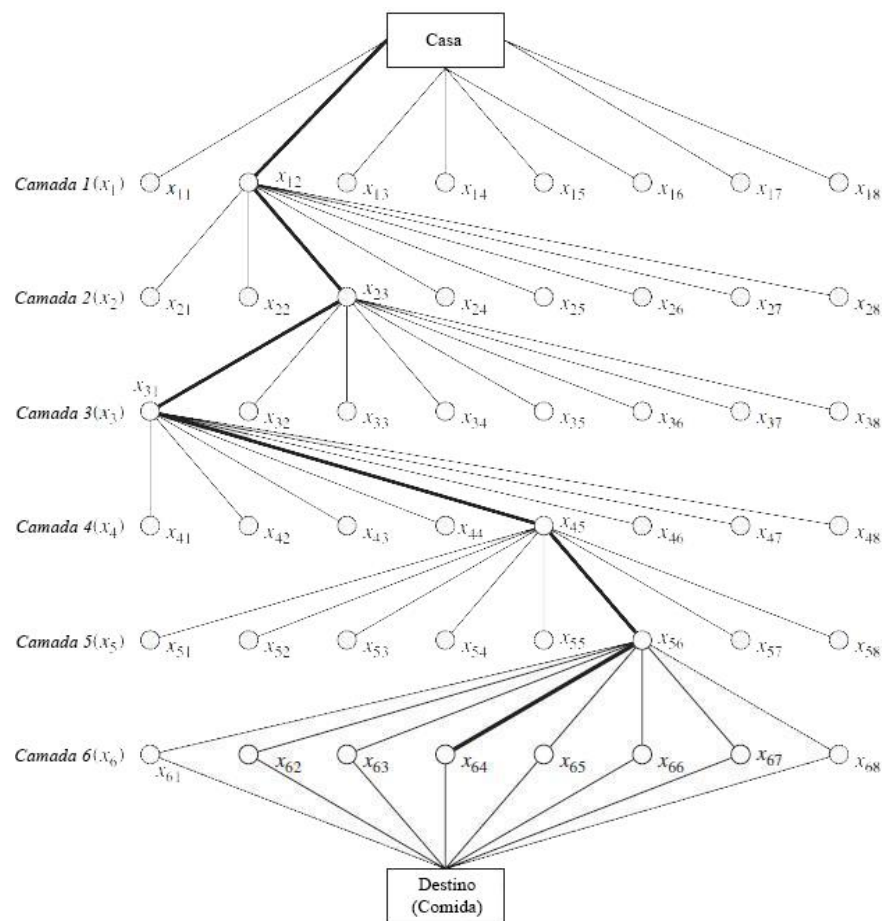


Figura 2.13 Representação gráfica do algoritmo colônia de formigas

Fonte Adaptado de Rao (2009 p. 714)

2.6.2 Enxame de Partículas

O algoritmo de otimização enxame de partícula simula o comportamento de organismos sociais como uma colônia ou enxame de insetos (por exemplo, cupins, abelhas e vespas), ou um bando de pássaros, ou mesmo um cardume de peixes. Cada indivíduo ou partícula se comporta de uma maneira distribuída, usando sua própria inteligência e a inteligência coletiva do grupo, isto é, se um indivíduo descobre um bom caminho para a comida, o restante da população também será capaz de seguir o melhor caminho imediatamente mesmo que sua localização seja longe do enxame. O algoritmo foi originalmente proposto por Kennedy et al. (1995) apud (SIMON, 2013).

O enxame é assumido com um tamanho fixo e cada indivíduo é localizado inicialmente no domínio de forma aleatória. Desta maneira, cada indivíduo é definido com duas características, posição e velocidade, e vagueia no domínio a ser estudado, memorizando a melhor posição que descobriu (em termos de fonte de alimento ou o valor da função objetivo).

A inteligência de cada pássaro resume-se em regras bem simples, que são:

- Cada indivíduo não pode chegar muito perto uns dos outros;
- Cada indivíduo é orientado para a direção média dos outros indivíduos;
- O algoritmo tenta se ajustar em relação à posição média dos outros indivíduos, sem grandes diferenças entre eles.

O comportamento do enxame baseia-se em uma combinação de três fatores:

- Coesão, ficar juntos;
- Separação, não ficar muito perto;
- Alinhamento, seguir o rumo do enxame.

Quando um indivíduo localiza um alvo (ou o ótimo da função objetivo), ele transmite instantaneamente a informação para os outros indivíduos. Ao longo do caminho, um indivíduo pode achar uma condição melhor (escape do ótimo local). Após a identificação do alimento todos os indivíduos gravitam ao redor do alvo (mas não diretamente). Os indivíduos têm um

pensamento próprio e uma memória do passado. Assim, o modelo simula uma pesquisa aleatória no espaço em busca do maior valor da função objetivo.

2.6.3 Recozimento Simulado

A técnica do recozimento simulado é uma técnica computacional que usa processos aleatórios derivados da mecânica estatística para conduzir uma busca guiada por estados de mínima energia. Basicamente, ela estabelece uma analogia entre a otimização e a maneira pela qual os metais se solidificam, cristalizam ou esfriam, como nos processos de têmpera e recozimento.

Quando um metal está no estado líquido, as moléculas estão se movendo livremente umas em relação às outras, de modo que o resfriamento ocorre lentamente. Desta forma, a estrutura vai se alinhando e ordenando até formar um cristal único e atingir o estado de mínima energia. Se resfriado rapidamente, as moléculas não se alinham e este se solidifica de forma policristalina ou amorfa (SONMEZ, 2007). O fenômeno de resfriamento de um metal fundido é simulado, introduzindo a temperatura como sendo um parâmetro sendo controlado, usando a distribuição de probabilidade de Boltzmann. A distribuição de probabilidade de Boltzmann implica que a energia (E) de um sistema de equilíbrio térmico na temperatura T é distribuída probabilisticamente de acordo com a relação:

$$P(E) = e^{-E/kT} \quad (2.2)$$

Onde $P(E)$ mostra a probabilidade de alcançar o nível de energia E , e k é chamado de constante de Boltzmann. A equação acima mostra que, em altas temperaturas, a probabilidade é uniforme em qualquer estado de energia e em baixas temperaturas a probabilidade é baixa. Isto indica que, quando o processo de busca é assumido, seguindo a distribuição de probabilidade de Boltzmann, a convergência do algoritmo pode ser controlada, manipulando a temperatura T . A

Equação 2.2 irá selecionar (ou não) uma determinada configuração (KIRKPATRICK et al., 1983 apud HO e WU, 2012).

Características do Método:

1. A qualidade da solução final não é afetada pelas estimativas iniciais, com exceção ao fato de que o esforço computacional pode aumentar com o pior ponto de partida.
2. Por causa da natureza discreta de avaliações, as características de convergência ou de transição não são afetadas pela continuidade ou não-continuidade das funções.
3. A convergência não é influenciada pela convexidade do problema.
4. As variáveis de projeto não precisam ser positivas.
5. O método pode ser usado para resolver problemas contínuos, discretos ou mistos.

A Figura 2.15 apresenta o fluxograma do algoritmo RS, abaixo segue uma descrição das etapas do método.

Descrição do Fluxograma

1. Escolher os parâmetros iniciais onde T = temperatura (tal temperatura é definida inicialmente em relação a uma média de algumas configurações realizadas de modo aleatório), n é o número de iteração e c é um termo relacionado ao fator de redução de temperatura.
2. Calcular a função objetivo.
3. Elaborar uma nova configuração de pontos.
4. Gerar um número aleatório de 0 a 1.
5. Utilizar a **Equação 2.2**, caso o número encontrado, utilizando a equação, for maior que o número aleatório obtido no passo anterior, os novos valores serão aceitos.

6. Atualizar a função objetivo.
7. Reduzir a temperatura.
8. Repetir o processo.

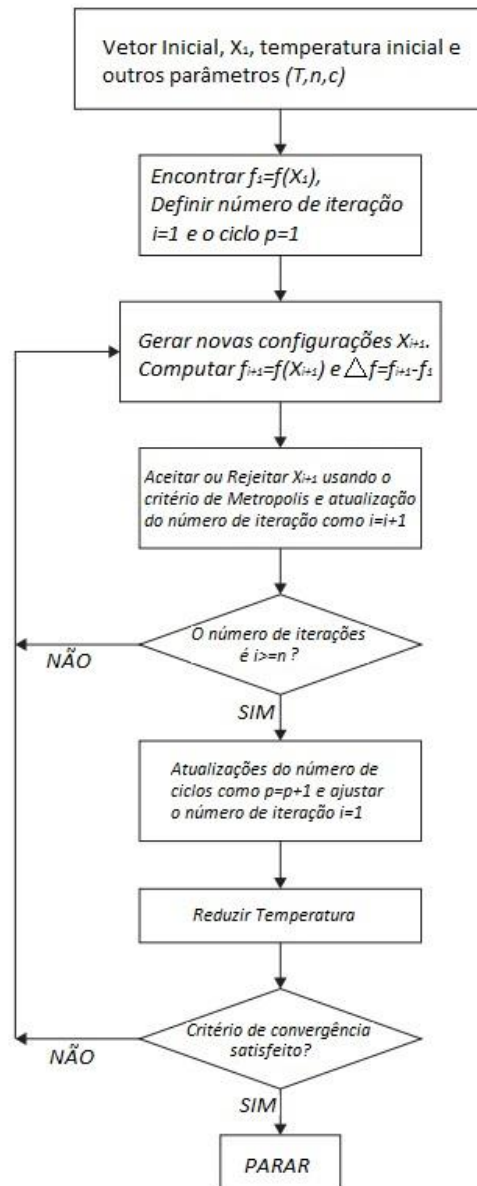


Figura 2.14 Fluxograma do algoritmo RS
Fonte Desenvolvido por Rao (2009 p. 706)

2.6.4 Busca Tabu

A ferramenta Busca Tabu (BT) é um algoritmo com característica metaheurística. Há no algoritmo um procedimento adaptativo que o guia na exploração contínua, dentro de um espaço de busca. A partir de uma solução inicial, tenta avançar para outra solução (melhor que a anterior) na sua vizinhança até que se satisfaça um determinado critério de parada Glover (1978) apud (GLOVER, 1986).

A ideia por trás do algoritmo BT é a utilização de uma memória para impedir que, durante o processo de busca, o algoritmo encontre e pare em um mínimo local. No método BT, a utilização de uma memória de curto prazo permite que o algoritmo possa escapar dos mínimos locais. Esta memória é utilizada para evitar que, durante o processo de busca, o algoritmo retorne a posições já visitadas. O algoritmo BT pode ser implementado por meio da memorização das soluções candidatas, visitadas anteriormente, afastando qualquer passo que o levaria de volta a eles (LIN et al., 2003).

2.6.5 Algoritmos Genéticos

AG é um algoritmo de busca computadorizada no qual a otimização é baseada na natureza, na mecânica da genética e na seleção natural. O algoritmo genético foi originalmente proposto por Holland (1975) apud (BUREERAT, 2008) e é base deste trabalho de mestrado.

AG é uma técnica de computação evolutiva, que é um ramo da inteligência artificial. As características dos algoritmos evolucionários são sua dependência à aptidão, informação aleatória e regras probabilísticas ou determinísticas de reprodução, para ajudar na busca a um ponto ótimo no espaço de projeto (WIDMAIER, 2005).

Os elementos básicos do algoritmo genético são reprodução, cruzamento e mutação e estes são utilizados no procedimento da pesquisa genética. AGs diferem dos métodos tradicionais de otimização com relação aos seguintes aspectos segundo (RAO, 2009):

1. A população de pontos (vetores de projeto em estudo) é usada para iniciar o procedimento em vez de um ponto único. Se o número de variáveis de projeto é n , normalmente o tamanho da população é especificado como de $2n$ a $4n$, uma vez que vários pontos são usados como soluções candidatas. AGs são menos propensos a acharem um ótimo local.
2. AGs usam apenas os valores da função objetivo. As derivadas não são usadas no procedimento de busca, característica dos métodos tradicionais.
3. Em AGs as variáveis de projeto são representadas como *strings* de variáveis binárias que correspondem aos cromossomos na genética natural. Assim, o método de pesquisa é naturalmente aplicável para resolver problemas de programação discretos e inteiros. Para as variáveis de em projeto contínuo, o comprimento da *string* pode ser variado para atingir qualquer resolução desejada.
4. O valor da função objetivo corresponde a um conjunto de valores adotados para cada variável em uma determinada iteração, o tamanho da população irá definir a quantidade de estudos realizados por iteração. Desta forma, a função objetivo em cada iteração irá corresponder a média dos valores obtidos. Tais valores desempenham o papel de aptidão no processo evolutivo do algoritmo.
5. A cada nova geração, uma nova *string* é produzida, usando os pais da antiga população, selecionados aleatoriamente para seleção e cruzamento (antigo conjunto de *strings*). O importante a ser notado aqui é que a busca aleatória não implica necessariamente em uma busca cega e sem direção. Os algoritmos genéticos utilizam o conhecimento adquirido durante as várias gerações, para se aproximar da solução ótima.

A solução de um problema de otimização utilizando AG inicia-se com uma string de gerada à partir de uma população aleatória para cada vetor de projeto (população). O tamanho da população no AG (n) é usualmente fixo. Cada vetor de projeto é calculado para encontrar o valor de função objetivo conhecida no AG como *fitness*. A população é operada por três operadores que são seleção, cruzamento e mutação para produzir uma nova combinação de pontos. A nova população é então calculada com o objetivo de encontrar um novo valor para a *fitness*. Tal valor

é testado com o intuito de verificar a convergência do processo. Um ciclo de reprodução, cruzamento e mutação é conhecido como geração do AG. Se o critério de convergência não for satisfeito, a população é novamente criada, utilizando os três operadores mencionados, resultando, assim, em uma nova população. Desta forma, o procedimento é continuado através de várias gerações até que o critério de convergência seja satisfeito e o processo é encerrado.

Existem diversos outros algoritmos, além dos apresentados como, por exemplo, o método de otimização fuzzy, que foi desenvolvido para resolver problemas de otimização envolvendo dados de projeto e foi apresentado pela primeira vez por Rao em 1986 (VANDERPLAATS, 2007). Também há o método chamado rede neural, baseado no imenso poder computacional do sistema nervoso animal para resolver problemas de percepção, na presença de grande quantidade de dados sensoriais, através de sua capacidade de processamento paralelo. Seu algoritmo foi desenvolvido por Hopfield et al. em 1985 (VANDERPLAATS, 2007). Nesse trabalho, buscou-se obter informações somente sobre os principais algoritmos aplicados em problemas de otimização de forma de componentes mecânicos, o que não é o caso dos dois métodos citados acima.

2.7 Estudos relacionados à otimização

Neste tópico, serão apresentados estudos que abordam a otimização como ferramenta para obtenção de eficácia de processos ou produtos.

Khoo e Ong (1998), com o objetivo de melhorar a eficiência no planejamento do processo de furação em placas de circuito impresso, foi utilizado o AG para a otimização no tempo de processo. Assim, o tempo de processo na furação foi otimizado, considerando os furos como um problema do caixeiro-viajante (PCV). Os autores afirmaram que a distância percorrida pela ferramenta para a realização do processo de furação na placa estudada foi de 19,80% menor, utilizando o algoritmo.

Parsons e Canfield (2001) demonstraram a utilização do AG para projetar um mecanismo flexível para resolução de problemas estruturais. O estudo consistia em criar um dispositivo cuja

finalidade era inserir uma força oposta à força externa aplicada à estrutura. O intuito era manter a estrutura na posição inicial. No estudo apresentado, os autores definiram algumas estruturas treliçadas e a adição ou remoções de barras foram definidas com o auxílio do AG. Os autores afirmam que tais metodologias tiveram como principais vantagens a sua robustez, a sua capacidade de busca e a não dependência do gradiente, além de conduzir a um produto ótimo.

Bishop e Striz (2004) utilizaram o AG para um estudo de posicionamento de amortecedores de vibração em estruturas flexíveis. Diversos carregamentos dinâmicos foram considerados em vários estudos de caso. Os autores concluem que a aplicação do AG, em problemas de posicionamento de amortecedores, em estruturas devido às suas características exaustivas de busca, torna-o extremamente útil.

O método de colônia de formiga foi utilizado por Yagmahan e Yenisey (2007) para realizar um estudo de programação e agendamento de tarefas. Os autores mencionam que a utilização de ferramentas metaheurísticas em situações similares é muito eficaz. Anos mais tarde, Yagmahan e Yenisey (2009) aprimoraram o algoritmo para as mesmas aplicações.

Degertekin et al. (2007) compararam o método BT com o AG para estudos de otimização de estruturas. Segundo os autores, o método de BT foi entre 7% e 9% melhor do que o método do AG. Os autores mencionam a arquitetura do algoritmo para memorizar os melhores resultados do passado foi sua principal virtude. Entretanto, apesar de ter um melhor desempenho, o tempo computacional foi superior.

Sonmez (2007) utilizou o método de RS para obter formas bidimensionais otimizadas de componentes sujeitos a cargas quasi-estáticas. O objetivo era a redução de peso (volume) da estrutura, tendo o limite de escoamento do material como restrição para a solução do problema topológico. O autor desenvolveu um programa para trabalhar em conjunto com o Ansys[®]. A posição dos pontos a cada iteração era definida de forma aleatória, baseada no critério de busca do RS. Tais pontos foram ligados através de uma curva do tipo *spline* e assim a forma era obtida. A técnica segundo o autor pode ser aplicada em problemas bidimensionais com uma alta precisão.

Lamberti (2008) demonstra a utilização de um algoritmo, modificado baseado no RS, cujo nome é *múltiplos níveis corrigidos e múltiplos pontos de recozimento simulado*, e a sigla em inglês é CMLPSA. Segundo o autor, a flexibilidade e a simplicidade do método do RS faz com

que o algoritmo seja ideal para alterações que melhorem o seu custo computacional. Os estudos de Lamberti foram voltados à otimização de estruturas treliçadas. Foram encontrados valores muito superiores aos métodos tradicionais e foi necessária uma quantidade bem menor de iterações para a resolução de um problema proposto pelo autor.

Gen et al. (2008) propuseram uma rotina de otimização, baseada em técnicas evolucionárias, de modo que todo o processo de desenvolvimento fabril fosse modelado como um grande problema de otimização. Isto incluiu o leiaute da fábrica, o planejamento na sequência de produção, a sequência das operações, durante os processos de usinagem, que serão necessários para a obtenção do produto final e a logística da distribuição dos produtos.

Bureerat e Limtragool (2008) utilizaram o método do RS para desenvolver estudos voltados à otimização topológica. A principal desvantagem da utilização destas ferramentas em problemas topológicos é o alto tempo computacional. Para resolver este problema, os autores utilizaram uma ferramenta matemática chamada distribuição de densidade aproximada. Tal ferramenta tem como finalidade remover elementos que estão sendo menos solicitados do domínio.

Vasan e Raju (2008) utilizaram os métodos RS e AG para estudos de problemas relacionados ao planejamento eficiente de recursos hídricos. Segundo os autores, os resultados apresentados foram muito próximos e ambos resolveram o problema sem muitas dificuldades. Entretanto, o tempo computacional do método do RS foi superior.

Um software desenvolvido por Li et al. (2009) foi proposto com a finalidade de criar modelos geométricos e integrar tais modelos com um algoritmo de otimização não-tradicional. Os autores utilizaram o método do RS. Entretanto, os autores efetuaram alterações na sua linguagem inicial a fim de ganhar desempenho computacional, isto é, implementaram o algoritmo para realizar as tarefas em modo paralelo. Desta forma, cada processador realizaria uma etapa do cálculo. O intuito foi estudar um leiaute tridimensional. Utilizando os conceitos mencionados, os autores tiveram um resultado em média 70% superior, comparado ao algoritmo RS desenvolvido na sua forma tradicional.

Gologlu e Zeyveli (2009) utilizaram os AGs para realizar estudos voltados à otimização de caixas de câmbio, visando minimizar seu volume sem perda de eficiência. Tais estudos auxiliariam no desenvolvimento inicial do projeto de uma nova caixa de transmissão. Com o

intuito de verificar o impacto, os autores alteraram os parâmetros (P_m =*Probabilidade de Mutação* e P_c =*Probabilidade de Cruzamento*) do AG. Os autores concluíram que na maioria dos estudos quando a probabilidade de P_m foi ajustado em 0,001 e a P_c foi de 0,9 obteve um desempenho superior a valores diferentes deste.

Foi também estudada a aplicação de AG em problemas relacionados à otimização da área de construção, levando em consideração o custo e o número de colunas de sustentação ideal, de modo que a área dos cômodos possa ser maximizada. Nimtawat e Nanakorn (2009) tiveram resultados muito promissores com a utilização do método.

Lin e Sheu (2009) desenvolveram uma alternativa para solução de problemas topológicos. O método, denominado restrição de volume adaptativa, tem como característica o ajuste do seu volume durante o processo de otimização topológica. Nos métodos anteriormente desenvolvidos, em sua grande maioria, os volumes não são tratados de forma adaptativa.

Lv et al. (2009), baseado no método do RS, propuseram uma nova ferramenta. A razão é que, apesar de o AG levar a resultados bastante satisfatórios, o custo computacional normalmente é alto. Desta forma, o algoritmo chamado teoria nas nuvens, baseado no recozimento simulado ou em inglês *Cloud Simulated Annealing (CSA)*, é proposto. Em diversos problemas realizados, a ferramenta apresentada obteve resultado superior em relação ao método RS de modo tradicional.

Leon (2009) propõe um novo conceito para o desenvolvimento de novos produtos cujo termo é chamado CAI (Computer Aided Innovation). O objetivo, segundo o autor, é desenvolver um produto ótimo, isto é, amarrar o modelo CAD com o CAE e softwares de otimização, utilizando os conceitos de inovação como TRIZ, Brainstorming, QFD, entre outros.

Wei et al. (2010) aborda problemas relacionados à otimização de estruturas. Foram utilizados diversos métodos em um único algoritmo, isto é, combina-se a robustez e a facilidade de encontrar o mínimo global, característica do AG, a habilidade de exploração do domínio da ferramenta simplex e o ganho de velocidade obtido, quando se trabalha em paralelo, formam o algoritmo denominado NHPGA (*Niche Hybrid Parallel Genetic Algorithm*). Os autores mencionam que embora o AG tenha explorado de modo eficiente o domínio estudado, este por sua vez, teve um custo computacional elevado. Esse método proposto pelos autores, NHPGA, é uma ferramenta que superou o AG quando avaliado considerando o tempo computacional.

Park e Dang (2010) demonstraram a interação entre as ferramentas CAD-CAE, aplicando-as em um estudo de otimização. Usando o método DOE e através de um arquivo no formato batch, as plataformas CAD e CAE foram integradas. A cada iteração o produto foi sofrendo alterações a fim de levar a um resultado ótimo com um custo mínimo. A integração entre as ferramentas, segundo os autores, reduz o custo da engenharia no desenvolvimento de novos produtos e estes, por sua vez, são otimizados.

Bhuvaraghan et al. (2010) utilizaram o DOE para prever e avaliar um estudo de propagação de trinca por fadiga gerada através do processo de *Shot Peening*, que é o bombardeio de esferas em uma superfície a fim de se obter uma característica superficial diferente em termos de tensão. Os autores mencionam que a utilização de um software de elementos finitos, trabalhando simultaneamente com alguma ferramenta voltada a otimização, auxilia no diagnóstico de falhas.

Cheng (2010) utilizou o método do AG para estudos de otimização de pontes, visando a minimização do peso. Tais resultados, foram comparados com as ferramentas tradicionais voltadas à otimização implementadas em softwares comerciais. Para o estudo em questão, foi utilizada a ferramenta chamada *First-Order*. O método dos AG foi muito promissor segundo o autor, sendo superior aos métodos tradicionais. Comparando os resultados obtidos, à medida que a quantidade de variáveis foi aumentando, a diferença entre os resultados foi se tornando cada vez maior. Com 4 variáveis a diferença entre os resultados foi de 5%, com 10 variáveis 12% e com 41 variáveis, foi de 45%.

Seo et al. (2010) desenvolveram uma técnica voltada para estudos topológicos na qual as superfícies estudadas são aparadas e é criada uma curva do tipo *spline* no vazio gerado no domínio sob estudo. Normalmente, em estudos topológicos, os elementos são removidos integralmente, gerando um vazio e criando *degraus* no seu domínio. Uma forma para resolver este tipo de problema é diminuindo o tamanho do elemento. Segundo os autores, o trabalho apresentado mostra uma alternativa a esse tipo de problema.

Vasant e Barsoum (2010) desenvolveram uma técnica chamada de técnica híbrida de busca entre Busca Padrão (BP) e o RS para estudos de planejamento dos processos de produção. Segundo os autores, a versatilidade do algoritmo, permite uma fácil adaptação dos critérios de deduções iniciais. Desta forma, os autores alteraram a forma das deduções, utilizando uma

ferramenta já conhecida como BP, substituindo no método tradicional. Após as deduções iniciais, o algoritmo evolui da forma tradicional. Os resultados apontados, utilizando tais técnicas, foram superiores aos obtidos com métodos tradicionais.

Hsiao et al. (2010) propuseram a elaboração de um banco de dados com diversos conceitos já modelados em CAD. Cada um destes já haviam sido usados na criação de um produto final. Utilizando o AG, a seleção dos componentes a partir desse banco de dados pode ser realizada, de modo a obter um produto ótimo.

O trabalho desenvolvido por Bénabès et al. (2011) demonstra a aplicação dos AGs para um estudo de leiaute. O estudo busca o melhor arranjo físico (localização e orientação) de mobílias em um determinado espaço, levando em consideração suas restrições. Assim, o estudo consistiu em otimizar uma sala aonde iriam diversos computadores, mesas e armários. Os autores ressaltam a abrangência deste método, e mencionado destas ferramentas pode ser aplicado em diversos segmentos.

Leung et al. (2010) realizaram um estudo empregando o método do RS híbrida para resolução de problemas voltados a otimização de leiaute, com foco na logística. Um exemplo estudado pelos autores foi a otimização no preenchimento de um container. Desta forma, os pacotes foram as variáveis e o volume do container o domínio a ser preenchido.

Kamran e Guozhu (2011), a partir de um modelo CAD, utilizaram o método de otimização AG para realizar um estudo de otimização de foguetes de modo que cada parte do foguete fosse estudada de forma independente. Para tal estudo o algoritmo utilizado foi o AG, a cada iteração o algoritmo gerava novos dados de entrada para o arquivo CAD. Assim havendo uma comunicação simultânea entre as duas ferramentas.

Shi (2011) realizou um estudo voltado à arquitetura utilizando ferramentas evolucionárias de otimização. O problema em questão consistia em obter a melhor estratégia para o aproveitamento de isolantes térmicos em uma determinada construção ao longo do dia. Para o estudo foi utilizado o AG. Entretanto, o software modeFrontier® foi também empregado, em conjunto, permitindo o estudo fosse realizado.

Chiandussi et al. (2011) realizaram um estudo de otimização, empregando o MEF com o intuito de verificar os esforços mecânicos, com o AG para realizar o estudo de otimização. Os autores realizaram diversos estudos, utilizando um pacote de métodos de otimização chamada

Dakota[®] e uma ferramenta voltada a elementos finitos, Abaqus[®]. Os valores dos esforços mecânicos eram dados de entrada para o software Dakota[®], software desenvolvido pela Sandia National Laboratories, cuja finalidade é o desenvolvimento de algoritmos relacionados à otimização. O software Abaqus[®] interpretava as posições das variáveis e realizava o estudo.

Ahari et al. (2011) desenvolveram um estudo adotando o AG para resolução de problemas relacionados a otimização de processos de laminação. Uma das variáveis do AG é o tamanho da população. Os autores realizaram estudos e demonstraram que o tamanho da população influencia nos resultados, como esperado.

Um estudo de caso com o intuito de melhorar o comportamento de uma estrutura foi realizado por Turrin et al. (2011). Os autores utilizam um termo recente chamado de CACD (*Computer Aided Conceptual Design*) proposto por Chong et al. (2008). O conceito consiste em parametrizar uma ferramenta CAD com uma voltada a algum mecanismo de otimização. Devido ao fato do estudo estar voltado a problemas de arquitetura, os autores utilizaram o software de elementos finitos Staad.Pro[®] e desenvolveram o AG para trabalhar em conjunto. Com o objetivo de o cliente final acompanhar todo o processo evolutivo do estudo, os autores inseriram os resultados em um banco de dados em SQL. Assim, via web o cliente consegue visualizar tais informações.

Wang et al. (2012) realizaram um estudo de otimização de forma utilizando o AG. Segundo os autores, nenhum software comercial é capaz de realizar um estudo de otimização da forma como o que foi proposto. Basicamente, através de uma subtração booleana o volume é gerado. As variáveis são inseridas somente em um volume e após uma subtração booleana a malha é gerada. Por exemplo, se o estudo fosse realizado em uma placa com um furo, existiriam dois volumes, o da placa a ser estudada e o do furo, que neste caso seria uma barra cilíndrica. Após uma subtração booleana esta barra irá se tornar um furo na placa. Porém, as variáveis de projeto somente estarão nesta barra cilíndrica. Tal método foi chamado pelos autores de método de perfuração virtual. Simplicidade, facilidade de implementação e forte conexão com os softwares CAD foram apontados pelos autores como suas principais virtudes.

An e Zhang (2012) realizaram um estudo de elementos finitos em problemas relacionados ao processo de estampagem. A estratégia dos autores foi utilizar o método DOE para a definição

da amostra inicial a ser estudada. Após definição inicial, o método do AG foi utilizado com o intuito de obter o produto ótimo.

Baseado em um trabalho desenvolvido pelos autores em 2009, Hamade et al. (2012) desenvolveram uma ferramenta para avaliar o desempenho de novos usuários CAD. O software, utilizado para estudo, foi o Pro-Engineer® (CREO®). Foram empregados dados experimentais obtidos pelos autores com o intuito de avaliar o desempenho. A partir destas informações, foi desenvolvida uma ferramenta, baseada no RS e em Redes Neurais (RN), para prever o desempenho de novos usuários com a qual os autores tiveram resultados 20% superior comparado a situações em que não havia tais mecanismos.

Empregar modelos de CAD flexíveis, isto é, criar variáveis em cada dimensão e estas, por sua vez, serem as variáveis para a otimização de forma é o que propõe o trabalho de Amadori et al. (2012). O método propõe um conceito diferente para o desenvolvimento de produtos, pois todas as *features* serão definidas como variáveis e estas trabalharão em comunicação com um banco de dados a fim de se obter um produto ótimo baseado em um histórico já desenvolvido.

Percebe-se que a aplicação das ferramentas de otimização em processos, produtos, leiautes e estruturas faz com que o objeto se torne mais eficiente em relação aos parâmetros que foram definidos inicialmente. Devido à grande evolução computacional, a utilização de tais conceitos torna os métodos metaheurísticos de otimização uma ferramenta extremamente promissora para o desenvolvimento de novos conceitos.

3. MODELAGEM TEÓRICA E MÉTODOS

Este capítulo descreve em detalhes os métodos tradicionais de otimização, empregados por um programa comercial, o Ansys[®], e os diversos métodos não-tradicionais, dentre os quais o AG, aplicados à otimização de componentes mecânicos. Tais conceitos são necessários para o entendimento da proposta do trabalho, uma vez que esta inclui a comparação entre os resultados obtidos através do programa comercial citado e a da proposta feita nesta dissertação. Em seguida, serão descritos as particularidades e recursos do software para estudos relacionados à otimização. A estrutura do AG, comparativos entre o AG e o RS, e o desenvolvimento de um software, a partir da linguagem Fortran[®], para estudos relacionados à otimização de forma.

3.1. Descrição do Programa de Elementos Finitos:

O Ansys[®] é um software de análise de elementos finitos, voltado a estudos de engenharia, que possui as seguintes características:

- Capacidade de construir os modelos a serem estudados ou mesmo importar a geometria CAD de outros softwares;
- Possibilidade de aplicar cargas ou condições encontradas em campo;
- Ferramentas de otimização que permitem obter um produto ótimo a um custo mínimo;
- Prototipagem virtual, para componentes que não podem ser testados, pois trabalham em ambientes onde não se deseja realizar experimentos ou mesmo não permitem tal experimentação.

Segundo Ansys (2009), a interface do software permite que um estudo em elementos finitos possa ser desenvolvido, utilizando os seguintes recursos:

- Modo Interativo: Este é o modo mais comum de interação entre o usuário e o software. O

modo é caracterizado pela ativação de uma plataforma chamada de Interface Gráfica com o Usuário (GUI em inglês), no qual a interface é composta de menus, caixa de diálogo, botões e janelas. O modo interativo é recomendado para os usuários iniciantes.

- **Modo em Batch:** Este método utiliza o software sem a ativação da GUI. O modo tem como característica a criação de um arquivo chamado de *Input File* escrito em APDL (Ansys Parametric Design Language), no qual permite a execução de comandos e parâmetros a partir de uma linguagem de programação própria baseada na linguagem Fortran[®]. Estas capacidades tornam o modo em Batch uma poderosa ferramenta de análise.
- **Modo Combinado:** Este método é a combinação do modo interativo com o modo em batch. Tal modo ativa uma GUI e a partir dela insere um arquivo via Input File.

3.1.1 APDL

É uma linguagem de programação baseada em *script* que permite automatizar tarefas comuns, ou mesmo construir um modelo, em função de parâmetros do problema. Estão disponíveis nessa linguagem todos os comandos do software para o desenvolvimento de rotinas, além de inúmeras outras características de um software de programação, tal como repetir um comando, macros, *if-then-else*, *do-loops* e operações com vetores, escalares e matrizes.

Para o trabalho apresentado, foi desenvolvido um programa em APDL. Nesse, os vértices que caracterizam a forma da peça a ser otimizada podem ser definidos como variáveis de projeto, de modo que, a cada iteração, estes poderão assumir uma nova posição. O programa desenvolvido foi baseado em AG, como já dito. A cada etapa, o programa definirá as posições dos vértices para a próxima iteração.

O programa comercial mencionado permite o modelamento de materiais estruturais, utilizando uma ampla biblioteca de elementos especializados, mais adequados para simular um fenômeno físico específico. O trabalho proposto adotará um elemento de alta ordem, o

PLANE183. Apesar de a quantidade de incógnitas ser maior do que em outros elementos mais simples, a vantagem de trabalhar com elementos de alta ordem é a facilidade de lidar com geometrias complexas, permitindo um resultado mais adequado para a maioria dos casos.

O elemento adotado possui características triangulares ou quadráticas, com 8 nós, conforme Figura 3.1. Este tipo de elemento é indicado para simular comportamentos estruturais para a análise de diversos fenômenos como, por exemplo, hiperelasticidade, propagação de trinca, viscoelasticidade entre outros fenômenos em problemas no campo bidimensional.

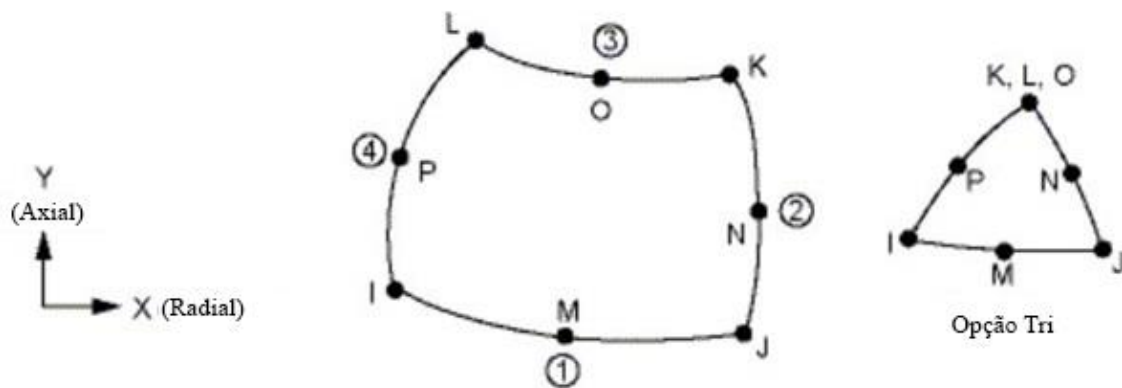


Figura 3.1 Elemento Plane183

Fonte Ansys (2009)

Para os estudos tridimensionais, será utilizado o elemento SOLID186, conforme a Figura 3.2 que possui as mesmas características que o elemento PLANE183. Dependendo da geometria a ser discretizado, o software preenche o domínio, utilizando elementos hexaédricos ou tetraédricos, as Figuras 3.1 e 3.2 mostram as variações permitidas para os elementos utilizados.

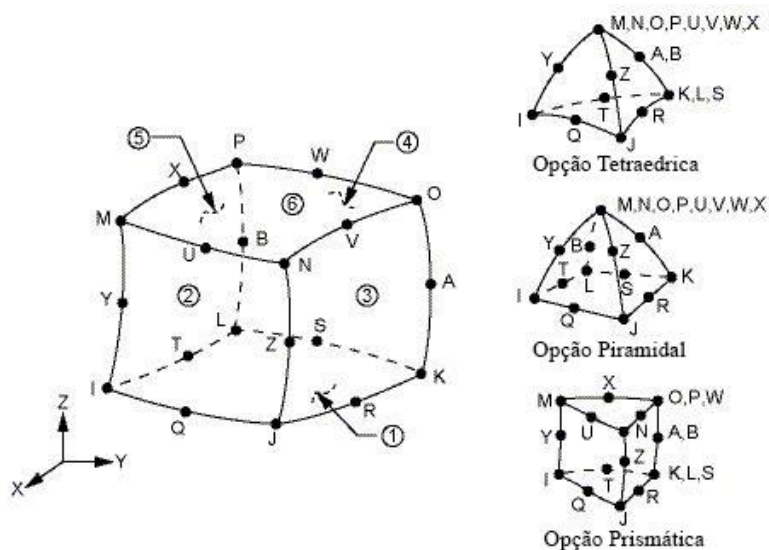


Figura 3.2 Elemento Solid186

Fonte Ansys (2009)

3.1.2 Algoritmos de Otimização disponíveis no Ansys®

O software comercial, que está sendo utilizado como base de estudo, possui diversas ferramentas para o auxílio no desenvolvimento de estudos direcionados à otimização. Algumas delas são voltadas a estudos, empregando otimização de forma e outras com otimização topológica.

Os algoritmos, que poderão ser empregados para estudos de otimização de forma no programa, são apresentados em seguida. Segundo Ansys (2009), o programa utiliza dois métodos de otimização, o *método de aproximação por subproblema* e o *método de primeira ordem*.

O método de *aproximação por subproblema* é um método que, devido à sua versatilidade, comporta-se melhor na maior parte dos problemas voltados a engenharia.

O método de *primeira ordem* baseia-se nas sensibilidades das variáveis e é mais adequado para os problemas onde requer uma precisão mais elevada.

Para ambos os métodos de aproximação, o software executa uma série de ciclos de análise e avaliação. Inicialmente, o componente estudado é analisado e os resultados obtidos são avaliados,

baseando-se nos critérios definidos. Quando necessário, o componente é modificado, respeitando os limites de cada variável. O processo é repetido até que se cumpram todos os critérios especificados.

O método de aproximação por subproblema é um método que requer somente os valores das variáveis, e não suas derivadas. Há duas particularidades que desempenham um papel fundamental neste método: o uso de curvas de aproximação para representar o comportamento da função objetivo e das variáveis de estado, além da conversão de um problema de otimização com restrição em um problema sem restrição.

Nessa abordagem, o programa determina a relação entre a função objetivo e as variáveis de estado por uma curva de ajuste. As variáveis dependentes são substituídas por valores aproximados através de uma curva de ajuste desenvolvida pelo método dos mínimos quadrados. Assim, cada ciclo de otimização gera um novo conjunto de dados para cada variável, e a função objetivo aproximada é atualizada. Desta forma, a curva a ser minimizada é a aproximada e não a curva real.

As variáveis de estado são tratadas da mesma maneira. Uma curva de aproximação é gerada para cada variável de estado e atualizada no final de cada iteração. A aproximação por ajuste de curva pode por ajustes lineares, quadráticos, ou de ordem superior.

As variáveis de estado e seus limites, utilizados para a realização de um determinado estudo, restringem o problema. A técnica tem como finalidade converter o estudo em um problema de otimização sem restrições, com o qual as técnicas de minimização são mais eficientes (ANSYS, 2009). A conversão é realizada pela adição de penalidades na função objetivo nos pontos em que as restrições foram impostas. Esta técnica é conhecida como técnica sequencial irrestrita de minimização ou Sequential Unconstrained Minimization Technique (SUMT).

Diferente do método de aproximação por subproblema, no método de primeira ordem a minimização é realizada nas suas condições reais e não de modo aproximado. Desta forma, o método é mais preciso (ANSYS, 2009). Entretanto, a técnica SUMT também é empregada neste algoritmo, isto é, o problema com restrição acaba se tornando sem restrição devido à adição de penalidades nas restrições (ANSYS, 2009).

O método de primeira ordem utiliza o gradiente das variáveis dependentes, em relação às variáveis de projeto. Para cada iteração, o cálculo dos gradientes é feito, a fim de determinar uma direção de busca. Para isso, podem ser utilizados o método da Descida Íngreme (*Steepest Descent*) ou o método de direção conjugada.

Cada iteração é composta de uma determinada quantidade de subiterações, que estão relacionadas ao sentido de busca e ao cálculo do gradiente. Desta forma, o método é computacionalmente mais caro.

3.2 Ferramentas de otimização não-tradicionais

As ferramentas, ditas como não-tradicionais ou métodos modernos de otimização e que serão empregadas no trabalho proposto, têm como principais características:

- Necessidade de um gerador de números aleatórios;
- Não haver necessidade de derivação;
- Facilidade de resolver problemas com grandes descontinuidades.

A escolha do AG para o estudo em questão se deve ao fato de que, segundo Andresen et al. (2008), o AG tende a ser mais rápido do que as outras ferramentas que possuem as mesmas características. Um algoritmo, muito utilizado para esse tipo de estudo é o RS, entretanto, ele é um pouco mais lento, se compararmos com o AG (HASAN et al., 2000).

3.2.1 Comparação entre o AG e RS

A flexibilidade na formação do código-fonte do método RS permite adaptações com o intuito de melhorar seu desempenho. Tais aprimoramentos o tornam uma ferramenta muito

utilizada (VASANT e BARSOUM, 2010). A tese de doutorado de Vasant impulsionou no desenvolvimento de um novo método chamado HSAPS (*Hybrid Simulated Annealing and Pattern Search*) cujo objetivo foi melhorar o desempenho do RS, utilizando as qualidades de outra ferramenta de otimização chamada BT.

A Tabela 3.1 apresenta a comparação entre o AG e o RS.

Tabela 3.1 Tabela comparativa entre o RS e o AG

Fonte Adaptado de Vasant e Barsoum (2010)

	Recozimento Simulado	Algoritmo Genético
Padrão de Busca	Ponto-a-Ponto	Baseado em População
Parâmetros Importantes	Número de iterações, Temperatura inicial, temperatura final e taxa de resfriamento	Número de gerações, tamanho da população, cruzamento e probabilidade de mutação
Analogia	Processo de resfriamento dos metais	Critério de sobrevivência de Darwin
Status de convergência ao ponto ótimo	Próximo ao ótimo	Próximo ao ótimo
Possibilidade do algoritmo permanecer preso no ótimo local	Quase nula	Mínimo
Tempo Computacional	Muito Alto	Baixo

Estudos preliminares realizados neste trabalho de mestrado mostraram que o desempenho do método AG foi aproximadamente 7% superior ao RS em problemas relacionados à otimização de forma. A escolha do AG como ferramenta para estudo deve-se a esse fato, dentre outros.

3.2.2 Algoritmo Genético

Ao invés de partir de um único ponto, o padrão de busca do AG é inicializado por uma população distribuída aleatoriamente no domínio. A evolução do algoritmo depende de três operadores: seleção, cruzamento e mutação. Tal evolução direciona a população em busca do ótimo global.

No centro da Figura 3.3 são mostradas as principais características do AG, como Estatística, Encadeamento, Observação e Números Aleatórios. A estrutura do AG é baseada no processo de evolução. Os termos derivados radialmente estão relacionados ao método e entre si e serão discutidos em detalhes ao longo dos tópicos que se seguem.

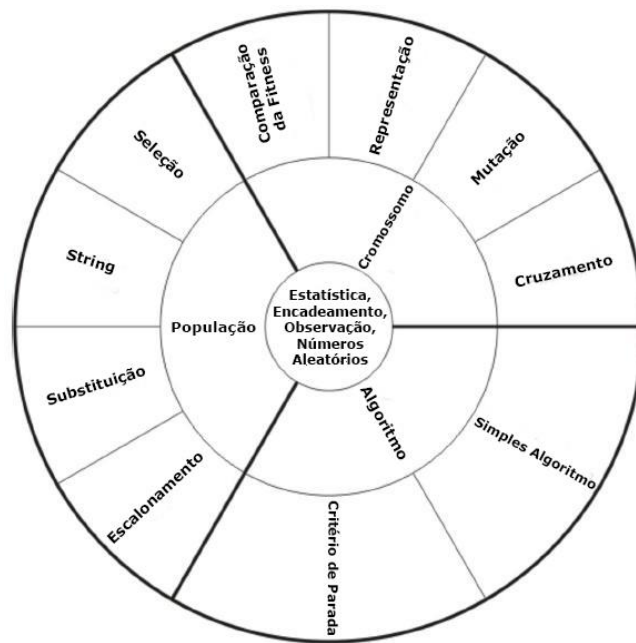


Figura 3.3 Estrutura do AG

Fonte Haupt, (2004 p. 52)

3.2.2.1 Precisão do algoritmo

De um modo geral, os valores de cada variável são convertidos em números binários e, posteriormente, retornam para a base decimal.

A conversão de decimal para binário é realizada, utilizando a Equação 3.1:

$$r = \frac{r_{max} - r_{min}}{2^l - 1} z + r_{min} \quad (3.1)$$

Onde r_{max} e r_{min} são os valores de máximo e mínimo adotados para a variável, l é o número de bits da *string* e z é o valor da *string* gerada de forma aleatória. *String* é o valor dado em forma de um número binário a uma determinada variável.

Por exemplo, se adotássemos uma *string* com um comprimento $l=5$, os valores poderão ser gerados de modo aleatório na faixa de $l=00000$ à $l=11111$.

Exemplo: Uma determinada incógnita poderá variar entre 2,2 e 3,9. O valor decimal da *string* cujo valor aleatório gerado é 10101:

Como $x=10101$ corresponde a $z=21$, utilizando a Equação 3.1, teremos:

$$r = \frac{3,9 - 2,2}{2^5 - 1} 21 + 2,2 = 3,3516$$

Suponha que fosse gerado o próximo número binário subsequente que, neste caso, seria 10110, ou seja, 22. Usando a fórmula da Equação 3.1, o resultado será $r = 3,4065$. Sendo assim, o tamanho da população e a variação entre os valores máximos e mínimos irão influenciar na precisão. Subtraindo $r = 3,4065$ do resultado anterior $r = 3,3516$, os valores para esta variável não poderão ser menores do que 0,0549.

3.2.2.2 População

A população é formada por indivíduos (*strings*) que levam em consideração a quantidade de variáveis que o problema apresenta. As incógnitas M são representadas como *sub-strings* de comprimento l . Assim, as outras *sub-strings* pertinentes às outras variáveis são concatenadas a fim de formar um membro da população, abrangendo todo o conjunto com comprimento L . Desta forma, tem-se:

$$L = \sum_{j=1}^M l_j \quad (3.2)$$

Concatenando os dados segundo o exemplo anterior, com $a=10101$ e $b=10110$, a população para um indivíduo será $c=1010110110$.

3.2.2.3 Mutação

No algoritmo, há um parâmetro denominado Pm (Probabilidade de Mutação). A finalidade da mutação é causar uma perturbação a fim de explorar o domínio de maneira eficiente e escapar do mínimo local. Utilizando o exemplo do item 3.2.2.2, nota-se que a população concatenada equivale a $c=1010110110$. Desta forma, para cada *bit* da *string*, será gerado um número aleatório, caso este número seja menor que Pm o algoritmo irá alterar o número binário de 0 para 1 e vice-versa do *bit* em estudo. Tipicamente o valor adotado é $Pm=0,001$, embora alguns autores como Schaeffer (1989) e Davis (1991) utilizaram valores diferentes deste (COLEY, 1999).

3.2.2.4 Seleção

O operador seleção do AG é baseado na proporcionalidade da *fitness*. Desta forma, a probabilidade de seleção está relacionada ao valor da *fitness* do indivíduo. Os maiores valores têm uma probabilidade maior de seleção em relação aos menores.

Na Figura 3.3, há uma população (*string*) composta de 6 itens. A cada iteração, a *fitness* é calculada em função dos valores da população (*string*) corrente. O valor encontrado da *fitness* para cada item é somado. Com o valor total, pode-se obter a porcentagem de cada item. Na Figura 3.3, os valores da *fitness* das *strings* 1, 2, 3, 4, 5 e 6 respectivamente equivalem a 12, 4, 16, 8, 36 e 24. A probabilidade da *string* 5 ser selecionada é maior devido ao maior valor da *fitness*. O tamanho da população (número de *strings*) define o número de seleção, até os cruzamentos formarem uma nova população.

O mecanismo de seleção será repetido duas vezes a fim de encontrar um par que se submeterá, ou não, ao cruzamento.

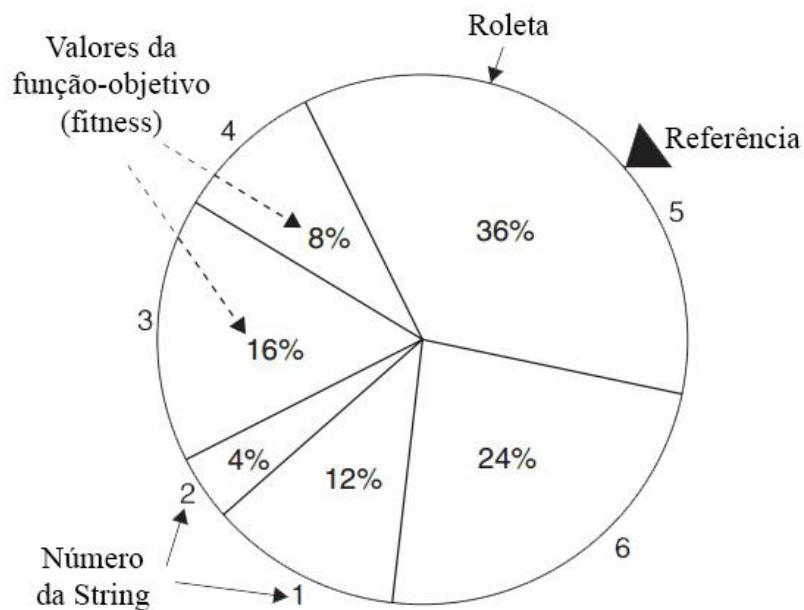


Figura 3.4 Esquema de seleção por *fitness*-proporcional (roleta)

Fonte Adaptado de Rao (2009 p. 698)

3.2.2.5 Elitismo

O critério de seleção que o AG adota é chamado de *fitness*-proporcional. Entretanto, tal mecanismo não garante a seleção de nenhum indivíduo em particular, ou seja, pode não incluir o melhor ajuste, a menos, que o indivíduo seja relativamente muito maior que os outros. De outra forma, há uma chance de que um determinado indivíduo não seja selecionado durante o processo de evolução.

Assegurar a sequência (continuidade) dos membros da elite é o que define o *elitismo*. Tal método permite que não somente a elite seja selecionada, mas garante que a elite não seja perdida pela mutação ou cruzamento durante o seu processo de evolução.

3.2.2.6 Cruzamento

O cruzamento que será utilizado neste trabalho será ponto único para a recombinação, embora múltiplos pontos já tenham sido adotados por outros autores, obtendo resultados bastante adequados (SIMON, 2013).

A probabilidade de cada um dos indivíduos que se submeterá ao cruzamento será definida pela variável P_c (Probabilidade de Cruzamento). Um número aleatório R_c será gerado na faixa de 0 à 1 e os indivíduos se submeterão ao cruzamento caso $R_c \leq P_c$; caso contrário, não haverá cruzamento entre os pares. Os valores típicos de P_c são de 0,4 à 0,9.

Por exemplo, se $P_c=0,5$ a metade da nova população serão formadas pela seleção e posteriormente cruzamento, e a outra metade somente pela seleção.

O cruzamento definido por um único ponto procede, quando a string é cortada em um único ponto, escolhido aleatoriamente, e unido para formar às novas strings.

Exemplo:

Pais:	1010 / 0010101	Filhos:	1010 / 1111111
	1111 / 1111111		1111 / 0010101

3.2.2.7 Escalonamento

O escalonamento linear *fitness* trabalha em função dos valores médios das *fitness*. Isto permite que haja uma evolução constante e de modo proporcional na direção de um ótimo global. Normalmente, diversos autores trabalham com valores entre 1.0 e 2.0. Por exemplo, quando há um valor de escalonamento $C_m=2.0$, aproximadamente duas vezes o número dos melhores indivíduos avançará para a próxima geração do que a média dos indivíduos.

3.3 GAnsys

O software, desenvolvido pelo autor desta dissertação, com o nome de GAnsys (Genetic Algorithm with Ansys), tem como característica a utilização do algoritmo genético para resolução de problemas relacionados a otimização de forma, em conjunto com o software de elementos finitos. O APÊNDICE D mostra o programa desenvolvido.

O programa foi criado, utilizando a linguagem Fortran[®]. A cada iteração é criado um arquivo que servirá como dado de entrada para o posicionamento dos vértices da forma em estudo, para que a avaliação pelo método dos elementos finitos seja realizada com esta nova configuração. Um programa desenvolvido em APDL irá realizar a leitura das posições definidas anteriormente, informando ao programa de Elementos Finitos. A Figura 3.5 apresenta o fluxograma de interação entre o software de elementos finitos e o software desenvolvido.

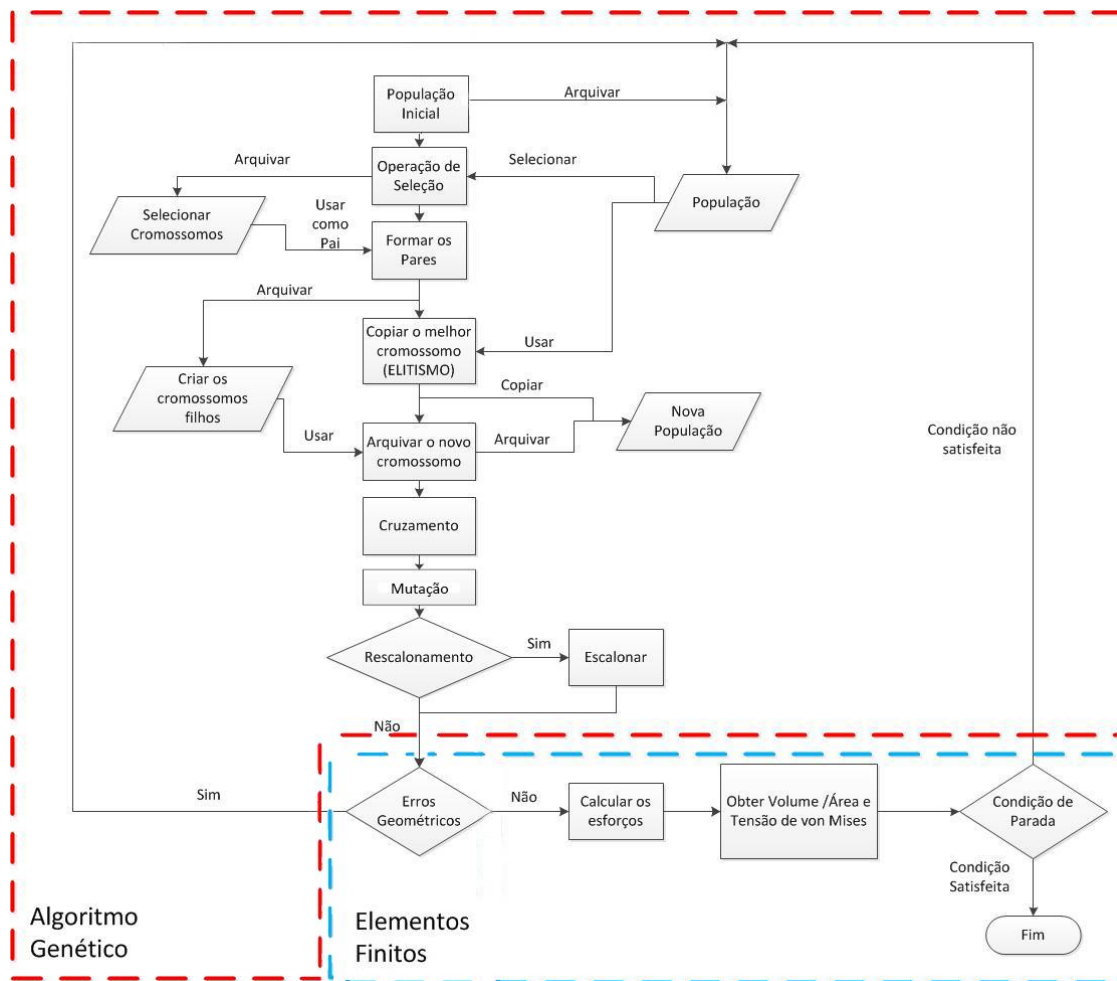


Figura 3.5 Fluxograma do software desenvolvido, mostrando a interação entre o programa de AG e o software de elementos finitos Ansys®.

3.4 Métodos

Como foi visto, existem diversos estudos em que o algoritmo é modificado a fim de ganhar desempenho, isto é, melhorar sua eficiência, diminuindo seu tempo computacional. Entretanto, o algoritmo para o estudo em questão foi implementado, adotando as configurações originais, isto é, nenhuma rotina de aprimoramento da eficiência foi usada, dado que este trabalho é o primeiro do grupo ligado à otimização de forma. Trabalhos futuros poderão empregar tais métodos para

aprimorar o que foi desenvolvido nesta dissertação. A Figura 3.6 é um fragmento do programa desenvolvido no APÊNDICE D.

```
PROGRAM GANSYS
USE DFPOR

----- AJUSTAR OS PARÂMETROS CORRETAMENTE PARA UM MELHOR DESEMPENHO -----

IMPLICIT REAL (A-H, J-Z)
INTEGER GENER, PSIZE, NUNKNO, SUBLEN, TSLENG, MAXG, NEWIND, MATE1,
+ MATE2, FITIND

Com o intuito de mudar a sequencia randomica, o ISEED devera ser alterado, desta forma, mudando o RND().
Deve ser PAR.
PARAMETER(Psize = 20)
PARAMETER(NUNKNO = 32)
Todas as sub-strings tem o mesmo comprimento.
PARAMETER(SUBLEN = 10)
PARAMETER(TSLENG = NUNKNO*SUBLEN)
PARAMETER(MAXG = 1000)
PARAMETER(CP = 0.50)
PARAMETER(MP = 0.001)
CHARACTER*3 ELITE
PARAMETER(ELITE = 'on')
PARAMETER(SCALEC = 1.2)

-----DECLARAR TODAS AS VARIÁVEIS (GLOBAIS)-----

Esta ARRAY define a tamanho do individuo da populacao atual
REAL UNKNO(Psize, NUNKNO), FIT(Psize)
INTEGER INTEG(SIZE, NUNKNO)
INTEGER STRING(Psize, TSLENG)

A nova populacao.
INTEGER NSTRIN(Psize, TSLENG)

Esta ARRAY define o RANGE das incognitas.
REAL RANGE(2, NUNKNO)
```

Figura 3.6 Detalhe parcial do software implementado

A comunicação, entre o software desenvolvido e o Ansys®, é realizada através de um arquivo batch (lote) que é chamado a cada iteração, durante a execução do programa, conforme Figura 3.7. Existe uma linha de comando no GAnsys que chama o arquivo batch. Este arquivo, por sua vez, tem a finalidade de abrir o arquivo desenvolvido em APDL que executará o estudo em elementos finitos da iteração corrente.

```
@echo off

"C:\Program Files\Ansys Inc\V100\ANSYS\bin\intel\ansys100" -b -i E:\Mestrado\GA\Placa_com_Furo\placacomfuro.txt -o
E:\Mestrado\GA\Placa_com_Furo\placacomfuro.out
```

Figura 3.7 Detalhe do arquivo batch

Nessa dissertação, como já dito, o objetivo é encontrar um perfil ótimo dado um determinado problema, utilizando as ferramentas modernas de otimização. O software comercial Ansys® realizará a simulação dos esforços físicos pelo MEF. No software Ansys® já existem algumas ferramentas relacionados à otimização. Comparar tais ferramentas com as novas ferramentas voltadas à otimização, especificamente o AG, faz parte da proposta para validar os resultados. Além disso, é importante também verificar as influências dos principais parâmetros que existem no algoritmo proposto. Segundo Ansys, (2009), dos algoritmos que o software possui para estudos de otimização, o algoritmo que abrange uma gama maior de aplicações, incluindo variações geométricas, é o método por subproblema. Assim, tal método será utilizado para validar os estudos de otimização que foram desenvolvidos. Obviamente, ao se comparar uma proposta com um método disponível em um programa comercial há o risco de que o próprio método adotado como base não seja acurado o suficiente, ou eficiente. Na falta de outras formas de comparação, para essa dissertação, foi considerada adequada comparação do procedimento proposto com o disponível no programa comercial.

Nem sempre com a utilização de ferramentas tradicionais é possível encontrar o ponto mínimo/máximo global. Situações mais complexas requerem que diversas considerações adicionais sejam feitas. Acredita-se que o método aplicado pelo software não atenderá parte dos problemas propostos devido à quantidade de variáveis. Busca-se avaliar se o algoritmo genético pode ser adequado em tais casos.

3.5 Definição do problema

Nesse tópico, são apresentados diversos estudos bidimensionais e tridimensionais. As geometrias iniciais são definidas, conforme as imagens demonstradas nos estudos. Todos os estudos serão analisados, utilizando o algoritmo desenvolvido pelo Ansys® chamado de método por subproblema e o AG. O objetivo é avaliar o desempenho e fazer um comparativo entre os métodos.

3.5.1 Primeiro estudo de caso

Para o primeiro estudo de caso, seja uma estrutura bidimensional como, por exemplo, a da Figura 3.8, de modo que a geometria possa ser definida pelos seus vértices. A estrutura estará sujeita a cargas no plano de modo estático com restrições em alguns pontos. As cargas aplicadas e as restrições serão definidas como condição de contorno. A estrutura deverá ser capaz de resistir às cargas sem falhas. Isso significa que o material deverá trabalhar abaixo do limite de escoamento, para esta análise específica. Além disso, nenhuma parte da estrutura poderá perder sua conexão com os apoios, ou seja, a estrutura deverá permanecer unida. O objetivo será minimizar o volume (ou peso) da estrutura, em outras palavras, a obtenção de uma configuração (forma) com o uso mais eficiente dos materiais sem violar as restrições comportamentais e geométricas. O problema de otimização de peso (minimização) pode ser formulado como segue:

Dado: Posições iniciais dos pontos, forças aplicadas, restrições e propriedades dos materiais.

Pesquisar: A forma ideal de estrutura.

Minimizar: A = área da estrutura.

Sujeito à: (i) a restrição da tensão ($\sigma_{\max} < 400 \text{ MPa}$).

(ii) a restrição geométrica (modelo conectividade).

Variáveis de Projeto: As posições dos pontos.

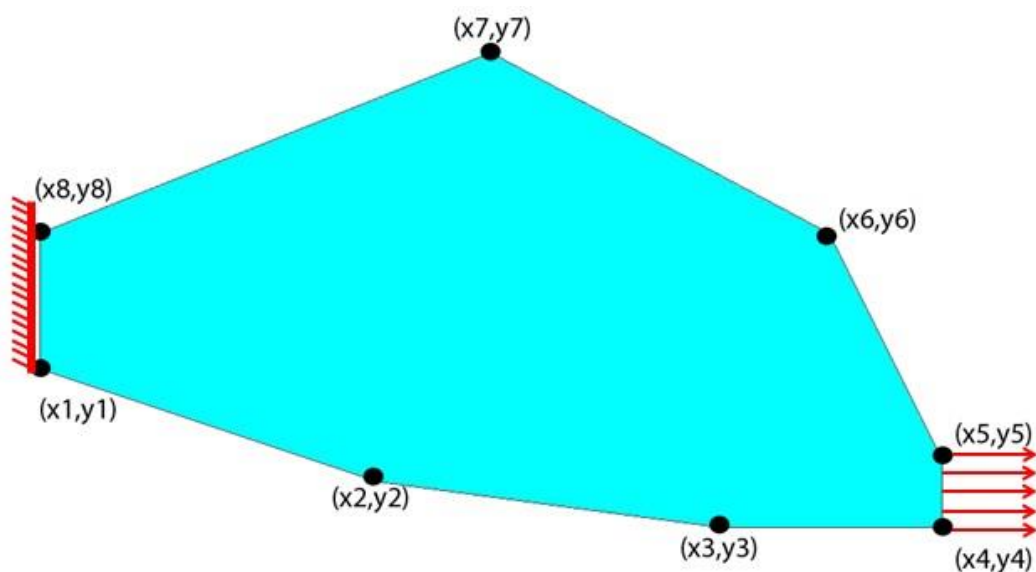


Figura 3.8 Representação da condição de contorno inicial para uma placa bidimensional

A tensão equivalente máximo (von Mises) da estrutura ($\sigma_{\max}$) deverá ser igual ou inferior à tensão máxima admissível (σ_{adm}) que, para o primeiro estudo de caso foi adotado $\sigma_{\max} < 400$ MPa e para o segundo estudo $\sigma_{\max} < 250$ MPa. O limite de escoamento do material para o terceiro estudo de caso será de $\sigma_{\max} < 1000$ MPa. Os pontos serão usados para definir os limites da estrutura e deverão mover-se sem a perda de conectividade.

Os atuais métodos de otimização, nos quais se pode incluir o AG, normalmente requerem a geração de um número aleatório para a definição de uma nova configuração (ou uma nova forma) a cada iteração. Assim, uma configuração posterior deverá ser definida com base na otimização das variáveis. A forma do componente é definida pelos seus limites e sua espessura. Na sua forma, poderá ser desenvolvida por uma curva do tipo *spline* ou por segmentos de reta, passando por uma série de pontos (nós).

Como ilustrado na Figura 3.8, as posições dos pontos poderão ser deslocadas em direções aleatórias através de distâncias também aleatórias. Desta forma, um novo contorno é obtido, ou seja, uma nova configuração de pontos. As coordenadas X, Y e Z dos pontos que se deslocam são as variáveis de otimização.

O arquivo APDL, criado para trabalhar no Ansys®, tem como finalidade ler a posição das

variáveis e realizar os estudos de pré e pós-processamento. Isto é, gerar o modelo a partir das variáveis definidas pelo GAnsys, criar a malha, calcular os esforços e mostrar os resultados. A Tabela 3.2 mostra os valores de máximo e mínimo admissível definido para cada variável para o primeiro estudo de caso. Uma das virtudes das ferramentas modernas de otimização é que elas se adaptam para problemas em que os limites máximos e mínimos das variáveis são elevados. Rao, (2009) menciona que os métodos tradicionais normalmente encontram dificuldades de convergência em problemas com tais características.

Tabela 3.2 Valores de máximo e mínimo de cada variável para o primeiro estudo de caso

NÓ	X		Y	
	MÍN.	MÁX	MÍN.	MÁX
1	0	0	50	50
2	0	100	10	80
3	110	175	5	70
4	160	240	5	5
5	160	240	20	20
6	120	195	50	100
7	30	120	40	140
8	0	0	100	100

3.5.2 Segundo estudo de caso

Como um segundo estudo de caso será realizado, um estudo bidimensional, a geometria do modelo é relativamente simples, conforme Figura 3.9. Entretanto, esse estudo apresenta uma quantidade elevada de variáveis, quando comparado ao problema estudado anteriormente. Os limites são mostrados na Tabela 3.3. Os procedimentos serão os mesmos. Como não há grandes variações nos limites superiores e inferiores das variáveis deste estudo, acredita-se que o método por subproblema possa ser eficaz. No entanto, a elevada quantidade de variáveis pode impossibilitar um bom desempenho do programa comercial. A tensão máxima adotada para o estudo será de $\sigma_{\max} < 250$ MPa.

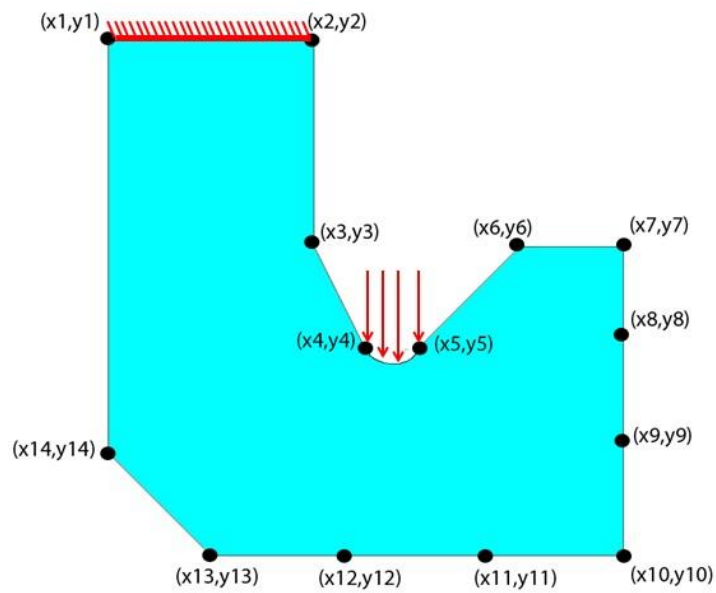


Figura 3.9 Representação da condição de contorno inicial de um gancho bidimensional

Tabela 3.3 Valores de máximo e mínimo de cada variável para o segundo estudo de caso

NÓ	X		Y	
	MÍN.	MÁX	MÍN.	MÁX
1	0	0	0	0
2	50	120	0	0
3	65	115	35	75
4	80	120	80	125
5	125	125	150	180
6	150	150	150	180
7	200	200	75	210
8	220	260	75	210
9	220	280	150	180
10	190	270	170	220
11	150	260	210	300
12	100	180	205	300
13	60	125	225	300
14	70	130	190	300
15	0	60	170	250
16	0	60	90	210

3.5.3 Terceiro estudo de caso

O terceiro estudo de caso será um estudo tridimensional, conforme Figura 3.10. Entretanto, o modelo será mais complexo devido às suas características geométricas. A quantidade de variáveis é relativamente pequena se for comparada com o analisado no problema estudado anteriormente. Contudo, o fato de a geometria ser tridimensional torna-se mais complexo o problema. Os procedimentos serão os mesmos, porém as dimensões dos volumes serão as variáveis para o estudo em questão. A Tabela 3.4 mostra os limites de cada variável. A tensão adotada para este estudo de caso será de $\sigma_{\max} < 1000 \text{ MPa}$.

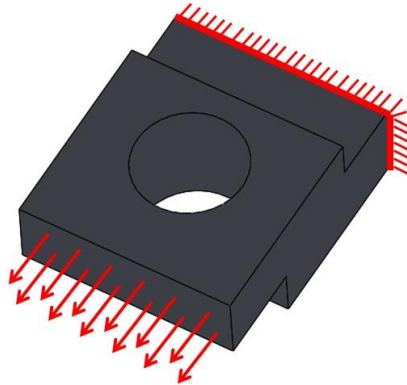


Figura 3.10 Representação da condição de contorno inicial para um componente tridimensional

Tabela 3.4 Valores de máximo e mínimo de cada variável para o terceiro estudo de caso

NÓ	Dimensão		Descrição
	MÍN.	MÁX	
1	20	50	Largura X1
2	30	80	Comprimento Y1
3	10	25	Posição Furo X2
4	15	40	Posição Furo Y2
5	2	20	Diâmetro D1
6	10	40	Extrusão 1
7	30	50	Dimensão Corte X3
8	5	35	Extrusão 2 (Corte)
9	30	50	Dimensão Corte Y3
10	5	35	Extrusão 3 (Corte)

Nas imagens da Figura 3.11, são mostradas as variáveis que serão estudadas para o terceiro estudo de caso.

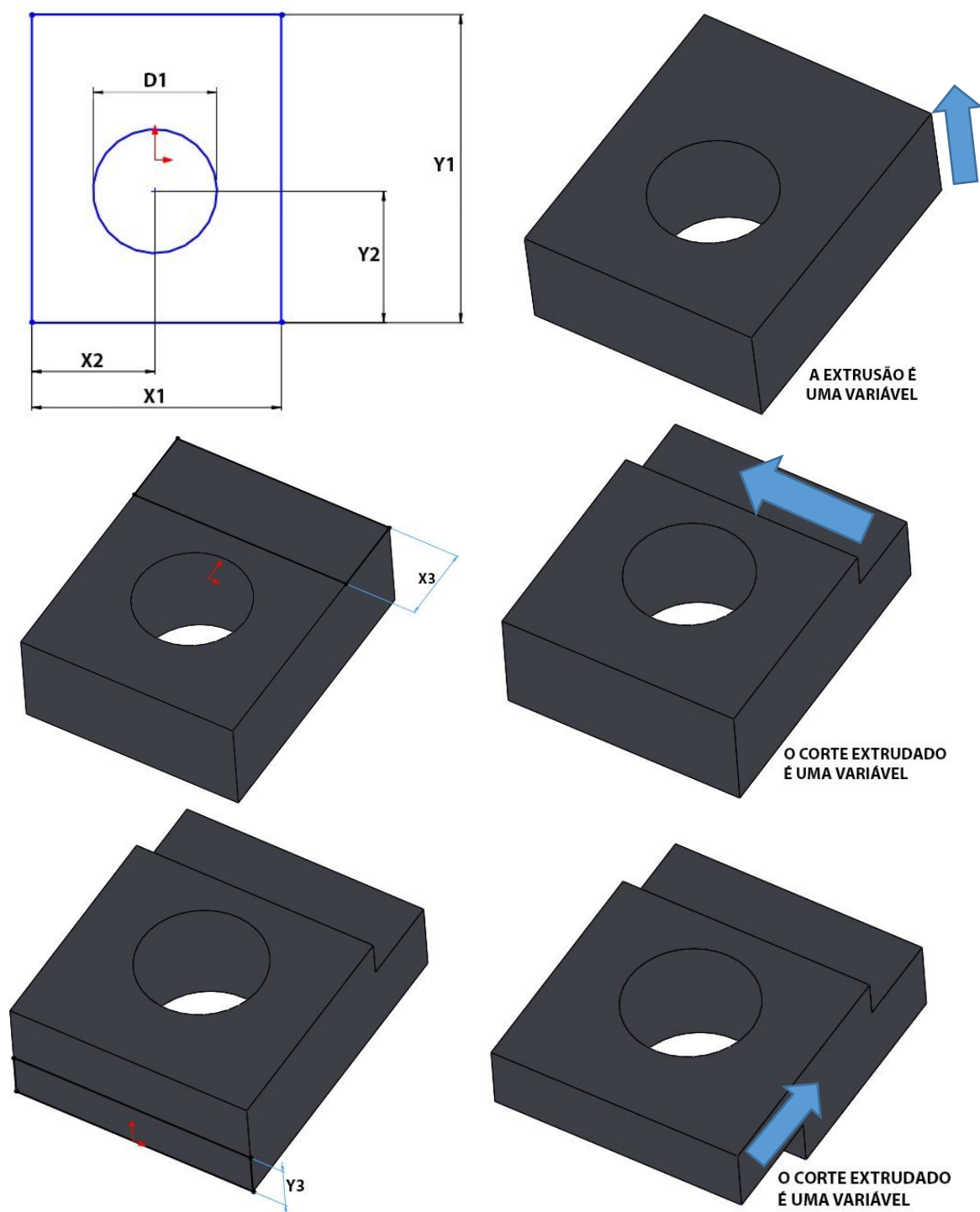


Figura 3.11 Demonstração das variáveis para o terceiro estudo de caso

3.5.4 Quarto estudo de caso

Para o quarto estudo de caso, será estudado uma placa tridimensional com um furo conforme Figura 3.12. Devido à quantidade de variáveis serem elevadas, acredita-se que o algoritmo, utilizado pelo Ansys®, encontre dificuldades para sua convergência. Os autores Annicchiarico e Cerrolaza (2001) realizaram um estudo similar. O objetivo deste estudo é comparar os resultados encontrados pelos autores citados com os obtidos com a ferramenta proposta nesta dissertação.

Para este estudo, os nós ao longo do contorno, além do diâmetro e a espessura da placa, são as variáveis de projeto. Assim, a posição do furo na placa e o seu diâmetro, além de todo o contorno da placa, definem a geometria a ser estudada, a Tabela 3.5 mostra os limites admissíveis. A força será aplicada na aresta superior da geometria e será engastado no lado oposto.

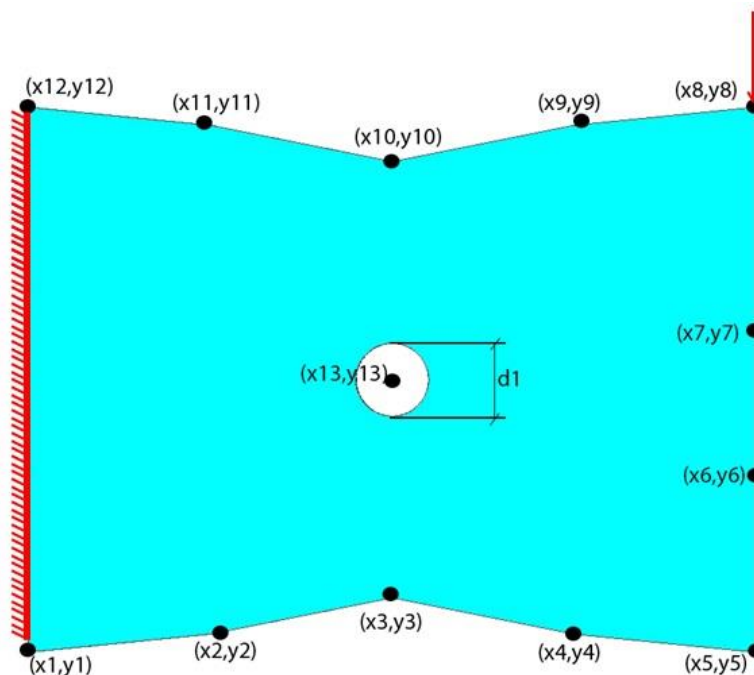


Figura 3.12 Representação da condição de contorno inicial para o quarto estudo de caso

Tabela 3.5 Valores de máximo e mínimo de cada variável para o quarto estudo de caso

NÓ	X		Y	
	MÍN.	MÁX	MÍN.	MÁX
1	0	0	-75	-75
2	80	120	-75	-15
3	180	220	-75	-15
4	280	320	-75	-15
5	380	500	-75	-20
6	380	500	-25	-5
7	380	500	5	25
8	380	500	20	75
9	280	320	15	75
10	180	220	15	75
11	80	120	15	75
12	0	0	75	75
13	100	300	0	0
14	5	30	Diâmetro	
15	50	250	Espessura	

4. RESULTADOS E DISCUSSÕES

Serão demonstrados a influência das alterações dos parâmetros do AG nos resultados para todos os estudos de caso, o impacto do tamanho da população no tempo computacional e nos resultados encontrados e a dependência dos processadores com alto desempenho para a viabilidade das ferramentas metaheurísticas.

4.1 Estudos de otimização de forma

Os resultados obtidos, para todos os estudos de caso propostos, serão apresentados neste capítulo. Detalhes referentes a resultados, ao longo das iterações para cada configuração, discretização do domínio e influência da população em função do tempo computacional, poderão ser encontrados nos APÊNDICES A, B e C.

4.1.1 Estudo de uma placa bidimensional

Diversas configurações do AG foram alteradas e avaliadas com o objetivo de verificar o impacto de tais alterações. A proposta de realizá-las se deve ao fato de que, dependendo do problema estudado, uma determinada configuração pode ser mais eficiente. Desta forma, é possível verificar a influência das alterações mencionadas em problemas simples e complexos.

Como visto, a probabilidade dos indivíduos que se submeterão ao cruzamento será definida pela variável P_c (*Probabilidade de Cruzamento*). Por exemplo, se $P_c=0,5$ a metade da nova população será formada pela seleção e posteriormente cruzamento, e a outra metade somente pela seleção. A variável, denominada P_m (*Probabilidade de Mutação*), tem como característica alterar o número binário de 0 para 1 e vice-versa em função de uma determinada

probabilidade. Por exemplo, $P_m=0,001$ um número, a cada mil, sofrerá alteração. O *elitismo* tem como característica garantir que os melhores resultados não se percam durante o processo evolutivo. Com o método de seleção apontado (*fitness*-proporcional), pode ocorrer da melhor solução seja jogada fora. O elitismo garante que a elite não seja perdida pela mutação ou cruzamento durante o seu processo evolutivo.

O *escalonamento linear* da *fitness* trabalha em função dos valores médios da *fitness*. Isto permite uma evolução constante e de modo proporcional na direção do ótimo global. Os valores usuais estão entre 1.0 e 2.0. Por exemplo, quando se adota um valor de escalonamento $C_m=2.0$, então aproximadamente duas vezes o número dos melhores indivíduos avançará para a próxima geração do que a média dos indivíduos. Os valores aplicados para a maioria dos estudos foi de $C_m=1.2$.

Para todos os estudos, as seguintes configurações foram analisadas:

1. Probabilidade de Cruzamento 0,65; Probabilidade de Mutação 0,001, Elitismo habilitado e Escalonamento de 1,2
2. Probabilidade de Cruzamento 0,65; Probabilidade de Mutação 0,1, Elitismo habilitado e Escalonamento de 1,2
3. Probabilidade de Cruzamento 0,65; Probabilidade de Mutação 0,3, Elitismo habilitado e Escalonamento de 1,2
4. Probabilidade de Cruzamento 0,9; Probabilidade de Mutação 0,1, Elitismo habilitado e Escalonamento de 1,2
5. Probabilidade de Cruzamento 0,65; Probabilidade de Mutação 0,001, Elitismo desabilitado e Escalonamento de 1,2
6. Probabilidade de Cruzamento 0,65; Probabilidade de Mutação 0,001, Elitismo habilitado e Escalonamento de 2
7. Probabilidade de Cruzamento 0,9; Probabilidade de Mutação 0,001, Elitismo habilitado e Escalonamento de 1,2

Na sequência, serão mostradas as geometrias obtidas para cada estudo. Foram realizadas 200 iterações para todos os estudos. No APÊNDICE A, com o intuito de verificar a evolução do domínio em todas as situações acima, foram geradas imagens nas iterações 1, 25, 50, 100, 150 e 200.

Os resultados encontrados para o estudo de uma placa bidimensional, mencionada no capítulo anterior, serão demonstrados na Figura 4.1.

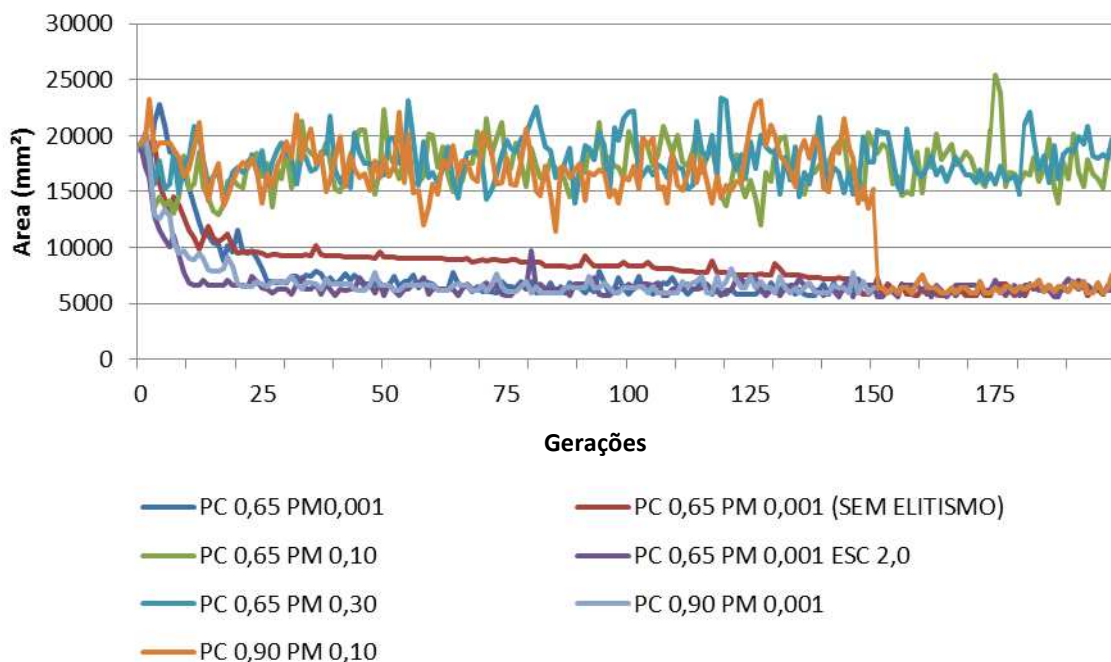


Figura 4.1 Influência das alterações dos parâmetros do AG para o estudo de uma placa bidimensional

Para o estudo bidimensional, situações nas quais a Pm foi elevada para valores maiores do que 10% ocorreram inúmeras perturbações no domínio, de modo que o algoritmo encontrasse dificuldade para sua convergência. No estudo com as configurações $Pc=0,9$; $Pm=0,1$; *Elitismo* habilitado e com *Escalonamento de 1,2*, (linha laranja), aparentemente parecia que não iria haver convergência, porém, na iteração 150, o programa encontrou o ponto ótimo global. Para este estudo, a quantidade de iterações necessárias para a obtenção do ponto ótimo global foi de

aproximadamente 6 vezes mais do que os outros resultados em que o valor da P_m foi menor.

No estudo em que não foi habilitado o elitismo, o algoritmo teve um bom desempenho. Entretanto, o algoritmo precisou de uma quantidade 6 vezes maior de iterações para obter os valores mínimos de área.

Na situação em que o escalonamento foi elevado de 1,2 para 2,0, o algoritmo encontrou o ponto ótimo global 50% mais rápido que os demais. Porém, valores muito elevados deste parâmetro pode fazer com que o algoritmo se perca, durante o processo evolutivo, podendo cair em um mínimo local. Devido aos limites máximos e mínimos das variáveis serem elevadas o método de aproximação por subproblema não convergiu nesta configuração. A Figura 4.2 mostra as condições iniciais e finais para o primeiro estudo de caso, a Tabela 4.1 mostra os limites de cada variável e os valores ótimos encontrados.

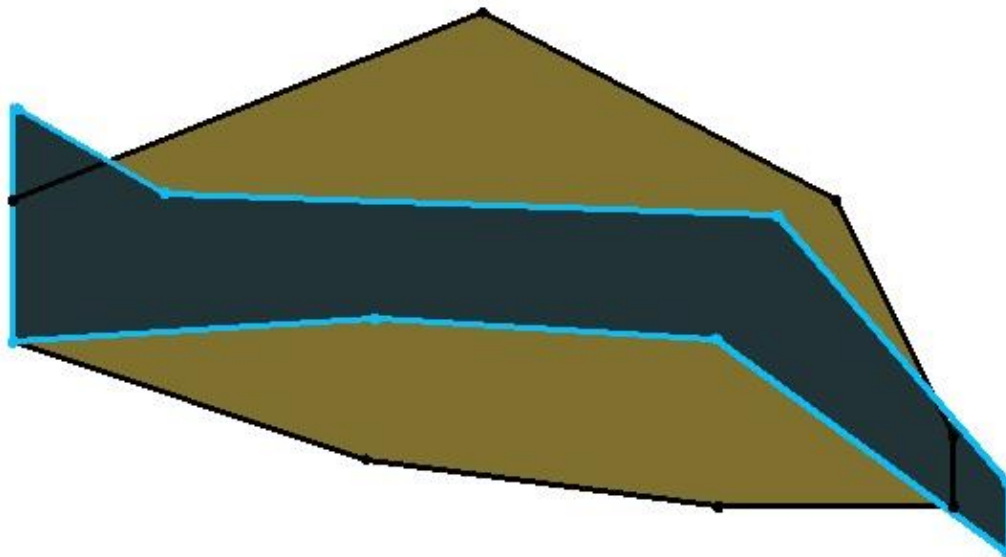


Figura 4.2 Representação das geometrias iniciais e finais de uma placa bidimensional, utilizando o AG como ferramenta de otimização

Tabela 4.1 Limites dos parâmetros máximo, mínimos, e os valores finais para cada variável referente ao primeiro estudo de caso

0	0	0	1	50	50	50
0	77,03	100	2	10	54,82	80
110	149,58	175	3	5	50,56	70
160	211,53	240	4	5	5	5
160	211,53	240	5	20	20	20
120	162,52	195	6	50	76,78	100
30	32,38	120	7	40	81,54	140
0	0	0	8	100	100	100
MÍNIMO	FINAL	MÁXIMO	NÓ	MÍNIMO	FINAL	MÁXIMO
X				Y		

A Tabela 4.2 mostra os valores relacionados a tensão (MPa) e área (mm²) para todas as configurações estudadas no primeiro estudo de caso.

Tabela 4.2 Valores de Tensão e Área para as configurações analisadas no primeiro estudo

	Tensão (MPa)	Área (mm ²)
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 1,2	398,03	6092,98
Pc= 0,65; Pm=0,001; Elitismo OFF e Esc. 1,2	364,93	5590,33
Pc= 0,65; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 2	399,24	6646,84
Pc= 0,65; Pm=0,30; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,90; Pm=0,001; Elitismo ON e Esc. 1,2	392,59	5897,75
Pc= 0,90; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Método por Subproblemas	NÃO CONVERGIU	

4.1.2 Estudo de um gancho bidimensional

No estudo anterior, a ausência da habilitação da função elitismo não impediu com que o algoritmo encontrasse o ponto máximo global. Neste estudo, em que a quantidade de variáveis foi

maior, a desabilitação da função elitismo fez com que o estudo não convergisse. Valores elevados na probabilidade de mutação, da mesma forma que no exemplo anterior, fizeram com que o domínio tivesse uma perturbação excessiva, impedindo que o algoritmo não encontrasse o ponto de convergência.

Uma questão apontada no exemplo anterior e que ocorreu neste estudo, surgiu quando o escalonamento foi elevado de 1,2 para 2,0. Com tal alteração, o algoritmo caiu em um ponto ótimo local não saindo mais ao longo das iterações. Desta forma, também não houve convergência nesta configuração.

Apesar da quantidade de variável ser relativamente alta, o algoritmo subproblema rodou e teve um desempenho de aproximadamente 20% inferior em relação aos melhores resultados encontrados utilizando o AG. Neste segundo estudo de caso, os dois métodos convergiram, a Figura 4.3 mostra a evolução dos comportamentos para cada configuração adotada.

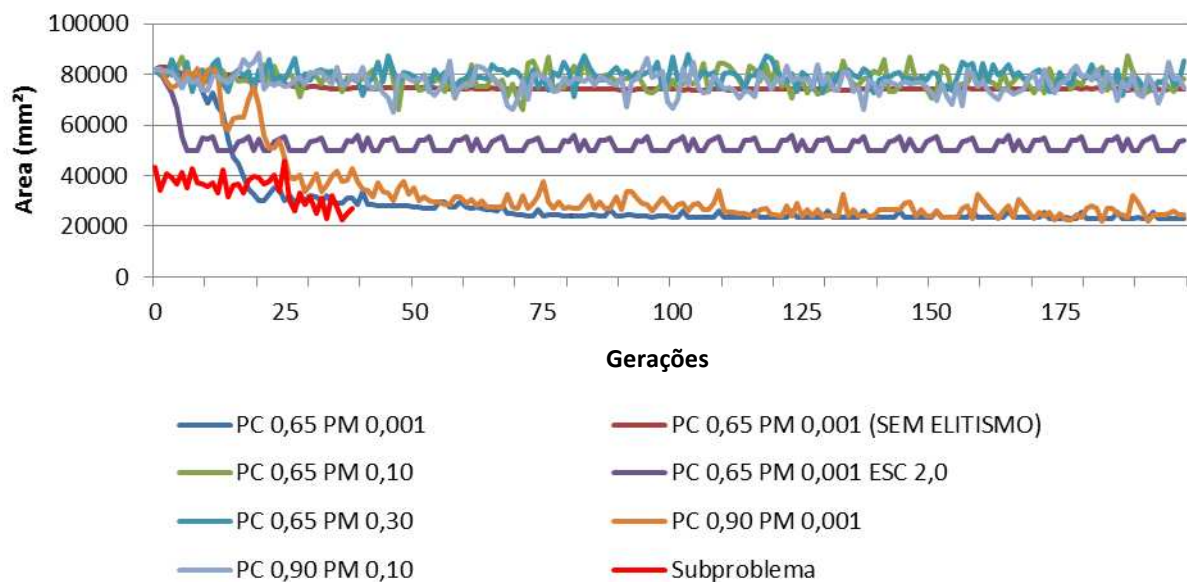


Figura 4.3 Influência das alterações dos parâmetros do AG para um estudo de um gancho bidimensional

A Figura 4.4 mostram as geometrias otimizadas. A imagem em vermelho corresponde à geometria ótima encontrada pelo método de aproximação por subproblema, em azul, o AG. A Tabela 4.3 mostra os limites de cada variável e os valores ótimos encontrados para cada método.

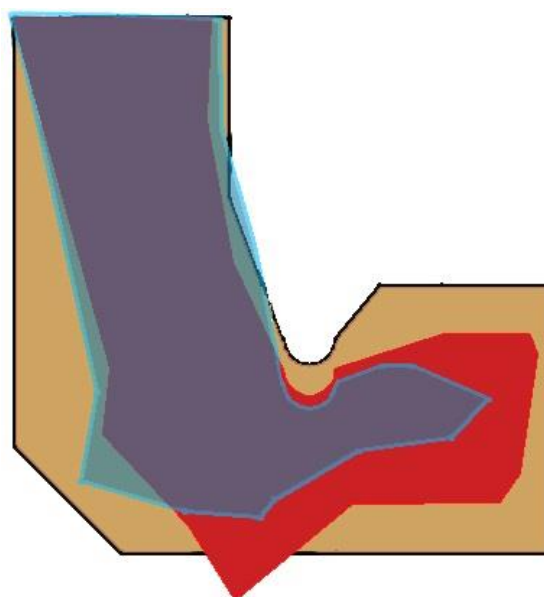


Figura 4.4 Geometrias iniciais e finais de um gancho bidimensional utilizando o AG e o método por subproblemas como ferramentas de otimização

Tabela 4.3 Limites dos parâmetros máximo, mínimos, e os valores finais para cada ferramenta de otimização referente ao segundo estudo de caso

0	0	0,0	0	1	0	0	0,0	0
50	95,85	90,7	120	2	0	0	0,0	0
65	96,67	88,8	115	3	35	51,93	47,6	75
80	111,83	100,8	120	4	80	105,65	113,4	125
125	125	125,0	125	5	150	170,49	164,5	180
150	150	150,0	150	6	150	170,79	164,5	180
160	171,34	200,0	180	7	75	162,55	148,1	210
180	185,83	240,1	260	8	75	162,55	148,1	210
220	221,29	243,5	280	9	150	178,03	156,7	180
190	203,92	235,1	270	10	170	196,25	212,4	220
150	160,65	226,2	260	11	200	202,44	225,7	300
100	121,19	157,1	180	12	205	225,06	226,4	300
60	114,49	103,4	125	13	225	233,42	271,2	300
70	83,08	81,5	130	14	190	230,48	236,3	300
0	30,79	41,5	60	15	170	216,45	193,8	250
0	38,18	44,9	60	16	90	174,69	163,4	210
MÍNIMO	FINAL	FINAL	MÁXIMO	NÓ	MÍNIMO	FINAL	FINAL	MÁXIMO
	(AG)	(Sub)				(AG)	(Sub)	
X					Y			

A Tabela 4.4 mostra os valores relacionados referentes à tensão (MPa) e área (mm²) para todas as configurações, utilizadas para o segundo estudo.

Tabela 4.4 Valores de Tensão e Área para as configurações analisadas no segundo estudo

	Tensão (MPa)	Área (mm ²)
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 1,2	244	23113,13
Pc= 0,65; Pm=0,001; Elitismo OFF e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,65; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 2	NÃO CONVERGIU	
Pc= 0,65; Pm=0,30; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,90; Pm=0,001; Elitismo ON e Esc. 1,2	248	21648,71
Pc= 0,90; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Método por Subproblemas	226	27029,35

4.1.3 Estudo de um modelo tridimensional

Neste estudo, não há uma quantidade elevada de variáveis, característica dos estudos anteriores. Entretanto, o estudo foi realizado a partir de um modelo tridimensional desenvolvido de forma paramétrica. Assim, as dimensões de cada geometria são as variáveis do estudo. Similarmente, aos estudos anteriores, um valor elevado na *Pm* (probabilidade de mutação) levou a uma alta perturbação no modelo, ocasionando assim sua não convergência.

Para o estudo em questão, a ausência da habilitação da função elitismo fez com que o modelo encontrasse o mínimo global em um tempo menor do que de forma habilitada.

No estudo em que o escalonamento foi elevado de 1,2 para 2,0, o algoritmo encontrou o ponto ótimo global aproximadamente 50% mais rápido que a configuração *Pc*=0,65, *Pm*=0,001, com *escalonamento* e com a opção *elitismo* ativada, situação muito parecida em relação ao primeiro estudo. O método de aproximação por subproblema não convergiu nesta situação. A Figura 4.5 mostra a evolução dos comportamentos para cada configuração adotada.

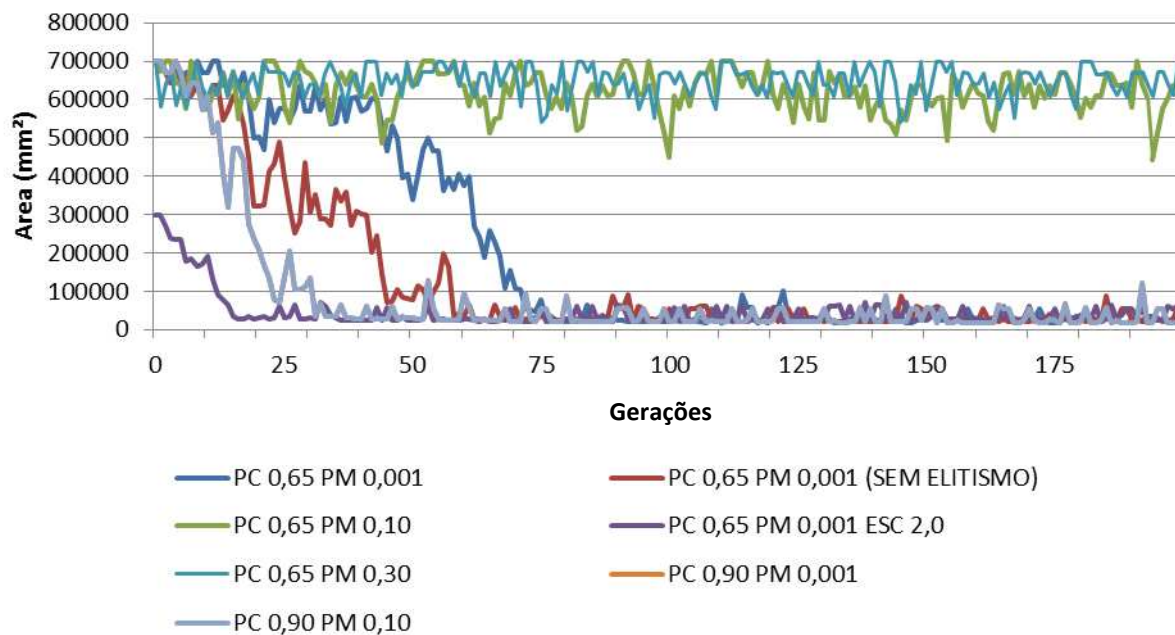


Figura 4.5 Influência das alterações dos parâmetros do AG para um estudo de uma placa com furo tridimensional

Devido aos valores elevados dos resultados que não convergiram nos estudos da Figura 4.5, a finalidade da Figura 4.6 é a de mostrar a evolução para os estudos em que houve convergência.

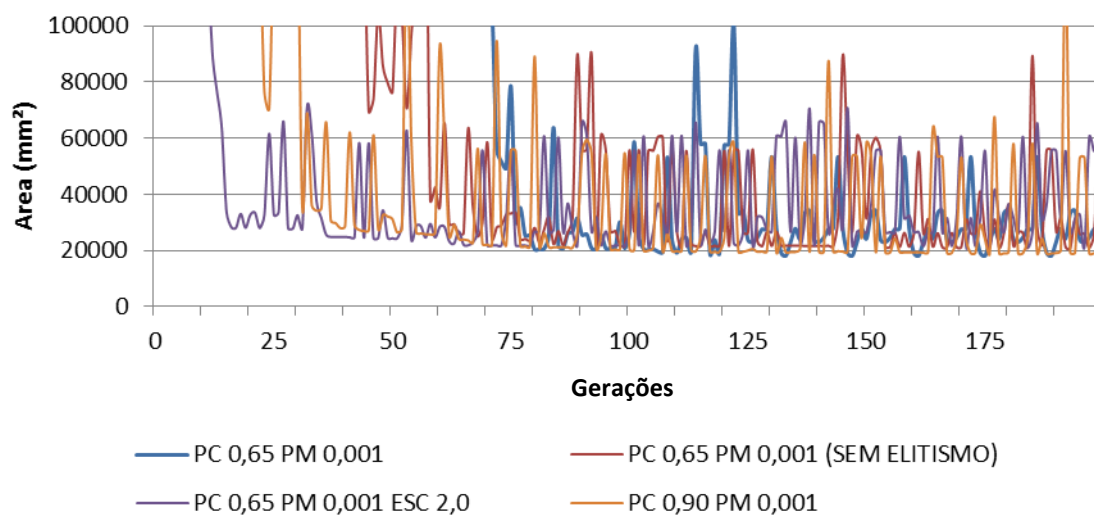


Figura 4.6 Detalhe da evolução para o primeiro estudo de caso

A Figura 4.7 demonstra a geometria otimizada sobreposta na condição original. A Tabela 4.5 mostra os limites adotados para cada variável e o valor ótimo encontrado.

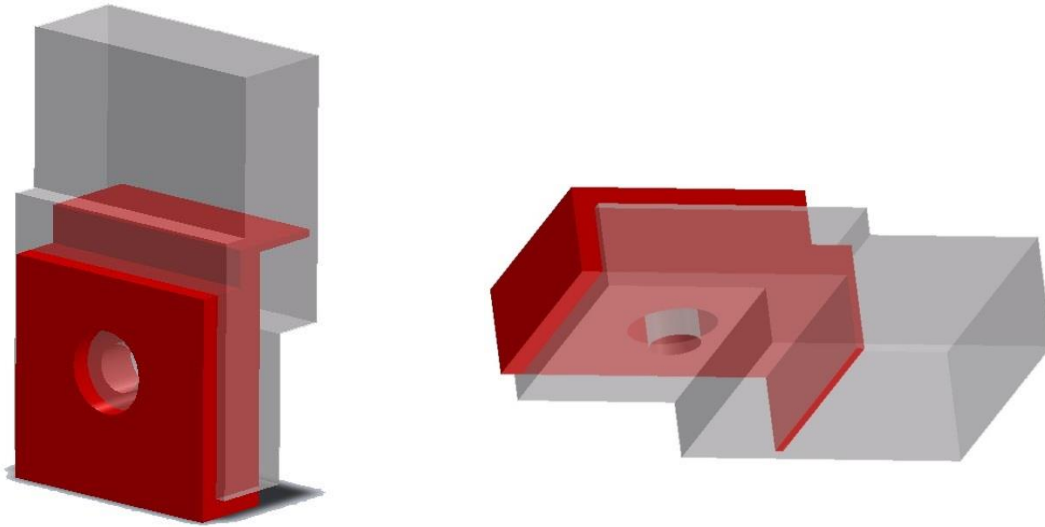


Figura 4.7 Representação das geometrias iniciais e finais para um estudo tridimensional, utilizando o AG como ferramenta de otimização

Tabela 4.5 Limites dos parâmetros máximo, mínimos e os valores finais para cada variável referente ao terceiro estudo de caso

NÓ	Dimensão			Descrição
	MÍN.	FINAL	MÁX	
1	20	29,38	50	Largura X1
2	30	49,35	80	Comprimento Y1
3	10	19,72	25	Posição Furo X2
4	15	21,21	40	Posição Furo Y2
5	2	3,28	20	Diâmetro D1
6	10	35,66	40	Extrusão 1
7	30	47,43	50	Dimensão Corte Y3
8	5	7,17	35	Prof. Rasgo 1
9	30	41,69	50	Dimensão Corte Y4
10	5	31,13	35	Prof. Rasgo 2

Na Tabela 4.6, os valores referentes à tensão (MPa) e volume (mm³) para todas as configurações, utilizadas ao terceiro estudo, são apontados.

Tabela 4.6 Valores de Tensão e Volume para as configurações analisadas no terceiro estudo

	Tensão (MPa)	Volume (mm³)
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 1,2	895,58	13014,82
Pc= 0,65; Pm=0,001; Elitismo OFF e Esc. 1,2	980,94	24615,99
Pc= 0,65; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,65; Pm=0,001; Elitismo ON e Esc. 2	937,21	21553,23
Pc= 0,65; Pm=0,30; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Pc= 0,90; Pm=0,001; Elitismo ON e Esc. 1,2	913,51	18980,95
Pc= 0,90; Pm=0,10; Elitismo ON e Esc. 1,2	NÃO CONVERGIU	
Método por Subproblemas	NÃO CONVERGIU	

4.2 Influências da alteração do parâmetro população

A finalidade da população é a de aumentar a precisão das variáveis, explorando melhor o domínio a ser estudado, como visto no capítulo 2.

Entretanto, o aumento da população está intimamente ligada ao tempo computacional, pois o tamanho da população está relacionada à quantidade de estudos efetuados a cada iteração antes que se possam aplicar os critérios de seleção subsequentemente.

Nessa etapa do estudo, busca-se verificar a influência do aumento da população em relação ao tempo computacional e também analisar a qualidade dos resultados encontrados para cada situação.

Novamente, será estudada a placa bidimensional analisada anteriormente. O objetivo de utilizá-la se deve ao fato de que o tempo computacional para uma população elevada aumentará exponencialmente. Assim, tal seleção está relacionada somente à quantidade de variáveis e à simplicidade deste estudo.

A condição de contorno é a mesma adotada no primeiro estudo de caso. O valor da probabilidade de cruzamento e da probabilidade de mutação é de $P_c=0,65$ e $P_m=0,001$, respectivamente. O *escalamento* utilizado foi de 1,2 e com a função *elitismo habilitada*. Tal condição foi a que melhor se comportou nos estudos realizados anteriormente. Em todos os estudos a seguir, foram utilizados os mesmos parâmetros. Foram utilizadas populações de 20, 50, 80 e 100. As imagens com as evoluções para todos os casos encontram-se no APÊNDICE C.

A Figura 4.8 mostra a evolução dos resultados em função do aumento da população, a Figura 4.9 aponta o tempo computacional em função do tamanho da população, a Tabela 4.7 demonstra os resultados finais encontrados em cada situação.

Todos os estudos foram realizados em um computador Intel® Celeron® CPU 287 @ 1,10 GHz, com dois processadores e com 8 GB de memória RAM.

Tabela 4.7 Resultados obtidos em função do aumento da população

População	Tensão (MPa)	Área (mm ²)	Tempo (min.)
20	351,57	7110,01	283
50	446,91	5371,87	681
80	389,27	5204,41	1104
100	394,02	4706,31	1677

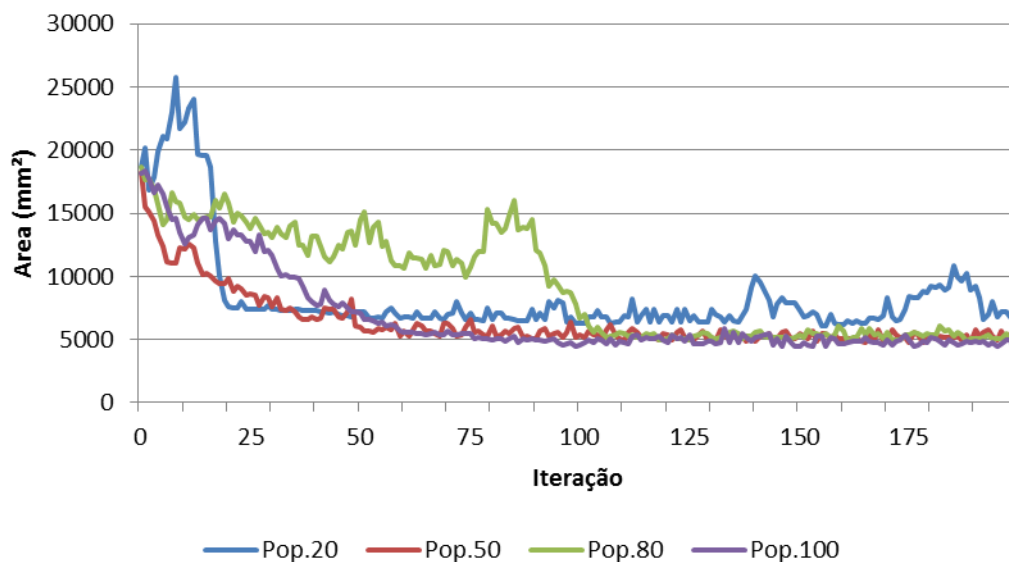


Figura 4.8 Influência da população em função do mínimo global

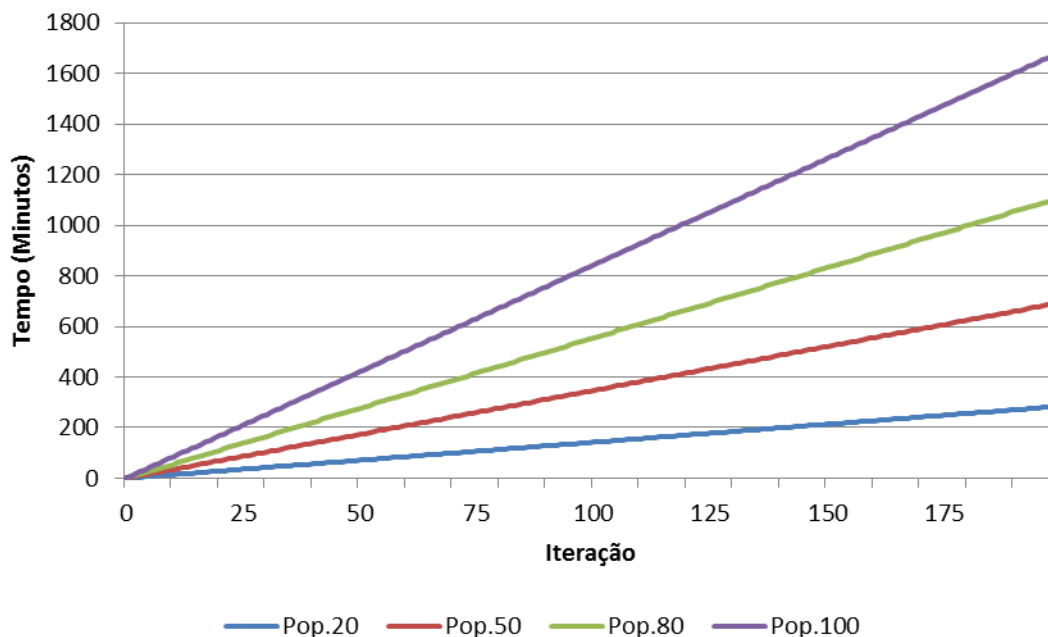


Figura 4.9 Tempo computacional em função da variação da população

O tempo computacional em otimizações, envolvendo os métodos metaheurísticos é sua principal desvantagem. Devido à exploração maior no domínio a ser estudado, inevitavelmente, o tempo computacional tende a ser maior. Com os recursos computacionais disponíveis atualmente, a tendência é que este tempo diminua cada vez mais. Para o caso de todos os algoritmos modernos de otimização apontados neste trabalho, apesar das particularidades de cada um deles, não há uma complexidade muito elevada na sua implementação. Desta forma, os processadores com múltiplos núcleos são muito eficientes para este tipo de estudo, pois cada núcleo trabalhará com uma determinada quantidade de tarefas.

Utilizando o software desenvolvido pelo autor, o primeiro estudo de caso foi analisado em diversas máquinas para verificar o impacto dos processadores, núcleos entre outras características na obtenção de uma resposta. O estudo foi realizado, utilizando os seguintes equipamentos:

- Intel(R) Pentium(R) Dual CPU E2180 @2 GHz e 1 Gb de memória RAM;
- Intel(R) Celeron(R) CPU 847 @1,1 GHz e 8 Gb de memória RAM;
- Intel(R) Core(TM) i5-2400 @3,1 GHz e 4 Gb de memória RAM.

A Figura 4.10 mostra o desempenho do algoritmo nos equipamentos mencionados. Verifica-se que a integração de um software CAD e CAE, em conjunto com as ferramentas metaheurísticas, dependerá de equipamento eficiente.

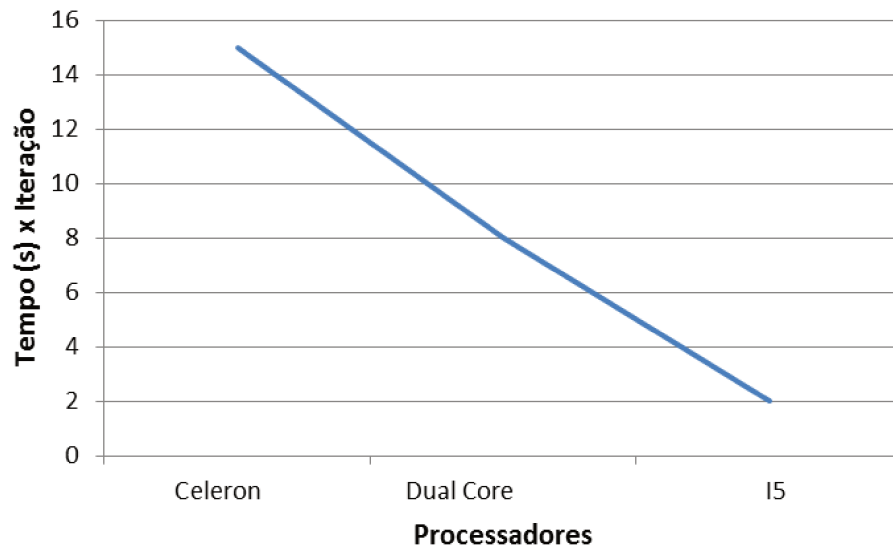


Figura 4.10 Tempo computacional por iteração em função do tipo de processador

4.3. Estudo comparativo em relação a trabalhos realizados anteriormente

Os autores Annicchiarico e Cerrolaza (2001) realizaram um estudo de uma placa com um furo, utilizando o método dos algoritmo genéticos em problemas de otimização de forma. O objetivo deste estudo é comparar os resultados obtido pelos autores com os resultados do trabalho apresentado.

Para tal estudo, os autores utilizaram uma placa com um furo, conforme a Figura 4.11, o material tem um módulo de elasticidade de 200 GPa com um coeficiente de Poisson de 0,3. A barra está engastada em uma extremidade e, na outra extremidade, há uma carga distribuída de 20.000N, ao longo da aresta superior, o limite de escoamento adotado pelos autores foi de 3000 MPa :

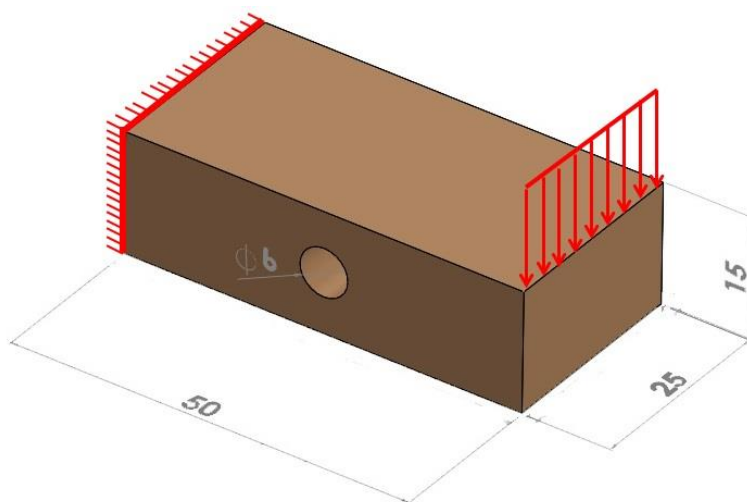


Figura 4.11 Condição de contorno do modelo proposto para o quarto estudo de caso

Fonte Annicchiarico e Cerrolaza, (2001)

A Tabela 4.8 mostra as configurações iniciais no método do AG para o estudo.

Tabela 4.8 Métodos adotados por ambos os autores

	Annicchiarico e Cerrolaza	Almeida, A.B.
População	100	100
Número de Gerações	50	50
Esquema de Cruzamento	Cruzamento “Single Point”	Cruzamento “Single Point”
Probabilidade de Cruzamento	0,80	0,80
Probabilidade de Mutação	0,005	0,005
Escalonamento	1,2	1,2
Elitismo	Habilitado	Habilitado

O volume encontrado pelos autores op. cit., nas configurações mencionadas na Tabela 4.8, foi de 9314,34 mm³ após as 50 iterações. Utilizando as mesmas configurações, porém com o software implementado, o volume obtido foi 8705,26 mm³, aproximadamente (6,5%) menor, a Figura 4.12 mostra o resultado obtido pelo autor.

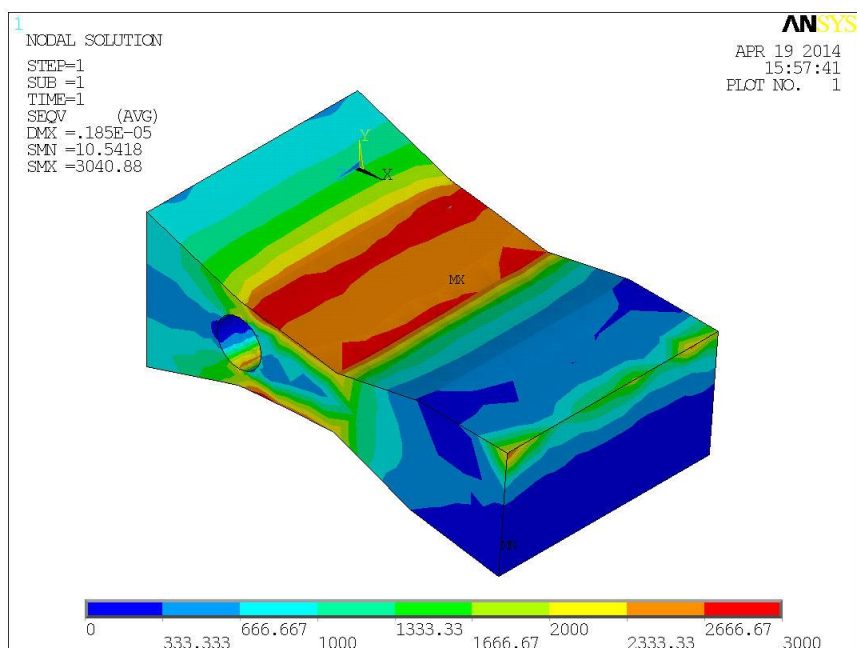


Figura 4.12 Geometria final do modelo proposto por Annicchiarico e Cerrolaza, (2001)

Para todos os estudos de caso relacionados à otimização de forma avaliados anteriormente, a configuração do AG, que melhor se comportou, foi o que apresenta a Tabela 4.9. O objetivo do estudo abaixo é verificar o impacto dessas alterações nos resultados obtidos, apontados na Figura 4.12.

Tabela 4.9 Métodos adotados por ambos os autores

	Almeida, A.B.
População	50
Número de Gerações	50
Esquema de Cruzamento	Cruzamento “Single Point”
Probabilidade de Cruzamento	0,65
Probabilidade de Mutação	0,001
Escalonamento	1,2
Elitismo	Habilitado

O volume encontrado, utilizando as configurações da Tabela 4.9 foi de 7683,37 mm³; tal valor corresponde a aproximadamente (17,5%) a menos do que os valores encontrados por Annicchiarico e Cerrolaza, (2001), a Figura 4.13 mostra o resultado encontrado.

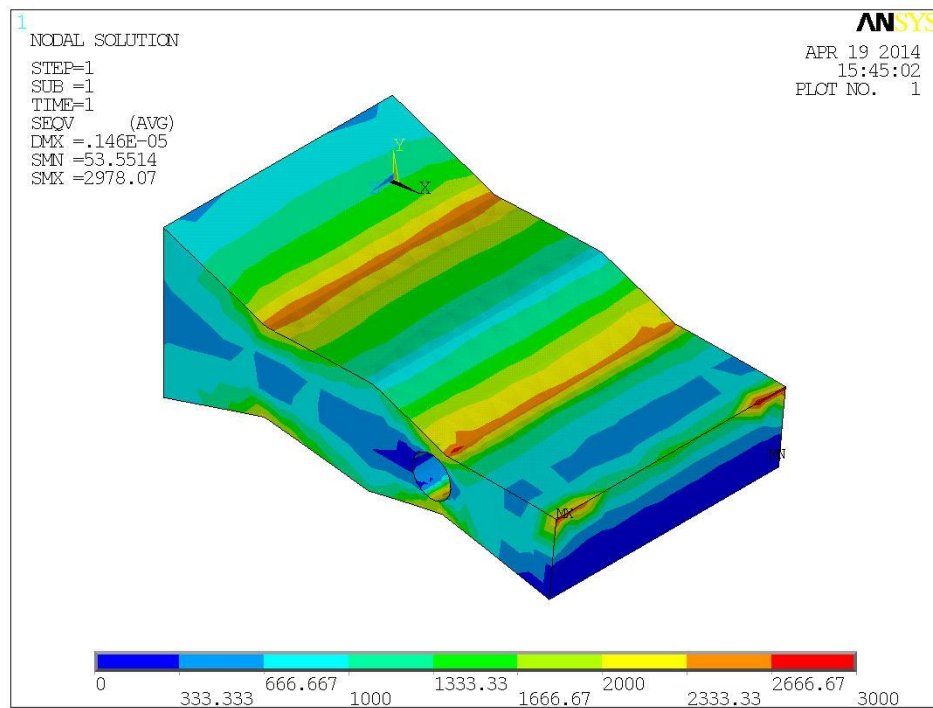


Figura 4.13 Geometria final do modelo proposto por Annicchiarico e Cerrolaza, (2001), utilizando as configurações da Tabela 4.9

5 CONCLUSÕES E SUGESTÕES PARA PRÓXIMOS TRABALHOS

Este trabalho analisou o emprego de Algoritmos Genéticos para a otimização de forma de componentes mecânicos. O trabalho envolveu o estudo das ferramentas usuais e de novas ferramentas para otimização, apresentando as principais e focando naquelas que foram usadas nesta dissertação. Com a finalidade de comparar o desempenho dos Algoritmos Genéticos, os resultados obtidos com este método foram comparados com os obtidos com o emprego de um programa comercial (Ansys®), que utiliza ferramentas tradicionais de otimização (método de aproximação por subproblema). Além disso, o trabalho buscou comparar os seus resultados com os obtidos em literatura. Por fim, mostrou que, utilizando parâmetros otimizados, é possível obter resultados mais adequados com o programa desenvolvido nesta dissertação.

As principais conclusões do trabalho são:

- É possível empregar AG para otimização de forma de componentes mecânicos e seu desempenho é adequado ou superior, quando comparado ao método de aproximação por subproblema, utilizado em programas comerciais.
- O algoritmo, independentemente da geometria ou da amplitude dos limites das variáveis, explorou muito bem o domínio, o que não aconteceu com os métodos tradicionais de otimização estudados neste trabalho.
- Há uma forte influência dos parâmetros do método dos AG no seu desempenho. Foi constatado que, dependendo dos ajustes, o algoritmo pode encontrar dificuldades para convergência.
- Em problemas relacionados à otimização de forma, a configuração que teve melhor desempenho foi $P_m = \text{Probabilidade de Cruzamento } 0,65$; $P_m = \text{Probabilidade de Mutação } 0,001$; *Elitismo habilitado*, *Escalonamento de 1,2* e uma *População de 50*. Conforme apresentado, valores de população superiores a 50 melhoraram os resultados. Porém, valores acima de 70 tiveram pouca influência nos resultados. Tal parâmetro é de extrema importância, pois está intimamente ligado ao tempo computacional.

Como sugestão para os próximos trabalhos têm-se:

- Dividir a *string* em várias posições. A técnica utilizada para o cruzamento foi o *single point*, isto é, a *string* foi dividida somente em uma posição. Atualmente, existem diversos trabalhos em que a *string* é dividida em várias posições.
- Realização de estudos similares, utilizando outros algoritmos. O algoritmo estudado foi um dos precursores das técnicas evolucionárias de otimização. Entretanto, atualmente existem muitas outras como, por exemplo, Evolução Diferencial, Otimização baseada na Biogeografia, Aprendizado baseado na Oposição entre outras.
- No estudo realizado, caso a Tensão (MPa) fosse superior estabelecido em uma determinada iteração, a função era severamente penalizada. A fim de explorar melhor o domínio, os valores de penalização poderiam ser definidos como uma função.
- Empregar geometria, ao invés de posição de pontos. Nos estudos realizados nessa dissertação, a maioria das variáveis que poderiam ser alteradas era relacionada à posição de pontos da geometria. Seria interessante realizar estudos com a geometria propriamente dita já definida (retas, raios, etc.) e variar os valores desses parâmetros geométricos, não somente da posição. Isso se aproximaria mais o estudo dos problemas normalmente encontrados em engenharia.

REFERÊNCIAS

- AHARI et al. **A genetic algorithm for optimization of laminated dies manufacturing**, Computer-Aided Design, v.43, p.730-737, 2011.
- AMADORI, K. et al. **Flexible and robust CAD models for design automation**, Advanced Engineering Informatics, v.26, p.180-195, 2012.
- AN et al. Optimization for Cylindrical Cup Drawing Based on Multi-Objective Genetic Algorithm. In:_____. **Software Engineering and Knowledge Engineering: Theory and Practice**, Springer, 2012. v.1, p.617-624.
- ANDRESEN et al. **Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop**, Mathematical and computer modeling, v.48, 1279-1293, 2008.
- ANTONIOU, A. e Lu, W.S **Practical Optimization**, 1º Edição, Springer, 2007, 669p.
- ANSYS User's Manual, **Procedures**, v.1, Swanson Analysis Systems Inc., 2009.
- ANSYS User's Manual, **Theory**, v.4, Swanson Analysis Systems Inc., 2009.
- BASKHARONE,E.A. **The Finite Element Method with Heat Transfer and Fluid Mechanics Applications**, Cambridge University Press, 2014, 373p.
- BELYTSHKO, T.e FISH,J. **Um Primeiro Curso em Elementos Finitos**, LTC, 2009, 241p.
- BÉNABÈS, J. et al. An Interactive-Based Approach to the Layout Design Optimization. In:_____. **Global Product Development**, Springer, 2011, p.511-520.
- BENDØE, M.P. e SIGMUND, O. N **Topology Optimization**, Springer, 2003, 370p.
- BHUVARAGHAN et al. **Optimization of the fatigue strength of materials due to shot peening: A Survey**, International Journal of Structural Changes in Solids – Mechanics and Applications, v.02, n.02, p.33-63, 2010.
- BISHOP, J.A. e STRIZ, A.G. **On using genetic algorithms for optimum damper placement in space trusses**, Struct Multidisc Optim, v.28, p.136-145, 2004.
- BROTCHIE, J. F. **Optimization and robustness of structural engineering systems**, Engineering Structures, v.19, n.4, p.289-292, 1997.

BUREERAT, S. e LIMTRAGOOL, J. **Structural topology optimization using simulated annealing with multiresolution design variables**, Finite Elements in Analysis and Design, v.44, p.738-747, 2008.

CHENG, J. **Optimum design of steel truss arch bridges using a hybrid genetic algorithm**, Journal of Constructional Steel Research, v.66, p.1011-1017, 2010.

CHIANDUSSI et al. **Comparison of multi-objective optimization methodologies for engineering applications**, Computers and Mathematics with Applications, v.63, p.912-942, 2012.

COOK, R.D. **Concepts and Applications of Finite Elements Analysis**, 4ª Edição, John Wiley & Sons Inc., 2002, 719p.

DAS et al. **Design of structures for optimal static strength Using ESO**, Engineering Failure Analysis, v.12, p.61–80, 2005.

DEGERTEKIN et al. **Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm**, Engineering Structures, v.30, p.197-205, 2008.

DENG et al. **Topology optimization of steady Navier–Stokes flow with body force**, Comput. Methods Appl. Mech. Engrg., v.255, p.306-321, 2013.

GEN et al. **Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey**, Computers & Industrial Engineering, v.56, p.779-808, 2009.

GHIANI et al. **Operations research in solid waste management: A survey of strategic and tactical issues**, Computers & Operations Research, v.44, p.22-32, 2014.

GOLUGLU, C e ZEYVELI, M. **A genetic approach to automate preliminar design of gear drives**, Computers & Industrial Engineering, v.57, p.1043-1051, 2009.

HAFTKA, R. T e GÜRDAL, Z. **Elements of Structural Optimization**, Kluwer Academic Publishers, 1991, 481p.

HAMADE, R.F. et al. **Toward predicting the performance of novice CAD users based on their profiled technical attributes**, Engineering Applications of Artificial Intelligence, v.25, p.628-639, 2012.

HASAN et al. **A comparison between simulated annealing, genetic algorithm and tabu search methods for the unconstrained quadratic Pseudo-Boolean function**, Computers & Industrial Engineering, v.38, p.323-340, 2000.

HAUPT, R. L. e HAUPT, S.E. **Practical genetic algorithms**, 2º Edição, John Wiley & Sons Inc., 2004, 263p.

HO, Z.P. e WU, C.S. **Application of Simulated Annealing Algorithm to Optimization Deployment of Mobile Wireless Base Stations**, Advances in CSIE, v.2, p.665-670, 2012.

HSIAO, S.W. et al. **Product-form design model based on genetic algorithm**, International Journal of Industrial Ergonomics, v.40, p.237-246, 2010.

IQBAL, K. **Fundamental Engineering Optimization Methods**, Kamran Iqbal &, 2013, 162p.

KAMRAN, A. e GUOZHU, L. **An integrated approach for optimization of solid rocket motor**, Aerospace Science and Technology, Aerospace Science and Technology, v.17, p.50-64, 2012.

KHOO, L.P. e ONG, N.S. **PCB Assembly Planning Using Genetic Algorithms**, The International Journal of Advanced Manufacturing Technology, v.14, p.363-368, 1998.

KIM, N.H. e BHAVANI, V.S. **Introduction to Finite Element Analysis and Design**, John Wiley & Sons Inc., 2009, 419p.

LAMBERTI, L. **An efficient simulated annealing algorithm for design optimization of truss structures**, Computers and Structures, v.86, p.1936-1953, 2007.

LEON, N. **The future of computer-aided innovation**, Computers in Industry, v.60, p.539-550, 2009.

LEUNG et al. **A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem**, Computers & Operations Research, v.39, p.64-73, 2012.

LI et al. **A parallel simulated annealing algorithm based on functional feature tree modeling for 3D engineering layout design**, Applied Soft Computing, v.10, p. 592–601, 2010.

LI, H. e MULAY, S.S. **Meshless Methods and their Numerical Properties**, CRC Press Taylor & Francis Group, 2013, 408p.

LIN et al. **Using Tabu Search to Solve MINLP Problems for PSE**, Process Systems Engineering, v.15, p.541-546, 2003.

LIN, C.Y. e SHEU, F.M. **Adaptive volume constraint algorithm for stress limit-based topology optimization**, Computer-Aided Design, v.41, p.685-694, 2009.

LOGAN, D.L. **A First Course in the Finite Element Method**, 4º Edição, Thomson Canada Limited, 2007, 836p.

LV et al. **Cloud theory-based simulated annealing algorithm and application**, Engineering Applications of Artificial Intelligence, v.22, p.742-749, 2009.

MADENCI, E. e GUVEN, I. **The Finite Element Method and Applications in Engineering Using Ansys®**, 1ª Edição, Springer, 2006, 686p.

NAKAZONE et al. **Engineering Analysis with ANSYS Software**, 1ª Edição, Elsevier Butterworth-Heinemann, 2006, 456p.

NETO et al. **Computational Methods for Plasticity Theory and Applications**, John Wiley & Sons Inc., 2008, 791p.

NIMTAWAT, A. e NANAKORN, P. **Automated layout design of beam-slab floors using a genetic algorithm**, Computers and Structures, v.87, p.1308-1330, 2009.

NORRIS, J.R. **Markov Chains**, 15ª Impressão, Cambridge University Press, 2009, 237p.

OKUZONO et al. **A finite-element method using dispersion reduced spline elements for room acoustics simulation**, Applied Acoustics, v.79,p.1-8,2014.

PARK, H.S. e DANG, X.P. **Structural optimization based on CAD-CAE integration and metamodeling techniques**, Computer-Aided Design, v.42, p.889-902, 2010.

PARSONS, R. e CANFIELD, S.L. **Developing genetic programming techniques for the design of compliant mechanisms**, Struct Multidisc Optim, v.24, p.78-86, 2002.

RAO, S. S. **Engineering optimization : Theory and practice**, 4ª Edição, John Wiley & Sons Inc., 2009, 813p.

SEO, Y. D et al. **Isogeometric topology optimization using trimmed spline surfaces**, Computer Methods in Applied Mechanics and Engineering, v.199, p.3270-3296, 2010.

SHI, X. **Design optimization of insulation usage and space conditioning load using energy simulation and genetic algorithm**, Energy, v.36, p.1659-1667, 2011.

SIMON, D. **Evolutionary Optimization Algorithms**, John Wiley & Sons Inc., 2013, 742p.

SONMEZ, F.O. **Shape optimization of 2D structures using simulated annealing**, Computer methods in applied mechanics and engineering, v.196, p.3279-3299, 2007.

TURRIN, M. et al. **Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms**, Advanced Engineering Informatics, v.25, p.656-675, 2011.

VASAN, A. e RAJU, K.S. **Comparative analysis of Simulated Annealing, Simulated Quenching and Genetic Algorithms for optimal reservoir operation**, Applied Soft Computing, v.9, p.274-281, 2009.

VASANT, P. e BARSOUM, N. **Hybrid pattern search simulated annealing for fuzzy production planning problems**, Computers and Mathematics with Applications, v.60, p.1058-1067, 2010.

WANG et al. **A virtual punching method for shape optimization of openings on curved panels using CAD-based Boolean operations**, Computer-Aided Design, v.44, p.388-399, 2012.

WEI et al. **Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm**, Struct. Multidisc. Optim, v.43, p.665-682, 2011.

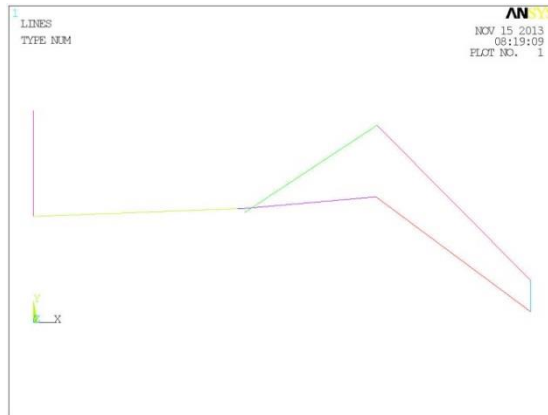
WIDMAIER, K. **Algoritmo Genético aplicado à Otimização de Asas de Material Compósito de Veículos Aéreos não Tripulados**, 2005. 182p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

YENISEY, M.M. e YAGMAHAN, B. **Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends**, Omega, v.45,p.119-135,2014.

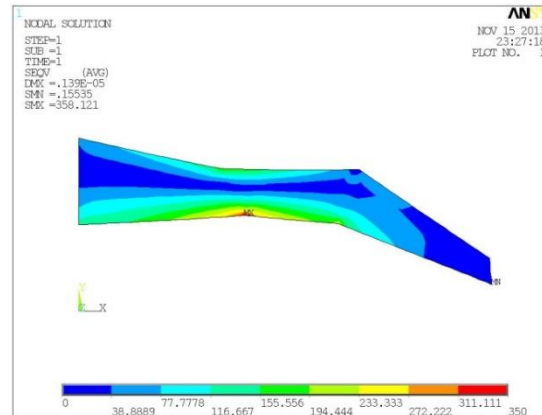
ZIENKIEWICZ,O.C. **The Finite Element Method**, 6º Edição, Elsevier Butterworth-Heinemann, 2005,792p.

APÊNDICE A – Influência das alterações dos parâmetros do AG

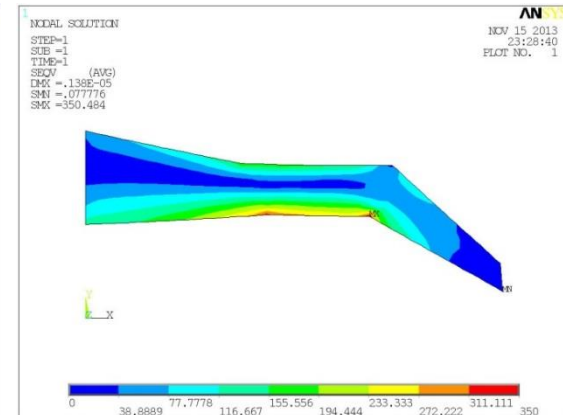
PC 0.65 PM 0.001



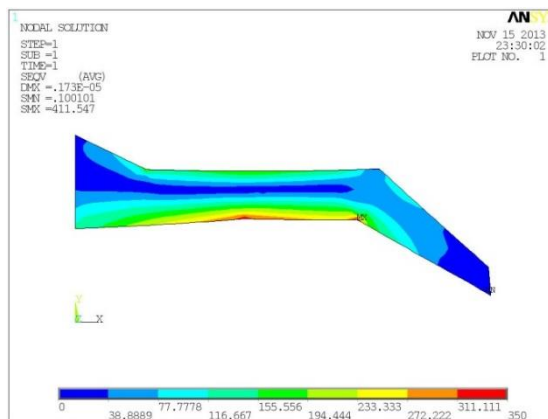
Iteração = 01
NÃO CONVERGIU



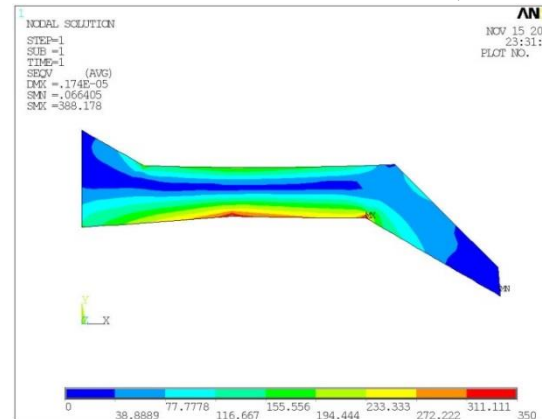
Iteração = 25
Tensão = 358 MPa / Área = 8121,35 mm²



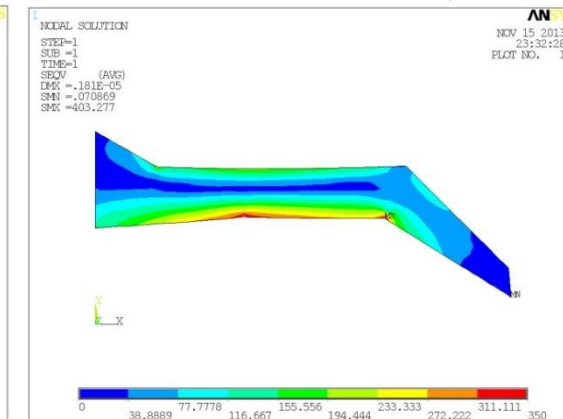
Iteração = 50
Tensão=350 MPa /Área=6478,22 mm²



Iteração = 100
Tensão = 411 MPa / Área = 6363,43mm²

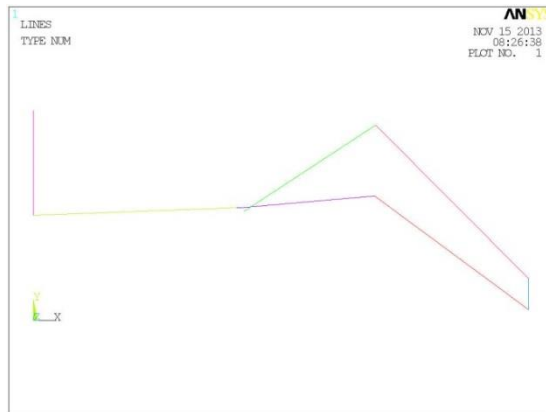


Iteração = 150
Tensão = 388 MPa / Área = 5773,01 mm²

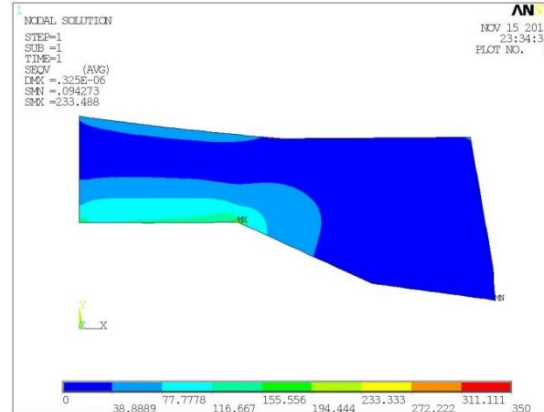


Iteração = 200
Tensão= 403 MPa/Área= 5725,12mm²

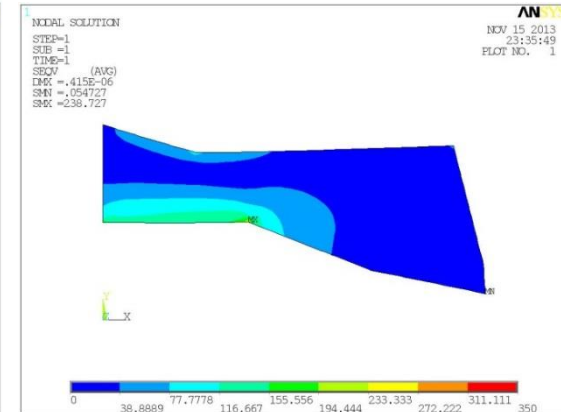
PC 0.65 PM 0.001 (SEM ELITISMO)



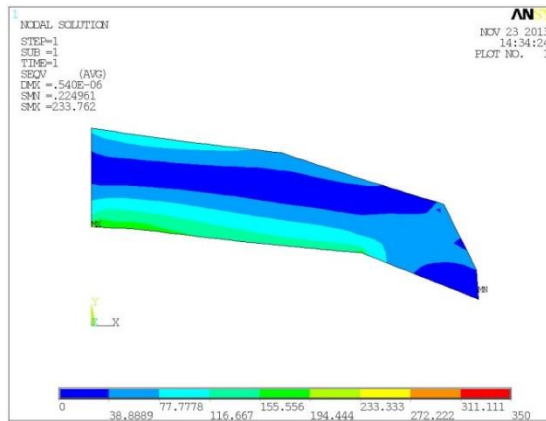
Iteração = 01
NÃO CONVERGIU



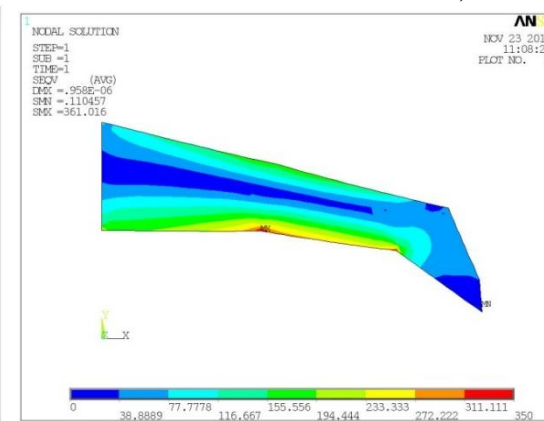
Iteração = 25
Tensão = 233 MPa / Área = 8288,30 mm²



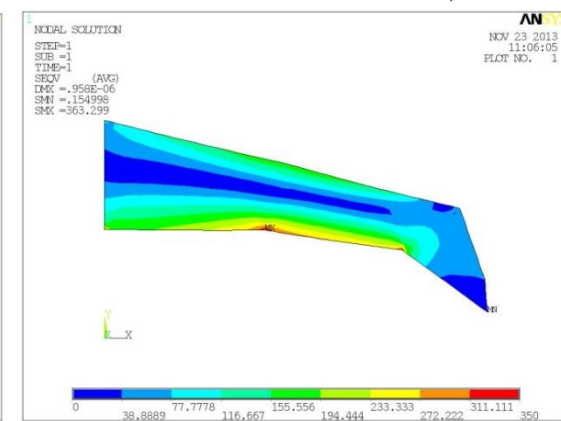
Iteração = 50
Tensão=238 MPa /Área=7852,23 mm²



Iteração = 100
Tensão = 233 MPa / Área = 6261,01 mm²

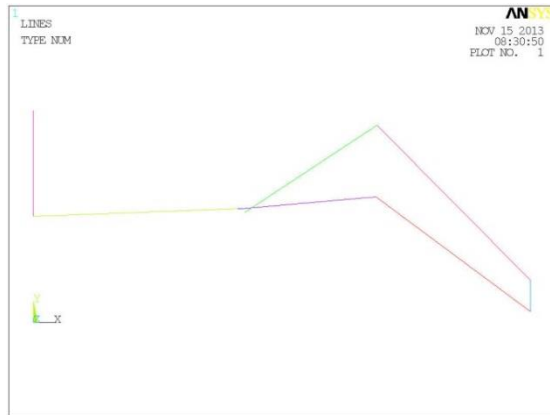


Iteração = 150
Tensão = 361 MPa / Área = 5677,98 mm²

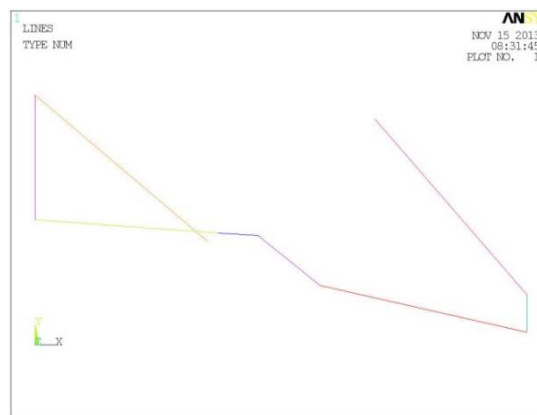


Iteração = 200
Tensão= 363 MPa/Área= 5590,33 mm²

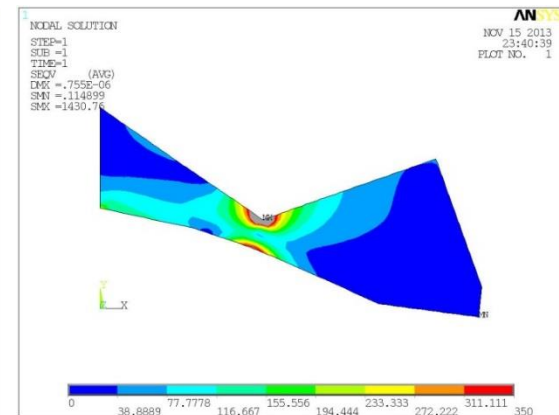
PC 0.65 PM 0.10



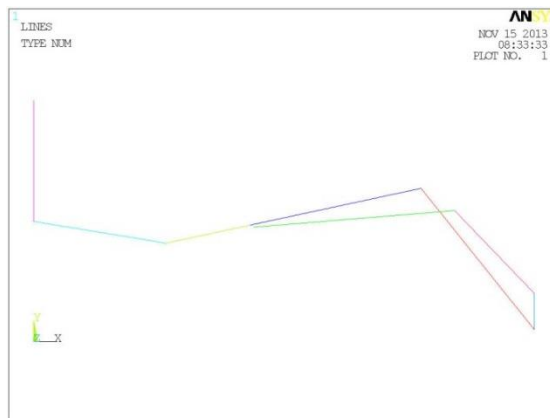
Iteração = 01
NÃO CONVERGIU



Iteração = 25
NÃO CONVERGIU



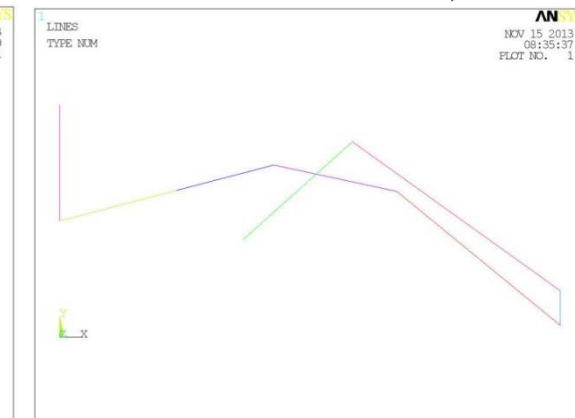
Iteração = 50
Tensão=1430 MPa /Área=17148,41 mm²*



Iteração = 100
NÃO CONVERGIU

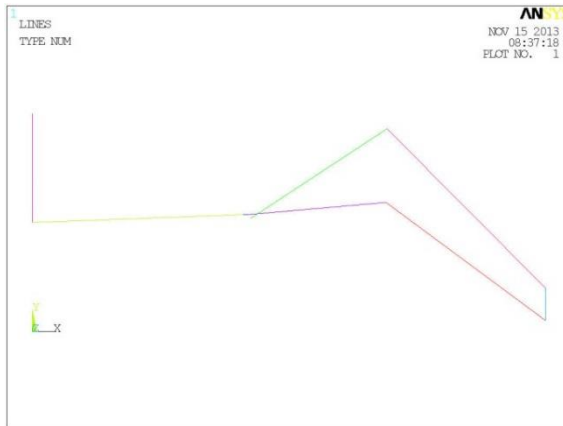


Iteração = 150
NÃO CONVERGIU

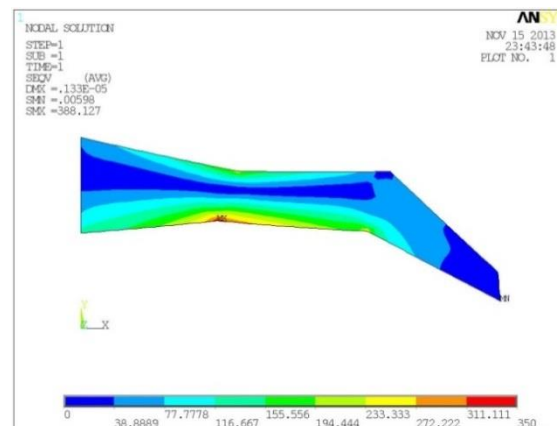


Iteração = 200
NÃO CONVERGIU

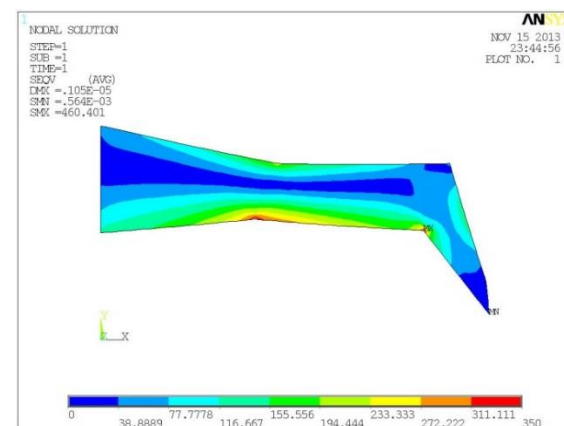
PC 0.65 PM 0.001 e ESCALONAMENTO 2.0



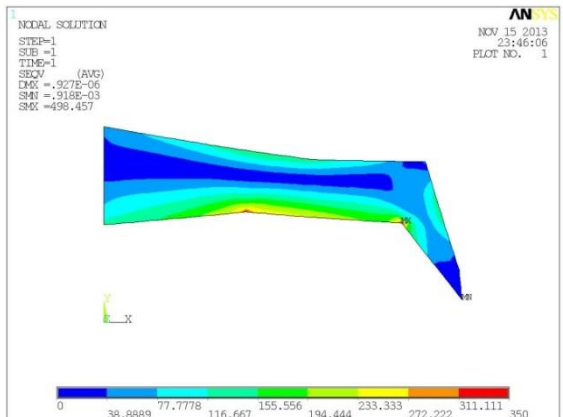
Iteração = 01
NÃO CONVERGIU



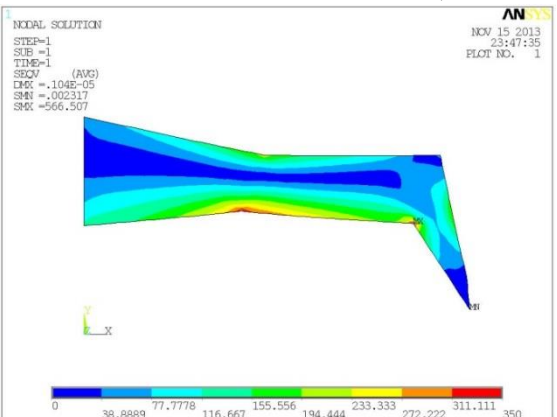
Iteração = 25
Tensão = 388 MPa / Área = 6996,22 mm²



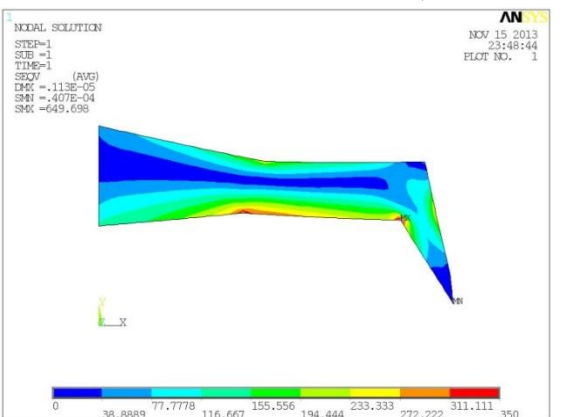
Iteração = 50
Tensão = 460 MPa / Área = 6225,38 mm²



Iteração = 100
Tensão = 498 MPa / Área = 6259,21 mm²

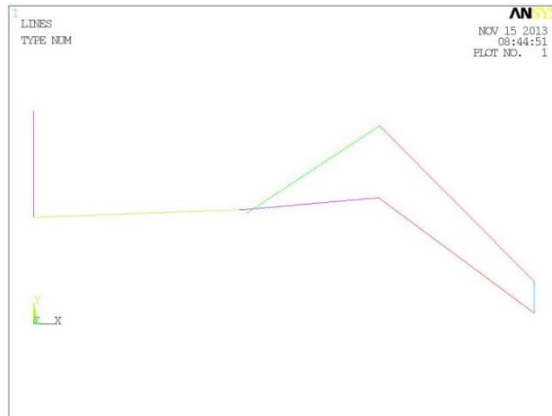


Iteração = 150
Tensão = 566 MPa / Área = 6623,25 mm²

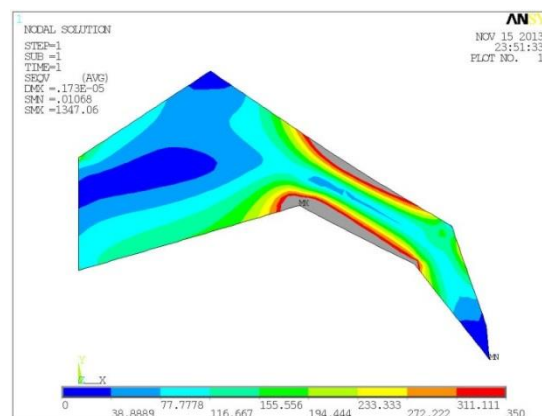


Iteração = 200
Tensão = 649 MPa / Área = 5655,05 mm²

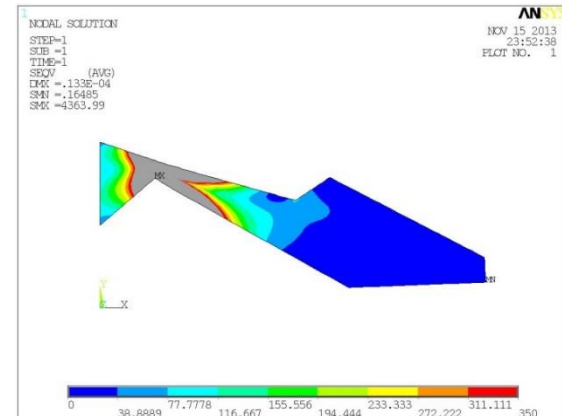
PC 0.65 PM 0.30 e ESCALONAMENTO 1.2



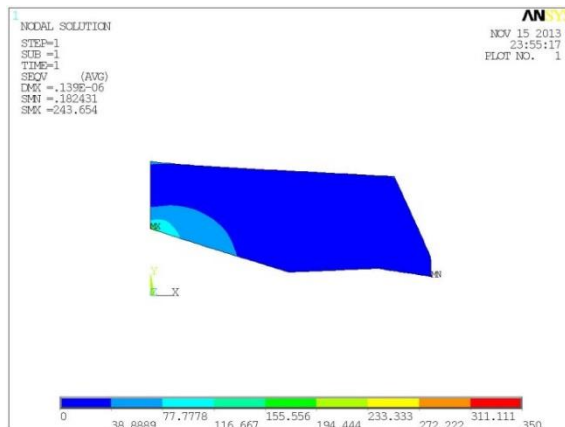
Iteração = 01
NÃO CONVERGIU



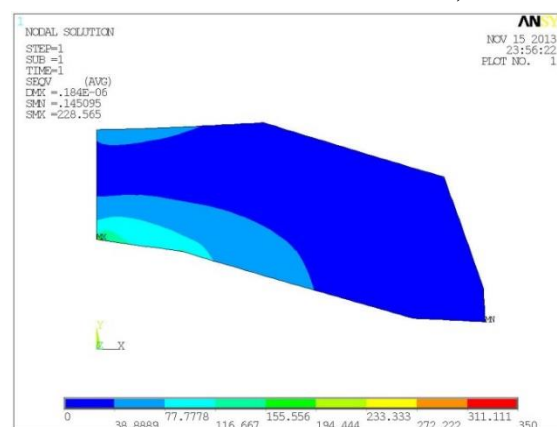
Iteração = 25
Tensão = 1347 MPa/Área = 16699,03mm²*



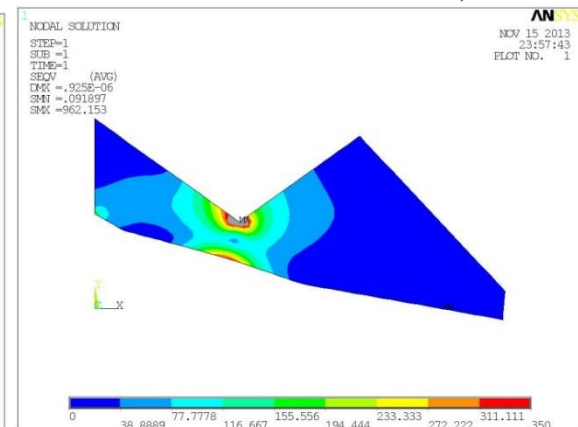
Iteração = 50
Tensão=4363 MPa/Área=16449,34 mm²*



Iteração = 100
Tensão = 243 MPa / Área = 13174,47 mm²

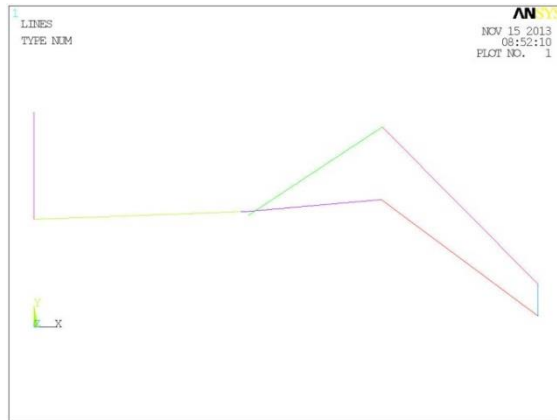


Iteração = 150
Tensão = 228 MPa / Área = 10749,53 mm²

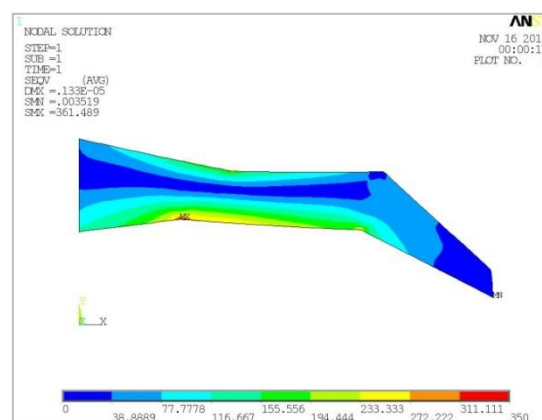


Iteração = 200
Tensão= 962 MPa/Área=19906,15 mm²*

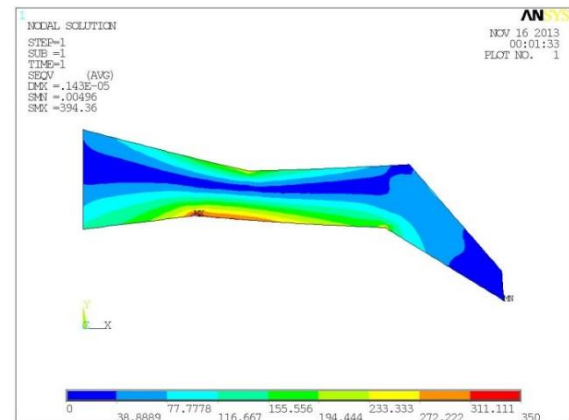
PC 0.90 PM 0.001 e ESCALONAMENTO 1.2



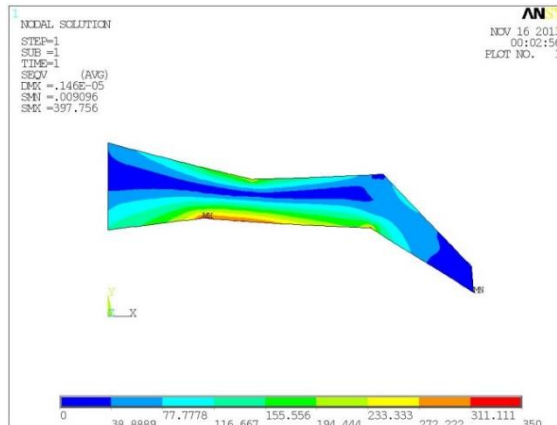
Iteração = 01
NÃO CONVERGIU



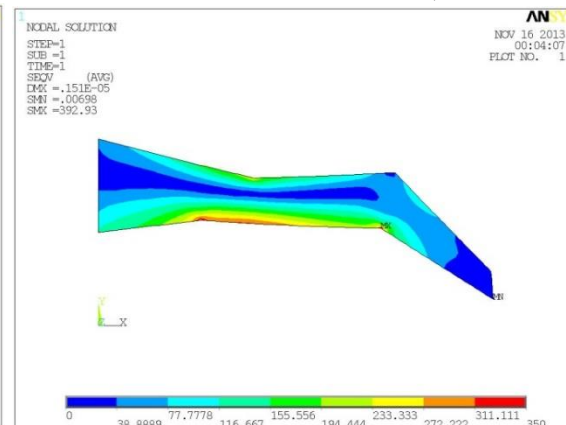
Iteração = 25
Tensão = 361 MPa/Área = 6972,10 mm²



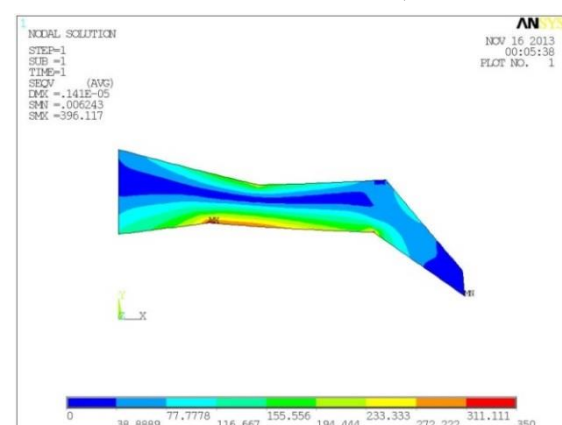
Iteração = 50
Tensão=394 MPa/Área=6479,99 mm²



Iteração = 100
Tensão = 397 MPa / Área = 6298,32 mm²

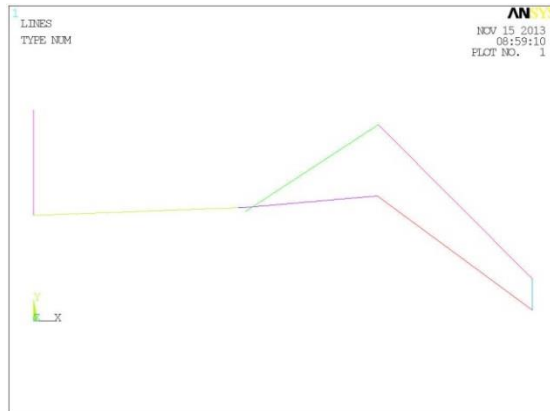


Iteração = 150
Tensão = 392 MPa / Área = 6231,30 mm²

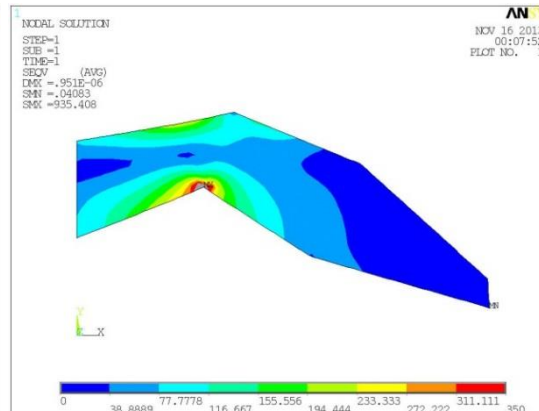


Iteração = 200
Tensão= 396 MPa/Área= 6168,14 mm²

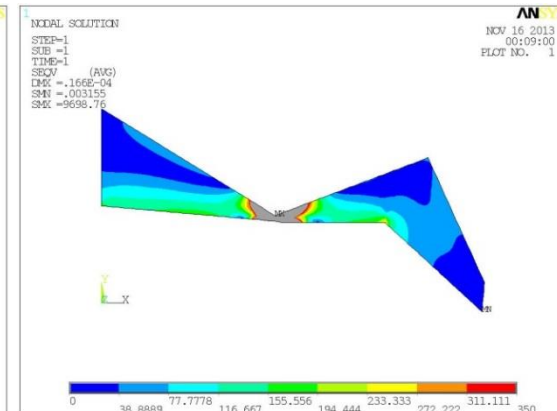
PC 0.90 PM 0.10 e ESCALONAMENTO 1.2



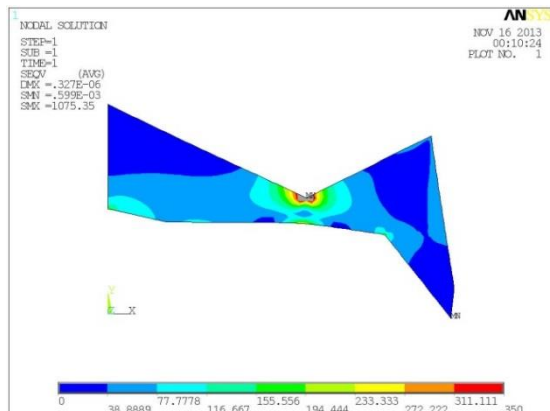
Iteração = 01
NÃO CONVERGIU



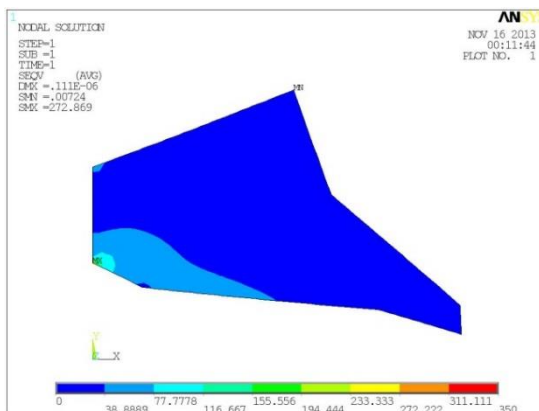
Iteração = 25
Tensão = 935 MPa/Área = 19539,89 mm²*



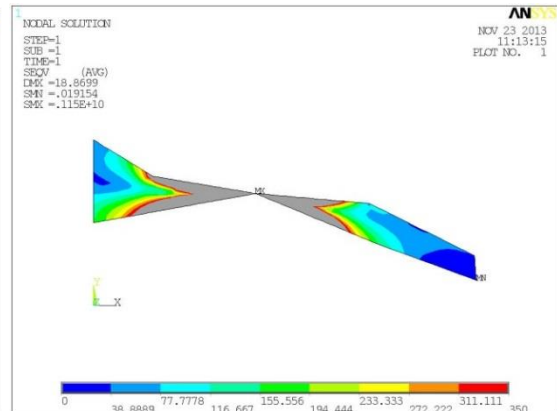
Iteração = 50
Tensão=9698 MPa/Área=16428.64 mm²*



Iteração = 100
Tensão=1075 MPa/Área = 19899,68 mm²*

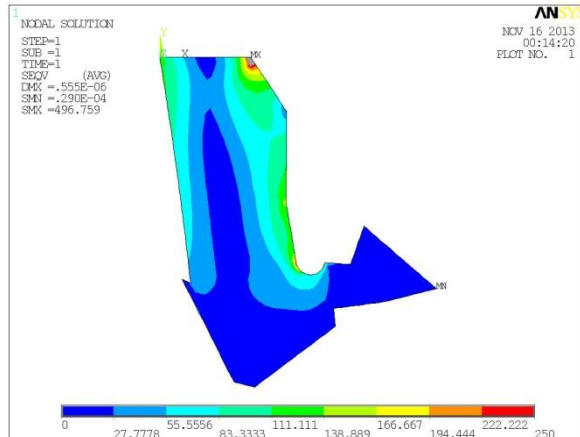


Iteração = 150
Tensão=272 MPa/Área = 12629,31 mm²

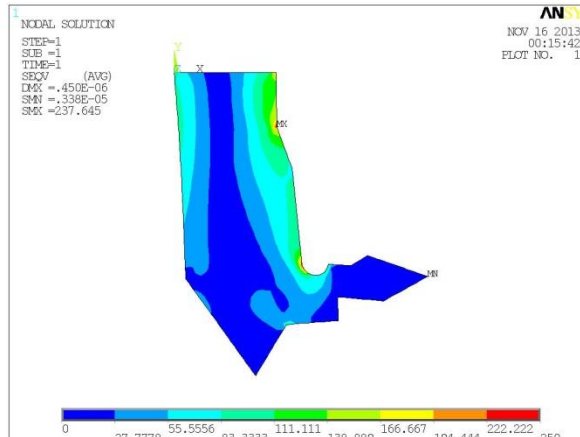


Iteração = 200
Tensão= 11500 MPa/Área=14763,31mm²*

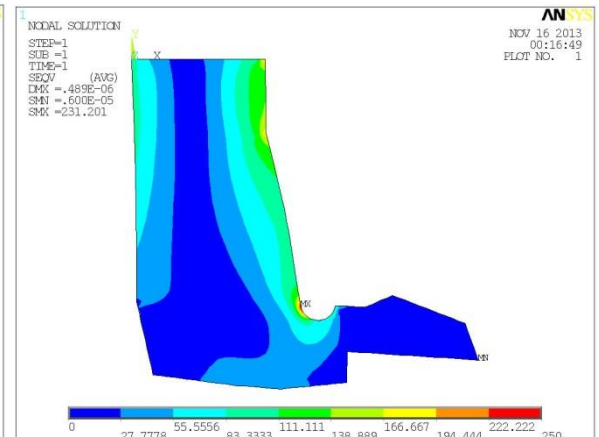
PC 0.65 PM 0.001 e ESCALONAMENTO 1.2



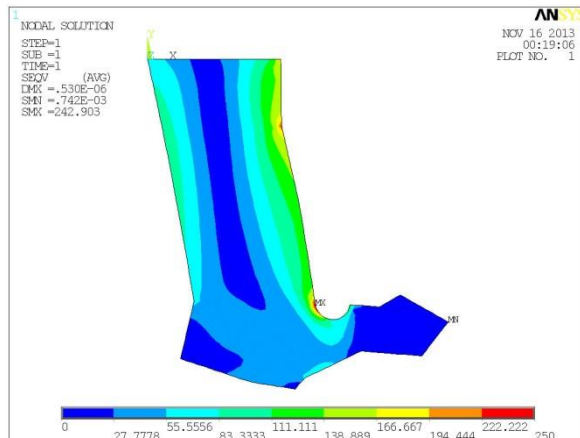
Iteração = 01
 Tensão = 496 MPa / Área = 36357,98 mm²



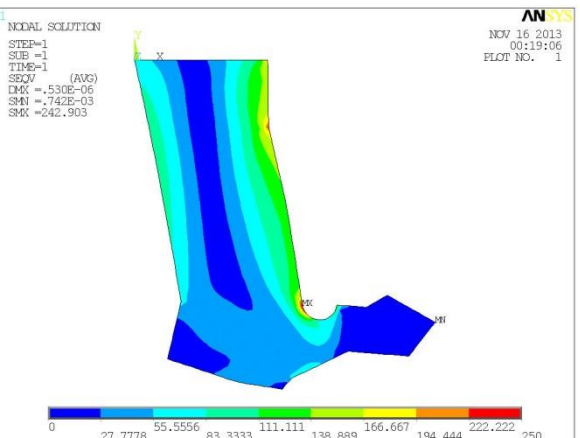
Iteração = 25
 Tensão = 237 MPa / Área = 29030,63 mm²



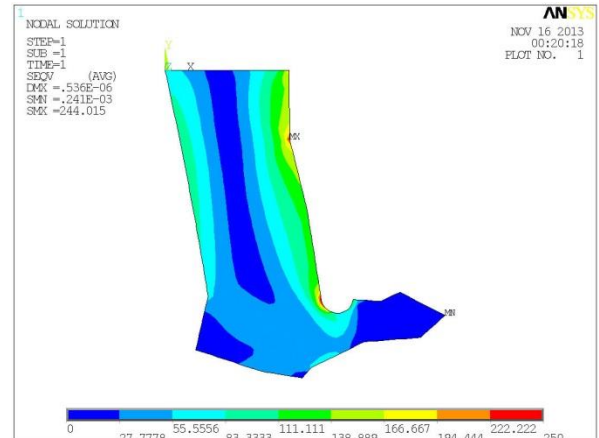
Iteração = 50
 Tensão = 231 MPa / Área = 28998,55 mm²



Iteração = 100
 Tensão = 241 MPa / Área = 23521,64 mm²

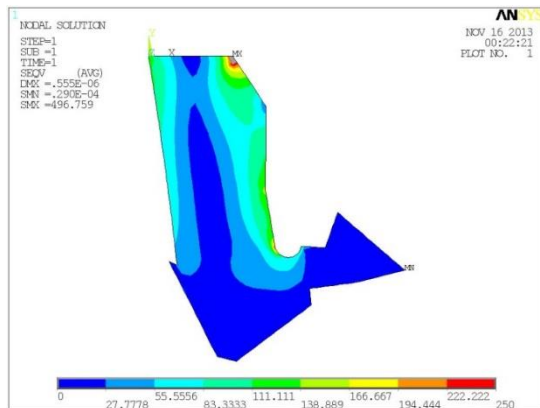


Iteração = 150
 Tensão = 242 MPa / Área = 23424,13 mm²



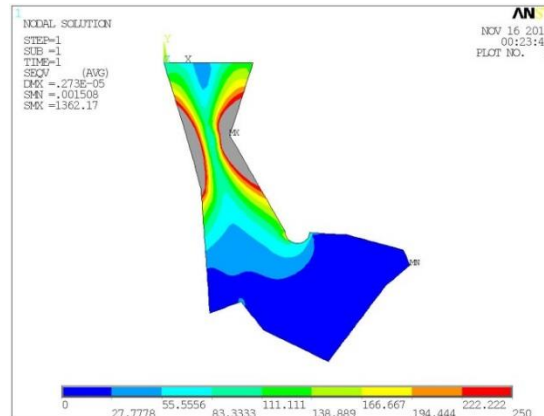
Iteração = 200
 Tensão = 244 MPa / Área = 23113,13 mm²

PC 0.65 PM 0.001 e ESCALONAMENTO 1.2 (SEM ELITISMO)



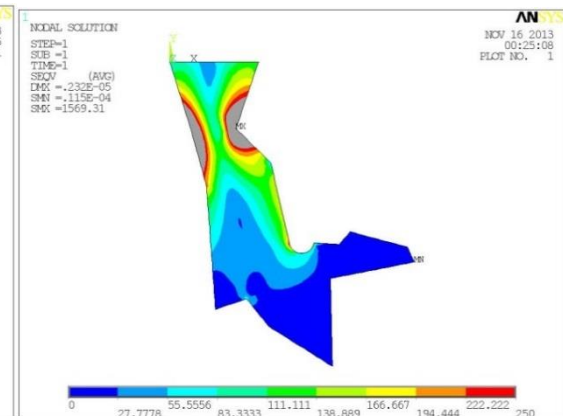
Iteração = 01

Tensão = 496 MPa / Área = 80297,09 mm²*



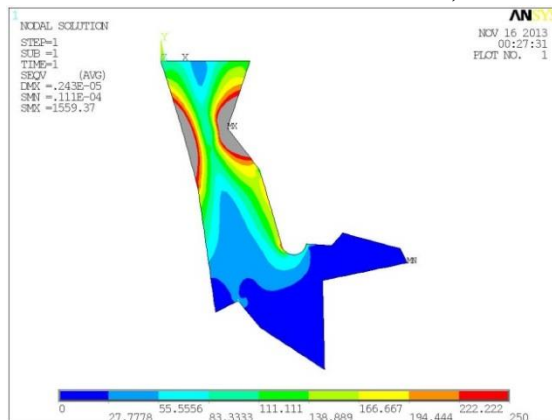
Iteração = 25

Tensão = 1362 MPa / Área = 77349,10 mm²*



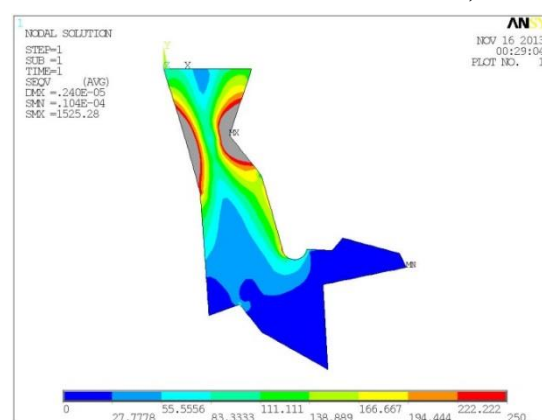
Iteração = 50

Tensão=1569 MPa /Área=75142,55mm²*



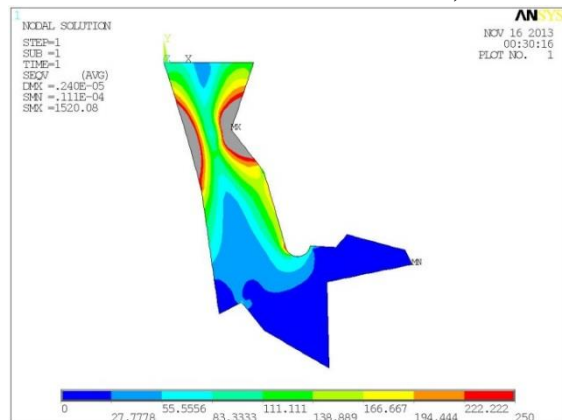
Iteração = 100

Tensão = 1559 MPa/Área = 73815,78 mm²*



Iteração = 150

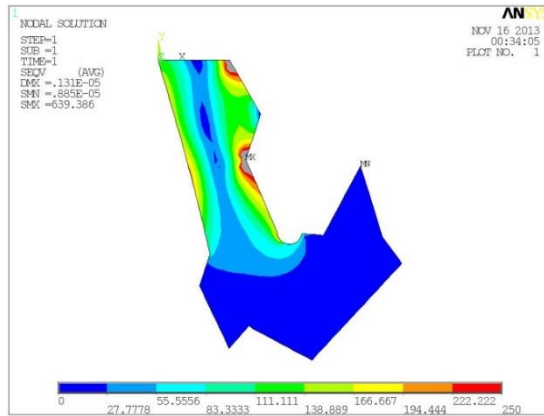
Tensão = 1525 MPa/Área = 74316,27 mm²*



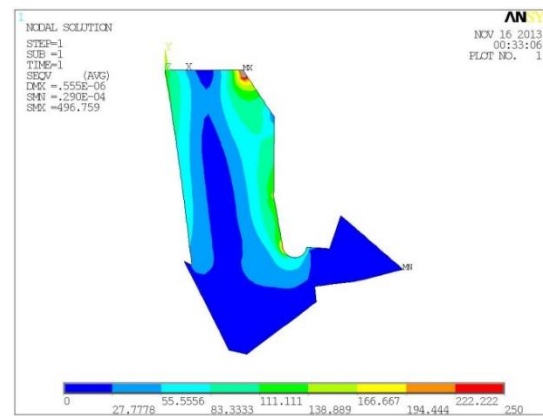
Iteração = 200

Tensão=1520 MPa /Área=73770,36 mm²*

PC 0.65 PM 0.10 e ESCALONAMENTO 1.2



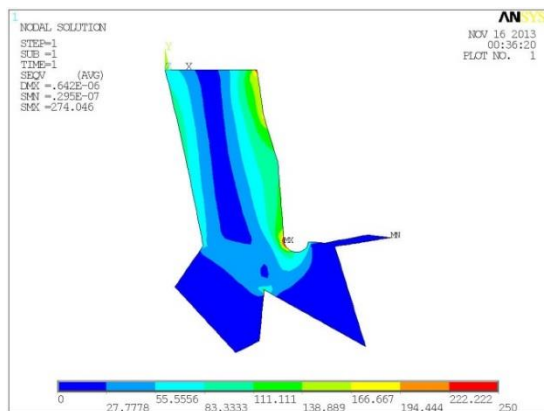
Iteração = 01
Tensão = 496 MPa / Área = 80297,09 mm² *



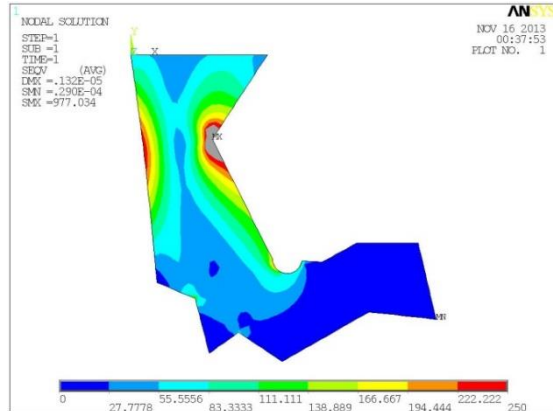
Iteração = 25
Tensão = 639 MPa / Área = 82327,68 mm² *



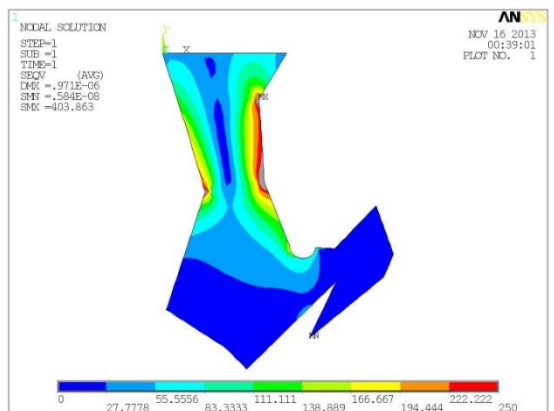
Iteração = 50
NÃO CONVERGIU



Iteração = 100
Tensão = 274 MPa/Área = 36858,43 mm²

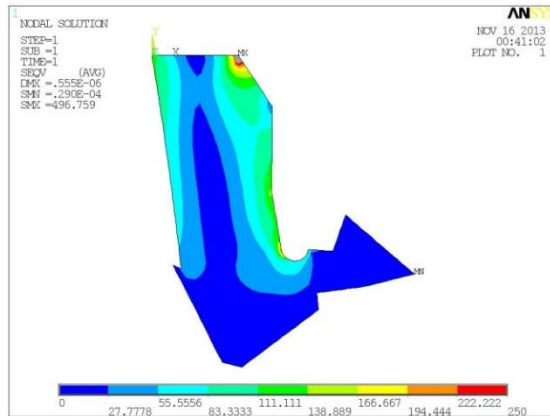


Iteração = 150
Tensão = 977 MPa/Área = 78736,10 mm² *

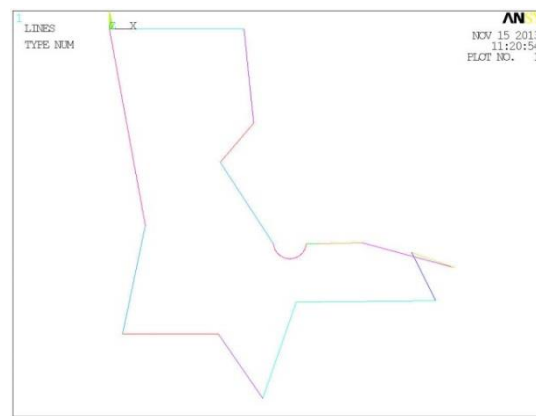


Iteração = 200
Tensão=403 MPa /Área=79285,62 mm² *

PC 0.65 PM 0.001 e ESCALONAMENTO 2.0



Iteração = 01
Tensão = 496 MPa / Área = 80297,09 mm² *



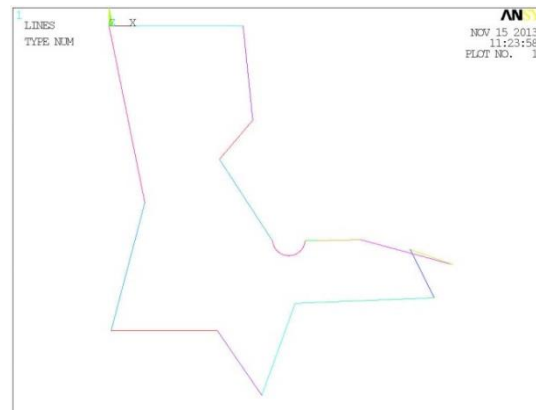
Iteração = 25
NÃO CONVERGIU



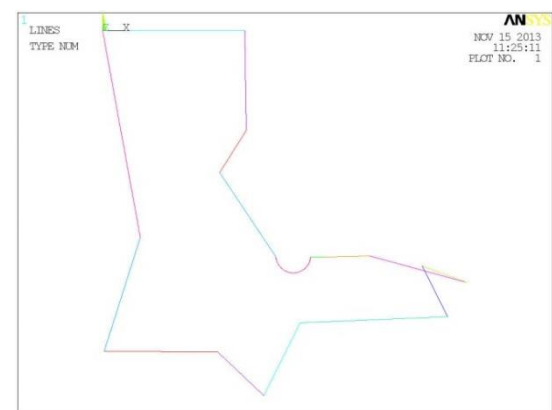
Iteração = 50
NÃO CONVERGIU



Iteração = 100
NÃO CONVERGIU

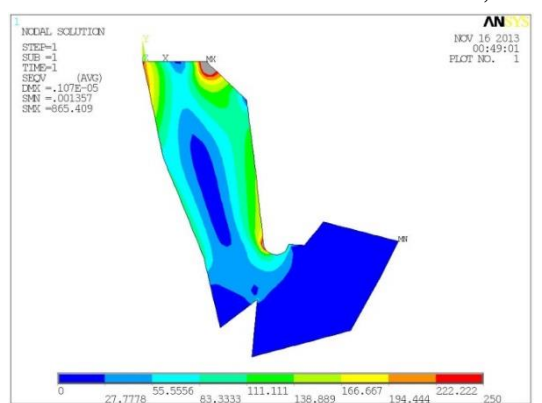
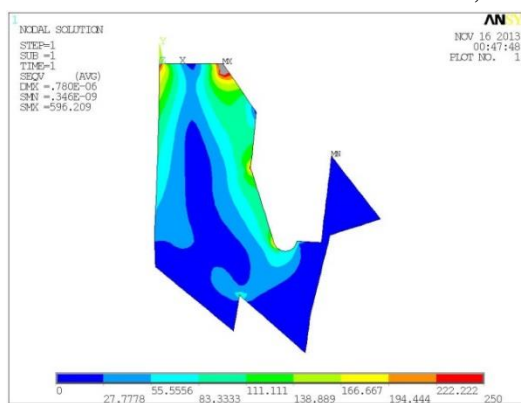
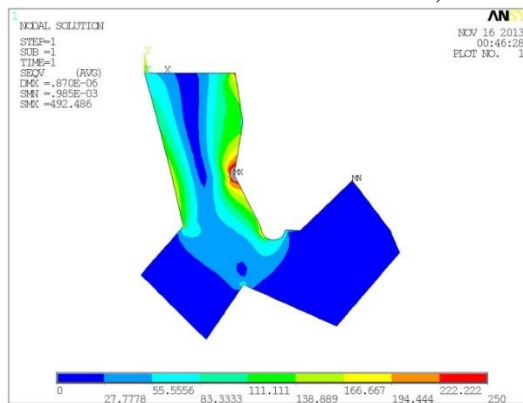
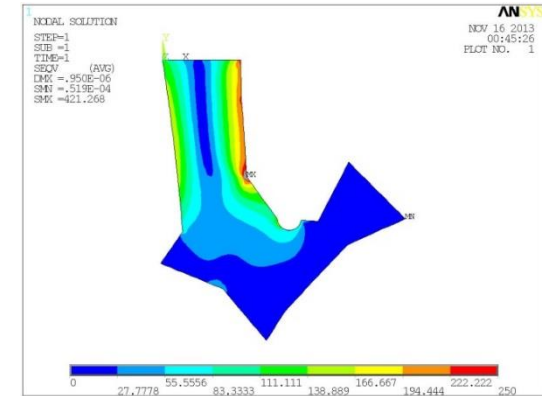
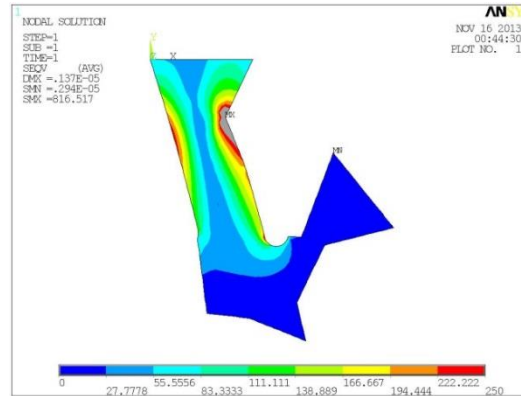
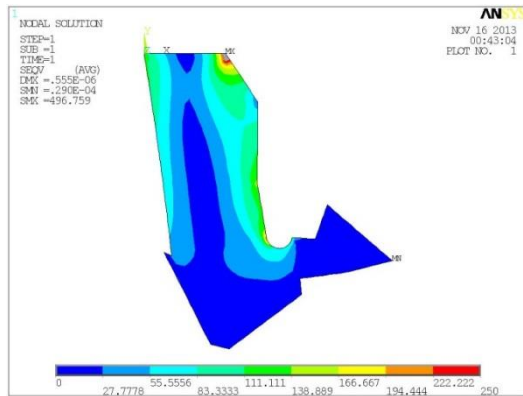


Iteração = 150
NÃO CONVERGIU

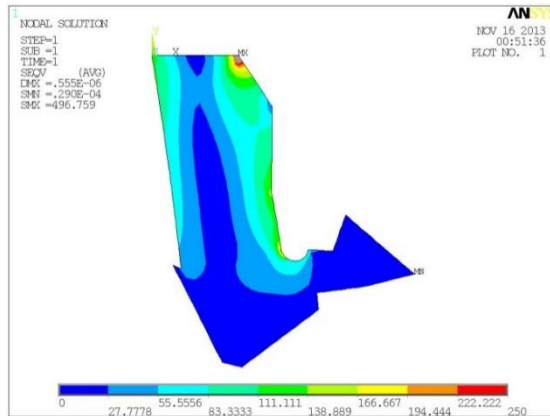


Iteração = 200
NÃO CONVERGIU

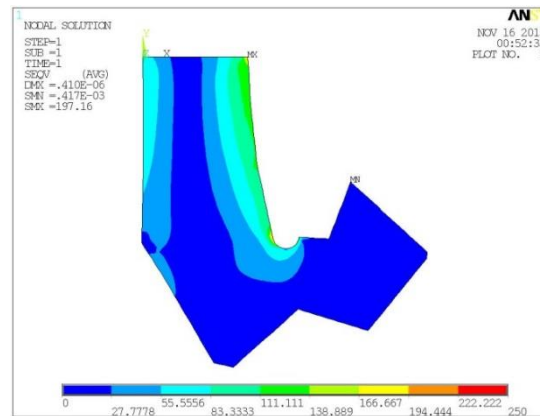
PC 0.65 PM 0.30 e ESCALONAMENTO 1.2



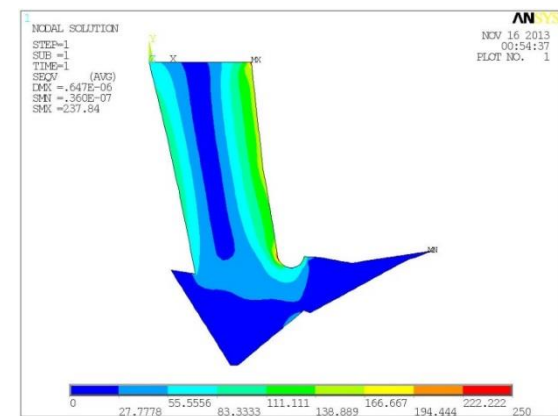
PC 0.90 PM 0.001 e ESCALONAMENTO 1.2



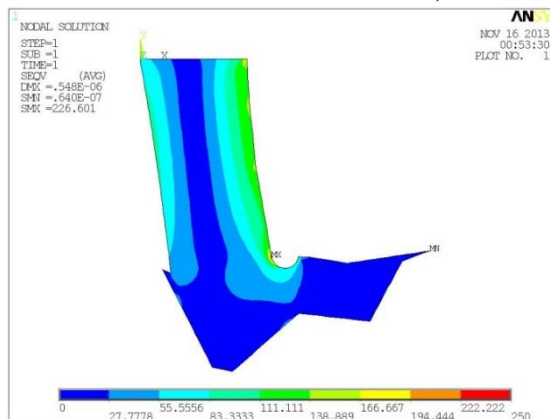
Iteração = 01
Tensão = 496 MPa/Área = 80297,09 mm² *



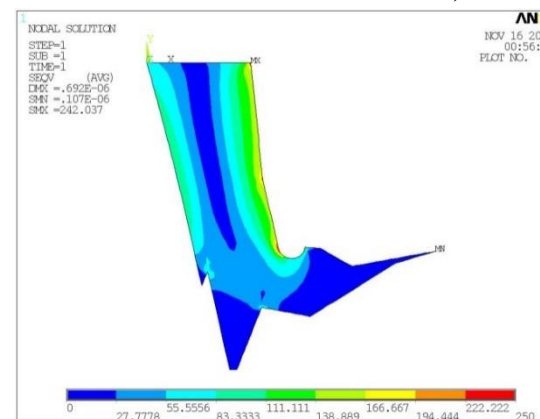
Iteração = 25
Tensão = 197 MPa/Área = 38928,29 mm²



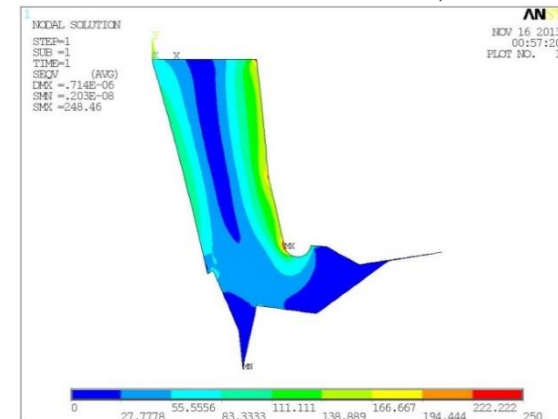
Iteração = 50
Tensão=237 MPa /Área=30150,41 mm²



Iteração = 100
Tensão = 226 MPa/Área = 26247,81 mm²

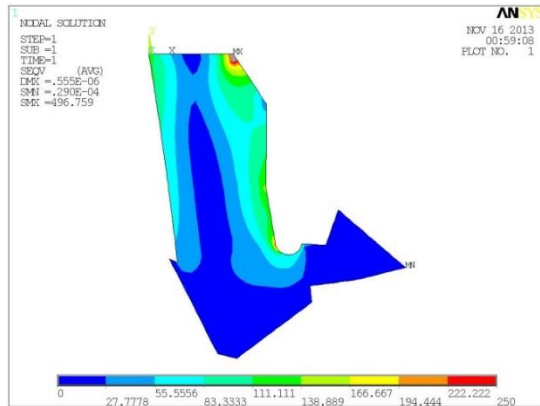


Iteração = 150
Tensão = 242 MPa/Área = 23377,57 mm²



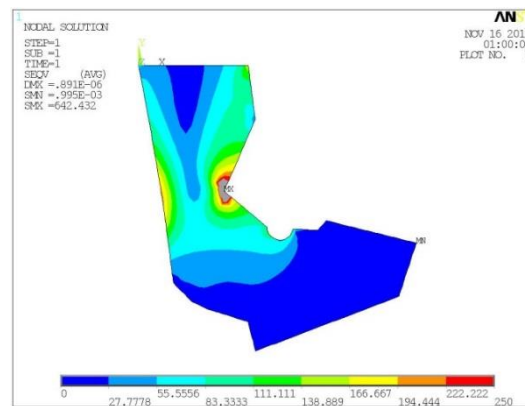
Iteração = 200
Tensão=248 MPa /Área=21648,71 mm²

PC 0.90 PM 0.10 e ESCALONAMENTO 1.2



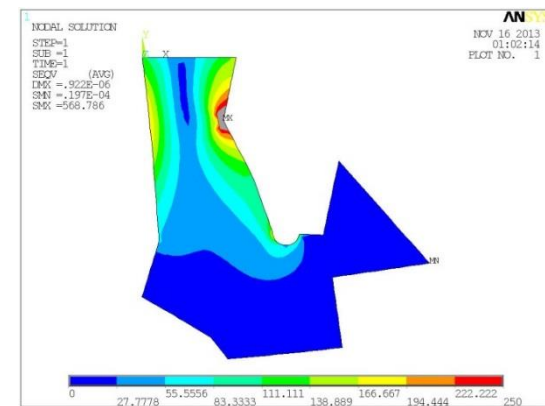
Iteração = 01

Tensão = 496 MPa/Área = 80297,09 mm² *



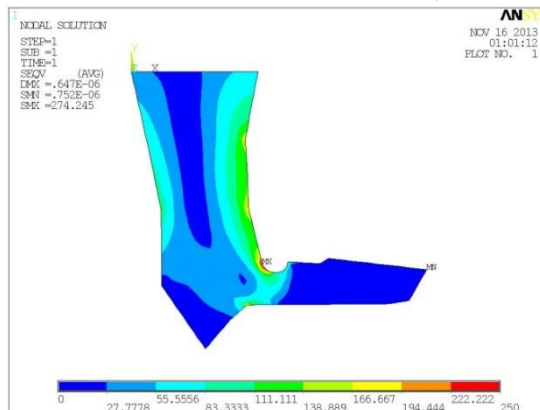
Iteração = 25

Tensão = 642 MPa/Área = 82520.89 mm² *



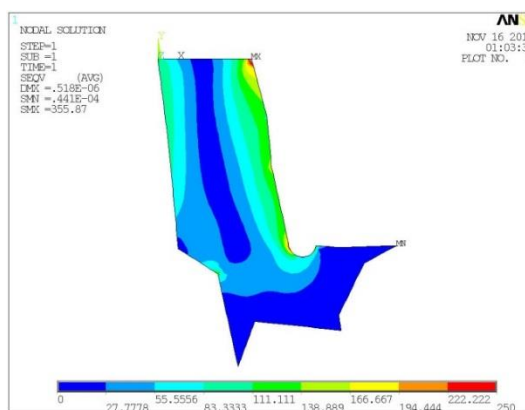
Iteração = 50

Tensão=568 MPa /Área=77112.02 mm² *



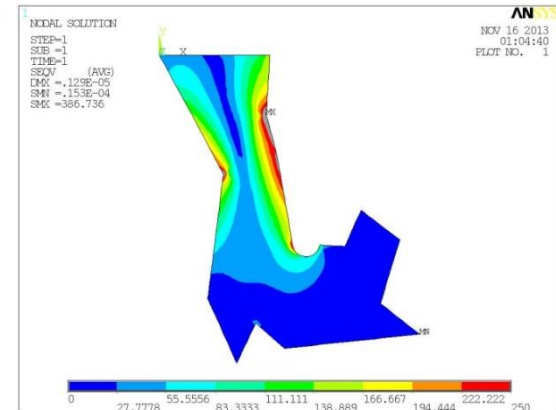
Iteração = 100

Tensão = 274 MPa/Área = 26464,09 mm²



Iteração = 150

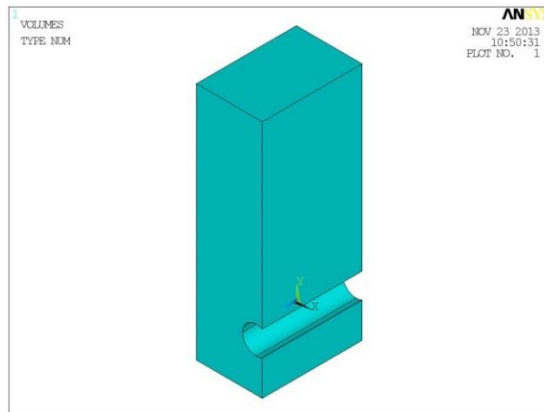
Tensão = 355 MPa/Área = 75742.88 mm² *



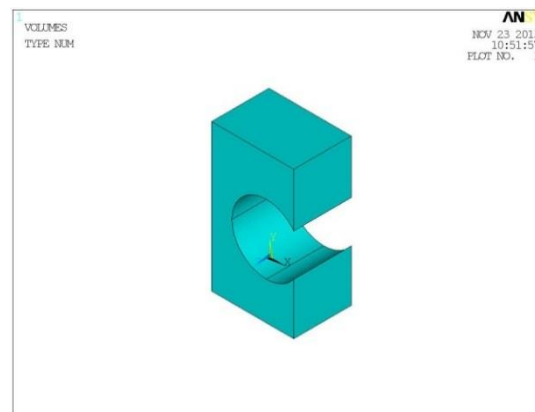
Iteração = 200

Tensão=386 MPa /Área=76507.06 mm² *

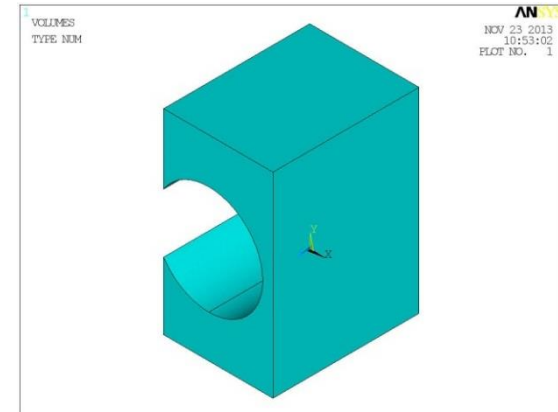
PC 0.65 PM 0.001 e ESCALONAMENTO 1.2



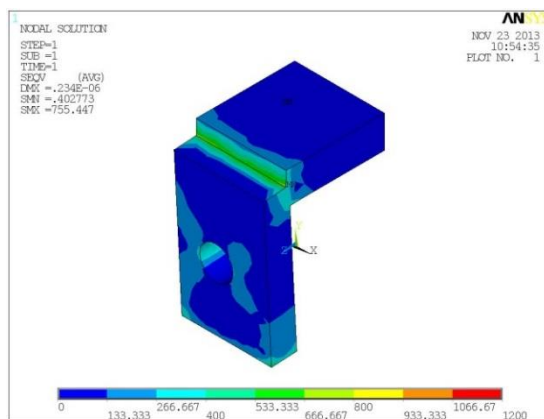
Iteração = 01
NÃO CONVERGIU



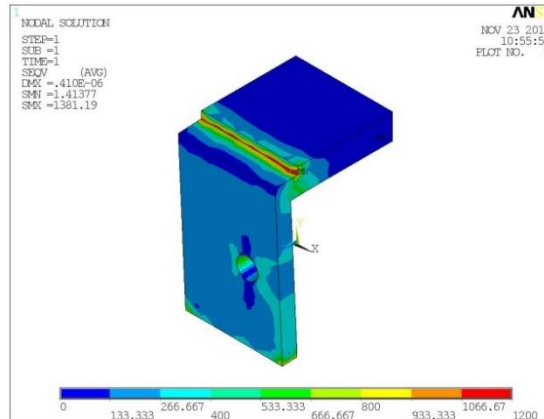
Iteração = 25
NÃO CONVERGIU



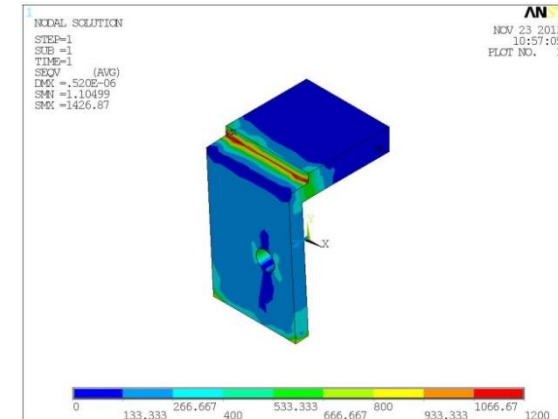
Iteração = 50
NÃO CONVERGIU



Iteração = 100
Tensão = 755 MPa/Área = 20462,39 mm³

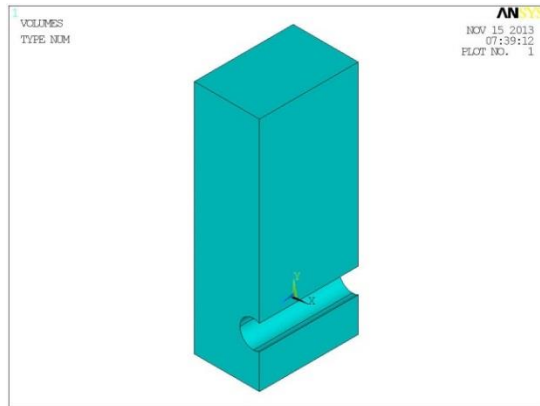


Iteração = 150
Tensão = 1381 MPa/Área = 12987,04 mm³

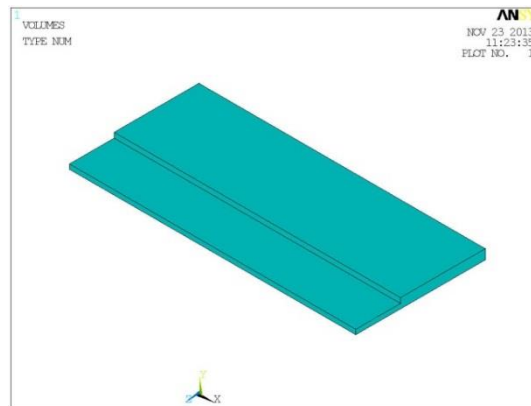


Iteração = 200
Tensão=1426 MPa /Área=12895,47 mm³

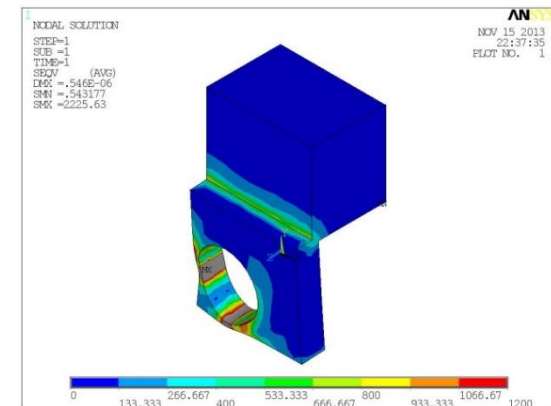
PC 0.65 PM 0.001 e ESCALONAMENTO 1.2 (SEM ELITISMO)



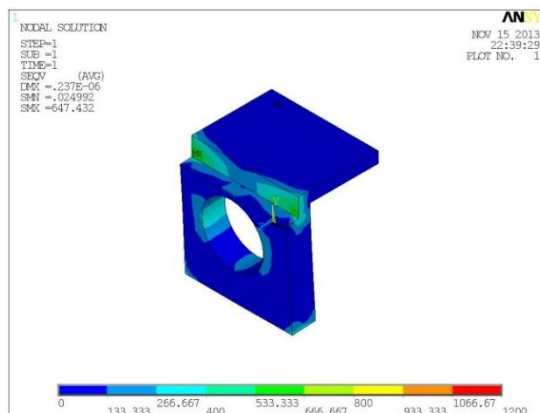
Iteração = 01
NÃO CONVERGIU



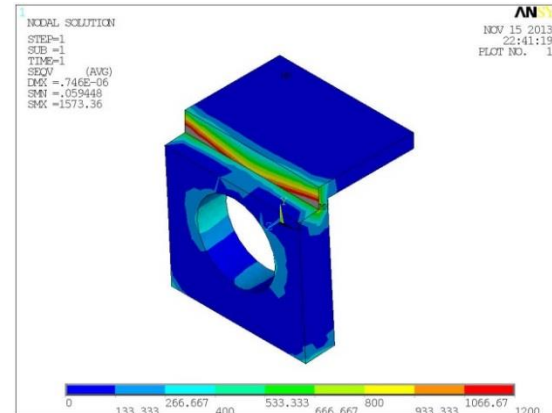
Iteração = 25
NÃO CONVERGIU



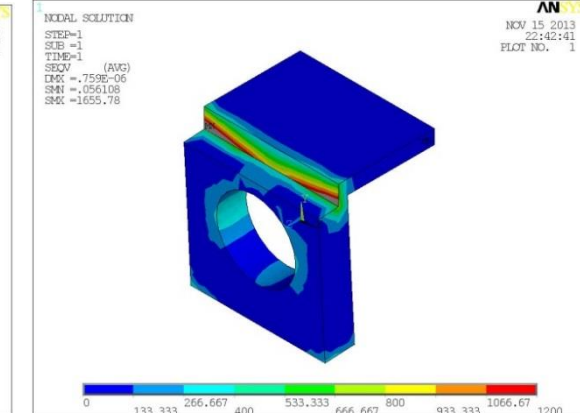
Iteração = 50
Tensão=2225 MPa /Área=168461,8 mm³*



Iteração = 100
Tensão = 647 MPa/Área = 21534,24 mm³

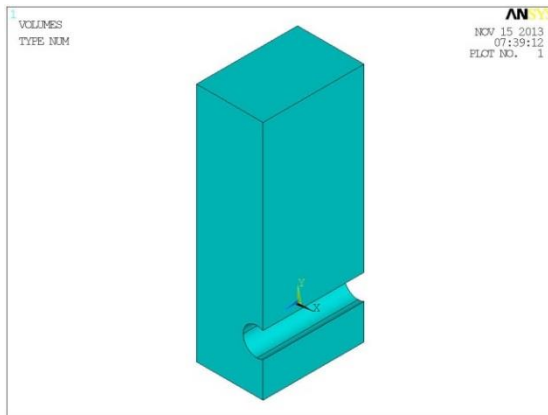


Iteração = 150
Tensão = 1573 MPa/Área = 21257,05 mm³

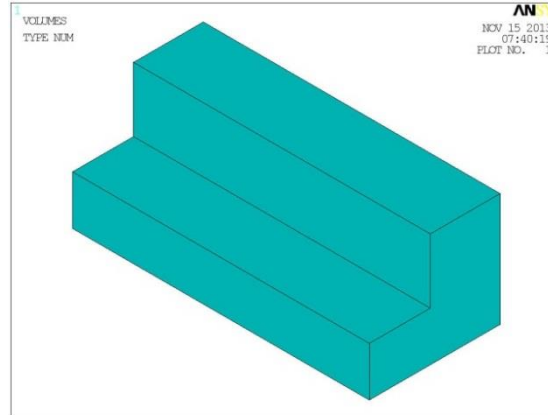


Iteração = 200
Tensão=1655 MPa /Área=20994,55 mm³

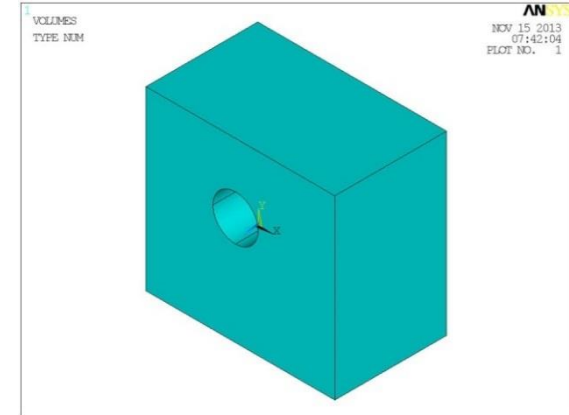
PC 0.65 PM 0.10 e ESCALONAMENTO 1.2



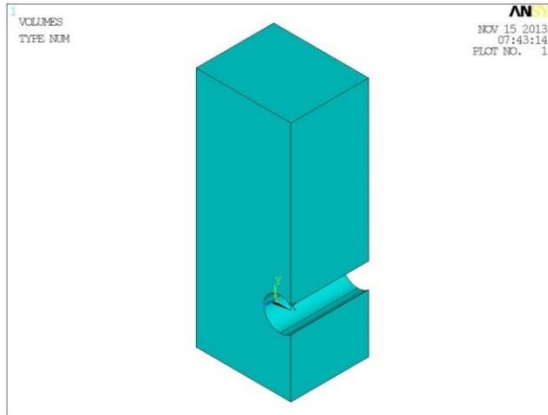
Iteração = 01
NÃO CONVERGIU



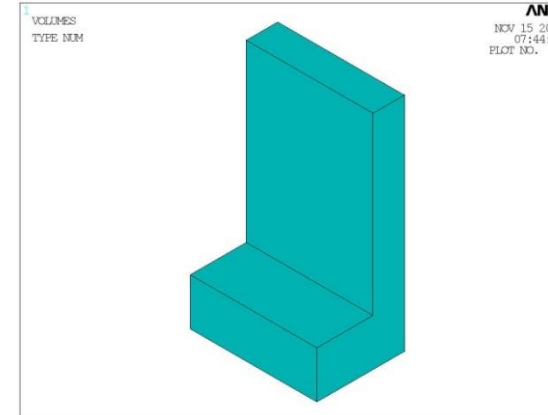
Iteração = 25
NÃO CONVERGIU



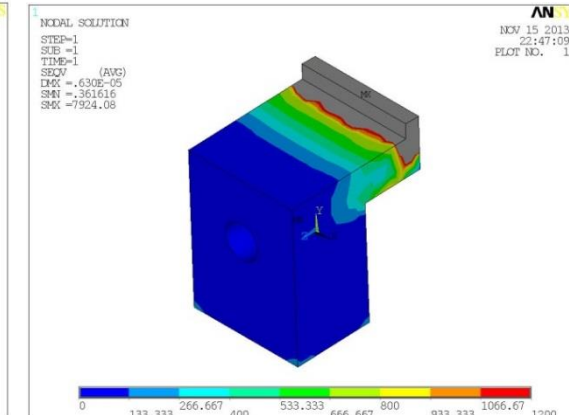
Iteração = 50
NÃO CONVERGIU



Iteração = 100
NÃO CONVERGIU

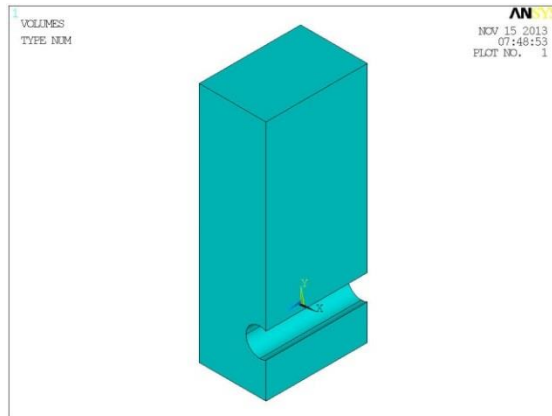


Iteração = 150
NÃO CONVERGIU

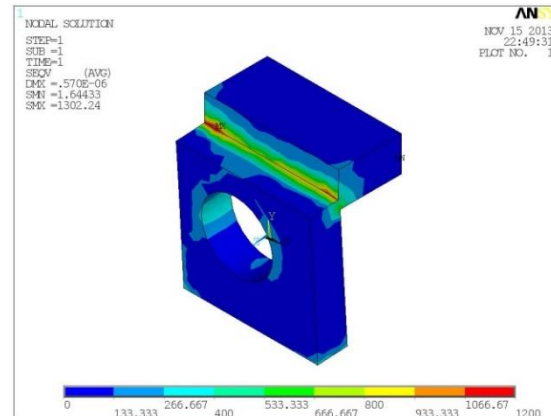


Iteração = 200
Tensão=7924 MPa /Área=130256,3 mm³ *

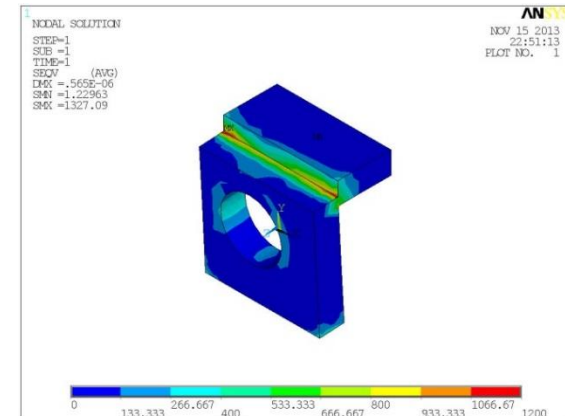
PC 0.65 PM 0.001 e ESCALONAMENTO 2.0



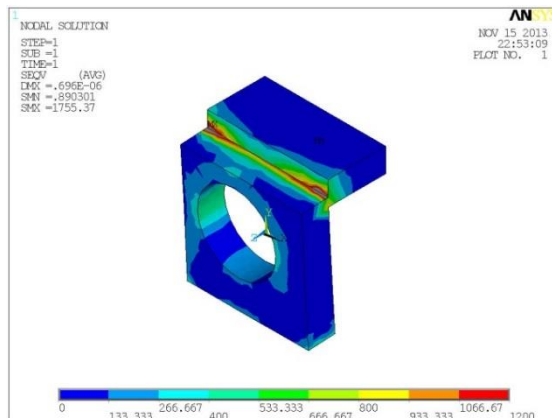
Iteração = 01
NÃO CONVERGIU



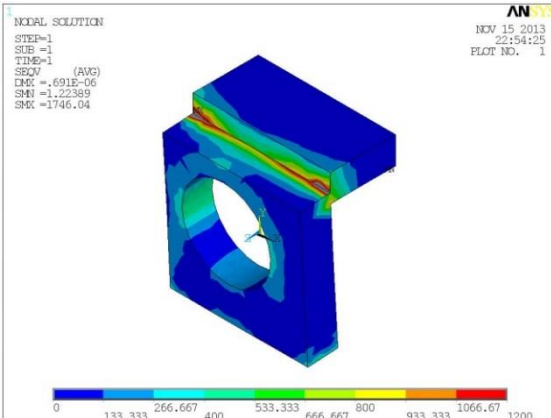
Iteração = 25
Tensão=1302 MPa /Área=27980.99 mm³



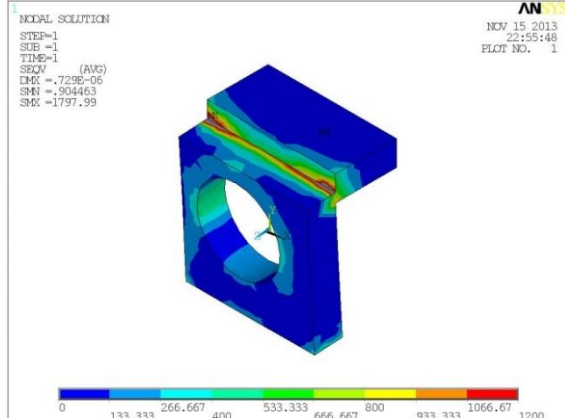
Iteração = 50
Tensão=1320 MPa /Área=24360,42 mm³



Iteração = 100
Tensão = 1755 MPa/Área = 21771,50 mm³

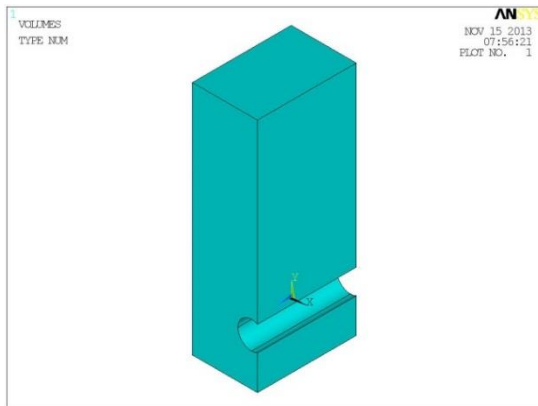


Iteração = 150
Tensão = 1746 MPa/Área = 21752,73 mm³

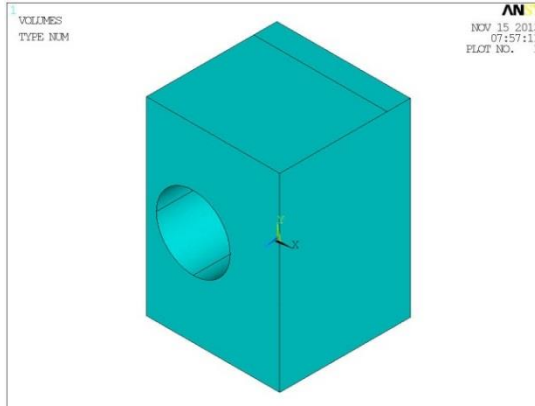


Iteração = 200
Tensão=1797 MPa /Área=21484,15mm³

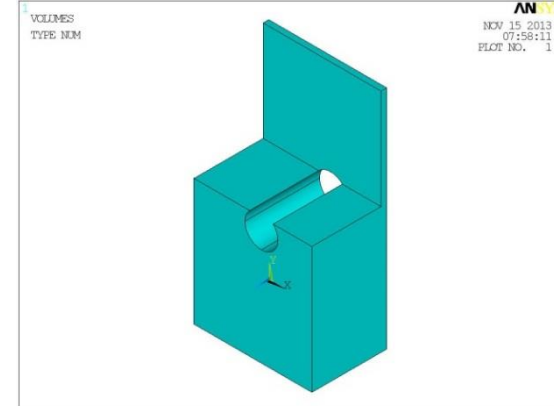
PC 0.65 PM 0.30 e ESCALONAMENTO 1.2



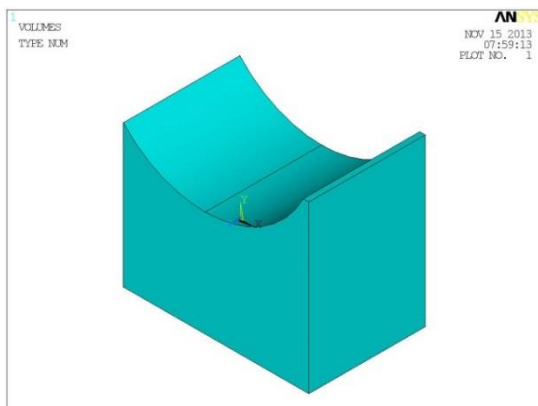
Iteração = 01
NÃO CONVERGIU



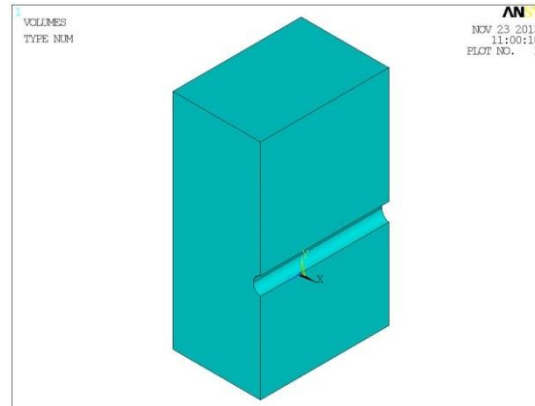
Iteração = 25
NÃO CONVERGIU



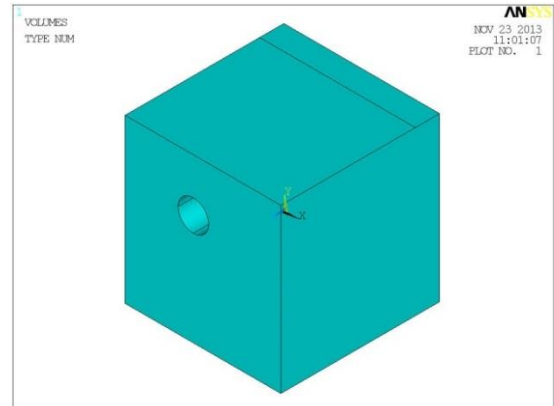
Iteração = 50
NÃO CONVERGIU



Iteração = 100
NÃO CONVERGIU

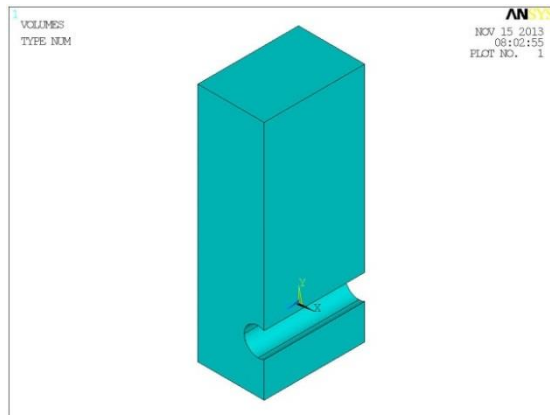


Iteração = 150
NÃO CONVERGIU

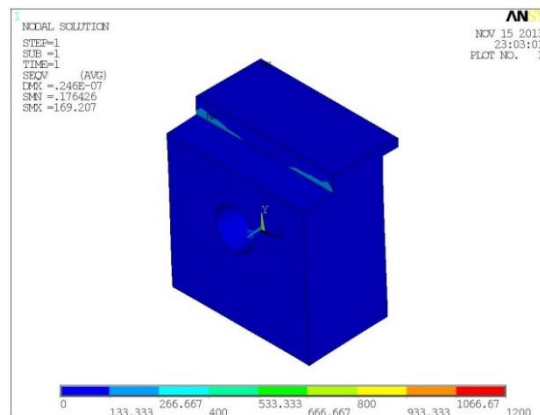


Iteração = 200
NÃO CONVERGIU

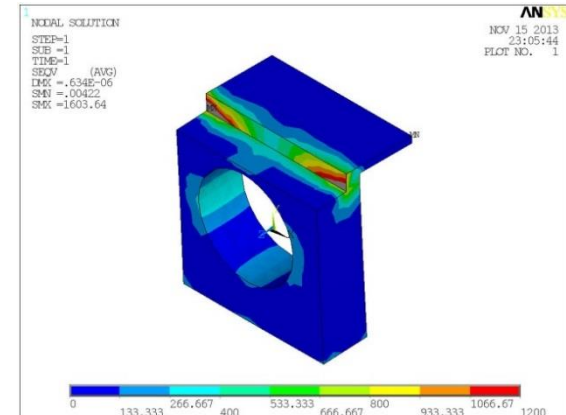
PC 0.90 PM 0.001 e ESCALONAMENTO 1.2



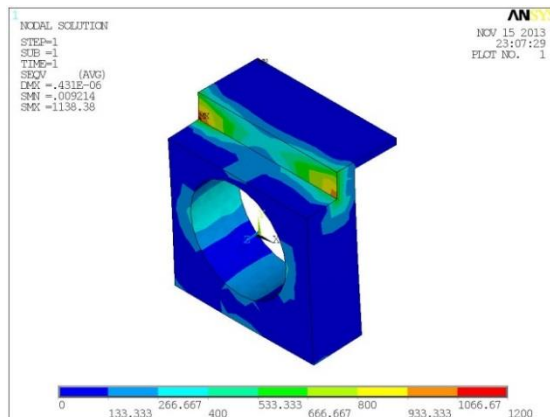
Iteração = 01
NÃO CONVERGIU



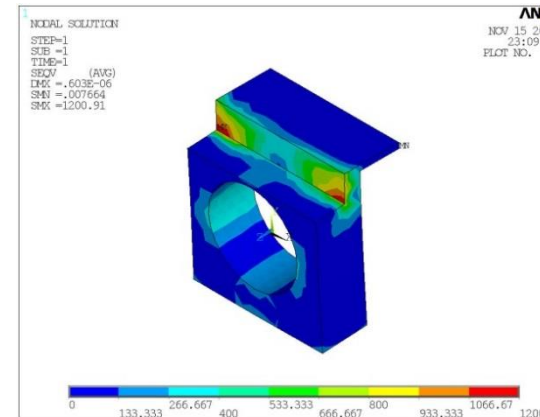
Iteração = 25
Tensão=169 MPa /Área=53594,94 mm³



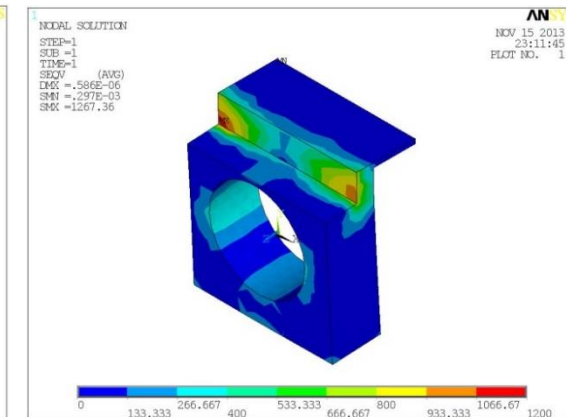
Iteração = 50
Tensão=1603 MPa /Área=28571,92 mm³



Iteração = 100
Tensão = 1138 MPa /Área = 20044,42 mm³

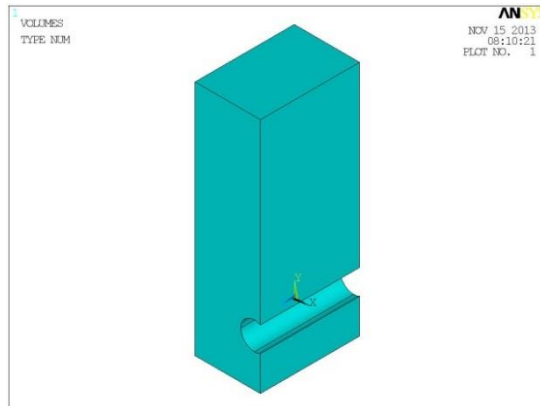


Iteração = 150
Tensão = 1200 MPa /Área = 19201,71 mm³

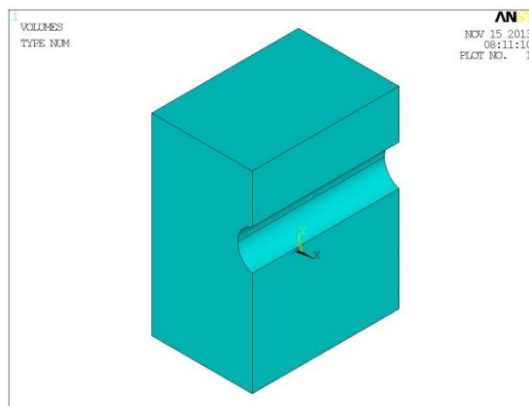


Iteração = 200
Tensão=1267 MPa /Área=18854,34 mm³

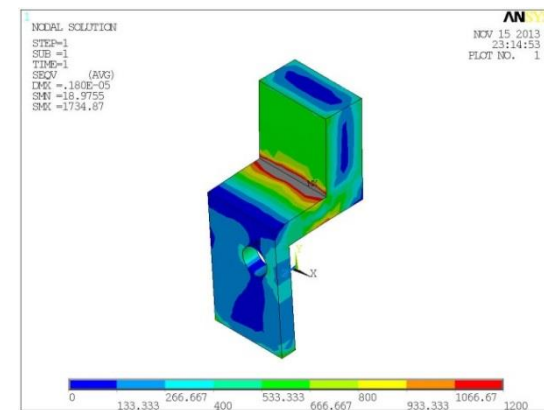
PC 0.90 PM 0.10 e ESCALONAMENTO 1.2



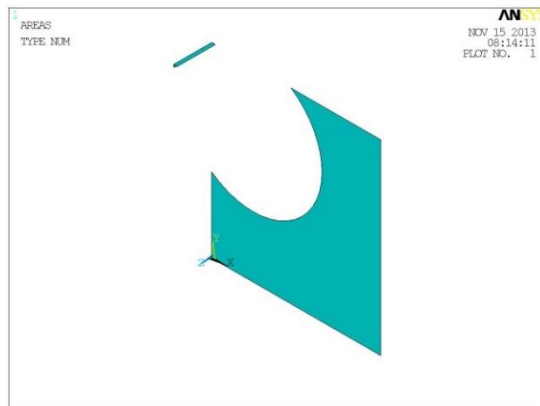
Iteração = 01
NÃO CONVERGIU



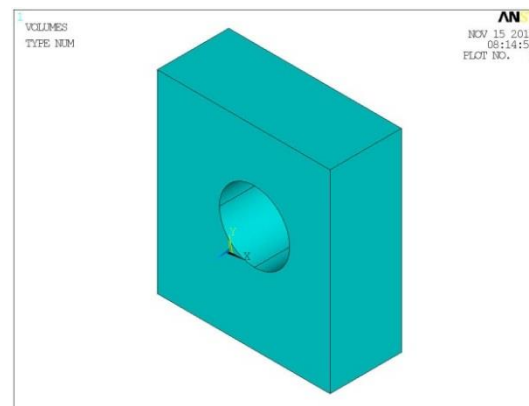
Iteração = 25
NÃO CONVERGIU



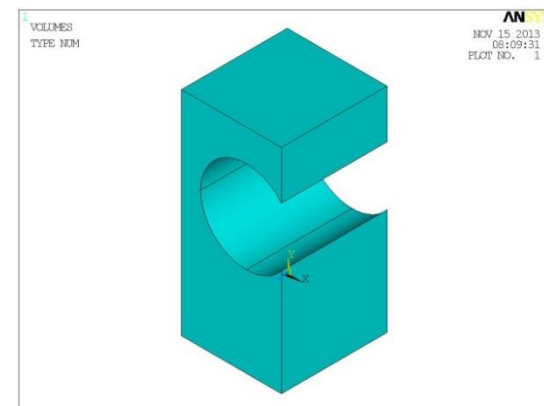
Iteração = 50
Tensão=1734 MPa /Área=128098.6 mm³ *



Iteração = 100
NÃO CONVERGIU



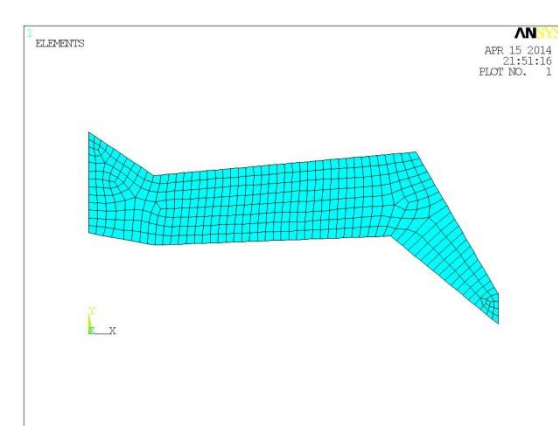
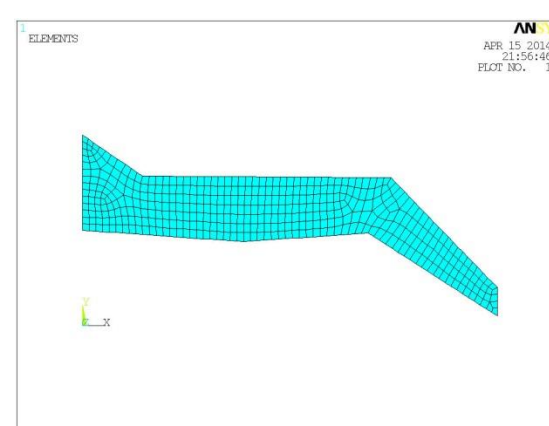
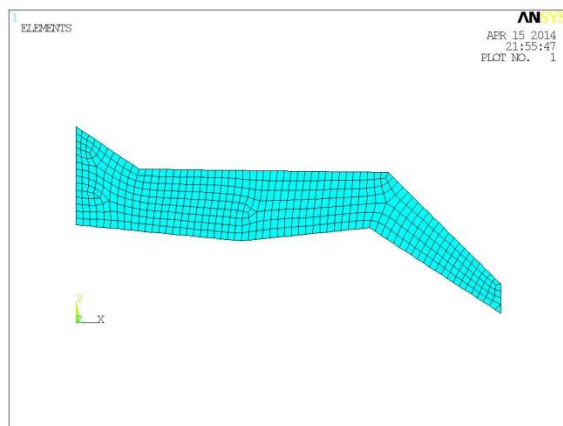
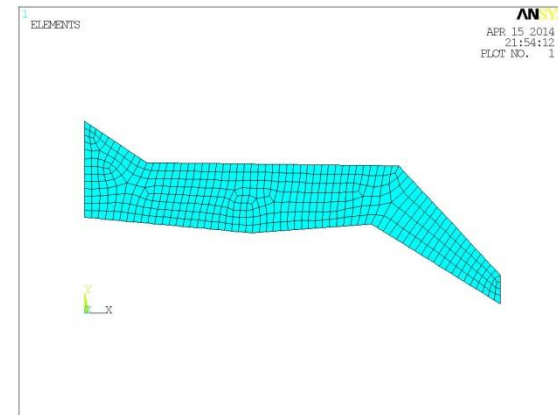
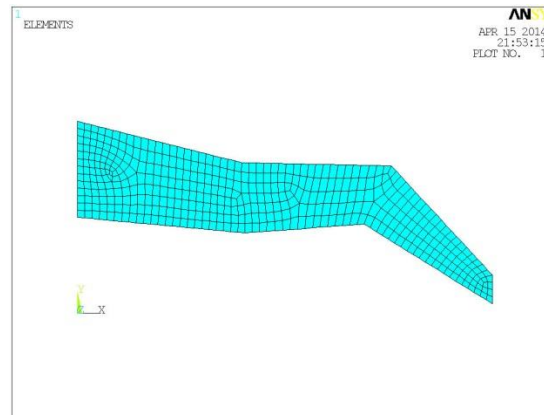
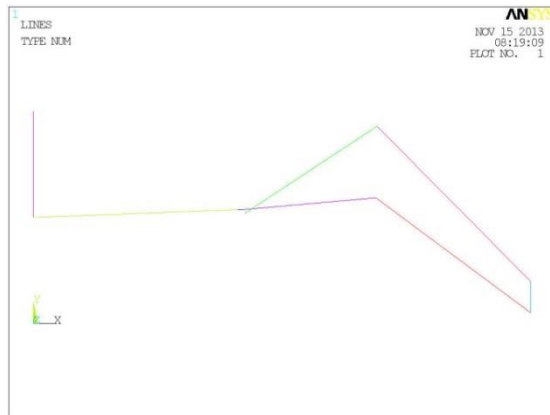
Iteração = 150
NÃO CONVERGIU



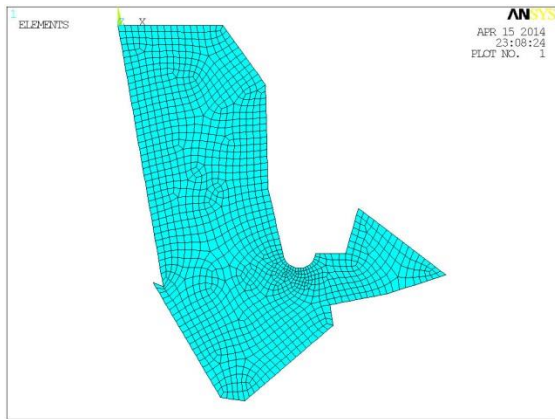
Iteração = 200
NÃO CONVERGIU

APÊNDICE B – Discretização do domínio em função do número de iteração

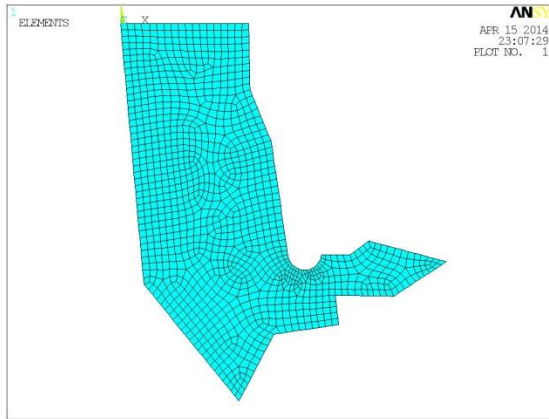
PC 0.65 PM 0.001 – Placa Bidimensional



PC 0.65 PM 0.001 – Gancho Bidimensional



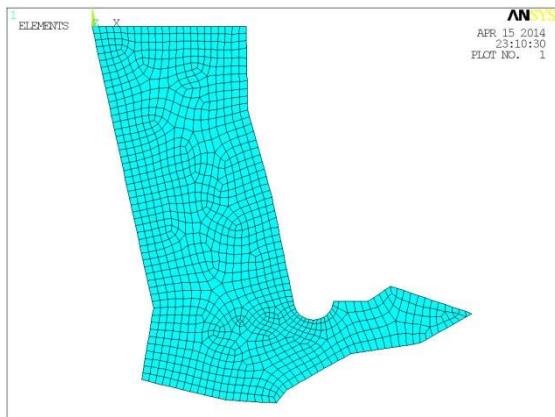
Iteração = 01



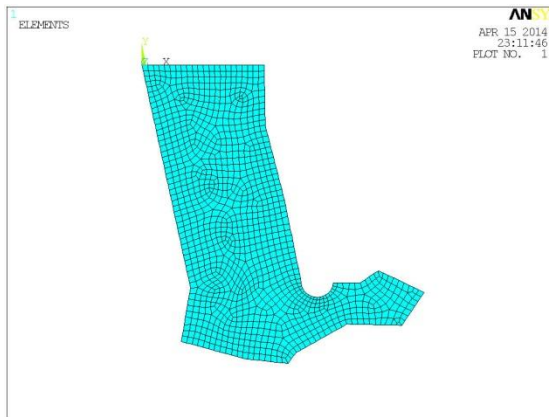
Iteração = 25



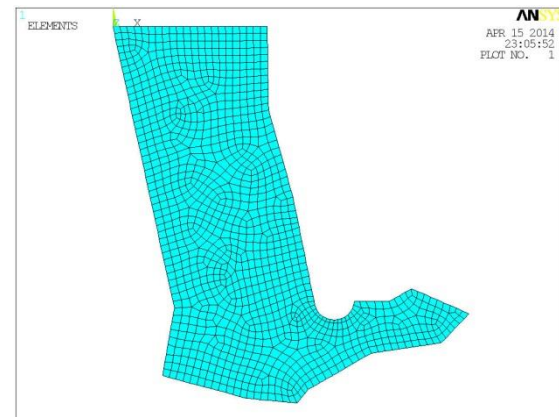
Iteração = 50



Iteração = 100

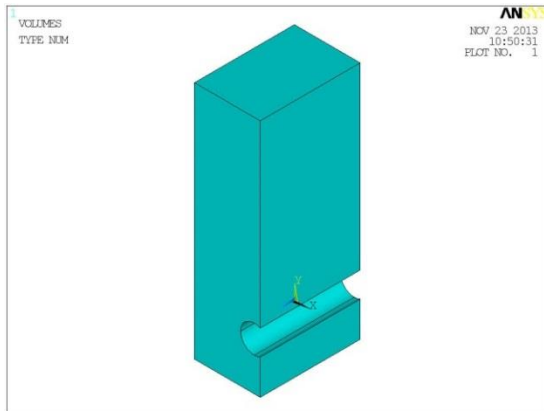


Iteração = 150

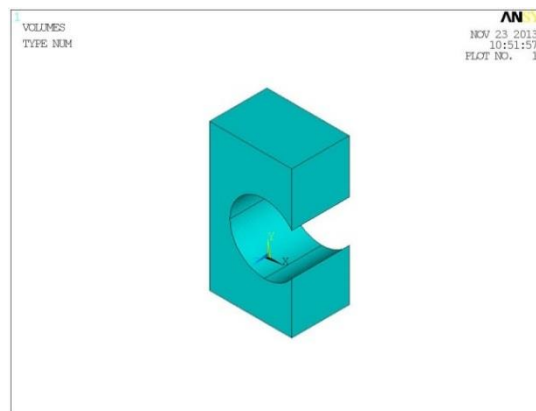


Iteração = 200

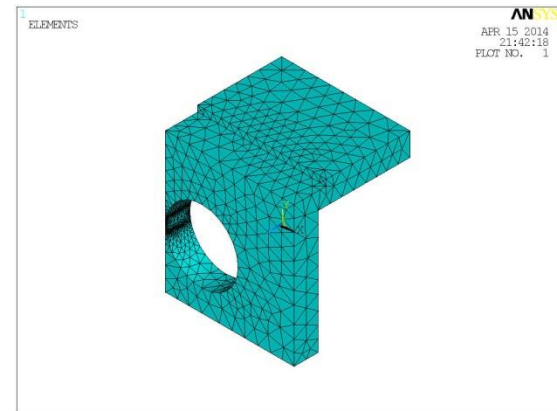
PC 0.65 PM 0.001 – Estudo Tridimensional



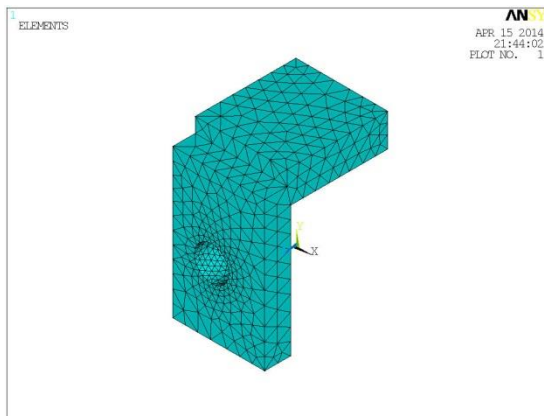
Iteração = 01



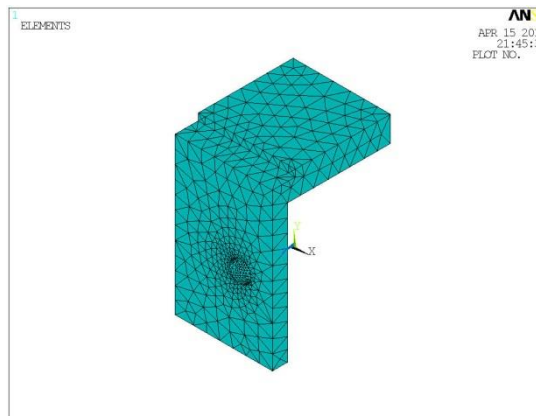
Iteração = 25



Iteração = 50



Iteração = 100

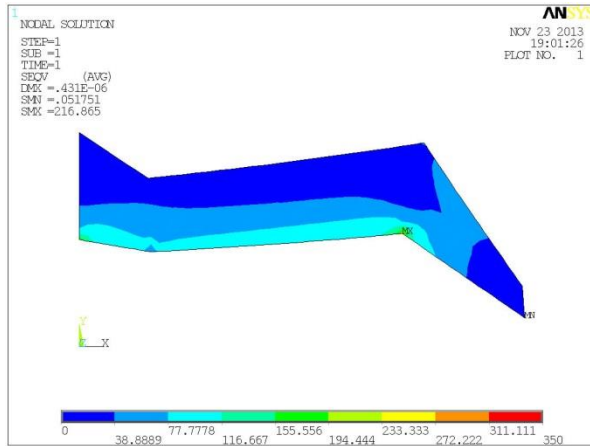


Iteração = 150

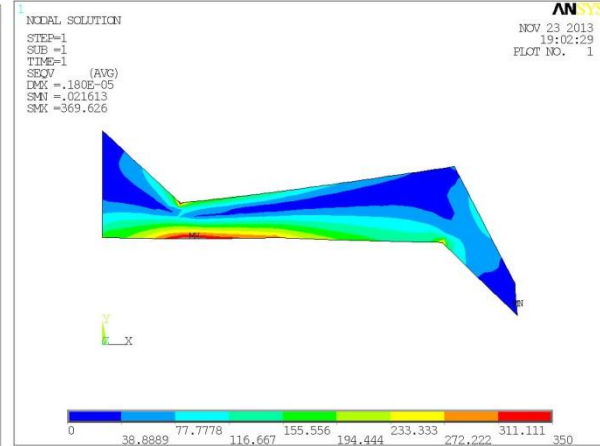


Iteração = 200

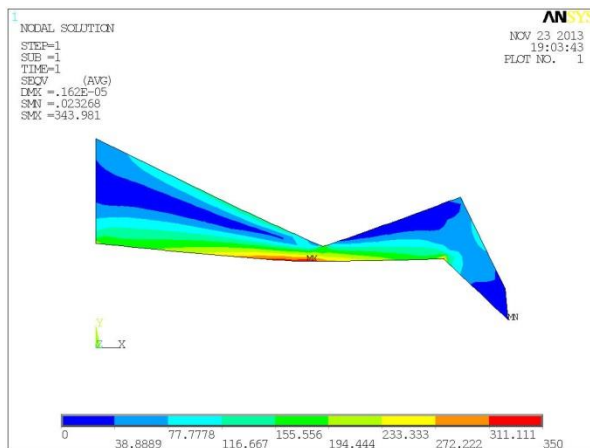
APÊNDICE C – Influência do aumento da população



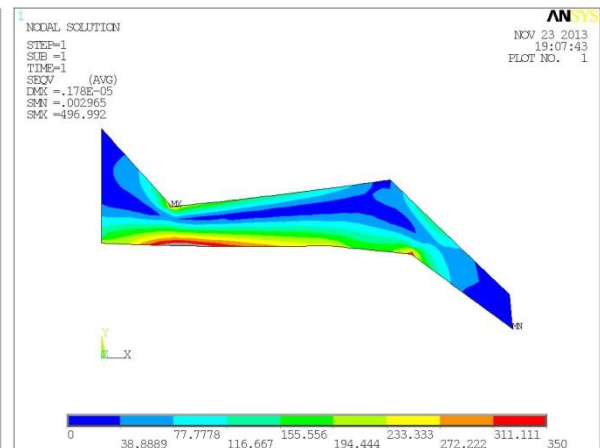
População = 20
Tensão = 351,57 MPa
Área = 7110,01 mm²
Tempo = 283 min.



População = 50
Tensão = 446,91 MPa
Área = 5371,87 mm²
Tempo = 681 min.



População = 80
Tensão = 389,27 MPa
Área = 5204,41 mm²
Tempo = 1104 min.



População = 100
Tensão = 394,02 MPa
Área = 4706,31 mm²
Tempo = 1677 min.

APÊNDICE D – Programa GAnsys

```
PROGRAM GANSYS
USE DFPORT
```

* ----- AJUSTAR OS PARÂMETROS CORRETAMENTE PARA UM MELHOR DESEMPENHO -----

```
IMPLICIT REAL (A-H,J-Z)
INTEGER GENER,PSIZE,NUNKNO,SUBLEN,TSLENG,MAXG,NEWIND,MATE1,
+ MATE2,FITIND
```

* Com o intuito de mudar a sequencia randomica, o ISEED devera ser alterado, desta forma, mudando o RND().

* Deve ser PAR.

```
PARAMETER(PSIZE = 20)
PARAMETER(NUNKNO = 32)

* Todas as sub-strings tem o mesmo comprimento.



```
PARAMETER(SUBLEN = 10)
PARAMETER(TSLENG = NUNKNO*SUBLEN)
PARAMETER(MAXG = 200)
PARAMETER(CP = 0.90)
PARAMETER(MP = 0.001)
CHARACTER*3 ELITE
PARAMETER(ELITE = 'on')
PARAMETER(SCALEC = 1.2)
```


```

* -----DECLARAR TODAS AS VARIÁVEIS (GLOBAIS)-----

* Esta ARRAY define a tamanho do individuo da populacao atual

```
REAL UNKNO(PSIZE,NUNKNO), FIT(PSIZE)
INTEGER INTEG(SIZE, NUNKNO)
INTEGER STRING(PSIZE, TSLENG)
```

* A nova populacao.

```
INTEGER NSTRIN(PSIZE, TSLENG)
```

* Esta ARRAY define o RANGE das incognitas.

```
REAL RANGE(2, NUNKNO)
```

* O melhor individuo do passado GENER. Usado se ELITE estiver ativado.

```

INTEGER ESTRIN(TSLENG)
INTEGER EINTEG(NUNKNO)
REAL EliteFIT
REAL EUNKNO(NUNKNO)

* -----

* Define o range de cada incognita.
CALL DefineRange(RANGE,NUNKNO)

* ----- INICIO DO ALGORITMO GENETICO -----

* ----- CRIA A POPULACAO INICIAL, GERACAO 1 -----

GENER = 1

* Constroe a populacao de modo randomico.
CALL InitialPopulation(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,
+ STRING,UNKNO,RANGE)

* Encontra o fitness de cada membro da populacao.
CALL FindFIT(Psize,UNKNO,NUNKNO,FIT,GENER)

* Encontra a media do fitness e o fittest individual.
CALL Statistics(MEANF,SUMFIT,FITIND,Psize,FIT,INTEGS,
+ STRING,ESTRIN,EINTEG,EUNKNO,NUNKNO,TSLENG,UNKNO,ELITE,EFIT)

* Caso esteja utilizando o escalamento linear para definicao da fitness, tais escalamentos serão
priorizados para uma nova selecao.
CALL Scaling(SCALEC,FITIND,SUMFIT,MEANF,FIT,Psize)

* ----- LOOP PARA TODAS AS GERACOES -----

DO 1 GENER = 2 , MAXG

* Loop sobre todas as populacoes selecionando-as em pares.
DO 2 NEWIND = 1 , Psize, 2

CALL Selection(MATE1,SUMFIT,Psize,FIT)

CALL Selection(MATE2,SUMFIT,Psize,FIT)

```

```

        WRITE(*,*)MATE1,MATE2
*   Chamar a populacao e efetuar (OU NAO) o cruzamento.
        IF (RND().LE.CP) THEN
            CALL CrossOver(MATE1, MATE2, NEWIND,TSLENG,NSTRIN,PSIZE,
+ STRING)
        ELSE
            CALL NoCrossover(MATE1,MATE2,NEWIND,PSIZE,TSLENG,STRING,
+ NSTRIN)
        ENDIF

2    CONTINUE

*   Efetuar a mutacao da populacao..
        CALL Mutate(PSIZE,TSLENG,NSTRIN,MP)

*   Substituir a populacao anterior, pela nova populacao criada.
        CALL Replace(PSIZE,TSLENG,STRING,NSTRIN)

*   Decodificar a populacao de numeros inteiros para numeros reais.
        CALL FindUNKNO(PSIZE,TSLENG,NUNKNO,SUBLEN,INTEGS,STRING,
+ UNKNO,RANGE)

*   Encontrar a fitness de cada membro da populacao.
        CALL FINDFIT(PSIZE,UNKNO,NUNKNO,FIT,GENER)

*   Encontra a media do fitness e o fittest individual.
        CALL Statistics(MEANF,SUMFIT,FITIND,PSIZE,FIT,INTEGS,
+ STRING,ESTRIN,EINTEG,EUNKNO,NUNKNO,TSLENG,UNKNO,ELITE,EFIT)

*   Caso esteja utilizando o escalamento linear para definicao da fitness, tais escalamentos serão
priorizados para uma nova selecao.
        CALL Scaling(SCALEC,FITIND,SUMFIT,MEANF,FIT,PSIZE)

1    CONTINUE

*   Fechar todos os arquivos
        CLOSE (1)
        CLOSE (2)
        CLOSE (11)

        END
        SUBROUTINE CrossOver (MATE1, MATE2, NEWIND,TSLENG,NSTRIN,PSIZE,
+ STRING)

```

- * Realiza o cruzamento em um unico ponto.
 IMPLICIT REAL (A-H,J-Z)
 INTEGER BIT,CSITE,NEWIND,TSLENG,PSIZE,MATE1,MATE2
 INTEGER STRING(PSIZE,TSLENG)
 INTEGER NSTRIN(PSIZE,TSLENG)
 write(*,*)'CROSS'
- * Selecao da posicao de forma randomica
 CSITE = (TSLENG - 1)* RND() + 1

- * Trocar os bits dos pais da esquerda
 DO 1 BIT =1 , CSITE
 NSTRIN(NEWIND, BIT) = STRING(MATE1, BIT)
 NSTRIN(NEWIND + 1, BIT) = STRING(MATE2, BIT)
- 1 CONTINUE

- * Trocar os bits dos pais da direita
 DO 2 BIT = CSITE + 1 , TSLENG
 NSTRIN(NEWIND, BIT) = STRING(MATE2, BIT)
 NSTRIN(NEWIND + 1, BIT) = STRING(MATE1, BIT)
- 2 CONTINUE

RETURN
 END

SUBROUTINE DefineRange(RANGE,NUNKNO)

- * Definir os limites superiores e inferiores

IMPLICIT REAL (A-H,J-Z)
 INTEGER UNKNOW,NUNKNO
 REAL RANGE(2,NUNKNO)

- * Primeira Incógnita X1
 UNKNOW = 1
- * Limite Inferior
 RANGE(1,UNKNOW) = 0
- * Limite Superior
 RANGE(2,UNKNOW) = 0

- * Segunda Incógnita Y1
 UNKNOW = 2
 RANGE(1,UNKNOW) = 0
 RANGE(2,UNKNOW) = 0

- * Terceira Incógnita X2

UNKNOWN = 3
RANGE(1,UNKNOWN) = 50
RANGE(2,UNKNOWN) = 120

* Quarta Incógnita Y2
UNKNOWN = 4
RANGE(1,UNKNOWN) = 0
RANGE(2,UNKNOWN) = 0

* Quinta Incógnita X3
UNKNOWN = 5
RANGE(1,UNKNOWN) = 65
RANGE(2,UNKNOWN) = 115

* Sexta Incógnita Y3
UNKNOWN = 6
RANGE(1,UNKNOWN) = 35
RANGE(2,UNKNOWN) = 75

* Sétima Incógnita X4
UNKNOWN = 7
RANGE(1,UNKNOWN) = 80
RANGE(2,UNKNOWN) = 120

* Oitava Incógnita Y4
UNKNOWN = 8
RANGE(1,UNKNOWN) = 80
RANGE(2,UNKNOWN) = 135

* Nona Incógnita X5
UNKNOWN = 9
RANGE(1,UNKNOWN) = 125
RANGE(2,UNKNOWN) = 125

* Décima Incógnita Y5
UNKNOWN = 10
RANGE(1,UNKNOWN) = 150
RANGE(2,UNKNOWN) = 180

* Décima-Primeira Incógnita X6
UNKNOWN = 11
RANGE(1,UNKNOWN) = 150
RANGE(2,UNKNOWN) = 150

- * Décima-Segunda Incógnita Y6
 $UNKNOWN = 12$
 $RANGE(1, UNKNOWN) = 150$
 $RANGE(2, UNKNOWN) = 180$
- * Décima-Terceira Incógnita X7
 $UNKNOWN = 13$
 $RANGE(1, UNKNOWN) = 160$
 $RANGE(2, UNKNOWN) = 180$
- * Décima-Quarta Incógnita Y7
 $UNKNOWN = 14$
 $RANGE(1, UNKNOWN) = 75$
 $RANGE(2, UNKNOWN) = 180$
- * Décima-Quinta Incógnita X8
 $UNKNOWN = 15$
 $RANGE(1, UNKNOWN) = 181$
 $RANGE(2, UNKNOWN) = 220$
- * Décima-Sexta Incógnita Y8
 $UNKNOWN = 16$
 $RANGE(1, UNKNOWN) = 75$
 $RANGE(2, UNKNOWN) = 180$
- * Décima-Setima Incógnita X9
 $UNKNOWN = 17$
 $RANGE(1, UNKNOWN) = 221$
 $RANGE(2, UNKNOWN) = 280$
- * Décima-Oitava Incógnita Y9
 $UNKNOWN = 18$
 $RANGE(1, UNKNOWN) = 150$
 $RANGE(2, UNKNOWN) = 190$
- * Décima-Nona Incógnita X10
 $UNKNOWN = 19$
 $RANGE(1, UNKNOWN) = 170$
 $RANGE(2, UNKNOWN) = 270$
- * Vigésima Incógnita Y10
 $UNKNOWN = 20$
 $RANGE(1, UNKNOWN) = 160$
 $RANGE(2, UNKNOWN) = 220$

- * Vigésima-Primeira Incógnita X11
 $\text{UNKNOWN} = 21$
 $\text{RANGE}(1, \text{UNKNOWN}) = 140$
 $\text{RANGE}(2, \text{UNKNOWN}) = 260$
- * Vigésima-Segunda Incógnita Y11
 $\text{UNKNOWN} = 22$
 $\text{RANGE}(1, \text{UNKNOWN}) = 200$
 $\text{RANGE}(2, \text{UNKNOWN}) = 300$
- * Vigésima-Terceira Incógnita X12
 $\text{UNKNOWN} = 23$
 $\text{RANGE}(1, \text{UNKNOWN}) = 100$
 $\text{RANGE}(2, \text{UNKNOWN}) = 180$
- * Vigésima-Quarta Incógnita Y12
 $\text{UNKNOWN} = 24$
 $\text{RANGE}(1, \text{UNKNOWN}) = 205$
 $\text{RANGE}(2, \text{UNKNOWN}) = 300$
- * Vigésima-Quinta Incógnita X13
 $\text{UNKNOWN} = 25$
 $\text{RANGE}(1, \text{UNKNOWN}) = 90$
 $\text{RANGE}(2, \text{UNKNOWN}) = 125$
- * Vigésima-Sexta Incógnita Y13
 $\text{UNKNOWN} = 26$
 $\text{RANGE}(1, \text{UNKNOWN}) = 230$
 $\text{RANGE}(2, \text{UNKNOWN}) = 300$
- * Vigésima-Setima Incógnita X14
 $\text{UNKNOWN} = 27$
 $\text{RANGE}(1, \text{UNKNOWN}) = 70$
 $\text{RANGE}(2, \text{UNKNOWN}) = 90$
- * Vigésima-Oitava Incógnita Y14
 $\text{UNKNOWN} = 28$
 $\text{RANGE}(1, \text{UNKNOWN}) = 230$
 $\text{RANGE}(2, \text{UNKNOWN}) = 300$
- * Vigésima-Nona Incógnita X15
 $\text{UNKNOWN} = 29$
 $\text{RANGE}(1, \text{UNKNOWN}) = 0$
 $\text{RANGE}(2, \text{UNKNOWN}) = 70$

```

*      Trigesima Incógnita Y15
      UNKNOW = 30
      RANGE(1,UNKNOW) = 170
      RANGE(2,UNKNOW) = 250

*      Trigesima-Primeira Incógnita X16
      UNKNOW = 31
      RANGE(1,UNKNOW) = 0
      RANGE(2,UNKNOW) = 60

*      Trigesima-Segunda Incógnita Y16
      UNKNOW = 32
      RANGE(1,UNKNOW) = 90
      RANGE(2,UNKNOW) = 210


      RETURN
      END


      SUBROUTINE ELITES (FITIND,PSIZE,NUNKNO,TSLENG,UNKNO,FIT,
+ INTEG,STRING,ESTRIN,EINTEG,EUNKNO,EFIT)
*      Aplica ELITE substituindo um individuo escolhido randomicamente por um membro da elite
*      da populacao anterior se o novo fitness for menor que o valor anterior.
      IMPLICIT REAL (A-H,J-Z)
      INTEGER BIT,FITIND,PSIZE,NUNKNO,TSLENG,UNKNOW
      REAL UNKNOW(PSIZE,NUNKNO), FIT(PSIZE)
      INTEGER INTEG(PSIZE, NUNKNO)
      INTEGER STRING(PSIZE, TSLENG)
      INTEGER ESTRIN(TSLENG)
      INTEGER EINTEG(NUNKNO)
      REAL EUNKNO(NUNKNO)
      write(*,*)'ELITE'
      IF (FIT(FITIND).LT.EFIT) THEN

*      Escolher o individuo que sera substituido.
      INDIV = INT(PSIZE* RND() + 1)

      DO 1 BIT = 1 , TSLENG
        STRING(INDIV, BIT) = ESTRIN(BIT)
1      CONTINUE

      FIT(INDIV) = EFIT

```

```

DO 2 UNKNOW = 1 , NUNKNO
  INTEG(SINDIV, UNKNOW) = EINTEG(UNKNOW)
  UNKNO(SINDIV, UNKNOW) = EUNKNO(UNKNOW)
2  CONTINUE

FITIND = SINDIV

ENDIF

DO 3 BIT = 1 , TSLENG
  ESTRIN(BIT) = STRING(FITIND, BIT)
3  CONTINUE

EFIT = FIT(FITIND)

DO 4 UNKNOW = 1 , NUNKNO
  EINTEG(UNKNOW) = INTEG(FITIND, UNKNOW)
  EUNKNO(UNKNOW) = UNKNO(FITIND, UNKNOW)
4  CONTINUE

RETURN
END

SUBROUTINE FindFIT(PSIZE,UNKNO,NUNKNO,FIT,GENER)

IMPLICIT REAL (A-H,J-Z)
INTEGER PSIZE,NUNKNO
  INTEGER :: II=0
  INTEGER :: GENER
  INTEGER :: COUNT_JOBS=1
REAL UNKNO(PSIZE,NUNKNO)
REAL FIT(PSIZE)
REAL x1,Stress,Area
  REAL DISTANCIA
  REAL :: AreaXLS=0
  REAL :: StressXLS=0

OPEN (11, FILE = 'E:\Mestrado\GA\Gancho\Evolucao.txt')
OPEN (12, FILE = 'E:\Mestrado\GA\Gancho\EvolucaoXLS.txt')
OPEN (3, FILE = 'E:\Mestrado\GA\Gancho\Resultados.txt')
OPEN (15, FILE = 'E:\Mestrado\GA\Gancho\ResultadosXLS.txt')

DO 1 SINDIV = 1 , PSIZE

```

```

DO I=1,NUNKNO,1

    IF (I == 12) THEN
        x1 = UNKNO(INDIV, I-2)
        WRITE(3,*) x1

    ELSEIF (I == 16) THEN
        x1 = UNKNO(INDIV, I-2)
        WRITE(3,*) x1

    ELSE

        x1 = UNKNO(INDIV, I)
        WRITE(3,*) x1

    ENDIF

    IF (INDIV == PSIZE) THEN
        WRITE(15,*) x1
        IF (I == NUNKNO) THEN
            WRITE(15,*) 'ITE=',GENER + 1,'-----'
        ELSE
            ENDIF
        ELSE
            ENDIF
    ENDIF

ENDDO

```

```

OPEN (13, FILE = 'E:\Mestrado\GA\Gancho\COUNT_JOBS.txt')
    WRITE(13,*)    COUNT_JOBS
    COUNT_JOBS=COUNT_JOBS+1
    REWIND(13)
    CLOSE(13)

    RESULT = SYSTEM ('E:\Mestrado\GA\Gancho\Gansys.bat')

    REWIND(3)

    OPEN (6, FILE = 'E:\Mestrado\GA\Gancho\Geracao.txt')
    READ(6,*) Geracao

```

```
DO WHILE (Geracao == 0) ! Tempo para rodar o Ansys
ENDDO
CLOSE (6)
```

```
Geracao2 = NINT(Geracao)
```

```
IF (Geracao2 .NE. 1) THEN
```

```
Stress = 7000
Area = 100000
```

```
PRINT *, 'ERRO DE GEOMETRIA'
WRITE(11,*) 'ITE=', GENER, 'ERRO DE GEOMETRIA'
```

```
ELSE
```

```
OPEN (4, FILE = 'E:\Mestrado\GA\Gancho\Stress.txt')
OPEN (5, FILE = 'E:\Mestrado\GA\Gancho\Area.txt')
```

```
READ(4,*) Stress
READ(5,*) Area
```

```
IF (Area < 15000 .OR. Stress > 250) THEN
    Area = Area+50000
    Stress = Stress + 1500
```

```
ELSE
```

```
ENDIF
```

```
REWIND(4)
REWIND(5)
```

```
PRINT *, 'TENSAO=', Stress, 'AREA =', Area
WRITE(11,*) 'ITE= ', GENER, ' AREA =', Area, ' Tensao =', Stress
```

```
ENDIF
```

```
FIT(INDIV)= 1/Area
```

```

        IF (INDIV == 20) THEN
            AreaXLS = Area + AreaXLS
            StressXLS = Stress + StressXLS
            AREAPRINT = AreaXLS/PSIZE
            STRESSPRINT = StressXLS/PSIZE
            WRITE(12,*)'ITE= ',GENER,' AREA =',AREAPRINT,
+ ' Tensao =', STRESSPRINT
            AreaXLS = 0
            StressXLS = 0
        ELSE
            AreaXLS = Area + AreaXLS
            StressXLS = Stress + StressXLS
        ENDIF

        CLOSE (4)
        CLOSE (5)

```

1 CONTINUE

```

        CLOSE (3)

```

```

RETURN
END

```

SUBROUTINE FindINTEGS(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,STRING)

* Decodificar o conjunto de strings de inteiros decimais.

```

IMPLICIT REAL (A-H,I-Z)

```

```

INTEGER Psize,TSLENG,NUNKNO,SUBLEN,BIT,SBIT

```

```

INTEGER INTEGS(Psize, NUNKNO)

```

```

INTEGER STRING(Psize, TSLENG)

```

```

DO 1 INDIV = 1 , Psize

```

```

    BIT = TSLENG + 1

```

```

    DO 2 UNKNOW = NUNKNO , 1 ,-1

```

```

        INTEGS(INDIV, UNKNOW) = 0

```

```

        DO 3 SBIT = 1 , SUBLEN

```

```

            BIT = BIT - 1

```

```

            IF (STRING(INDIV, BIT).EQ.1) THEN

```

```

                INTEGS(INDIV, UNKNOW) = INTEGS(INDIV, UNKNOW) +
+      2** (SBIT - 1)

```

```

            ENDIF

```

3 CONTINUE

2 CONTINUE

1 CONTINUE

```
RETURN
END
```

```
SUBROUTINE FindUNKNO(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,STRING,
+ UNKNO,RANGE)
```

```
* Decodificar as strings para numeros reais.
```

```
IMPLICIT REAL (A-H,J-Z)
```

```
INTEGER Psize,TSLENG,NUNKNO,SUBLEN,UNKNO
```

```
INTEGER INTEGS(Psize, NUNKNO)
```

```
INTEGER STRING(Psize, TSLENG)
```

```
REAL UNKNO(Psize,NUNKNO)
```

```
REAL RANGE(2, NUNKNO)
```

```
* Decodificar o conjunto de strings dos numeros inteiros decimais
```

```
CALL FindINTEGS(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,STRING)
```

```
* Converter os inteiros para reais
```

```
DO 1 INDIV = 1 , Psize
```

```
DO 2 UNKNOW = 1 , NUNKNO
```

```
UNKNO(INDIV, UNKNOW) = RANGE(1, UNKNOW) +
```

```
+ INTEGS(INDIV, UNKNOW)*
```

```
+ (RANGE(2, UNKNOW) - RANGE(1, UNKNOW)) / (2**SUBLEN - 1)
```

```
2 CONTINUE
```

```
1 CONTINUE
```

```
RETURN
```

```
END
```

```
SUBROUTINE InitialPopulation(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,
+ STRING,UNKNO,RANGE)
```

```
* Cria a populacao inicial de forma randomica.
```

```
IMPLICIT REAL (A-H,J-Z)
```

```
INTEGER BIT,Psize,TSLENG,NUNKNO,SUBLEN
```

```
INTEGER STRING(Psize, TSLENG)
```

```
DO 1 INDIV = 1 , Psize
```

```
DO 2 BIT = 1 , TSLENG
```

```
IF (RND().GT.0.5) THEN
```

```
STRING(INDIV, BIT) = 1
```

```
ELSE
```

```
STRING(INDIV, BIT) = 0
```

```
ENDIF
```

```
2 CONTINUE
```

```

1  CONTINUE

*   Converter os inteiros para reais
    CALL FindUNKNO(Psize,TSLENG,NUNKNO,SUBLEN,INTEGS,STRING,UNKNO,
+ RANGE)

    RETURN
    END

SUBROUTINE Mutate(Psize,TSLENG,NSTRIN,MP)
*   Inverter ocasionalmente o bit de "1" para "0" ou vice versa.
    IMPLICIT REAL (A-H,J-Z)
    INTEGER Psize,TSLENG,BIT
    INTEGER NSTRIN(Psize,TSLENG)
    write(*,*)'mutate',MP
    DO 1 INDIV = 1, Psize
        DO 2 BIT = 1, TSLENG
*       Verificar se o numero randomico e menor que a probabilidade de mutacao
            IF (RND().LE.MP) THEN
*           Mutate.
                IF (NSTRIN(INDIV, BIT).EQ.1) THEN
                    NSTRIN(INDIV, BIT) = 0
                ELSE
                    NSTRIN(INDIV, BIT) = 1
                ENDIF
            ENDIF
        ENDIF
    ENDIF
2  CONTINUE
1  CONTINUE

    RETURN
    END

SUBROUTINE NoCrossover(MATE1,MATE2,NEWIND,Psize,TSLENG,STRING,
+ NSTRIN)
*   Selecionar a string da nova populacao temporaria sem efetuar o cruzamento
    IMPLICIT REAL (A-H,J-Z)
    INTEGER MATE1,MATE2,NEWIND,Psize,TSLENG,BIT
    INTEGER STRING(Psize, TSLENG)
    INTEGER NSTRIN(Psize, TSLENG)
    write(*,*)'NOCROSS'
    DO 1 BIT = 1 , TSLENG
        NSTRIN(NEWIND, BIT) = STRING(MATE1, BIT)
        NSTRIN(NEWIND + 1, BIT) = STRING(MATE2, BIT)
*       WRITE(*,*)NEWIND,MATE1,MATE2,BIT

```

1 CONTINUE

RETURN
END

SUBROUTINE Replace(Psize,TSLENG,STRING,NSTRIN)

* Substituir a populacao antiga pela nova

IMPLICIT REAL (A-H,J-Z)

INTEGER Psize,TSLENG,BIT

INTEGER STRING(Psize, TSLENG)

INTEGER NSTRIN(Psize, TSLENG)

DO 1 INDIV = 1 , Psize

DO 2 BIT = 1 , TSLENG

STRING(INDIV, BIT) = NSTRIN(INDIV, BIT)

2 CONTINUE

1 CONTINUE

RETURN
END

SUBROUTINE Scaling(SCALEC,FITIND,SUMFIT,MEANF,FIT,Psize)

* Aplicar o escalonamento linear,

* $scaledfitness = a * fitness + b$.

* Sujeito a,

* $meanscaledfitness = meanfitness$

* e

* $bestscaledfitness = c * MEANF$,

* aonde c, e a constante de escalonamento definida pelo usuario.

IMPLICIT REAL (A-H,J-Z)

INTEGER FITIND,Psize

REAL FIT(Psize)

INTEGER INDIV

* Se a constante de escalonamento for zero, ou todos os individuos possuem o mesmo

* *fitness*, o escalonamento nao sera aplicado.

IF ((SCALEC.NE.0).AND.(FIT(FITIND) - MEANF.GT.0)) THEN

* Encontrar A e b.

$A = (SCALEC - 1) * MEANF / (FIT(FITIND) - MEANF)$

$B = (1 - a) * MEANF$

* Ajustar o *fitness* de todos os membros da populacao.

```

SUMFIT = 0
DO 1 INDIV = 1 , PSIZE
  FIT(INDIV) = A* FIT(INDIV) + B
*   ----
  IF (FIT(INDIV).LT.0) FIT(INDIV) = 0
*   Ajusta a soma de todos os fitnesses.
  SUMFIT = SUMFIT + FIT(INDIV)
1  CONTINUE

*   Ajustar a media de todos os fitnesses.
  MEANF = SUMFIT / PSIZE
ENDIF

RETURN
END

SUBROUTINE Selection(MATE,SUMFIT,PSIZE,FIT)
*   Selecionar um simples individuo pela selecao de fitness proportional.
IMPLICIT REAL (A-H,J-Z)
INTEGER MATE,PSIZE,I
REAL FIT(PSIZE)
write(*,*)'SELECT',PSIZE
SUM = 0
INDIV = 0

RWHEEL = RND()* SUMFIT
write(*,*)SUMFIT,RWHEEL
DO 1 I=1,PSIZE
  INDIV = INDIV + 1
  SUM = SUM + FIT(INDIV)
  WRITE(*,*)SUM,FIT(INDIV),INDIV
  IF (SUM.GE.RWHEEL) GO TO 2
1  CONTINUE

2  MATE = INDIV

RETURN
END

SUBROUTINE Statistics(MEANF,SUMFIT,FITIND,PSIZE,FIT,INTEGS,
+ STRING,ESTRIN,EINTEG,EUNKNO,NUNKNO,TSLENG,UNKNO,ELITE,EFIT)
*   Calcular a soma dos fitness atraves da populacao e encontrar o melhor individuo
*   e entao aplicar a ELITE se selecionado.
IMPLICIT REAL (A-H,J-Z)
INTEGER FITIND,PSIZE,NUNKNO,TSLENG,INDIV

```

```

REAL FIT(Psize)
INTEGER STRING(Psize, TSLENG)
INTEGER ESTRIN(TSLENG)
INTEGER EINTEG(NUNKNO)
CHARACTER*3 ELITE

FITIND = 0
MAXFIT = 0

DO 1 INDIV = 1 , Psize
  IF (FIT(INDIV).GT.MAXFIT) THEN
    MAXFIT = FIT(INDIV)
    FITIND = INDIV
  ENDIF
1 CONTINUE

IF (ELITE.EQ.'on ') THEN
  CALL ELITES (FITIND,Psize,NUNKNO,TSLENG,UNKNO,FIT,
+ INTEG,STRING,ESTRIN,EINTEG,EUNKNO,EFIT)
ENDIF

*   Soma dos fitness.
SUMFIT = 0
DO 2 INDIV = 1 , Psize
  SUMFIT = SUMFIT + FIT(INDIV)
2 CONTINUE

*   Encontrar a media dos fitness da populacao.
MEANF = SUMFIT / Psize

RETURN
END

FUNCTION RND()
*   Um simples gerador de numeros randomicos
*   adaptado do "FORTRAN 77" por D.M.Monro.

SAVE NEW,I

*   Mudar o ISEED a cada vez que se for realizar um estudo ira gerar uma
*   nova sequencia de numeros randomicos.

ISEED=15625

```

```
IF (I.EQ.0) NEW=ISEED
I=I+1
NEW=NEW*ISEED
NEW=MOD(NEW,16384)
IF (NEW.LT.0) NEW=16384+NEW

RND=NEW/16384.

RETURN
END
```

APÊNDICE E – Programa em APDL de um Gancho bidimensional

```
/UIS,MSGPOP,4
/NERR,,,-1
/NOPR

/PREP7

*SET,GERACAO , 0

*CFOPEN,Geracao,txt,E:\Mestrado\GA\Gancho
*VWRITE,GERACAO,,,,,,,,
(I1)
*CFCLOS

*DIME,return,,1,32
*VREAD,return(1,1,1),Resultados,txt,,IJK,1,32
(F10.2)

MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,2e11
MPDATA,PRXY,1,,0.3

K,1,return(1,1,1),return(1,2,1)*(-1),0,
K,2,return(1,3,1),return(1,4,1)*(-1),0,
K,3,return(1,5,1),return(1,6,1)*(-1),0,
K,4,return(1,7,1),return(1,8,1)*(-1),0,
K,5,return(1,9,1),return(1,10,1)*(-1),0,
K,6,return(1,11,1),return(1,12,1)*(-1),0,
K,7,return(1,13,1),return(1,12,1)*(-1),0,
K,8,return(1,15,1),return(1,16,1)*(-1),0,
K,9,return(1,17,1),return(1,18,1)*(-1),0,
K,10,return(1,19,1),return(1,20,1)*(-1),0,
K,11,return(1,21,1),return(1,22,1)*(-1),0,
K,12,return(1,23,1),return(1,24,1)*(-1),0,
K,13,return(1,25,1),return(1,26,1)*(-1),0,
K,14,return(1,27,1),return(1,28,1)*(-1),0,
K,15,return(1,29,1),return(1,30,1)*(-1),0,
K,16,return(1,31,1),return(1,32,1)*(-1),0,

kx = return(1,9,1) + ((return(1,11,1)-return(1,9,1))/2)
```

ky = return(1,10,1)*(-1) + ((return(1,11,1)-return(1,9,1))/2.2)*(-1)

K,17,kx,ky,0,

/REP,FAST

1,1,2

1,2,3

1,3,4

1,4,5

larc,5,6,17

1,6,7

1,7,8

1,8,9

1,9,10

1,10,11

1,11,12

1,12,13

1,13,14

1,14,15

1,15,16

1,16,1

A,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

ET,1,PLANE183

KEYOPT,1,3,0

KEYOPT,1,6,0

KEYOPT,1,10,0

SMRT,1

MSHAPE,0,2D

MSHKEY,0

CM,_Y,AREA

ASEL, , , , 1

CM,_Y1,AREA

CHKMSH,'AREA'

CMSEL,S,_Y

AMESH,_Y1

```
CMDELE,_Y
CMDELE,_Y1
CMDELE,_Y2
```

```
/SOL
```

```
LSEL,S,,,1
ALLSEL,BELOW,LINE
NSLL,S,1
D,ALL,ALL,
ALLSEL
```

```
LSEL,S,,,5
ALLSEL,BELOW,LINE
NSLL,S,1
*GET,numnod,NODE,0,COUNT
NODES= numnod
CARGA=-2000/NODES
F,all,FY,CARGA
ALLSEL
```

```
SOLVE
```

```
*GET,smax,SORT, ,MAX
*GET,area,AREA,1,AREA
```

```
/output,Stress,txt
*vwrite,smax
(F14.5)
/output
```

```
/output,Area,txt
*vwrite,area
(F10.2)
/output
```

```
GERACAO = GERACAO + 1
```

```
*CFOPEN,Geracao,txt,E:\Mestrado\GA\Gancho
*VWRITE,Geracao,,,,,,,,
(F3.1)
*CFCLOS
```

APÊNDICE F – Programa em APDL para a utilização do método por subproblemas – Estudo referente a um Gancho bidimensional

```
/UIS,MSGPOP,4  
/NERR,,,-1  
/NOPR
```

```
/PREP7
```

```
*SET,x2, 100  
*SET,x3, 100  
*SET,y3, 50  
*SET,x4, 100  
*SET,y4, 100  
*SET,y5, 150  
*SET,x6, 150  
*SET,y6, 150  
*SET,y7, 100  
*SET,x8, 250  
*SET,x9, 250  
*SET,y9, 150  
*SET,x10, 250  
*SET,y10, 200  
*SET,x11, 250  
*SET,y11, 250  
*SET,x12, 150  
*SET,y12, 250  
*SET,x13, 100  
*SET,y13, 250  
*SET,x14, 90  
*SET,y14, 250  
*SET,x15, 50  
*SET,y15, 250  
*SET,x16, 0  
*SET,y16, 200
```

```
MPTEMP,,,,,,,,  
MPTEMP,1,0  
MPDATA,EX,1,,2e11  
MPDATA,PRXY,1,,0.3
```

```
K,1,0,0,0,  
K,2,x2,0,0,
```

K,3,x3,y3*(-1),0,
K,4,x4,y4*(-1),0,
K,5,125,y5*(-1),0,
K,6,150,y5*(-1),0,
K,7,200,y7*(-1),0,
K,8,x8,y7*(-1),0,
K,9,x9,y9*(-1),0,
K,10,x10,y10*(-1),0,
K,11,x11,y11*(-1),0,
K,12,x12,y12*(-1),0,
K,13,x13,y13*(-1),0,
K,14,x14,y14*(-1),0,
K,15,x15,y15*(-1),0,
K,16,x16,y16*(-1),0,

/REP,FAST

1,1,2
1,2,3
1,3,4
1,4,5
1,5,6
1,6,7
1,7,8
1,8,9
1,9,10
1,10,11
1,11,12
1,12,13
1,13,14
1,14,15
1,15,16
1,16,1

A,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

ET,1,PLANE183
KEYOPT,1,3,0
KEYOPT,1,6,0
KEYOPT,1,10,0

```
SMRT,1
MSHAPE,0,2D
MSHKEY,0
```

```
CM,_Y,AREA
ASEL,,,1
CM,_Y1,AREA
CHKMSH,'AREA'
CMSEL,S,_Y
```

```
AMESH,_Y1
CMDELE,_Y
CMDELE,_Y1
CMDELE,_Y2
```

```
/SOL
```

```
LSEL,S,,,1
ALLSEL,BELOW,LINE
NSLL,S,1
D,ALL,ALL,
ALLSEL
```

```
LSEL,S,,,5
ALLSEL,BELOW,LINE
NSLL,S,1
*GET,numnod,NODE,0,COUNT
NODES= numnod
CARGA=-2000/NODES
F,all,FY,CARGA
ALLSEL
```

```
SOLVE
```

```
*GET,smax,SORT, ,MAX
*GET,area,AREA,1,AREA
```

```
/output,Stress,txt
*vwrite,smax
(F14.5)
```

/output

/output,Area,txt

*vwrite,area

(F10.2)

/output

/OPT

OPANL,'placacomfurosub','txt',' '

OPVAR,X2,DV,50,100, ,

OPVAR,X3,DV,85,115, ,

OPVAR,Y3,DV,45,55, ,

OPVAR,X4,DV,80,110, ,

OPVAR,Y4,DV,100,125, ,

OPVAR,Y5,DV,150,170, ,

OPVAR,Y7,DV,75,150, ,

OPVAR,X8,DV,240,260, ,

OPVAR,X9,DV,230,260, ,

OPVAR,Y9,DV,150,170, ,

OPVAR,X10,DV,230,260, ,

OPVAR,Y10,DV,200,220, ,

OPVAR,X11,DV,200,260, ,

OPVAR,Y11,DV,225,300, ,

OPVAR,X12,DV,150,180, ,

OPVAR,Y12,DV,225,300, ,

OPVAR,X13,DV,90,115, ,
OPVAR,Y13,DV,225,300, ,
OPVAR,X14,DV,70,90, ,
OPVAR,Y14,DV,225,300, ,
OPVAR,X15,DV,0,50, ,
OPVAR,Y15,DV,190,250, ,
OPVAR,X16,DV,0,50, ,
OPVAR,Y16,DV,90,200, ,
OPVAR,SMAX,SV, ,225, ,
OPVAR,AREA,OBJ, , , ,
OPTYPE,SUBP
OPSUBP,999,999,
OPEQN,0,0,0,1,0,