

APLICAÇÃO DE ALGORITMOS GENÉTICOS EM OTIMIZAÇÃO ESTRUTURAL

Afonso Celso de Castro Lemonge

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

Prof. Alvaro Luiz G. A. Coutinho, D.Sc.
(Presidente)

Prof. Helio José Corrêa Barbosa, D.Sc.

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. João Nisan Correia Guerreiro, D.Sc.

Prof. Raul Rosas e Silva, Ph.D.

Prof. Djalma Mosqueira Falcão, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 1999

**LEMONGE, AFONSO CELSO DE
CASTRO**

Aplicação de Algoritmos Genéticos em
Otimização Estrutural [Rio de Janeiro] 1999
XVI, 218p., 29,7cm (COPPE/UFRJ, D.
Sc., Engenharia Civil, 1999)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1 – Algoritmos Genéticos 2 – Otimização Es-
trutural 3 – Método dos Elementos Finitos

I. COPPE/UFRJ II. Título (Série)

Aos meus pais

Joaquim e Maria Ephygenia

Agradecimentos

Ao Prof. Helio Barbosa pela orientação, confiança, incentivo e amizade permanentes durante o desenvolvimento deste trabalho.

Ao Prof. Alvaro Coutinho pelo apoio, amizade e incentivo constantes durante todo tempo no Programa de Engenharia Civil da COPPE.

Aos amigos do LNCC pela convivência enriquecedora, apoio e incentivo constantes. Em especial ao Elson Toledo, João Nisan, Renato Simões, Eduardo Garcia e Cristiano, que contribuíram de diversas formas para este trabalho.

Aos amigos da Universidade Federal de Juiz de Fora. Em especial ao Vinicius Viana, Dário Furtado, Luis Paulo e Patrícia Hallack.

A todos os professores do Programa de Engenharia Civil da COPPE, pelo apoio recebido. Em especial aos Profs. Luiz Landau, José Alves e Fernando Ribeiro.

Ao LNCC, pelo apoio computacional e administrativo.

À CAPES, pela bolsa de doutorado concedida e à FAPEMIG pela concessão de auxílio para participação no IV World Congress on Computational Mechanics.

A todos que de forma direta ou indireta contribuíram para a realização deste trabalho.

À Maria Eugênia pelo carinho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D. Sc.)

APLICAÇÃO DE ALGORITMOS GENÉTICOS EM OTIMIZAÇÃO ESTRUTURAL

Afonso Celso de Castro Lemonge

Junho/1999

Orientadores: Alvaro L. G. A. Coutinho
Helio José Corrêa Barbosa

Programa: Engenharia Civil

Neste trabalho são discutidos, desenvolvidos e implementados algoritmos genéticos para resolver problemas de otimização estrutural. Mostra-se a potencialidade dos algoritmos para obtenção de soluções ótimas eficientes e sua flexibilização para ser aplicado em vários tipos de problemas de forma fácil.

São analisados problemas de minimização de peso, minimização de trabalho de deformação, reação máxima de apoio, otimização de parâmetros e topologia, em estruturas reticuladas planas e espaciais.

São utilizados dois tipos de algoritmos genéticos: um do tipo “steady-state” com codificação real e outro do tipo “geracional” com codificação binária.

Mostra-se a potencialidade dos algoritmos genéticos para analisar problemas variados com diversos objetivos e as facilidades de implementação sem necessidade de formulações ou reformulações extensas.

Utiliza-se o método dos elementos finitos na análise estrutural das soluções candidatas no processo de otimização.

São demonstradas as vantagens que os algoritmos genéticos apresentam sobre os métodos clássicos para o tratamento de alguns problemas de otimização estrutural.

São apresentados vários exemplos numéricos de aplicação dos algoritmos implementados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

APPLICATION OF GENETIC ALGORITHMS TO STRUCTURAL OPTIMIZATION

Afonso Celso de Castro Lemonge

June/1999

Advisors: Alvaro L. G. A. Coutinho
Helio José Corrêa Barbosa

Department: Civil Engineering

In this work genetic algorithms are discussed, developed and implemented to solve structural optimization problems. The potential of the algorithms to find optimal solutions is shown together with their flexibility in dealing with different types of problems.

Problems with weight minimization, deformation work minimization, maximum support reaction minimization, parameter and topology optimization, are analysed in space and planar framed structures.

Two types of genetic algorithm are used: a “steady-state” with real coding and a “generational” one with binary coding.

The potential of the algorithm to analyse several problems with different objectives is shown together with its ease of implementation, which avoids extensive formulations or reformulations.

The finite element method is used to solve the structural problems considered.

The advantages of the genetic algorithms as compared to classical methods are shown in several problems of structural optimization.

Numerical examples of the different genetic algorithms implemented are presented and discussed.

Índice

1	Introdução	1
1.1	Introdução	1
1.2	Motivação	2
1.3	Algoritmos genéticos e engenharia	4
1.4	Objetivos	4
1.5	Contribuições	5
1.6	Organização do texto	5
2	Otimização	6
2.1	Introdução	6
2.2	A formulação do problema de otimização	6
2.3	O algoritmo	7
2.3.1	Ingredientes necessários a um algoritmo de busca	7
2.3.2	Classificação	8
2.4	O problema	9
2.5	O usuário	11
2.6	Otimização estrutural	12
2.6.1	Projeto estrutural	12
2.6.2	Otimização estrutural	12
2.6.3	Breve histórico	14
3	Computação Evolucionista e Algoritmos Genéticos	16
3.1	Introdução	16
3.2	Computação Evolucionista	16
3.3	Algoritmos Genéticos	18
3.3.1	Codificação	20
3.3.2	Inicialização da população	22
3.3.3	Função aptidão	22
3.3.4	Seleção	23
3.3.5	Dois esquemas (algoritmos) de reprodução	25
3.3.6	Operadores genéticos	25
3.3.7	Operadores para codificação binária	26
3.3.8	Operadores para codificação real	27
3.3.9	Codificação real ou binária ?	32
4	Tópicos adicionais em Algoritmos Genéticos	33
4.1	Introdução	33
4.2	Morfogênese	33
4.2.1	Funções de desenvolvimento translativas	34

4.2.2	Funções de desenvolvimento generativas	34
4.2.3	Funções de desenvolvimento adaptativas	34
4.3	Problemas com restrições	35
4.3.1	Funções de penalização	36
4.3.2	Tratamento das restrições em várias etapas	38
4.3.3	Um algoritmo genético com dois parâmetros de penalização	38
4.3.4	Operadores especiais	39
4.4	Adaptação	39
4.4.1	Parametrização externa	39
4.4.2	Parametrização dinâmica	40
4.4.3	Adaptação do ambiente	41
4.4.4	Adaptação da população	41
4.4.5	Adaptação do indivíduo	41
4.4.6	Adaptação da componente	42
4.5	Diversidade da população	42
4.5.1	Diversidade genotípica	42
4.5.2	Diversidade fenotípica	43
4.6	Hibridização e busca local	43
4.6.1	Hibridização	43
4.6.2	Busca local	44
4.6.3	Lamarckianismo e Efeito Baldwin	46
4.7	Algoritmos Genéticos Paralelos	48
4.7.1	AG com paralelização do cálculo da aptidão	48
4.7.2	AG com modelo de “ilhas”	49
4.7.3	AG com <i>granulação fina</i>	49
4.7.4	AG misto	50
5	Experimentos Numéricos	52
5.1	Introdução	52
5.2	Treliça de 10 barras	52
5.2.1	Problema P1	53
5.2.2	Problema P2	57
5.2.3	Resultados comparativos - Problema P2	66
5.2.4	Problema P3	69
5.2.5	Resultados comparativos - Problema P3	70
5.2.6	Problema P4	71
5.2.7	Resultados comparativos - Problema P4	73
5.2.8	Problema P5	75
5.2.9	Resultados comparativos - Problemas P5	81
5.2.10	Problema P6	83
5.2.11	Resumo dos resultados	87
5.2.12	Comentários	88
5.3	Treliça de 38 barras	90
5.4	Treliça de 52 barras	95
5.5	Treliça de 18 barras	99
5.5.1	Relaxando a função de penalização	105
5.6	Treliça de 47 barras	107
5.6.1	Comentários	119

5.7	Otimização da estrutura de uma ponte	120
5.7.1	Introdução	120
5.7.2	A geometria	120
5.7.3	A função objetivo	121
5.7.4	O desenvolvimento do genótipo e avaliação da aptidão	121
5.7.5	Parametrização	127
5.7.6	Minimização da <i>compliance</i>	128
5.7.7	Minimização da reação máxima	134
5.7.8	Comentários	139
5.8	Estratégias para evolução de parâmetros	139
5.8.1	Introdução	139
5.8.2	Uma segunda estratégia	149
5.8.3	Uma terceira estratégia	155
5.8.4	Comentários	168
5.9	Estrutura de uma torre de transmissão	169
5.9.1	Comentários	182
5.10	Evoluindo a configuração de um projeto estrutural	182
5.10.1	Introdução	182
5.10.2	Características do projeto	183
5.10.3	Variáveis de projeto	183
5.10.4	Codificação do cromossomo	185
5.10.5	Resultados	186
5.10.6	Comentários	187

6 Conclusões **190**

Bibliografia **192**

Lista de Figuras

2.1	Algoritmo “Hill-Climbing” – exemplo de maximização de $f(x)$	9
2.2	Algoritmo “Simulated Annealing” – exemplo de maximização de $f(x)$	10
3.1	Algoritmo Evolucionista.	18
3.2	Algoritmo genético genérico.	20
3.3	Seleção proporcional à aptidão.	24
3.4	Algoritmo genético geracional.	25
3.5	Algoritmo genético “steady-state”.	26
3.6	Crossover para codificação real – Tabela reproduzida da referência [58].	31
4.1	Mapeamento translativo.	34
4.2	Mapeamento generativo.	35
4.3	Mapeamento adaptativo (a).	35
4.4	Mapeamento adaptativo (b).	36
4.5	Desenvolvimento do genótipo e Lamarckianismo.	47
4.6	Mapeamento inverso não possível.	47
4.7	AG paralelo do tipo “mestre-escravos”.	49
4.8	AG paralelo com granulação grossa.	50
4.9	AG paralelo com granulação fina.	50
4.10	Escolha do vizinho para reprodução – O quadrado central representa o indivíduo escolhido para reproduzir e o restante ao seu redor representam os possíveis elementos que podem participar da recombinação.	50
4.11	AG paralelo onde cada “ilha” tem aptidão avaliada em paralelo.	51
4.12	AG paralelo misto com modelo de “ilhas” associado a um AG com granulação fina.	51
5.1	Treliça de 10 barras.	52
5.2	Probabilidades iniciais e finais dos operadores.	54
5.3	População com 400 elementos e 20000 avaliações.	62
5.4	População com 400 elementos e 30000 avaliações.	62
5.5	População com 400 elementos e 40000 avaliações.	63
5.6	População com 500 elementos e 20000 avaliações.	63
5.7	População com 500 elementos e 30000 avaliações.	64
5.8	População com 500 elementos e 40000 avaliações.	64
5.9	População com 600 elementos e 21000 avaliações.	65
5.10	População com 600 elementos e 30000 avaliações.	65
5.11	População com 600 elementos e 42000 avaliações.	66
5.12	Problema P3 - somente restrições de tensões.	71
5.13	Problema P4.	75
5.14	Caso I.	81

5.15	Caso II.	81
5.16	Caso III.	82
5.17	Caso discreto com consideração de peso próprio.	84
5.18	Caso discreto – cargas nos nós 2 e 4 – cargas em Kips $\times 10^3$	85
5.19	Caso contínuo com consideração de peso próprio.	87
5.20	Caso contínuo – cargas nos nós 2 e 4 – cargas em Kips $\times 10^3$	88
5.21	Treliça de 38 barras.	90
5.22	Função de penalização adotada no Problema da Treliça de 38 barras.	91
5.23	Caso discreto.	94
5.24	Caso contínuo.	94
5.25	Treliça de 52 barras.	95
5.26	Função aptidão do melhor elemento da população em 10 execuções.	98
5.27	Treliça de 18 barras.	99
5.28	Função aptidão do melhor elemento da população em 10 execuções.	104
5.29	Forma ótima - (a) contínuo (tensões normais); (b) misto (tensões normais); (c) contínuo (tensões normais e de Euler); (d) misto (tensões normais e de Euler).	104
5.30	Peso do melhor elemento da população em 10 execuções.	106
5.31	Treliça de 47 barras.	108
5.32	Forma ótima - (a) contínuo (tensões normais); (b) misto (tensões normais); (c) contínuo (tensões normais e de Euler); (d) misto (tensões normais e de Euler).	118
5.33	Peso do melhor elemento da população em 10 execuções.	119
5.34	A estrutura da ponte.	120
5.35	Morfogênese: do genótipo para o fenótipo.	122
5.36	Modelo com apoios rígidos.	123
5.37	Modelo com apoios elásticos.	126
5.38	Modelo para determinação das constantes elásticas do pilar + aparelho de apoio.	127
5.39	Pista com carga uniformemente distribuída.	127
5.40	Pista com carga do veículo padrão.	128
5.41	Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.	131
5.42	Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.	132
5.43	Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.	136
5.44	Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.	137
5.45	O algoritmo de Gauss-Seidel com relaxação – GSR.	140
5.46	O algoritmo para o processo de co-evolução.	142
5.47	Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução.	146
5.48	População de ω no processo de evolução.	146
5.49	Melhor ω da população durante a evolução.	147
5.50	Aptidão do melhor ω ao longo da evolução.	147
5.51	Peso da treliça de 10 barras.	148

5.52	Número de iterações no GSR para $\omega \in [1.0, 1.95]$ referente ao sistema de equações da estrutura final otimizada.	148
5.53	Segunda estratégia – Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução.	152
5.54	Segunda estratégia – População de ω no processo de evolução.	152
5.55	Segunda estratégia – Melhor ω da população durante a evolução.	153
5.56	Segunda estratégia – Aptidão do melhor ω ao longo da evolução.	153
5.57	Segunda estratégia – Peso da treliça de 10 barras.	154
5.58	Segunda estratégia – Sistema de equações da estrutura final otimizada resolvido com vários valores de ω	154
5.59	Algoritmo que controla a geração de novos filhos – Terceira estratégia.	155
5.60	Algoritmo para a determinação da função aptidão e controle da política de inserção – Terceira estratégia.	156
5.61	Terceira estratégia – Distribuição de valores de ω e seu respectivos números de iterações ao longo da evolução par $p_{new} = 0.002$ e $p_{mut} = 1.0$	158
5.62	Terceira estratégia – População de ω no processo de evolução – $p_{new} = 0.002$ e $p_{mut} = 1.0$	158
5.63	Terceira estratégia – Melhor ω da população durante a evolução – $p_{new} = 0.002$ e $p_{mut} = 1.0$	159
5.64	Terceira estratégia – Aptidão do melhor ω ao longo da evolução – $p_{new} = 0.002$ e $p_{mut} = 1.0$	159
5.65	Terceira estratégia – Peso da treliça de 10 barras – $p_{new} = 0.002$ e $p_{mut} = 1.0$	160
5.66	Terceira estratégia – Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$	160
5.67	Terceira estratégia – População de ω no processo de evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$	161
5.68	Terceira estratégia – Melhor ω da população durante a evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$	161
5.69	Terceira estratégia – Aptidão do melhor ω ao longo da evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$	162
5.70	Terceira estratégia – Peso da treliça de 10 barras – $p_{new} = 0.5$ e $p_{mut} = 0.1$	162
5.71	Cromossomo para o caso discreto – $n=38$	170
5.72	Cromossomo para o caso contínuo – $n=38$	171
5.73	Torre de transmissão com 160 barras.	172
5.74	Módulo componente da treliça.	183
5.75	Áreas de influências para determinação de carregamento devido a sobrecarga.	184
5.76	Variáveis de forma.	184
5.77	Função aptidão.	188
5.78	Forma final do módulo padrão.	189

Lista de Tabelas

3.1	Código binário usual e Código de Gray.	22
3.2	Seleção proporcional à aptidão – $F_T = 148$, F_{pi} = soma parcial das aptidões.	24
3.3	Número aleatório n e elemento selecionado.	24
5.1	Propriedades do material e restrições.	53
5.2	Probabilidades iniciais e finais dos operadores.	54
5.3	Resultados obtidos na otimização – peso da estrutura em lbs.	55
5.4	Estatística dos operadores - população = 600, avaliações = 42000.	55
5.5	Resultados comparativos do Problema P1.	56
5.6	Resultados obtidos na otimização - prob.=0.1 – peso da estrutura em lbs.	57
5.7	Resultados obtidos na otimização - prob.=0.3 – peso da estrutura em lbs.	58
5.8	Resultados obtidos na otimização - prob.=1.0 – peso da estrutura em lbs.	58
5.9	Resultados comparativos do Problema P2 - prob.=0.1.	59
5.10	Resultados comparativos do Problema P2 - prob.=0.3.	60
5.11	Resultados comparativos do Problema P2 - prob.=1.0.	61
5.12	Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.	67
5.13	Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.	67
5.14	Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.	68
5.15	Pesos apresentados e calculados e deslocamentos correspondentes às soluções apresentadas nas referências e neste trabalho.	69
5.16	Resultados obtidos na otimização – peso da estrutura em lbs.	69
5.17	Resultados comparativos entre a literatura e a melhor solução do Problema P3.	70
5.18	Resultados comparativos entre a literatura e a melhor solução do Problema P3.	70
5.19	Problema P3 - tensões nas barras.	71
5.20	Resultados comparativos do Problema P3.	72
5.21	Probabilidades iniciais e finais dos operadores.	73
5.22	Resultados obtidos na otimização – peso da estrutura em lbs.	73
5.23	Resultados comparativos entre a literatura e a melhor solução do Problema P4.	74

5.24	Resultados comparativos entre a literatura e a melhor solução do Problema P4.	74
5.25	Problema P4 - Pesos apresentados e calculados e deslocamentos correspondentes às soluções apresentadas.	75
5.26	Resultados comparativos do Problema P4.	76
5.27	Intervalos que definem sub-conjuntos de \mathcal{S} como possibilidades de escolha para as variáveis de projeto do Caso I.	77
5.28	Conjunto \mathcal{S} de seções transversais disponíveis.	77
5.29	Resultados obtidos na otimização - Caso I – peso da estrutura em lbs.	78
5.30	Resultados comparativos do Problema P5 - Caso I.	79
5.31	Resultados comparativos do Problema P5 - Caso I.	80
5.32	Resultados comparativos entre a literatura e a melhor solução do Problema P5.	82
5.33	Problema P5 - Pesos apresentados e calculados e deslocamentos correspondentes às soluções encontradas na literatura e obtidas nesta tese.	83
5.34	Caso discreto com peso próprio – peso da estrutura em lbs.	83
5.35	Cargas finais nos nós (lbs) – caso discreto.	84
5.36	Resultados obtidos na otimização – peso em lbs.	85
5.37	Cargas finais nos nós (lbs) – caso contínuo.	85
5.38	Caso contínuo com peso próprio.	86
5.39	Características dos Problemas.	87
5.40	Características dos Problemas.	88
5.41	Melhores resultados para a treliça de 10 barras - caso contínuo.	89
5.42	Melhores resultados para a treliça de 10 barras - caso discreto.	89
5.43	Propriedades do material e restrições.	91
5.44	Solução para 10 populações iniciais diferentes - peso em kgs.	92
5.45	Probabilidades iniciais e finais dos operadores.	93
5.46	Resultados comparativos - variáveis contínuas - $A_i(\text{mm}^2 \times 10^3)$	93
5.47	Dados de carregamento - cargas em kN.	96
5.48	Propriedades do material.	96
5.49	Seções transversais disponíveis.	97
5.50	Variáveis de projeto da melhor solução em 10 execuções.	98
5.51	Dados de carregamento - cargas em lb.	100
5.52	Propriedades do material e restrições.	100
5.53	Variáveis de forma.	100
5.54	Probabilidades dos operadores.	100
5.55	Variáveis de projeto da melhor solução para os casos contínuo e misto (sem restrições de Euler).	101
5.56	Tensões normais nas barras (sem restrições de Euler).	102
5.57	Variáveis de projeto da melhor solução para os casos contínuo e misto (com restrições de Euler).	102
5.58	Tensões normais e tensões de Euler - caso contínuo.	103
5.59	Tensões normais e de Euler - caso misto.	103
5.60	Variáveis de projeto da melhor solução para o caso misto (sem restrições de Euler) – Flexibilização das restrições em 2% e 5%.	105
5.61	Tensões normais nas barras – caso misto (sem restrições de Euler).	106
5.62	Dados de carregamento - cargas em lb.	107

5.63	Propriedades do material e restrições.	107
5.64	Variáveis de forma.	109
5.65	Probabilidades dos operadores.	109
5.66	Variáveis de projeto da melhor solução para o casos contínuo e misto (sem restrições de Euler).	110
5.67	Tensões normais nas barras - caso contínuo (sem restrições de Euler).	111
5.68	Tensões normais nas barras - caso misto (sem restrições de Euler).	112
5.69	Variáveis de projeto da melhor solução para os casos contínuo e misto (com restrições de Euler).	113
5.70	Tensões normais e de Euler – caso contínuo.	114
5.71	Tensões normais e de Euler – caso contínuo – Ref [125].	115
5.72	Tensões normais e de Euler – caso contínuo – Ref [126].	116
5.73	Tensões normais e de Euler – caso misto.	117
5.74	Variáveis de projeto para minimização da <i>compliance</i>	129
5.75	Vãos obtidos pela minimização da <i>compliance</i>	129
5.76	Dados dos pilares para minimização da <i>compliance</i> para todos critérios.	130
5.77	Dimensões dos aparelhos de apoio obtidas pela minimização da <i>compliance</i>	133
5.78	Resumo das análises para o caso de minimização da <i>compliance</i>	133
5.79	Variáveis de projeto para minimização da reação máxima.	134
5.80	Vãos obtidos pela minimização da reação máxima.	134
5.81	Dados dos pilares para minimização da reação máxima.	135
5.82	Dimensões dos aparelhos de apoio obtidas pela minimização da reação máxima.	138
5.83	Resumo das análises para o caso de minimização da reação máxima.	138
5.84	Probabilidades dos operadores para a população de estruturas.	141
5.85	Valores das melhores soluções em 10 execuções independentes.	143
5.86	Valores médios em 10 execuções.	143
5.87	Valores constantes de ω e o correspondente total de iterações para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 500.	144
5.88	Valores constantes de ω , número médio de iterações por resolução de sistema e número total de iterações médio em 10 execuções para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-9}$ e limite = 500.	144
5.89	Segunda estratégia – Valores das melhores soluções em 10 execuções independentes.	149
5.90	Segunda estratégia – Valores médios de 10 execuções.	150
5.91	Segunda estratégia – Valores constantes de ω e número total de ite- rações para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 1000.	150
5.92	Segunda estratégia – Valores constantes de ω , número médio de ite- rações e número total de iterações médio – Valores médios em 10 execuções – $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-7}$ e limite = 1000.	151
5.93	Terceira estratégia – Valores de ω variando com p_{new} e p_{mut}	157
5.94	Terceira estratégia – Valores médios de ω variando com p_{new} e p_{mut}	157
5.95	Terceira estratégia – Número de iterações da melhor solução variando com p_{new} e p_{mut}	163
5.96	Terceira estratégia – Número de iterações do melhor indivíduo va- riando com p_{new} e p_{mut} - Valores médios de 10 execuções.	163

5.97 Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} .	164
5.98 Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} - Valores médios de 10 execuções.	164
5.99 Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} .	165
5.100 Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} – Valores médios de 10 execuções.	165
5.101 Terceira estratégia – Valores finais médios de ω variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.	166
5.102 Terceira estratégia – Número de iterações do melhor indivíduo variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.	166
5.103 Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.	167
5.104 Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.	167
5.105 Dados de carregamento - cargas em kg.	173
5.106 Conectividades dos elementos e variáveis de projeto.	174
5.107 Seções transversais disponíveis para o caso discreto – cm^2 .	175
5.108 Propriedades do material e restrições.	175
5.109 Coordenadas dos nós – cm.	176
5.110 Codificação do cromossomo – caso discreto.	176
5.111 Codificação do cromossomo – caso contínuo.	176
5.112 Peso obtido: caso discreto (kg).	177
5.113 Peso obtido: caso contínuo (kg).	177
5.114 Seções transversais selecionadas – caso discreto.	177
5.115 Seções transversais selecionadas – caso contínuo.	177
5.116 Designação das seções transversais escolhidas para as áreas das barras – caso discreto.	178
5.117 Designação das seções transversais escolhidas para as áreas das barras – caso contínuo.	179
5.118 Melhores execuções – caso discreto.	180
5.119 Melhores execuções – caso contínuo.	180
5.120 Média – caso discreto.	181
5.121 Média – caso contínuo.	181
5.122 Propriedades do material.	183
5.123 Seções transversais disponíveis – cm^2 .	185
5.124 Variáveis de projeto e suas designações.	186
5.125 Codificação e limites para as variáveis de projeto.	187
5.126 Variáveis de projeto obtidas na otimização.	188

Capítulo 1

Introdução

1.1 Introdução

A competição entre os seres vivos induz a busca de condições ideais de manutenção da vida de cada um deles no ambiente em que vivem. Esta manutenção, também conhecida como aptidão à sobrevivência, é representada por uma série de fatores complexos. Os seres vivos podem ser comparados a máquinas complicadas e extremamente delicadas que são dinamicamente submetidas a transformações nos seus componentes (órgãos) com o intuito de melhorar a sua eficiência, ou seja, sua capacidade de sobrevivência. Mas, como devem ser feitas as transformações para que se tenha uma maior eficiência? Esta resposta não é trivial e, ao contrário, de extrema complexidade pois, nem sempre grandes transformações induzem uma melhora na eficiência e algumas vezes, pequenas transformações feitas aleatoriamente podem provocar sensíveis melhoras.

As habilidades adquiridas pelos seres vivos durante o processo de desenvolvimento ou maturação no meio em que vivem, não podem ser transmitidos para as gerações seguintes. O que é possível é que certas habilidades podem tornar-se essenciais à sobrevivência de certas espécies ao longo dos tempos, e assim, os indivíduos que as possuem estarão mais aptos a se manterem no ambiente e conseqüentemente com maiores chances de seleção para reprodução.

As anomalias provocadas por mutações não felizes provocam a diminuição da eficiência, comprometendo a aptidão à sobrevivência de alguns seres vivos. Mudanças radicais não ocorrem de forma repentina nem existe perda de eficiência dos seres durante a evolução, ou seja, as gerações seguintes devem ser mais eficientes que as anteriores, dado que não existam grandes transformações no meio ambiente.

Segundo a teoria da evolução de Charles Darwin [1], qualquer ser vivo que tenha sido bem sucedido, com eficiência comprovada, a ponto de chegar na idade de se tornar genitor, deve ser considerado bom.

O processo de evolução intrinsecamente ligado ao código genético (genoma) de cada ser vivo, não é ainda totalmente conhecido. As pesquisas são intensas nessa área, porém algumas certezas já existem [2]:

- Evolução é um processo que atua sobre uma população ou uma espécie;
- Seleção natural é o processo que acarreta uma reprodução mais acentuada daqueles indivíduos mais bem adaptados (de melhor desempenho no seu meio-ambiente) e a correspondente difusão de seu material genético na população;

- O processo de reprodução é o ponto principal da evolução. O processo de recombinação pode criar cromossomos bem diferentes para a prole a partir do cromossomo paterno e materno e mutações podem introduzir pequenas modificações adicionais;
- Evolução biológica não tem memória nem planejamento. Não existe transmissão de conhecimento adquirido entre pais e filhos.

Na década de 70 o pesquisador John Holland [3, 4] da Universidade de Michigan atraído por questões envolvidas no processo de seleção natural das espécies, desenvolveu uma técnica, implementada computacionalmente através de um algoritmo, que batizou de Algoritmo Genético, com o objetivo de resolver problemas complexos espelhando-se no processo de evolução dos seres vivos.

Algoritmos genéticos são procedimentos de busca baseados no processo de seleção natural das espécies [5, 6, 7, 2, 8]. São algoritmos probabilísticos com forte aplicação em problemas de otimização nas mais diversas áreas. São considerados algoritmos robustos por serem capazes de ser aplicados a vários tipos de problemas. Sua difusão ocorre de forma rápida e sua popularidade se dá, também, pelo fato de resolver problemas complexos das mais distintas áreas como finanças, medicina, biologia, engenharia, etc, ainda não resolvidos por outras técnicas.

1.2 Motivação

A motivação em desenvolver pesquisa em aplicação de algoritmos genéticos em otimização estrutural pode ser justificada pelos seguintes aspectos:

1. Métodos clássicos apresentam dificuldades de identificar soluções ótimas globais, em geral;
2. Métodos clássicos apresentam dificuldades quando o problema envolve variáveis contínuas e discretas - problemas típicos em engenharia;
3. Métodos clássicos, em geral, não são eficazes quando a otimização tem multi-objetivos;
4. Métodos clássicos não são indicados para programação em códigos paralelos;
5. Cada método clássico, em geral, tem domínio de aplicação restrito;

Geralmente, em grande parte dos problemas as funções objetivo são multi-modais contendo ótimos locais e ótimos globais. Logo, achar a solução ótima global se torna essencial para que os objetivos requeridos sejam alcançados da melhor forma possível. Esta é uma característica vital em engenharia que está estritamente ligada à obtenção das melhores soluções (ótimos globais), que sejam eficientes e baratas. Muitos dos métodos clássicos apresentam dificuldades para identificação de soluções ótimas globais restringindo-se apenas a descobrir soluções ótimas locais.

Para grande parte dos problemas de otimização em engenharia os métodos clássicos também não são muito eficazes já que, em geral, as funções objetivo podem ser não-lineares e não-diferenciáveis que são condições necessárias para utilização dos mesmos e, dessa forma, dificultando suas aplicações.

É frequente em problemas de otimização estrutural, a utilização de espaços de busca com variáveis discretas. Por exemplo, ao se tentar otimizar as áreas das barras de uma treliça o problema fica restrito às barras encontradas em tabelas fornecidas pelos fabricantes. O espaço de busca é discreto criando dificuldades para aplicação dos métodos clássicos. Esta situação é contornada assumindo-se um espaço de busca com variáveis contínuas que, após a otimização, fornecerão uma aproximação das variáveis de projeto para as disponíveis no espaço discreto. Todavia, torna-se uma tarefa difícil a escolha das variáveis discretas mais próximas das contínuas encontradas, pois existirão duas opções de variáveis discretas para cada variável contínua, ou seja, uma imediatamente anterior e outra imediatamente posterior e se o problema tem n variáveis serão 2^n possibilidades de escolha.

Diversos problemas em engenharia envolvem também otimização com multi-objetivos a serem satisfeitos simultaneamente [9, 10]. É comum na elaboração de um projeto se requerer condições de economia, segurança e eficiência, aspectos às vezes conflitantes para serem plenamente satisfeitos em uma única solução. Geralmente, nos métodos clássicos, contorna-se esta situação criando-se artificialmente uma única função objetivo que pondera os múltiplos objetivos. Este tipo de procedimento é fortemente subjetivo, devido à introdução de uma ponderação, que interfere diretamente na busca da solução ótima. Como o algoritmo genético trabalha com uma população de soluções, eles são indicados para resolver problemas de otimização multi-objetivos. Por modificações adequadas no algoritmo é possível identificar simultaneamente na população final de soluções as soluções que melhor satisfazem aos múltiplos objetivos.

As formulações e modelagens dos problemas em engenharia estão se tornando mais complexas e a avaliação de uma possível solução pode demandar alto custo computacional. Para tanto, uma questão interessante é a paralelização dos códigos computacionais em problemas de otimização. Visto que a maioria dos métodos clássicos utilizam algoritmos iterativos onde a solução do passo seguinte depende da solução do passo anterior, encontram-se dificuldades quanto à utilização plena de recursos oferecidos pela computação paralela. A computação paralela está se tornando uma ferramenta bastante popular e decisiva na eficiência de vários códigos computacionais e os algoritmos genéticos podem utilizá-la sem nenhuma restrição.

Alguns aspectos interessantes dos algoritmos genéticos podem ser resumidos em:

- Algoritmos genéticos não requerem que as funções objetivo sejam contínuas ou diferenciáveis;
- Algoritmos genéticos, em geral, não requerem formulações ou reformulações complexas para os problemas;
- Algoritmos genéticos não requerem estudos preliminares para definição de pontos de partida, factíveis ou não, no espaço de busca;
- Algoritmos genéticos são de natureza intrinsecamente paralela.
- Algoritmos genéticos proporcionam grande flexibilidade na definição da função objetivo que pode ser modificada facilmente sem a necessidade de recodificações extensas.

1.3 Algoritmos genéticos e engenharia

A difusão dos algoritmos genéticos como ferramenta em problemas de otimização, ocorrida a partir de meados da década de oitenta, despertou o interesse em pesquisadores da área de engenharia e ciências afins. Eles têm sido aplicados em uma variedade de problemas e frequentemente em problemas de otimização. Inicialmente, foram utilizados em análises de projetos de turbinas de aeronaves e, mais recentemente, em análises de estruturas de proteínas. Já no ano de 1989 [7] pode-se encontrar uma série de aplicações em diversas áreas.

Em otimização estrutural dois trabalhos recebem destaque na literatura e estão entre os pioneiros desta área: 1) *Projeto Ótimo de uma treliça de 10 barras* (Goldberg e Samtani-1986) [11] e 2) *Projetos Ótimos de Ligações Soldadas* (Deb 1990) [12]. Estes trabalhos mostram a capacidade de utilização dos algoritmos genéticos em problemas de otimização estrutural. Neles os problemas foram inicialmente estudados através de técnicas de programação matemática e depois transformados em problemas sem restrições, incorporando funções de penalização nas funções objetivo e, então, analisados via algoritmos genéticos. A comprovação da robustez dos algoritmos genéticos fez com que o número de aplicações crescesse em todas as direções, inclusive, na direção de problemas de otimização estrutural.

Uma bibliografia contendo aproximadamente 4300 referências sobre algoritmos genéticos pode ser obtida em [13]. Algumas aplicações, entre tantas, em engenharia, são listadas a seguir [14, 15, 16, 17, 18, 19, 20, 21]:

- Otimização de forma de elementos estruturais;
- Otimização de topologia;
- Localização ótima de sensores e atuadores;
- Detecção, localização e quantificação de danos em estruturas;
- Problemas de identificação e problemas inversos;
- Concepção de projetos;
- Problemas de locação/alocação;
- Problemas que envolvem sequenciamento.

1.4 Objetivos

O objetivo desta tese é desenvolver, implementar e integrar algoritmos genéticos e métodos de análise estrutural com o intuito de resolver problemas de otimização estrutural. Mostrar, também, a robustez dos algoritmos genéticos e como podem resolver problemas que apresentam dificuldades significativas quando analisados por métodos clássicos.

Na maioria dos problemas apresentados neste texto, utiliza-se um módulo genérico de elementos finitos [22, 23, 24] destinado à análise estrutural. Em outros casos são desenvolvidos e implementados módulos (também em elementos finitos), que incorporam características particulares do problema em discussão.

São utilizados elementos de barra (treliça plana e espacial), e todas as análises são lineares.

1.5 Contribuições

- Utilização de conhecimentos específicos já adquiridos pelo projetista sobre o problema em questão como informação adicionada ao algoritmo genético para tornar a busca mais rápida e eficiente;
- Solução de problemas com variáveis contínuas e discretas tanto através de algoritmos genéticos com codificação binária quanto com codificação real;
- Discussão de estratégias de co-evolução de parâmetros de resolução em problemas de otimização.
- Desenvolvimento e implementação de estratégias para obtenção de solução de problemas de interesse prático em engenharia.
- Utilização de algoritmos genéticos na busca de soluções ótimas de concepções de projetos estruturais.
- Discussão de exemplos clássicos de otimização estrutural mostrando a capacidade dos algoritmos genéticos na obtenção de melhores soluções verificando-se a factibilidade de resultados encontrados na literatura;

1.6 Organização do texto

O Capítulo 2 é destinado à otimização em um aspecto mais genérico abordando a classificação dos algoritmos, o tratamento dos problemas e o papel do usuário. Introduce-se o problema de otimização estrutural, apresentando-se um breve histórico e a formulação do problema.

No Capítulo 3 encontram-se informações sobre as origens da computação evolucionista com trabalhos relevantes encontrados na literatura e a introdução aos algoritmos genéticos. Destacam-se os seus pontos principais, como: codificação, inicialização da população, função aptidão, tipos de algoritmos genéticos, codificação e operadores.

No Capítulo 4 são discutidos tópicos adicionais em algoritmos genéticos destacando-se o processo de morfogênese, problemas com restrições, busca local (lamarckianismo e efeito Baldwin), diversidade da população e algoritmos genéticos paralelos.

No Capítulo 5 encontram-se os experimentos numéricos realizados neste trabalho e, finalmente, as conclusões desta tese são sumarizadas no Capítulo 6, onde possíveis desenvolvimentos adicionais também são propostos.

Capítulo 2

Otimização

2.1 Introdução

Um problema de otimização **PO** está sempre associado a um problema de minimização ou maximização de uma ou mais funções. No caso de minimização o **PO** pode ser colocado da seguinte forma:

$$\text{minimize } W = f(x_1, x_2, x_3, \dots, x_n)$$

onde a minimização da função $f(x)$, $x \in \mathbb{R}^n$, é equivalente à maximização da função $g(x)$, onde $g(x) = -f(x)$. Assim, tem-se:

$$\text{mínimo } f(x) = \text{máximo } g(x) = \text{máximo } \{-f(x)\}$$

As funções $f(x)$ e $g(x)$ são conhecidas como *funções objetivo* e representam a quantidade que se deseja minimizar ou maximizar, respectivamente.

2.2 A formulação do problema de otimização

O problema de otimização pode ser formulado matematicamente da seguinte forma:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{submetido a } \begin{cases} g_i(x) \leq 0 & \text{para } i = 1, \dots, m \\ h_i(x) = 0 & \text{para } i = 1, \dots, l \\ x \in X \subset \mathbb{R}^n \\ X = \{x \in \mathbb{R}^n : x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, n\} \end{cases} \end{aligned}$$

O vetor x é designado como o vetor de incógnitas ou vetor de variáveis de projeto, $f(x)$ é a função objetivo e $g_i(x)$ e $h_i(x)$ são as restrições de desigualdade e igualdade, respectivamente, e estas podem ser funções lineares ou não lineares do vetor de variáveis de projeto. A otimização com restrições é mais complexa e pode requerer estratégias específicas na formulação do problema para que essas sejam satisfeitas.

O conjunto X é um paralelepípedo em \mathbb{R}^n definido pelos limites inferior e superior pré-estabelecidos para as variáveis de projeto. Um vetor $x \in X$ que satisfaz a todas essas restrições é chamado de uma *solução factível* do problema. O conjunto de todas as soluções factíveis é chamado de *região factível*. A solução do problema

de minimização é dita *solução ótima* e existindo mais de uma solução ótima estas são ditas *soluções ótimas alternativas*.

O conjunto de variáveis de projeto que proporciona o menor valor da função objetivo – no caso de minimização – entre todas as combinações possíveis entre os valores das variáveis, é chamado de mínimo global. Em geral, é difícil afirmar que tal valor é global devido à possibilidade de existência de vários mínimos locais e, assim, somente um deles será o global. O que pode-se afirmar é que o valor encontrado é mínimo numa vizinhança do espaço de busca.

Ao se resolver um problema de otimização são necessários três elementos atuantes no processo:

- O algoritmo
- O problema
- O usuário

2.3 O algoritmo

2.3.1 Ingredientes necessários a um algoritmo de busca

Segundo [25] um algoritmo de busca clássico, numérico ou evolucionário deve conter alguns dos seguintes ingredientes:

- Uma **representação** para as soluções candidatas ao problema;
- Um procedimento de **inicialização** que gera as soluções candidatas;
- Um procedimento de **avaliação da qualidade** de cada solução candidata;
- Uma **estrutura de memória** contendo a lista de soluções candidatas;
- Uma **história** contendo as soluções anteriormente testadas;
- Um **critério de comparação** entre as soluções candidatas para identificar qual a melhor;
- Um procedimento de **gradiente de qualidade** para indicar direções em que serão obtidas melhores soluções;
- Um conjunto de **operadores** que atuam sobre uma ou mais soluções gerando novas soluções;
- Uma estratégia de **controle** para a aplicação dos operadores.
- Um procedimento de **seleção** que determina quais soluções devem ser operadas;
- Um procedimento de **filtro** ou **reparo** que elimina ou corrige soluções inválidas;
- Um procedimento de **gerência de memória** que seleciona soluções a serem adicionadas ou removidas da memória.
- Um **critério de parada** pelo qual a busca é interrompida;

2.3.2 Classificação

Existem várias maneiras de se classificar os algoritmos de otimização a partir de diferentes pontos de vista. Uma possível classificação é a que se baseia na natureza determinística ou não-determinística dos algoritmos de busca [26].

Dessa forma os algoritmos podem ser classificados como:

- Determinísticos
- Probabilísticos
 - Algoritmos de busca *não informada*
 - Algoritmos de busca *informada*

Determinísticos

Quando qualquer decisão a ser tomada dentro do algoritmo obedece a algum tipo de regra de forma rigorosa não havendo oportunidade para alguma decisão de caráter (pseudo)-aleatório. Um exemplo simples é a ordenação de membros de uma lista que obedece a uma regra pré-estabelecida.

Probabilísticos

Quando são introduzidos elementos (pseudo)-aleatórios dentro do algoritmo com o objetivo de testar novas direções para a busca de novas soluções, diminuindo o aspecto *rigoroso* de um algoritmo determinístico. Essa é uma característica atraente para várias aplicações práticas. Dentro da classe de algoritmos probabilísticos existem os de busca *não informada* e os de busca *informada*.

Algoritmos probabilísticos com busca *não informada*

Talvez sejam esses os mais simples dentro da classe de algoritmos probabilísticos. Neste caso, a única informação disponível durante a busca é a resposta verdadeiro ou falso, à pergunta: é esta a solução do problema?

Algoritmos probabilísticos com busca *informada*

Esses algoritmos têm como característica a obtenção de informações colhidas a partir das regiões do espaço de busca, qualificando-as e até mesmo sugerindo uma ordem de exploração das mesmas. Entre eles encontram-se os algoritmos do “Montanhista” (“Hill-Climbing”), “Recozimento Simulado” (“Simulated Annealing”) e Algoritmos Evolucionistas.

- Algoritmo “Hill-Climbing” – Figura 2.1: é um tipo de algoritmo que tem como característica a escolha aleatória de uma solução de partida no espaço de busca gerando soluções na sua vizinhança de tal forma que qualquer nova solução gerada poderá substituir a atual se apresentar uma aptidão melhor para o problema.
- Algoritmo “Simulated Annealing” – Figura 2.2: tem sido bastante utilizado a partir da década de 80. É inspirado no fenômeno físico do comportamento de

moléculas durante o processo de cristalização. É similar ao algoritmo “Hill-Climbing” com a diferença de que pode haver substituição de uma solução melhor por uma pior, com probabilidade decrescente ao longo do processo, com o intuito de possibilitar o afastamento de um possível mínimo ou máximo local.

- Algoritmos Evolucionistas: são baseados em estratégias inspiradas pelo processo de evolução das espécies em biologia.

Algoritmo “Hill-Climbing”

Início

$t \leftarrow 0$

Repita

$local \leftarrow FALSE$

selecionar uma solução candidata x_k aleatoriamente

avaliar $f(x_k)$

Repita

selecionar n soluções candidatas na vizinhança de x_k

selecionar x_m que retorna o maior valor para $f(x)$

se $f(x_k) < f(x_m)$ **faça**

$x_k \leftarrow x_m$

senão

$local \leftarrow VERDADEIRO$

fim-se

Até $local$

$t \leftarrow t + 1$

Até $t =$ número máximo de iterações

Fim

Figura 2.1: Algoritmo “Hill-Climbing” – exemplo de maximização de $f(x)$.

2.4 O problema

Cada vez mais cresce a complexidade dos problemas que se colocam para serem resolvidos nos campos da ciência e engenharia. Pesquisadores convivem constantemente com os desafios e na área de otimização eles podem ser resumidos nos seguintes itens [27]:

1. O tempo necessário para avaliar uma única função é grande. Logo, a minimização do número de avaliações da função é vital;
2. Derivadas analíticas (em relação aos parâmetros), da função objetivo e das restrições, podem ser caras ou complexas;
3. As variáveis de projeto podem ser contínuas, discretas ou mistas;

Algoritmo “Simulated-Annealing”**Início** $t \leftarrow 0$ inicializar temperatura T selecionar uma solução x_k aleatoriamenteavaliar $f(x_k)$ **Repita****Repita**selecionar uma nova solução x_n na vizinhança de x_k através de uma pequena modificação em x_k **se** $f(x_k) < f(x_n)$ **faça** $x_k \leftarrow x_n$ **senão**calcular $p = \exp\{(f(x_n) - f(x_k))/T\}$ escolher r entre $[0, 1)$ aleatoriamente**se** $r < p$ **faça** $x_k \leftarrow x_n$ **fim-se****fim-se****Até** critério de terminação satisfeito $T \leftarrow g(T, t)$ $t \leftarrow t + 1$ **Até** critério de parada satisfeito**Fim**

Figura 2.2: Algoritmo “Simulated Annealing” – exemplo de maximização de $f(x)$.

4. As funções objetivo e as funções de restrições podem não ser *bem comportadas* e as superfícies de respostas podem ser descontínuas ou até mesmo indefinidas em algumas regiões do espaço de busca. A existência de mínimos/máximos locais é comum, dificultando a busca;
5. Critérios de convergência embutidos em procedimentos iterativos podem introduzir ruídos na avaliação da função objetivo podendo acarretar gradientes numéricos imprecisos;
6. Pode-se necessitar de uma solução inicial de partida para a busca através de alguns algoritmos e se essas não forem bem escolhidas poderá ocorrer um distanciamento de regiões com boas soluções.

Quando existem incertezas ou conhecimentos insuficientes sobre as soluções adotadas e mesmo sobre o espaço onde foram encontradas, deve-se usar métodos capazes de explorar melhor esse espaço à procura de uma melhor solução.

No processo de busca em espaço de soluções, dois aspectos são importantes: o primeiro refere-se à identificação de áreas interessantes onde podem ser encontradas boas soluções e é o que denomina-se como exploração – e o segundo refere-se ao processo de extração das soluções encontradas nessas áreas interessantes, é o que denomina-se como exploração (como tradução do termo da língua inglesa *exploitation*). Deve haver um balanceamento entre os processos de exploração e exploração

pois, uma exploração demasiada pode incluir áreas pouco interessantes e uma exploração exaustiva pode estar concentrada em uma área pouco promissora. Por outro lado, havendo pouca exploração, áreas interessantes podem ser excluídas da busca e poderá ocorrer exploração demasiada da área menos frutífera.

Após Thomas Edison ter investigado 100 combinações de materiais para o filamento elétrico contido no bulbo da lâmpada, seu assistente de laboratório expressou uma frustração, reclamando que não havia visto nenhum sinal de progresso próximo dos 100 fracassos. Edison replicou: “Nós estamos conseguindo um progresso excepcional. Agora, nós conhecemos 100 possibilidades que não devemos trabalhar” [28].

Qualquer tipo de experiência ou conhecimento, sobre uma classe de problemas, adquirido através de algum processo, pode ser utilizado de forma eficiente em um processo de busca. Em geral esta é uma característica importante em problemas de engenharia onde a experiência do projetista gera uma base de conhecimento para a busca de novas soluções. A aplicação do conhecimento na busca necessita de uma forma eficiente para considerar dados já existentes para proporcionar uma busca mais rápida e com melhores soluções evitando um desperdício de tempo avaliando-se soluções de pouco ou nenhum interesse. Exemplos de incorporação de conhecimento do problema podem ser encontrados, entre outros, em [29, 30]

2.5 O usuário

O papel do usuário que está envolvido em algum processo de otimização é de caráter qualitativo. Em geral, um dado importante é sua experiência que poderá interferir nas decisões envolvidas no processo de otimização desde a escolha do algoritmo até os detalhes de sua parametrização.

Para a obtenção da solução de um problema de otimização, poderá existir a necessidade de uma associação de usuários, cada um com uma habilidade específica. Entre eles serão necessários aqueles com conhecimento sobre as ferramentas de otimização e aqueles com conhecimento do problema. Por exemplo, um projetista ou fabricante de algum componente mecânico, que domina as técnicas de fabricação e as restrições de projeto, se associa a grupos que desenvolvem plataformas computacionais gerando uma interface extremamente proveitosa.

Um outro aspecto relevante é a interdisciplinaridade envolvida em problemas de alta complexidade que requerem uma estrutura organizacional eficiente para enfrentar os desafios impostos [9]. Por exemplo, ao se projetar uma aeronave serão necessárias informações sobre aerodinâmica, estruturas, termodinâmica, arquitetura de aeronaves, etc, que devem estar presentes na plataforma computacional.

A modelagem matemática de um sistema pode requerer sofisticação na implementação computacional devido à necessidade de acoplamento entre vários códigos, cada um deles representando um fenômeno físico. A transferência de dados entre os módulos deve ser feita de maneira eficiente evitando análises com alto custo computacional. E assim, deverá existir uma interface entre aqueles que dominam os conhecimentos específicos.

Um importante papel do usuário é a capacidade de decidir de forma rápida sobre algumas questões: escolher um tipo de análise a partir de uma variedade de opções que vão desde as “mais baratas” e aproximadas até as “mais caras” e precisas; re-análises “inteligentes” que repetem somente parte da análise original; mudanças

adequadas nos dados de entrada em função de resultados da análise anterior; visualização e controle da infraestrutura necessária para o manuseio de grande volumes de dados, etc.

Um grande desafio, também, para o usuário é criar uma sistemática que envolva métodos rápidos e eficientes capazes de gerar soluções ótimas, melhorando o desempenho e confiabilidade nas análises reduzindo a dependência de custos adicionais no processo.

Comercialmente, são disponíveis várias plataformas computacionais que atingem vários níveis de expectativas dos usuários, das mais simples às mais complexas. Entretanto, algumas dificuldades são presentes em relação às decisões a serem tomadas, como por exemplo, qual o melhor programa? O programa aceita modelagens complexas? ou, São possíveis simplificações? Qual o custo de uma análise através de vários programas? Além dessas, surgem algumas questões após a escolha do modelo que representa o problema e do programa que realizará a análise: realmente foi encontrado o melhor projeto? A sua execução é possível? Qual o nível de segurança e confiabilidade do projeto? Qual o nível de confiança nas respostas das perguntas anteriores?

Paralelamente às plataformas comerciais existe uma infinidade de plataformas “domésticas” que são desenvolvidas com o objetivo de atender as necessidades específicas de cada usuário, podendo haver inclusive interfaces com as comerciais.

2.6 Otimização estrutural

2.6.1 Projeto estrutural

Em um projeto estrutural é necessário se obter uma maneira adequada para a disposição das peças estruturais de tal forma que a estrutura seja capaz de suportar o carregamento imposto com segurança. Escolhida a disposição estrutural buscase qual a melhor distribuição de material - garantindo as condições de segurança - que ofereça o menor custo. Para tanto são necessárias informações sobre topologia, forma e dimensões da estrutura (tamanho das peças estruturais).

Para uma estrutura, a determinação de uma configuração eficiente é inerente a um processo de tentativa em “erro e acerto”, cujo resultado não garante com segurança que tenha sido encontrado o projeto ótimo absoluto.

Logo, o projeto estrutural concentra-se na determinação de proporções convenientes (ótimas) de uma variedade de tipos de formas para carregamentos atuantes, feita através de várias análises visando a concepção ou determinação da estrutura de melhor desempenho global.

2.6.2 Otimização estrutural

Otimização é uma fusão de áreas da engenharia e matemática capaz de adicionar dados ao projeto além da experiência do projetista.

Para um problema de otimização estrutural o interesse está em se definir uma ou mais funções objetivo, sujeitas ou não a restrições, que sejam capazes de gerar um projeto que atenda às condições de resistência e rigidez com o menor custo.

Em geral, a função objetivo não incorpora um único interesse a ser alcançado, e sim uma série de quesitos envolvidos no problema, isto é, otimização multi-objetivos.

Entre eles incluem-se, por exemplo: escolha do material (custo); a melhor combinação de peças estruturais disponíveis no mercado (perfis metálicos para uma treliça ou elementos pré-moldados para uma edificação civil); topologia; durabilidade; confiabilidade; funcionabilidade; eficiência; capacidade e tempo de execução; recursos para análise (plataformas computacionais e testes de laboratório); etc. Entretanto, são adotadas simplificações devido à complexidade de se conseguir uma parametrização que considera todos esses fatores de forma adequada.

São comuns problemas cujo objetivo principal está em determinar estruturas com peso mínimo (custo mínimo), considerando-se que este objetivo é bastante atraente quanto ao aspecto econômico. Este fato é muito comum, por exemplo, em projetos de estruturas metálicas ou em confecção de peça da indústria automotiva, onde o custo do material é um fator importante. Deve-se ressaltar que na indústria aeronáutica o peso é de extrema importância sendo, geralmente, preponderante em relação ao custo. Em projetos de torres de transmissão ou peças mecânicas com um grande número de unidades fabricadas, uma economia por menor que seja em cada unidade corresponderá a um valor global considerável.

Por exemplo, para uma estrutura reticulada, como uma treliça, a função objetivo, que considera a minimização do peso da estrutura, pode ser escrita da seguinte forma:

$$W = \sum_{i=1}^n \rho_i A_i L_i$$

onde ρ_i é a massa específica do material, A_i é a área da seção transversal e L_i é o comprimento da i -ésima barra da estrutura.

A título de ilustração, outros exemplos de otimização estrutural seriam: otimização da forma da asa de uma aeronave, otimização da seção transversal de uma ponte, otimização da espessura de uma placa ou de um vaso de pressão ou, para um projeto de cascas, encontrar uma forma que sempre satisfaça um estado de tensões dentro dos limites especificados. Vários exemplos acadêmicos e aplicações reais podem ser encontrados em [31], por exemplo.

Cabe ao projetista a capacidade de definir a função ou funções objetivo, identificar todas as restrições importantes, identificar os níveis ou limites aos quais as restrições devem ser satisfeitas e incluir essas informações no processo de otimização estrutural.

Em geral, durante o processo de otimização do projeto, restrições adicionais não previstas inicialmente podem se tornar importantes. O sistema de otimização deve conter, também, facilidades de flexibilização referentes a decisões a serem tomadas durante o processo com o intuito de gerar soluções alternativas. Essas soluções alternativas podem, por exemplo, não respeitar rigorosamente as restrições impostas mas podem basear-se em experiência do projetista adquirida ao longo dos anos de prática.

As restrições podem ser explícitas ou implícitas dependendo do problema. Corriqueiramente, em projetos usuais, estão associadas às tensões, deslocamentos, frequências de vibração, cargas críticas de flambagem, etc.

Uma medida de complexidade dos problemas pode ser dada pelo número de variáveis de projeto ou de restrições impostas. Com o aumento dos recursos computacionais estas medidas sofrem variações. Segundo [32] podem ser estabelecidas três classes de problemas: problemas de menor escala, contendo por volta de 5 incógnitas

ou restrições, problemas de escala intermediária, contendo entre 5 e 100 variáveis e problemas em grande escala, contendo mais do que 100 ou até mesmo 1000 variáveis.

Em resumo, aspectos importantes da otimização estrutural que o projetista deve estar atento, incluem [33]:

- Tamanho, forma e topologia sendo otimizados no mesmo problema em diferentes partes da estrutura;
- Critérios diferentes de otimização em partes diferentes da estrutura;
- Estrutura submetida a vários carregamentos;
- Estrutura com possíveis variações no sistema de apoios;
- Estrutura composta por vários tipos de materiais;
- Consideração de análises bi e tridimensionais;
- Otimização considerando-se análise estática, dinâmica e estabilidade simultaneamente;
- Otimização com não-linearidades físicas e geométricas.

2.6.3 Breve histórico

O interesse pela pesquisa em otimização teve início na década de 50 onde destaca-se o trabalho de J. Heyman [34] que utilizou programação linear para otimizar estruturas no regime plástico. Na década de 60 destaca-se o trabalho de L.A. Schmit [35] que utilizou técnicas de programação não-linear em otimização de projetos estruturais. As maiores aplicações estavam concentradas em engenharia aeronáutica e algumas, por exemplo, envolviam estruturas reticuladas enrijecidas por cascas e placas.

Até então, os projetos que satisfaziam as condições técnicas e de segurança eram aceitos pela indústria. Devido as complexidades matemáticas e falta de recursos computacionais suficientes, as pesquisas permaneceram durante vários anos quase que estritamente com interesse acadêmico.

Entretanto, dois fatores contribuíram para o crescimento do interesse na área de otimização estrutural: o desenvolvimento acelerado de recursos computacionais e a criação de algoritmos eficientes capazes de suportar formulações complexas. Assim, cientistas, engenheiros e fabricantes passaram a investir esforços na busca de projetos mais eficientes e mais econômicos, dando origem a uma forte atividade na área de otimização estrutural.

Na década de 70 vários trabalhos se destacaram e hoje fazem parte da literatura clássica [36]. Vários temas foram tratados, como por exemplo: projeto com tensões limites, algoritmos com critérios de otimalidade, programação matemática e algoritmos de programação linear.

Linhas de pesquisa abordaram problemas de otimização de forma e para isso, foram estudados e implementados algoritmos numéricos através de métodos discretos como elementos finitos, elementos de contorno e diferenças finitas. Os interesses estavam, também, sobre problemas de análise de sensibilidade, programação linear sequencial, funções de penalização para problemas com restrições e programação dinâmica.

A partir desta década pacotes computacionais foram e continuam sendo desenvolvidos, principalmente devido ao grande interesse das indústrias aeronáutica, automotiva e mecânica. Para estes pacotes, tem sido dado ênfase à capacidade de modelagem geométrica (geração automática de malhas), segurança e rapidez na análise (algoritmos eficientes), e facilidades de visualização (estações gráficas).

Em [9] é feito um levantamento sobre os desenvolvimentos recentes associados à área de otimização de projetos aeroespaciais. São discutidos tópicos sobre modelagem matemática, análise de projeto orientado, conceitos de aproximação, processos de otimização, sensibilidade de sistemas e interface humana. Ainda, para métodos e algoritmos podem-se citar, por exemplo, [37, 31, 38, 39, 40, 41, 42] e para aplicações em otimização estrutural [43, 28].

Recentemente [44, 45], vários trabalhos têm sido publicados com destaque nas seguintes áreas: análise de sensibilidades e re-análise, otimização estrutural não-linear, topologia e otimização não-linear, otimização de materiais, critérios de otimalidade, algoritmos genéticos e estratégias de evolução.

Durante os últimos anos os algoritmos genéticos foram apresentados como uma ferramenta de otimização com aplicação em várias áreas: redes neurais, controle ótimo, controle de processos, robótica, otimização estrutural, etc. Sua grande vantagem sobre os outros métodos é de que são métodos de busca global e que não requerem requisitos adicionais como diferenciabilidade da função objetivo. Logo, funções não diferenciáveis com vários mínimos locais podem ser consideradas em problemas de otimização através de algoritmos genéticos. O algoritmo pode ser considerado caro em algumas situações, mas por outro lado pode ser aplicado em situações muito complexas. Inclusive em situações muito específicas, as quais se indica a aplicação de métodos clássicos, pode-se associar tais métodos ao algoritmo genético com a finalidade de se melhorar a busca local mantendo a visão global do algoritmo genético. Esta associação é conhecida como hibridização, e será abordada adiante, na seção 4.6.

Capítulo 3

Computação Evolucionista e Algoritmos Genéticos

3.1 Introdução

Neste capítulo serão apresentados alguns comentários sobre as origens da computação evolucionista e dos algoritmos genéticos. Serão abordadas as principais características dos algoritmos genéticos como: codificação, inicialização da população, função aptidão, esquemas de seleção e reprodução e operadores genéticos tanto para codificações binárias quanto para codificações reais.

3.2 Computação Evolucionista

Antes da utilização de algoritmos genéticos propriamente ditos como ferramentas ou sistemas artificiais para algum tipo de busca, biólogos usaram computadores digitais com o objetivo de simular sistemas genéticos associados à evolução. Entre eles destacam-se N.A. Barricelli, com o trabalho “*Processos de Evolução Simbiogenética através de Métodos Artificiais*” no ano de 1957 e “*Teste Numérico da Teoria da Evolução*”, no ano de 1962 e também A.S. Fraser, com os trabalhos “*Simulação de Sistemas Genéticos por computadores automáticos digitais: acoplamento-S, dominância e epistasia* e *Simulação de Sistemas Genéticos*, nos anos de 1960 e 1962, respectivamente. As simulações desenvolvidas por Fraser não estavam tão distantes da noção de algoritmos genéticos, pois ele trabalhou, inclusive, com um conjunto de cadeias de caracteres codificadas em um alfabeto binário que foram analisadas em sucessivas gerações [46].

Durante os últimos anos cresceu o interesse em se resolver problemas, das naturezas mais diversas, através de procedimentos que utilizam algoritmos que consideram características de hereditariedade entre as soluções candidatas. Na literatura, um termo bastante comum que representa o conjunto de tais procedimentos é denominado *Computação Evolucionista* [47] ou *Computação Evolucionária (CE)*.

Apesar de não se poder definir exatamente onde estão as origens da computação evolucionista, sabe-se que experiências importantes foram realizadas na década de 50. Entretanto, os trabalhos conhecidos nesta área não tiveram forte impacto no meio científico durante algumas décadas.

Um marco importante em pesquisas com utilização de computação evolucionista

é creditado aos experimentos realizados no início da década de 60 pelos Professores Ingo Rechenberg e Hans-Paul Schwefel da Universidade de Berlin na Alemanha [47]. Eles estavam interessados em resolver um problema de otimização que envolvia a forma de placas de metal cujos apoios poderiam estar localizados em várias posições. As placas estavam submetidas a um fluxo de ar que experimentalmente foi produzido em um túnel de vento e o número de apoios era de 5, gerando um espaço de busca da ordem de $3,45 \times 10^8$ configurações [48]. A forma ótima foi encontrada após 300 iterações.

Na década de 70 alguns trabalhos são considerados fundamentais e responsáveis pelo aumento, pode-se dizer exponencial, de pesquisas em computação evolucionista. Destacam-se os trabalhos de Holland [3, 4], Rechenberg [49], Schwefel [50] e Fogel [51].

A computação evolucionista está intrinsecamente associada à implementação de algoritmos evolucionistas que podem ser classificados em três categorias: Algoritmos Genéticos (**AG**), Programação Evolucionista (**PE**) e Estratégias Evolucionistas (**EE**). Todas essas estratégias se baseiam em um conceito de população de indivíduos que são ditos candidatos à solução do problema em questão e são manipulados por operadores probabilísticos conhecidos como operadores de seleção, de recombinação e de mutação. A definição mais clara e abrangente desses operadores está detalhada mais adiante neste texto.

Os algoritmos genéticos foram introduzidos por Holland [3] e posteriormente estudados por vários pesquisadores, destacando-se os trabalhos de De Jong [52], Goldberg [7] e Davis [2]. A programação evolucionista foi introduzida por Fogel [51] e também destaca-se nos trabalhos de seu filho David Fogel [53].

Finalmente, as estratégias evolucionistas foram desenvolvidas por Rechenberg [54] e Schwefel [55], e estudadas por vários outros pesquisadores. Foram inicialmente projetadas com o objetivo de resolver problemas de parametrização em otimização e aplicadas principalmente em problemas experimentais.

As principais diferenças entre as três estratégias estão

- na forma de representação dos indivíduos;
- na definição dos operadores de recombinação e mutação;
- no mecanismo de seleção e reprodução.

Um algoritmo evolucionista em um modelo mais geral é mostrado na Figura 3.1:

Tipicamente, um algoritmo evolucionista inicializa aleatoriamente uma população de soluções candidatas. O processo de avaliação fornece a medida da aptidão de cada indivíduo da população, podendo ser simples ou complexa dependendo do problema analisado. Em geral, o mecanismo de seleção e reprodução é realizado em duas etapas: seleção de pais e teste de sobrevivência. A seleção decide quais serão os pais e quantos serão seus descendentes (filhos). Os filhos são criados a partir da operação de recombinação e mutação. A recombinação é a troca de material genético entre os pais durante o processo de reprodução e a mutação é a operação que introduz perturbações no genótipo. Criados os filhos, estes podem ser avaliados segundo os critérios de sobrevivência.

Existe uma série de variações em torno dos algoritmos evolucionistas. Algumas recebem destaque e entre elas pode-se citar os algoritmos genéticos com codificação real, sistemas de classificação e programação genética.

Algoritmo Evolucionista AE

```

Início
  t = 0
  inicialize P(t)
  avalie P(t)
  enquanto não termina processo faça:
    Início
      t = t + 1
      selecione P(t) a partir de P(t-1)
      altere P(t)
      avalie P(t)
    Fim
Fim

```

Figura 3.1: Algoritmo Evolucionista.

- ALGORITMOS GENÉTICOS COM CODIFICAÇÃO REAL [8]: Utilização de variáveis reais para codificação das soluções candidatas.
- SISTEMAS DE CLASSIFICAÇÃO [5]: Em geral são constituídos por algoritmos evolucionistas que procuram espaços de definição de regras através da manipulação de dados. Geralmente os espaços são codificados por cadeias de caracteres sobre um alfabeto ternário ou através de regras simbólicas. Tradicionalmente, são conhecidas duas técnicas: a abordagem de Michigan e a abordagem de Pittsburgh que diferem entre si, respectivamente, pelo fato de um elemento da população estar associado a uma única regra ou com uma base completa de regras.
- PROGRAMAÇÃO GENÉTICA [56]: Aplicação de algoritmos genéticos onde os elementos manipulados são programas escritos em linguagens adequadas e sobre os mesmos atuam operadores de recombinação e mutação. Uma linguagem utilizada nesses programas é o LISP.

3.3 Algoritmos Genéticos

Os algoritmos genéticos foram desenvolvidos por John Holland no início da década de 70 na Universidade de Michigan[3, 6]. Ele acreditava na possibilidade de implementação de um algoritmo capaz de resolver problemas complexos espelhando-se no processo de evolução encontrado na natureza.

Os objetivos de suas pesquisas estavam concentrados em dois pontos fundamentais: abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais e criar programas de computador para simular sistemas artificiais mantendo os mecanismos importantes dos sistemas naturais [7].

Algoritmos genéticos podem ser definidos como procedimentos de busca baseados na genética e seleção natural das espécies. Eles combinam o conceito de sobrevivência artificial através de testes de classificação com o de operadores genéticos

abstraídos da natureza para formar um mecanismo eficiente de busca. Testes de classificação têm como objetivo promover a distinção entre os melhores e os piores indivíduos (soluções), ou seja, a interação entre o indivíduo e o meio em que vive fornece sua aptidão à sobrevivência.

John Holland direcionou suas pesquisas para três pontos: cromossomo, indivíduo e população. A tentativa era de se encontrar uma maneira de se codificar um cromossomo que representasse um indivíduo e uma população.

As suas primeiras codificações surgiram através de um alfabeto binário composto de 0's e 1's representando artificialmente um "código genético".

Duas definições são importantes: a de população e a de operadores genéticos. A população é definida como o conjunto de indivíduos codificados por seus cromossomos e que serão avaliados pelo meio em que vivem. Por um processo de competição, cada indivíduo da população é testado e classificado. Isto permitirá a escolha adequada dos indivíduos desta população (genitores) que originarão os indivíduos novos (filhos) da próxima geração. Três operações básicas são feitas sobre a população: seleção, recombinação e mutação. Pelo operador de seleção são escolhidos os genitores (pais), e sobre eles atuam os operadores de recombinação e mutação. Se os genitores são *bons*, isto é, se as suas soluções são as que melhor satisfazem o problema naquela geração é de se esperar que os seus filhos também o sejam, trazendo também boas soluções. Dessa forma a população evoluirá para uma solução desejada. Esta analogia se faz com o processo de seleção e sobrevivência na natureza.

Basicamente, um algoritmo genético apresenta cinco aspectos fundamentais quando usado para resolver um problema:

1. uma codificação genética de soluções para o problema;
2. um procedimento para criar uma população inicial de soluções;
3. uma função de avaliação que retorna a aptidão de cada indivíduo;
4. operadores genéticos que manipulam a codificação dos pais durante o processo de reprodução dando origem a novos indivíduos;
5. parâmetros a serem utilizados no algoritmo durante os processos de reprodução e mutação.

E, pode ser descrito da seguinte forma [2]:

1. Inicializar uma população de cromossomos;
2. Avaliar cada cromossomo na população;
3. Criar novos cromossomos através da troca de material genético entre cromossomos;
4. Remover membros da população para dar lugar a novos cromossomos;
5. Avaliar os novos cromossomos e inserí-los na população;
6. Se o procedimento convergiu, terminar, se não, voltar ao passo 3.

A manipulação concatenada de todos esses aspectos tornou viável a codificação e implementação de algoritmos genéticos de uma forma robusta e independente, capaz de ser aplicada a uma gama de problemas de maneira eficiente.

Um pseudo-código que representa um algoritmo genético genérico é mostrado na Figura 3.2:

```

Algoritmo genético genérico
  Inicialize a população
  Avalie indivíduos na população
  Repita
    Selecione indivíduos para reprodução
    Aplique operadores de recombinação e mutação
    Avalie indivíduos na população
    Selecione indivíduos para sobreviver
  Até critério de parada satisfeito
Fim

```

Figura 3.2: Algoritmo genético genérico.

3.3.1 Codificação

A codificação é a forma utilizada para se representar possíveis soluções do problema. Existem vários “*alfabetos*” utilizados neste procedimento destacando-se entre eles a codificação binária e a codificação real. Sendo $x = (x_1, x_2, \dots, x_n)$, $x_i \in \mathbb{R}$ uma possível solução para um dado problema, a sua codificação em alfabeto binário, por exemplo, seria obtida pela concatenação da codificação binária de cada elemento x_i desse conjunto. O resultado final desta é o cromossomo.

É comum a utilização de codificações binárias para problemas com variáveis discretas e de codificações reais para problemas com variáveis contínuas. Para ilustrar, em um problema com quatro variáveis, estas poderiam ser codificadas, usando-se, por exemplo, cinco bits para cada uma delas, da seguinte maneira:

$$\begin{aligned}
 x_1 &= 10100 \\
 x_2 &= 01001 \\
 x_3 &= 11010 \\
 x_4 &= 10001
 \end{aligned}$$

A associação destas codificações representaria, então, uma possível solução para o problema, através do cromossomo:

$$10100 \cdot 01001 \cdot 11010 \cdot 10001 \rightarrow 10100010011101010001$$

Para recuperar os valores originais (físicos) das variáveis é necessário um procedimento de decodificação.

Para as variáveis discretas a decodificação fornece um número inteiro que representa o índice da variável em uma lista que é o espaço de busca correspondente a essa variável.

Por exemplo, seja $x_i = 101$. Sua decodificação indicará o índice $IND = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$, que apontará para a quinta variável discreta do espaço de busca dessa variável.

Para as variáveis contínuas tem-se a decodificação:

$$x = x^L + IND \times \frac{x^U - x^L}{2^{nb} - 1}$$

Por exemplo, seja $\varepsilon = 0.001$ e $x^U = 50.0$ e $x^L = 30.0$, e $nb = 15$. Se $x_i = 100000000000001$, encontra-se $IND = 1 \times 2^{14} + 0 \times 2^{13} + 0 \times 2^{12} + 0 \times 2^{11} + 0 \times 2^{10} + 0 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16385$ e, finalmente, a variável x_i .

$$x_i = 30 + 16385 \times \frac{(50 - 30)}{2^{15} - 1} = 40.00091556$$

Vale lembrar que a escolha do número de bits para cada variável, o tamanho do cromossomo e sua decodificação, dependem de cada problema.

A escolha do número de bits para cada variável é feita da seguinte forma [8, 7]:

- para um espaço de busca com variáveis discretas, tem-se

$$2^{nb} = nv$$

onde nb é o número de bits e nv é o número de possíveis valores assumidos.

- para um espaço com variáveis reais (contínuas), nb bits possibilitam a representação de 2^{nb} valores discretos que podem ser distribuídos uniformemente no intervalo $[x^L, x^U]$, acarretando uma resolução (precisão), de:

$$\varepsilon = \frac{x^U - x^L}{2^{nb} - 1}$$

Consequentemente, o número de bits necessários para garantir uma resolução ε é:

$$nb \geq \log_2 \frac{x^U - x^L}{\varepsilon}$$

onde x^U e x^L são os limites superior e inferior do espaço de busca. Por exemplo, se $\varepsilon = 0.001$ e $x^U = 50.0$ e $x^L = 30.0$, então, $nb \geq 15$. Na representação de um espaço de variáveis reais através de uma codificação binária, é interessante que dois pontos contíguos (próximos) na representação também o sejam no espaço propriamente dito.

Nota-se que isso pode não ocorrer em uma codificação binária. Verifica-se, por exemplo, na Tabela 3.1 que a codificação contígua ao binário 0111 é o binário 1000 com bits diferentes em todas as posições e, para se passar de um para o outro é necessário a troca de todos os bits. Esta discrepância pode ser reduzida utilizando-se o código de Gray [8, 7].

Nota-se que a representação de dois valores consecutivos pelo código de Gray difere somente pela permuta de 1 bit, fazendo com que a mudança de uma unidade na variável corresponda à troca de um bit na codificação.

Na Tabela 3.1 associa-se 0000 ao zero, 0001 ao número 1, 0010 ao número 2, e assim sucessivamente até o número 15 associado ao binário 1111. Algoritmos para transformação do código de Gray para código podem ser encontrados, por exemplo, em [8].

Inteiro	Binário	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Tabela 3.1: Código binário usual e Código de Gray.

3.3.2 Inicialização da população

Após a definição do processo de codificação pode-se escolher uma população inicial que é a representação de possíveis soluções do problema a ser avaliado. Os elementos que constituem a população inicial são escolhidos, geralmente, de forma aleatória utilizando-se funções (pseudo)-randômicas nas rotinas de códigos computacionais. O número de elementos é escolhido de forma adequada para cada problema e pode ficar entre algumas dezenas e poucas centenas, dependendo também da codificação.

3.3.3 Função aptidão

A função aptidão, representa uma medida que avalia a capacidade de sobrevivência de um elemento da população durante o processo de evolução. A decodificação do cromossomo fornecerá parâmetros (variáveis), para a função aptidão, que servirão para retornar o valor de sua avaliação. Aqueles que obtiverem as melhores avaliações terão mais chances de sobreviver e conseqüentemente de se reproduzirem mais frequentemente.

Cada elemento da população (possível solução) terá a medida de sua aptidão que servirá como elemento de classificação dessas soluções indicando as chances de reprodução de cada uma. A classificação pode ser feita através de uma ordenação das soluções em ordem decrescente ou crescente de suas aptidões, respectivamente, se o problema é de minimização ou maximização.

Genericamente, a função aptidão para um problema de otimização estrutural, pode ser definida como:

$$F(x) = f(x) + penal(x)$$

onde $f(x)$ é a função objetivo e $penal(x)$ é conhecida como uma função de penalização.

Se o problema não apresenta restrições em sua formulação a função de penalização não é considerada e a função aptidão é a própria função objetivo. A penalização é diretamente ligada as restrições e somente será ativada quando alguma delas for violada. Problemas com restrições não são fáceis de resolver, em geral, justamente devido à dificuldade de se ponderar a função de penalização que tem relação direta com o próprio problema. A dificuldade na maioria dos casos é de se encontrar a melhor forma de se penetrar na região factível. Por exemplo, se a penalização é pequena a população poderá evoluir para soluções não factíveis e se a penalização é grande a população poderá convergir prematuramente para uma solução factível mas não ótima. Funções de penalização serão tratadas mais a frente neste texto.

3.3.4 Seleção

O objetivo da seleção é escolher os indivíduos que servirão de pais no processo de reprodução. Pela teoria da evolução de Darwin, os melhores indivíduos da população devem ser escolhidos para sobreviver e gerar filhos. Existem várias formas de efetuar a seleção, podendo-se citar a seleção por posição (*rank selection*), seleção proporcional à aptidão (*roulette wheel*), seleção de Boltzman, seleção por torneio. A seguir examina-se algumas técnicas de seleção em detalhes.

1. Seleção por posição: Neste tipo de seleção somente a posição do elemento dentro de uma lista ordenada (*rank*) é utilizada. O primeiro colocado do *rank* tem maior chance de reprodução. Este tipo de seleção não tem analogia na biologia.
2. Seleção proporcional à aptidão: Este tipo de seleção é bastante encontrado na literatura. Pode ser entendido através do seguinte algoritmo [2]:

- (a) Obtenha a soma das aptidões de todos os elementos da população F_T ;

$$F_T = \sum_{i=1}^{popsiz} F_i$$

onde *popsiz* é o tamanho da população e F_i é a aptidão de um indivíduo da população.

- (b) Escolha um número aleatório n entre 0 e F_T ;
- (c) Retorne o primeiro elemento da população, cuja soma das aptidões dos elementos até ele, inclusive, é maior ou igual a n .

A Tabela 3.2 mostra a aptidão de cada indivíduo da população e a soma parcial das aptidões e a Tabela 3.3 mostra a escolha de um valor de aptidão representada por n e o respectivo indivíduo selecionado para reprodução.

Elemento	F_i	F_{pi}
1	13	13
2	8	21
3	27	48
4	5	53
5	43	96
6	12	108
7	25	133
8	15	148

Tabela 3.2: Seleção proporcional à aptidão – $F_T = 148$, F_{pi} = soma parcial das aptidões.

n	Elemento
55	5
28	3
103	6
125	7

Tabela 3.3: Número aleatório n e elemento selecionado.

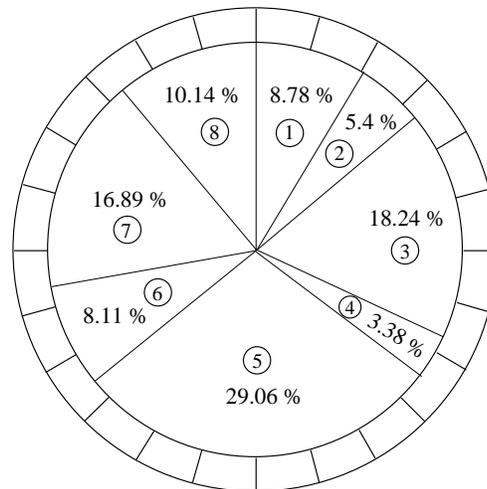


Figura 3.3: Seleção proporcional à aptidão.

A Figura 3.3 ilustra a seleção proporcional à aptidão comparando o processo de escolha a um “sorteio” através de uma “roleta”, onde cada elemento é associado a uma probabilidade de ser escolhido.

Quando uma geração (de “pais”) é totalmente substituída por outra mais nova (de “filhos”) corre-se o risco de se perder um bom indivíduo. Para preservar parte da população que tem desempenho superior pode-se definir uma porcentagem dela

que será mantida, isto é, copiada diretamente para a geração seguinte. Esta prática é conhecida como *elitismo* [2].

3.3.5 Dois esquemas (algoritmos) de reprodução

Os dois esquemas de reprodução mais encontrados na literatura são conhecidos como “Geracional” e “Steady-state” e são detalhados a seguir.

1) “Geracional”: Este tipo de algoritmo tem como principal característica a substituição total da população a cada geração. Tem a desvantagem da possibilidade de perda de material genético de boa qualidade. Para evitar isso, muitas vezes adota-se um processo de seleção elitista mantendo-se parte dos melhores indivíduos da geração anterior. O pseudo-código mostrado na Figura 3.4 representa este algoritmo:

```

Algoritmo genético geracional
Início
  Inicialize a população P aleatoriamente
  Avalie indivíduos na população P
Repita
  Repita
    Selecione 2 indivíduos em P
    Aplique operador de recombinação com probabilidade  $p_c$ 
    Aplique operador de mutação com taxa  $p_m$ 
    Insira novos indivíduos em P'
  Até população P' completa
  Avalie indivíduos na população P'
   $P \leftarrow P'$ 
Até critério de parada satisfeito
Fim

```

Figura 3.4: Algoritmo genético geracional.

2) “Steady-state”: De maneira diferente do “geracional”, este algoritmo tem como característica principal a geração de somente um indivíduo de cada vez. A cada nova criação uma avaliação é feita. Se o novo indivíduo for melhor que o pior na lista de classificação, ele terá direito à sobrevivência e o pior será eliminado. Esta seria uma “política de inserção” tradicional, sendo possíveis outras variações. Pode-se, por exemplo, inserir somente indivíduos que tenham aptidões superiores que a aptidão mediana, inserir indivíduos nos lugares dos piores ou pais mais próximos, etc. O pseudo-código correspondente é mostrado na Figura 3.5.

3.3.6 Operadores genéticos

Nesta seção serão discutidos os operadores de recombinação e mutação com codificações binárias e codificações reais. O operador de recombinação é conhecido na língua inglesa como *crossover*. A partir daqui neste texto, também se utilizará esta denominação.

Algoritmo genético “Steady-state”**Início**

Inicialize a população P aleatoriamente
 Avalie indivíduos na população P
 Ordene a população P de acordo com a aptidão

Repita

Selecione operador genético
 Selecione indivíduo(s) para reprodução
 Aplique operador genético
 Avalie indivíduo(s) gerado(s)
 Selecione indivíduo f para sobreviver

Se f é melhor que o pior elemento de P Então

Insira f em P de acordo com seu “ranking”

Até critério de parada satisfeito

Fim

Figura 3.5: Algoritmo genético “steady-state”.

3.3.7 Operadores para codificação binária**Crossover de um ponto**

Este operador é o mais clássico dentro dos algoritmos genéticos. Sua função é selecionar aleatoriamente um ponto localizado entre os limites $\{1, \dots, L - 1\}$ dos cromossomos selecionados como pais e promover a recombinação de material entre eles, gerando dois filhos. Os exemplos a seguir esclarecem esta operação:

Pai 1	1111 111	Filho 1	11111000
Pai 2	0000 000	Filho 2	00001111

onde foi sorteada a 5ª posição de corte.

Analogamente, pode-se ter o crossover de dois pontos. Agora, escolhem-se dois pontos aleatoriamente no intervalo $\{1, \dots, L - 1\}$ e troca-se o material genético entre eles.

Pai 1	1111 111 111	Filho 1	1111000111
Pai 2	0000 000 000	Filho 2	0000111000

onde foram sorteadas a 4ª e 7ª posições de corte.

Crossover de n pontos

Este é uma generalização natural do crossover de um ou dois pontos. Aqui, escolhem-se aleatoriamente n pontos pertencentes ao intervalo $\{1, \dots, L - 1\}$ e troca-se o material genético.

Crossover uniforme

Neste caso procede-se a uma escolha uniforme dos números aleatórios que indicarão a troca de material genético. O exemplo a seguir esclarece.

Pai 1	11111111	Filho 1	10010110
“máscara”	01101001		
Pai 2	01001100	Filho 2	00100101

Percebe-se que a codificação dos filhos é obtida pela troca ou não de bits dos pais. Se a “máscara” tiver o bit 1 haverá troca do bit correspondente a sua posição. Se o bit for 0 nada acontecerá.

Crossover com vários pais

Este operador [57] generaliza a idéia de crossover para N pais selecionando-se $N - 1$ pontos de recombinação e compondo N filhos através de N segmentos de cromossomos a partir dos pais participantes. O exemplo a seguir mostra um crossover com três pais e três filhos ou com três pais e um único filho (diagonal).

Pai 1	101 110 101	Filho 1	101 001 110
Pai 2	111 001 101	Filho 2	010 110 101
Pai 3	010 110 110	Filho 3	111 110 101

ou

Pai 1	101 110 101		
Pai 2	111 001 101	Filho	101 001 110
Pai 3	010 110 110		

Todos estes operadores são extensíveis à codificação real.

Mutação

Este operador é utilizado após a aplicação dos operadores de recombinação. Seu objetivo principal é introduzir diversidade entre os novos indivíduos da população. Em códigos binários o operador mutação efetua uma troca no bit. Se o bit for 1 ele passa para 0 e vice-versa. Cada bit é mudado com probabilidade p_m sendo geralmente da ordem de $1/L$ onde L é o tamanho do cromossomo.

	0110 0 110
mutação	0110 1 110

Neste exemplo, na posição escolhida aleatoriamente, houve a troca do bit 0 para o bit 1.

3.3.8 Operadores para codificação real

A codificação real é naturalmente a mais indicada em problemas cujas variáveis estão associadas a domínios contínuos. O cromossomo é um vetor em \mathbb{R}^n onde n é o número de variáveis do problema. Neste caso, o tamanho do cromossomo se confunde com o tamanho do vetor. Enquanto na codificação binária a precisão das variáveis está associada ao número de bits que as representam, na codificação real a precisão é equivalente à precisão da máquina. Uma precisão mais ou menos rigorosa não implicará na alteração do tamanho do cromossomo. Outra vantagem é o fato de não haver necessidade de processos de codificação e decodificação, isto é,

o indivíduo e sua codificação se confundem. Os exemplos de operadores que serão tratados nesta seção seguem, basicamente, a apresentação discutida em [58] onde alguns são mostrados esquematicamente na Figura 3.6, também encontrada em [58]. Será utilizada a seguinte notação para os cromossomos selecionados para a aplicação dos operadores:

$$\begin{array}{ll} \text{cromossomo } 1 & C_1 = (c_1^1, c_2^1, \dots, c_n^1) \\ \text{cromossomo } 2 & C_2 = (c_1^2, c_2^2, \dots, c_n^2) \end{array}$$

Crossover uniforme

Este operador é o mais simples que pode ser aplicado a uma codificação real. É gerado um filho $F = (f_1, f_2, \dots, f_n)$ cujas componentes f_i são escolhidas aleatoriamente (uniformemente) no intervalo $[c_i^1, c_i^2] : c_i^1 \leq f_i \leq c_i^2$.

Crossover simples

Analogamente ao crossover de um ponto para codificação real, escolhe-se um ponto no intervalo $\{1, 2, \dots, n-1\}$, e procede-se à troca de material genético entre os cromossomos C_1 e C_2 .

$$\begin{array}{l} F_1 = (c_1^1, c_2^1, \dots, c_i^1, c_{i+1}^2, \dots, c_n^2) \\ F_2 = (c_1^2, c_2^2, \dots, c_i^2, c_{i+1}^1, \dots, c_n^1) \end{array}$$

Crossover aritmético

São gerados dois filhos F_1 e F_2 :

$$\begin{array}{l} F_1 = (f_1^1, f_2^1, \dots, f_n^1) \\ F_2 = (f_1^2, f_2^2, \dots, f_n^2) \end{array}$$

onde:

$$\begin{array}{l} f_i^1 = \lambda c_i^1 + (1 - \lambda)c_i^2 \\ f_i^2 = \lambda c_i^2 + (1 - \lambda)c_i^1 \end{array}$$

onde λ é uma constante ou uma variável cujo valor depende, por exemplo, da geração em que se encontra o processo de evolução (crossover aritmético não-uniforme) [8].

Crossover $BLX - \alpha$ (*Blended crossover*)

É gerado um filho $F_1 = (f_1^1, f_2^1, \dots, f_n^1)$ onde f_i é escolhido aleatoriamente no intervalo $[c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$, onde $c_{min} = \min(c_i^1, c_i^2)$ e $c_{max} = \max(c_i^1, c_i^2)$ e $I = c_{max} - c_{min}$. Onde α é uma constante, geralmente, escolhida no intervalo $0 \leq \alpha \leq 0.5$. Nota-se aqui que para $\alpha = 0$ recupera-se o crossover uniforme.

Crossover linear

São gerados três filhos

$$F_1 = (f_1^1, \dots, f_n^1)$$

$$F_2 = (f_1^2, \dots, f_n^2)$$

$$F_3 = (f_1^3, \dots, f_n^3)$$

onde os elementos de cada um são definidos da seguinte forma:

$$f_i^1 = \frac{1}{2}c_i^1 + \frac{1}{2}c_i^2$$

$$f_i^2 = \frac{3}{2}c_i^1 - \frac{1}{2}c_i^2$$

$$f_i^3 = -\frac{1}{2}c_i^1 + \frac{3}{2}c_i^2$$

Crossover discreto

Gera-se um filho $F_i = (f_1^i, \dots, f_n^i)$ onde f_i é escolhido aleatoriamente (uniformemente) no conjunto $\{c_i^1; c_i^2\}$.

Crossover linha estendida

Este tipo de operador (*extended line crossover*) gera um elemento do cromossomo do filho da seguinte forma: $f_i = c_i^1 + \alpha(c_i^2 - c_i^1)$ onde α é gerado aleatoriamente no intervalo $[-0.25, 1.25]$. Uma possível variação para este operador seria considerar um valor de α variável (escolhido aleatoriamente), para cada f_i , ou seja, $f_i = c_i^1 + \alpha_i(c_i^2 - c_i^1)$. Esta variação é conhecida como crossover intermediário estendido.

Crossover heurístico de Wright

Para este operador, escolhe-se um número aleatório r no intervalo $[0, 1]$. Entre os cromossomos C_1 e C_2 , selecionados, escolhe-se o de maior aptidão – digamos C_1 – determinando-se cada variável do cromossomo do filho da seguinte forma:

$$f_i = r \cdot (c_i^1 - c_i^2) + c_i^1$$

Crossover linear BGA (*Breeder Genetic Algorithm*)

Neste caso, cada elemento f_i do cromossomo a ser gerado é obtido pela seguinte expressão:

$$f_i = c_i^1 \pm rang_i \cdot \gamma \cdot \Lambda$$

onde

$$\Lambda = \frac{c_i^2 - c_i^1}{\|C_1 - C_2\|}$$

O sinal negativo na expressão de f_i é escolhido com uma probabilidade de 0.9. Na maioria dos casos, escolhe-se para $rang_i$ o valor $0.5 \cdot (b_i - a_i)$. A determinação de γ é feita por

$$\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$$

onde $\alpha_i \in \{0, 1\}$ é obtido aleatoriamente com probabilidade $p(\alpha_i = 1) = \frac{1}{16}$.

Crossover ls-x (*line search crossover*)

Primeiramente, gera-se um filho que é a média entre os pais

$$F_1 = 0.5(c_1^1 + c_1^2, \dots, c_n^1 + c_n^2)$$

sendo C_1 o pai com melhor aptidão. Se a aptidão de F_1 for melhor que a aptidão de C_1 , então o novo filho será definido como F_1 . Se isto não ocorrer, gera-se um novo filho F_2 mais próximo do melhor pai C_1 , ou seja

$$F_2 = (c_1^1 + 0.5(c_1^1 - c_1^2), \dots, c_n^1 + 0.5(c_n^1 - c_n^2))$$

e, então, escolhe-se o melhor filho entre F_1 e F_2 .

Para facilitar os procedimentos de mutação mostrados a seguir define-se $c_i \in [a_i, b_i]$ como sendo o elemento (*gene*) a sofrer o processo de mutação. Este pertence ao cromossomo $C_i = (c_1, c_2, \dots, c_n)$. Após a aplicação do operador de mutação seu valor passa a ser c_i^* .

Mutação aleatória

Escolhe-se aleatoriamente um elemento no cromossomo e c_i^* tem seu valor determinado aleatoriamente no intervalo $[a_i, b_i]$.

Mutação não-uniforme

Se este operador é aplicado em uma certa geração gen e se o número máximo de gerações é $maxgen$, então:

$$c_i^* = \begin{cases} c_i + \Delta(gen, b_i - c_i) & \text{se } \tau = 0 \\ c_i - \Delta(gen, c_i - a_i) & \text{se } \tau = 1 \end{cases}$$

Na expressão anterior τ é escolhido aleatoriamente e seu valor pode ser 0 ou 1 e a função $\Delta(gen, y)$ é dada por:

$$\Delta(gen, y) = y(1 - r^{(1 - \frac{gen}{maxgen})^b})$$

onde r é um número aleatório escolhido no intervalo $[0, 1]$ e o parâmetro b é definido pelo usuário, determinando o grau de dependência com o número de iterações. Essa função fornece um valor no intervalo $[0, y]$. Com o aumento de gerações o intervalo $[a_i, b_i]$ diminui de comprimento, imaginando-se que o processo está convergindo. Esta característica acarreta uma busca uniforme nas gerações iniciais, ou seja, quando gen é pequeno e, bem mais localizada nas gerações finais.

Mutação *Small-Creep* e *Big-Creep*

Quando a função analisada apresenta vários mínimos ou máximos locais e em um dado instante do processo aproxima-se do máximo global é conveniente fazer com que a operação de mutação gere novos elementos de tal forma que não se distanciem dessas proximidades, isto é, no entorno do máximo global. Para isso introduz-se “deslocamentos” nos cromossomos controlados por pequenas quantidades escolhidas aleatoriamente. Definem-se, então, os operadores *small-creep* e *big-creep* que têm suas diferenças associadas ao valor máximo dos deslocamentos a serem aplicados.

CROSS.	REPRESENTAÇÃO
Aritmético	
BLX- α	
Linear	
Discreto	
Estendido	
Linear BGA	
Heurístico de Wright	

Figura 3.6: Crossover para codificação real – Tabela reproduzida da referência [58].

Mutação de Mühlenbein

Neste caso [58], a obtenção do elemento c_i^* que sofreu a mutação é dada por:

$$c_i^* = c_i \pm rang_i \cdot \gamma$$

onde $rang_i$ define o campo de mutação, geralmente, escolhido no intervalo $0.1 \cdot (b_i - a_i)$. Os sinais + ou - na expressão de c_i^* são escolhidos com probabilidade de 0.5 e γ é dado por:

$$\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$$

onde $\alpha_i \in \{0, 1\}$ é obtido aleatoriamente com probabilidade $p(\alpha_i = 1) = \frac{1}{16}$. São gerados, por este operador, valores no intervalo $[c_i - rang_i, c_i + rang_i]$ com probabilidade alta de se obter valores nas proximidades de c_i . A menor proximidade possível é obtida com uma precisão de $rang_i = 2^{-15}$. Podem ser encontrados na literatura [58], variações deste operador apresentando diferenças na forma em que se determina γ .

Mutação de uma unidade

Esse operador gera um filho pela adição ou subtração aleatória de uma unidade da variável x_i , também escolhida aleatoriamente.

3.3.9 Codificação real ou binária ?

Em [8] é feita uma análise detalhada sobre a utilização de codificação binária ou real (ponto flutuante). É estudado um caso teste através de duas implementações. As conclusões obtidas mostram que a codificação real é mais rápida, mais consistente, alcançando uma melhor precisão, pois no caso da codificação binária necessita-se de cromossomos de tamanhos extensos para uma boa precisão dada por $(x^U - x^L)/(2^{nb} - 1)$ que torna-se, em geral, proibitiva para problemas com grande número de variáveis.

Conclui-se, também, que torna-se mais fácil a implementação de novos operadores que introduzem conhecimento do problema quando utiliza-se a codificação real, principalmente em problemas com restrições.

Ainda, uma sugestão que pode ser feita é a utilização de codificação real quando o problema apresenta espaço de busca com variáveis contínuas (otimização contínua) e codificação binária quando o problema apresenta espaço de busca com variáveis discretas (otimização discreta).

Capítulo 4

Tópicos adicionais em Algoritmos Genéticos

4.1 Introdução

Neste capítulo serão apresentados alguns tópicos adicionais em algoritmos genéticos destacando-se morfogênese, problemas com restrições, utilização de procedimentos de busca local e hibridização, adaptação, medidas de diversidade da população e algoritmos genéticos paralelos.

4.2 Morfogênese

Na biologia, morfogênese é o processo de desenvolvimento de um organismo, a partir do seu genoma, cujo desempenho será avaliado pelo meio em que vive.

A implementação do processo de morfogênese nos algoritmos genéticos [59], estabelece os algoritmos genéticos morfogenéticos, que objetivam uma distinção entre o genótipo e o seu fenótipo correspondente, a ser avaliado pela função aptidão.

Cada elemento da população é transformado (*mapeado*) antes de ser avaliado pela função aptidão, e este mapeamento, ocorre através da criação de funções de desenvolvimento ou funções de maturação. Utilizando-se o conceito de morfogênese pode-se conseguir algoritmos genéticos mais eficientes. Algumas vantagens decorrentes da utilização de algoritmos genéticos morfogenéticos:

- Compactação da estrutura codificada reduzindo-se o tamanho do cromossomo e conseqüentemente reduzindo-se o tempo de manipulação desta;
- Introdução de conhecimento específico sobre o problema na fase de desenvolvimento facilitando a busca;
- Construção de fenótipo viável evitando-se avaliações de elementos infactíveis;
- Deseja-se: aumento da *evolvabilidade* das soluções (fenótipos) aumentando as probabilidades de criação de prole factível e melhorada.

Três tipos de funções de desenvolvimento recebem atenção especial e estão ilustradas pelas Figuras 4.1, 4.2, 4.3 e 4.4. Essas Figuras foram reproduzidas de [59]

que mostra também uma ampla discussão sobre as aplicações de algoritmos genéticos morfogenéticos.

As funções de desenvolvimento podem ser classificadas como:

4.2.1 Funções de desenvolvimento translativas

Envolvem mapeamentos triviais entre estruturas codificadas e estruturas avaliadas. Um exemplo simples é o código de Gray e um mais complexo é o da decodificação de parâmetros em uma rede neural. Nota-se pela Figura 4.1 que os elementos encontrados na estrutura codificada são mapeados diretamente na estrutura a ser avaliada cuja “conexão” se manterá inalterada, ou seja, as variáveis encontradas em cada estrutura codificada terão destinos definidos na estrutura avaliada.

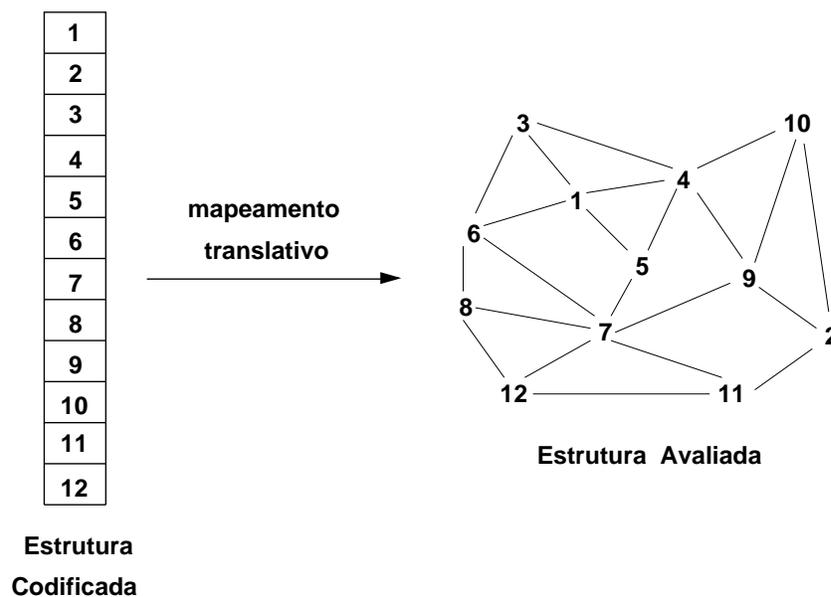


Figura 4.1: Mapeamento translativo.

4.2.2 Funções de desenvolvimento generativas

Envolvem criação recursiva de definições entre estruturas codificadas e estruturas avaliadas. Este é um desenvolvimento mais complexo que o caso anterior onde as estruturas codificadas estão “compactadas”, necessitando-se de uma função de “descompactação”. Pela Figura 4.2 percebe-se que a estrutura a ser avaliada é obtida por uma função de descompactação que interpreta progressivamente os elementos codificados. São encontradas aplicações no desenvolvimento de arquitetura de redes neurais e programação genética.

4.2.3 Funções de desenvolvimento adaptativas

Envolvem criação dinâmica de funções de mapeamento entre estruturas codificadas e estruturas avaliadas. Na Figura 4.3 é realizada uma operação de decomposição através da criação de “módulos” da estrutura codificada e estes novos “módulos”

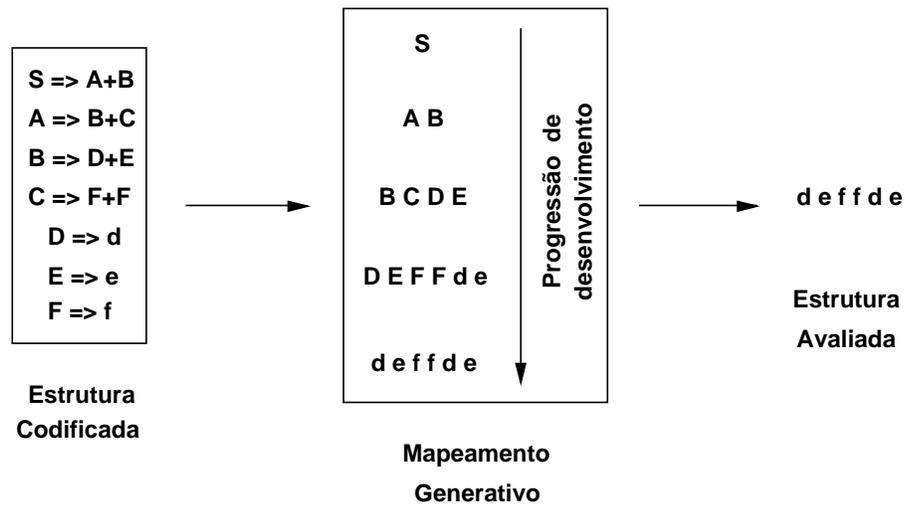


Figura 4.2: Mapeamento generativo.

passam a fazer parte de uma biblioteca de opções a serem utilizadas em estruturas avaliadas como mostra a Figura 4.4. Este tipo de mapeamento tem forte aplicação em programação genética.

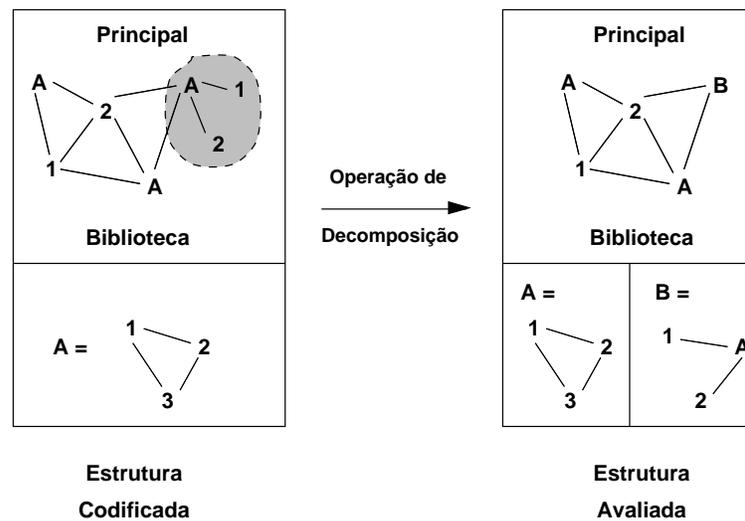


Figura 4.3: Mapeamento adaptativo (a).

4.3 Problemas com restrições

Várias técnicas para tratamento de restrições são encontradas na literatura. Estas podem ser agrupadas em várias categorias de acordo com o tipo de algoritmo, tipo de restrições, tipo de problema, etc. Em geral, as estratégias estão associadas a algum procedimento, que pode ser, por exemplo: utilização de funções de penalização, utilização de operadores especiais, utilização de técnicas de otimização multi-objetivo,

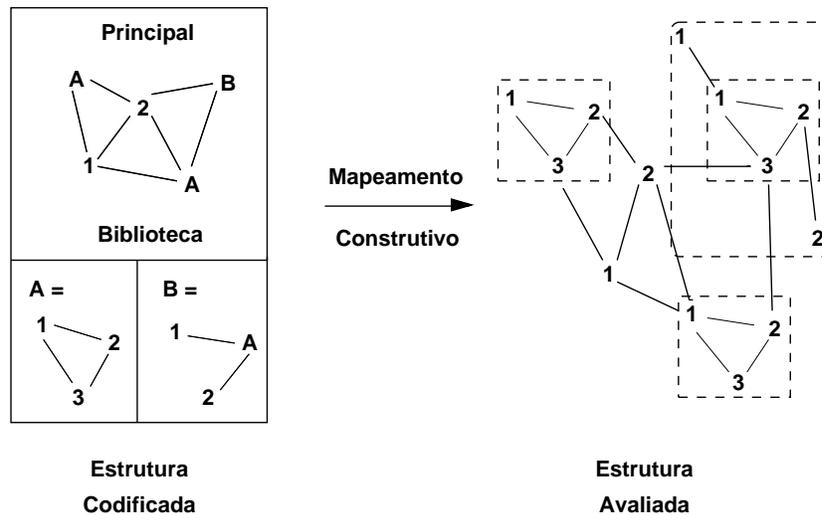


Figura 4.4: Mapeamento adaptativo (b).

utilização de modelos de co-evolução, utilização de operadores de reparo, etc.

4.3.1 Funções de penalização

Geralmente, um problema com restrições pode ser transformado em um problema sem restrições pela associação de uma função de penalização. Estas funções são incorporadas à função objetivo gerando uma única função a ser otimizada. O caso mais simples é a utilização de funções de penalizações constantes associadas a um parâmetro de penalização que atua sobre as restrições violadas. No entanto, a determinação e ajuste desse parâmetro é uma tarefa bastante árdua que pode demandar bastante tempo em função da complexidade do problema em questão.

Segundo experiências relatadas em [60], a utilização de funções de penalização pode gerar uma forma eficaz de se tratar as restrições, desde que a região factível seja “larga” ou quando o problema apresenta uma “suavização” suficiente do espaço de busca. Se o problema apresentar essas características, este é o melhor caminho para o tratamento das restrições.

A função aptidão modificada pode ser colocada na seguinte forma:

$$F(x) = f(x) + penal(x)$$

Na maioria dos métodos a parcela $penal(x)$ são funções de penalização, que diferem entre si de acordo com o método e a maneira em que são aplicados às soluções infactíveis.

Em [61, 62, 63] são descritos vários métodos e alguns são apresentados resumidamente neste texto, a seguir.

Penalidades estáticas

O método proposto em [64] sugere a determinação de intervalos com distintos valores de parâmetros de penalização associados a cada restrição f_j .

A idéia é definir vários níveis de violação (l). Para cada nível de violação e para cada restrição criar um coeficiente de penalização definido por R_{ij} , com $i = 1, 2, \dots, l$ e $j = 1, 2, \dots, m$, onde m é o número de violações.

A função aptidão modificada é dada por:

$$F(x) = f(x) + \sum_{j=1}^m R_{ij} \times f_j^2(x)$$

Esta é uma estratégia bastante interessante que evita penalizar de forma mais rigorosa pequenas violações, mas que tem a desvantagem de requerer um maior número de parâmetros a serem ajustados gerando-se uma maior complexidade na aplicação prática do método.

Penalidades dinâmicas

Diferentemente do método das penalidades estáticas, são estabelecidos parâmetros de penalização dinâmicos que dependem, por exemplo, da geração em que se encontra o processo de evolução. Por exemplo, em [65] a função $F(x)$ pode ser colocada na forma:

$$F(x) = f(x) + (C \times t)^\alpha \sum_{j=1}^m f_j^\beta(x)$$

onde C , α e β são constantes e têm valores sugeridos $C = 0.5$ e $\alpha = \beta = 2$. Nota-se que o parâmetro de penalização $(C \times t)^\alpha$ cresce com o número de gerações t atingindo seu valor máximo no final do processo de evolução.

Penalidades adaptativas

Estratégias que envolvem adaptação dos parâmetros de penalização são de grande interesse em computação evolucionista. Os parâmetros de penalização podem ser definidos (adaptados) de acordo com informações obtidas durante o processo de evolução. São apresentados aqui dois métodos que utilizam funções de penalização adaptadas de acordo com informações do processo de evolução. O primeiro deles [66], propõe a seguinte função aptidão modificada:

$$F(x) = f(x) + \lambda(t) \sum_{j=1}^m f_j^2(x)$$

onde o parâmetro de penalização $\lambda(t)$ é adaptado a cada geração t de acordo com as seguintes regras:

$$\lambda(t+1) = \begin{cases} (\frac{1}{\beta_1})\lambda(t), & \text{se } b^i \in \mathcal{F} \text{ para todo } t-k+1 \leq i \leq t \\ \beta_2\lambda(t), & \text{se } b^i \notin \mathcal{F} \text{ para todo } t-k+1 \leq i \leq t \\ \lambda(t) & \text{em caso contrário} \end{cases}$$

onde b^i é o melhor indivíduo da geração i , \mathcal{F} é a região factível, $\beta_1 \neq \beta_2$ e $\beta_1, \beta_2 > 1$. Neste método o parâmetro de penalização da geração seguinte $\lambda(t+1)$ diminui se todos os indivíduos na geração t são factíveis, aumenta se todos são infactíveis e se mantém inalterado se o caso for diferente de um desses dois.

O segundo método, proposto em [67], utiliza uma parametrização que manipula regiões infactíveis “interessantes” próximas à região factível. A função de aptidão é dada por:

$$F(x) = f(x) + (F_{feas}(t) - F_{all}(t)) \sum_{j=1}^m (f_j(x)/q_j(t))^k$$

onde F_{all} representa a melhor solução, obtida até a geração t , sem consideração de penalização, F_{feas} representa a melhor solução factível e k é uma constante.

4.3.2 Tratamento das restrições em várias etapas

Em [60] apresenta-se uma estratégia que trata problemas com restrições em várias etapas:

- Evoluir uma população gerada aleatoriamente com o objetivo de minimizar a primeira restrição, até que uma certa porcentagem da população ϕ (estabelecida pelo usuário) seja factível em relação a essa restrição;
- Esta população é o ponto de partida para a fase seguinte da evolução que tem como objetivo a minimização da restrição seguinte. Durante esta etapa, os indivíduos que violarem a restrição anterior serão considerados falhas e não competirão no processo de evolução. O critério de encerramento do processo é, agora, a satisfação da restrição seguinte, pela mesma porcentagem ϕ da população;
- Repetir o processo até que todas as restrições sejam analisadas.

O inconveniente dessa estratégia é sua aplicação em problemas que apresentam um número maior de restrições o que levará inevitavelmente à uma evolução cara quanto ao número de avaliações e também quanto à escolha da ordem em que as restrições serão analisadas.

4.3.3 Um algoritmo genético com dois parâmetros de penalização

Em [68] é apresentado um algoritmo genético que incorpora uma estratégia para tratamento de restrições onde são utilizados dois parâmetros constantes p_1 e p_2 distintos que atuam independentemente em duas populações. A idéia é gerar dois conjuntos de soluções candidatas ao problema onde uma delas pertence ao grupo avaliado por p_1 e outra ao grupo avaliado por p_2 . Se $p_1 \neq p_2$, conseqüentemente, tem-se diferentes níveis de satisfação das restrições nos dois grupos. As vantagens, nesta estratégia, podem ser representadas pelo fato de existirem diferentes trajetórias em que os grupos se direcionam e também pelo fato de que em problemas com restrições a solução está na fronteira entre a região factível e a infactível. Logo, isto pode ser alcançado adotando-se, por exemplo, $p_1 \gg p_2$, esperando-se encontrar uma região factível no grupo avaliado por p_1 e uma região que contém elementos infactíveis no grupo avaliado por p_2 .

Resumidamente, a estratégia pode ser entendida como:

- Gerar $2 \times$ POPSIZ indivíduos aleatoriamente (POPSIZ é o tamanho da população);
- Avaliar cada indivíduo para os 2 parâmetros de penalização e ordená-los em duas listas;
- Criar uma única lista ordenada de tamanho POPSIZ;
- Aplicar operadores genéticos;
- Avaliar novos filhos ($2 \times$ POPSIZ), com os parâmetros p_1 e p_2 gerando uma nova população;
- Verificar critério de convergência.

4.3.4 Operadores especiais

São propostas na literatura estratégias que utilizam operadores especiais para o tratamento de problemas com restrições. Em [69] são discutidos operadores chamados “operadores de fronteira” que basicamente têm como objetivo principal buscar e explorar os limites (fronteira) entre a região factível e a infactível. De posse de informações sobre a fronteira são desenvolvidos operadores que objetivam manter os novos indivíduos na região factível.

Outros métodos e aplicações para tratamento de problemas com restrições podem ainda serem encontrados em [70, 71, 72].

4.4 Adaptação

Adaptação de parâmetros e operadores é uma das mais importantes e promissoras áreas dentro do contexto da computação evolucionista. Ela não é utilizada somente para encontrar soluções ótimas para os problemas, mas também para melhorar o desempenho dos algoritmos.

Quando pretende-se analisar um problema utilizando-se algum tipo de algoritmo evolucionista, não basta simplesmente a escolha do tipo de algoritmo, mas além disso necessita-se a definição de parâmetros a serem utilizados no mesmo. Tais parâmetros podem ser, por exemplo, o tamanho da população, probabilidades de recombinação e mutação bem como as probabilidades específicas dos operadores. A escolha da parametrização de forma automática não é trivial sendo um procedimento trabalhoso que necessita um esforço considerável.

A adaptação da parametrização pode ser classificada [73, 74] quanto ao tipo e quanto ao nível. Quanto ao tipo se dá em duas categorias: externa e dinâmica. A dinâmica pode ser subdividida em exógena, adaptativa e auto-adaptativa. Quanto ao nível a adaptação de parâmetros se dá em quatro categorias: ambiente, população, indivíduo e componente.

4.4.1 Parametrização externa

A parametrização é mantida constante durante toda a execução do algoritmo. As alterações somente são permitidas através de mecanismos externos podendo ser através

do usuário ou de um programa específico (meta-algoritmo) [75]. Às vezes necessita-se de um esforço externo e exclusivo do usuário, dependendo de uma infinidade de testes podendo muitas vezes tornar-se proibitiva devido ao número de parâmetros a serem ajustados.

4.4.2 Parametrização dinâmica

Exógena

Neste caso a adaptação da parametrização é feita respeitando-se uma regra externa (fixa), que não está associada a nenhum tipo de informação obtida durante o processo de evolução. É muito comum a utilização de regras que utilizam uma variável “tempo” que pode representar as gerações, por exemplo. A mutação não-uniforme proposta por Michalewicz [8] é um exemplo deste caso. Uma outra utilização seria a definição de parâmetros de penalização em problemas com restrições onde esses têm seus valores alterados em função da geração [8].

Adaptativa

Acontece quando se utiliza qualquer tipo de retro-alimentação (*feedback*), obtida durante o processo de evolução definindo quanto e como se alterar a parametrização.

Na literatura existem técnicas que propõem uma estratégia para adaptação de operadores genéticos de recombinação e mutação. Estuda-se a distribuição de “créditos” inerente à eficiência do operador (produtividade), onde os mais eficientes têm suas probabilidades aumentadas e os menos, diminuídas [76]. A dificuldade é saber quem receberá crédito, quando e quanto.

A distribuição de créditos pode ser feita de várias maneiras. Uma seria proporcional ao ganho que pode se dar quando o filho gerado tem aptidão melhor que um certo valor de referência. A referência pode ser local [77] quando o filho tem aptidão melhor que a dos pais, por exemplo, ou global quando a aptidão é melhor que um certo valor definido como referência como, por exemplo, a aptidão média ou mediana. O valor do crédito a receber pode ser proporcional ao ganho de aptidão ou um valor fixo independente do quanto se melhorou a aptidão [78, 79, 80, 81].

A distribuição de créditos pode considerar a ancestralidade ou descendência dos operadores fazendo com que “pais”, “avós”, “bisavós”, etc, também sejam recompensados pelo sucesso de seus descendentes.

Entretanto, alguns cuidados devem ser tomados quando adotam-se políticas de distribuição de créditos pois pode acontecer que em um dado momento da evolução um certo operador pode ter um desempenho extremamente favorável fazendo com este receba um crédito bastante significativo. Sua probabilidade deve aumentar de forma significativa e conseqüentemente será mais utilizado. Apesar de ser bastante solicitado devido a sua alta probabilidade pode não ser mais tão eficiente quanto naquele momento “fértil” do “passado” e outros operadores, talvez agora mais adequados, podem não estarem sendo utilizados com mais frequência por terem probabilidades baixas. Para minimizar este tipo de problema introduz-se um parâmetro que considera a “memória” do processo de evolução que pode ser feito estabelecendo-se um limite de avaliações para que os operadores sejam reavaliados (“caso janela” [76]), ou distribuindo-se créditos considerando-se uma ponderação (parâmetro c), de

valores acumulados no passado (“caso contínuo” [77]), dada por:

$$W_{t+1} = W_t(1 - c) + cR_t$$

onde W_t é o crédito acumulado do operador até o instante t , em função do seu desempenho, R_t é o crédito a ser dado no instante t e c é o parâmetro de memória. O seu valor deve ser escolhido no intervalo $[0, 1]$. O extremo 0 considera a memória total – o passado é considerado totalmente e a distribuição de créditos no presente torna-se sem efeito – e o extremo 1 não considera memória – somente o presente é considerado. Em [77] são discutidos vários experimentos considerando-se valores intermediários do parâmetro c no intervalo $[0, 1]$.

Auto-adaptativa

Baseia-se na “evolução da evolução” para se definir a auto-adaptação da parametrização. A parametrização é codificada dentro dos cromossomos juntamente com as outras variáveis do problema. Essa codificação conjunta não altera as medidas de aptidão, mas, boas parametrizações podem acarretar bons indivíduos e esses podem estar mais aptos à sobrevivência e reprodução. Consequentemente, também propagando para a prole boas parametrizações.

4.4.3 Adaptação do ambiente

Durante o processo de evolução a relação entre o ambiente e o indivíduo é modificada como no caso de problemas com restrições onde parâmetros de penalização são alterados tornando o ambiente “mais rigoroso” (penalização maior), ou “menos rigoroso” (penalização menor).

4.4.4 Adaptação da população

Em geral as parametrizações são globais e válidas para todos os elementos da população. Definindo-se parametrizações diferentes para grupos diferentes da população tem-se uma adaptação da população.

4.4.5 Adaptação do indivíduo

Ocorre quando a modificação de parametrização afeta um único indivíduo da população, por exemplo, a adaptação de pontos de crossover [82] exclusiva para alguns ou todos indivíduos.

Um exemplo interessante também é encontrado em [83] onde adapta-se um parâmetro que representa a idade do indivíduo acarretando consequentemente uma adaptação do tamanho da população dado por:

$$Popsiz(t + 1) = Popsiz(t) + N(t) - D(t)$$

onde $D(t)$ é o número de indivíduos que morrem e $N(t)$ é o número de filhos que são gerados na geração t , respectivamente.

4.4.6 Adaptação da componente

Ocorre um ajuste local definido para um gene (componente), ou grupo de genes dentro do indivíduo. Um exemplo simples é quando se tem valores de taxa de mutação específicos para cada componente do cromossomo.

4.5 Diversidade da população

Segundo classificação proposta em [84] as medidas de diversidade podem ser divididas em dois grupos: um referente às diversidades genóticas e outro às fenóticas. Enquanto o primeiro concentra-se no material genético da população o segundo concentra-se na aptidão dos indivíduos da população.

4.5.1 Diversidade genotípica

Existe uma infinidade de medidas que podem representar a diversidade no espaço de genótipos. Elas podem ser medidas de dispersão baseadas em dados estatísticos, distância de Hamming, medidas de entropia, etc.

Diversidade genotípica através de medida Euclidiana

Comumente utilizada em AGs com codificação real é dada pela seguinte expressão:

$$me = \frac{\bar{d} - d_{min}}{d_{max} - d_{min}}$$

onde:

$$\bar{d} = \frac{1}{popsize} \sum_{i=1}^{popsize} d(C_{best}, C_i)$$

$$d_{max} = \max \{d(C_{best}, C_i) | C_i \in P\}$$

$$d_{min} = \min \{d(C_{best}, C_i) | C_i \in P\}$$

d_{max} representa a distância do cromossomo mais afastado do melhor indivíduo e d_{min} a distância do cromossomo mais próximo do melhor indivíduo. Se me está próximo de 0 significa que muitos cromossomos estão próximo do melhor indivíduo e existirá convergência.

Diversidade genotípica através da distância de Hamming

É proposta uma medida de diversidade por distância de Hamming (DH) em [85] onde determina-se uma média de DH entre todos os cromossomos que não sejam duplicatas. O custo computacional pode ser alto caso exista uma população grande com cromossomos longos. Uma alternativa é proposta em [86] onde determina-se DH entre o melhor e o pior indivíduo da população.

Diversidade genotípica por medidas de dispersões estatísticas

Estas medidas de diversidade podem ser dadas de vários parâmetros de dispersões estatísticas. Mais detalhes sobre essas medidas podem ser obtidos em [84].

4.5.2 Diversidade fenotípica

As medidas de diversidades fenotípicas são obtidas através de valores de aptidão no espaço de fenótipos. Em geral, são calculadas usando-se valores de aptidão do melhor indivíduo (f_{best}), aptidão do pior indivíduo (f_{worst}), aptidão média (\bar{f}), aptidão da mediana da população (f_{med}), entre outras. Algumas podem ser dadas por f_{best}/f_{med} , f_{med}/f_{worst} , propostas em [87], cujos valores devem estar no intervalo $[0, 1]$.

Uma outra medida de diversidade pode ser encontrada em [8] definida por:

$$l = \frac{\sqrt{\frac{1}{popsiz-1} \sum_{i=1}^{popsiz} (f_i - \bar{f})^2}}{\frac{1}{popsiz} \sum_{i=1}^{popsiz} f_i}$$

Outra medida de diversidade pode também ser dada por:

$$dvs = \frac{f_{max} - f_{med}}{f_{max} - f_{min}}$$

onde f_{max} e f_{min} representam os valores de aptidão máxima e mínima, respectivamente e f_{med} representa o valor da aptidão mediana. Se esta razão estiver próxima de 1 significa pouca diversidade na população.

Uma outra medida seria a razão entre a aptidão máxima e a média f_{max}/\bar{f} , onde valores próximos de 1 também indicam pouca diversidade na população.

Em [88] utiliza-se uma medida para se avaliar a diversidade e convergência, dada pela diferença entre a aptidão do melhor indivíduo e a aptidão média \bar{f} , $f_{best} - \bar{f}$.

4.6 Hibridização e busca local

4.6.1 Hibridização

É sabido que algoritmos genéticos são mais eficientes na determinação de regiões que apresentam soluções ótimas globais do que métodos clássicos baseados em técnicas de gradiente. Entretanto, após descoberta a região que contém a solução ótima global o algoritmo genético pode não ser tão eficiente na sua localização. Neste ponto os métodos clássicos de busca podem mostrar-se mais eficientes. Para tentar solucionar este tipo de problema adotam-se algumas estratégias que são conhecidas como *buscas locais* e que podem ser implementadas de várias maneiras. Em geral, podem ser utilizados operadores especiais dentro do algoritmo genético ou uma associação a outro método já existente de eficiência comprovada. Em ambos os casos as estratégias implementadas podem variar de acordo com o problema em questão. A associação a um método já existente é conhecida como hibridização.

Por quê hibridizar?

Segundo [2] algumas questões importantes devem ser consideradas e justificam uma resposta afirmativa para esta questão:

- hibridizando-se algoritmos genéticos com outros algoritmos aplicáveis ao problema pode-se gerar um algoritmo melhor que cada um deles isoladamente;
- hibridização pode acarretar novas heurísticas para otimização;

4.6.2 Busca local

Busca-se na vizinhança do indivíduo um elemento mais apto. Por exemplo, em uma codificação binária a forma mais simples de se fazer uma busca local é trocar os bits de 1 para 0 e 0 para 1 bit a bit recodificando-se o cromossomo e calculando-se uma nova aptidão para cada troca efetuada[7]. Isto é feito até que todos os bits sejam trocados. A cada passo, se a nova aptidão for melhor que a anterior ela é adotada. Neste ponto cabe uma observação: duas opções podem ocorrer: a primeira é de que o cromossomo pode ser mantido com a codificação original e somente a aptidão é alterada e, na outra além da nova aptidão adota-se também a recodificação para o cromossomo, conhecida como Lamarckianismo.

São encontradas na literatura outras formas de busca. Por exemplo, a proposta [89] que ocorre em três etapas:

1. *superset test*: este teste é parecido com a estratégia anterior com a exceção de que as mudanças ocorrem somente nos bits no sentido de 0 para 1;
2. *substitution test*: este troca simultaneamente os valores de dois bits. Troca-se os valores de 0 para 1 e 1 para 0. Permanece o valor de maior aptidão assumindo-se a recodificação do cromossomo original;
3. *subset test*: este é similar ao *superset test*, sendo que agora somente são trocados os bits no sentido de 1 para 0.

Nos casos anteriores, os procedimentos de busca são terminados somente quando todos os bits forem submetidos a pelo menos uma troca tornando-se estratégias caras.

Uma outra opção seria a utilização da técnica denominada “*first ascent*” onde o procedimento de busca local é interrompido na primeira troca que produz melhora.

Em [90], são sugeridas duas estratégias de busca através da implementação de um operador utilizado em problemas com restrições. Basicamente, estas estratégias combinam características favoráveis de soluções factíveis e infactíveis em uma dada população e em uma certa geração, evitando a introdução de parâmetros de penalização associados às violações das restrições.

Sejam, por exemplo, os seguintes cromossomos referentes a dois elementos da população, um factível e outro infactível:

10011101000000111	factível - cromossomo A
01011111011100100	infactível - cromossomo B

O operador tem a função de substituir um bit em uma certa posição do cromossomo B (infactível), pelo correspondente valor do cromossomo A (factível) com uma probabilidade p_E .

- Estratégia 1:

1. Gerar população aleatoriamente;
2. Avaliar elementos na população;
3. Combinar elementos infactíveis com factíveis da seguinte maneira:
 - (a) Identificar o melhor elemento factível por X_{best} ;

- (b) Classificar os N elementos inactíveis de acordo com suas violações às restrições, ou seja, o que mais violar assume o último lugar na lista;
- (c) Combinar cada elemento inactível com X_{best} . A probabilidade p_E de haver troca em um bit é dada de acordo com a classificação no (*rank*) do elemento inactível da seguinte maneira: definindo-se g_i^E como o bit da i -ésima posição do elemento inactível que se deseja definir, g_i^B como o bit da i -ésima posição do X_{best} e g_{ij}^V o bit do j -ésimo elemento inactível, tem-se:

$$g_i^E = \begin{cases} g_i^B & \text{se } r_i < R_j \\ g_{ij}^V & \text{se } r_i \geq R_j \end{cases}$$

$$j = 1, \dots, N$$

onde r_i é um inteiro gerado aleatoriamente entre 1 e o tamanho da população e R_j é a posição no *rank* do j -ésimo elemento inactível.

4. Avaliar elementos recodificados;
 5. Inserir novos elementos factíveis na população no lugar dos factíveis com menor aptidão;
 6. Combinar factíveis com inactíveis e completar a nova população;
 7. Aplicar operadores de recombinação e mutação na nova população;
 8. Repetir o processo até a convergência.
- Estratégia 2: A estratégia 2 é basicamente a mesma da 1 exceto pelo item 3. Ela apresenta um aspecto mais seletivo que o anterior e se dá pela combinação de elementos factíveis com inactíveis que têm valores similares de aptidão. Sendo J o elemento inactível e I o factível, tem-se:

$$\delta_{IJ} = F(I) - F(J)$$

onde $F(I)$ é a função objetivo do elemento I e $F(J)$ é a função objetivo do elemento J . Seleciona-se, desta forma, o elemento factível I mais próximo do inactível J . Identificados os elementos, o valor de g_i^E , é dado por:

$$g_i^E = \begin{cases} g_i^{BF} & \text{se } r_i < p_E \\ g_{ij}^V & \text{se } r_i \geq p_E \end{cases}$$

$$j = 1, \dots, N$$

Onde g_i^{BF} é o i -ésimo bit do elemento factível selecionado, g_{ij}^V é o correspondente bit no elemento inactível e r_i é um número aleatório escolhido entre 0 e 1. Os passos restantes são idênticos aos da estratégia 1.

4.6.3 Lamarckianismo e Efeito Baldwin

Em estratégias evolucionistas, com sustentação na biologia, existe uma diferença bem definida entre espaço de genótipos (G) e espaço de fenótipos (F). Um elemento do espaço de fenótipos, também denominado como indivíduo, é obtido por um mapeamento a partir do espaço de genótipos. Este mapeamento é realizado por funções de desenvolvimento ou funções de decodificação [91]. Durante o processo de desenvolvimento não existe qualquer tipo de influência a partir dos elementos restantes da população. A função $f(g)$ na Figura 4.5 representa uma função de desenvolvimento. O mapeamento só acontece do espaço de genótipos para o espaço de fenótipos exatamente como ocorre na natureza. No espaço de fenótipos, o conhecimento refere-se a uma plasticidade a qual o indivíduo pode adquirir, provocando seu “amadurecimento” tal como evidenciado por processos adaptativos encontrados na natureza.

Entretanto, o biólogo Jean Batiste Lamarck em 1800 em seu discurso sobre a teoria da evolução, defendia um mapeamento inverso ($f^{-1}(f')$), como ilustra a Figura 4.5, do espaço de fenótipos para o espaço de genótipos. Para ele as características obtidas através do conhecimento ou “maturação” (m), do indivíduo no ambiente em que vive poderiam ser herdadas pelas gerações seguintes. Dessa forma haveria uma alteração do material genético do indivíduo em função do conhecimento obtido durante a sua existência e sua defesa se baseava no exemplo clássico em que tentava explicar a existência dos pescoços longos das girafas.

Em algoritmos genéticos, denomina-se *busca local Lamarckiana* a modificação do genótipo ($f^{-1}(f')$), em alusão ao biólogo Lamarck. Este mapeamento inverso é possível de ser implementado em alguns casos. Por exemplo, se várias codificações do espaço de genótipos acarretam um único indivíduo no espaço de fenótipos como ilustra a Figura 4.6, o caminho inverso fica indefinido e não é possível realizar a busca Lamarckiana.

Segundo [91], algoritmos genéticos que utilizam *busca local Lamarckiana* são frequentemente mais eficientes que aqueles que não as usam, o que pode ser atraente em muitas aplicações.

O “efeito Baldwin” é uma teoria que se opõe ao Lamarckianismo e tem fundamentos puramente Darwinianos e foi introduzido independentemente por Baldwin, Morgan e Osborn no ano de 1896 [92]. Neste caso, ocorrem *buscas locais não-Lamarckianas* explorando informações obtidas por uma “busca fenotípica” sem utilização de modificações no código genético.

Lamarckianismo

No AG Lamarckiano em função de uma melhora da aptidão à sobrevivência por algum processo de aprendizado ou maturação do indivíduo existe a recodificação do genótipo que é inserido na população no lugar do antigo, competindo no processo de reprodução.

Efeito Baldwin

Diferentemente da situação anterior, qualquer tipo de conhecimento adquirido pelo indivíduo (fenótipo), não acarreta em uma recodificação de seu genótipo. Como na

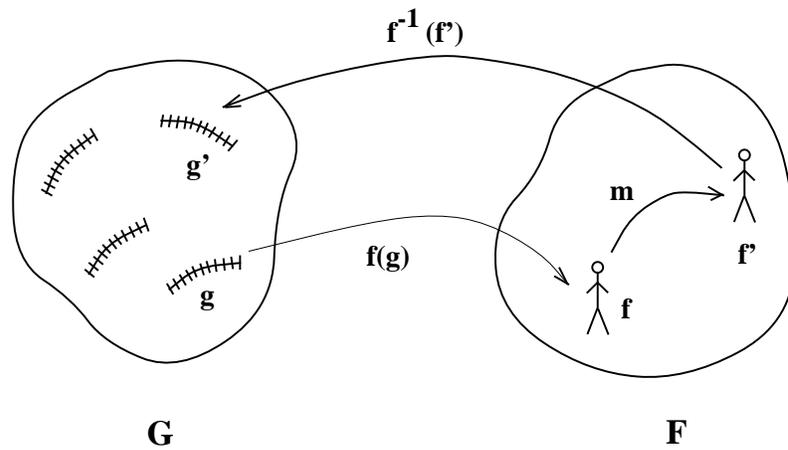


Figura 4.5: Desenvolvimento do genótipo e Lamarckianismo.

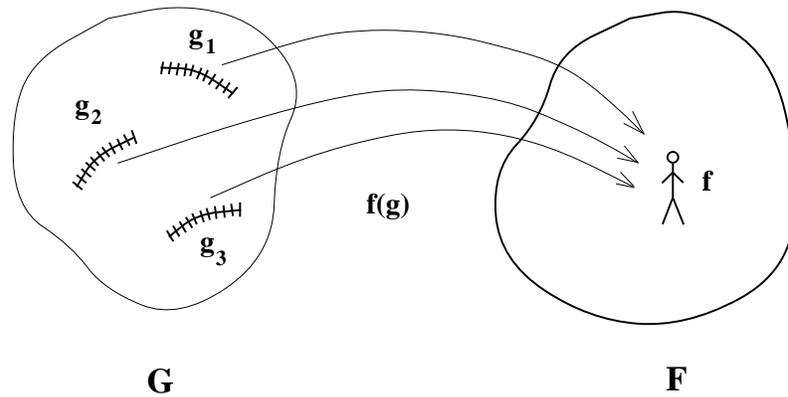


Figura 4.6: Mapeamento inverso não possível.

natureza, o indivíduo melhora sua aptidão à sobrevivência sem acarretar mudanças em seu cromossomo.

Em [92] são discutidos vários aspectos inerentes à aplicação do “efeito Baldwin” e entre ele destacam-se a plasticidade fenotípica, rigidez fenotípica, plasticidade do comportamento e plasticidade do conhecimento, descritos a seguir.

Plasticidade fenotípica

A plasticidade fenotípica tem a característica de suavizar a “paisagem” do espaço de busca objetivando condições melhores para se dar a evolução. Possibilita ao indivíduo uma melhor exploração do espaço e sua aptidão pode ser obtida considerando-se a aptidão do melhor indivíduo encontrado na sua vizinhança através de saltos ou aproximações, no espaço de busca.

Rigidez fenotípica

O organismo pode, lentamente, evoluir características de seu comportamento ou de sua estrutura física. Um aspecto interessante é o conhecimento proveitoso adquirido

pelas experimentações praticadas no meio ambiente, que podem ser bem ou mal sucedidas.

Plasticidade do comportamento

Explora-se o fato de que mudanças no comportamento de um indivíduo são mais fáceis de ocorrer do que mudanças em uma estrutura física. O organismo pode desenvolver formas alternativas para melhor se adaptar a um novo ambiente do que submeter-se às mudanças radicais em sua estrutura física. Por exemplo, o ser humano não está apto para receber fortes cargas de raios ultra-violeta mas, é suficientemente capaz para desenvolver protetores solares para suavizar tal inconveniente.

Plasticidade do conhecimento

Características instintivas encontradas em um ser vivo possibilitam a sua adaptação ao meio em que viverá. Entretanto, o conhecimento adquirido, que não tem o aspecto instintivo, poderá proporcionar ao indivíduo oportunidades maiores de adaptação e sobrevivência em um maior número de ambientes diferentes, o que talvez não seja possível somente com características instintivas.

4.7 Algoritmos Genéticos Paralelos

Os algoritmos genéticos podem ser aplicados a uma gama de problemas. Entre eles, muitos são complexos, principalmente na obtenção da medida da aptidão, gerando análises com alto custo computacional. Para minimizar esse custo são desenvolvidos vários tipos de algoritmos genéticos que utilizam recursos de computação paralela [93, 94, 95, 96, 97, 98, 99, 100, 101, 102].

A idéia principal concebida na maioria dos códigos computacionais com técnicas de paralelização é de simplesmente dividir o lote total de tarefas de natureza repetitiva em sub-lotes de tarefas e resolvê-los simultaneamente. Isto pode ser feito de várias maneiras.

Principais tipos de AGs paralelos:

- AG com paralelização do cálculo da aptidão;
- AG com modelo de “ilhas”;
- AG com *granulação fina*;
- AG misto.

4.7.1 AG com paralelização do cálculo da aptidão

Esta é a estratégia mais simples e de fácil implementação para se gerar um AG paralelo. Trabalha-se neste caso com uma única população subdividida com indivíduos avaliados simultaneamente. Este é um AG paralelo do tipo “mestre-escravos” como mostra a Figura 4.7. Podem haver diferenças no tempo de avaliação total de cada grupo de indivíduos em função das variações do tipo de arquitetura do(s) computador(s) utilizado(s). Neste caso pode-se ter um único computador com arquitetura

paralela ou redes de estações homogêneas ou heterogêneas, dedicados ou não a um único usuário.

As operações de seleção, recombinação e mutação são feitas na população inteira pelo mestre e a única diferença deste AG paralelo para o sequencial está no tempo de processamento. Aos escravos cabe a obtenção das aptidões dos indivíduos pertencentes ao grupo designado para cada um deles.

Podem ocorrer no processo de comunicação entre os escravos e o mestre um tempo de espera para que o mestre tenha os dados de todos os grupos e, assim, duas decisões podem ser tomadas: uma em que o mestre espera por todos os dados dos grupos, tendo-se um AG síncrono e outra em que o mestre não espera por todos os dados tendo-se um AG assíncrono. No caso de assincronismo o AG paralelo dá origem a um processo evolutivo distinto do caso sequencial.

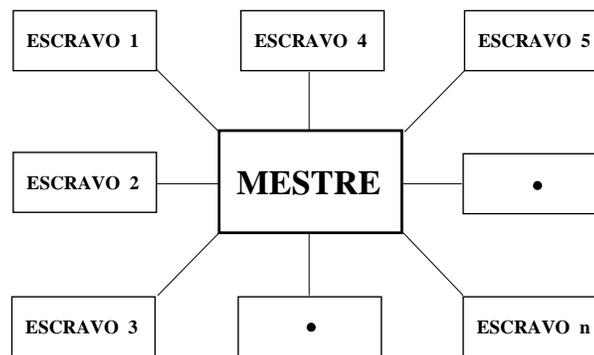


Figura 4.7: AG paralelo do tipo “mestre-escravos”.

4.7.2 AG com modelo de “ilhas”

Este tipo de algoritmo é mais sofisticado que o do tipo “mestre-escravos”. A população é subdividida em várias sub-populações mas agora as operações de seleção, recombinação e mutação são feitas localmente em cada sub-população, ou seja em cada “ilha”. Um aspecto interessante neste caso é a possibilidade de troca ou migração de indivíduos entre as sub-populações. Uma possível topologia é a do tipo “anel” mostrada na Figura 4.8 mostrando as possibilidades de migração. Deve haver para esse modelo uma parametrização que controla quando haverá migração, qual o seu sentido e quantos serão os indivíduos que migrarão.

4.7.3 AG com *granulação fina*

Neste caso os indivíduos distribuem-se espacialmente num dado arranjo com uma certa topologia de ligações (vizinhança). Um exemplo seria o “toro” bidimensional mostrado na Figura 4.9. A seleção ocorre localmente onde cada indivíduo se reproduz com outro selecionado numa certa vizinhança sua, como mostra a Figura 4.10. A avaliação pode ser síncrona ou não.

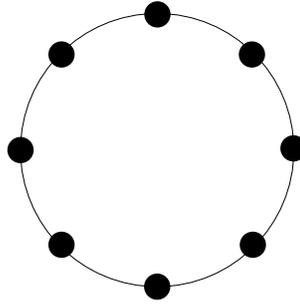


Figura 4.8: AG paralelo com granulação grossa.

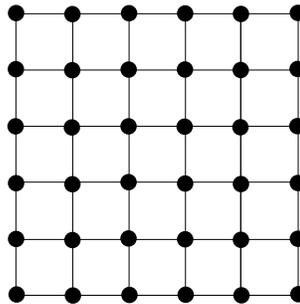


Figura 4.9: AG paralelo com granulação fina.

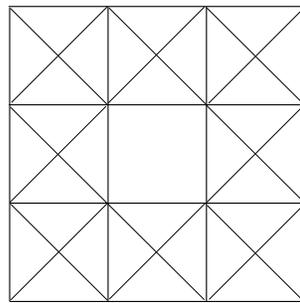


Figura 4.10: Escolha do vizinho para reprodução – O quadrado central representa o indivíduo escolhido para reproduzir e o restante ao seu redor representam os possíveis elementos que podem participar da recombinação.

4.7.4 AG misto

Finalmente, os AGs mistos agrupam características dos vários tipos de AGs paralelos num único algoritmo. Por exemplo, um AG misto pode ser obtido através de uma técnica de paralelização para a obtenção da aptidão com um modelo de “ilhas”, como esquematizado na Figura 4.11 ou um modelo de “ilhas” associado a um AG paralelo com granulação fina como mostrado na Figura 4.12.

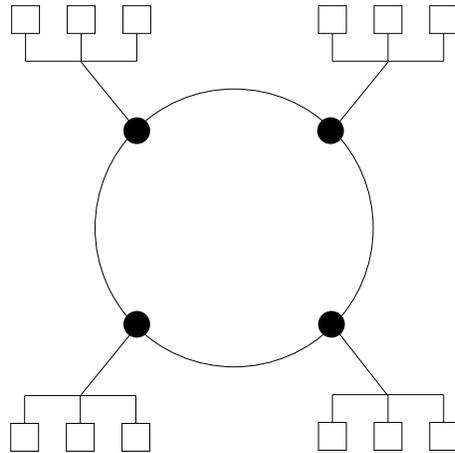


Figura 4.11: AG paralelo onde cada “ilha” tem aptidão avaliada em paralelo.

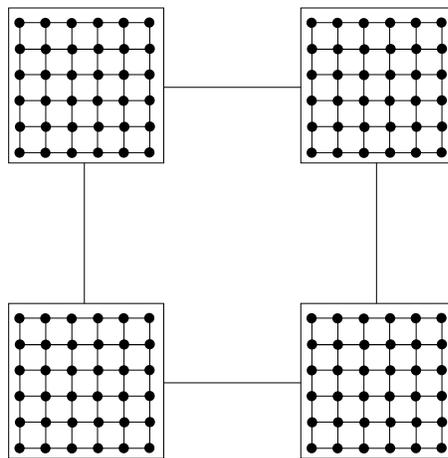


Figura 4.12: AG paralelo misto com modelo de “ilhas” associado a um AG com granulação fina.

Capítulo 5

Experimentos Numéricos

5.1 Introdução

Neste capítulo são discutidos alguns experimentos numéricos de aplicação dos algoritmos e formulações discutidas em capítulos anteriores deste texto. As implementações foram realizadas em linguagem de programação FORTRAN 77. Parte dos resultados foram publicados nas referências [103, 30, 104, 105].

5.2 Treliça de 10 barras

É frequente na literatura a discussão de exemplos de otimização de peso em treliças metálicas. Um exemplo clássico é a treliça de 10 barras mostrada na Figura 5.1. Este exemplo é exaustivamente estudado através de métodos clássicos e algoritmos genéticos [106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118].

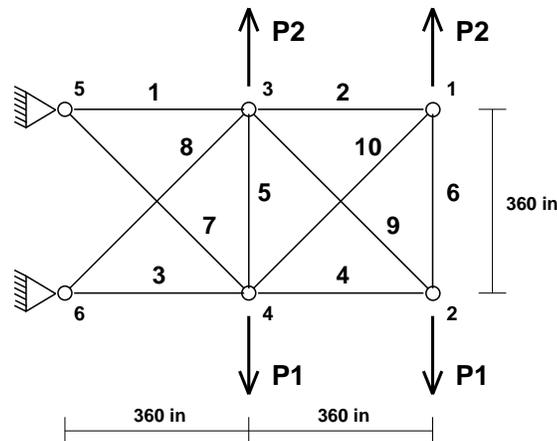


Figura 5.1: Treliça de 10 barras.

O problema de otimização pode ser colocado da seguinte forma: Encontre o conjunto de áreas $\mathbf{a} = \{A_1, A_2, \dots, A_n\}$ que minimiza o peso da estrutura

$$W(\mathbf{a}) = \sum_{i=1}^n \rho A_i L_i$$

onde L_i é o comprimento do i -ésimo membro da treliça e ρ a densidade do material. O problema está submetido às restrições de tensões

$$\frac{|\sigma_i|}{\sigma_{max}} - 1 \leq 0 \quad \forall i = 1, 2, \dots, n$$

onde n é o número de barras, e restrições de deslocamentos para todos os graus de liberdade globais u_j

$$\frac{|u_j|}{u_{max}} - 1 \leq 0 \quad \forall j = 1, 2, \dots, m$$

onde m é o número de graus de liberdade da estrutura.

A otimização dessa treliça será estudada através de vários problemas que apresentam variações entre si em função do carregamento, espaço de busca (variáveis contínuas ou discretas), restrições e tipos de algoritmos genéticos utilizados.

A Tabela 5.1 mostra as propriedades do material e os limites admissíveis para as tensões e deslocamentos. As cargas P1 estão aplicadas nos nós 2 e 4 e P2 nos nós 1 e 3 e suas intensidades variam de acordo com o problema.

propriedade	valor
Módulo de elasticidade	10.0 ⁷ psi
Massa específica	0.1 lb/in ³
Tensões admissíveis	± 25000.0 psi
Deslocamentos admissíveis	2.0 in

Tabela 5.1: Propriedades do material e restrições.

Para todos os problemas apresentados a seguir utilizou-se uma função aptidão modificada, a partir da função a ser otimizada, que incorpora uma função de penalização transformando o problema com restrições em um problema sem restrições. Essa função é dada pela seguinte expressão:

$$f(\mathbf{a}) = W(\mathbf{a}) \times (1 + penal * S)$$

onde a variável S é soma dos resíduos de cada restrição violada e $penal = 600$, para todos os problemas analisados neste exemplo.

Os resultados obtidos nos processos de otimizações envolvidos em todos os problemas deste exemplo referem-se ao melhor resultado entre todas as execuções realizadas nos algoritmos genéticos.

5.2.1 Problema P1

O Problema **P1** corresponde a uma análise feita considerando-se o carregamento definido com os valores de P1 = 100 Kips e P2 = 0, impondo-se restrições de deslocamentos máximos admissíveis de $u_{max} = \pm 2.0$ in para todos os nós em todas as direções e restrições de tensões para todas as barras. O espaço de busca é formado por variáveis contínuas e tem o limite inferior $A_{min} = 0.1$ in² e superior $A_{max} = 40.0$ in².

Utilizou-se um algoritmo do tipo “steady-state” com codificação real, com os seguintes operadores de recombinação: *crossover* de dois pontos, discreto, BLX- α ($\alpha = 0.3$), e o heurístico. Adotou-se uma mutação randômica. Todos os operadores foram definidos com probabilidades variáveis durante as avaliações e dadas na Tabela 5.2.

<i>prob</i>	cr. 2 pontos	cr. discreto	BLX- α	cr. heurístico	mut. randômica
<i>prob_i</i>	0.3	0.2	0.0	0.2	0.3
<i>prob_f</i>	0.1	0.3	0.2	0.2	0.2

Tabela 5.2: Probabilidades iniciais e finais dos operadores.

A probabilidade *prob* de um operador em uma determinada avaliação *naval* é dada pela seguinte equação:

$$prob = prob_i + (prob_f - prob_i) \times \frac{naval}{maxaval}$$

onde *prob_i* representa a probabilidade inicial do operador, *prob_f* a probabilidade final, *naval* o número da avaliação corrente e *maxaval* o número total de avaliações.

O gráfico da Figura 5.2 ilustra a variação das probabilidades dos operadores escolhidos para este problema.

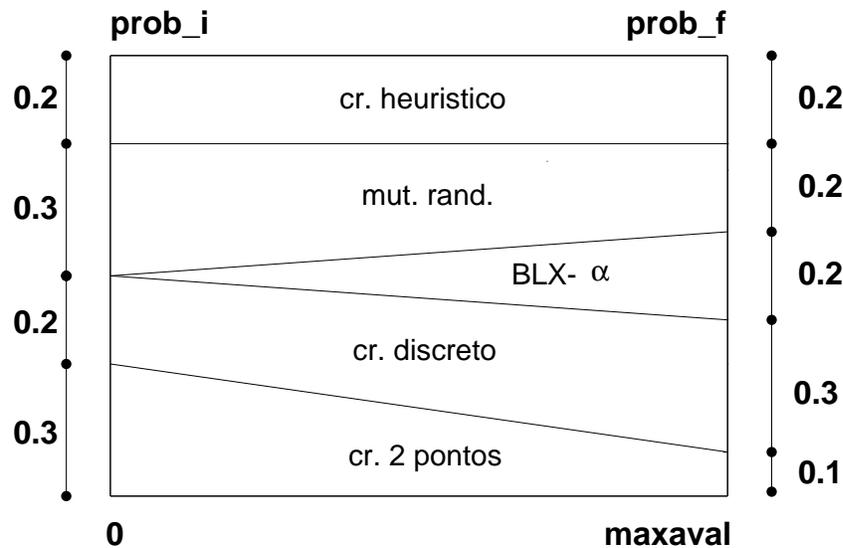


Figura 5.2: Probabilidades iniciais e finais dos operadores.

Foram definidas três populações iniciais com os tamanhos de 400, 500 e 600 elementos. Para as de 400 e 500 foram realizadas 20000, 30000 e 40000 avaliações e para a de 600, realizadas 21000, 30000 e 42000 avaliações. Para cada população inicial foram realizadas 10 execuções com 10 sementes diferentes cujos valores foram escolhidos aleatoriamente entre 0 e 1.

A Tabela 5.3 resume os valores obtidos na otimização, detalhados na Tabela 5.5.

avaliações	população		
	400	500	600
20000	5065.795	5064.539	-
21000	-	-	5066.664
30000	5062.059	5061.880	5063.907
40000	5076.684	5061.442	
42000	-	-	5061.330

Tabela 5.3: Resultados obtidos na otimização – peso da estrutura em lbs.

Para o Problema **P1** o melhor resultado obtido foi o da análise com uma população com 600 elementos e 42000 avaliações com o valor de 5061.330 lbs para o peso total da estrutura como mostra a Tabela 5.3.

Observa-se na Tabela 5.3 que para a população com 400 elementos e 40000 avaliações ocorreu um resultado pior do que os com 20000 e 30000 avaliações. Isto se deveu ao fato de terem sido adotadas probabilidades variáveis de operadores durante a evolução associados com diferentes tamanhos de populações e número de avaliações, gerando, então diferentes parametrizações que não garantem que uma população maior com um maior número de avaliações forneça melhores resultados. Este fato é observado também em análises seguintes deste exemplo, onde também são mostrados resultados detalhados de todas as análises realizadas.

A Tabela 5.4 totaliza o desempenho dos operadores genéticos para o conjunto das 10 execuções que inclui o melhor resultado da otimização. Nessa Tabela são apresentados o número de aplicações de cada operador e o número de sucessos obtidos, ou seja, quando cada um deles, conseguiu gerar um novo melhor indivíduo da população. A taxa é a razão entre o número de sucessos e o número total de aplicações que mostra que a mutação randômica, para este exemplo, foi o operador de maior eficiência com 212 sucessos e o *crossover* heurístico o de menor com 19 sucessos.

A Tabela 5.5 mostra os valores das áreas das barras A_i da treliça, referente a cada população inicial, “naval” representa o número de avaliações e “aval” o número da avaliação em que foi encontrado a melhor solução.

operador	núm. aplicações	sucessos	taxa
cr. 2 pontos	134720	184	1.3658×10^{-3}
discreto	87571	106	1.2104×10^{-3}
BLX- α	36346	33	0.9079×10^{-3}
heurístico	86203	19	0.2204×10^{-3}
mut. randômica	69161	212	3.0653×10^{-3}

Tabela 5.4: Estatística dos operadores - população = 600, avaliações = 42000.

Tabela 5.5: Resultados comparativos do Problema P1.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	31.50807	29.99145	30.77946	30.60234	30.22844	30.27438	30.28590	30.88026	30.45486
2	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
3	23.31610	23.23086	23.94125	23.15699	23.00941	23.23628	23.17837	23.45432	23.08036
4	15.03513	15.11057	14.74322	14.96404	15.40381	15.20492	16.19389	15.74594	15.05050
5	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
6	0.45738	0.53993	0.10000	0.62277	0.47992	0.54791	0.68535	0.48641	0.57944
7	7.54193	7.50561	8.53791	7.57811	7.50810	7.49426	7.41424	7.36322	7.49861
8	21.32678	21.54992	20.90450	20.99726	21.38214	21.41151	20.49145	20.36052	21.28839
9	20.67006	21.43084	20.84254	21.62502	21.41670	21.29224	21.63119	21.60101	21.47835
10	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
W(lb)	5065.795	5062.059	5076.684	5064.539	5061.880	5061.442	5066.664	5063.907	5061.330
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19248	29929	39927	19873	29561	38960	20904	28906	41441

5.2.2 Problema P2

Um fato que deve ser observado cuidadosamente no exemplo da treliça de 10 barras em estudo é de que além das restrições impostas em relação às tensões e deslocamentos admissíveis define-se um limite inferior de 0.1 in^2 para a área mínima das seções transversais das barras. Ao se utilizar alguns operadores de recombinação, BLX- α por exemplo, existe a possibilidade de escolha de um valor para um certo parâmetro do cromossomo que esteja fora dos limites estabelecidos para o mesmo, ou seja, menor que o limite inferior ou maior que o limite superior. Em alguns problemas este tipo de ocorrência pode ser indiferente em nada interferindo a busca. Entretanto, particularmente no caso em questão não se pode admitir a geração de uma estrutura onde uma barra tenha área negativa.

Nota-se pelos resultados da análise anterior que um número significativo de barras assumem os valores de 0.1 in^2 para a área de suas seções transversais. Por esta característica do problema e pelos operadores utilizados, provavelmente, foram geradas estruturas com algumas barras com seções transversais com valores negativos. Algumas opções para reparar este tipo de problema podem ser adotadas e basicamente duas escolhas podem ser feitas. Uma seria deixar a codificação da forma em que foi gerada (com valor negativo para áreas) e fazer uma projeção, como se fosse um processo de aprendizado do indivíduo, para o limite inferior somente no momento da avaliação, sem recodificar o cromossomo, explorando o “Efeito Baldwin”.

A outra opção seria o oposto e os cromossomos gerados após a recombinação seriam recodificados (lamarckianismo), para garantir que nenhuma barra tenha sua seção transversal com valor negativo. O Problema **P1** foi analisado adotando-se a opção de projeção para o limite inferior somente no momento da avaliação.

Decidiu-se, ainda, dentro da análise do Problema **P2** uma variação no processo de lamarckianismo utilizado. Foram feitos alguns experimentos imaginando-se uma recodificação aleatória, e para isso definiu-se uma probabilidade (prob.) de se recodificar o cromossomo após a operação de recombinação. Lembrando que para garantir o limite inferior para os valores das seções transversais sempre procedeu-se a uma verificação no momento da avaliação.

As Tabelas 5.6, 5.7 e 5.8 resumem os valores obtidos na otimização, detalhados nas Tabelas 5.9, 5.10 e 5.11, com probabilidades 0.1, 0.3 e 1.0 de recodificação, respectivamente.

avaliações	população		
	400	500	600
20000	5062.713	5071.952	-
21000	-	-	5065.884
30000	5062.692	5061.916	5063.386
40000	5061.324	5060.920	
42000	-	-	5061.119

Tabela 5.6: Resultados obtidos na otimização - prob.=0.1 – peso da estrutura em lbs.

Observa-se através desses resultados que os melhores valores para a otimização analisada no problema **P2**, foram com uma população inicial de 500 indivíduos

avaliações	população		
	400	500	600
20000	5062.985	5063.530	-
21000	-	-	5065.304
30000	5061.392	5061.484	5061.515
40000	5061.196	5060.984	
42000	-	-	5061.241

Tabela 5.7: Resultados obtidos na otimização - prob.=0.3 – peso da estrutura em lbs.

avaliações	população		
	400	500	600
20000	5064.471	5067.096	-
21000	-	-	5064.522
30000	5061.703	5061.957	5063.361
40000	5061.134	5061.032	
42000	-	-	5061.321

Tabela 5.8: Resultados obtidos na otimização - prob.=1.0 – peso da estrutura em lbs.

e um total de 40000 avaliações (5050.920 lbs), adotando-se uma probabilidade de recodificação igual a 0.1 como mostra a Tabela 5.6. Este resultado indica que mesmo com probabilidades maiores de recodificação, ou seja, 0.3 e 1.0, não foram alcançadas melhores soluções, o que poderia ter se tornado uma expectativa.

As Figuras 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10 e 5.11, mostram resultados comparativos, do peso da treliça durante o processo de evolução, entre os problemas **P1** (sem recodificação - prob. = 0) e **P2** (com recodificação - prob.=0.1, 0.3 e 1.0).

Tabela 5.9: Resultados comparativos do Problema P2 - prob.=0.1.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	30.24690	31.22010	30.74501	31.29434	30.79093	30.56091	30.99152	30.42410	30.52264
2	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
3	22.99815	23.58067	23.35654	23.30847	23.09824	23.16999	22.03626	23.58091	22.98128
4	15.28034	15.03023	14.93309	15.70378	15.27799	15.11224	15.35900	14.50070	15.13942
5	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
6	0.62241	0.60793	0.58570	0.61258	0.47821	0.54910	0.54581	0.53080	0.55567
7	7.49033	7.40407	7.44899	7.54626	7.48716	7.47048	7.58110	7.49271	7.47665
8	21.19814	20.57777	20.97934	21.24821	21.02268	21.09910	21.05323	21.37340	21.05390
9	21.61629	21.40957	21.51502	20.43923	21.42744	21.52714	21.88490	21.53059	21.70672
10	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
W(lb)	5062.713	5062.692	5061.324	5071.952	5061.916	5060.920	5065.884	5063.386	5061.119
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19503	29668	39450	19484	29900	39363	20333	29568	41027

Tabela 5.10: Resultados comparativos do Problema P2 - prob.=0.3.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	30.36927	30.36304	30.33009	30.73520	30.66295	30.46298	30.97641	30.37576	30.59562
2	0.10184	0.10002	0.10000	0.10920	0.10013	0.10000	0.10000	0.10000	0.10000
3	23.01809	22.99683	23.19911	23.37190	23.29230	23.14617	23.41872	23.19919	23.26966
4	15.63544	15.45859	15.50266	15.24374	15.13862	15.09405	15.19114	15.23543	15.21645
5	0.10207	0.10000	0.10000	0.10000	0.10012	0.10000	0.10000	0.10000	0.10000
6	0.53430	0.52976	0.54979	0.65613	0.50478	0.54768	0.39658	0.52704	0.61509
7	7.49119	7.46066	7.43951	7.47416	7.48188	7.47749	7.50277	7.48854	7.44091
8	21.27103	21.00526	20.94632	20.96140	21.18348	21.17240	21.10845	21.29987	20.90464
9	21.25571	21.67060	21.58235	21.27102	21.29772	21.54801	21.15398	21.35841	21.54210
10	0.10000	0.10057	0.10002	0.10021	0.10000	0.10000	0.10000	0.10034	0.10000
W(lb)	5062.985	5061.392	5061.196	5063.530	5061.484	5060.984	5065.304	5061.515	5061.241
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19296	28994	38630	19076	29606	39878	16806	28147	41583

Tabela 5.11: Resultados comparativos do Problema P2 - prob.=1.0.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	29.69128	30.64071	30.61077	29.42393	30.30948	30.59611	31.07248	30.76797	30.46216
2	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10002
3	23.71597	23.07979	23.12721	23.79172	23.50978	23.17079	22.89847	23.21570	23.47391
4	15.46022	14.80784	14.95542	14.76352	14.88539	15.19302	15.36411	15.32003	15.10480
5	0.10545	0.10006	0.10000	0.10027	0.10000	0.10000	0.10000	0.10000	0.10004
6	0.50258	0.55031	0.53406	0.43839	0.62674	0.59202	0.48602	0.57245	0.58554
7	7.35547	7.51582	7.47486	7.49584	7.43973	7.46628	7.62222	7.54048	7.42497
8	20.89625	21.32187	21.09044	21.43685	21.10680	21.05058	21.09494	20.92230	20.93994
9	21.92658	21.49605	21.65212	21.97434	21.61359	21.46913	21.14707	21.33959	21.57404
10	0.10000	0.10004	0.10001	0.10000	0.10000	0.10000	0.10000	0.10012	0.10000
W(lb)	5064.471	5061.703	5061.134	5067.096	5061.957	5061.032	5064.522	5063.361	5061.321
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19884	29883	39267	19890	28506	38479	19229	29132	40476

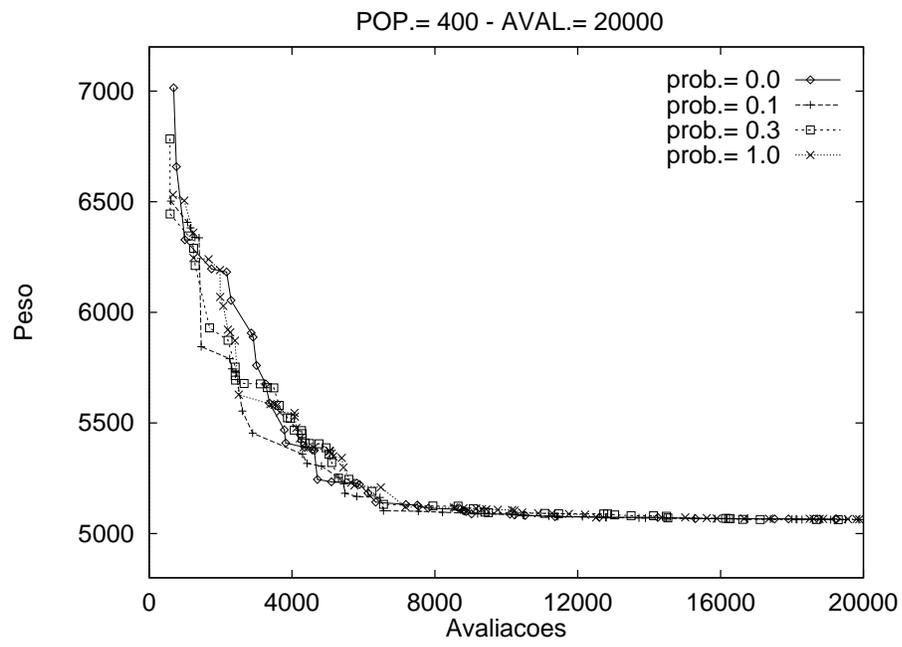


Figura 5.3: População com 400 elementos e 20000 avaliações.

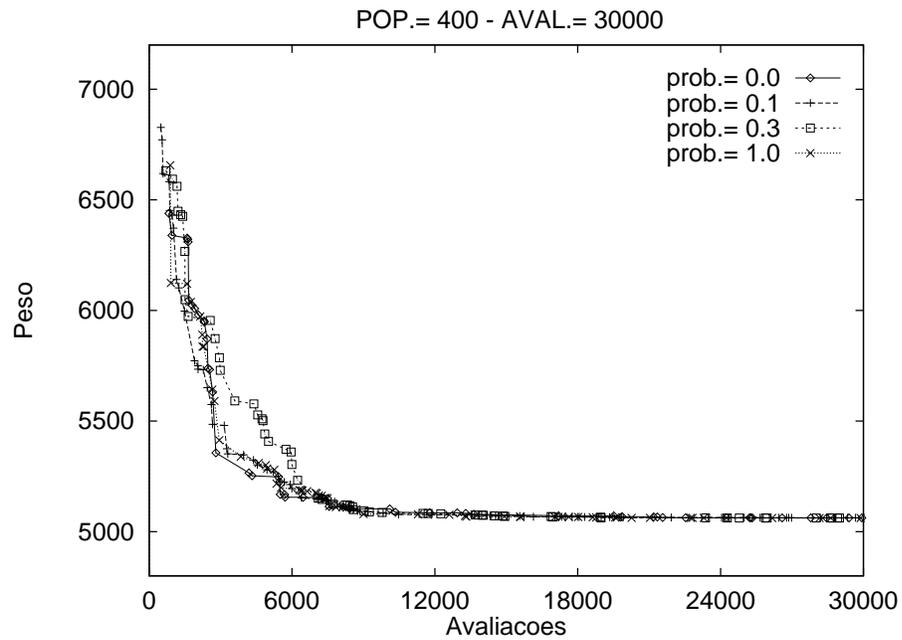


Figura 5.4: População com 400 elementos e 30000 avaliações.

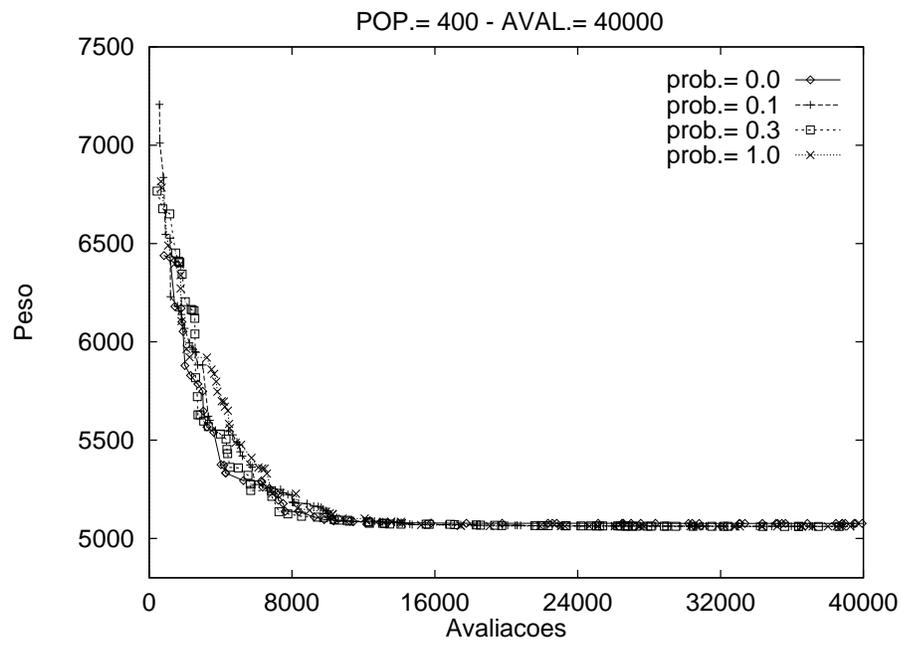


Figura 5.5: População com 400 elementos e 40000 avaliações.

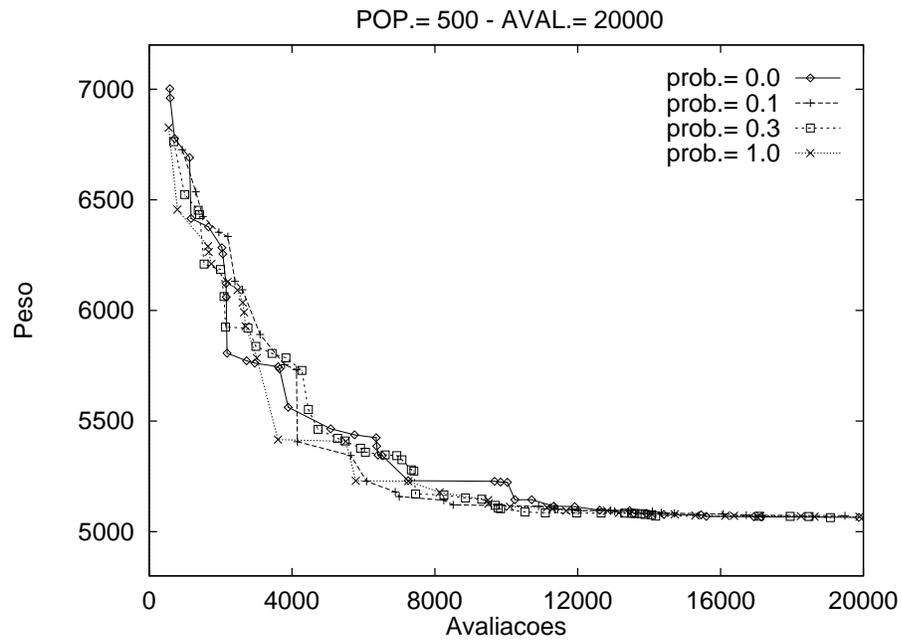


Figura 5.6: População com 500 elementos e 20000 avaliações.

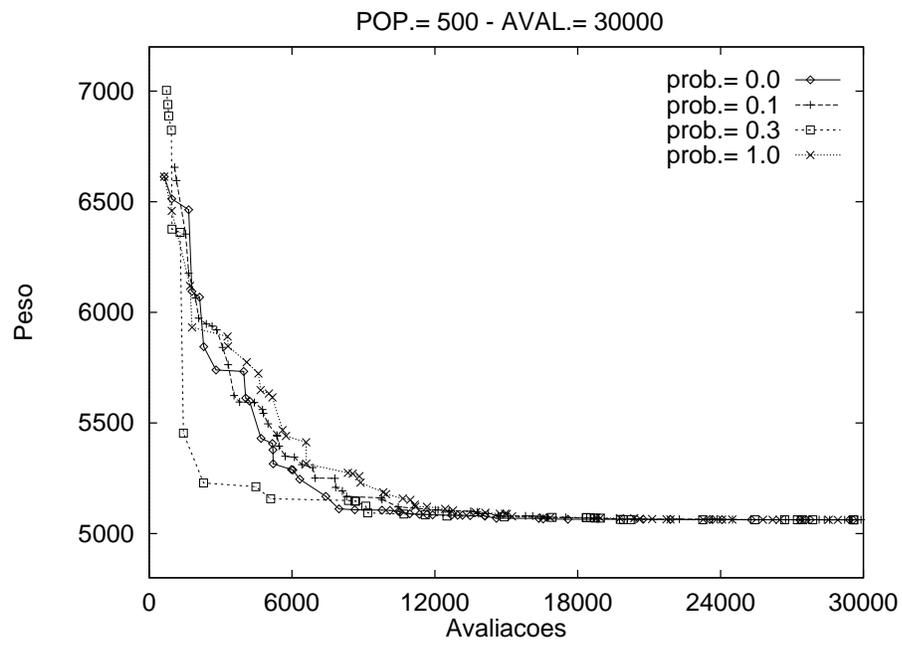


Figura 5.7: População com 500 elementos e 30000 avaliações.

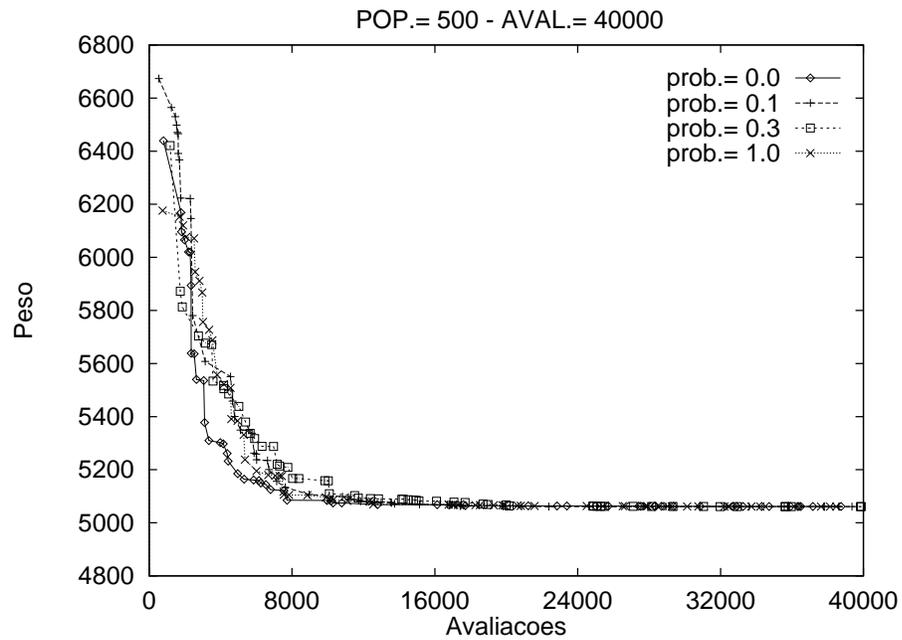


Figura 5.8: População com 500 elementos e 40000 avaliações.

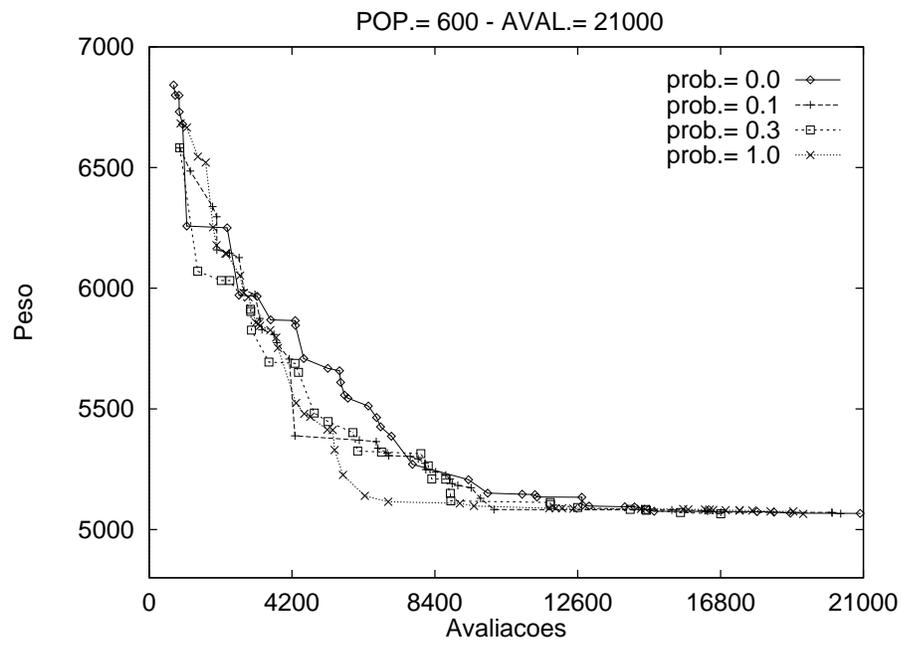


Figura 5.9: População com 600 elementos e 21000 avaliações.

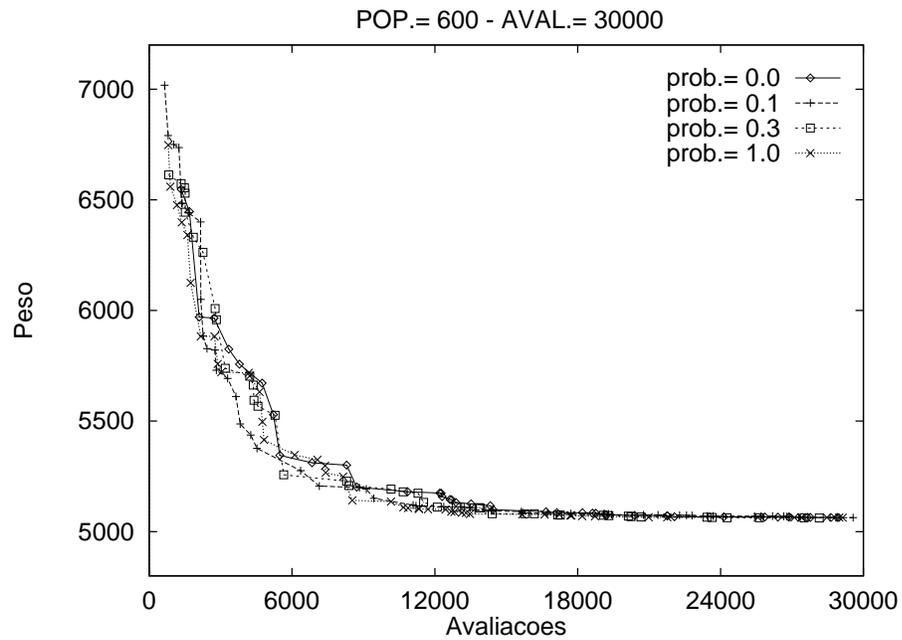


Figura 5.10: População com 600 elementos e 30000 avaliações.

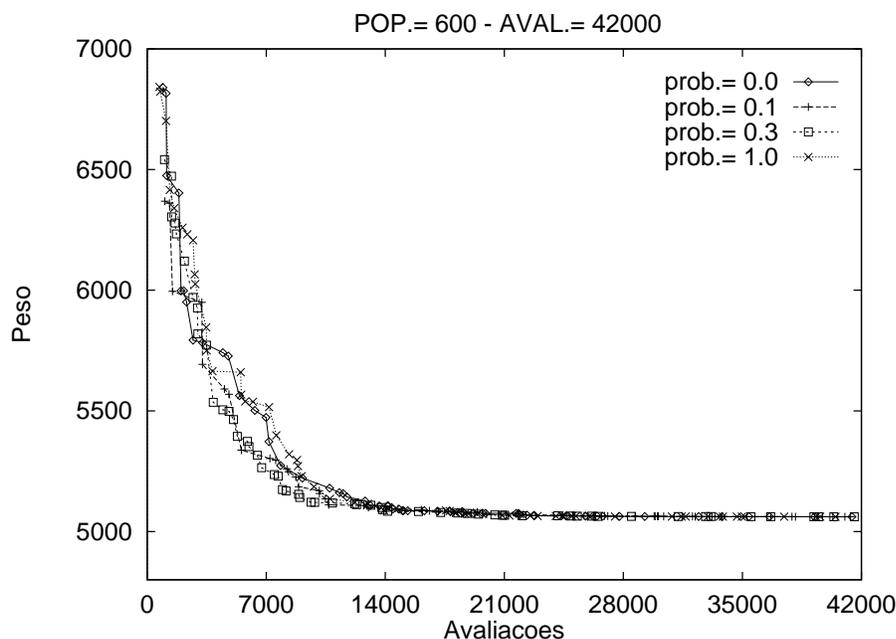


Figura 5.11: População com 600 elementos e 42000 avaliações.

5.2.3 Resultados comparativos - Problema P2

Serão relacionados e comparados resultados frequentemente encontrados na literatura com os obtidos nas análises do **P2** como mostram as Tabelas 5.12, 5.13 e 5.14. Além disso serão feitos alguns comentários e críticas em relação a alguns desses resultados.

Em geral, os resultados encontrados na literatura referentes a esta treliça fornecem os valores das seções transversais das barras com dois ou três algarismos significativos após o ponto (vírgula). Raramente são apresentados resultados com 4, 5 ou mais algarismos significativos após o ponto e isto pode ser justificado pelo fato de que as áreas são dadas em in^2 e uma precisão de dois algarismos já seria bastante rigorosa em uma fase de execução da treliça.

Entretanto, por se tratar de um exemplo clássico e de ser exaustivamente estudado por vários tipos de algoritmos alguns cuidados devem ser tomados tanto na obtenção quanto na apresentação de resultados.

Um detalhe importante está relacionado ao comprimento das barras adotados durante o momento de avaliação de uma possível solução. Alguns pesquisadores são flexíveis em relação à essas medidas. Para as barras 1, 2, 3, 4, 5, e 6 de comprimento 360 in^2 não haveriam problemas já que trata-se de um valor inteiro mas, para as barras de número 7, 8, 9 e 10 (inclinadas) necessita-se da adoção de uma precisão mais rigorosa evitando-se possíveis *arredondamentos* que acarretam um peso menor da estrutura. Nas análises realizadas neste trabalho adotou-se precisão dupla para a variável que representa o comprimento das barras inclinadas.

Um outro detalhe está relacionado às restrições impostas na formulação do problema de otimização, principalmente as associadas aos deslocamentos máximos permitidos, as quais têm sido violadas na apresentação de alguns resultados. Esta é uma

A_i	Ref.[106]	Ref.[119]	Ref.[108]	Ref.[109]	Ref.[110]	Ref.[107]
1	31.350	30.570	33.432	30.416	30.500	30.670
2	0.100	0.369	0.100	0.128	0.100	0.100
3	20.030	23.970	24.260	23.408	23.290	23.760
4	15.600	14.730	14.260	14.905	15.428	14.590
5	0.140	0.100	0.100	0.101	0.100	0.100
6	0.240	0.364	0.100	0.101	0.210	0.100
7	8.350	8.547	8.388	8.696	7.649	8.578
8	22.210	21.110	20.740	21.084	20.980	21.070
9	22.060	20.770	19.690	21.077	21.818	20.960
10	0.100	0.320	0.100	0.186	0.100	0.100
W	5112.00	5107.30	5089.00	5084.90	5080.00	5076.85

Tabela 5.12: Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.

A_i	Ref.[111]	Ref.[112]	Ref.[113]	Tese	Ref.[114]	Ref.[115]
1	30.731	30.980	30.0310	30.56091	30.520	30.350
2	0.100	0.100	0.1000	0.10000	0.100	0.100
3	23.934	24.169	23.2740	23.16999	23.200	23.620
4	14.733	14.805	15.2860	15.11224	15.220	15.950
5	0.100	0.100	0.1000	0.10000	0.100	0.100
6	0.100	0.406	0.5565	0.54910	0.551	0.760
7	8.542	7.547	7.4683	7.47048	7.457	7.260
8	20.954	21.046	21.1980	21.09910	21.040	19.670
9	21.836	20.937	21.6180	21.52714	21.530	22.100
10	0.100	0.100	0.1000	0.10000	0.100	0.100
W	5076.66	5066.98	5061.60	5060.920	5060.80	5053.00

Tabela 5.13: Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.

A_i	Ref.[115]	Ref.[116]	Ref.[117]	Ref.[117]	Ref.[118]	Ref.[115]
1	30.64	31.28	28.7878	32.9657	30.440	30.561
2	0.10	0.10	0.1000	0.1000	0.100	0.100
3	23.71	24.65	22.3483	22.7988	21.790	27.946
4	14.39	15.39	16.7847	14.1463	14.260	13.619
5	0.10	0.10	0.1000	0.1000	0.100	0.100
6	0.497	0.10	0.5516	0.7393	0.451	0.100
7	7.58	7.90	7.5633	6.3807	7.628	7.907
8	22.24	21.53	21.9861	20.9121	21.630	19.345
9	20.21	19.07	21.0175	20.9779	21.360	19.273
10	0.10	0.10	0.1000	0.1000	0.100	0.100
W	5052.50	5052.00	5051.75	5013.24	4987.00	4981.10

Tabela 5.14: Resultados comparativos entre a literatura e a melhor solução do Problema P2 – peso da estrutura em lbs.

questão de maior importância que não pode deixar de ser verificada pois dessa forma o problema original torna-se modificado. Além disso os limites máximos para os deslocamentos e tensões devem ser estabelecidos com variáveis definidas com precisão dupla, ou seja, neste caso, $u_{max} = 2.0d+00$ in e $\sigma_{max} = 25000.0d+00$ psi, no momento da avaliação de uma solução candidata.

Na Tabela 5.15, com resultados comparativos, tem-se “W apresentado” como valor do peso (correspondente à função objetivo apresentado nas referências, em lbs), “W calculado” (calculado nesta tese para todas as referências), representando o valor do peso calculado através das áreas apresentadas e utilizando-se o comprimento das barras inclinadas por uma variável de precisão dupla e u_{y2} e u_{y4} os deslocamentos verticais (eixo-y), dos nós 2 e 4 (restrições “mais representativas”).

Observando-se os resultados das comparações percebe-se que existem variações significativas no valor do peso apresentado e do peso calculado onde algumas têm peso calculado maior que o apresentado, outras com um simples truncamento e *arredondamento* na quantidade de algarismos significativos e, finalmente, algumas com peso apresentado maior que o calculado. Este é um detalhe importante pois trata-se de um exemplo clássico e por menores que sejam as variações estas são representativas.

Entretanto, o detalhe que mais deve ser observado é o relacionado com as violações às restrições. Nota-se que as referências [106, 119, 108, 109, 110, 111, 113, 114] e os resultados dessa tese satisfazem plenamente às restrições de deslocamentos e as referências [115, 116, 117, 118] não as satisfazem e dessa forma não podem ser comparados.

Nota-se que comparando-se os resultados obtidos nessa tese, para este exemplo, 5060.920 com os apresentados por [114] 5060.80 (supostamente os melhores) uma diferença em relação ao peso de 5060.80 (apresentado) para 5060.926 (calculado) tornando o resultado obtido nessa tese como sendo o melhor no estudo comparativo.

Referência	W apresentado	W calculado	u_{y2}	u_{y4}
[106]	5112.00	5112.624	-1.99999	-1.98696
[119]	5107.30	5107.323	-1.99991	-1.99998
[108]	5089.00	5091.569	-1.99844	-1.99982
[109]	5084.90	5084.809	-1.99860	-1.99998
[110]	5080.00	5080.041	-1.99997	-1.98164
[107]	5076.85	5077.149	-1.99989	-1.99988
[111]	5076.66	5127.617	-1.98233	-1.97777
[112]	5066.98	5066.907	-2.00007	-1.99116
[113]	5061.60	5061.659	-1.99999	-1.99149
Tese	5060.92	5060.920	-1.99999	-1.99147
[114]	5060.80	5060.926	-1.99996	-1.99138
[115]	5053.00	5052.971	-2.00713	-2.00024
[115]	5052.50	5051.935	-2.00758	-2.00031
[116]	5052.00	5052.628	-2.01949	-2.00657
[117]	5051.75	5051.744	-2.00904	-1.99999
[117]	5013.24	5013.237	-2.01312	-2.00000
[118]	4987.00	4999.215	-2.02798	-2.01850
[115]	4981.10	4981.093	-2.04994	-2.06047

Tabela 5.15: Pesos apresentados e calculados e deslocamentos correspondentes às soluções apresentadas nas referências e neste trabalho.

5.2.4 Problema P3

O Problema **P3** corresponde a uma análise feita considerando-se o carregamento definido com os valores de $P1 = 100$ Kips, $P2 = 0$ e impondo-se somente restrições de tensões. O tipo de algoritmo utilizado bem como os operadores e probabilidades são os mesmos dos Problema **P1** e **P2**. Baseando-se nos resultados anteriores, optou-se por um lamarkianismo aplicado com probabilidade 0.1.

A Tabela 5.16 resume os resultados encontrados na otimização apresentados na Tabela 5.20. A melhor solução (1593.184 lbs) foi obtida com uma população de 500 elementos e 40000 avaliações.

avaliações	população		
	400	500	600
20000	1593.516	1597.410	-
21000	-	-	1603.499
30000	1593.189	1593.258	1593.481
40000	1593.186	1593.184	
42000	-	-	1593.187

Tabela 5.16: Resultados obtidos na otimização – peso da estrutura em lbs.

5.2.5 Resultados comparativos - Problema P3

A Tabela 5.17 mostra um estudo comparativo entre o melhor resultado para o Problema P2 e soluções encontradas na literatura cujos valores reais para o peso da treliça são mostrados na Tabela 5.18. A Figura 5.12 mostra o peso da estrutura durante o processo de evolução.

A_i	Ref.[115]	Tese	Ref.[113]	Ref.[117]
1	7.960	7.93788	7.9379	7.9392
2	0.100	0.10000	0.1000	0.1000
3	8.074	8.06214	8.0621	8.0616
4	3.939	3.93787	3.9379	3.9332
5	0.100	0.10000	0.1000	0.1013
6	0.100	0.10000	0.1000	0.1000
7	5.765	5.74475	5.7447	5.7433
8	5.562	5.56899	5.5690	5.5711
9	5.558	5.56899	5.5690	5.5699
10	0.100	0.10000	0.1000	0.1000
W (lb)	1594.50	1593.184	1593.18	1593.17

Tabela 5.17: Resultados comparativos entre a literatura e a melhor solução do Problema P3.

Referência	W apresentado	W calculado
Ref.[115]	1594.50	1594.563
Tese	1593.184	1593.184
Ref.[113]	1593.18	1593.182
Ref.[117]	1593.170	1593.170

Tabela 5.18: Resultados comparativos entre a literatura e a melhor solução do Problema P3.

Nota-se pelos resultados apresentados na Tabela 5.19 que algumas barras violam as restrições de tensões impostas na formulação do problema: barras de números 8 e 9 da Referência [115], barras de números 3 e 7 da Referência [113] e barras de números 3,4 e 7 da Referência [117]. As restrições impostas pelo problema foram rigorosamente satisfeitas pelos resultados encontrados neste trabalho.

Barra	tensões nas barras			
	Ref.[115]	Tese	Ref.[113]	Ref.[117]
1	24927.6	24999.9	24999.9	24996.3
2	15587.8	15533.0	15532.9	15530.7
3	-24966.1	-24999.9	-25000.1	-25001.2
4	-24991.4	-24999.9	-24999.8	-25029.7
5	-176.975	-0.17074	44189.2	32759.7
6	15587.8	15533.0	15532.9	15530.7
7	24917.8	24999.9	25000.1	25005.3
8	-25025.5	-24999.9	-25000.0	-24991.4
9	25048.0	24999.9	24999.9	24996.0
10	-22044.4	-21967.0	-21966.8	-21963.7

Tabela 5.19: Problema P3 - tensões nas barras.

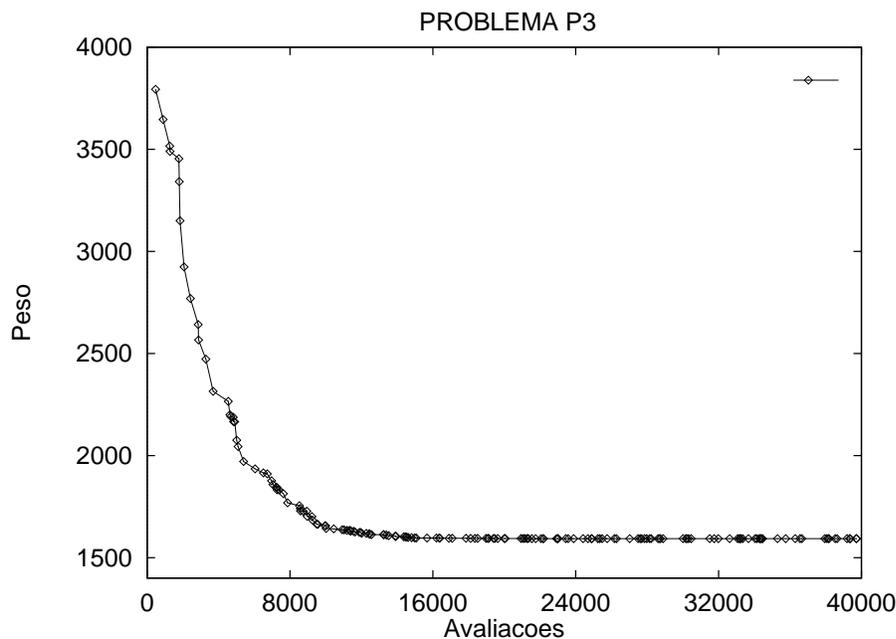


Figura 5.12: Problema P3 - somente restrições de tensões.

5.2.6 Problema P4

O Problema **P4** corresponde a uma análise feita considerando-se o carregamento definido com os valores de $P1 = 150$ Kips, $P2 = 50$ Kips e impondo-se restrições de tensões e de deslocamentos. Como nos problemas anteriores, o tipo de algoritmo genético utilizado é o “steady-state” e as probabilidades para os operadores utilizados são mostrados na Tabela 5.21.

Neste caso, manteve-se a aplicação de um lamarckianismo com probabilidade $prob.=0.1$. Diferentemente dos casos anteriores, as avaliações foram de 30000, 40000 e 50000 para as populações de tamanho 400 e 500 e de 30000, 42000 e 54000 para a população de 600.

Tabela 5.20: Resultados comparativos do Problema P3.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	7.93912	7.93791	7.93789	7.94336	7.93876	7.93788	7.95579	7.94085	7.93791
2	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
3	8.06435	8.06215	8.06214	8.07454	8.06237	8.06214	8.12168	8.06281	8.06217
4	3.93873	3.93787	3.93788	3.94751	3.93806	3.93787	3.98378	3.93839	3.93787
5	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
6	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
7	5.74688	5.74477	5.74474	5.74752	5.74486	5.74475	5.78323	5.74518	5.74476
8	5.56939	5.56901	5.56902	5.59035	5.56908	5.56899	5.60362	5.57056	5.56900
9	5.56994	5.56902	5.56902	5.60842	5.56933	5.56899	5.61126	5.56989	5.56900
10	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
W(lb)	1593.516	1593.189	1593.186	1597.410	1593.258	1593.184	1603.499	1593.481	1593.187
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19469	28810	30325	19677	29914	38968	20770	29568	41921

A Tabela 5.22 resume os resultados obtidos na otimização detalhados na Tabela 5.26, observando-se que a melhor solução (4676.931 lbs) é obtida com uma população de 400 elementos e 40000 avaliações. A Figura 5.13 mostra a variação do peso da estrutura ao longo da evolução.

<i>prob</i>	cr. 2 pontos	cr. discreto	cr. heurístico	mut. randômica
<i>prob_i</i>	0.3	0.2	0.3	0.2
<i>prob_f</i>	0.0	0.1	0.3	0.4

Tabela 5.21: Probabilidades iniciais e finais dos operadores.

avaliações	população		
	400	500	600
20000	4677.182	4677.114	-
21000	-	-	4677.301
30000	4677.979	4677.030	4676.980
40000	4676.931	4676.934	
42000	-	-	4676.953

Tabela 5.22: Resultados obtidos na otimização – peso da estrutura em lbs.

5.2.7 Resultados comparativos - Problema P4

Os resultados comparativos entre soluções encontradas na literatura e os obtidos nesta tese são mostrados nas Tabelas 5.23, 5.24 e 5.25. Nas Tabelas 5.23 e 5.24 são feitas comparações entre os pesos das treliças e as respectivas áreas das barras para as várias soluções. Na Tabela 5.25 são realizadas comparações entre o peso apresentado e o peso calculado bem como os deslocamentos nodais obtidos para as várias soluções.

Analisando-se os resultados apresentados na Tabela 5.25, nota-se que existem violações de restrições referentes aos deslocamentos máximos permitidos: deslocamento u_{y2} da referência [109], deslocamento u_{y2} da referência [120], deslocamento u_{y2} da referência [119] e deslocamento u_{y2} da referência [113], o que não acontece com os resultados desta tese.

A_i	Ref.[110]	Ref.[109]	Ref.[112]	Ref.[120]	Ref.[108]
1	25.813	25.190	24.716	24.28	24.289
2	0.100	0.363	0.100	0.10	0.100
3	27.233	25.419	26.541	24.28	23.346
4	16.653	14.327	13.219	15.98	13.654
5	0.100	0.417	0.108	0.10	0.100
6	2.024	3.144	4.835	3.84	1.969
7	12.776	12.083	12.664	12.13	12.670
8	14.218	14.612	13.775	13.06	12.544
9	22.137	20.261	18.438	18.67	21.971
10	0.100	0.513	0.100	0.10	0.100
W	5095.50	4895.60	4792.52	4707.00	4691.84

Tabela 5.23: Resultados comparativos entre a literatura e a melhor solução do Problema P4.

A_i	Ref.[107]	Ref.[119]	Tese	Ref.[111]	Ref.[113]
1	23.550	23.550	23.55490	23.5330	23.56
2	0.176	0.100	0.10000	0.1000	0.10
3	25.200	25.290	25.26433	25.2910	25.28
4	14.390	14.360	14.40471	14.3740	14.34
5	0.100	0.100	0.10000	0.1000	0.10
6	1.967	1.970	1.96980	1.9697	1.97
7	12.400	12.390	12.38889	12.3890	12.40
8	12.860	12.810	12.80681	12.8250	12.85
9	20.410	20.340	20.32755	20.3280	20.29
10	0.100	0.100	0.10000	0.1000	0.10
W	4684.11	4676.96	4676.931	4676.92	4676.21

Tabela 5.24: Resultados comparativos entre a literatura e a melhor solução do Problema P4.

Referência	W apresentado	W calculado	u_{y1}	u_{y2}	u_{y3}	u_{y4}
[110]	5059.50	5095.661	-0.94166	-1.81688	-0.595454	-1.49537
[109]	4895.60	4895.687	-1.45136	-2.00006	-0.661961	-1.50188
[112]	4792.52	4792.539	-1.63011	-1.99999	-0.614870	-1.51670
[120]	4707.00	4706.958	-1.54430	-2.00865	-0.641279	-1.59891
[108]	4691.84	4691.847	-1.09997	-1.99999	-0.659620	-1.55963
[107]	4684.11	4684.016	-1.09962	-1.99910	-0.656863	-1.55656
[119]	4676.96	4676.929	-1.10021	-2.00009	-0.657589	-1.55721
Tese	4676.931	4676.931	-1.10004	-2.00000	-0.657662	-1.55766
[111]	4676.92	4676.948	-1.09997	-1.99999	-0.657276	-1.55728
[113]	4676.21	4676.209	-1.10041	-2.00031	-0.656180	-1.55637

Tabela 5.25: Problema P4 - Pesos apresentados e calculados e deslocamentos correspondentes às soluções apresentadas.

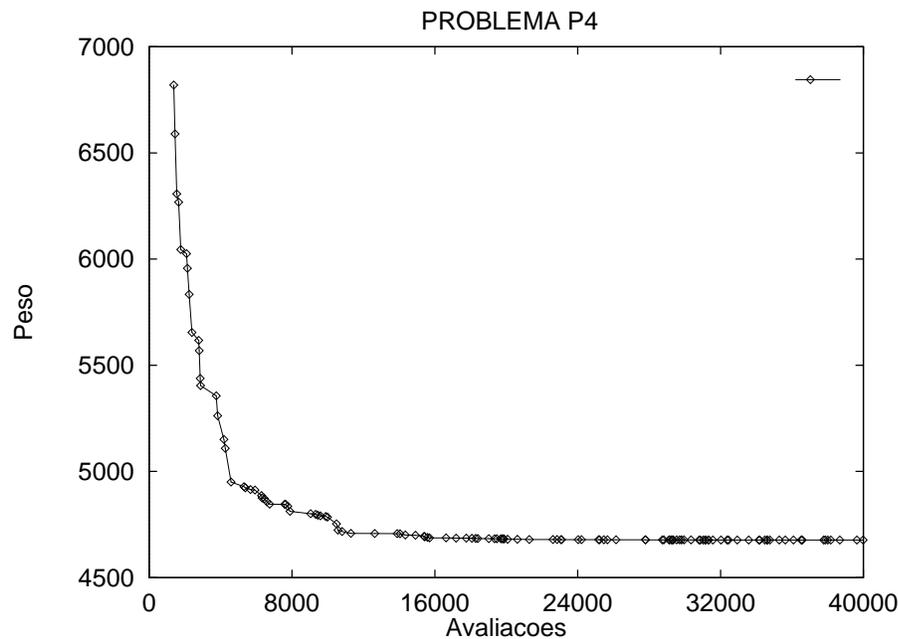


Figura 5.13: Problema P4.

5.2.8 Problema P5

O Problema **P5** corresponde a uma análise feita considerando-se o carregamento definido com os valores de $P1 = 100$ Kips, $P2 = 0$ e impondo-se as restrições de deslocamentos e de tensões. Utilizou-se primeiramente um algoritmo do tipo “steady-state”, com codificação real, onde o espaço de busca para as áreas A_i é discreto.

São 42 seções transversais disponíveis a serem utilizadas na otimização e cada barra tem uma opção de escolha em um dado intervalo como mostra a Tabela 5.27. Por exemplo, a área da barra 9 da treliça deve ser escolhida entre as áreas de número 27 e número 42, incluindo ambas. As seções disponíveis são fornecidas pelo *AISC* -

Tabela 5.26: Resultados comparativos do Problema P4.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	23.42485	23.58510	23.55490	23.33976	23.61526	23.53896	23.56819	23.59830	23.53211
2	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
3	25.25629	25.19425	25.26433	25.44484	25.18008	25.29264	25.50345	25.36195	25.33163
4	14.54284	14.32849	14.40471	14.45127	14.36428	14.36944	14.47142	14.35207	14.32459
5	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
6	1.96979	1.97003	1.96980	1.97048	1.97131	1.97007	1.97019	1.97030	1.96978
7	12.42888	12.42521	12.38889	12.37141	12.42362	12.38808	12.36707	12.37244	12.38126
8	12.96079	12.87356	12.80681	12.89577	12.85313	12.82082	12.83169	12.78547	12.81793
9	20.13849	20.30734	20.32755	20.25073	20.29284	20.33040	20.10582	20.30345	20.34968
10	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
W(lb)	4677.182	4676.979	4676.931	4677.114	4677.030	4676.934	4677.301	4676.980	4676.953
naval	30000	40000	50000	30000	40000	50000	30000	42000	54000
aval	29645	39480	49478	29706	39894	49637	29529	41414	53912

American Institute of Steel Construction e estão dispostas na Tabela 5.28.

Esta primeira análise foi denominada como Caso I. Para ela foram adotadas os mesmos operadores e probabilidades do Problema P1 e as seguintes variações no tamanho da população (100, 200, 300, 400, 500 e 600), e no número de avaliações (5000, 5100, 10000, 10200, 15000, 20000, 21000, 30000, 40000 e 42000). O resumo dos resultados obtidos na otimização encontra-se na Tabela 5.29 e estão detalhados nas Tabelas 5.30 e 5.31. Para este caso o melhor resultado (5490.738 lbs), foi observado em várias análises e, entre elas, a de menor custo computacional foi a definida com uma população de 300 elementos e 10200 avaliações. A Figura 5.14 mostra a variação do peso da estrutura ao longo da evolução.

variável	intervalo	variável	intervalo
1	[27, 42]	6	[1, 16]
2	[1, 16]	7	[20, 35]
3	[27, 42]	8	[27, 42]
4	[25, 40]	9	[27, 42]
5	[1, 16]	10	[1, 16]

Tabela 5.27: Intervalos que definem sub-conjuntos de \mathcal{S} como possibilidades de escolha para as variáveis de projeto do Caso I.

No.	in ²	No.	in ²	No.	in ²
1	1.62	15	3.63	29	11.5
2	1.80	16	3.84	30	13.5
3	1.99	17	3.87	31	13.9
4	2.13	18	3.88	32	14.2
5	2.38	19	4.18	33	15.5
6	2.62	20	4.22	34	16.0
7	2.63	21	4.49	35	16.9
8	2.88	22	4.59	36	18.8
9	2.93	23	4.80	37	19.9
10	3.09	24	4.97	38	22.0
11	3.13	25	5.12	39	22.9
12	3.38	26	5.74	40	26.5
13	3.47	27	7.22	41	30.0
14	3.55	28	7.97	42	33.5

Tabela 5.28: Conjunto \mathcal{S} de seções transversais disponíveis.

Em uma segunda análise (caso II), foi realizada uma otimização onde não houve limitações, por intervalos de escolha a partir do conjunto \mathcal{S} da Tabela 5.28 para as variáveis de projeto, e todas elas tiveram as 42 seções disponíveis para escolha. Baseado no melhor resultado da análise anterior (Caso I) adotou-se uma única população com 300 elementos e 10200 avaliações, mantendo-se o mesmo AG com a mesma parametrização.

avaliações	população					
	100	200	300	400	500	600
5000	5549.317	5513.423	5533.656	–	–	–
10200	5491.717	5491.717	5490.738	–	–	–
15000	5530.457	5525.534	5490.738	–	–	–
20000	–	–	–	5490.738	5497.217	–
21000	–	–	–	–	–	5491.717
30000	–	–	–	5490.738	5490.738	5507.758
40000	–	–	–	5491.717	5490.738	–
42000	–	–	–	–	–	5491.717

Tabela 5.29: Resultados obtidos na otimização - Caso I – peso da estrutura em lbs.

O resultado encontrado é idêntico ao obtido na primeira análise (5490.738 lbs), com a mesma distribuição de seções transversais para as barras. A variação do peso da treliça durante o processo de evolução é mostrado na Figura 5.15 observando-se que não haveria necessidade de 10200 avaliações visto que por volta de 6000 avaliações já foi encontrada a melhor solução e esta não foi superada até o fim da evolução.

Ainda, para este exemplo foi realizada uma terceira análise (Caso III), através de um algoritmo genético do tipo “geracional” com codificação binária, com restrições de escolhas para as barras da treliça como no Caso I (Tabela 5.27).

Foram definidos os seguintes dados: uma população de 40 elementos evoluída em 40 gerações e 4 bits para cada variável gerando um cromossomo com comprimento igual a 40. Os operadores genéticos de recombinação usados foram três: *crossover* de um ponto, dois pontos e uniforme, aplicados com as seguintes probabilidades relativas: $p_c^1 = 0.2$, $p_c^2 = 0.4$ e $p_c^u = 0.4$. A operação de recombinação foi definida com probabilidade $p_c = 0.8$ e a de mutação com taxa $p_m = 0.02$. Foram realizados 40 execuções para esta análise.

Os melhores resultados foram exatamente iguais aos obtidos nas análises anteriores, ou seja, mesmo peso para a estrutura com a mesma distribuição de seções transversais. A variação do peso ao longo da evolução é mostrada na Figura 5.16.

Tabela 5.30: Resultados comparativos do Problema P5 - Caso I.

A_i	população								
	100	100	100	200	200	200	300	300	300
1	33.50	33.50	30.00	33.50	33.50	33.50	30.00	33.50	33.50
2	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
3	26.50	22.90	22.90	26.50	22.90	26.50	22.90	22.90	22.90
4	13.50	15.50	13.50	14.20	15.50	13.90	13.90	14.20	14.20
5	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
6	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
7	7.97	7.97	7.22	7.97	7.97	7.22	11.50	7.97	7.97
8	22.00	22.00	26.50	22.90	22.00	22.00	22.00	22.90	22.90
9	22.00	22.00	22.90	19.90	22.00	22.00	22.90	22.00	22.00
10	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
W(lb)	5549.317	5491.717	5530.457	5513.423	5491.717	5525.534	5533.656	5490.738	5490.738
naval	5000	10000	15000	5000	10000	15000	5100	10200	15000
aval	2402	1606	1729	3640	4005	4392	5067	8620	9692

Tabela 5.31: Resultados comparativos do Problema P5 - Caso I.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	33.500	33.50	33.50	33.50	33.50	33.50	33.50	30.00	33.50
2	1.620	1.62	1.62	1.80	1.62	1.62	1.62	1.62	1.62
3	22.900	22.90	22.90	22.90	22.90	22.90	22.90	22.90	22.90
4	14.200	14.20	15.50	14.20	14.20	14.20	15.50	16.90	15.50
5	1.620	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
6	1.620	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
7	7.970	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.97
8	22.900	22.90	22.00	22.90	22.90	22.90	22.00	22.90	22.00
9	22.000	22.00	22.00	22.00	22.00	22.00	22.00	22.90	22.00
10	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
W(lb)	5490.738	5490.738	5491.717	5497.217	5490.738	5490.738	5491.717	5507.758	5491.717
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	8055	8055	8203	14413	10064	8721	11914	14373	17113

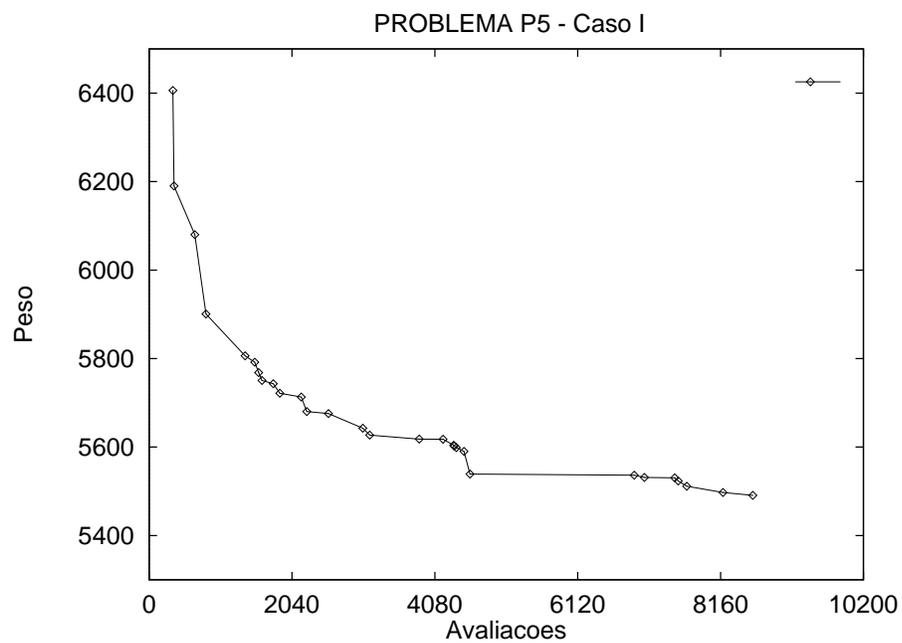


Figura 5.14: Caso I.

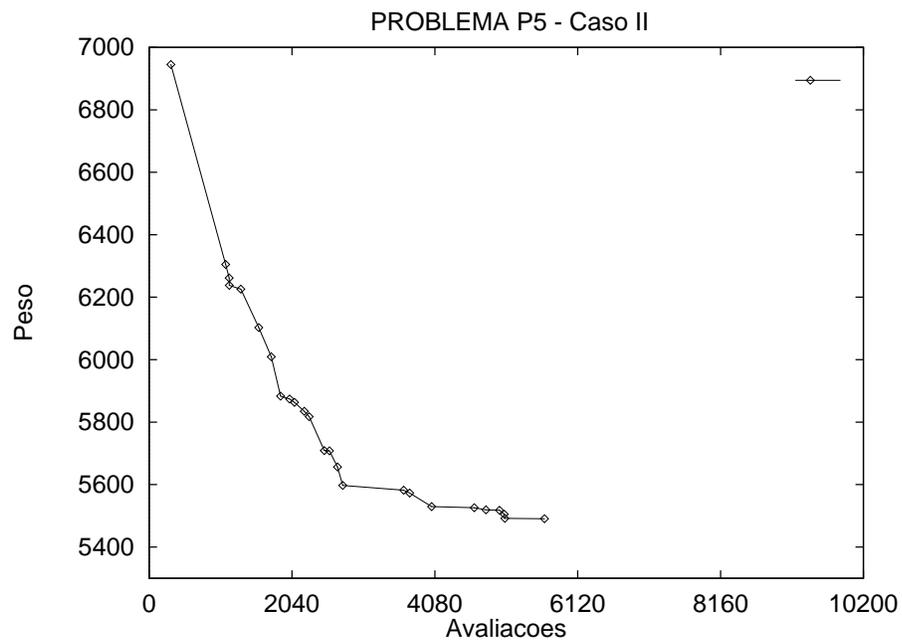


Figura 5.15: Caso II.

5.2.9 Resultados comparativos - Problemas P5

Os resultados comparativos entre soluções encontradas na literatura e os obtidos nesta tese são mostrados nas Tabelas 5.32 e 5.33. Na Tabela 5.32 são feitas comparações entre os pesos das treliças e as respectivas áreas das barras para as várias

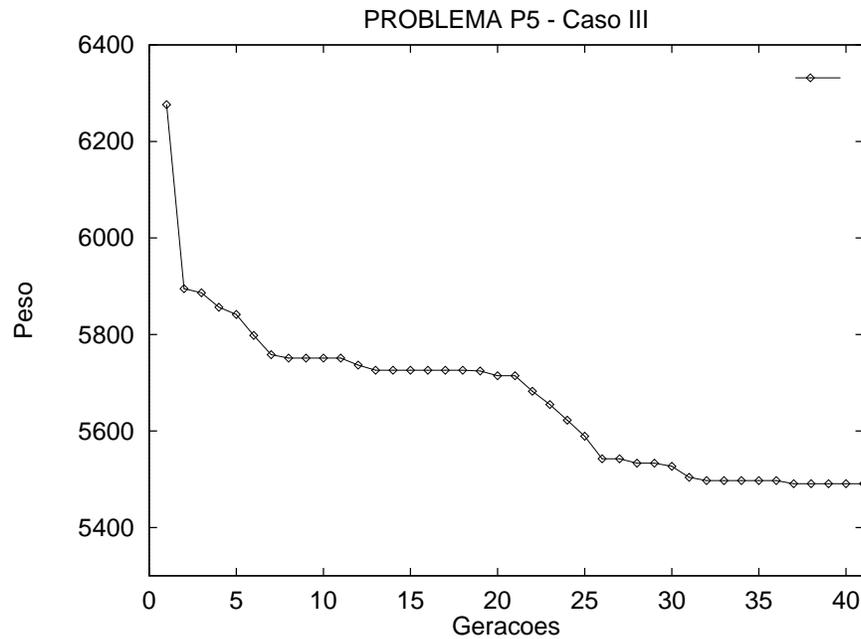


Figura 5.16: Caso III.

soluções. Na Tabela 5.33 são realizadas comparações entre o peso apresentado e o peso calculado bem como os deslocamentos nodais obtidos para as várias soluções.

A_i	Ref.[121](a)	Ref.[121](b)	Tese	Ref.[122]
1	33.50	33.50	33.50	33.50
2	1.62	1.62	1.62	1.62
3	22.00	22.00	22.90	22.00
4	15.50	15.50	14.20	14.20
5	1.62	1.62	1.62	1.62
6	1.80	1.62	1.62	1.62
7	14.20	14.20	7.97	7.97
8	19.90	19.90	22.90	22.90
9	19.90	19.90	22.00	22.00
10	2.62	2.62	1.62	1.62
W (lb)	5620.08	5613.84	5490.738	5458.53

Tabela 5.32: Resultados comparativos entre a literatura e a melhor solução do Problema P5.

Analisando-se os resultados comparativos apresentados na Tabela 5.33 percebe-se que existem pequenas diferenças entre os valores dos pesos apresentados e calculados e violações às restrições de deslocamentos nos resultados apresentados na referência [121](b) e também em [122]. Comparativamente com estas referências nota-se que os melhores resultados foram os obtidos neste trabalho, com coerência de resultados apresentados e sem violações de restrições.

Referência	W apresentado	W calculado	u_{y2}	u_{y4}
[121](a)	5620.08	5620.059	-1.88567	-1.99938
[121](b)	5613.84	5613.579	-1.87722	-2.00075
[122]	5458.3	5458.337	-1.97349	-2.01227
Tese	5490.738	5490.738	-1.95909	-1.98894

Tabela 5.33: Problema P5 - Pesos apresentados e calculados e deslocamentos correspondentes às soluções encontradas na literatura e obtidas nesta tese.

5.2.10 Problema P6

Em todos os problemas estudados nesta seção não haviam sido consideradas a inclusão do peso próprio das barras em suas configurações de carregamento.

O Problema **P6** corresponde a uma análise feita considerando-se o carregamento definido com os valores de $P1 = 100$ Kips, $P2 = 0$, incluindo-se o carregamento referente ao peso próprio das barras da estrutura, considerado como uma carga nodal equivalente em cada nó da treliça.

Percebe-se neste caso que cada estrutura avaliada tem uma configuração de carregamento diferente das demais. São impostas as restrições de deslocamentos e de tensões, estudando-se os casos contínuo e o discreto. Para o caso contínuo não foi considerado nenhum tipo de lamarkianismo nas análises. Todos os dados referentes a tamanho de populações, número de avaliações e operadores são os mesmos do Problema P1.

Para o caso discreto todas as análises efetuadas, em função das variações em relação ao tamanho da população e número de avaliações, levaram a um mesmo resultado (melhor entre todas as execuções), mostrado na Tabela 5.34. A Tabela 5.35 mostra os valores finais das cargas nos nós para o caso discreto representadas por F_y . A Figura 5.17 mostra a variação do peso da estrutura durante a evolução.

A Figura 5.18 mostra as variações das cargas nos nós 2 e 4.

barra	A_i
1	33.50
2	1.62
3	22.00
4	16.00
5	1.62
6	1.62
7	7.97
8	22.90
9	22.90
10	1.62
W(lb)	5568.958

Tabela 5.34: Caso discreto com peso próprio – peso da estrutura em lbs.

Nó	1	2	3	4
F_y	-99.56	-100900.10	-1827.20	-100957.29

Tabela 5.35: Cargas finais nos nós (lbs) – caso discreto.

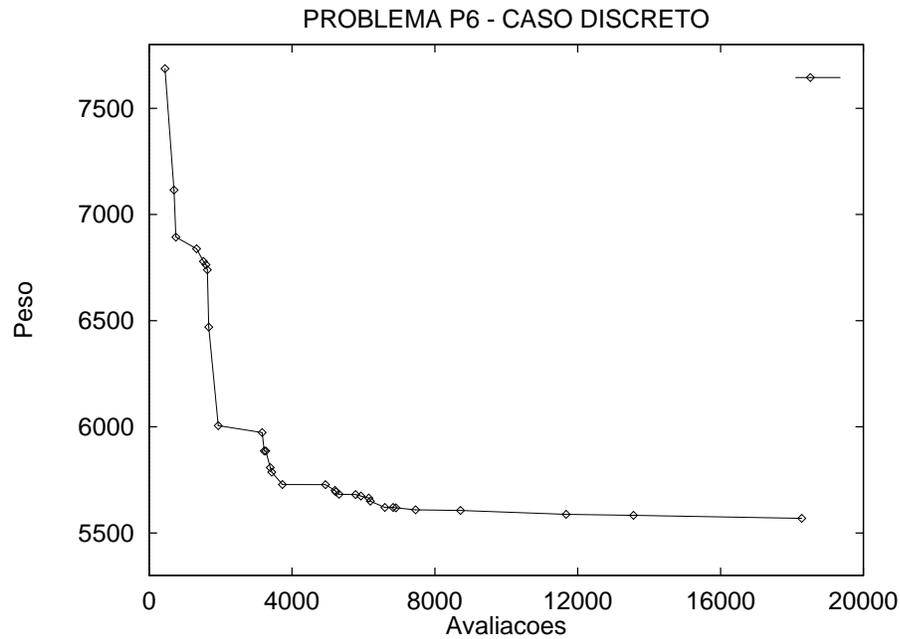


Figura 5.17: Caso discreto com consideração de peso próprio.

Para o caso contínuo a Tabela 5.36 resume os resultados obtidos na otimização para as variações de tamanho de população e número de avaliações detalhados na tabela 5.38. Observa-se que a melhor solução (5132.676 lbs), é encontrada com uma população de 400 elementos e 40000 avaliações. A Tabela 5.37 mostra os valores finais das cargas nos nós para o caso discreto A Figura 5.19 mostra a variação do peso da estrutura durante a evolução.

A Figura 5.20 mostra as variações das cargas nos nós 2 e 4.

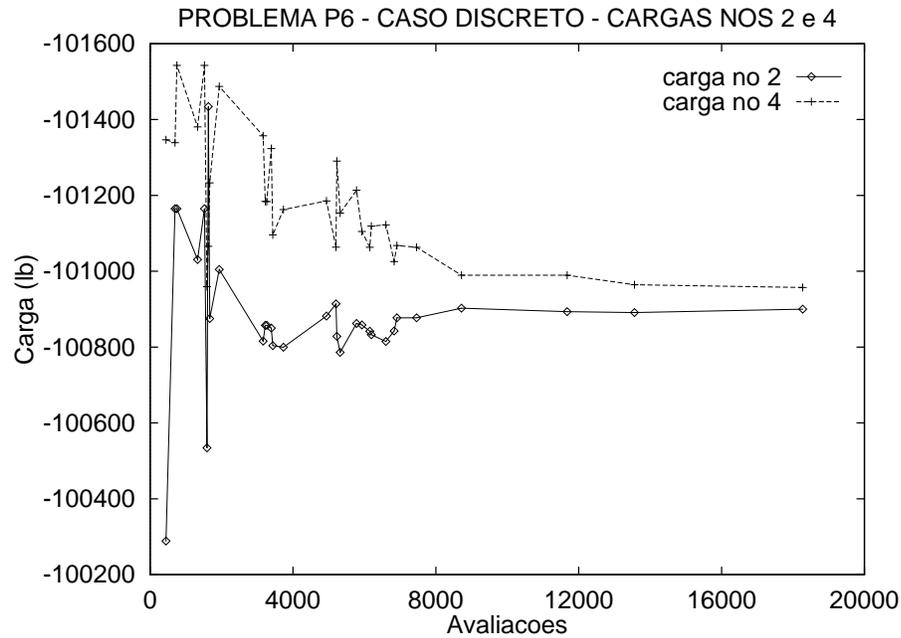


Figura 5.18: Caso discreto – cargas nos nós 2 e 4 – cargas em Kips $\times 10^3$.

avaliações	população		
	400	500	600
20000	5133.337	5139.116	-
21000	-	-	5138.591
30000	5133.243	5133.484	5136.333
40000	5132.676	5132.750	
42000	-	-	5134.127

Tabela 5.36: Resultados obtidos na otimização – peso em lbs.

Nó	1	2	3	4
F_y	-14.83	-100834.84	-1659.40	-100898.27

Tabela 5.37: Cargas finais nos nós (lbs) – caso contínuo.

Tabela 5.38: Caso contínuo com peso próprio.

A_i	população								
	400	400	400	500	500	500	600	600	600
1	31.04744	31.51655	30.95193	31.01476	31.42144	30.80030	31.64981	32.10498	30.89975
2	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000
3	23.98966	23.39820	23.62162	24.07783	23.21714	23.60061	23.44397	23.48412	24.15809
4	15.15301	15.59055	15.39973	15.42416	15.34352	15.18256	14.47654	15.08809	15.11685
5	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000
6	.50286	.59450	.58209	.47228	.55327	.55309	.56790	.58427	.54273
7	7.50501	7.53014	7.52443	7.59636	7.60272	7.52294	7.70947	7.62001	7.51757
8	21.63617	21.39952	21.66482	22.12259	21.77839	21.71663	22.16853	21.78041	21.82118
9	21.45820	21.38018	21.49444	20.78457	21.33258	21.74170	21.21681	20.85580	21.25883
10	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000	.10000
W(lb)	5133.337	5133.243	5132.676	5139.116	5133.484	5132.750	5138.591	5136.333	5134.127
naval	20000	30000	40000	20000	30000	40000	21000	30000	42000
aval	19566	29929	39273	19259	29711	38479	20753	29187	41595

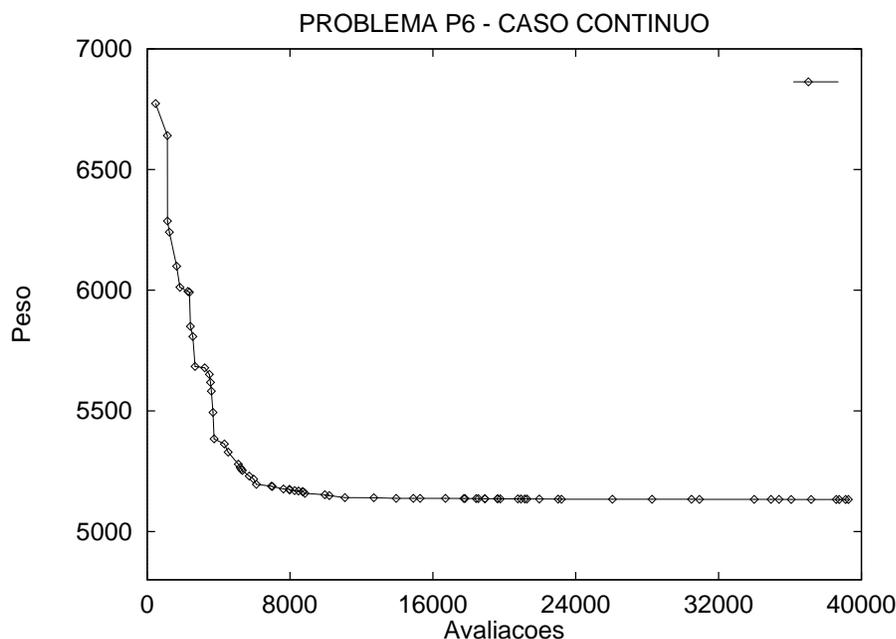


Figura 5.19: Caso contínuo com consideração de peso próprio.

5.2.11 Resumo dos resultados

A Tabela 5.39 resume os dados das análises referentes ao carregamento (P1 e P2), consideração ou não de restrições de deslocamentos (r. desl.) e restrições de tensões (r. tens.), variáveis discretas ou contínuas (var.) e finalmente, consideração ou não de peso próprio da estrutura (p. próprio).

A Tabela 5.40 resume os dados das análises referentes ao tipo de algoritmo genético utilizado (AG), tamanho das populações iniciais (popsiz), número máximo de avaliações (maxaval), número máximo de gerações (maxgen) e finalmente, número total de execuções no AG (exec.).

Prob.	P1	P2	r. desl.	r. tens.	var.	p. próprio
P1	100.0	0.0	sim	sim	cont.	não
P2	100.0	0.0	sim	sim	cont.	não
P3	100.0	0.0	não	sim	cont.	não
P4	150.0	50.0	sim	sim	cont.	não
P5	100.0	0.0	sim	sim	disc.	não
P6	100.0	0.0	sim	sim	cont.	sim
P6	100.0	0.0	sim	sim	disc.	sim

Tabela 5.39: Características dos Problemas.

Nas Tabelas 5.41 e 5.42 encontram-se os melhores resultados obtidos, respectivamente para os casos discreto e contínuo, em todos problemas analisados neste exemplo.

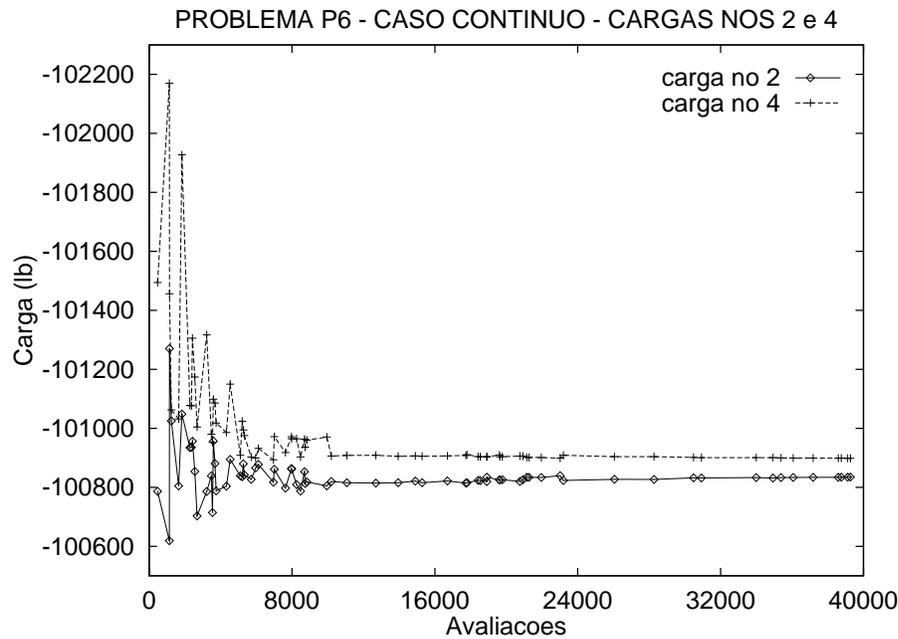


Figura 5.20: Caso contínuo – cargas nos nós 2 e 4 – cargas em Kips $\times 10^3$.

Prob.	AG	popsiz	maxaval	maxgen	exec.
P1	s. steady	600	42000	–	10
P2	s. steady	500	40000	–	10
P3	s. steady	500	50000	–	10
P4	s. steady	400	20000	–	10
P5	s. steady	300	10200	–	10
P5	geracional	40	–	40	40
P6	s. steady	400	40000	–	10
P6	s. steady	400	20000	–	10

Tabela 5.40: Características dos Problemas.

5.2.12 Comentários

Neste exemplo, subdivido em 6 problemas (**P1**, **P2**, ..., **P6**), foram analisadas várias situações. As variações ocorreram na consideração das restrições (tensões e deslocamentos), tipos de carregamentos, espaço de busca (variáveis contínuas ou discretas) e tipos de algoritmos genéticos utilizados (“steady-state” com codificação real e “geracional” com codificação binária).

Os algoritmos genéticos utilizados em todas as análises mostraram-se eficientes o que pode ser notado pelos aspectos das curvas apresentadas nos gráficos referentes às variações de peso da treliça durante os processos de evolução, notando-se que rapidamente encontra-se o caminho para uma região com algum tipo de solução ótima.

Foram analisadas várias situações para cada problema onde variou-se o tamanho da população inicial, número de avaliações, operadores genéticos e suas probabili-

A_i	problema				
	P1	P2	P3	P4	P6
1	30.45486	30.56091	7.93788	23.55490	30.95193
2	0.10000	0.10000	0.10000	0.10000	0.10000
3	23.08036	23.16999	8.06214	25.26433	23.39820
4	15.05050	15.11224	3.93787	14.40471	15.59055
5	0.10000	0.10000	0.10000	0.10000	0.10000
6	0.57944	0.54910	0.10000	1.96980	0.59450
7	7.49861	7.47048	5.74475	12.38889	7.53014
8	21.28839	21.09910	5.56899	12.80681	21.39952
9	21.47835	21.52714	5.56899	20.32755	21.38018
10	0.10000	0.10000	0.10000	0.10000	0.10000
W	5061.330	5060.920	1593.184	4676.931	5132.676
popsiz	600	500	500	400	400
naval	42000	40000	50000	20000	40000
aval	41441	39363	38968	49978	39273

Tabela 5.41: Melhores resultados para a treliça de 10 barras - caso contínuo.

A_i	problema	
	P5	P6
1	33.50	33.50
2	1.62	1.62
3	22.90	22.00
4	14.20	16.00
5	1.62	1.62
6	1.62	1.62
7	7.97	7.97
8	22.90	22.90
9	22.00	22.90
10	1.62	1.62
W	5490.738	5568.958
popsiz	300	400
naval	10200	20000
aval	8620	6237

Tabela 5.42: Melhores resultados para a treliça de 10 barras - caso discreto.

dades.

Mostrou-se a consideração de técnicas lamarckianas e efeito Baldwin cujos resultados satisfatórios dessas simulações foram aplicados em várias análises dos problemas subsequentes.

Procedeu-se a uma otimização estrutural com o objetivo de minimizar o peso total da estrutura onde foram introduzidas funções de penalização, constantes em todas as análises, transformando os problemas com restrições em problemas sem restrições.

Foram feitos estudos comparativos entre os resultados obtidos neste texto e alguns encontrados na literatura. Mostrou-se que os resultados obtidos neste texto respeitaram rigorosamente as restrições impostas aos problemas, ao contrário do que ocorre em alguns encontrados na literatura. Observou-se diferenças tanto em relação às violações quanto à apresentação dos próprios resultados propriamente ditos, principalmente em relação ao peso da estrutura.

Nenhum tipo de conhecimento do problema foi informado aos algoritmos, baseados somente na ordenação das soluções candidatas no *rank* de aptidão.

Foram testados vários critérios para se definir estratégias de penalização. Entretanto, para todos os problemas, adotou-se funções de penalização através de parâmetros constantes. Esta escolha foi devida ao fato de ser esta uma estratégia “relativamente” fácil de ser controlada permitindo o ajuste de um único parâmetro de maneira mais rápida e eficiente e, auxiliando, também, a definição de vários outros parâmetros como: tamanho da população, número de avaliações, número de gerações, etc. Outras estratégias apresentadas neste texto e que constam da literatura não foram exaustivamente testadas devido à complexidade das parametrizações envolvidas. Este procedimento será adotado nos experimentos seguintes deste texto.

5.3 Treliça de 38 barras

Este exemplo é de uma treliça de 38 barras descrita em [123] mostrada na Figura 5.21. As propriedades do material e restrições encontram-se na Tabela 5.43.

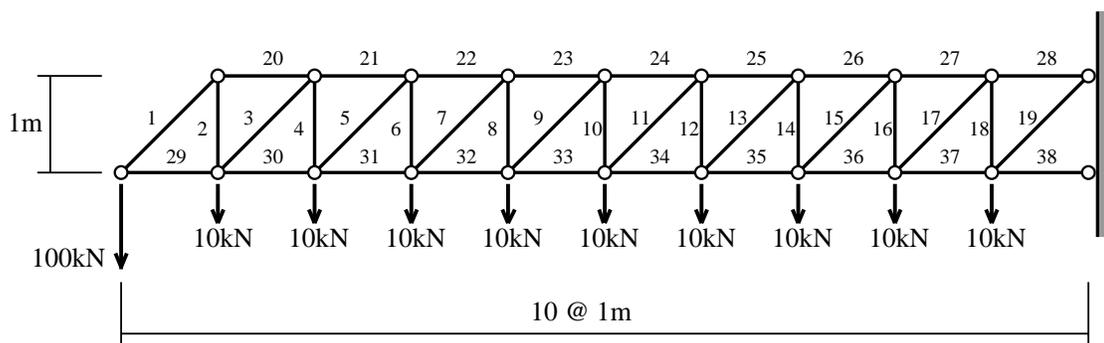


Figura 5.21: Treliça de 38 barras.

Pretende-se minimizar o peso total da estrutura impondo-se a restrição de que o deslocamento na direção vertical do nó da extremidade livre (com a carga de 100 kN), não seja superior a 10 mm.

propriedade	valor
Módulo de elasticidade	210.0 kN/mm ²
Massa específica	7.85 × 10 ⁻⁶ kg/mm ³
Deslocamento admissível	10 mm

Tabela 5.43: Propriedades do material e restrições.

As variáveis de projeto são as seções transversais das barras e estão agrupadas da seguinte forma: $A_2 = A_{20} = A_{29}$; $A_{21} = A_{30}$; $A_{22} = A_{31}$; $A_{23} = A_{32}$; $A_{24} = A_{33}$; $A_{25} = A_{34}$; $A_{26} = A_{35}$; $A_{27} = A_{36}$ e $A_{28} = A_{37}$, totalizando 28 variáveis de projeto.

Foram realizadas duas análises: a primeira onde o espaço de busca para as variáveis é discreto e formado pelo seguinte conjunto $S = \{5.0, 10.0, 20.0, 40.0, 75.0\}$ (mm² × 10³) (otimização discreta) e a segunda onde o espaço de busca para as variáveis é contínuo e limitado no intervalo [7.0, 90.0] (mm² × 10³).

Para o caso discreto, utilizou-se um algoritmo genético do tipo “geracional” com codificação binária onde foram analisadas 10 execuções. Adotou-se uma população com 600 elementos evoluída em 40 gerações.

O problema apresenta suas 28 variáveis de projeto codificadas com 3 bits cada perfazendo um cromossomo com comprimento igual a 84 bits.

O problema com restrições foi transformado em um problema sem restrições pela introdução de uma função de penalização. Neste caso, utilizou-se uma penalização variável como mostra o gráfico da Figura 5.22 onde a variável *penal* cresce de 0 a 30000 até a 20^a geração e partir desta permanece constante com este valor. A função a ser otimizada é:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|u_j|}{u_{max}} - 1 \right]^+ \right)^2$$

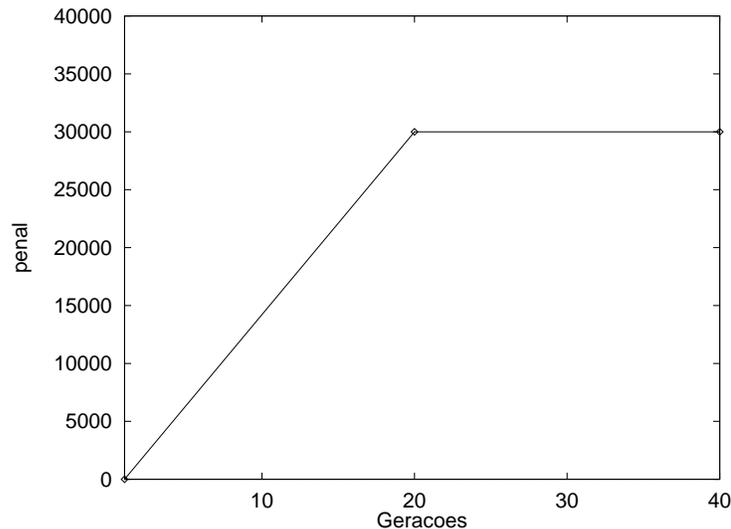


Figura 5.22: Função de penalização adotada no Problema da Treliça de 38 barras.

Os operadores genéticos de recombinação usados são três: *crossover* de um ponto, dois pontos e uniforme, aplicados com as seguintes probabilidades relativas: $p_c^1 = 0.2$, $p_c^2 = 0.4$ e $p_c^u = 0.4$. A operação de recombinação foi definida com probabilidade $p_c = 0.8$ e para a mutação adotou-se uma taxa de $p_m = 0.01$.

A Tabela 5.44 mostra o resultado obtido para as 10 diferentes populações iniciais adotadas para este caso.

A_i	1	2	3	4	5	6	7	8	9	10
1	10.	10.	10.	10.	5.	10.	10.	10.	10.	10.
2,20,29	10.	5.	10.	5.	10.	5.	5.	5.	5.	5.
3	10.	10.	20.	5.	10.	20.	10.	10.	10.	10.
4	10.	10.	10.	10.	20.	10.	10.	10.	10.	10.
5	10.	10.	20.	5.	10.	10.	10.	10.	10.	10.
6	10.	10.	5.	10.	10.	5.	10.	10.	10.	10.
7	20.	10.	10.	10.	10.	10.	10.	10.	10.	10.
8	5.	10.	10.	10.	20.	5.	20.	10.	5.	10.
9	10.	10.	10.	10.	10.	5.	20.	10.	10.	10.
10	10.	10.	10.	5.	5.	10.	10.	10.	5.	20.
11	10.	20.	10.	20.	10.	10.	20.	10.	20.	10.
12	5.	20.	20.	10.	10.	10.	10.	10.	10.	5.
13	20.	10.	10.	10.	10.	20.	10.	10.	10.	10.
14	10.	10.	10.	10.	10.	5.	5.	10.	5.	10.
15	10.	20.	10.	10.	10.	10.	20.	10.	20.	10.
16	10.	10.	10.	10.	10.	10.	10.	10.	20.	5.
17	20.	10.	10.	20.	20.	10.	10.	10.	10.	20.
18	10.	10.	10.	10.	5.	10.	10.	10.	5.	10.
19	20.	20.	20.	20.	20.	20.	20.	10.	10.	10.
21,30	20.	10.	20.	20.	10.	20.	20.	20.	20.	20.
22,31	20.	20.	20.	20.	20.	20.	20.	20.	20.	20.
23,32	20.	40.	20.	40.	40.	20.	20.	40.	40.	40.
24,33	40.	40.	40.	40.	40.	40.	40.	40.	40.	40.
25,34	40.	75.	40.	40.	40.	75.	40.	40.	40.	40.
26,35	75.	40.	75.	75.	75.	75.	75.	75.	75.	75.
27,36	75.	75.	75.	75.	75.	75.	75.	75.	75.	75.
28,37	75.	75.	75.	75.	75.	75.	75.	75.	75.	75.
38	75.	75.	75.	75.	75.	75.	75.	75.	75.	75.
W	8658.5	8743.7	8665.2	8671.9	8694.9	8884.4	8658.5	8489.2	8632.7	8600.17

Tabela 5.44: Solução para 10 populações iniciais diferentes - peso em kgs.

Entre as 10 populações iniciais testadas para este exemplo a de número 8 encontrou os melhores resultados e estes são comparáveis aos da Referência [123]. A evolução da otimização é mostrada na Figura 5.23.

Para o caso de otimização com variáveis contínuas adotou-se um algoritmo genético “steady-state” com codificação real.

Foram utilizados os seguintes operadores de recombinação: *crossover* de dois pontos, discreto, heurístico e *ls-x*. As probabilidades iniciais e finais de cada um deles se encontram na Tabela 5.45. Foi adotada uma população de 1000 elementos e executadas 300000 avaliações em 10 execuções e o melhor resultado é apresentado na Tabela 5.46 comparado com o encontrado na Referência [123]. Adotou-se um valor de *penal* = 100 em todas as execuções. A evolução da otimização é mostrada

na Figura 5.24.

<i>prob</i>	cr. 2 pontos	cr. discreto	cr. heurístico	cr. ls-x
<i>prob_i</i>	0.3	0.2	0.4	0.1
<i>prob_f</i>	0.2	0.2	0.4	0.2

Tabela 5.45: Probabilidades iniciais e finais dos operadores.

A_i	Ref.[123]	Tese
1	9.953	10.219081
2,20,29	7.038	7.0
3	10.439	10.529027
4	7.382	7.375649
5	10.903	10.703953
6	7.710	7.451641
7	11.348	11.332751
8	8.024	7.951734
9	11.777	11.815087
10	8.328	8.217712
11	12.190	12.170449
12	8.620	8.692951
13	12.590	12.597849
14	8.903	8.940743
15	12.978	2.950578
16	9.176	9.270661
17	13.354	13.446911
18	9.442	9.415315
19	13.720	13.774452
21,30	14.424	14.346538
22,31	22.145	22.320715
23,32	30.190	30.295357
24,33	38.549	38.782520
25,34	47.213	47.260029
26,35	56.172	56.291019
27,36	65.420	65.060045
28,37	74.948	74.717652
38	84.749	84.740997
W (kg)	8165.7	8166.044

Tabela 5.46: Resultados comparativos - variáveis contínuas - $A_i(\text{mm}^2 \times 10^3)$.

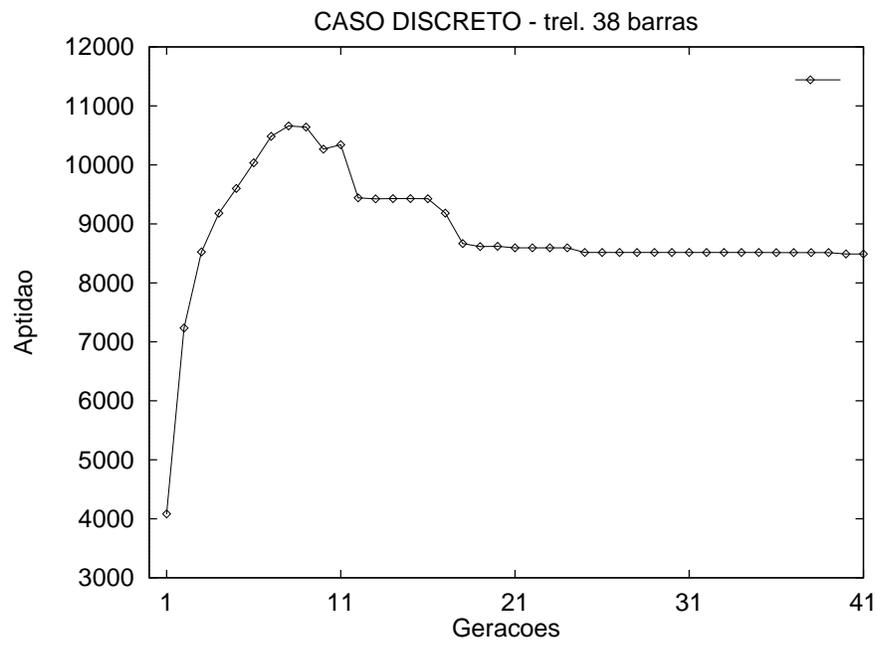


Figura 5.23: Caso discreto.

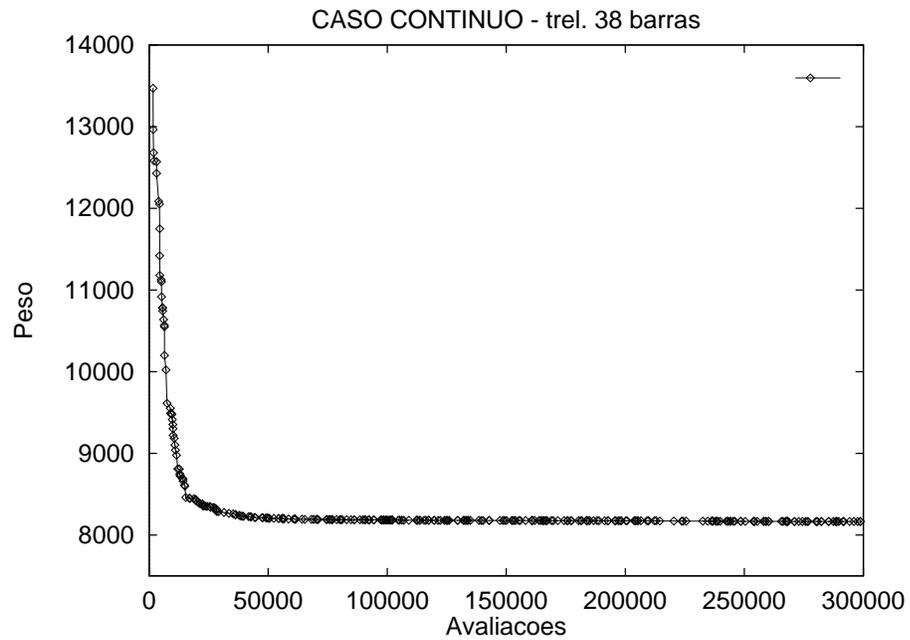


Figura 5.24: Caso contínuo.

Para o caso discreto os resultados encontrados foram exatamente iguais aos apresentados na referência [123]. Entretanto, o caso contínuo mostrou-se mais trabalhoso, acarretando um expressivo número de avaliações não alcançando-se o valor apresentado em [123]. Apesar disso, observa-se que a diferença é extremamente pequena entre os resultados comparados e é da ordem de 0.122 %. Pode-se dizer que o algoritmo genético obteve resultados aceitáveis. Também como no exemplo anterior a busca efetuou-se sem qualquer tipo de informação adicional sobre o problema e soluções boas já são obtidas por volta de 50000 avaliações.

Neste caso, os resultados encontrados nesta tese bem como os apresentados na literatura satisfazem plenamente às restrições impostas.

5.4 Treliça de 52 barras

A treliça de 52 barras mostrada na Figura 5.25 é submetida ao carregamento indicado na Tabela 5.47 e tem as propriedades do material mostradas na Tabela 5.48. As restrições impostas são as tensões máximas admissíveis.

As áreas das barras da estrutura compõem 12 grupos para garantir a simetria do projeto totalizando 12 variáveis de projeto: $A_{1-4} = x_1$, $A_{5-10} = x_2$, $A_{11-13} = x_3$, $A_{14-17} = x_4$, $A_{18-23} = x_5$, $A_{24-26} = x_6$, $A_{27-30} = x_7$, $A_{31-36} = x_8$, $A_{37-39} = x_9$, $A_{40-43} = x_{10}$, $A_{44-49} = x_{11}$ e $A_{50-52} = x_{12}$. Os valores discretos para as seções transversais das barras são escolhidas a partir da Tabela 5.49.

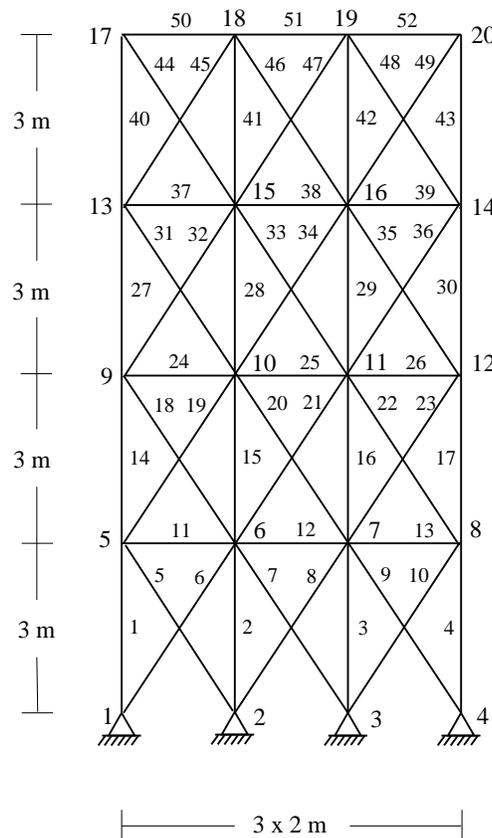


Figura 5.25: Treliça de 52 barras.

NÓ	F_x	F_y
17	100.0	200.0
18	100.0	200.0
19	100.0	200.0
20	100.0	200.0

Tabela 5.47: Dados de carregamento - cargas em kN.

propriedade	valor
Módulo de elasticidade	2.07×10^5 MPa
Tensões admissíveis	± 180.0 Mpa
Massa específica	7860.0 kg/m ³

Tabela 5.48: Propriedades do material.

Utilizou-se um algoritmo genético do tipo “geracional” com codificação binária e foram usados os seguintes operadores genéticos de recombinação: crossover de um ponto, dois pontos e uniforme, aplicados com as seguintes probabilidades relativas: $p_c^1 = 0.2$, $p_c^2 = 0.4$ e $p_c^u = 0.4$. A operação de recombinação foi definida com probabilidade $p_c = 0.8$ e para a mutação $p_m = 0.01$.

A função a ser otimizada é:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{max}} - 1 \right]^+ \right)^2$$

onde *penal* é o coeficiente de penalização definido com o valor de 200.

Cada variável foi codificada com 6 bits que corresponde aos 64 valores disponíveis na Tabela 5.49, perfazendo um cromossomo com 72 bits.

Foi adotada uma população de 100 elementos evoluídos em 200 gerações em 10 execuções totalizando 20000 avaliações cada. Os resultados são apresentados na Tabela 5.50, onde “maxaval” representa o total de avaliações. A Figura 5.26 que mostra a evolução da otimização.

Obteve-se neste exemplo um resultado melhor para a otimização do que o apresentado pela referência [124] com um número menor de avaliações. Enquanto neste texto utilizou-se um algoritmo genético “geracional”, na referência [124] utilizou-se um algoritmo “steady-state” binário com operadores de recombinação de 1, 2, 3 e 4 pontos. Tanto os resultados encontrados na literatura quanto os apresentados nesta tese respeitam rigorosamente as restrições impostas ao problema.

No.	in ²	mm ²	No.	in ²	mm ²
1	0.111	71.613	33	3.840	2477.414
2	0.141	90.968	34	3.870	2496.769
3	0.196	126.451	35	3.880	2503.221
4	0.250	161.290	36	4.180	2696.769
5	0.307	198.064	37	4.220	2722.575
6	0.391	252.258	38	4.490	2896.768
7	0.442	285.161	39	4.590	2961.284
8	0.563	363.225	40	4.800	3096.768
9	0.602	388.386	41	4.970	3206.445
10	0.766	494.193	42	5.120	3303.219
11	0.785	506.451	43	5.740	3703.218
12	0.994	641.289	44	7.220	4658.055
13	1.000	645.160	45	7.970	5141.925
14	1.228	792.256	46	8.530	5503.215
15	1.266	816.773	47	9.300	5999.988
16	1.457	940.000	48	10.850	6999.986
17	1.563	1008.385	49	11.500	7419.340
18	1.620	1045.159	50	13.500	8709.660
19	1.800	1161.288	51	13.900	8967.724
20	1.990	1283.868	52	14.200	9161.272
21	2.130	1374.191	53	15.500	9999.980
22	2.380	1535.481	54	16.000	10322.560
23	2.620	1690.319	55	16.900	10903.204
24	2.630	1696.771	56	18.800	12129.008
25	2.880	1858.061	57	19.900	12838.684
26	2.930	1890.319	58	22.000	14193.520
27	3.090	1993.544	59	22.900	14774.164
28	3.130	2019.351	60	24.500	15806.420
29	3.380	2180.641	61	26.500	17096.740
30	3.470	2238.705	62	28.000	18064.480
31	3.550	2290.318	63	30.000	19354.800
32	3.630	2341.191	64	33.500	21612.860

Tabela 5.49: Seções transversais disponíveis.

VAR. PROJETO	Ref. [124]	Tese
$x_1 = A_{1-4}$	4658.055	4658.055
$x_2 = A_{5-10}$	1161.288	1161.288
$x_3 = A_{11-13}$	645.160	363.225
$x_4 = A_{14-17}$	3303.219	3303.219
$x_5 = A_{18-23}$	1045.159	940.000
$x_6 = A_{24-26}$	494.193	641.289
$x_7 = A_{27-30}$	2477.414	2238.705
$x_8 = A_{31-36}$	1045.159	1008.385
$x_9 = A_{37-39}$	285.161	494.193
$x_{10} = A_{40-43}$	1696.771	1283.868
$x_{11} = A_{44-49}$	1045.159	1161.288
$x_{12} = A_{50-52}$	641.289	494.193
PESO (kg)	1970.142	1903.366
maxaval	60000	20000

Tabela 5.50: Variáveis de projeto da melhor solução em 10 execuções.

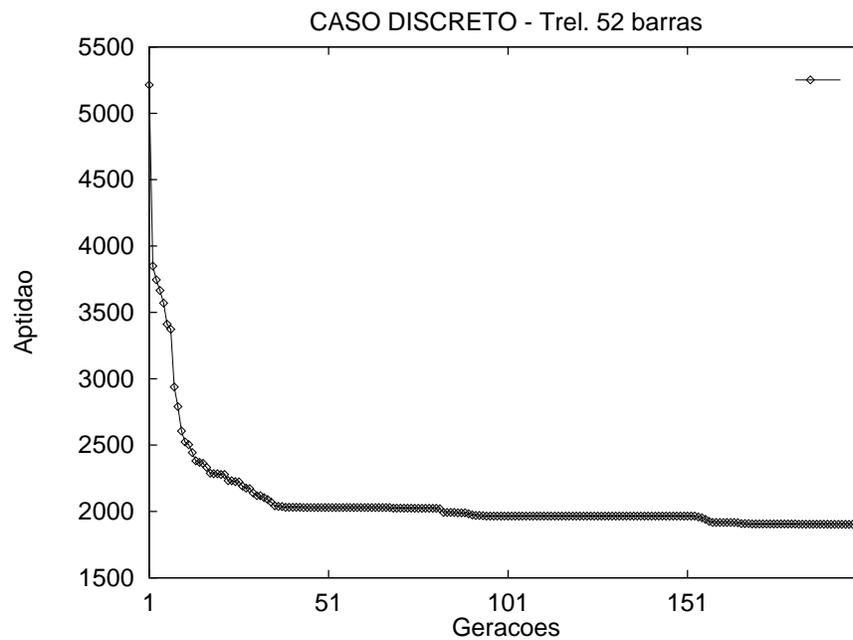


Figura 5.26: Função aptidão do melhor elemento da população em 10 execuções.

5.5 Treliça de 18 barras

Neste exemplo um algoritmo genético do tipo “steady-state” com codificação real é utilizado para resolver um problema de minimização de peso de uma treliça plana de 18 barras. Este problema é descrito nas Referências [125, 126, 104].

Foram consideradas variáveis contínuas tanto para as áreas das barras quanto para as coordenadas dos nós (caso contínuo) e variáveis discretas para as áreas das barras e contínuas para as coordenadas dos nós (caso misto).

A treliça analisada neste exemplo tem a possível configuração de referência mostrada na Figura 5.27, com carregamento, propriedades do material e restrições mostrados nas Tabelas 5.51 e 5.52, respectivamente.

As barras das treliças estão agrupadas da seguinte forma: $A_1 = A_4 = A_8 = A_{12} = A_{16} = x_1$; $A_2 = A_6 = A_{10} = A_{14} = A_{18} = x_2$; $A_3 = A_7 = A_{11} = A_{15} = x_3$; $A_5 = A_9 = A_{13} = A_{17} = x_4$, perfazendo 4 variáveis de projeto. Os nós 3, 5, 7 e 9 podem ter suas coordenadas alteradas tanto em relação ao eixo- x quanto eixo- y de acordo com os limites dados na Tabela 5.53, adicionando-se desta forma mais 8 variáveis de projeto: $X_3 = x_5, Y_3 = x_6, \dots, Y_9 = x_{12}$. Assim, o cromossomo tem um total de 12 variáveis de projeto.

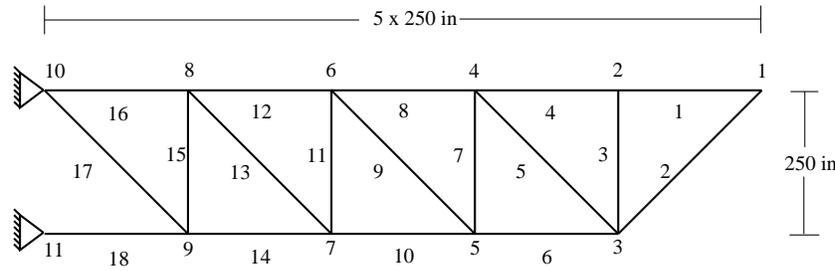


Figura 5.27: Treliça de 18 barras.

Duas situações serão analisadas neste exemplo: uma em que se consideram somente as tensões normais de tração e compressão como restrições e outra em que adicionam-se à essas restrições as tensões de flambagem de Euler, dadas pelas seguinte expressão:

$$\sigma_{Ei} = -\frac{K_i E_i A_i}{L_i^2}$$

onde K_i é o coeficiente de flambagem (igual a 4.0 para este caso), E_i o módulo de elasticidade do material, A_i e L_i a área e o comprimento, respectivamente, da i -ésima barra.

Para o caso discreto, as áreas das barras x_1 a x_4 são obtidas no conjunto $T = \{0.1, 0.2, 0.3, \dots, 30\}$ e para o caso contínuo obtidas no intervalo $[0.1, 30.0]$. Para o caso misto (áreas discretas e coordenadas nodais contínuas), as variáveis x_1 a x_4 variam no intervalo $[1, 31]$ e sua parte inteira é então usada como índice para a escolha no conjunto T de valores discretos para as seções transversais das barras.

A função aptidão a ser otimizada é definida como:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{max}} - 1 \right]^+ \right)^2 + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{Ei}} - 1 \right]^+ \right)^2$$

NÓ	F_x	F_y
1	0.0	-20000.0
2	0.0	-20000.0
4	0.0	-20000.0
6	0.0	-20000.0
8	0.0	-20000.0

Tabela 5.51: Dados de carregamento - cargas em lb.

propriedade	valor
Módulo de elasticidade	1.0×10^7 psi
Tensões admissíveis	± 20000.0 psi
Massa específica	0.10 lb/in ³
Coefficiente de flambagem K_i	4.0

Tabela 5.52: Propriedades do material e restrições.

$VAR.$	X_3	Y_3	X_5	Y_5	X_7	Y_7	X_9	Y_9
X^{INF}	876.0	0.0	555.0	0.0	376.0	0.0	126.0	0.0
X^{SUP}	1124.0	230.0	874.0	230.0	550.0	100.0	374.0	50.0

Tabela 5.53: Variáveis de forma.

onde $W = \sum \rho A_i L_i$ é o peso da estrutura e $penal$, definido com o valor de 600, é o coeficiente de penalização.

Utilizou-se uma população de 1800 indivíduos e 540000 avaliações em 10 execuções, exceto para a análise do caso contínuo com consideração de tensões normais e de flambagem de Euler, onde foram executadas 560000 avaliações com populações iniciais com 800 indivíduos. Foram adotados os seguintes operadores de recombinação: discreto, 2 pontos, BLX- α ($\alpha = 0.3$), e heurístico, e mutação de uma unidade. Os operadores e suas respectivas probabilidades são mostrados na Tabela 5.54.

caso	discreto	2 pontos	BLX- α	cr. heur.	mut. 1 unid
contínuo	0.2	0.2	0.2	0.4	-
contínuo (+ rest. Euler)	0.2	0.3	-	0.4	0.1
misto	0.2	0.2	0.2	0.4	-
misto (+ rest. Euler)	0.2	0.2	-	0.3	0.3

Tabela 5.54: Probabilidades dos operadores.

A Tabela 5.55 mostra os resultados correspondentes à melhor solução encontrada no caso onde as tensões de Euler não são consideradas, comparadas com a referência

[125] que apresenta resultados somente para o caso contínuo. É importante notar que a solução da Referência [125], (3906.8 lbs) não é rigorosamente factível, apresentando violações nas restrições.

VAR. PROJETO	Figura 5.27	contínuo	Ref.[125]	misto
A_1	10.00	11.02706	10.71	11.00
A_2	21.65	15.16515	15.19	16.00
A_3	12.50	1.38557	1.94	2.00
A_5	7.07	4.84354	5.19	5.00
X_3	1000.00	973.63529	881.4	917.76236
Y_3	0.00	222.499	178.8	202.52007
X_5	750.00	584.54103	628.9	593.38205
Y_5	0.00	131.86472	124.9	138.35909
X_7	500.00	397.31537	390.5	389.33700
Y_7	0.00	76.72122	66.8	73.39527
X_9	250.00	295.70768	313.2	283.01887
Y_9	0.00	43.99531	45.0	37.73585
PESO (lb)	6430.00	3871.489	3906.8	4011.342

Tabela 5.55: Variáveis de projeto da melhor solução para os casos contínuo e misto (sem restrições de Euler).

A Tabela 5.56 mostra as tensões normais σ_{Ni} referentes aos resultados apresentados na Tabela 5.55. Nota-se que existem barras com tensões violadas (3 e 17), nos resultados apresentados em [125], o que não acontece com os resultados obtidos nesta tese. Além disso o peso apresentado (3906.8 lbs), difere do compatível com os dados mostrados, ou seja, para esses, o valor correto do peso é 3905.08 lbs.

A Tabela 5.57 mostra os resultados correspondentes à melhor solução no caso em que são consideradas as tensões de Euler, comparadas também com os obtidos em [126]. Aqui, também, as soluções $W=4505.0$ lbs da referência [125] e $W=4506.09$ lbs da referência [126], não são rigorosamente factíveis como as obtidas pelo algoritmo genético dessas análises como mostram as Tabelas 5.58 e 5.59. Na Tabela 5.59 σ_{Ei} corresponde às tensões de flambagem de Euler para a barra i .

Nota-se na Tabela 5.58 que os resultados apresentados pela Ref. [125] tem a barra 17 com uma violação “excessiva” ($31868.6 \gg 20000.0$ psi) em relação ao limite para as tensões normais e as barras de número 2, 11, 15 e 18 violam as restrições referentes às tensões de Euler. Além disso o peso da estrutura (4505.0 lbs), não é compatível com os dados apresentados e para esses o valor correto é 4577.432 lbs.

Ainda na Tabela 5.58, verifica-se que os resultados da Ref.[126] apresentam violações referentes às tensões de Euler nas barras 2, 6 e 10 e neste caso o peso apresentado (4506.09 lbs), tem uma pequena diferença com o correto (4506.236 lbs de acordo com os dados apresentados).

Finalmente, a Figura 5.28 mostra o valor do peso da estrutura para o melhor elemento da população em 10 execuções independentes e a Figura 5.29 mostra as formas ótimas encontradas considerando-se os casos contínuo e misto bem como a inclusão ou não das tensões de Euler.

BARRA	contínuo	Ref.[125]	misto
	σ_{Ni}	σ_{Ni}	σ_{Ni}
1	18226.5	9667.54	12722.6
2	-13318.6	-6942.28	-8835.65
3	-19996.2	-20029.4	-19999.9
4	19965.3	12778.1	15871.7
5	-6594.52	3129.53	2363.35
6	-12759.7	-10173.8	-11847.6
7	-19998.9	-19899.7	-19999.9
8	19135.5	16618.6	19866.5
9	-450.451	3675.94	-1971.77
10	-14417.7	-12987.9	-13918.8
11	-15290.2	-19989.6	-7338.95
12	19999.8	19754.6	19999.9
13	1907.93	4059.23	1251.24
14	-15692.9	-15348.6	-14758.2
15	-19990.5	-19805.4	-12605.6
16	19998.6	19894.8	20000.0
17	19997.5	20017.7	19999.9
18	-19999.9	-19952.6	-18915.9

Tabela 5.56: Tensões normais nas barras (sem restrições de Euler).

VAR. PROJETO	caso contínuo			caso misto	
	Tese	Ref.[125]	Ref.[126]	Tese	Ref.[126]
A_1	12.19179	12.76	12.50	12.0	13.0
A_2	17.56473	17.77	17.72	18.0	18.0
A_3	5.48721	5.55	5.41	5.0	6.0
A_5	4.22398	3.26	3.71	5.0	4.0
X_3	909.77238	881.4	909.85	914.50884	909.85
Y_3	182.35215	178.8	184.33	188.79335	184.33
X_5	640.71597	628.9	641.3	649.87669	641.3
Y_5	143.41630	124.9	145.84	153.34165	145.84
X_7	411.21728	390.5	411.86	420.40269	411.86
Y_7	93.33045	66.8	97.03	98.46753	97.03
X_9	200.54943	313.2	201.09	205.87465	201.09
Y_9	24.66361	45.0	30.37	23.2811	30.37
PESO (lb)	4508.760	4505.0	4506.09	4568.557	4668.83

Tabela 5.57: Variáveis de projeto da melhor solução para os casos contínuo e misto (com restrições de Euler).

BARRA	Tese		Ref.[125]		Ref.[126]	
	σ_{Ni}	σ_{Ei}	σ_{Ni}	σ_{Ei}	σ_{Ni}	σ_{Ei}
1	8250.45		8114.37		8287.50	
2	-5838.79	-5838.80	-5934.34	-5043.45	-5954.10	-5905.95
3	-6076.04	-17259.1	-7001.25	-11601.5	-6278.68	-17396.1
4	10438.4		10725.2		10483.9	
5	9767.15		4982.29		11157.2	
6	-9506.33	-9506.34	-8696.65	-10662.8	-9653.94	-9630.36
7	-9418.84	-9418.86	-6955.94	-7323.05	-9545.72	-9547.76
8	16589.4		13948.7		16530.0	
9	4827.63		5852.19		5655.82	
10	-12733.0	-12733.1	-11102.2	-11805.3	-12895.9	-12881.4
11	-6768.47	-6768.54	-6987.34	-4873.50	-6922.72	-6942.91
12	19424.6		16580.9		19377.0	
13	714.455		6462.39		998.459	
14	-14310.4	-14310.5	-13120.2	-110192.	-14495.5	-14504.5
15	-4123.98	-4124.03	-6922.96	-4824.07	-4269.24	-4274.19
16	19999.9		16698.5		19994.0	
17	19999.9		31868.6		19987.4	
18	-17208.3	-17208.4	-17055.8	-7099.53	-17122.0	-17137.5

Tabela 5.58: Tensões normais e tensões de Euler - caso contínuo.

BARRA	Tese		Ref.[126]	
	σ_{Ni}	σ_{Ei}	σ_{Ni}	σ_{Ei}
1	9135.48		7968.75	
2	-6190.84	-6190.86	-5861.48	-5999.27
3	-6871.33	-18091.4	-5661.28	-19293.3
4	11463.4		10080.7	
5	9096.89		10348.3	
6	-10100.0	-10100.0	-9503.77	-9782.53
7	-10326.3	-10326.6	-8607.06	-10589.0
8	18111.4		15894.2	
9	2161.21		5245.77	
10	-12933.4	-12933.5	-12695.3	-13084.9
11	-5841.37	-6826.45	-6241.98	-7700.09
12	20000.0		18631.7	
13	-888.079	-3846.22	926.071	
14	-13933.2	-13933.2	-14270.1	-14733.7
15	-3473.84	-3748.93	-3849.43	-4740.32
16	20000.0		19225.0	
17	17850.4		18538.3	
18	-16772.9	-16772.9	-16855.7	-17408.3

Tabela 5.59: Tensões normais e de Euler - caso misto.

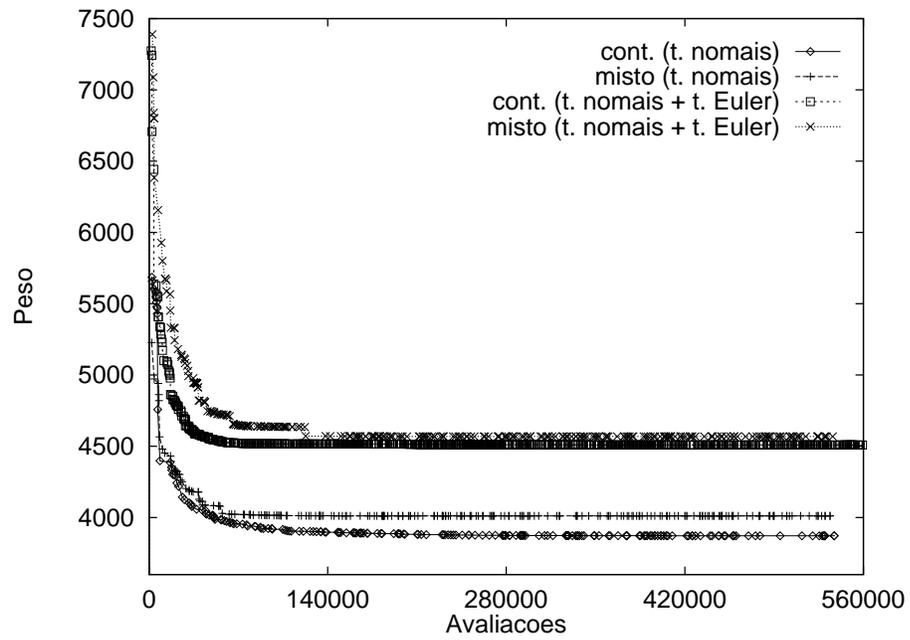


Figura 5.28: Função aptidão do melhor elemento da população em 10 execuções.

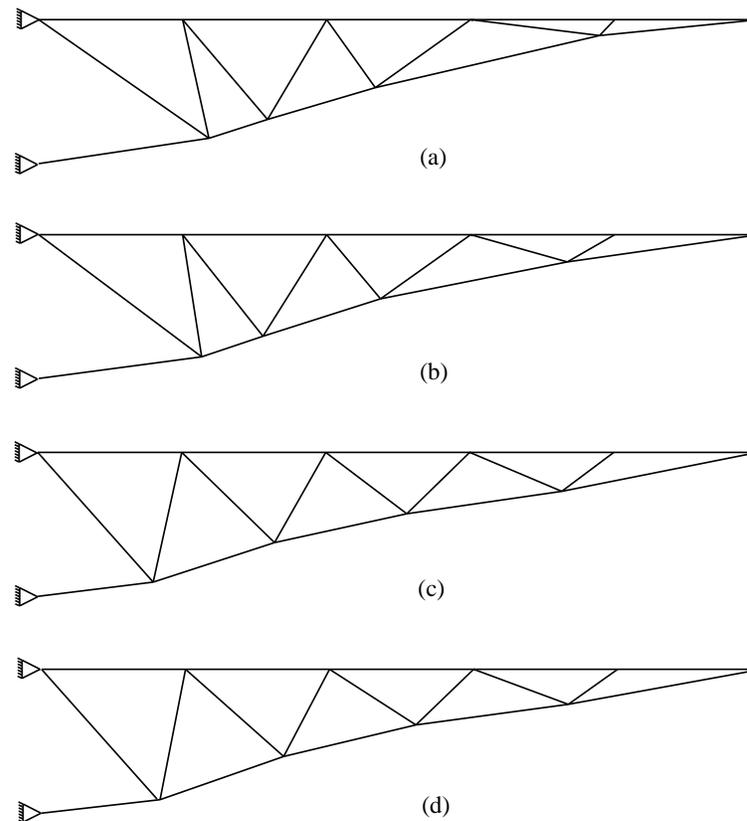


Figura 5.29: Forma ótima - (a) contínuo (tensões normais); (b) misto (tensões normais); (c) contínuo (tensões normais e de Euler); (d) misto (tensões normais e de Euler).

5.5.1 Relaxando a função de penalização

Um dado interessante, principalmente em problemas de engenharia, é a possibilidade de flexibilização dos limites impostos para as restrições. É comum na prática de projetos em engenharia a permissão de ligeiras violações de restrições. Muitas vezes a experiência do projetista permite a definição de valores aceitáveis que não comprometerão a segurança e eficiência do projeto já que existem normas, em geral conservadoras, que sugerem coeficientes de segurança que asseguram possíveis variações nas restrições impostas.

Observando-se a função objetivo a ser otimizada, que incorpora a função de penalização, nota-se que é possível um controle da porcentagem máxima de violação permitida. A função objetivo é dada por:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{max}} - 1 \right]^+ \right)^2 + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{Ei}} - 1 \right]^+ \right)^2$$

Para que a solução candidata seja factível é necessário que a parcela $|\sigma_i|/\sigma_{max}$ seja menor que 1, como mostra a expressão anterior. Dessa forma, adotando-se um valor maior do que 1 no termo anterior, significa uma relaxação da restrição. Vale lembrar, que o problema torna-se diferente do originalmente proposto.

Ilustrativamente, para o problema anterior, foram realizadas duas análises considerando-se flexibilizações nas restrições com valores de 2% e 5%, ou seja, adotando-se 1.02 e 1.05, respectivamente, no lugar do valor 1. A Tabela 5.60 mostra os resultados obtidos para o peso da estrutura considerando-se o caso misto sem restrições de Euler.

VAR. PROJETO	2 %	5 %
A_1	10.00	10.00
A_2	15.00	15.00
A_3	2.00	2.00
A_5	6.00	6.00
X_3	894.90691	886.74556
Y_3	190.89523	188.66817
X_5	585.66424	589.90811
Y_5	125.45134	127.45221
X_7	408.91321	402.97319
Y_7	67.22021	70.17939
X_9	287.02256	263.17953
Y_9	23.04510	18.17429
PESO (lb)	3871.926	3856.516

Tabela 5.60: Variáveis de projeto da melhor solução para o caso misto (sem restrições de Euler) – Flexibilização das restrições em 2% e 5%.

Para esta análise adotou-se a mesma parametrização do caso misto estudado anteriormente exceto o número de avaliações que foi de 90000. Percebe-se pela Tabela

5.60 que os valores encontrados para os pesos são menores que àquele mostrado na Tabela 5.55 com o valor de 4011.342 lbs, 3.6% e 4.01% a menos, respectivamente. Além disso, o número de avaliações caiu de 540000 para 90000. As tensões normais para os dois casos são mostradas na Tabela 5.61.

BARRA	2 %	5 %
1	12015.7	11845.5
2	-8120.68	-8008.79
3	-20399.9	-20999.7
4	15571.8	15538.7
5	2047.15	2219.31
6	-11386.1	11404.1
7	-20395.8	20934.2
8	19960.1	20078.3
9	-949.807	255.789
10	-13783.6	13936.3
11	-8549.84	10659.8
12	20399.9	20999.8
13	1018.93	312.080
14	-14749.9	15023.6
15	-12469.5	10730.4
16	20399.5	20999.4
17	20398.5	19990.8
18	-20064.3	20047.6

Tabela 5.61: Tensões normais nas barras – caso misto (sem restrições de Euler).

A Figura 5.30 mostra a evolução da função aptidão.

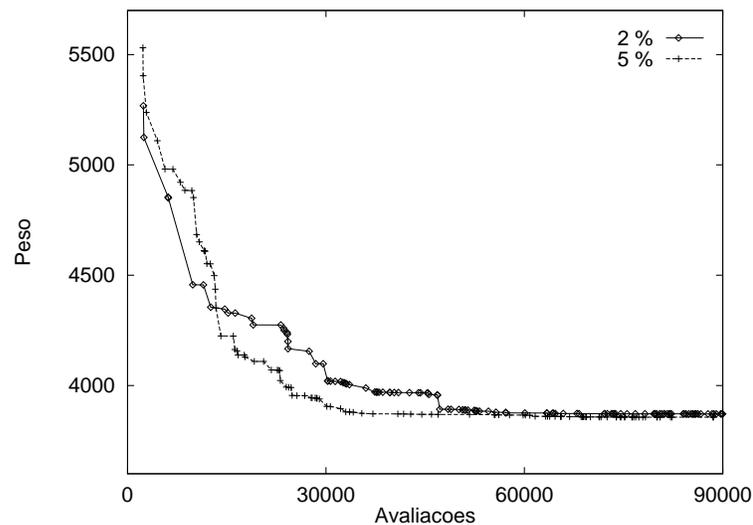


Figura 5.30: Peso do melhor elemento da população em 10 execuções.

5.6 Treliça de 47 barras

A treliça analisada neste exemplo descrita em [125, 126, 105], tem a configuração de referência mostrada na Figura 5.31, com carregamento, propriedades do material e restrições mostrados nas Tabelas 5.62 e 5.63, respectivamente.

A treliça apresenta simetria em relação ao seu eixo y e suas barras estão agrupadas da seguinte forma: $A_1 = A_3; A_2 = A_4; A_5 = A_6; A_8 = A_9; A_{11} = A_{12}; A_{13} = A_{14}; A_{15} = A_{16}; A_{17} = A_{18}; A_{19} = A_{20}; A_{21} = A_{22}; A_{23} = A_{24}; A_{25} = A_{26}; A_{29} = A_{30}; A_{31} = A_{32}; A_{34} = A_{35}; A_{36} = A_{37}; A_{39} = A_{40}; A_{41} = A_{42}; A_{44} = A_{45}; A_{46} = A_{47}$, gerando 27 variáveis de projeto, x_1 à x_{27} .

Os nós 15, 16, 17 e 22 têm suas coordenadas fixas, os nós 1 e 2 poderão ter suas coordenadas alteradas em relação ao eixo- x e o restante poderá ter suas coordenadas alteradas em relação aos eixos x e y , gerando mais 17 variáveis de projeto, x_{28} à x_{42} totalizando 44 variáveis para o problema.

Os objetivos a serem atingidos no processo de otimização são idênticos aos da treliça de 18 barras do exemplo anterior.

Serão considerados os casos contínuo e misto com inclusão ou não das tensões de flambagem de Euler. Para o caso contínuo as áreas das barras x_1 a x_{27} são obtidas no intervalo $[0.1, 50.0]$ e para o caso misto obtidas no conjunto $T = \{0.1, 0.2, 0.3, \dots, 50.0\}$ e as coordenadas dos nós para ambos os casos variam segundo os limites dados na Tabela 5.64.

Carregamento	NÓ	F_x	F_y
1	17	6000.0	-14000.0
	22	6000.0	-14000.0
2	17	6000.0	-14000.0
	22	0.0	0.0
3	17	0.0	0.0
	22	6000.0	-14000.0

Tabela 5.62: Dados de carregamento - cargas em lb.

propriedade	valor
Módulo de elasticidade	3.0×10^7 psi
Tensões admissíveis de tração	20000.0 psi
Tensões admissíveis de compressão	-15000.0 psi
Massa específica	0.30 lb/in ³
Coefficiente de flambagem K_i	3.96

Tabela 5.63: Propriedades do material e restrições.

A função aptidão a ser otimizada é definida como:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{max}} - 1 \right]^+ \right)^2 + penal \sum_i \left(\left[\frac{|\sigma_i|}{\sigma_{Ei}} - 1 \right]^+ \right)^2$$

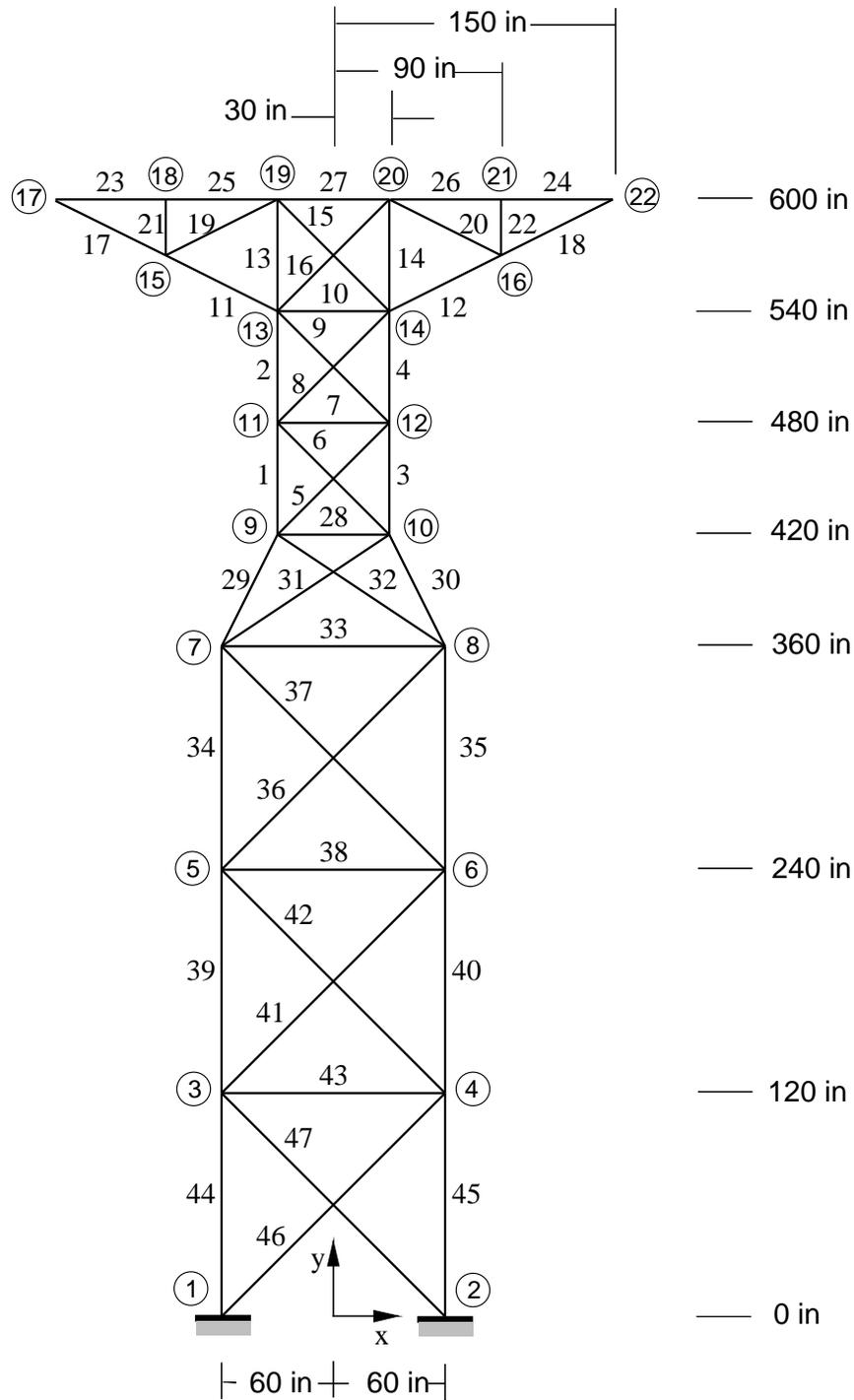


Figura 5.31: Treliza de 47 barras.

<i>VAR.</i>	X^{INF}	X^{SUP}	<i>VAR.</i>	X^{INF}	X^{SUP}
X_2	60.0	120.0	X_{12}	30.0	60.0
X_4	60.0	120.0	Y_{12}	460.0	490.0
Y_4	100.0	140.0	X_{14}	30.0	60.0
X_6	60.0	120.0	Y_{14}	500.0	540.0
Y_6	230.0	270.0	X_{20}	10.0	40.0
X_8	60.0	120.0	Y_{20}	570.0	630.0
Y_8	330.0	370.0	X_{21}	70.0	110.0
X_{10}	30.0	60.0	Y_{21}	600.0	650.0
Y_{10}	380.0	440.0			

Tabela 5.64: Variáveis de forma.

onde $W = \sum \rho A_i L_i$ é o peso da estrutura e $penal$, definido com o valor de 600, é o coeficiente de penalização.

Utilizou-se um algoritmo genético do tipo “steady-state” com uma população de 1200 indivíduos e 480000 avaliações em 10 execuções. Os operadores genéticos utilizados e suas respectivas probabilidades são encontrados na Tabela 5.65 e estes são válidos para todos os casos.

operadores	discreto	2 pontos	cr. heurístico	mut. randômica
probabilidades	0.2	0.3	0.2	0.3

Tabela 5.65: Probabilidades dos operadores.

A Tabela 5.66 mostra as variáveis de projeto da melhor solução em 10 execuções para os casos contínuo e misto sem restrições das tensões de Euler. As Tabelas 5.67 e 5.68 mostram as tensões normais σ_{Ni}^j nas barras $i = 1, 47$ para os três tipos carregamentos diferentes $j = 1, 3$ para os casos contínuo e misto, respectivamente. Nota-se que as restrições são rigorosamente satisfeitas nos resultados obtidos das análises desta tese.

A Tabela 5.69 mostra as variáveis de projeto da melhor solução em 10 execuções para os casos contínuo e misto com restrições das tensões de Euler comparadas com resultados encontrados em [125, 126].

As Tabelas 5.70, 5.71 e 5.72 mostram as tensões normais σ_{Ni}^j e tensões de Euler σ_{Ei}^j nas barras $i = 1, 47$ (caso contínuo), para os três tipos carregamentos diferentes $j = 1, 3$. A Tabela 5.73 refere-se ao caso misto que não foi abordado nas referências [125, 126].

Finalmente, a Figura 5.33 mostra o valor do peso da estrutura, para o melhor elemento da população, em 10 execuções independentes e a Figura 5.32 mostra as formas ótimas encontradas considerando-se os casos contínuo e misto bem como as inclusões ou não das tensões de Euler.

VAR. PROJETO	Figura 5.31	contínuo	misto
A_3	3.8	2.54286	2.4
A_4	3.4	2.50948	2.3
A_5	0.8	.33897	.4
A_7	0.9	.1	.1
A_8	0.9	.51708	.6
A_{10}	1.8	1.22544	.9
A_{12}	2.1	1.73460	1.6
A_{14}	1.2	.56487	.5
A_{15}	1.6	.81887	.7
A_{18}	2.1	1.40415	1.3
A_{20}	0.7	.21166	.3
A_{22}	0.9	.89333	.9
A_{24}	1.7	1.05333	1.0
A_{26}	1.7	1.09603	.9
A_{27}	1.4	.79425	.6
A_{28}	0.9	.1	.1
A_{30}	3.7	2.45778	2.4
A_{31}	1.5	.33112	.3
A_{33}	0.7	.1	.1
A_{35}	2.9	2.64507	2.7
A_{36}	0.7	.31165	.3
A_{38}	1.6	.1	.1
A_{40}	3.7	2.84948	3.0
A_{41}	1.6	.21849	.3
A_{43}	0.7	.1	.1
A_{45}	4.5	2.99613	3.3
A_{46}	1.6	.18441	.2
X_2	60.0	112.81926	97.76432
X_4	60.0	96.10662	85.10024
Y_4	120.0	131.40587	121.54254
X_6	60.0	76.95502	71.52607
Y_6	240.0	252.43600	261.49615
X_8	60.0	59.54597	65.94714
Y_8	360.0	363.28276	334.01452
X_{10}	30.0	51.09242	57.47306
Y_{10}	420.0	439.98245	414.88301
X_{12}	30.0	44.71505	50.13195
Y_{12}	480.0	484.57353	466.55141
X_{14}	30.0	42.47738	48.77601
Y_{14}	540.0	515.46528	508.67982
X_{20}	30.0	10.58937	16.88877
Y_{20}	600.0	594.85717	606.73294
X_{21}	90.0	94.34576	87.71766
Y_{21}	600.0	619.85756	628.73893
PESO (lb)	2446.8	1415.583	1432.993

Tabela 5.66: Variáveis de projeto da melhor solução para o casos contínuo e misto (sem restrições de Euler).

BARRA	σ_{Ni}^1	σ_{Ni}^2	σ_{Ni}^3
1	1128.65	-8180.60	9309.24
2	211.689	-9525.71	9737.39
3	-12489.3	2502.95	-14992.3
4	-10747.1	4252.83	-14999.9
5	13636.5	17040.6	-3404.06
6	-9458.11	-14998.9	5540.79
7	5693.82	2746.39	2947.44
8	5550.31	6394.50	-844.199
9	-14994.0	-11041.2	-3952.76
10	-14999.6	-7314.03	-7685.61
11	-14532.9	-14532.9	0
12	-14999.8	0	-14999.8
13	15454.2	-4027.69	19481.9
14	-6161.09	8837.34	-14998.4
15	-4546.73	10453.2	-14999.9
16	12992.7	-6804.24	19796.9
17	-14999.8	-14999.8	0
18	-11020.9	0	-11020.9
19	-14999.3	-14999.3	0
20	10341.1	0	10341.1
21	-9705.71	-9705.71	0
22	-14999.7	0	-14999.7
23	12941.0	12941.0	0
24	19999.8	0	19999.8
25	12941.1	12941.1	0
26	19999.9	0	19999.9
27	19999.0	6240.90	13758.1
28	1934.83	1009.28	925.547
29	3085.04	-6273.29	9358.32
30	-14370.5	629.145	-14999.7
31	12274.2	13878.8	-1604.63
32	-14551.6	-14999.6	447.956
33	195.192	83.1213	112.071
34	4498.85	-3735.29	8234.15
35	-14999.9	-1514.78	-13485.1
36	6836.29	13567.6	-6731.36
37	-9688.82	-14999.8	5311.01
38	2205.85	1107.11	1098.73
39	5022.79	-1853.55	6876.34
40	-15000.0	-3135.14	-11864.9
41	8929.96	15323.0	-6393.05
42	-8287.33	-14999.6	6712.30
43	3826.85	1912.40	1914.46
44	5578.89	-614.145	6193.03
45	-14999.9	-4096.35	-10903.6
46	13525.4	15004.5	-1479.08
47	-13514.5	-14999.7	1485.26

Tabela 5.67: Tensões normais nas barras - caso contínuo (sem restrições de Euler).

BARRA	σ_{Ni}^1	σ_{Ni}^2	σ_{Ni}^3
1	1433.71	-7536.40	8970.11
2	316.889	-9075.85	9392.74
3	-13421.2	1539.47	-14960.7
4	-11517.5	3482.46	-15000.0
5	11083.0	13289.1	-2206.15
6	-8288.08	-11847.9	3559.85
7	5303.60	2752.09	2551.51
8	5287.53	3848.06	1439.47
9	-14864.9	-8704.51	-6160.36
10	-14586.1	-7474.68	-7111.41
11	-14660.5	-14660.5	0
12	-14038.4	0	-14038.4
13	17791.8	-487.631	18279.4
14	-7570.56	6754.59	-14325.1
15	-5146.11	9797.65	-14943.8
16	14167.8	-5522.01	19689.8
17	-15000.0	-15000.0	0
18	-10046.8	0	-10046.8
19	-14978.3	-14978.3	0
20	5124.66	0	5124.66
21	-9705.88	-9705.88	0
22	-15000.0	0	-15000.0
23	12600.6	12600.6	0
24	19473.6	0	19473.6
25	12917.4	12917.4	0
26	19963.2	0	19963.2
27	19737.6	5325.53	14412.0
28	984.569	446.814	537.755
29	3263.97	-5529.48	8793.45
30	-14970.9	-322.572	-14648.4
31	14651.7	14454.5	197.220
32	-14993.9	-14646.5	-347.421
33	3629.42	1829.57	1799.85
34	4555.55	-3406.15	7961.70
35	-14981.8	-1807.34	-13174.5
36	15230.4	12054.8	3175.57
37	-14744.8	-11805.1	-2939.67
38	-84.4118	-48.1227	-36.2892
39	5806.93	-1620.64	7427.58
40	-15000.0	-2975.73	-12024.3
41	11977.8	11844.8	132.959
42	-14727.4	-13222.0	-1505.38
43	2567.24	1285.12	1282.12
44	6497.45	-266.965	6764.42
45	-14999.2	-3983.94	-11015.2
46	13523.0	13980.4	-457.414
47	-14384.2	-14410.0	25.7824

Tabela 5.68: Tensões normais nas barras - caso misto (sem restrições de Euler).

VAR. PROJETO	Figura 5.31	contínuo	Ref.[125]	Ref.[126]	misto
A_3	3.8	2.89343	2.42	2.61	2.7
A_4	3.4	2.62198	2.35	2.56	2.7
A_5	0.8	0.71548	0.82	0.69	0.8
A_7	0.9	0.1	0.10	0.47	0.1
A_8	0.9	0.91066	0.86	0.80	0.8
A_{10}	1.8	1.28430	1.15	1.13	1.4
A_{12}	2.1	1.76153	1.77	1.71	2.0
A_{14}	1.2	0.69899	0.67	0.77	0.7
A_{15}	1.6	1.08315	0.86	1.09	1.0
A_{18}	2.1	1.39413	1.24	1.34	1.5
A_{20}	0.7	0.24637	0.33	0.36	0.3
A_{22}	0.9	0.87163	1.22	0.97	1.1
A_{24}	1.7	1.04533	0.93	1.00	1.1
A_{26}	1.7	1.13265	0.86	1.03	1.2
A_{27}	1.4	0.87627	0.69	0.88	0.9
A_{28}	0.9	0.14230	0.15	0.55	0.1
A_{30}	3.7	2.92196	2.46	2.59	2.5
A_{31}	1.5	0.71886	0.90	0.84	0.9
A_{33}	0.7	0.10002	0.10	0.25	0.1
A_{35}	2.9	3.04432	2.74	2.86	2.7
A_{36}	0.7	0.88077	0.92	0.92	1.0
A_{38}	1.6	0.1	0.10	0.67	0.1
A_{40}	3.7	3.12594	2.94	3.06	3.0
A_{41}	1.6	1.06141	1.13	1.04	1.2
A_{43}	0.7	0.1	0.10	0.10	0.1
A_{45}	4.5	3.20907	3.12	3.13	3.3
A_{46}	1.6	1.13375	1.10	1.12	1.2
X_2	60.0	103.68612	107.1	107.76	98.09310
X_4	60.0	84.73633	91.2	89.15	85.86931
Y_4	120.0	136.14462	122.8	137.98	123.81572
X_6	60.0	64.15087	74.2	66.75	73.79714
Y_6	240.0	262.48916	241.4	254.47	251.10542
X_8	60.0	49.35989	65.5	57.38	62.03375
Y_8	360.0	346.81739	324.6	342.16	351.65057
X_{10}	30.0	43.94032	57.1	49.85	51.28319
Y_{10}	420.0	404.62385	400.4	417.17	430.62751
X_{12}	30.0	41.49469	49.3	44.66	40.99478
Y_{12}	480.0	466.17855	472.3	475.35	480.36005
X_{14}	30.0	41.05254	47.4	41.09	38.75064
Y_{14}	540.0	522.90204	507.5	513.15	512.03611
X_{20}	30.0	23.33406	3.9	17.90	15.76198
Y_{20}	600.0	599.23280	586.5	597.92	585.10683
X_{21}	90.0	98.68882	83.3	93.54	89.68743
Y_{21}	600.0	618.77405	636.0	623.94	618.62011
PESO (lb)	2446.8	1875.154	1850.4	1900.0	1913.751

Tabela 5.69: Variáveis de projeto da melhor solução para os casos contínuo e misto (com restrições de Euler).

BARRA	σ_{Ni}^1	σ_{Ei}^1	σ_{Ni}^2	σ_{Ei}^2	σ_{Ni}^3	σ_{Ei}^3
1	2955.20		-7029.14	-90577.8	9984.34	
2	1079.09		-9167.88	-96804.1	10247.0	
3	-12965.3	-90577.8	2028.23		-14993.5	-90577.8
4	-10598.8	-96804.1	4399.48		-14998.3	-96804.1
5	9885.31		6612.02		3273.30	
6	-7636.12	-7665.77	-5516.10	-7665.77	-2120.02	-7665.77
7	10816.7		5332.05		5484.61	
8	4887.85		2729.83		2158.02	
9	-10782.7	-10784.6	-5633.86	-10784.6	-5148.80	-10784.6
10	-14999.6	-22633.0	-7355.65	-22633.0	-7643.99	-22633.0
11	-14922.0	-45354.7	-14922.0	-45354.7	0	
12	-14999.7	-45354.7	0		-14999.7	-45354.7
13	11125.6		-4564.41	-13523.7	15690.0	
14	-5284.38	-13523.7	8238.65		-13523.0	-13523.7
15	-4045.97	-12903.9	8857.33		-12903.3	-12903.9
16	9564.01		-6775.39	-12903.9	16339.4	
17	-14999.6	-36805.0	-14999.6	-36805.0	0	
18	-10933.1	-36805.0	0		-10933.1	-36805.0
19	-5522.46	-5523.55	-5522.46	-5523.55	0	
20	13801.7		0		13801.7	
21	-9704.00	-42189.3	-9704.00	-42189.3	0	
22	-14997.1	-42189.3	0		-14997.1	-42189.3
23	12940.8		12940.8		0	
24	19999.4		0		19999.4	
25	12940.0		12940.0		0	
26	19998.1		0		19998.1	
27	19999.8		6885.94		13113.8	
28	-2188.39	-2188.95	-1063.68	-2188.95	-1124.71	-2188.95
29	5279.45		-5124.04	-102976.	10403.5	
30	-14469.7	-102976.	527.412		-14997.1	-102976.
31	3742.42		5341.95		-1599.53	-7089.23
32	-7080.18	-7089.23	-6999.25	-7089.23	-80.9349	-7089.23
33	1124.65		548.487		576.168	
34	5791.28		-3380.31	-49340.3	9171.59	
35	-14999.8	-49340.3	-1223.12	-49340.3	-13776.7	-49340.3
36	384.223		4858.58		-4474.35	-5232.84
37	-1122.33	-5232.84	-5232.51	-5232.84	4110.18	
38	3039.27		1525.28		1513.99	
39	5877.37		-1722.26	-22662.4	7599.63	
40	-14999.9	-22662.4	-2839.47	-22662.4	-12160.5	-22662.4
41	1035.89		3413.93		-2378.04	-3306.96
42	-824.137	-3306.96	-3306.02	-3306.96	2481.88	
43	4207.12		2101.82		2105.30	
44	6257.45		-413.497	-20177.2	6670.94	
45	-14999.8	-20177.2	-3957.48	-20177.2	-11042.4	-20177.2
46	1251.99		2330.79		-1078.80	-2492.48
47	-1572.65	-2492.48	-2492.18	-2492.48	919.526	

Tabela 5.70: Tensões normais e de Euler – caso contínuo.

BARRA	σ_{Ni}^1	σ_{Ei}^1	σ_{Ni}^2	σ_{Ei}^2	σ_{Ni}^3	σ_{Ei}^3
1	1599.91		-7439.00	-54965.8	9038.91	
2	212.738		-9161.09	-224665.	9373.82	
3	-13577.6	-54965.8	1468.04		-15045.6	-54965.8
4	-11400.1	-224665.	3548.82		-14948.9	-224665.
5	6805.53		6524.35		281.175	
6	-5026.16	-5907.38	-5728.38	-5907.38	702.218	
7	9419.27		4463.57		4955.70	
8	3543.48		3297.01		246.465	
9	-9485.57	-9647.66	-6119.68	-9647.66	-3365.89	-9647.66
10	-14664.1	-15201.9	-6915.39	-15201.9	-7748.66	-15201.9
11	-13797.3	-36755.0	-13797.3	-36755.0	0	
12	-14935.1	-36755.0	0		-14935.1	-36755.0
13	14052.4		-141.243	-9786.49	14193.6	
14	-3062.23	-9786.49	5171.32		-8233.56	-9786.49
15	-2867.12	-11514.9	8682.43		-11549.6	-11514.9
16	13366.6		-4207.98	-11514.9	17574.5	
17	-14948.9	-32736.0	-14948.9	-32736.0	0	
18	-9332.33	-32736.0	0		-9332.33	-32736.0
19	-5092.41	-5101.06	-5092.41	-5101.06	0	
20	19604.7		0		19604.7	
21	-9535.11	-32933.3	-9535.11	-32933.3	0	
22	-14736.1	-32933.3	0		-14736.1	-32933.3
23	12927.2		12927.2		0	
24	19978.4		0		19978.4	
25	12886.9		12886.9		0	
26	19916.1		0		19916.1	
27	19766.0		5278.42		14487.6	
28	-1249.60	-1366.39	-524.283	-1366.39	-725.320	-1366.39
29	3818.02		-5133.03	-50247.2	8951.04	
30	-15077.9	-50247.2	-503.756	-50247.2	-14574.2	-50247.2
31	3998.20		4508.08		-509.880	-5146.22
32	-4989.43	-5146.22	-4968.43	-5146.22	-20.9996	-5146.22
33	4250.62		2076.42		2174.20	
34	4808.38		-3131.83	-46515.5	7940.21	
35	-15089.6	-46515.5	-2005.62	-46515.5	-13084.0	-46515.5
36	4022.90		3998.66		24.2402	
37	-3985.15	-4134.00	-3998.09	-4134.00	12.9373	
38	-498.367	-539.447	-226.326	-539.447	-272.042	-539.447
39	5678.42		-1571.05	-24331.1	7249.47	
40	-15073.0	-24331.1	-3128.27	-24331.1	-11944.8	-24331.1
41	1319.84		2667.30		-1347.46	-3240.80
42	-2321.03	-3240.80	-3159.01	-3240.80	837.983	
43	3948.43		1966.01		1982.41	
44	5927.18		-473.844	-24143.2	6401.02	
45	-15034.7	-24143.2	-4078.86	-24143.2	-10955.8	-24143.2
46	1895.27		2471.84	-2391.43	-576.567	-2391.43
47	-1614.88	-2391.43	-2337.24	-2391.43	722.363	

Tabela 5.71: Tensões normais e de Euler – caso contínuo – Ref [125].

BARRA	σ_{Ni}^1	σ_{Ei}^1	σ_{Ni}^2	σ_{Ei}^2	σ_{Ni}^3	σ_{Ei}^3
1	1978.38		-7582.31	-90879.8	9560.70	
2	854.635		-9202.86	-210968.	10057.5	
3	-12928.1	-90879.8	2110.47		-15038.6	-90879.8
4	-10692.0	-210968.	4274.98		-14967.0	-210968.
5	7913.52		7329.42		584.094	
6	-6625.70	-6655.16	-6707.30	-6655.16	81.5975	
7	6078.60		2999.50		3079.11	
8	1761.30		4082.02		-2320.72	-10822.3
9	-10834.7	-10822.3	-8546.13	-10822.3	-2288.54	-10822.3
10	-15001.9	-19877.5	-7372.51	-19877.5	-7629.41	-19877.5
11	-14742.6	-36120.9	-14742.6	-36120.9	0	
12	-14942.9	-36120.9	0		-14942.9	-36120.9
13	11903.9		-2781.78	-11843.5	14685.7	
14	-4821.67	-11843.5	6780.65		-11602.3	-11843.5
15	-4169.66	-12140.9	8021.37		-12191.0	-12140.9
16	10005.0		-5628.83	-12140.9	15633.8	
17	-14938.7	-35376.0	-14938.7	-35376.0	0	
18	-10344.3	-35376.0	0		-10344.3	-35376.0
19	-6186.74	-7154.31	-6186.74	-7154.31	0	
20	9884.26		0		9884.26	
21	-9664.09	-39436.6	-9664.09	-39436.6	0	
22	-14935.4	-39436.6	0		-14935.4	-39436.6
23	12930.5		12930.5		0	
24	19983.5		0		19983.5	
25	12852.8		12852.8		0	
26	19863.5		0		19863.5	
27	17563.8		5547.95		12015.8	
28	-102.210	-6573.38	-42.5888	-6573.38	-59.6210	-6573.38
29	4184.15		-5545.35	-54140.6	9729.50	
30	-14817.0	-54140.6	228.224		-15045.2	-54140.6
31	4524.63		5134.57		-609.938	-5827.35
32	-5767.81	-5827.35	-5752.45	-5827.35	-15.3569	-5827.35
33	3523.03		1758.89		1764.14	
34	5410.63		-3277.63	-43687.0	8688.25	
35	-14991.6	-43687.0	-1512.64	-43687.0	-13479.0	-43687.0
36	3149.95		3993.87		-843.918	-4731.88
37	-4569.26	-4731.88	-4704.83	-4731.88	135.572	
38	-799.104	-4466.10	-399.020	-4466.10	-400.083	-4466.10
39	5719.41		-1717.47	-25834.0	7436.88	
40	-14964.4	-25834.0	-2905.08	-25834.0	-12059.4	-25834.0
41	-147.018	-3262.12	3101.18		-3248.20	-3262.12
42	-205.061	-3262.12	-3277.02	-3262.12	3071.96	
43	5704.28		2851.96		2852.32	
44	5809.79		-564.208	-19182.2	6374.00	
45	-14984.7	-19182.2	-4023.21	-19182.2	-10961.5	-19182.2
46	2156.62		2660.60		-503.979	-2301.53
47	-1441.44	-2301.53	-2303.12	-2301.53	861.683	

Tabela 5.72: Tensões normais e de Euler – caso contínuo – Ref [126].

BARRA	σ_{Ni}^1	σ_{Ei}^1	σ_{Ni}^2	σ_{Ei}^2	σ_{Ni}^3	σ_{Ei}^3
1	1564.03		-7925.77	-124365.	9489.80	
2	1108.28		-9241.99	-318085.	10350.3	
3	-12553.4	-124365.	2446.43		-14999.8	-124365.
4	-10450.4	-318085.	4549.56		-15000.0	-318085.
5	4428.56		8930.16		-4501.60	-8649.00
6	-1646.19	-8649.00	-7645.81	-8649.00	5999.62	
7	4800.34		2114.65		2685.70	
8	1779.83		3498.60		-1718.77	-12908.3
9	-11393.8	-12908.3	-8110.83	-12908.3	-3282.98	-12908.3
10	-14457.5	-27690.2	-6714.14	-27690.2	-7743.31	-27690.2
11	-13638.3	-39690.6	-13638.3	-39690.6	0	
12	-15000.0	-39690.6	0		-15000.0	-39690.6
13	14252.4		-2467.04	-14172.2	16719.4	
14	-3089.44	-14172.2	7794.01		-10883.5	-14172.2
15	-3845.30	-14294.4	10083.3		-13928.6	-14294.4
16	13147.3		-6493.02	-14294.4	19640.3	
17	-14610.2	-39600.0	-14610.2	-39600.0	0	
18	-11195.8	-39600.0	0		-11195.8	-39600.0
19	-4992.44	-6209.60	-4992.44	-6209.60	0	
20	16849.9		0		16849.9	
21	-9395.84	-55278.9	-9395.84	-55278.9	0	
22	-14520.8	-55278.9	0		-14520.8	-55278.9
23	12941.0		12941.0		0	
24	19999.7		0		19999.7	
25	12384.2		12384.2		0	
26	19139.3		0		19139.3	
27	19289.9		6133.21		13156.7	
28	2205.86		1244.16		961.700	
29	3222.42		-5942.89	-46750.1	9165.32	
30	-14474.0	-46750.1	307.168		-14781.2	-46750.1
31	3960.88		5451.87		-1490.99	-5604.34
32	-4209.63	-5604.34	-5528.50	-5604.34	1318.87	
33	3038.48		1454.55		1583.93	
34	4535.00		-3538.80	-31300.7	8073.79	
35	-15000.0	-31301.9	-1688.90	-31301.9	-13311.1	-31301.9
36	3705.92		4109.59		-403.666	-4159.76
37	-3598.27	-4159.82	-4077.43	-4159.82	479.160	
38	2985.85		1521.18		1464.67	
39	5638.58		-1611.11	-21800.3	7249.69	
40	-15000.0	-21799.6	-3072.14	-21799.6	-11927.9	-21799.6
41	3251.85		2968.01		283.839	
42	-3306.97	-3419.03	-2985.44	-3419.03	-321.530	-3419.03
43	2961.44		1470.81		1490.63	
44	6453.17		-264.201	-25326.0	6717.37	
45	-14853.7	-25326.0	-3934.77	-25326.0	-10918.9	-25326.0
46	2249.73		2258.76		-9.02720	-2899.18
47	-2865.10	-2899.18	-2572.83	-2899.18	-292.276	-2899.18

Tabela 5.73: Tensões normais e de Euler – caso misto.

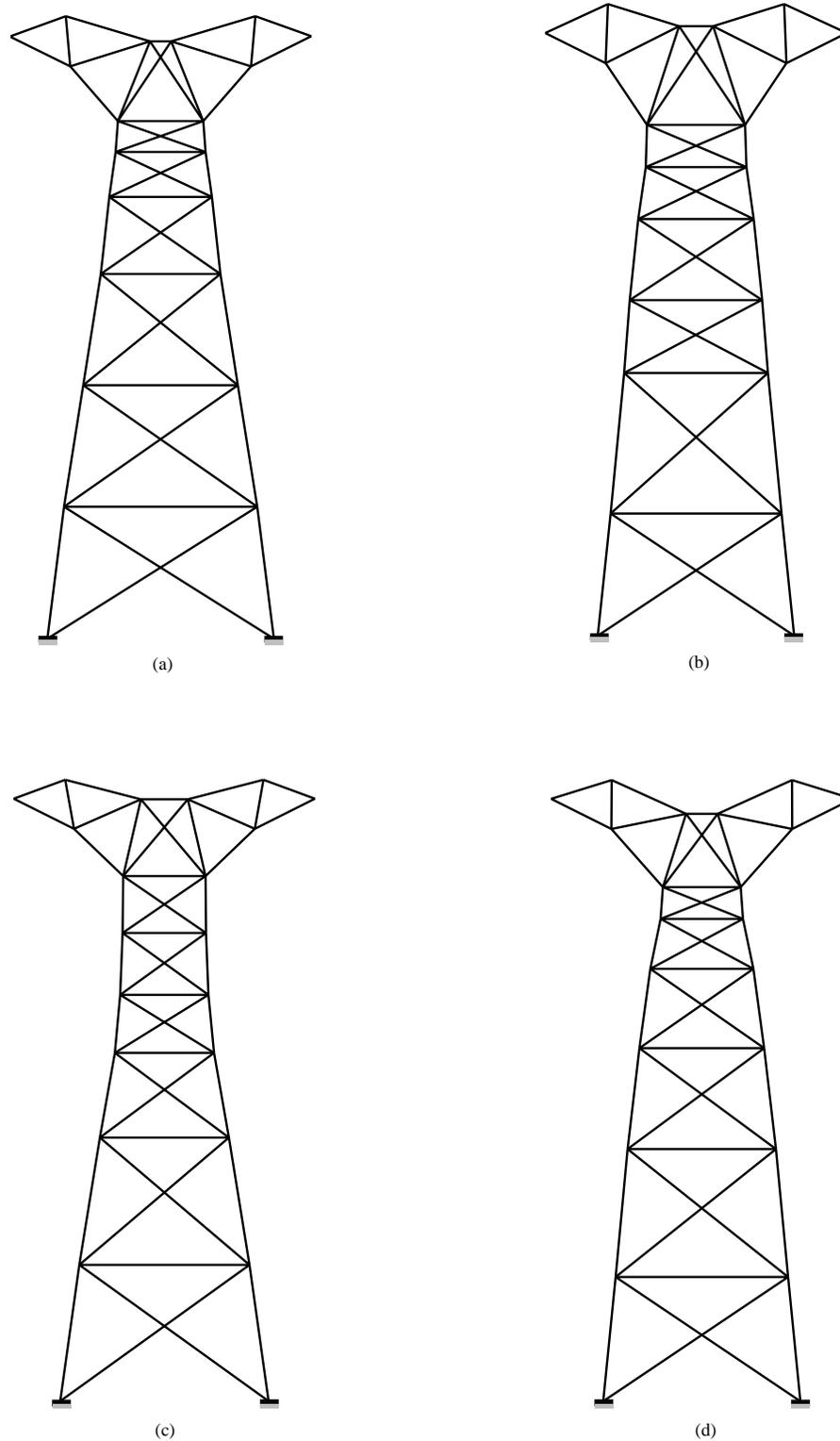


Figura 5.32: Forma ótima - (a) contínuo (tensões normais); (b) misto (tensões normais); (c) contínuo (tensões normais e de Euler); (d) misto (tensões normais e de Euler).

Analisando-se os resultados comparativos desse exemplo, verifica-se que são encontradas algumas restrições violadas no caso contínuo da Tabela 5.71 respectivo à

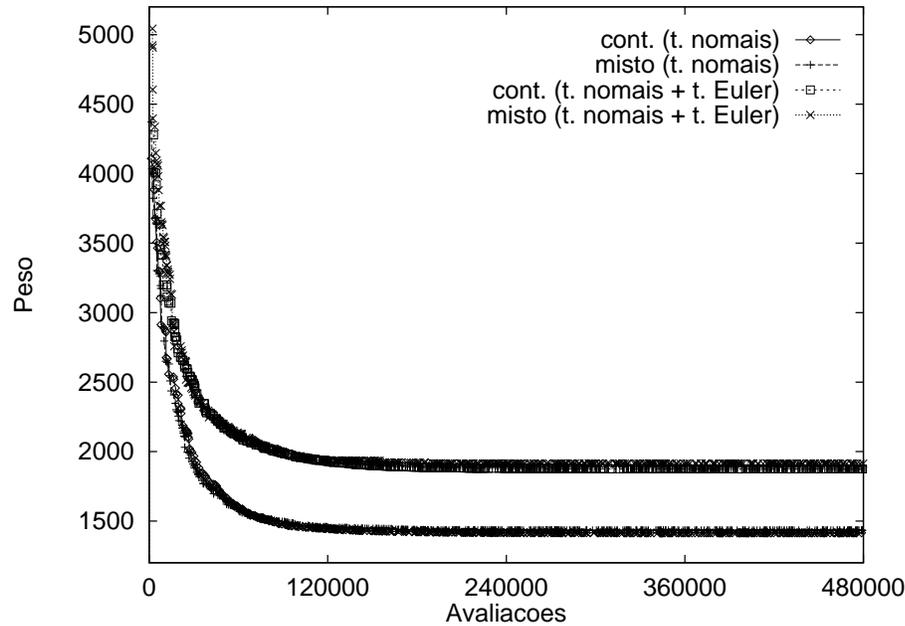


Figura 5.33: Peso do melhor elemento da população em 10 execuções.

Referência [125] na barra 15 (tensão de Euler - carregamento 3), e também para o caso contínuo da Tabela 5.72 respectivo à [126] na mesma barra 15 (tensão de Euler - carregamento 3). Todos os resultados aqui obtidos respeitaram rigorosamente as restrições não apresentando nenhum tipo de violação.

5.6.1 Comentários

Foi utilizado um algoritmo genético “steady-state” com codificação real em dois problemas de otimização de peso, incluindo variáveis de forma, de duas treliças, com variáveis mistas. Os resultados obtidos nas análises são bons e até melhores do que os encontrados na literatura. Apesar do número de avaliações das funções objetivo, ter sido elevado, as restrições impostas ao problema foram rigorosamente satisfeitas, o que não ocorreu plenamente com os resultados apresentados na literatura. Alternativas podem ser introduzidas para reduzir o número de avaliações como, por exemplo, a introdução de conhecimento sobre o problema [30], a utilização de um algoritmo genético geracional paralelo [98] ou a utilização de algoritmos híbridos [2].

5.7 Otimização da estrutura de uma ponte

5.7.1 Introdução

Neste exemplo analisa-se a estrutura de uma ponte rodoviária em concreto protendido com 178 metros de comprimento e apoiada em quatro pilares verticais engastados em suas bases e com aparelhos de apoio elásticos em seus topos [30].

5.7.2 A geometria

A viga principal é uma seção em “caixão” com duas células, mostrada esquematicamente na Figura 5.34, onde HG é uma variável de projeto que assume um valor mínimo de 0.2 m. A espessura da laje inferior tem 0.2 m e a superior 0.3 m. Todas as almas têm valores iguais a 0.4 m. Os ângulos mostrados na Figura 5.34 são $A=30^\circ$, $B=40^\circ$, $C=45^\circ$, $D=20^\circ$ e os pontos 2 e 3 estão a uma distância de 62.0 m da viga principal e os pontos 1 e 4 a 17.886 m e 22.912 m, respectivamente, da viga principal. Uma distância mínima de 60 m entre os pilares P2 e P3 deve ser satisfeita, devido as limitações da construção, bem como uma distância de 5 m entre P1 e P2 e entre P3 e P4. Os comprimentos mínimos dos balanços são de 5 m.

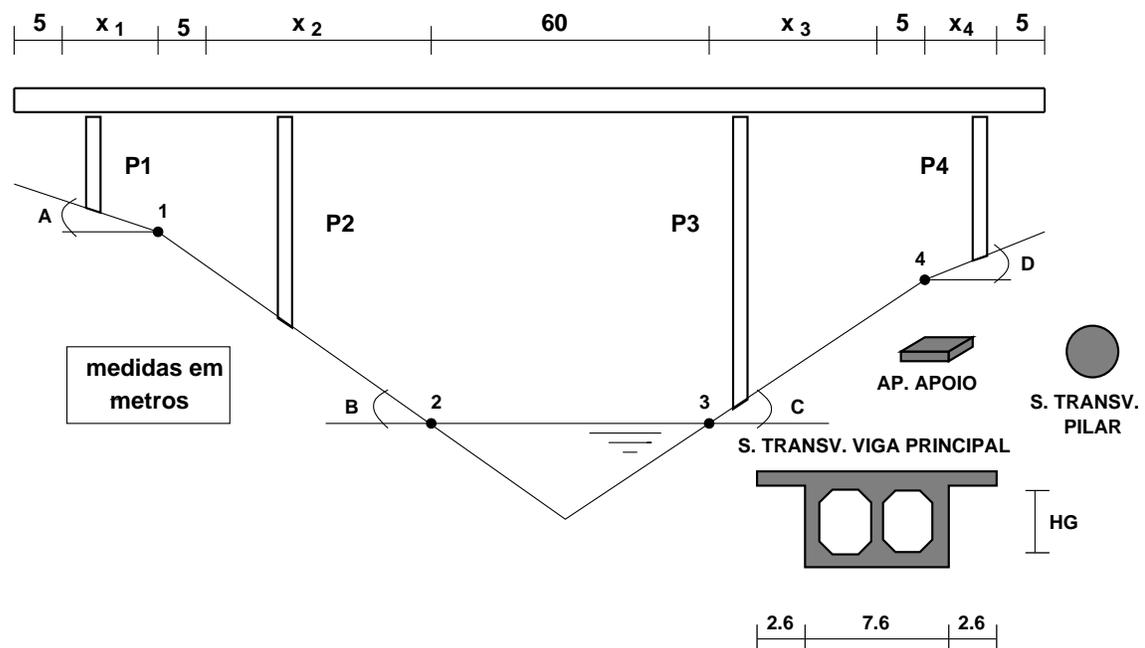


Figura 5.34: A estrutura da ponte.

As variáveis de projeto que definem a locação dos pilares são denotadas por: $x_i, i = 1, 4$ e são submetidas aos limites inferior e superior da seguinte forma:

$$0 \leq x_1 \leq 5 \quad 1 \leq x_2 \leq 50 \quad 1 \leq x_3 \leq 35 \quad 0 \leq x_4 \leq 8 \quad (5.1)$$

A última variável x_5 é necessária para definir completamente o modelo estrutural da viga principal e está associada à distância HG (ver Figura 5.34). Neste ponto alguns conhecimentos do domínio são introduzidos: para este tipo de ponte a razão

entre o maior vão L_{max} (ver Figura 5.36) e a altura da viga principal é usualmente dada no intervalo $[24, 28]$. Logo, a variável x_5 é definida como

$$x_5 = \frac{L_{max}}{0.2 + HG + 0.3} \quad (5.2)$$

com limites

$$24 \leq x_5 \leq 28.$$

Com essa decisão todos os projetos infactíveis onde HG não é compatível com

$$L_{max} = \max\{L_1, L_2, \dots, L_5\} = \text{função}(x_1, x_2, x_3, x_4) \quad (5.3)$$

não serão gerados nem testados pelo algoritmo. Com esse tipo de restrição diminui-se o espaço de busca de forma vantajosa. Por simplicidade somente as cargas devidas ao peso próprio e cargas horizontais devidas ao esforço de frenagem dos veículos são consideradas aqui. Também, a geometria da seção transversal é simples e parametrizações mais complexas podem ser facilmente acomodadas no AG.

5.7.3 A função objetivo

Neste problema busca-se encontrar uma locação ótima para os pilares bem como a altura da viga principal para que se minimize uma função objetivo. Duas funções objetivo são ilustradas:

1. minimização da *compliance* (trabalho de deformação)
2. minimização da reação de apoio máxima

Quando se usa um AG em sua forma geral ou especificamente para uma situação como proposta aqui, o projetista está livre para criar funções objetivo sem preocupar-se com a existência ou não de convexidade, diferenciabilidade ou continuidade. Uma otimização multi-objetivo - não tratada neste trabalho - também pode ser realizada com pequenas modificações no AG [10], [127].

5.7.4 O desenvolvimento do genótipo e avaliação da aptidão

O desenvolvimento do genótipo, também conhecido como morfogênese [59], é o processo que associa cada cromossomo ao seu correspondente fenótipo. No problema considerado aqui, o cromossomo é capaz de gerar o projeto de uma ponte como esquematicamente mostrado na Figura 5.35.

A função aptidão usada foi simplesmente associada à função objetivo sem a necessidade de termos adicionais de penalização devido ao fato de não existirem restrições explícitas devidas às escolhas feitas para:

- variáveis de projeto
- processo de desenvolvimento de genótipo
- processo de avaliação da função aptidão

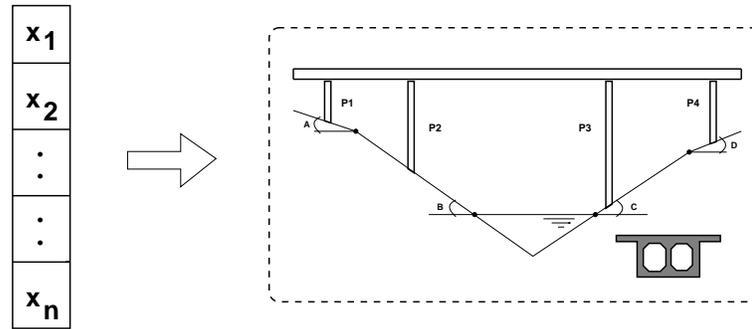


Figura 5.35: Morfogênese: do genótipo para o fenótipo.

Para o primeiro caso, minimização da *compliance*, tem-se

$$f(x) = \int_L q u dL \quad (5.4)$$

onde q representa as forças externas aplicadas à estrutura ao longo de seu comprimento L e u os deslocamentos em suas direções correspondentes.

Para o segundo caso, minimização da reação máxima, tem-se:

$$f(x) = \max\{r_1, r_2, r_3, r_4\} \quad (5.5)$$

onde as reações r_i são positivas (compressão nos pilares). Durante o processo de evolução os projetos que apresentam reações negativas são descartados. O desenvolvimento do processo de avaliação da função aptidão é dado pelos seguintes passos:

1. decodificação do cromossomo em variáveis de projeto $\{x_1, x_2, x_3, x_4, x_5\}$;
2. definição da geometria e cálculo do carregamento;
3. consideração da ponte com apoios rígidos e cálculo das reações;
4. definição da seção transversal para cada pilar de acordo com as reações obtidas no passo 2;
5. cálculo das dimensões dos aparelhos de apoio elastoméricos;
6. cálculo das constantes elásticas para cada conjunto pilar + aparelho de apoio;
7. resolução da estrutura usando as constantes elásticas obtidas no passo 5;
8. obtenção da aptidão.

Todos esses passos são detalhados a seguir.

Passo 1

As variáveis de projeto $\{x_1, x_2, x_3, x_4, x_5\}$ são decodificadas a partir do cromossomo – uma cadeia de caracteres binários.

Passo 2

Com a utilização da equação (5.3) e (5.2) HG é calculada e a geometria é definida. Adotando-se $\gamma_c = 2.5 \text{ tf/m}^3$ para o peso específico do concreto, o carregamento permanente é calculado.

Passo 3

As reações de apoio são calculadas para o modelo mostrado na Figura 5.36 para que se defina a seção transversal de cada pilar (cinco elementos finitos de viga foram utilizados no modelo discreto).

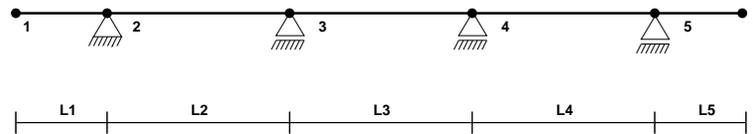


Figura 5.36: Modelo com apoios rígidos.

Passo 4

As seções circulares dos pilares foram escolhidas pela análise de vários critérios. Duas situações são consideradas, uma em que satisfaz-se a área mínima de concreto ($A_{c_{min}}$) necessária para a carga de compressão e outra em que considera-se o coeficiente de flambagem (λ). Para a área mínima, tem-se

$$A_{c_{min}} \geq \frac{N_d}{0.85f_{cd} + \rho f_{sd}}$$

onde N_d é a carga normal aplicada no pilar acrescida de um coeficiente de segurança ($N_d = N \times \gamma_f$), f_{cd} é a máxima resistência à compressão do concreto dividida por um coeficiente de segurança ($f_{cd} = f_{ck}/\gamma_f$). Um $f_{ck} = 30 \text{ MPa}$ foi adotado e $\gamma_f = 1.4$ [128] foi utilizado. A razão entre a área de aço e a área de concreto na seção transversal, $\rho = A_s/A_c$, não é conhecida *a priori* e como estimativa foi considerada como 0.8% neste exemplo. Finalmente, a resistência máxima à compressão do aço foi definida como $f_{sd} = 420 \text{ MPa}$. Para uma seção circular, tem-se

$$D_{min} \geq \sqrt{\frac{4A_{c_{min}}}{\pi}}.$$

O coeficiente de flambagem de um pilar é definido como $\lambda = l_e/i$ onde l_e é o comprimento de flambagem e i denota o raio de giração da seção transversal. Neste exemplo l_e é duas vezes o comprimento do pilar, $l_e = 2LP$, e o raio de giração

$$i = \sqrt{\frac{J_p}{A}}$$

acarretando $i = D/4$ onde D é o diâmetro da seção transversal. A norma brasileira [128] estabelece três classes de pilares de acordo com seus coeficientes de flambagem:

$$0 \leq \lambda \leq 40 \quad , \quad 40 \leq \lambda \leq 80 \quad \text{e} \quad 80 \leq \lambda \leq 200$$

correspondendo a curto, médio e esbelto, respectivamente.

Como uma ilustração da flexibilidade do AG proposto, quatro diferentes critérios para definição de D são considerados a seguir:

Critério 1

O primeiro critério corresponde a uma escolha subjetiva de projetista em particular. Ele/Ela escolhe o valor final de D de acordo com um processo racional que pode ser resumido pelo conjunto de regras dadas a seguir:

se	$(D_l < 1.5m)$	e	$(D_i < 1.5m)$	e	$(D_s \leq 1.5m)$	então	$D = D_s$	
se	$(D_l < 1.5m)$	e	$(D_i \leq 1.5m)$	e	$(D_s \geq 1.5m)$	então	$D = D_i$	
se	$(D_l \leq 1.5m)$	e	$(D_i \geq 1.5m)$	e	$(D_s > 1.5m)$	então	$D = D_i$	
			se	$(D_l \geq 1.5m)$	e	$(D_s \leq 3.0m)$	então	$D = D_i$
se	$(D_l < 3.0m)$	e	$(D_i \leq 3.0m)$	e	$(D_s \geq 3.0m)$	então	$D = D_i$	
se	$(D_l \leq 3.0m)$	e	$(D_i \geq 3.0m)$	e	$(D_s > 3.0m)$	então	$D = D_i$	
se	$(D_l \geq 3.0m)$	e	$(D_i > 3.0m)$	e	$(D_s > 3.0m)$	então	$D = D_l$	
				se	$(D < D_{min})$	então	$D = D_{min}$	

Critério 2

O segundo critério emprega o coeficiente de flambagem λ . Quatro variáveis adicionais $\{x_6, x_7, x_8, x_9\}$ são introduzidas e correspondem aos coeficientes de flambagem dos pilares 1, 2, 3 e 4, respectivamente.

Dois possibilidades são consideradas:

- opção (a): O primeiro e o último pilar devem ser curtos:

$$20 \leq \lambda_1, \lambda_4 \leq 40 \quad \text{e} \quad 40 \leq \lambda \leq 200$$

- opção (b): Qualquer coeficiente de flambagem é permitido no intervalo:

$$20 \leq \lambda_1, \lambda_2, \lambda_3, \lambda_4 \leq 200$$

Critério 3

Este é uma simples modificação do critério anterior. Após a aplicação do critério 2, um único valor de D é adotado: o maior deles. Como no caso anterior, as duas opções são consideradas.

Critério 4

Neste critério o coeficiente de flambagem não é explicitamente utilizado. Para tanto, é introduzida uma variável adicional, x_6 , representando um único valor do diâmetro para todos os pilares. Essa variável é codificada com 14 bits e limitada no intervalo $x_6 \in [2.4, 10]$ (em metros).

Observação

Para os critérios 2, 3 e 4, duas variantes da função aptidão original são consideradas:

$$f(x) = \int_L q u dL \times VP \quad (5.6)$$

$$f(x) = \max\{r_1, r_2, r_3, r_4\} \times VP \quad (5.7)$$

onde VP representa o volume total de concreto dos pilares. A idéia aqui é penalizar projetos onde são encontrados pequenos valores de *compliance* pela utilização de pilares com grandes diâmetros o que leva a grandes valores de VP.

Adotou-se “(x)” como notação para indicar que o critério utilizado tem a função aptidão original multiplicada por VP.

Passo 5

Usando-se as reações de apoio obtidas no Passo 2, as dimensões dos aparelhos de apoio podem ser calculadas. Por simplicidade um único tipo de aparelho foi considerado para todos os pilares e o mesmo foi calculado através da reação de apoio máxima obtida na avaliação. O material do aparelho de apoio tem módulos de elasticidade longitudinal e transversal iguais a $E_{bp} = 4$ MPa e $G_{bp} = 1.3$ MPa, respectivamente. O dimensionamento leva em conta as forças normais, transversais bem como a rotação e a flambagem do aparelho de apoio como mostrado a seguir:

- para a força normal deve-se ter

$$A_{bp} \geq \frac{N}{\sigma_c}$$

onde $A_{bp} = a \times b$ é a área do aparelho de apoio e σ_c é o limite da resistência à compressão (15 MPa). Por simplicidade foi assumida uma seção quadrada com $a = b$.

- para a carga transversal

$$h_{bp} \geq \frac{G_{bp} \times a \times b}{(a + b) \times \sigma_c}$$

onde h_{bp} é a altura do aparelho de apoio.

- para a rotação deve-se ter

$$h_{bp} \geq a \sqrt{\frac{\tan \alpha}{3}}$$

onde α é a rotação do apoio e considerada no dimensionamento como a de maior valor no conjunto de pilares. Sua obtenção se faz no Passo 3.

- e para a flambagem

$$h_{bp} \geq \frac{a}{\sigma_f}$$

onde σ_f tem o valor de 5 MPa.

Passo 6

Considerando-se as características estruturais de cada pilar e de seus respectivos aparelhos de apoio, o modelo para a viga principal pode ser melhor representado pela introdução de apoios elásticos.

O resultado é o modelo mostrado na Figura 5.37 e as constantes elásticas são calculadas a partir do modelo mostrado na Figura 5.38 e dadas por

- rigidez transversal do pilar + aparelho de apoio

$$K_L = \left(\frac{h_{bp}}{G_{bp} \times A_{bp}} + \frac{LP^3}{3 \times E_p \times J_p} \right)^{-1}$$

onde E_p é o módulo de elasticidade do concreto.

- rigidez axial do pilar + aparelho de apoio

$$K_A = \left(\frac{h_{bp}}{E_{bp} \times A_{bp}} + \frac{LP}{E_p \times A_p} \right)^{-1}$$

onde A_p é a área da seção transversal do pilar.

- rigidez à rotação do pilar + aparelho de apoio

$$K_R = \left(\frac{h_{bp}}{E_{bp} \times J_{bp}} + \frac{LP}{E_p \times J_p} \right)^{-1}$$

onde J_{bp} é o momento de inércia da seção transversal do aparelho de apoio.

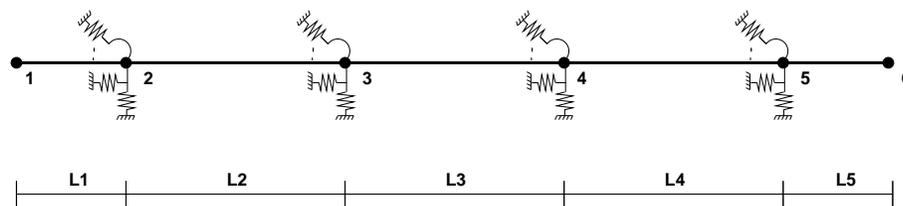


Figura 5.37: Modelo com apoios elásticos.

Passo 7

Nesse passo, as cargas permanentes bem como a carga horizontal devida à frenagem são aplicadas no modelo da Figura 5.37. De acordo com a Norma Brasileira [129], para uma pista de 12 m de largura de uma ponte de Classe 45, a carga de frenagem é calculada a partir dos esquemas mostrados nas Figuras 5.39 e 5.40. No primeiro caso deve-se ter uma carga distribuída de

$$5\%(0.5tf/m^2 \times 3.0m + 0.3tf/m^2 \times 9.0m) \times 178m = 37.38tf$$

enquanto no segundo caso a carga é

$$0.3 \times 45.0tf = 13.5tf$$

e a maior delas deve ser adotada. Como a pista de rolamento é considerada a mesma para todos os projetos, a carga de frenagem também será constante.

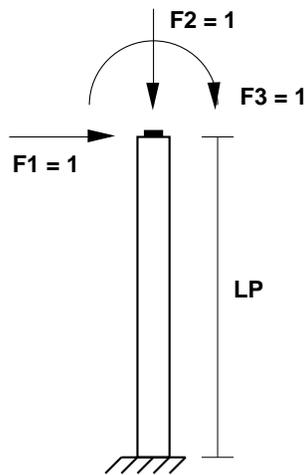


Figura 5.38: Modelo para determinação das constantes elásticas do pilar + aparelho de apoio.

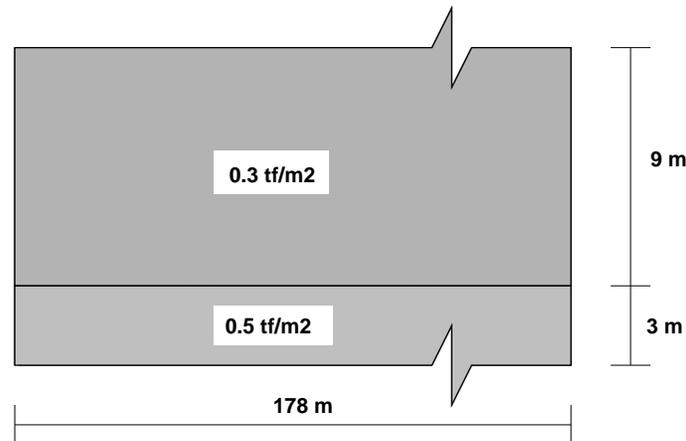


Figura 5.39: Pista com carga uniformemente distribuída.

Passo 8

Finalmente, com o modelo estrutural resolvido, pode-se calcular a função aptidão para um indivíduo/projeto particular de acordo com as expressões 5.4 ou 5.5.

5.7.5 Parametrização

Uma codificação binária e um algoritmo genético “geracional” foram usados e 10 bits foram empregados para codificar todas as variáveis de projeto exceto x_1 (8 bits) e x_2 (12 bits) gerando um cromossomo de 50 bits de comprimento para o critério 1. Para os critérios 2 e 3, 40 bits adicionais (correspondendo às variáveis x_6, \dots, x_9) foram usados levando a um cromossomo com 90 bits. Finalmente, para o critério 4, foi empregado um cromossomo de 64 bits de comprimento.

Foram realizadas 10 execuções onde cada uma iniciou o processo de evolução

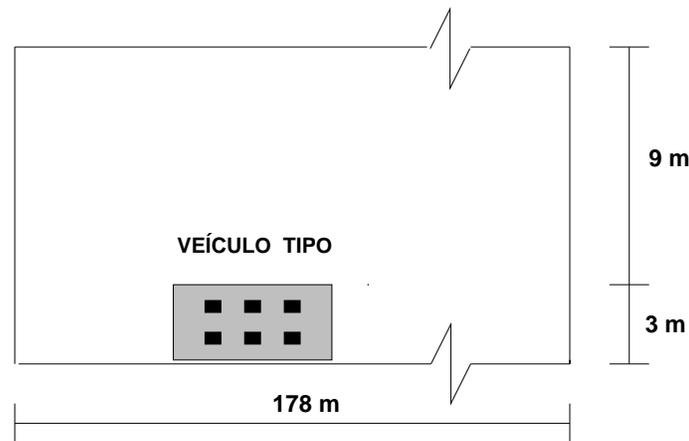


Figura 5.40: Pista com carga do veículo padrão.

com uma população diferente gerada aleatoriamente com 100 indivíduos e evoluída em 120 gerações.

A probabilidade de recombinação foi definida com o valor $p_c = 0.8$ e a de mutação $p_m = 0.04$. Três operadores de recombinação – um ponto, dois pontos e uniforme – foram usados, aplicados com as probabilidades de 0.16, 0.32 e 0.32, respectivamente.

Foi adotado um procedimento elitista: o melhor indivíduo (mais uma cópia com mutação), em cada geração, é copiado para a próxima geração a fim de não haver uma possível perda de material genético com boa qualidade da geração anterior.

5.7.6 Minimização da *compliance*.

A Tabela 5.74 mostra as variáveis de projeto obtidas na otimização, a Tabela 5.75 mostra os vãos, a altura HG da viga principal, o volume de concreto dos pilares VP e o volume de concreto total da ponte VC, a Tabela 5.76 mostra o diâmetro, a altura dos pilares, as reações (passo 3 e passo 7) e as constantes dos apoios elásticos, para cada projeto referente a cada critério. A Tabela 5.77 mostra as dimensões dos aparelhos de apoio.

As Figuras 5.41 e 5.42 mostram a aptidão do melhor indivíduo da população ao longo das gerações para a melhor e a pior execução e também a média em uma série de 10 execuções independentes.

A Tabela 5.78 resume os resultados obtidos considerando-se a minimização da *compliance*. Observa-se que o critério 3b acarreta o mínimo valor da *compliance* que é encontrado usando-se o maior volume de concreto, $VC=63305 \text{ m}^3$, que não é aceitável como solução devido a existência de pilares com diâmetros excessivos. O critério 1 produz o maior valor de *compliance* com o menor valor de concreto $VC=3140 \text{ m}^3$. Entretanto, boas soluções devem ser aquelas obtidas pela minimização do produto da *compliance* por VP, e neste caso, o critério 4(x) leva a uma solução com o volume de concreto $VC=2180.7 \text{ m}^3$ usando-se pilares com um único diâmetro de 2.4 m sendo, então, a melhor solução encontrada na Tabela 5.78 e a que deve ser adotada.

critério	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	5.0	3.748	0.0	5.521	24.0	-	-	-	-
2a(x)	5.0	4.310	0.0	3.808	24.0	40.0	186.08	148.231	40.0
2a	5.0	4.066	0.0	5.654	24.0	20.0	40.0	40.0	20.0
2b(x)	5.0	5.397	0.0	5.419	24.0	115.718	200.0	200.0	148.973
2b	5.0	4.139	0.0	5.591	24.0	20.0	20.0	20.0	20.0
3a(x)	5.0	5.092	0.0	2.315	24.0	38.749	175.914	181.075	40.0
3a	5.0	4.066	0.0	5.63	24.0	32.532	40.0	190.928	24.125
3b(x)	5.0	5.25	0.0	5.45	24.0	110.968	200.0	192.258	146.686
3b	5.0	4.127	0.0	5.591	24.0	30.029	20.0	91.789	65.572
4(x)	5.0	4.115	0.0	5.505	24.0	2.4	-	-	-
4	5.0	4.139	0.0	5.591	24.0	20.0	-	-	-

Tabela 5.74: Variáveis de projeto para minimização da *compliance*.

critério	L_1	L_2	L_3	L_4	L_5	HG	VP	VC
1	10.0	51.251	63.748	42.479	10.521	2.156	1611.5	3140.0
2a(x)	10.0	50.689	64.310	44.191	8.808	2.179	984.4	2518.0
2a	10.0	50.934	64.066	42.346	10.654	2.169	11248.6	12780.0
2b(x)	10.0	49.603	65.397	42.580	10.419	2.225	398.6	1941.8
2b	10.0	50.861	64.139	42.409	10.591	2.172	38707.4	40239.4
3a(x)	10.0	49.908	65.091	45.685	7.315	2.212	1930.5	3471.0
3a	10.0	50.934	64.066	42.369	10.630	2.169	15477.4	17008.7
3b(x)	10.0	49.749	65.250	42.549	10.451	2.225	593.9	2135.8
3b	10.0	50.873	64.127	42.409	10.591	2.172	61773.0	63305.0
4(x)	10.0	50.885	64.115	42.495	10.505	2.171	648.9	2180.7
4	10.0	50.861	64.139	42.409	10.591	2.172	45065.7	46697.8

Tabela 5.75: Vãos obtidos pela minimização da *compliance*.

critério	pilar	diam. (m)	altura (m)	reação (tf) (passo 3)	reação (tf) (passo 7)	K_L	K_A	K_R
1	1	1.788	17.887	643.982	643.840	691.651	486861	71.193
	2	4.708	58.854	1374.186	1374.720	860.126	1024229	71.241
	3	3.6	45.0	1236.138	1235.330	707.540	783340	71.237
	4	2.201	22.010	567.097	567.513	806.306	598863	71.217
2a(x)	1	3.577	17.887	639.656	641.085	2397.917	1944644	71.136
	2	2.510	58.383	1372.822	1370.703	98.125	294095	71.095
	3	2.429	45.0	1282.262	1282.315	182.132	357020	71.1
	4	4.277	21.386	539.161	539.797	2464.243	2324915	71.137
2a	1	7.155	17.887	640.768	640.779	2839.412	7775781	71.466
	2	11.718	58.588	1377.032	1377.081	2711.197	6367586	71.466
	3	9.0	45.0	1241.915	1241.755	2665.497	4890980	71.466
	4	8.823	22.058	568.751	568.853	2845.963	9588846	71.466
2b(x)	1	1.236	17.887	627.244	627.516	194.12	233182	72.019
	2	2.299	57.472	1387.127	1387.801	73.046	250749	72.19
	3	1.8	45.0	1281.107	1278.392	57.511	196539	72.122
	4	1.18	21.972	562.601	564.370	90.168	173077	71.907
2b	1	7.155	17.887	640.084	640.074	2839.778	7775782	71.494
	2	23.411	58.527	1377.392	1377.419	2863.991	25440824	71.494
	3	18.0	45.0	1245.022	1244.995	2860.764	19561118	71.494
	4	8.814	22.035	567.598	567.609	2846.302	9578949	71.494
3a(x)	1	4.169	17.887	632.760	633.299	2594.349	2640111	71.251
	2	4.169	57.728	1374.288	1373.429	625.37	818675	71.247
	3	4.169	45.0	1328.193	1328.384	1063.005	1049962	71.248
	4	4.169	20.842	516.045	516.176	2456.31	2265840	71.250
3a	1	11.718	17.887	640.784	640.857	2869.408	20854983	71.46
	2	11.718	58.588	1376.955	1376.833	2711.126	6367586	71.46
	3	11.718	45.0	1242.362	1242.402	2798.007	8290072	71.46
	4	11.718	22.049	568.365	568.375	2865.13	16918052	71.46
3b(x)	1	2.304	17.887	628.778	630.628	1341.19	807040	72.144
	2	2.304	57.594	1385.971	1382.948	73.197	251283	72.101
	3	2.304	45.0	1276.701	1277.508	149.306	321349	72.114
	4	2.304	21.984	563.37	563.734	919.756	656807	72.139
3b	1	23.415	17.887	640.206	640.226	2874.272	83271326	71.487
	2	23.415	58.537	1377.298	1377.265	2863.895	25445277	71.487
	3	23.415	45.0	1244.703	1244.712	2869.771	33099529	71.487
	4	23.415	22.035	567.619	567.622	2874.103	67594756	71.487
4(x)	1	2.4	17.887	640.388	642.029	1456.604	875779	71.442
	2	2.4	58.547	1376.921	1374.13	81.813	268207	71.405
	3	2.4	45.0	1246.016	1246.937	174.218	348670	71.417
	4	2.4	22.004	566.229	566.458	1022.184	712090	71.438
4	1	20.0	17.887	640.084	640.111	2874.199	60754107	71.495
	2	20.0	58.527	1377.393	1377.348	2854.593	18568100	71.495
	3	20.0	45.0	1245.022	1245.035	2865.568	24149309	71.495
	4	20.0	22.035	567.598	567.602	2873.695	49316652	71.495

Tabela 5.76: Dados dos pilares para minimização da *compliance* para todos critérios.

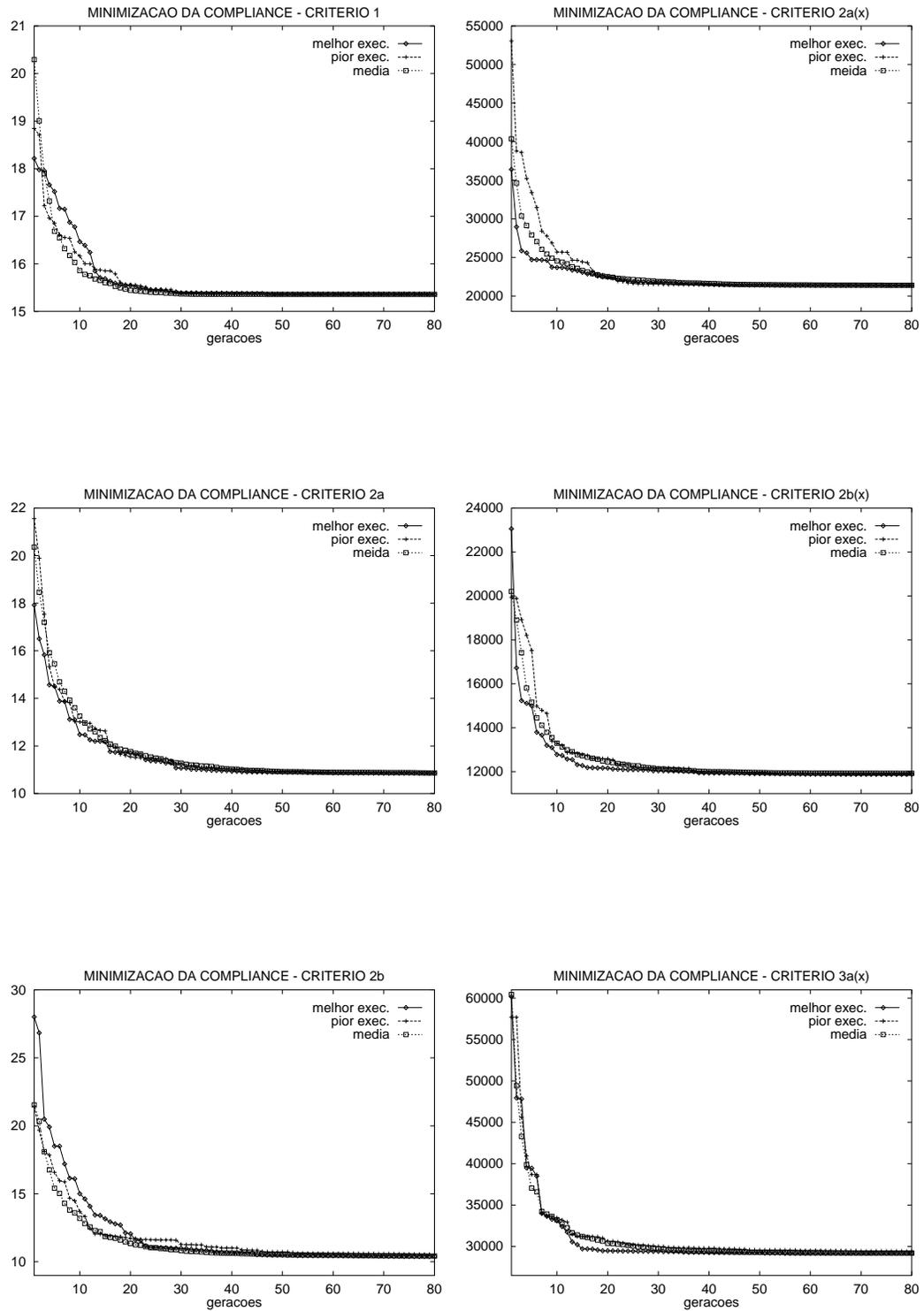


Figura 5.41: Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.

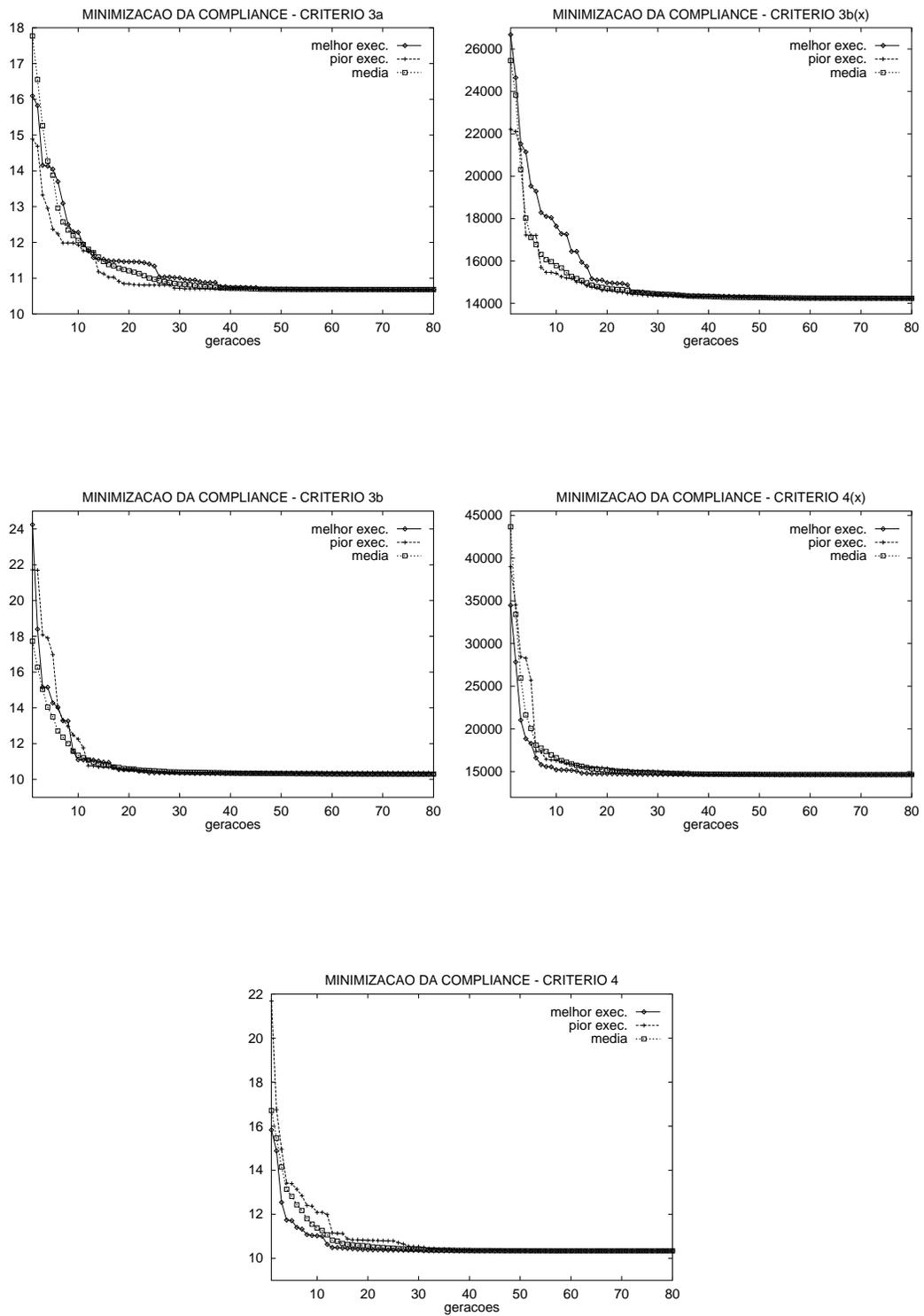


Figura 5.42: Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.

critério	lados (m)	altura (m)
1	0.957	0.127
2a(x)	0.957	0.127
2a	0.958	0.128
2b(x)	0.962	0.128
2b	0.958	0.128
3a(x)	0.957	0.127
3a	0.958	0.128
3b(x)	0.961	0.128
3b	0.958	0.128
4(x)	0.958	0.128
4	0.958	0.128

Tabela 5.77: Dimensões dos aparelhos de apoio obtidas pela minimização da *compliance*.

critério	D_1 (m)	D_2 (m)	D_3 (m)	D_4 (m)	VP (m^3)	VC (m^3)	comp.	comp. $\times VP$
1	1.788	4.708	3.6	2.201	1611.5	3140.0	15.36	
2a(x)	3.577	2.510	2.429	4.277	984.4	2518.0		21379
2a	7.155	11.718	9.0	8.823	11248.6	12780.0	10.86	
2b(x)	1.236	2.299	1.8	1.18	398.6	1941.8		11875
2b	7.155	23.411	18.0	8.814	38707.4	40239.4	10.4	
3a(x)	4.169	4.169	4.169	4.169	1930.5	3471.0		291717
3a	11.718	11.718	11.718	11.718	15477.4	17008.7	10.68	
3b(x)	2.304	2.304	2.304	2.304	593.9	2135.8		14237
3b	23.415	23.415	23.415	23.415	61773.0	63305.0	10.29	
4(x)	2.4	2.4	2.4	2.4	648.9	2180.7		14651
4	20.0	20.0	20.0	20.0	45065.7	46697.8	10.34	

Tabela 5.78: Resumo das análises para o caso de minimização da *compliance*.

5.7.7 Minimização da reação máxima

A Tabela 5.79 mostra as variáveis de projeto obtidas na otimização, a Tabela 5.80 mostra os vãos, a altura HG da viga principal, o volume de concreto dos pilares VP e o volume de concreto total da ponte VC, a Tabela 5.81 mostra o diâmetro, a altura dos pilares, as reações (passo 3 e passo 7) e as constantes dos apoios elásticos, para cada projeto referente a cada critério. A Tabela 5.82 mostra as dimensões dos aparelhos de apoio.

critério	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	5.0	0.0	0.0	0.0	28.0	-	-	-	-
2a(x)	0.0	13.419	1.197	0.0	28.0	40.0	200.0	200.0	40.0
2a	5.0	0.0	0.0	0.0	28.0	20.0	200.0	40.0	40.0
2b(x)	0.0	27.741	12.077	0.0	28.0	199.824	200.0	200.0	200.00
2b	5.0	0.0	0.0	0.0	28.0	20.0	200.0	20.0	200.0
3a(x)	0.0	29.06	7.39	0.0	28.0	189.619	200.0	200.0	193.314
3a	5.0	0.0	0.0	0.0	28.0	135.249	200.0	158.651	159.179
3b(x)	5.0	5.092	0.0	0.0	28.0	37.087	186.237	157.458	40.0
3b	5.0	0.0	0.0	0.0	28.0	39.374	178.104	152.766	40.0
4(x)	5.0	5.018	0.0	0.0	28.0	2.4	-	-	-
4	5.0	0.0	0.0	0.0	28.0	2.4	-	-	-

Tabela 5.79: Variáveis de projeto para minimização da reação máxima.

critério	L_1	L_2	L_3	L_4	L_5	HG	VP	VC
1	10.0	55.0	60.0	48.0	5.0	1.643	1763.8	3182.7
2a(x)	5.0	46.581	74.616	46.803	5.0	2.165	627.1	2157.5
2a	5.0	43.59	80.413	43.997	5.0	2.372	2159.6	1258.5
2b(x)	5.0	32.25	99.818	35.923	5.0	3.065	132.1	1845.8
2b	10.0	55.0	60.0	48.0	5.0	1.643	12479.8	13898.7
3a(x)	5.0	30.94	96.45	40.61	5.0	2.945	196.0	1893.0
3a	10.0	55.0	60.0	48.0	5.0	1.643	699.9	2118.8
3b(x)	10.0	49.908	65.092	48.0	5.0	1.825	1767.0	3224.8
3b	10.0	55.0	60.0	48.0	5.0	1.643	1820.7	3239.6
4(x)	10.0	49.982	65.018	48.0	5.0	2.072	636.4	2093.6
4	10.0	55.0	60.0	48.0	5.0	1.643	655.5	2074.4

Tabela 5.80: Vãos obtidos pela minimização da reação máxima.

As Figuras 5.43 e 5.44 mostram a aptidão do melhor indivíduo da população ao longo das gerações para a melhor e o pior execução e também a média em uma série de 10 execuções independentes.

Observa-se na Tabela 5.83 que os valores das reações máximas de apoio obtidas são semelhantes e foram encontradas com volumes similares de concreto com exceção do critério 2b. Os critérios 3a, 3b e 4 definem pilares com o mesmo diâmetro e com

critério	pilar	diam. (m)	altura (m)	reação (tf) (passo 3)	reação (tf) (passo 7)	K_L	K_A	K_R
1	1	1.786	17.864	645.985	645.870	684.514	486822	62.699
	2	4.96	62.0	1262.517	1262.923	879.985	1078877	62.737
	3	3.6	45.0	1173.429	1171.907	700.073	783301	62.733
	4	2.0	20.0	465.354	465.585	743.538	544232	62.710
2a(x)	1	3.0	15.0	426.87	428.094	2399.875	1630985	80.116
	2	2.029	50.74	1483.463	1482.968	64.732	221523	80.10
	3	1.752	43.803	1486.091	1482.98	56.047	191367	79.947
	4	4.0	20.0	429.615	431.996	2523.731	2174324	80.119
2a	1	3.0	15.0	367.116	368.946	2471.933	1631022	89.505
	2	1.929	48.23	1596.336	1595.495	61.641	210649	89.35
	3	1.64	40.997	1600.057	1595.448	52.554	179209	89.247
	4	4.0	20.0	373.085	376.705	2603.543	2174360	89.51
2b(x)	1	0.6	15.0	-10.67	0.336	19.522	66475	125.14
	2	1.549	38.722	2122.565	2111.906	49.79	169474	136.029
	3	1.317	32.923	2099.896	2088.702	42.457	144327	135.572
	4	0.8	20.0	94.887	105.734	25.891	88093	131.59
2b	1	7.155	17.887	645.985	647.862	2720.192	7775742	62.739
	2	2.48	62.0	1262.517	1256.959	78.542	270390	62.702
	3	18.0	45.0	1173.429	1179.562	2739.442	19561078	62.739
	4	0.8	20.0	465.354	462.903	25.835	87828	61.628
3a(x)	1	1.505	15.0	-14.156	0.321	633.072	411159	127.184
	2	1.505	37.616	2025.186	2007.95	48.371	164638	126.765
	3	1.505	37.61	1998.8	1997.079	48.393	164664	126.765
	4	1.505	20.0	232.606	237.086	298.409	308653	127.09
3a	1	2.48	17.887	645.985	646.783	1514.402	935035	62.729
	2	2.48	62.0	1262.517	1261.022	78.542	270390	62.702
	3	2.48	45.0	1173.429	1174.083	196.365	372199	62.712
	4	2.48	20.0	465.354	465.397	1284.491	836331	62.727
3b(x)	1	4.0	17.887	600.429	600.796	2481.96	2431032	65.025
	2	4.0	57.728	1293.019	1292.445	545.17	753873	65.021
	3	4.0	45.0	1292.274	1292.379	945.334	966840	65.023
	4	4.0	20.0	458.666	458.768	2378.846	2174259	65.025
3b	1	4.0	17.887	645.985	646.296	2455.684	2431021	62.738
	2	4.0	62.0	1262.517	1261.936	456.416	701976	62.734
	3	4.0	45.0	1173.429	1173.682	941.497	966829	62.736
	4	4.0	20.0	465.354	465.372	2354.698	2174248	62.738
4(x)	1	2.4	17.887	601.174	602.18	1433.249	875750	64.972
	2	2.4	57.789	1292.466	1290.888	84.899	271685	64.942
	3	2.4	45.0	1290.576	1290.873	173.879	348641	64.952
	4	2.4	20.0	458.776	459.05	1201.241	783312	64.97
4	1	2.4	17.887	645.985	646.836	1424.603	875740	62.727
	2	2.4	62.0	1262.517	1260.922	69.129	253283	62.696
	3	2.4	45.0	1173.429	1174.127	173.751	348631	62.708
	4	2.4	20.0	465.354	465.399	1195.162	783301	62.725

Tabela 5.81: Dados dos pilares para minimização da reação máxima.

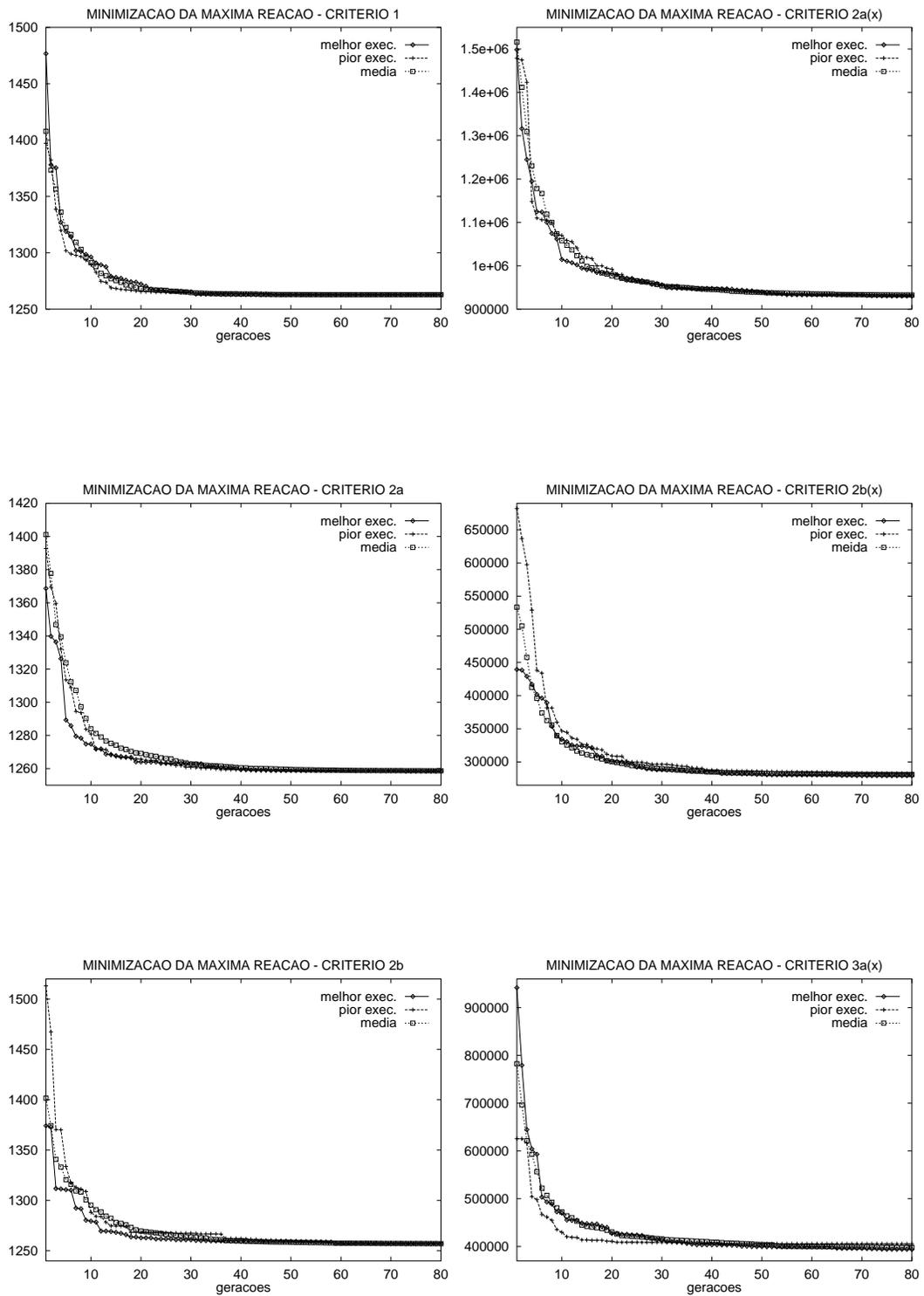


Figura 5.43: Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.

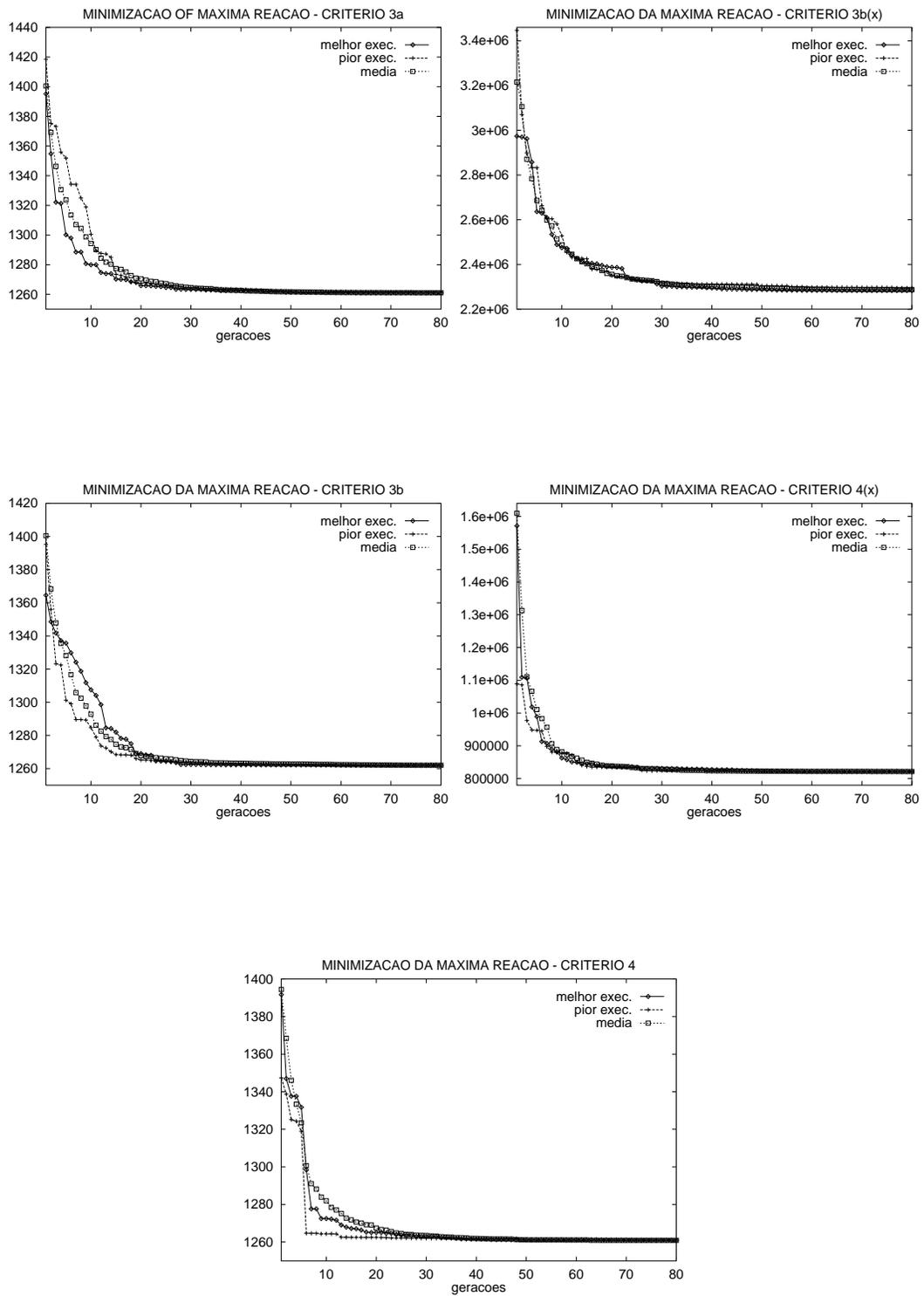


Figura 5.44: Aptidão do melhor indivíduo na melhor e pior execução juntamente com o valor médio em 10 execuções independentes.

critério	lados (m)	altura (m)
1	0.917	0.122
2a(x)	0.995	0.132
2a	1.033	0.138
2b(x)	1.189	0.158
2b	0.917	0.122
3a(x)	1.162	0.155
3a	0.917	0.122
3b(x)	0.928	0.124
3b	0.917	0.122
4(x)	0.928	0.123
4	0.917	0.122

Tabela 5.82: Dimensões dos aparelhos de apoio obtidas pela minimização da reação máxima.

critério	D_1 (m)	D_2 (m)	D_3 (m)	D_4 (m)	VP (m^3)	VC (m^3)	max.r.	max.r. $\times VP$
1	1.786	4.96	3.6	2.0	1763.8	3182.7	1262.9	
2a(x)	3.0	2.029	1.752	4.0	627.1	2157.5		930017
2a	3.0	1.929	1.64	4.0	584.9	2159.6	1258.5	
2b(x)	0.6	1.549	1.317	0.83	132.1	1854.8		279039
2b	7.155	2.48	18.0	0.8	12479.8	13898.7	1257.0	
3a(x)	1.505	1.505	1.505	1.505	196.0	1893.0		393541
3a	2.48	2.48	2.48	2.48	699.9	2118.8	1261.0	
3b(x)	4.0	4.0	4.0	4.0	1767.0	3224.8		2283767
3b	4.0	4.0	4.0	4.0	1820.7	3239.6	1261.9	
4(x)	2.4	2.4	2.4	2.4	636.4	2093.6		8215257
4	2.4	2.4	2.4	2.4	655.5	2074.4	1260.9	

Tabela 5.83: Resumo das análises para o caso de minimização da reação máxima.

valores similares e razoáveis para o projeto. Entre eles, a solução obtida usando-se o critério 4 é a melhor delas, com o menor volume de concreto, $VC=2074.4 \text{ m}^3$ e pilares com diâmetro de 2.4 m.

Quando minimiza-se o produto da reação máxima multiplicada por VP, o mínimo valor é alcançado usando-se o critério $2b(x)$, mas com pilares de diferentes diâmetros.

Os critérios $3a(x)$, $3a$, $3b(x)$, $3b$, $4(x)$ e 4 definiram pilares com o mesmo diâmetro e com valores razoáveis. Entre eles, a solução obtida pelo critério $3a(x)$ usa o menor volume de concreto, $VC=1893 \text{ m}^3$.

Finalmente, comparando-se as duas soluções selecionadas correspondentes aos critérios 4 e $3a(x)$, a obtida pelo critério $3a(x)$ sugere pilares mais “leves”, como esperado (desde que a função objetivo seja multiplicada por VP).

5.7.8 Comentários

Neste exemplo foi apresentado um AG para a localização ótima de pilares de uma ponte.

No lugar de uma codificação em um cromossomo contendo vários parâmetros geométricos necessários para a definição completa da geometria, o que acarretaria um espaço de busca com um grande número de projetos ineficazes ou anti-econômicos, o conhecimento do domínio é introduzido no sentido de que somente a posição dos pilares da ponte e um parâmetro que define a altura da viga, a partir de valores de vãos, são usados como variáveis de projeto.

É introduzido conhecimento adicional no processo de desenvolvimento que gera o projeto da ponte a partir de cada genótipo e também no processo de avaliação da função aptidão de cada projeto candidato.

Como resultado, projetos com as alturas das vigas incompatíveis com os comprimentos dos vãos não são gerados nem testados pelo algoritmo. O espaço de busca é bastante reduzido não existindo a necessidade de explicitar todas as restrições referentes ao comportamento estrutural dentro do algoritmo genético.

Diferentes funções objetivo podem ser facilmente consideradas e, neste exemplo, a minimização da *compliance* e a minimização da reação máxima de apoio foram apresentadas.

5.8 Estratégias para evolução de parâmetros

5.8.1 Introdução

Uma série de métodos iterativos podem ser utilizados na resolução de sistemas de equações como gradiente conjugado, Jacobi, Gauss-Seidel, Gauss-Seidel com relaxação, etc.

Na resolução de certos tipos de problemas pode ser conveniente ou até mesmo necessário a utilização de algum desses métodos iterativos.

Na utilização de alguns desses métodos é necessário definir uma parametrização com o objetivo de se obter desempenho e eficiência. Entretanto, a determinação dessa parametrização pode não ser trivial e se escolhida de maneira inadequada poderá acarretar prejuízos de várias naturezas para a obtenção de uma solução razoável. A parametrização depende, em geral, do problema em questão.

Propõe-se neste exemplo, uma estratégia de co-evolução [130] para se determinar o parâmetro ω a ser utilizado em um algoritmo iterativo do tipo Gauss-Seidel com relaxação (GSR).

Aplica-se esta idéia ao problema de otimização de peso da treliça de 10 barras, já estudada neste texto (Seção 4.1), onde o sistema de equações $Ku = f$ montado em cada avaliação é resolvido de forma iterativa pelo GSR.

Sabe-se que no método de Gauss-Seidel com relaxação, a escolha adequada do parâmetro ω acarreta um menor número de iterações. Pretende-se, então, adaptar o valor de ω ao sistema de equações na solução da treliça de 10 barras. Com isto a avaliação de cada estrutura durante o processo de evolução será obtida mais rapidamente e conseqüentemente o esforço computacional durante todo o processo de evolução será menor.

Algoritmo “Gauss-Seidel com relaxação”

Início

escolha u^0

escolha $\omega \in (0, 2)$

Para $k = 1, \dots$, *limite faça*

Para $i = 1, \dots, n$ *faça*

$$u_i^* = \frac{1}{K_{ii}} \left(f_i - \sum_{j=1}^{i-1} K_{ij} u_j^{k+1} - \sum_{j=i+1}^n K_{ij} u_j^k \right)$$

$$u_i^{k+1} = (1 - \omega) u_i^k + \omega u_i^*$$

Se $\|u^{k+1} - u^k\| \leq \varepsilon$ **então pare**

Fim - Para

Fim

Figura 5.45: O algoritmo de Gauss-Seidel com relaxação – GSR.

Evidentemente, pela natureza do problema não haveria necessidade da utilização de tal técnica, visto que um método direto de solução seria suficiente. Mas, o objetivo aqui é explorar o processo de co-evolução.

O processo de co-evolução desenvolvido nesta análise é implementado em dois algoritmos genéticos que envolvem a evolução de duas populações. Uma população é constituída por estruturas candidatas a um processo de minimização de peso com imposição de restrições de tensões e de deslocamentos. A aptidão dessa população é avaliada por uma função objetivo que envolve o peso da estrutura e uma função de penalização, como visto na Seção 4.1. Esta população é submetida a um processo evolutivo totalmente independente da outra população participante da co-evolução.

A outra população é constituída por elementos que são possíveis valores do parâmetro ω de relaxação do GSR. Nesta população, os elementos concorrem entre si por uma aptidão que é medida pelo número de iterações que eles necessitam ao resolverem os sistemas de equações envolvidos nas obtenções das aptidões dos elementos da população de estruturas.

As duas populações são evoluídas através de dois algoritmos genéticos do tipo “steady-state” com codificação real, onde os processos de seleção e recombinação ocorrem independentemente em cada um deles.

Para a população de estruturas as áreas das barras representam as variáveis de projeto sendo variáveis contínuas definidas no intervalo $[0.1, 40.0]$ e para a população de ω os valores são escolhidos no intervalo $[1, 1.95]$.

Utilizou-se 400 indivíduos para a população de estruturas (pop_A) e 40 indivíduos para a população de ω (pop_B) com 20000 avaliações em cada execução em um total de 10.

O algoritmo que representa ao processo de co-evolução pode ser descrito como: primeiramente, a população inicial de estruturas é dividida em grupos de tal forma que cada um terá o seu sistema de equações resolvido com um valor fixo de ω . O número de grupos ng é obtido pela razão entre a população de estruturas e a população de ω , $ng = pop_A/pop_B$. Os valores iniciais de ω designados para cada um desses grupos são obtidos por uma distribuição uniforme no intervalo $\omega \in [1.0; 1.95]$, Como $pop_A = 400$ e $pop_B = 40$, serão 40 valores de ω distribuídos uniformemente neste intervalo, ou seja, 1.0, 1.025, 1.05, ..., 1.95, significando que cada grupo de 10 indivíduos da população de estruturas será avaliado com um mesmo ω , isto é, grupo 1 (1.0), grupo 2 (1.025), grupo 3 (1.05), ..., grupo 40 (1.95). Dessa forma, tem-se as aptidões das estruturas (peso) e dos parâmetros (número de iterações).

Após a obtenção das aptidões e ordenações dos elementos das populações iniciais de estruturas e parâmetros, passa-se aos processos de seleção, recombinação e inserção tanto na população de estruturas quanto na população de parâmetros, que é repetido até o fim da co-evolução.

O pseudo-código mostrado na Figura 5.46 esclarece o algoritmo utilizado.

Foram realizadas várias análises envolvendo variações em relação à parametrização do GSR, adotando-se valores de 100, 500 e 1000 para o limite máximo do número de iterações e tolerâncias constantes e variáveis durante o processo de evolução. Foi adotado $\varepsilon = 10^{-8}$ como valor constante durante todo processo e também os seguintes valores iniciais e finais: $\varepsilon_i = 10^{-3}$ e $\varepsilon_f = 10^{-7}$, $\varepsilon_i = 10^{-3}$ e $\varepsilon_f = 10^{-8}$ e finalmente $\varepsilon_i = 10^{-3}$ e $\varepsilon_f = 10^{-9}$. Para o caso de variação de ε durante as avaliações, seu valor é dado pela expressão:

$$\varepsilon = \varepsilon_i \left(\frac{\varepsilon_f}{\varepsilon_i} \right)^{\frac{naval}{maxaval}}$$

onde *naval* é o número da avaliação corrente e *maxaval* é o número máximo de avaliações.

Os operadores de recombinação escolhidos para atuarem sobre a população de estruturas foram os *crossovers* de dois pontos, discreto e heurístico e adotou-se uma mutação randômica, com probabilidades constantes mostradas na Tabela 5.84. Para a população de ω adotou-se somente o operador de recombinação BLX- α ($\alpha = 0.3$).

operadores	cr. 2 pontos	cr. discreto	cr. heurístico	mut. randômica
probabilidades	0.2	0.3	0.4	0.1

Tabela 5.84: Probabilidades dos operadores para a população de estruturas.

A Tabela 5.85 mostra os valores finais da evolução de ω , o número total de iterações (NT_{it}) e o peso obtido na otimização estrutural, com variações de parametrizações do GSR, onde as soluções são as melhores obtidas em 10 execuções

Algoritmo**Início**

Inicializar população de estruturas (pop_A)

Inicializar população de ω (pop_B)

para $i = 1$ **até** pop_B **faça**

$ii = 1$

$jj = pop_A / pop_B$

para $j = ii$ **até** jj **faça**

 resolver estruturas com $\omega(i)$

 armazenar número de iterações para cada estrutura $NIT(i)$

 calcular aptidão para cada estrutura

fim

 aptidão para $\omega(i) = \sum_{k=ii}^{jj} NIT(i) / jj - ii + 1$

$ii = ii + pop_A / pop_B$

$jj = jj + pop_A / pop_B$

fim

Repita

 selecionar pais para estruturas

 aplicar operadores genéticos para estrutura

 selecionar pais para ω

 aplicar operadores genéticos para ω

 avaliar nova estrutura com novo ω

 inserir ou não nova estrutura em pop_A

 inserir ou não novo ω em pop_B

Até critério de parada satisfeito

Fim

Figura 5.46: O algoritmo para o processo de co-evolução.

independentes. A Tabela 5.86 mostra os valores médios obtidos em 10 execuções incluindo o número médio de iterações (N_{it-med}), por resolução de sistema, ilustrando o comportamento das soluções.

Percebe-se na Tabela 5.85 que o menor peso obtido na otimização estrutural é de 5061.26015 lbs com $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 500. Para esse caso o valor de ω adaptado foi de 1.458347 com um total de 757260 iterações.

A Tabela 5.87 mostra resultados referentes ao número total de iterações considerando-se valores de ω constantes durante toda evolução para a melhor solução em 10 execuções e a Tabela 5.88 para a média em 10 execuções. Para o GSR adotou-se 500 como limite de iterações, $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$, que é a parametrização da melhor solução da otimização estrutural.

Observa-se que o valor evoluído final não é igual ao melhor ω fixo.

Nota-se que, por esses resultados, que uma boa escolha para ω constante neste problema deve estar no intervalo [1.1, 1.5] que apresenta uma região com menores números totais de iterações e o valor evoluído para $\omega = 1.458347$, escolhido a partir da Tabela 5.85, está dentro desse intervalo. Observa-se também que o valor evoluído final não é igual ao melhor ω fixo.

ε_i	ε_f	<i>limite</i>	ω	NT_{it}	peso(lb)
10^{-8}	10^{-8}	100	1.533026	961238	5062.657
10^{-3}	10^{-7}	100	1.520803	599782	5062.581
10^{-3}	10^{-8}	100	1.508408	649928	5062.304
10^{-3}	10^{-9}	100	1.484235	686843	5061.315
10^{-8}	10^{-8}	500	1.488525	970356	5061.633
10^{-3}	10^{-7}	500	1.404079	622511	5064.410
10^{-3}	10^{-8}	500	1.458347	757260	5061.260
10^{-3}	10^{-9}	500	1.501123	816062	5061.799
10^{-8}	10^{-8}	1000	1.366672	1097310	5062.695
10^{-3}	10^{-7}	1000	1.559401	829078	5061.627
10^{-3}	10^{-8}	1000	1.515435	890394	5061.840
10^{-3}	10^{-9}	1000	1.553076	1012490	5062.621

Tabela 5.85: Valores das melhores soluções em 10 execuções independentes.

ε_i	ε_f	<i>limite</i>	ω	N_{it-med}	NT_{it}	peso(lb)
10^{-8}	10^{-8}	100	1.438454	23.6	838978.4	5096.065
10^{-3}	10^{-7}	100	1.468690	10.0	584534.7	5094.381
10^{-3}	10^{-8}	100	1.482720	10.7	664216.2	5093.527
10^{-3}	10^{-9}	100	1.464320	10.3	690099.2	5115.645
10^{-8}	10^{-8}	500	1.465354	24.7	973621.6	5095.493
10^{-3}	10^{-7}	500	1.444232	9.81	674876.2	5076.056
10^{-3}	10^{-8}	500	1.481407	10.8	748896.4	5114.479
10^{-3}	10^{-9}	500	1.487602	11.0	808702.7	5073.859
10^{-8}	10^{-8}	1000	1.465311	24.4	1060460.5	5096.331
10^{-3}	10^{-7}	1000	1.530734	10.8	819560.3	5095.735
10^{-3}	10^{-8}	1000	1.506596	10.8	859157.9	5095.728
10^{-3}	10^{-9}	1000	1.498975	10.7	919288.2	5096.102

Tabela 5.86: Valores médios em 10 execuções.

A Figura 5.47 mostra a distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução. Nota-se na Figura 5.47 que existe uma grande concentração de valores de ω próximos a 1.5 mostrando uma tendência de evolução para este valor. A Figura 5.48 mostra a população de ω no processo de evolução. A Figura 5.49 mostra o melhor valor de ω na população durante a evolução, observando-se que ele ocupou a primeira posição do *rank* desde o seu surgimento no início do processo evolutivo. A Figura 5.50 mostra a aptidão do melhor ω ao longo da evolução, que é constante, visto que não ocorreram mudanças na primeira posição do *rank*. A Figura 5.51 mostra a variação do peso da estrutura ao longo do processo de evolução para a melhor execução.

De posse da estrutura otimizada, o sistema de equações gerado por ela foi resolvido pelo GSR com valores de ω variando no intervalo $[1.0, 1.95]$, que é o espaço de busca para a população de ω , e o gráfico ($\omega \times$ iterações) é mostrado na Figura

ω	NT_{it}
1.0	1037061
1.1	775593
1.2	693538
1.3	647471
1.4	672349
1.458347	698885
1.5	776887
1.6	879448
1.7	1237899
1.8	1799026
1.9	3505677
1.95	7164054

Tabela 5.87: Valores constantes de ω e o correspondente total de iterações para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 500.

ω	NT_{it}	N_{it-med}	peso (lb)
1.0	1493881.0	38.7	5073.984
1.1	1155798.3	27.1	5095.713
1.2	974634.7	20.2	5092.599
1.3	883039.8	18.5	5092.719
1.4	898228.9	21.2	5075.803
1.487602	984101.3	25.6	5096.625
1.5	1002034.9	26.0	5073.995
1.6	1228956.4	34.4	5096.008
1.7	1652285.6	48.1	5096.354
1.8	2523666.5	76.9	5095.472
1.9	4985611.9	159.8	5095.892
1.95	9602680.7	321.7	5098.963

Tabela 5.88: Valores constantes de ω , número médio de iterações por resolução de sistema e número total de iterações médio em 10 execuções para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-9}$ e limite = 500.

5.52. Observa-se nesse gráfico que o valor de ω que proporciona o menor número de iterações está entre 1.2 e 1.4 (1.291175 com 31 iterações), afastado, portanto, do valor 1.458347 encontrado na evolução.

Pelo aspecto do gráfico mostrado na Figura 5.49, nota-se que o melhor ω (melhor indivíduo) manteve-se constante ocupando a primeira posição do *rank* durante todo o processo de evolução e sua aptidão é mostrada na Figura 5.50. A explicação para este fato é que o valor $\omega = 1.458347$ encontrou um sistema de equações onde foram necessárias somente 11 iterações para a obtenção da solução e isso ocorreu no início do processo de evolução, na avaliação de número 120 da população de estruturas, e nenhum outro valor de ω gerado foi capaz de resolver um sistema com um número

de iterações menor. Além, disso, percebe-se através da distribuição da população de ω mostrada na Figura 5.48 que houve perda de diversidade rapidamente e os indivíduos dessa população ficaram concentrados entre 1.45 e 1.60 aproximadamente, e conseqüentemente valores na vizinhança de 1.3 não mais fizeram parte da população.

O que pode-se concluir pelos resultados das análises desse processo de co-evolução é que para a população de estruturas não são observadas interferências na evolução em função de valores de ω .

O cenário de aptidão é a representação da função aptidão através de todos as suas variáveis envolvidas no processo de busca. Entretanto, o cenário de aptidão da população de ω depende das matrizes geradas pela população de estruturas e estas variam durante a evolução. Dessa forma, observa-se que o cenário de aptidão da população de ω é variável (“cenário de aptidão móvel”), ou seja, para cada sistema de equações existirá um ω ótimo que proporciona um menor número de iterações no GSR. Dessa forma, observa-se que o valor de ω ótimo é diferente para cada estrutura, principalmente, no princípio da evolução onde é maior a diversidade na população de estruturas.

Isto introduz dificuldades na evolução do ω . Além disso, a função aptidão medida unicamente pelo número de iterações (função aptidão, pode-se dizer, “ruidosa”), fornece poucas informações do cenário de aptidão não indicando quais as variações necessárias para “acompanhar” as mudanças do cenário. Outra dificuldade presente refere-se ao tipo de algoritmo genético utilizado. No tipo “steady-state” não há reposição de elementos, pais convivem com filhos e não existe elitismo. Pode ocorrer, então, pelo fato de não haver reposições, a permanência de indivíduos na população ao longo da evolução, que não são removidos por não serem testados. Estes podem ocupar boas posições no *rank* devido aos sucessos alcançados em períodos anteriores da evolução e não serem superados pelos descendentes.

Baseado nisso, observou-se que um certo indivíduo da população obteve um desempenho extremamente favorável quando foi submetido a um determinado sistema de equações gerado por uma estrutura que não representou uma “boa solução” na população de estruturas e, conseqüentemente, teve pouco material genético herdado pelos seus descendentes. Pelo fato da população de ω ser pequena, em relação à população de estruturas, ocorreu uma convergência prematura e perda de diversidade e assim indivíduos da população de ω que seriam interessantes ao GSR não mais fizeram parte da população.

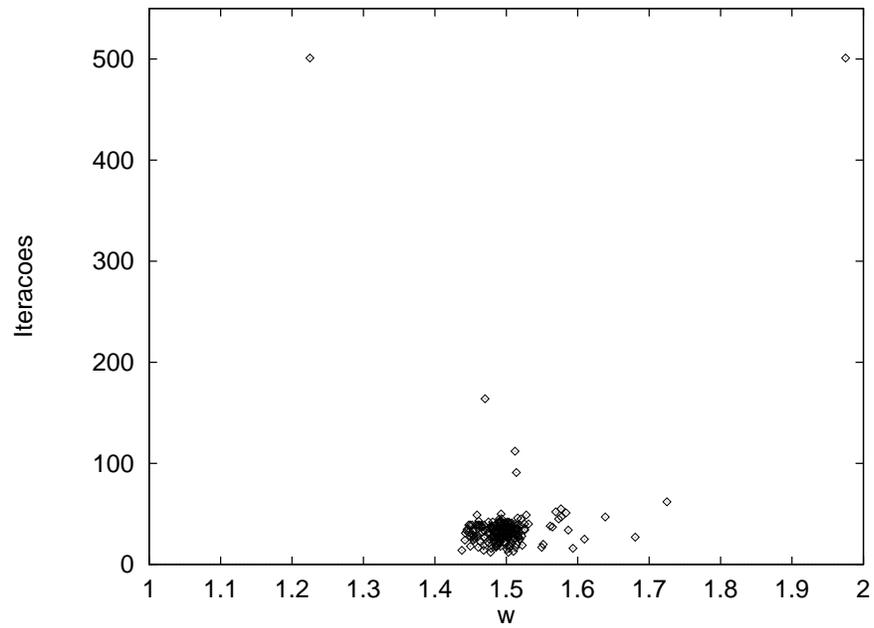


Figura 5.47: Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução.

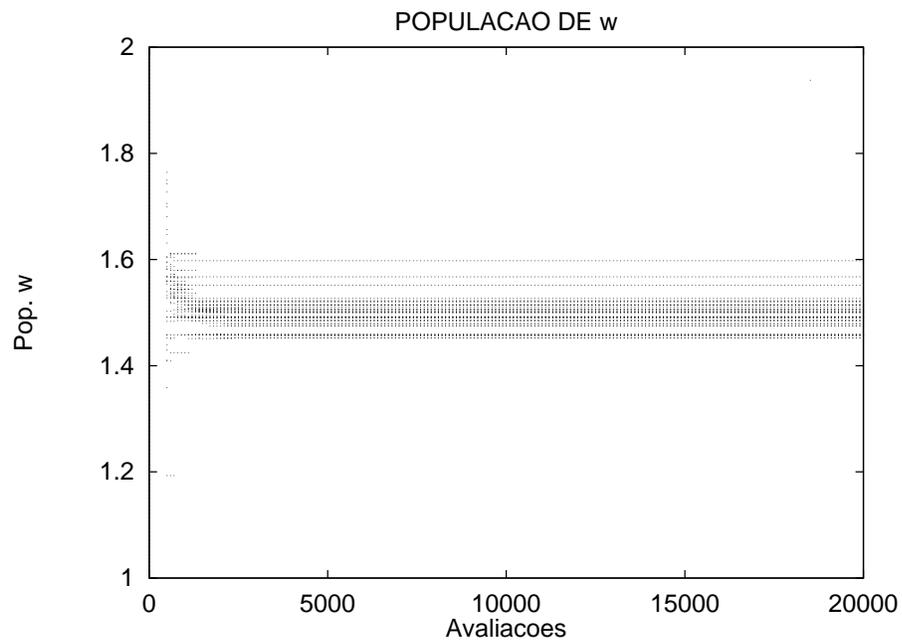


Figura 5.48: População de ω no processo de evolução.

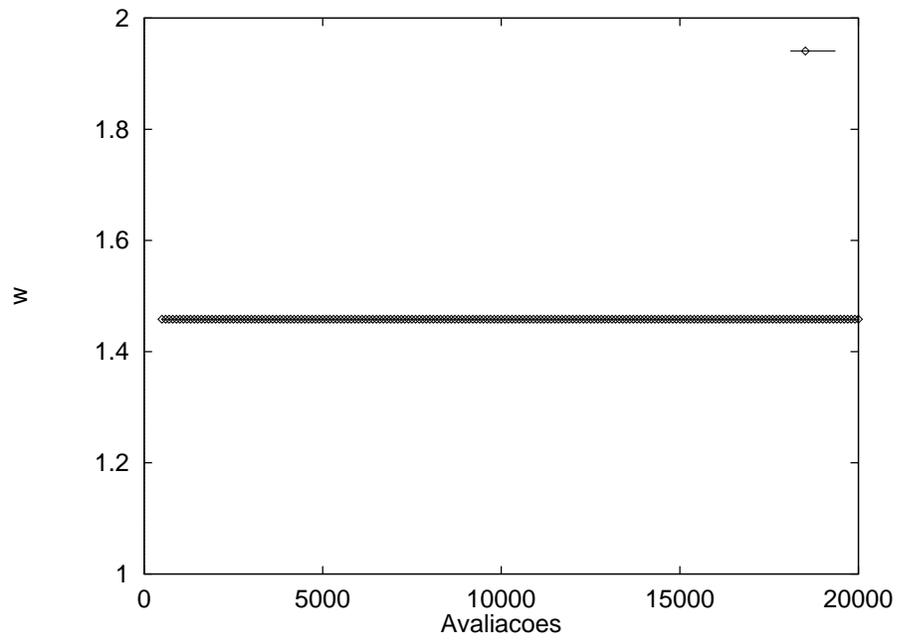


Figura 5.49: Melhor ω da população durante a evolução.

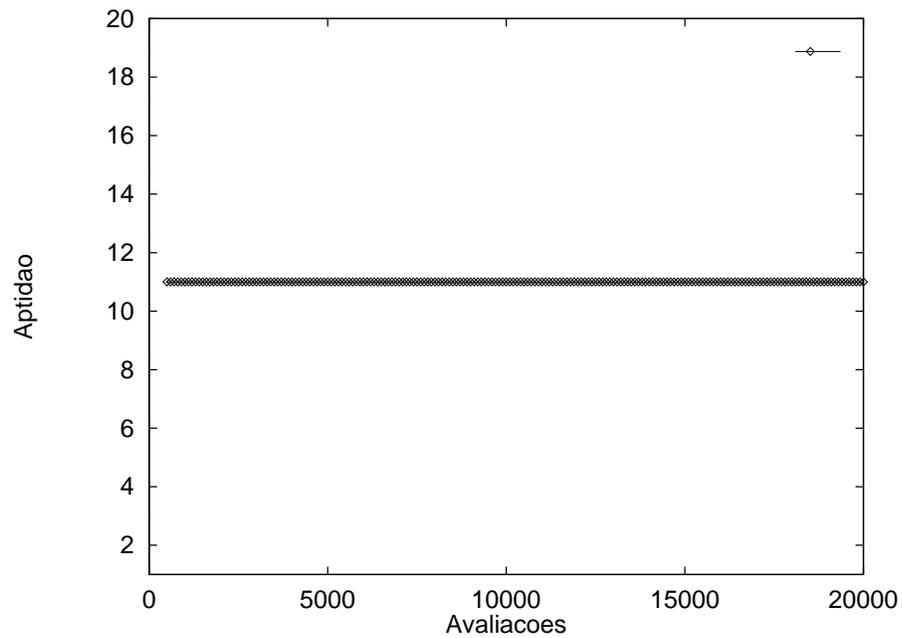


Figura 5.50: Aptidão do melhor ω ao longo da evolução.

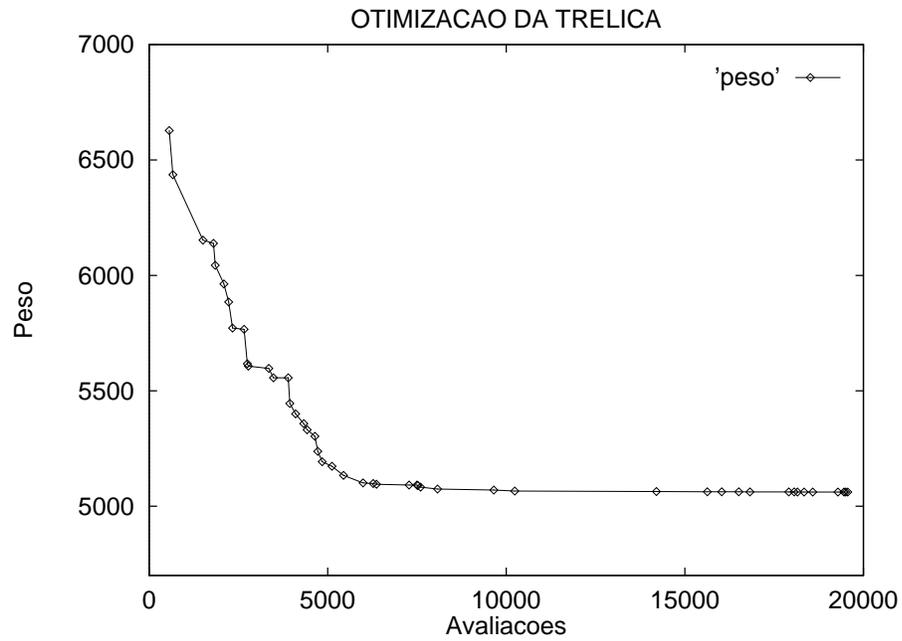


Figura 5.51: Peso da treliça de 10 barras.

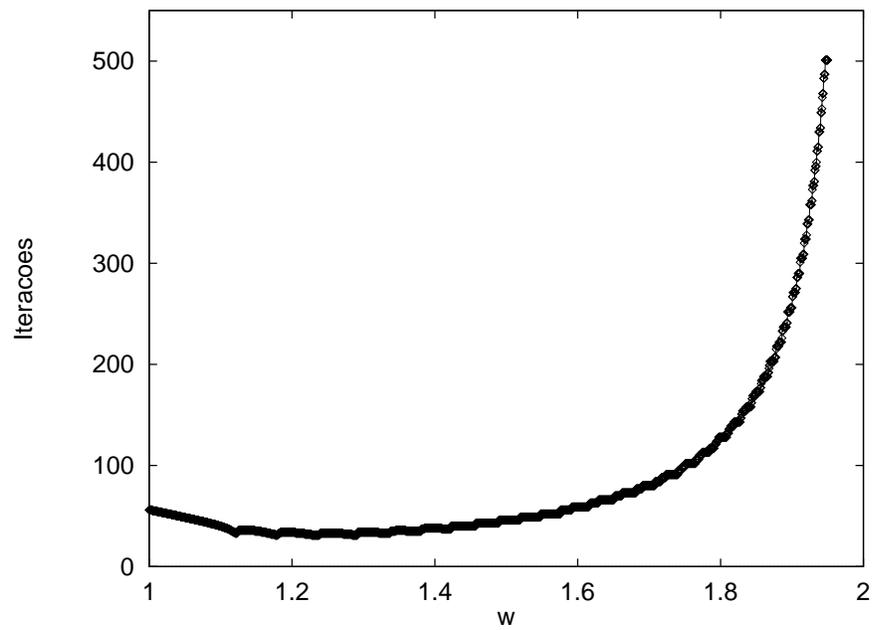


Figura 5.52: Número de iterações no GSR para $\omega \in [1.0, 1.95]$ referente ao sistema de equações da estrutura final otimizada.

5.8.2 Uma segunda estratégia

Percebeu-se pelas análises anteriores que a função aptidão para a população de ω é um pouco “ruidosa” (“cenário de aptidão móvel”).

Este fato motivou o desenvolvimento de uma segunda estratégia que é uma variação da anterior com alteração na política de inserção dos elementos da população de ω . Agora a inserção do filho ocorrerá no lugar do pai mais próximo, independentemente de sua aptidão. Imaginando-se que o pai tenha conseguido um bom lugar na ordenação e com isso é constantemente escolhido para reprodução ele será substituído pelo seu filho que poderá apresentar uma aptidão bastante diferente da sua em função do sistema de equações a ser resolvido.

A Tabela 5.89 mostra os valores de ω , o número total de iterações (NT_{it}) e o peso obtido na otimização estrutural, com variações de parametrizações do GSR, onde as soluções são as melhores obtidas em 10 execuções independentes. A Tabela 5.90 mostra os valores médios obtidos em 10 execuções incluindo o número médio de iterações (N_{it-med}), por resolução de sistema, ilustrando o comportamento das soluções.

ε_i	ε_f	limite	ω	NT_{it}	peso(lb)
10^{-8}	10^{-8}	100	1.476295	871769	5066.668
10^{-3}	10^{-7}	100	1.506007	614548	5076.924
10^{-3}	10^{-8}	100	1.502171	661074	5061.267
10^{-3}	10^{-9}	100	1.500294	730577	5061.338
10^{-8}	10^{-8}	500	1.520880	1050249	5063.234
10^{-3}	10^{-7}	500	1.505504	708595	5062.635
10^{-3}	10^{-8}	500	1.492474	889021	5062.319
10^{-3}	10^{-9}	500	1.483916	780114	5061.789
10^{-8}	10^{-8}	1000	1.438168	1064796	5061.376
10^{-3}	10^{-7}	1000	1.512096	805405	5062.030
10^{-3}	10^{-8}	1000	1.497292	853488	5061.158
10^{-3}	10^{-9}	1000	1.518748	910639	5063.978

Tabela 5.89: Segunda estratégia – Valores das melhores soluções em 10 execuções independentes.

Percebe-se na Tabela 5.89 que o menor peso obtido na otimização estrutural é de 5061.15838 lbs com $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 1000. Para esse caso o valor final evoluído para ω é 1.497292 e o número total de iterações é 853488.

A Figura 5.53 mostra a distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução. Percebe-se através da Figura 5.53 que, diferentemente, da primeira análise, existe uma melhor distribuição de valores de ω e uma concentração menor próxima do valor 1.5. A Figura 5.54 mostra a população de ω que mantém alguma diversidade até 5000 avaliações e depois converge rapidamente para valores próximos de 1.5.

Da mesma forma como feito na primeira análise, de posse da estrutura otimizada, o sistema de equações gerado pela estrutura final foi resolvido pelo GSR com valores

ε_i	ε_f	limite	ω	N_{it-med}	NT_{it}	peso(lb)
10^{-8}	10^{-8}	100	1.485735	48.4	907023.8	5097.956
10^{-3}	10^{-7}	100	1.504217	40.7	614820.4	5099.472
10^{-3}	10^{-8}	100	1.491010	45.6	661665.9	5091.881
10^{-3}	10^{-9}	100	1.476322	50.2	696562.5	5125.494
10^{-8}	10^{-8}	500	1.491988	50.2	1015270.6	5138.521
10^{-3}	10^{-7}	500	1.493955	41.6	706970.1	5075.644
10^{-3}	10^{-8}	500	1.477212	46.9	740122.3	5074.124
10^{-3}	10^{-9}	500	1.488521	52.3	806330.8	5075.554
10^{-8}	10^{-8}	1000	1.493504	50.2	1158285.3	5073.185
10^{-3}	10^{-7}	1000	1.489517	41.6	789525.3	5072.046
10^{-3}	10^{-8}	1000	1.491051	46.9	850137.1	5116.579
10^{-3}	10^{-9}	1000	1.499473	52.3	916041.1	5096.447

Tabela 5.90: Segunda estratégia – Valores médios de 10 execuções.

ω	NM_{it}
1.0	1060007
1.1	853440
1.2	770971
1.3	770774
1.4	757689
1.497292	853086
1.5	835332
1.6	996815
1.7	1272734
1.8	1872621
1.9	3622662
1.95	7334614

Tabela 5.91: Segunda estratégia – Valores constantes de ω e número total de iterações para $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-8}$ e limite de iterações igual a 1000.

de ω variando no intervalo $[1.0, 1.95]$ e o gráfico ($\omega \times$ iterações) é mostrado na Figura 5.58. Observa-se nesse gráfico, também, que o valor de ω que proporciona o menor número de iterações está entre 1.2 e 1.4 (1.238925 com 30 iterações), não confirmando o valor 1.497292 encontrado na evolução.

Diferentemente do ocorrido inicialmente, pelo aspecto do gráfico mostrado na Figura 5.55, nota-se que houve variações no valor do melhor ω (melhor indivíduo) até, aproximadamente, as primeiras 20000 avaliações. O último valor de ω a ocupar o primeiro lugar no *rank* foi exatamente 1.497292 na avaliação de número 19878 da população de estruturas. O valor da aptidão, representada pelo número de iterações, para o melhor ω é mostrada na Figura 5.56. Nota-se, também, neste gráfico que no começo do processo de evolução valores de ω próximos de 1.25 proporcionaram número de iterações menores que no final, mas estes deixaram de ocupar as melhores

ω	NT_{it}	N_{it-med}	peso (lb)
1.0	1019257.0	146.4	5121.900
1.1	810420.5	120.1	5105.934
1.2	724030.4	97.3	5073.963
1.3	680898.9	77.4	5072.707
1.4	701946.9	58.4	5074.643
1.489517	766359.9	44.2	5073.771
1.5	779110.3	43.1	5093.416
1.6	914784.7	36.9	5100.661
1.7	1144242.3	46.6	5093.943
1.8	1688241.7	71.9	5076.587
1.9	3262969.5	148.6	5089.406
1.95	6486132.6	300.4	5122.601

Tabela 5.92: Segunda estratégia – Valores constantes de ω , número médio de iterações e número total de iterações médio – Valores médios em 10 execuções – $\varepsilon_i = 10^{-3}$, $\varepsilon_f = 10^{-7}$ e limite = 1000.

posições no *rank* justamente pelo fato de terem sido substituídos pelos seus filhos, exatamente como é a proposta da segunda estratégia. A Figura 5.57 mostra a variação do peso da estrutura ao longo da evolução da melhor execução.

Também, neste caso, ocorreu perda rápida de diversidade e valores na vizinhança de 1.25 deixaram de fazer parte da população de ω , que seriam os ideais para o final do processo de otimização para a população de estruturas.

Entretanto, ao final da evolução notou-se também para a segunda estratégia que o número máximo de iterações gerado com a adaptação de ω está dentro do intervalo de valores aceitáveis para ω como mostra a Tabela 5.91, obtidos nas melhores execuções e a Tabela 5.92 para a média em 10 execuções. Esta observação mostra que o procedimento de co-evolução é eficiente.

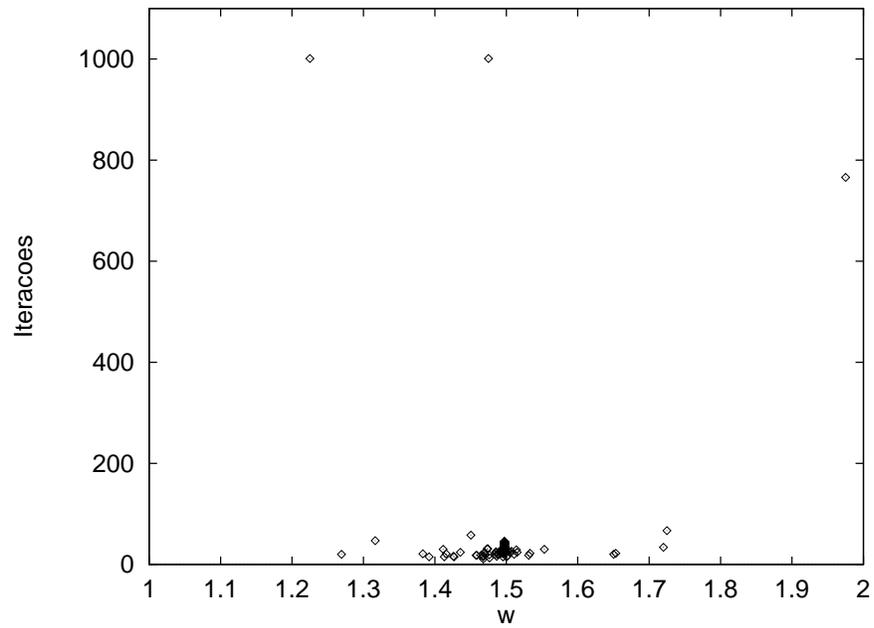


Figura 5.53: Segunda estratégia – Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução.

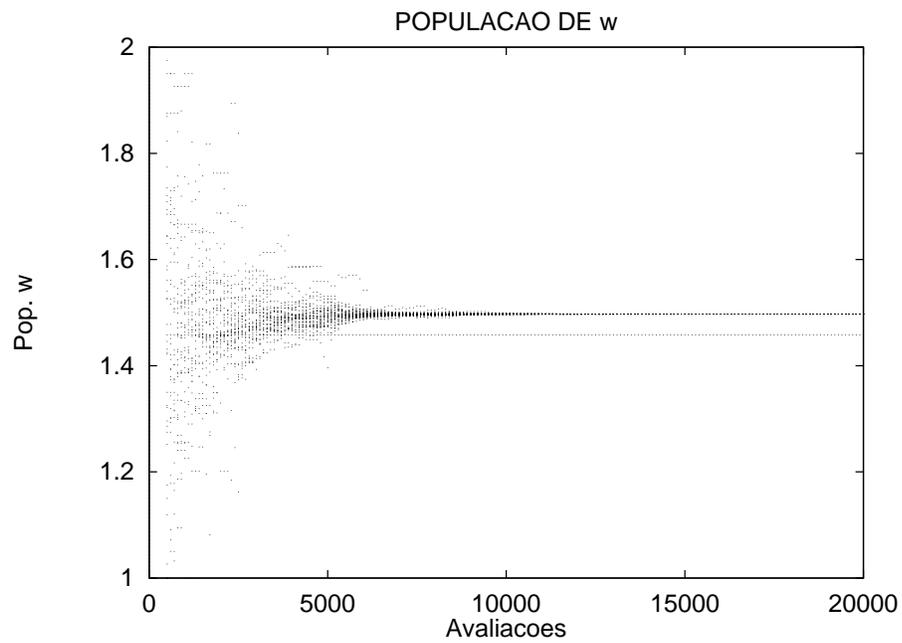


Figura 5.54: Segunda estratégia – População de ω no processo de evolução.

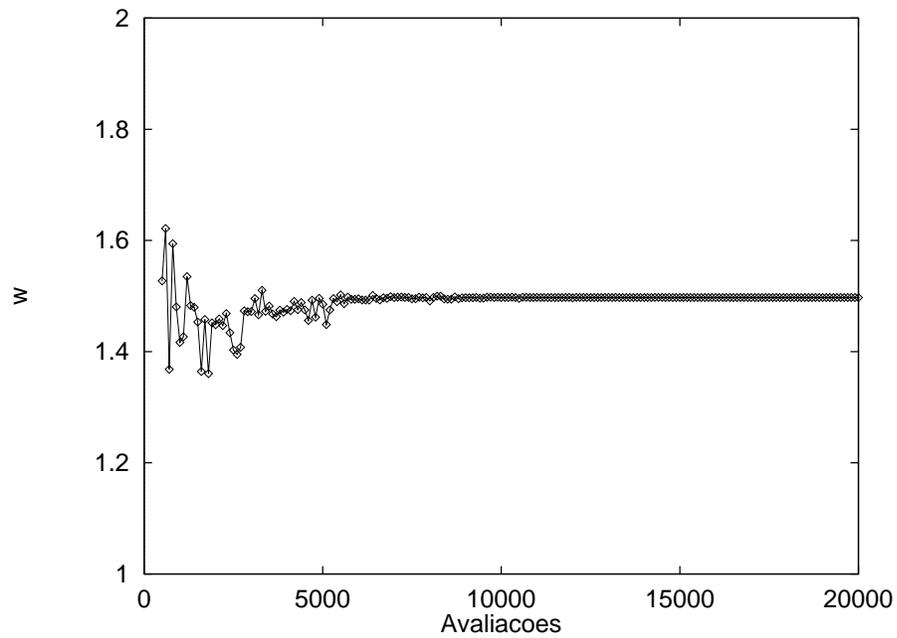


Figura 5.55: Segunda estratégia – Melhor ω da população durante a evolução.

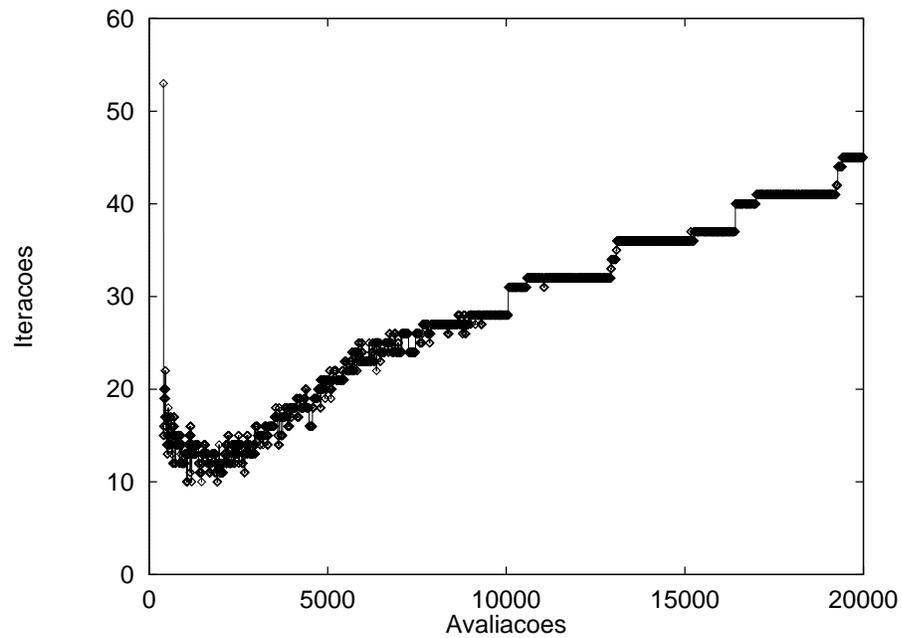


Figura 5.56: Segunda estratégia – Aptidão do melhor ω ao longo da evolução.

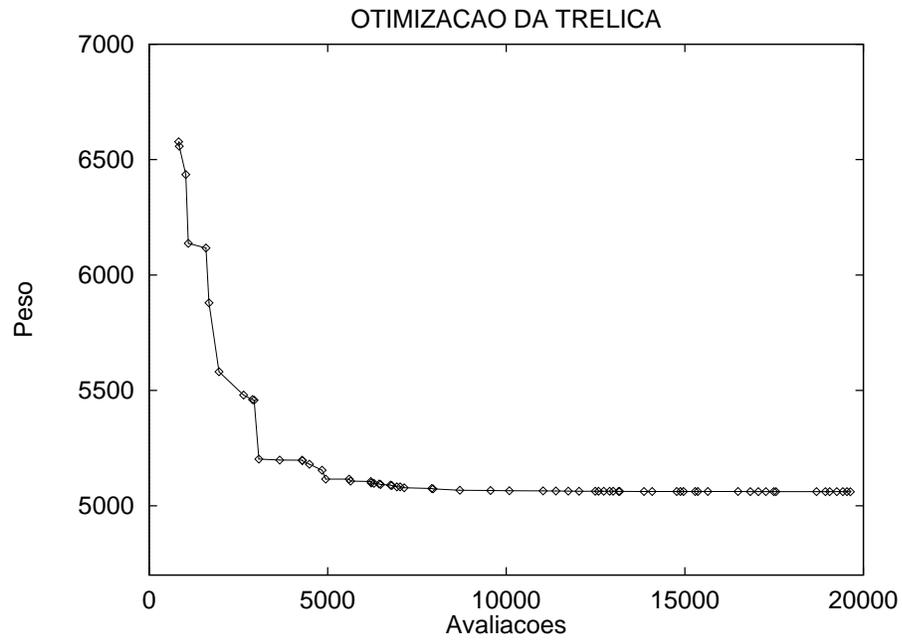


Figura 5.57: Segunda estratégia – Peso da treliça de 10 barras.

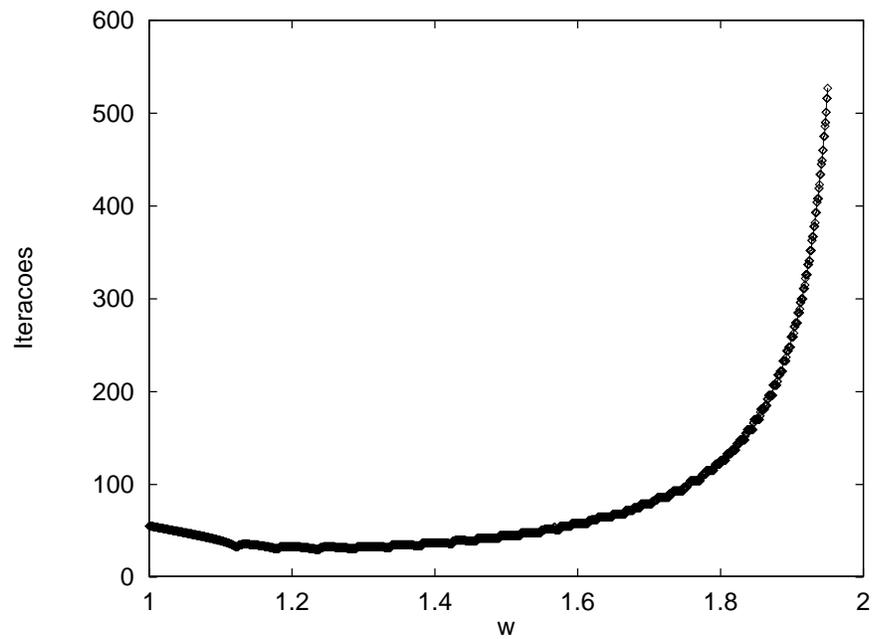


Figura 5.58: Segunda estratégia – Sistema de equações da estrutura final otimizada resolvido com vários valores de ω .

5.8.3 Uma terceira estratégia

Pelas análises das estratégias anteriores decidiu-se, ainda, pela implementação de uma terceira estratégia que tem como objetivo principal o controle da diversidade da população de ω , visto que este é o aspecto de maior sensibilidade dessas análises. Para tanto foram implementadas alterações nas operações de seleção e recombinação e também na determinação da função aptidão. A política de inserção dos novos filhos se dá exatamente como implementado na primeira análise, ou seja, se houver inserção esta ocorrerá no lugar do indivíduo de pior aptidão.

A terceira estratégia introduz modificações no processo de seleção dos pais, no processo de geração dos novos filhos e na obtenção da aptidão da população de ω . A intenção é preservar a diversidade da população de ω .

A estratégia pode ser descrita como: no processo de seleção existe a possibilidade de haver novo filho ou não e, se não houver novo filho a seleção se dará sobre um indivíduo já pertencente à população e este terá sua aptidão determinada diferentemente das estratégias anteriores. Agora, a aptidão será dada pelo valor médio de todas as vezes em que o indivíduo foi avaliado (média das iterações de todas as vezes em que foi utilizado). Se houver novo filho este poderá ser procedente de uma operação de recombinação ou mutação e será avaliado e inserido (ou não) na população.

O pseudo-código apresentado na Figura 5.59 mostra as alterações no algoritmo referentes à geração ou não de novos filhos:

Início

```

selecionar pais  $\omega_1$  e  $\omega_2$ 
com probabilidade  $p_{new}$  faça  $\omega_{novo} = verdadeiro$ 
se  $\omega_{novo}$  então:
    escolher operador de recombinação ou mutação
    gerar novo  $\omega$ 
    utilizar novo  $\omega$  em GSR
senão
    utilizar  $\omega_1$  em GSR
fim-se
atualizar aptidão do  $\omega$  utilizado

```

Fim

Figura 5.59: Algoritmo que controla a geração de novos filhos – Terceira estratégia.

O pseudo-código apresentado na Figura 5.60 mostra as alterações no algoritmo referentes à determinação da função aptidão e política de inserção.

A Tabela 5.93 mostra resultados de ω , das melhores execuções, obtidos através de variações nas parametrizações referente à geração de novos filhos onde p_{new} representa a probabilidade de haver novos filhos e p_{mut} representa a probabilidade do mesmo surgir por uma operação de mutação. Neste caso foi adotado o valor de 500 como limite máximo de iterações no GSR e tolerância constante $\varepsilon = 10^{-8}$. A Tabela 5.94 mostra valores médios de ω em 10 execuções independentes, variando com p_{new}

```

Início
  avaliar estrutura com  $\omega$ 
  obter aptidão de  $\omega$ :  $aptid\tilde{a}o(i)$ 
  se  $\omega$  não é novo faça
     $nit(i) = nit(i) + niter$ 
     $nvezes(i) = nvezes(i) + 1$ 
     $aptid\tilde{a}o(i) = nit(i)/nvezes(i)$ 
  senão
     $nit(i) = niter$ 
     $nvezes(i) = 1$ 
     $aptid\tilde{a}o(i) = niter$ 
  inserir (ou não)  $\omega$  em  $pop_B$ 
fim-se
  atualizar  $rank$  de  $pop_B$ 
Fim

```

Figura 5.60: Algoritmo para a determinação da função aptidão e controle da política de inserção – Terceira estratégia.

e p_{mut} . Para essa estratégia adotou-se também o operador BLX- α ($\alpha = 0.3$), para operações de recombinação, e uma mutação ($mutaw$) que acrescenta ou diminui 0.15 no valor do ω selecionado.

Pelos resultados apresentados na Tabela 5.93, nota-se que com probabilidades pequenas de serem gerados novos filhos p_{new} , os valores de ω encontrados são bastante próximos daqueles obtidos nas estratégias anteriores, que indicam os menores números de iterações na estrutura final otimizada. Nas estratégias anteriores os valores são 1.291175 com 31 iterações e 1.238925 com 30 iterações, respectivamente. A Tabela 5.95 confirma este fato onde são mostradas as iterações da melhor solução.

Na Tabela 5.96 encontram-se os valores do número de iterações do melhor indivíduo variando com p_{new} e p_{mut} , sendo valores médios em 10 execuções independentes.

Nas Tabelas 5.97 e 5.99 encontram-se o número total de iterações e o peso obtido na otimização da estrutura envolvida neste exemplo. Na Tabela 5.98 mostra-se o número total de iterações variando com p_{new} e p_{mut} e na Tabela 5.100 mostra-se o peso da estrutura otimizada (lbs) variando com p_{new} e p_{mut} , sendo para ambas Tabelas valores médios em 10 execuções.

A Figura 5.61 mostra a distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução para as probabilidades $p_{new} = 0.002$ e $p_{mut} = 1.0$. A Figura 5.61 mostra que com essas probabilidades mantém-se uma boa diversidade da população, não ocorrendo concentração de elementos próximos ao valor 1.5. A Figura 5.62 mostra a população de ω confirmando a distribuição da Figura 5.61. A Figura 5.63 mostra o melhor ω da população durante a evolução e a Figura 5.64 mostra a aptidão do melhor ω ao longo da evolução. Essas Figuras mostram que foi possível o controle da diversidade da população e do “acompanhamento” do espaço de busca com “cenário de aptidão móvel”, podendo-se verificar a eficiência dessa estratégia. A Figura 5.65 mostra a variação do peso da estrutura ao longo da

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	1.30	1.325	1.275	1.475	1.55	1.50	1.5	1.525
0.9	1.275	1.337573	1.338408	1.422799	1.436657	1.530568	1.521407	1.525051
0.8	1.275	1.337573	1.285794	1.464859	1.529061	1.55	1.494230	1.55
0.7	1.290474	1.275	1.275	1.459133	1.450008	1.525107	1.510131	1.517519
0.6	1.280064	1.275	1.275	1.475	1.484325	1.380541	1.547578	1.515534
0.5	1.325	1.283062	1.30	1.467545	1.543702	1.540297	1.457325	1.492263
0.4	1.325	1.179509	1.259926	1.475052	1.489477	1.509150	1.484389	1.520795
0.3	1.325	1.357171	1.237667	1.414987	1.583814	1.516655	1.484398	1.455021
0.2	1.325	1.281043	1.279649	1.463573	1.417597	1.506640	1.511899	1.485838
0.1	1.325	1.281382	1.221883	1.400323	1.496481	1.488280	1.519340	1.537612

Tabela 5.93: Terceira estratégia – Valores de ω variando com p_{new} e p_{mut} .

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	1.23	1.26	1.2525	1.4775	1.5175	1.5225	1.5375	1.51
0.9	1.226933	1.255077	1.270710	1.472186	1.505402	1.520818	1.530575	1.529538
0.8	1.242075	1.272811	1.255464	1.446317	1.524165	1.534794	1.515298	1.519715
0.7	1.250617	1.260928	1.255823	1.453810	1.489783	1.507714	1.510229	1.514107
0.6	1.255303	1.257301	1.280119	1.452911	1.485331	1.498180	1.521677	1.514190
0.5	1.258624	1.236118	1.256659	1.451535	1.476960	1.505098	1.503971	1.489594
0.4	1.275334	1.231950	1.259789	1.448358	1.483040	1.487582	1.491378	1.521245
0.3	1.261802	1.245423	1.241956	1.423657	1.499542	1.504534	1.500871	1.513782
0.2	1.260229	1.247757	1.265332	1.439163	1.470260	1.516816	1.496428	1.519247
0.1	1.261723	1.236775	1.270565	1.424204	1.497941	1.491999	1.513601	1.504036

Tabela 5.94: Terceira estratégia – Valores médios de ω variando com p_{new} e p_{mut} .

evolução.

A Figura 5.66 mostra a distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução para o caso $p_{new} = 0.5$ e $p_{mut} = 0.1$. A Figura 5.67 mostra a população de ω . A Figura 5.68 mostra o melhor ω da população durante a evolução e a Figura 5.69 mostra a aptidão do melhor ω ao longo da evolução e, finalmente, a Figura 5.70 mostra a variação do peso da estrutura durante a evolução. Pelas Figuras 5.66, 5.67, 5.68 e 5.69 confirma-se que aumentando-se as probabilidades de geração de novos filhos utilizando-se baixas taxas de mutação, ocorre perda de diversidade da população dificultando o “acompanhamento” de modificações do espaço de busca.

Todas as estratégias apresentadas anteriormente consideraram o vetor de partida do algoritmo de Gauss-Seidel com valor nulo. Mostra-se a seguir resultados de experimentos realizados considerando-se o vetor de partida como sendo a solução obtida na avaliação anterior.

A Tabela 5.101 mostra os valores médios de ω variando com p_{new} e p_{mut} . A Tabela 5.102 mostra o número de iterações do melhor elemento variando com p_{new}

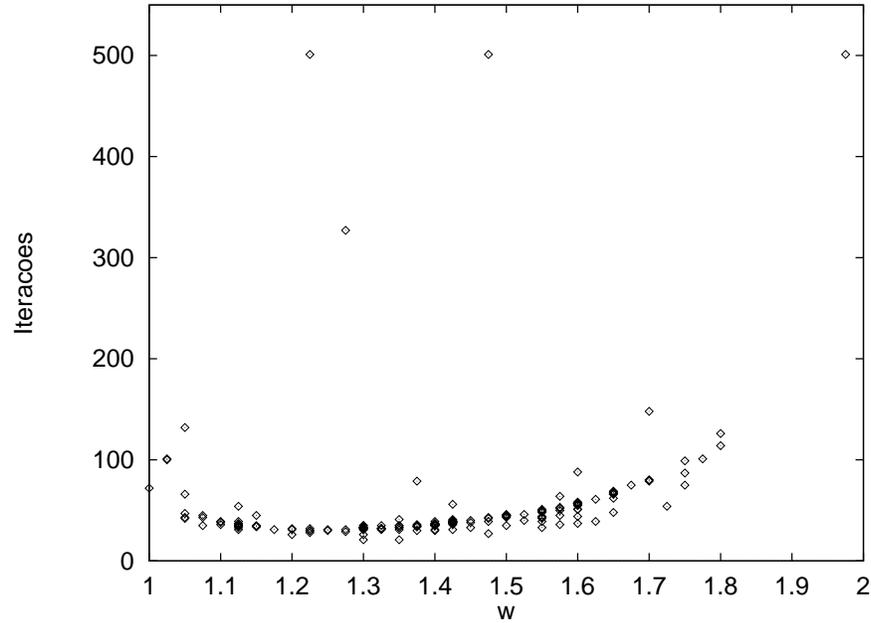


Figura 5.61: Terceira estratégia – Distribuição de valores de ω e seu respectivos números de iterações ao longo da evolução par $p_{new} = 0.002$ e $p_{mut} = 1.0$.

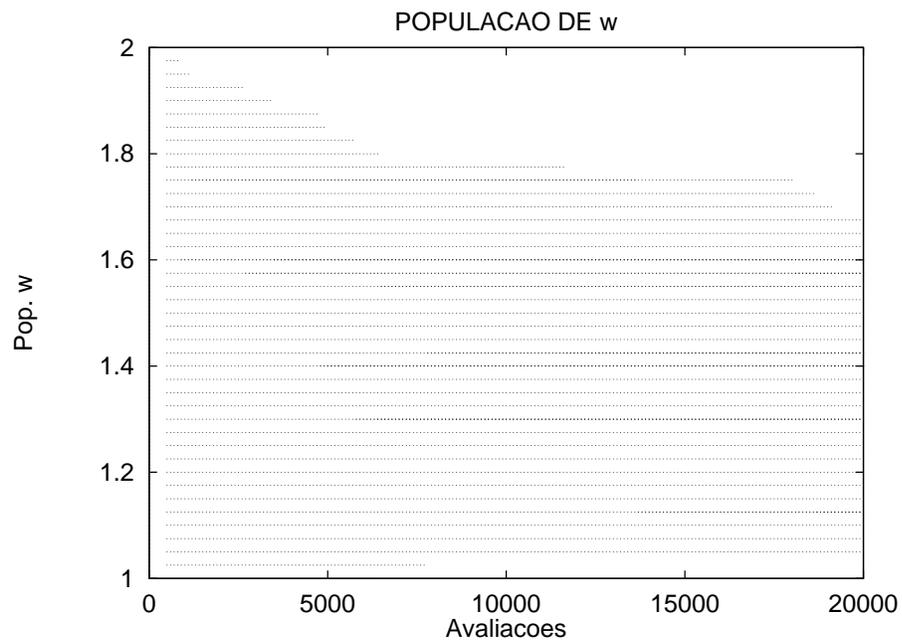


Figura 5.62: Terceira estratégia – População de ω no processo de evolução - $p_{new} = 0.002$ e $p_{mut} = 1.0$.

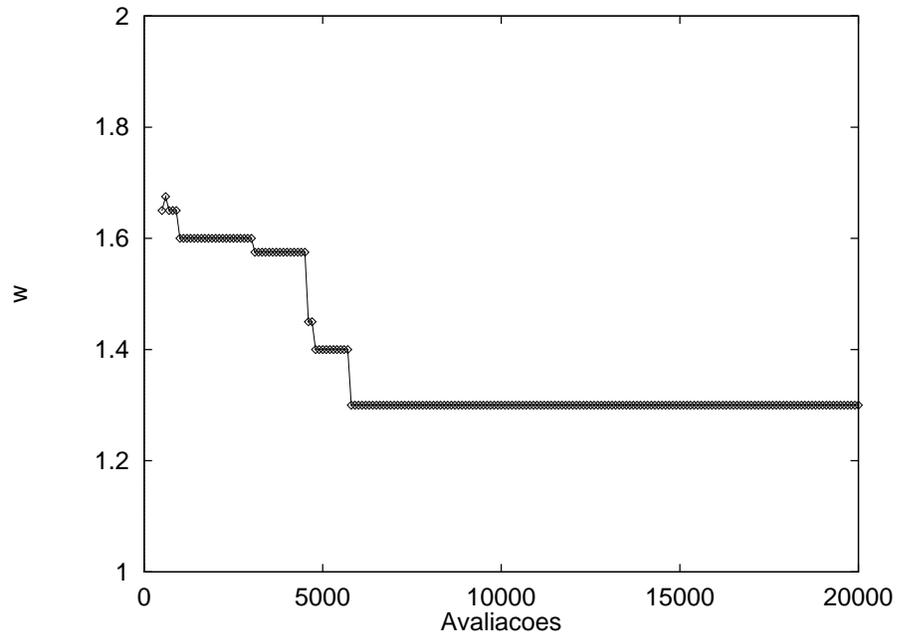


Figura 5.63: Terceira estratégia – Melhor ω da população durante a evolução – $p_{new} = 0.002$ e $p_{mut} = 1.0$.

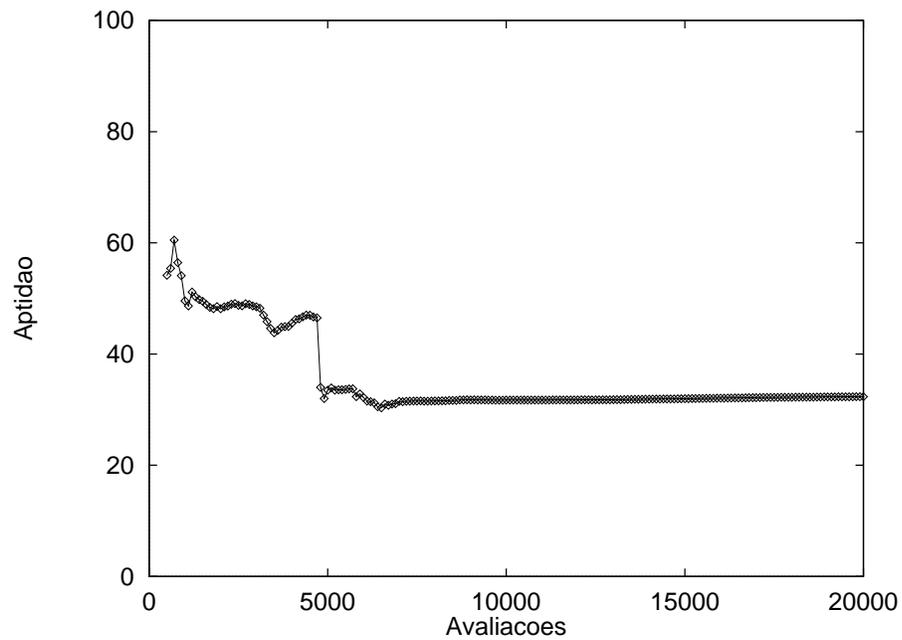


Figura 5.64: Terceira estratégia – Aptidão do melhor ω ao longo da evolução – $p_{new} = 0.002$ e $p_{mut} = 1.0$.

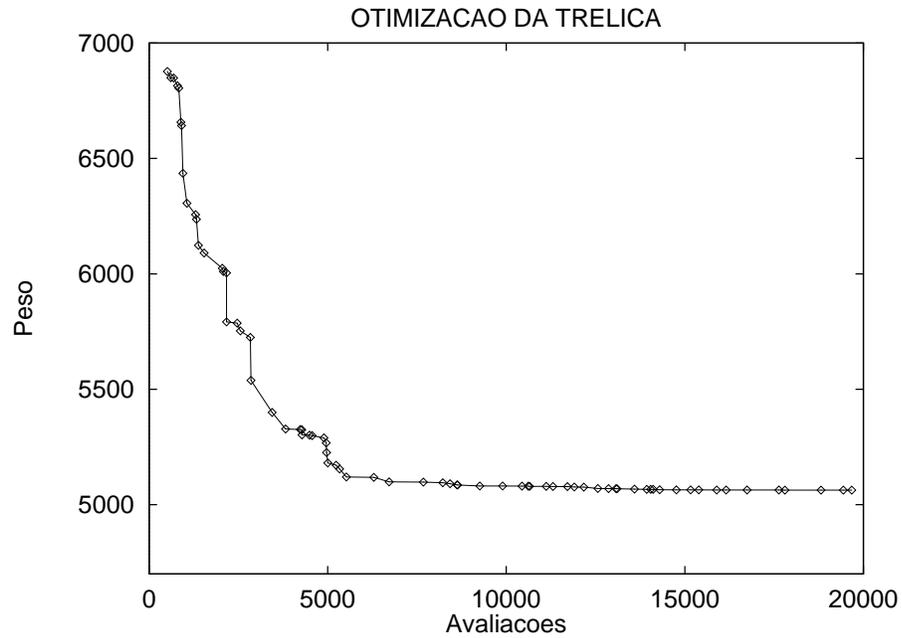


Figura 5.65: Terceira estratégia – Peso da treliça de 10 barras – $p_{new} = 0.002$ e $p_{mut} = 1.0$.

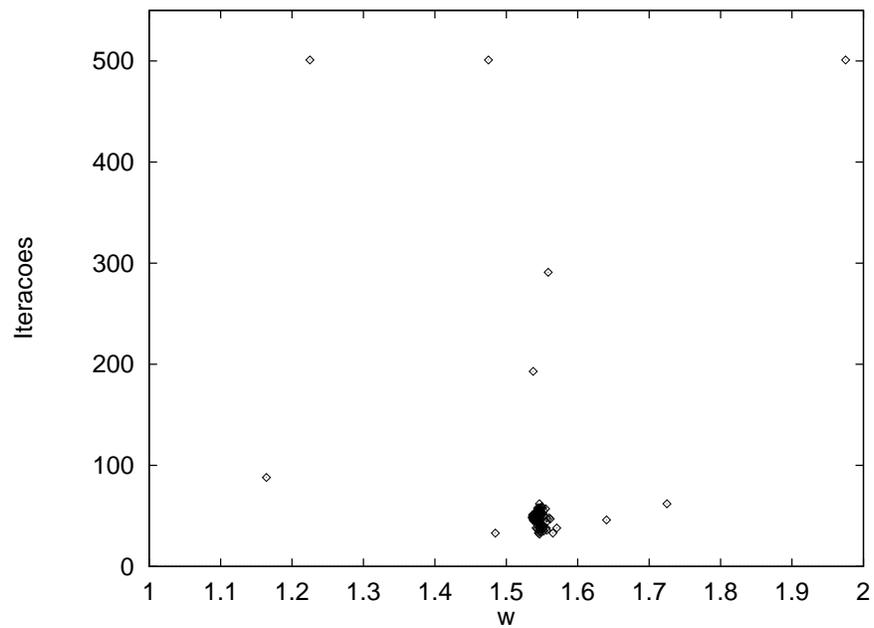


Figura 5.66: Terceira estratégia – Distribuição de valores de ω e seus respectivos números de iterações ao longo da evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$.

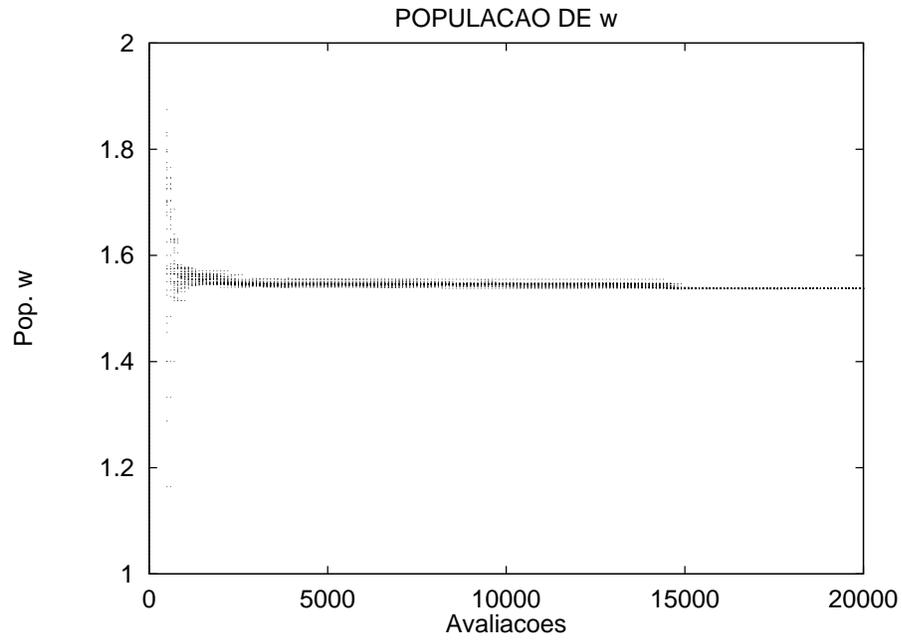


Figura 5.67: Terceira estratégia – População de ω no processo de evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$.

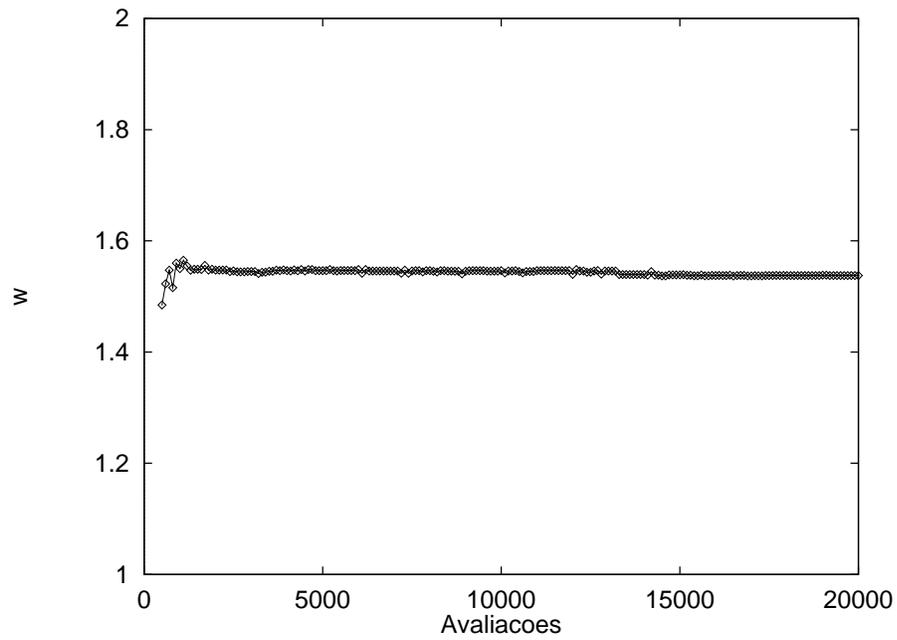


Figura 5.68: Terceira estratégia – Melhor ω da população durante a evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$.

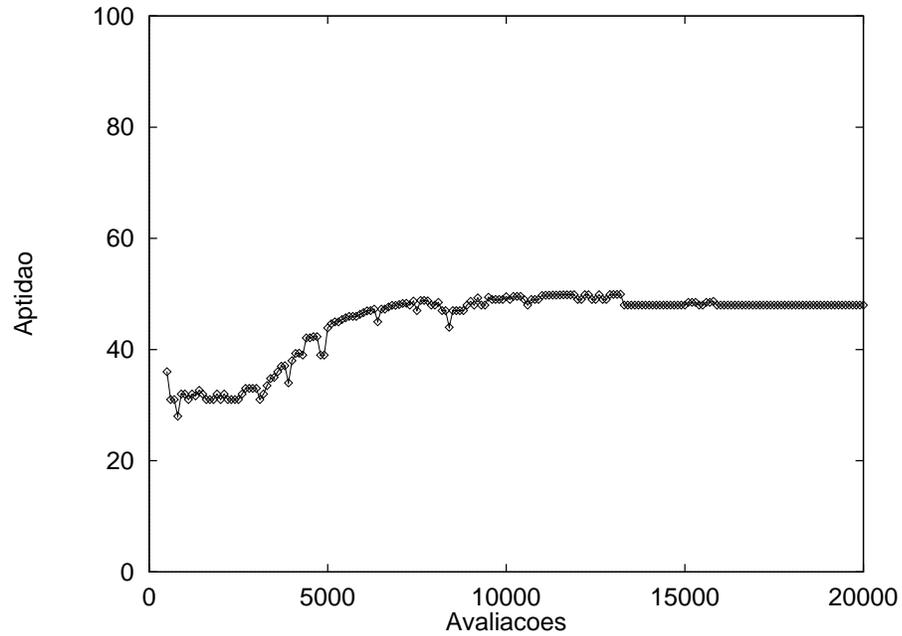


Figura 5.69: Terceira estratégia – Aptidão do melhor ω ao longo da evolução – $p_{new} = 0.5$ e $p_{mut} = 0.1$.

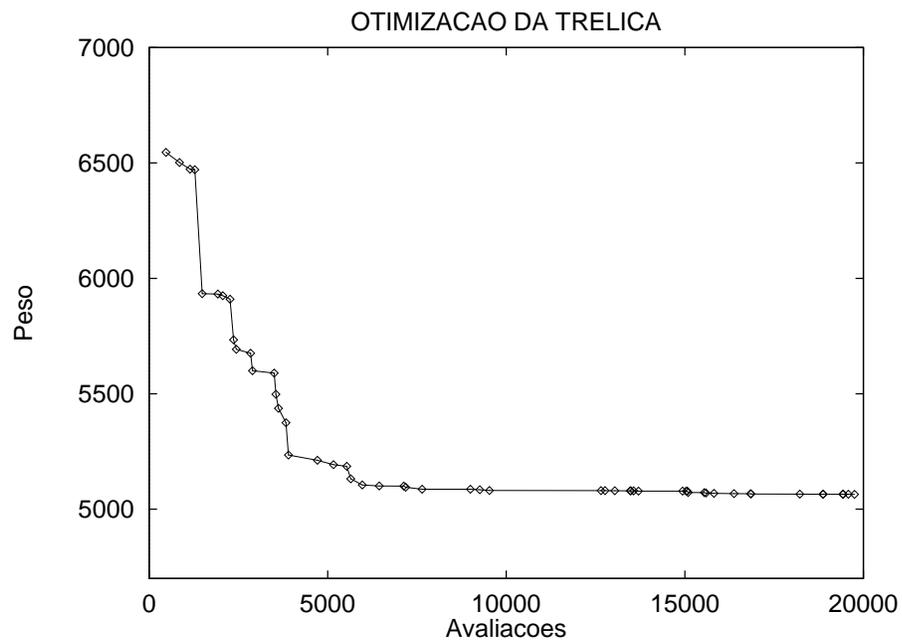


Figura 5.70: Terceira estratégia – Peso da treliça de 10 barras – $p_{new} = 0.5$ e $p_{mut} = 0.1$.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	32.329	32.775	31.970	41.175	48.703	43.611	43.247	47.218
0.9	32.686	32.625	32.0	37.60	39.503	45.750	45.0	45.0
0.8	32.686	33.0	31.0	40.380	44.808	49.0	46.611	48.0
0.7	31.281	31.759	31.922	40.0	39.878	45.633	45.154	44.168
0.6	34.537	31.759	31.696	39.968	40.106	35.0	48.0	44.028
0.5	33.054	31.576	33.231	40.133	47.0	46.356	39.0	41.667
0.4	33.054	31.857	33.031	40.039	41.788	43.823	42.897	44.5
0.3	33.054	34.802	31.474	37.983	54.0	45.0	41.0	38.0
0.2	33.054	31.695	32.0	40.0	36.0	44.674	45.0	41.667
0.1	33.054	31.0	31.633	36.358	42.568	41.80	44.135	48.0

Tabela 5.95: Terceira estratégia – Número de iterações da melhor solução variando com p_{new} e p_{mut} .

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	31.8	30.8	30.5	39.1	42.2	42.5	45.6	41.3
0.9	30.8	31.2	31.3	39.2	42.1	42.4	44.7	43.3
0.8	30.1	32.1	31.0	36.9	42.7	43.6	44.1	42.7
0.7	29.8	30.3	31.0	36.2	42.0	42.9	43.8	42.4
0.6	29.4	30.2	31.1	36.4	39.3	41.2	45.1	41.9
0.5	29.2	30.0	31.1	38.0	39.9	41.8	42.6	40.3
0.4	30.3	29.8	31.3	37.4	38.7	40.8	41.5	43.2
0.3	30.3	31.1	29.9	37.8	42.1	41.7	42.8	43.2
0.2	31.4	29.8	31.0	39.3	40.7	42.2	42.0	44.6
0.1	30.4	30.4	29.9	36.8	41.1	40.2	43.1	43.5

Tabela 5.96: Terceira estratégia – Número de iterações do melhor indivíduo variando com p_{new} e p_{mut} - Valores médios de 10 execuções.

e p_{mut} . A Tabela 5.103 mostra o número total de iterações variando com p_{new} e p_{mut} e, finalmente, a Tabela 5.104 mostra o peso da estrutura otimizada variando com p_{new} e p_{mut} . Para todas estas Tabelas tem-se valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.

Observou-se que adotando-se o vetor de partida como solução obtida na avaliação anterior no GSR obteve-se, simplesmente, um menor número total de iterações e consequentemente um menor número de iterações médio. Dessa forma, o custo computacional foi menor mas não foram observadas melhores soluções quanto à otimização propriamente dita.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	1113446	1086871	974830	991080	1128625	1018960	1019326	1094777
0.9	1161962	1102582	994444	955778	948011	1059559	1090242	1086767
0.8	1161960	1103505	993879	1001406	1050360	1136025	1053465	1117975
0.7	1160100	1102685	1000178	995506	956163	1049878	1057920	1019624
0.6	1176725	1102684	992480	983585	970583	884372	1128633	1020342
0.5	1179239	1124858	1032323	982178	1136173	1092520	960982	1011610
0.4	1179243	1143309	1034023	984024	994054	1039688	1027325	1097860
0.3	1179276	1102800	1022687	932478	1248757	1047774	975903	952856
0.2	1179293	1089871	993505	972984	947971	1041219	1062444	998848
0.1	1176403	1133922	998132	910724	1030711	1007269	1029490	1112907

Tabela 5.97: Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} .

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	1149818.5	1064886.0	993944.4	1007002.1	1063199.9	1062574.9	1099650.1	1035747.5
0.9	1142621.2	1074770.2	993022.5	997617.7	1038526.4	1065571.1	1090087.0	1072456.0
0.8	1144643.7	1074282.6	993779.5	972496.3	1072814.1	1089564.6	1057509.8	1067084.4
0.7	1144506.4	1072704.1	992442.9	969212.5	1026197.5	1048168.2	1061247.5	1057222.0
0.6	1148217.8	1073365.7	997703.4	969541.7	1002443.9	1021638.0	1071788.2	1030145.2
0.5	1147230.0	1072859.7	1000209.7	974845.0	1000575.3	1024891.4	1033089.8	1009395.2
0.4	1144763.8	1073765.2	993800.5	979236.2	1002464.1	1010064.5	1018524.3	1073204.9
0.3	1145141.7	1075414.9	994616.9	959872.9	1057884.4	1038928.5	1029818.1	1050811.6
0.2	1147394.4	1067729.5	989560.6	986074.8	998104.9	1056058.5	1032499.8	1065941.8
0.1	1139201.3	1076210.5	987736.7	939400.2	1035138.8	1013471.6	1058735.1	1044301.3

Tabela 5.98: Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} - Valores médios de 10 execuções.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	5062.951	5061.150	5061.796	5061.616	5061.326	5061.231	5063.724	5062.047
0.9	5061.044	5061.451	5062.919	5061.803	5061.441	5061.649	5061.186	5062.010
0.8	5061.044	5061.451	5062.919	5061.803	5061.987	5061.944	5061.521	5062.798
0.7	5061.044	5061.451	5062.919	5061.803	5064.366	5061.833	5076.820	5061.567
0.6	5061.044	5061.457	5062.919	5061.440	5061.520	5061.944	5062.981	5061.567
0.5	5061.044	5061.457	5062.939	5061.440	5063.117	5061.584	5062.981	5062.324
0.4	5061.044	5061.457	5062.939	5061.440	5061.192	5061.584	5061.967	5062.246
0.3	5061.044	5061.451	5062.939	5061.803	5063.117	5060.941	5061.979	5065.136
0.2	5061.044	5061.451	5062.939	5061.803	5061.713	5060.941	5061.876	5065.254
0.1	5061.044	5064.241	5062.919	5064.869	5061.713	5063.154	5066.302	5063.922

Tabela 5.99: Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} .

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	5115.101	5092.122	5075.509	5099.131	5093.448	5094.295	5138.490	5093.277
0.9	5071.149	5116.579	5072.011	5105.290	5095.327	5081.990	5073.078	5070.141
0.8	5071.149	5116.608	5072.011	5122.965	5116.051	5081.113	5115.539	5075.358
0.7	5071.111	5116.326	5072.011	5123.840	5093.440	5093.817	5119.866	5076.775
0.6	5071.110	5116.449	5072.003	5110.636	5071.142	5115.848	5113.998	5094.197
0.5	5071.111	5116.776	5073.510	5108.871	5093.183	5112.903	5094.126	5075.091
0.4	5071.111	5117.827	5073.678	5113.420	5073.058	5092.532	5117.043	5096.601
0.3	5071.111	5116.313	5073.914	5115.582	5075.058	5090.964	5118.801	5076.410
0.2	5071.111	5116.372	5073.521	5095.651	5095.514	5091.980	5079.644	5095.173
0.1	5077.630	5118.045	5094.049	5115.949	5073.835	5116.990	5118.726	5086.746

Tabela 5.100: Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} – Valores médios de 10 execuções.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	1.235	1.2425	1.27	1.465	1.4975	1.5375	1.5125	1.525
0.9	1.248164	1.260236	1.265749	1.480908	1.513283	1.522506	1.514930	1.506986
0.8	1.231507	1.281864	1.276242	1.446462	1.509018	1.492936	1.520673	1.499741
0.7	1.229949	1.250790	1.255393	1.451281	1.465859	1.500186	1.502727	1.511115
0.6	1.210693	1.230652	1.265683	1.450452	1.470169	1.502355	1.510424	1.516766
0.5	1.248322	1.241399	1.250080	1.438926	1.494753	1.495860	1.495803	1.510668
0.4	1.253809	1.248011	1.298155	1.435391	1.497545	1.506394	1.509575	1.516672
0.3	1.305617	1.241251	1.283549	1.443241	1.482049	1.502284	1.494277	1.497563
0.2	1.248492	1.253215	1.277395	1.444979	1.463867	1.485226	1.510865	1.496066
0.1	1.228817	1.240912	1.242759	1.453668	1.487451	1.491232	1.490793	1.518376

Tabela 5.101: Terceira estratégia – Valores finais médios de ω variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	22.6	21.6	20.9	24.9	23.2	27.1	24.3	26.1
0.9	21.0	20.8	21.0	26.4	26.1	25.9	26.5	25.6
0.8	20.5	20.9	20.6	24.1	26.5	23.9	26.9	25.1
0.7	20.5	20.5	21.3	24.9	24.2	24.2	25.8	24.4
0.6	21.4	19.7	20.9	24.4	25.7	24.3	25.9	25.2
0.5	22.3	20.6	20.9	23.9	26.0	25.2	25.3	24.6
0.4	21.3	20.2	20.5	22.7	26.2	26.0	27.1	25.4
0.3	21.2	21.0	21.4	23.6	24.6	25.3	24.4	25.2
0.2	20.7	20.9	20.9	23.3	23.5	23.9	26.3	24.4
0.1	20.6	21.4	20.1	23.5	25.3	25.0	23.9	25.8

Tabela 5.102: Terceira estratégia – Número de iterações do melhor indivíduo variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	970851.7	907568.1	854477.8	832165.8	856186.4	904676.6	864778.7	892920.8
0.9	974106.7	904697.4	857902.5	873516.2	884329.4	900580.7	883240.5	853550.0
0.8	968731.6	898898.4	856117.1	828428.6	885354.6	848512.8	886830.8	860120.7
0.7	971832.7	899854.3	851282.7	825072.5	837291.3	853400.2	866185.8	877791.6
0.6	969802.1	903382.2	860108.8	827221.1	848370.7	859231.1	877863.2	877487.9
0.5	969009.1	901577.1	847341.0	810541.4	864186.8	850998.8	865802.3	869077.3
0.4	969678.2	902339.5	849389.1	806049.8	865619.4	862765.0	882232.0	877853.0
0.3	974860.9	901167.3	842526.7	809096.4	843673.0	853580.8	848885.6	868663.9
0.2	966456.0	903578.2	851231.9	812991.7	829379.0	835323.5	863815.6	852727.2
0.1	964909.9	901122.2	850349.4	819606.7	851702.0	846486.4	843889.6	878185.9

Tabela 5.103: Terceira estratégia – Número total de iterações variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.

p_{mut}	p_{new}							
	0.002	0.005	0.01	0.1	0.2	0.3	0.4	0.5
1.0	5092.454	5071.099	5072.712	5092.823	5093.951	5095.815	5117.681	5095.598
0.9	5115.351	5112.531	5087.740	5077.396	5116.971	5096.743	5116.753	5093.790
0.8	5115.377	5112.527	5087.853	5076.318	5074.277	5095.683	5095.575	5073.392
0.7	5115.344	5112.573	5089.351	5071.577	5121.257	5092.846	5094.763	5097.953
0.6	5115.375	5112.536	5068.890	5076.776	5077.511	5094.394	5077.192	5074.254
0.5	5113.990	5112.589	5069.161	5071.836	5076.890	5106.268	5075.314	5075.584
0.4	5113.992	5112.618	5069.161	5074.077	5117.779	5099.214	5074.334	5094.993
0.3	5115.355	5112.523	5069.161	5073.407	5097.701	5099.444	5074.943	5103.203
0.2	5115.374	5112.521	5070.642	5068.690	5092.724	5083.516	5072.140	5076.911
0.1	5115.340	5090.435	5069.137	5070.631	5093.688	5093.352	5077.273	5095.701

Tabela 5.104: Terceira estratégia – Peso da estrutura otimizada variando com p_{new} e p_{mut} - Valores médios em 10 execuções considerando-se o vetor de partida como a solução da avaliação anterior.

5.8.4 Comentários

Neste exemplo foi proposta uma estratégia de co-evolução para a determinação do parâmetro ω a ser utilizado no algoritmo iterativo Gauss-Seidel com relaxação (GSR), para a resolução sistemas de equações lineares. Foi resolvido um problema de otimização estrutural através da evolução de uma população de estruturas e a busca do parâmetro ω , do GSR, que auxilia (“coopera”), no processo de obtenção da função aptidão de cada solução candidata da população de estruturas.

Foram propostas 3 estratégias para a análise deste problema, especificamente para a evolução da população de parâmetros. Observou-se – fato evidenciado pela primeira análise – que o espaço de busca ao qual se submete a população de parâmetros tem características de um “cenário de aptidão móvel” ao longo da evolução. Isto cria dificuldades para a definição exata da aptidão ao longo da evolução.

A segunda estratégia alterou a política de inserção dos novos elementos na população e a terceira introduziu modificações no processo de seleção, geração de filhos e na obtenção da aptidão.

Foram realizadas várias análises envolvendo várias parametrizações tanto no algoritmo genético quanto no GSR. Ainda, para a terceira estratégia, foram feitos estudos considerando-se para o GSR o vetor de partida como solução obtida na avaliação anterior.

Mostrou-se que as estratégias de co-evolução discutidas aqui foram eficientes. Das 3 propostas a que melhor se adequa ao problema de busca com “cenário de aptidão móvel” é a terceira onde mostrou-se que é possível o controle da diversidade da população através de probabilidades de geração de novos filhos e da escolha do tipo de operador genético a ser aplicado para obtenção dos novos filhos.

Evidentemente, na terceira estratégia, encontram-se dois parâmetros adicionais (p_{new} e p_{mut}), o que, a princípio, poderia dificultar a parametrização. Todavia, não foi o objetivo a indicação de valores ou intervalos ideais para estes parâmetros e sim mostrar que foram úteis para o problema.

Foram mostrados valores médios obtidos em 10 execuções de vários parâmetros envolvidos na co-evolução. Por estes não foi possível identificar vantagens sobre as melhores soluções para cada execução. Para uma comparação mais apurada devem ser realizadas outras análises, inclusive, com um maior número de execuções que acarretariam maior diversidade nas populações iniciais.

Foram realizadas, também, na terceira estratégia estudos com vetores de partida do GSR como solução obtida do sistema de equações da avaliação anterior de um elemento da população de estruturas e, neste caso, não foi observada nenhuma melhora quanto aos resultados da otimização tanto da estrutura quanto do parâmetro ω . Somente foi observado um número de iterações menor.

Pode-se concluir, também, principalmente das informações contidas nas Tabelas 5.93 e 5.94, que existe perda de diversidade aumentando-se a probabilidade de ocorrência de novos filhos p_{new} e, também, diminuindo-se a probabilidade dos novos filhos serem gerados por operadores de mutação p_{mut} .

O processo de co-evolução mostrado aqui pode ser abstraído para problemas de diversas naturezas que se enquadram nas situações analisadas neste exemplo.

5.9 Estrutura de uma torre de transmissão

Na prática da otimização estrutural, as variáveis podem ser discretas ou contínuas como visto nos problemas de otimização de peso das treliças dos exemplo anteriores. Geralmente deseja-se que as seções transversais escolhidas para as barras estejam disponíveis comercialmente. Para o caso de variáveis contínuas este fato leva a uma solução mais difícil e uma hipótese seria escolher as seções disponíveis mais próximas daquelas definidas no processo de otimização e que satisfaçam as restrições impostas. Entretanto, isto não é uma tarefa trivial pois para cada valor contínuo, existirá um valor discreto imediatamente anterior e um imediatamente posterior e, se n for o número de variáveis, serão 2^n possibilidades de escolha. Dessa forma, como escolher os valores discretos? Pode-se pensar em escolher todos os valores imediatamente posteriores como uma solução conservadora imaginando-se atender restrições de tensões e deslocamentos, por exemplo, mas isso pode gerar uma solução ineficaz (estrutura ineficaz, onde o peso próprio é considerado como carregamento, por exemplo). Consequentemente, haverá a necessidade de uma segunda etapa para analisar a aproximação. Estes problemas são comuns quando utiliza-se métodos clássicos no processo de otimização.

Uma situação interessante que pode ser analisada é a de que dado o espaço de busca (discreto ou contínuo), define-se um número m máximo de diferentes seções possíveis a serem usadas no processo de otimização, objetivando-se uma simplificação no processo de compra, armazenagem e montagem da estrutura. Este número é definido pelo projetista de acordo com o projeto em questão [103].

Para o caso de variáveis discretas, o algoritmo busca simultaneamente, entre as seções disponíveis comercialmente, o sub-conjunto ótimo (de cardinalidade não superior a m), de propriedades a serem utilizadas e a respectiva designação para cada variável de projeto. Para o caso de variáveis contínuas, o algoritmo busca simultaneamente o conjunto (de cardinalidade não superior a m) de propriedades a serem usadas, dentro de um dado intervalo, e a respectiva atribuição para cada variável de projeto.

O Problema pode ser colocado na forma: Encontre o conjunto de áreas $\mathbf{a} = \{A_1, A_2, \dots, A_n\}$ que minimiza o peso da estrutura

$$W(\mathbf{a}) = \sum_{i=1}^n \rho A_i L_i$$

onde L_i é o comprimento do i -ésimo membro da treliça e ρ a densidade do material. O problema está submetido às restrições de tensões

$$\frac{|\sigma_i|}{\sigma_{max}} - 1 \leq 0 \quad \forall i = 1, 2, \dots, n$$

e a restrição adicional de que no máximo m diferentes áreas podem ser utilizadas, isto é,

$$A_i \in C_m = \{S_1, S_2, \dots, S_m\} \quad \forall i = 1, 2, \dots, n$$

onde as áreas $S_i, i = 1, 2, \dots, m$ são incógnitas que pertencem:

- a um dado conjunto $S = \{A_1, A_2, \dots, A_p\}$ para o caso discreto e
- ao intervalo prescrito $R = [A_{min}, A_{max}]$ para o caso contínuo.

Para exemplificar o esquema de codificação, imagine um conjunto com 32 seções disponíveis comercialmente onde no máximo 4 tipos serão utilizadas. O cromossomo de uma estrutura é dado por:

tipo(1) tipo(2) tipo(3) tipo(4) pt(1) pt(2)...pt(n)

onde

tipo(1) ... tipo(4)

são os ponteiros para as 4 diferentes seções a serem escolhidas a partir de uma tabela de 32 seções disponíveis, por exemplo.

tab(1) tab(2) ... tab(32)

e

pt(1) pt(2) ... pt(n)

definem cada variável de projeto apontando para um dos 4 tipos listados no começo do cromossomo para cada variável de projeto. Então, a área da i -ésima barra da estrutura é dada por:

area(i)=tab(tipo(pt(i)))

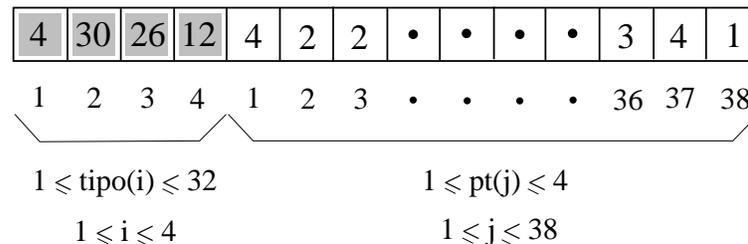


Figura 5.71: Cromossomo para o caso discreto – n=38.

Para o caso contínuo, analogamente ao caso discreto, tem-se o cromossomo:

size(1) size(2) size(3) size(4) pt(1) pt(2)...pt(n)

e a área da i -ésima barra da estrutura é dada por:

area(i)=size(pt(i))

Para o exemplo de codificação para o caso discreto, mostrado na Figura 5.71 a variável $\text{pt}(36)$, por exemplo, aponta para a variável $\text{tipo}(3)$ que aponta para o índice 26 de uma tabela de 32 seções disponíveis, como no caso da Tabela 5.107, ou seja:

area(36)=tab(tipo(pt(36)))=27.5

No exemplo de codificação para o caso contínuo mostrado na Figura 5.72 a variável $\text{pt}(3)$, por exemplo, aponta para a variável $\text{size}(2)$ que possui o valor de 30.91, ou seja:

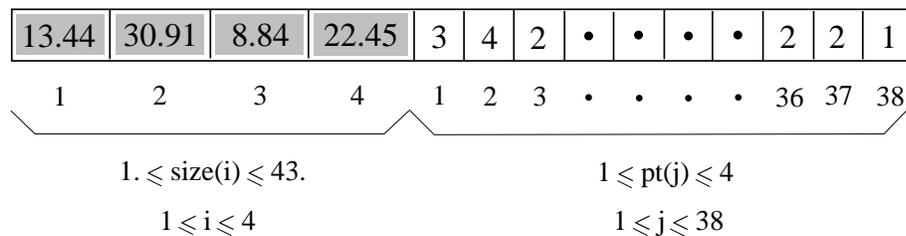


Figura 5.72: Cromossomo para o caso contínuo – n=38.

$\text{area}(3)=\text{size}(\text{pt}(2))=30.91$

A torre mostrada na Figura 5.73, tem 38 variáveis de projeto referentes às áreas das barras. A Tabela 5.106 mostra a designação das variáveis e a conectividade entre os elementos.

Para o caso discreto, as áreas das seções transversais das barras são selecionadas a partir de opções mostradas nas Tabelas 5.107 e para o caso contínuo as áreas são escolhidas no intervalo $[1.0, 43.0]$ cm².

A Tabela 5.108 mostra as propriedades do material e a Tabela 5.105 mostra os 8 casos de carregamento aos quais a estrutura é submetida e a Tabela 5.109 mostra as coordenadas dos nós.

A torre será submetida a um processo de otimização para a minimização do seu peso com restrições nas tensões normais admissíveis.

A função a ser otimizada é escrita da seguinte forma:

$$f(\mathbf{a}) = W(\mathbf{a}) + \text{penal} \sum_i \left(\left[\frac{|\sigma_j|}{\sigma_{max}} - 1 \right]^+ \right)^2$$

onde *penal* é o coeficiente de penalização definido com o valor de 200.

Utilizou-se um algoritmo genético do tipo “geracional” com codificação binária em todas as análises deste exemplo. Para todos os casos os operadores genéticos de recombinação usados são três: *crossover* de um ponto, dois pontos e uniforme, aplicados com as seguintes probabilidades relativas: $p_c^1 = 0.2$, $p_c^2 = 0.4$ e $p_c^u = 0.4$. A operação de recombinação foi definida com probabilidade $p_c = 0.8$ e para a mutação adotou-se uma taxa de $p_m = 0.01$.

As codificações dos cromossomos (variáveis *tipo* e *pt*), tamanhos dos cromossomos (*lcrom*) e número de variáveis de projeto (*nvarp*) são mostrados nas Tabelas 5.110 e 5.111 para os casos discreto e contínuo, respectivamente.

Foram realizadas 10 execuções para cada análise e os tamanhos das populações (*popsiz*), número de gerações (*maxgen*) e os valores obtidos na otimização (média e melhor execução), são mostrados nas Tabelas 5.112 e 5.113 referentes aos casos discreto e contínuo, respectivamente.

Nota-se, como esperado, pela Tabelas 5.112 e 5.113, que à medida que cresce o valor de “m” o peso da estrutura diminui, ou seja, diminuem-se as restrições de escolha das seções transversais e, conseqüentemente, aumentando-se o espaço de busca.

As Tabelas 5.114 e 5.115 mostram, para cada valor de “m”, as seções transversais escolhidas para os casos discreto e contínuo, respectivamente. Nota-se na Tabela

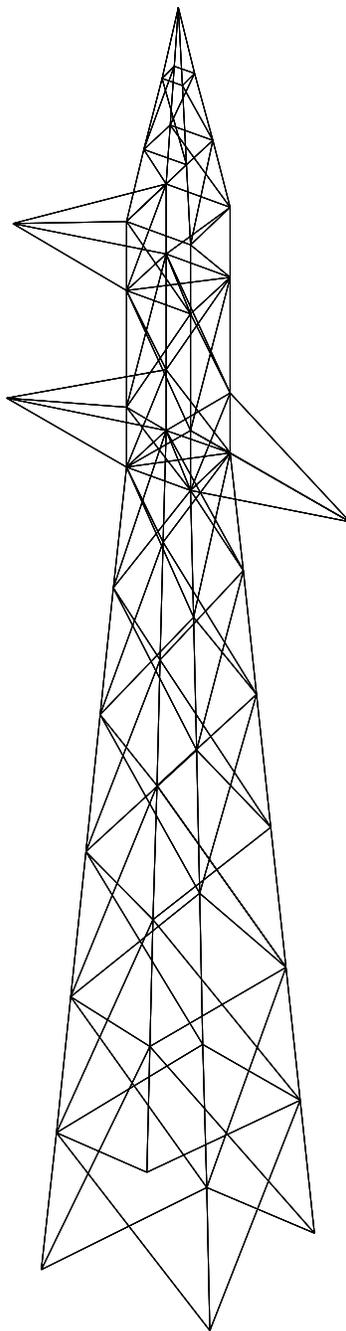


Figura 5.73: Torre de transmissão com 160 barras.

Caso de carregamento	Nó	F_x	F_y	F_z
1	52	-868.0	0.0	-491.0
	37	-996.0	0.0	-546.0
	25	-1091.0	0.0	-546.0
	28	-1091.0	0.0	-546.0
2	52	-493.0	1245.0	-363.0
	37	-996.0	0.0	-546.0
	25	-1091.0	0.0	-546.0
	28	-1091.0	0.0	-546.0
3	52	-917.0	0.0	-491.0
	37	-951.0	0.0	-546.0
	25	-1015.0	0.0	-546.0
	28	-1015.0	0.0	-546.0
4	52	-917.0	0.0	-546.0
	37	-572.0	1259.0	-428.0
	25	-1015.0	0.0	-546.0
	28	-1015.0	0.0	-546.0
5	52	-917.0	0.0	-491.0
	37	-951.0	0.0	-546.0
	25	-1015.0	0.0	-546.0
	28	-1015.0	1259.0	-428.0
6	52	-917.0	0.0	-491.0
	37	-572.0	1303.0	-428.0
	25	-1015.0	0.0	-546.0
	28	-1015.0	0.0	-546.0
7	52	-917.0	0.0	-491.0
	37	-951.0	0.0	-546.0
	25	-1015.0	0.0	-546.0
	28	-636.0	1303.0	-428.0
8	52	-498.0	1460.0	-363.0
	37	-951.0	0.0	-546.0
	25	-1015.0	0.0	-546.0
	28	-1015.0	0.0	-546.0

Tabela 5.105: Dados de carregamento - cargas em kg.

No.	i	j	x_j												
1	1	5	1	41	13	18	8	81	25	31	17	121	36	40	29
2	2	6	1	42	14	17	8	82	28	32	17	122	38	41	29
3	3	7	1	43	14	19	8	83	28	33	17	123	39	42	29
4	4	8	1	44	15	18	8	84	25	34	17	124	35	43	29
5	1	6	2	45	15	20	8	85	26	31	18	125	40	41	30
6	2	5	2	46	16	19	8	86	27	32	18	126	41	42	30
7	2	7	2	47	16	17	8	87	29	33	18	127	42	43	30
8	3	6	2	48	13	20	8	88	30	34	18	128	43	40	30
9	3	8	2	49	17	21	9	89	26	32	19	129	35	36	31
10	4	7	2	50	18	22	9	90	27	31	19	130	36	38	31
11	4	5	2	51	19	23	9	91	29	34	19	131	38	39	31
12	1	8	2	52	20	24	9	92	30	33	19	132	39	35	31
13	5	9	3	53	17	22	10	93	27	33	20	133	40	44	32
14	6	10	3	54	18	21	10	94	29	32	20	134	41	45	32
15	7	11	3	57	19	24	10	95	30	31	20	135	42	46	32
16	8	12	3	58	20	23	10	96	26	34	20	136	43	47	32
17	5	10	4	55	18	23	11	97	26	29	21	137	40	45	33
18	6	9	4	56	19	22	11	98	27	30	21	138	41	46	33
19	6	11	4	59	20	21	11	99	31	35	22	139	42	47	33
20	7	10	4	60	17	24	11	100	32	36	22	140	43	44	33
21	7	12	4	61	21	26	12	101	33	38	22	141	44	45	34
22	8	11	4	62	22	27	12	102	34	39	22	142	45	46	34
23	8	9	4	63	23	29	12	103	33	39	23	143	46	47	34
24	5	12	4	64	24	30	12	104	32	35	23	144	44	47	34
25	9	13	5	65	21	27	13	105	31	36	23	145	44	48	35
26	10	14	5	66	22	26	13	106	34	38	23	146	45	49	35
27	11	15	5	67	23	30	13	107	32	38	24	147	46	50	35
28	12	16	5	68	24	29	13	108	33	36	24	148	47	51	35
29	9	14	6	69	22	29	14	109	34	35	24	149	45	48	36
30	10	13	6	70	23	27	14	110	31	39	24	150	46	49	36
31	10	15	6	71	24	26	14	111	37	35	25	151	47	50	36
32	11	14	6	72	21	30	14	112	37	39	25	152	44	51	36
33	11	16	6	73	26	27	15	113	37	40	26	153	48	49	37
34	12	15	6	74	27	29	15	114	37	43	26	154	49	50	37
35	12	13	6	75	29	30	15	115	35	40	27	155	50	51	37
36	9	16	6	76	30	26	15	116	36	41	27	156	48	51	37
37	13	17	7	77	25	26	16	117	38	42	27	157	48	52	38
38	14	18	7	78	27	28	16	118	39	43	27	158	49	52	38
39	15	19	7	79	25	30	16	119	35	38	28	159	50	52	38
40	16	20	7	80	29	28	16	120	36	39	28	160	51	52	38

Tabela 5.106: Conectividades dos elementos e variáveis de projeto.

No.	cm ²						
1	1.84	9	5.75	17	11.38	25	25.12
2	2.26	10	6.25	18	12.21	26	27.50
3	2.66	11	6.84	19	13.79	27	29.88
4	3.07	12	7.44	20	15.39	28	32.76
5	3.47	13	8.06	21	17.03	29	33.90
6	3.88	14	8.66	22	19.03	30	34.47
7	4.79	15	9.40	23	21.12	31	39.16
8	5.27	16	10.47	24	23.20	32	43.00

Tabela 5.107: Seções transversais disponíveis para o caso discreto – cm².

propriedade	valor
Módulo de elasticidade	$2.047 \times 10^6 \text{ kg/cm}^2$
Tensões admissíveis	$\pm 15000.0 \text{ kg/cm}^2$
Massa específica	0.00785 kg/cm^3

Tabela 5.108: Propriedades do material e restrições.

5.114 que ocorreu a repetição da seção transversal de valor 1.84, para os casos com “m=6” e “m=8”. Esta repetição é possível de acontecer pois “m” representa o número máximo de seções a serem escolhidas não havendo restrições impostas quanto a repetição de valores escolhidos. Na Tabela 5.114 nota-se, também, a presença da seção transversal 29.88 cm² que não foi designada para nenhuma variável de projeto e esta, provavelmente, desapareceria das possibilidades de escolha se a evolução continuasse por mais gerações. No caso contínuo não houve nenhuma ocorrência dessa natureza como observado na Tabela 5.115.

As Tabelas 5.116 e 5.117 mostram as designações das seções transversais escolhidas para as áreas das barras para os casos discreto e contínuo, respectivamente.

As Figuras 5.118 e 5.119 ilustram a variação da função aptidão da melhor solução entre 10 execuções independentes ao longo do processo de evolução, comparando-se as soluções para os diferentes valores de “m” escolhidos. Pelos aspectos das curvas nessas Figuras, nota-se que o algoritmo genético encontrou “rapidamente” os caminhos que mostram boas soluções.

As Figuras 5.120 e 5.121 mostram os valores médios da função aptidão entre 10 execuções independentes ao longo do processo de evolução, comparando-se as soluções para os diferentes valores de “m” escolhidos.

Nó	X	Y	Z	Nó	X	Y	Z
1	-105.000	-105.000	0.000	27	40.000	40.000	1027.500
2	105.000	-105.000	0.000	28	214.000	0.000	1027.500
3	105.000	105.000	0.000	29	40.000	40.000	1027.500
4	-105.000	105.000	0.000	30	-40.000	40.000	1027.500
5	-93.929	-93.929	175.000	31	-40.000	-40.000	1105.500
6	93.929	-93.929	175.000	32	40.000	-40.000	1105.500
7	93.929	93.929	175.000	33	40.000	40.000	1105.500
8	-93.929	93.929	175.000	34	-40.000	40.000	1105.500
9	-82.859	-82.859	350.000	35	-40.000	-40.000	1256.500
10	82.859	-82.859	350.000	36	40.000	-40.000	1256.500
11	82.859	82.859	350.000	37	-207.000	0.000	1256.500
12	-82.859	82.859	350.000	38	40.000	40.000	1256.500
13	-71.156	-71.156	535.000	39	-40.000	40.000	1256.500
14	71.156	-71.156	535.000	40	-40.000	-40.000	1346.500
15	71.156	71.156	535.000	41	40.000	-40.000	1346.500
16	-71.156	71.156	535.000	42	40.000	40.000	1346.500
17	-60.085	-60.085	710.000	43	-40.000	40.000	1346.500
18	60.085	-60.085	710.000	44	-26.592	-26.592	1436.500
19	60.085	60.085	710.000	45	26.592	-26.592	1436.500
20	-60.085	60.085	710.000	46	26.592	26.592	1436.500
21	-49.805	-49.805	872.500	47	-26.592	26.592	1436.500
22	49.805	-49.805	872.500	48	-12.737	-12.737	1526.500
23	49.805	49.805	872.500	49	12.737	-12.737	1526.500
24	-49.805	49.805	872.500	50	12.737	12.737	1526.500
25	-214.000	0.000	1027.500	51	-12.737	12.737	1526.500
26	-40.000	-40.000	1027.500	52	0.000	0.000	1615.000

Tabela 5.109: Coordenadas dos nós – cm.

m	tipo	pt	lcrom	nvarp
2	2 x 5	38 x 1	48	40
4	4 x 5	38 x 2	96	42
6	6 x 5	38 x 3	144	44
8	8 x 5	38 x 3	154	46

Tabela 5.110: Codificação do cromossomo – caso discreto.

m	size	pt	lcrom	nvarp
2	2 x 16	38 x 1	48	70
4	4 x 16	38 x 2	96	140
6	6 x 16	38 x 3	144	210
8	8 x 16	38 x 3	154	242

Tabela 5.111: Codificação do cromossomo – caso contínuo.

m	popsiz	maxgen	média	melhor execução
2	160	100	952.4255	888.5458
4	160	100	812.1743	735.7504
6	160	100	786.1055	690.5617
8	160	100	747.8376	683.3990

Tabela 5.112: Peso obtido: caso discreto (kg).

m	popsiz	maxgen	média	melhor execução
2	160	100	989.1697	818.3813
4	160	100	721.8355	582.7344
6	160	140	657.9024	573.5612
8	160	160	735.2885	550.8472

Tabela 5.113: Peso obtido: caso contínuo (kg).

m=2	m=4	m=6	m=8
11.38	11.38	11.38	11.38
2.66	10.47	10.47	10.47
–	9.40	3.88	9.40
–	1.84	2.66	3.88
–	–	1.84	2.66
–	–	1.84	1.84
–	–	–	1.84
–	–	–	29.88

Tabela 5.114: Seções transversais selecionadas – caso discreto.

m=2	m=4	m=6	m=8
10.65997	10.99258	10.64651	10.84710
2.40032	9.36859	9.90116	10.29915
–	3.54108	4.95037	9.12570
–	1.00000	2.97070	4.77990
–	–	2.32661	2.98928
–	–	1.00128	2.32277
–	–	–	1.67100
–	–	–	1.00000

Tabela 5.115: Seções transversais selecionadas – caso contínuo.

x_j	m=2	m=4	m=6	m=8
1	11.38	11.38	11.38	11.38
2	2.66	1.84	1.84	1.84
3	11.38	10.47	10.47	10.47
4	2.66	1.84	1.84	1.84
5	11.38	10.47	10.47	10.47
6	2.66	1.84	1.84	1.84
7	11.38	9.40	10.47	9.40
8	2.66	1.84	1.84	1.84
9	11.38	9.40	10.47	9.40
10	2.66	1.84	1.84	1.84
11	2.66	1.84	1.84	1.84
12	11.38	9.40	10.47	9.40
13	2.66	1.84	1.84	1.84
14	2.66	1.84	1.84	1.84
15	2.66	1.84	1.84	1.84
16	2.66	1.84	1.84	1.84
17	2.66	1.84	1.84	1.84
18	11.38	9.40	10.47	9.40
19	2.66	1.84	1.84	1.84
20	2.66	1.84	1.84	1.84
21	2.66	1.84	1.84	1.84
22	11.38	9.40	3.88	3.88
23	2.66	1.84	1.84	1.84
24	2.66	1.84	1.84	1.84
25	2.66	1.84	1.84	1.84
26	2.66	1.84	1.84	1.84
27	11.38	9.40	3.88	9.40
28	2.66	1.84	1.84	1.84
29	2.66	1.84	1.84	1.84
30	2.66	1.84	1.84	1.84
31	2.66	1.84	1.84	1.84
32	2.66	9.40	2.66	2.66
33	2.66	1.84	1.84	1.84
34	2.66	1.84	1.84	1.84
35	2.66	9.40	3.88	2.66
36	2.66	1.84	1.84	1.84
37	2.66	1.84	1.84	1.84
38	2.66	9.40	2.66	2.66
PESO (kg)	888.5458	735.7504	690.5617	683.3990

Tabela 5.116: Designação das seções transversais escolhidas para as áreas das barras – caso discreto.

x_j	m=2	m=4	m=6	m=8
1	10.65997	10.99258	10.64651	10.84710
2	2.40032	1.00000	1.00128	1.00000
3	10.65997	10.99258	10.64651	10.29915
4	2.40032	1.00000	1.00128	1.00000
5	10.65997	10.99258	9.90116	10.29915
6	2.40032	1.00000	1.00128	1.00000
7	10.65997	9.36857	9.90116	9.12570
8	2.40032	1.00000	1.00128	1.00000
9	10.65997	9.36857	10.64651	9.12570
10	2.40032	1.00000	1.00128	1.67100
11	2.40032	1.00000	1.00128	1.00000
12	10.65997	9.36857	10.64651	9.12570
13	2.40032	1.00000	1.00128	1.00000
14	2.40032	1.00000	1.00128	1.00000
15	2.40032	1.00000	1.00128	1.00000
16	2.40032	1.00000	1.00128	1.00000
17	2.40032	1.00000	1.00128	1.00000
18	10.65997	9.36857	9.90116	9.12570
19	2.40032	1.00000	1.00128	1.00000
20	2.40032	1.00000	1.00128	1.67100
21	2.40032	1.00000	2.97070	1.00000
22	10.65997	9.36857	4.95037	4.77990
23	2.40032	1.00000	1.00128	1.00000
24	2.40032	1.00000	1.00128	1.00000
25	2.40032	1.00000	1.00128	1.00000
26	2.40032	1.00000	1.00128	1.00000
27	10.65997	3.54108	2.97070	2.98928
28	2.40032	1.00000	2.97070	1.00000
29	2.40032	1.00000	1.00128	1.00000
30	2.40032	1.00000	1.00128	1.67100
31	2.40032	1.00000	1.00128	1.67100
32	2.40032	3.54108	2.97070	2.32277
33	2.40032	1.00000	1.00128	1.00000
34	2.40032	1.00000	1.00128	1.67100
35	2.40032	3.54108	2.32661	2.32277
36	2.40032	1.00000	2.97070	1.67100
37	2.40032	1.00000	1.00128	1.00000
38	2.40032	3.54108	2.97070	2.98928
PESO (kg)	818.3813	582.7344	573.5612	550.8472

Tabela 5.117: Designação das seções transversais escolhidas para as áreas das barras – caso contínuo.

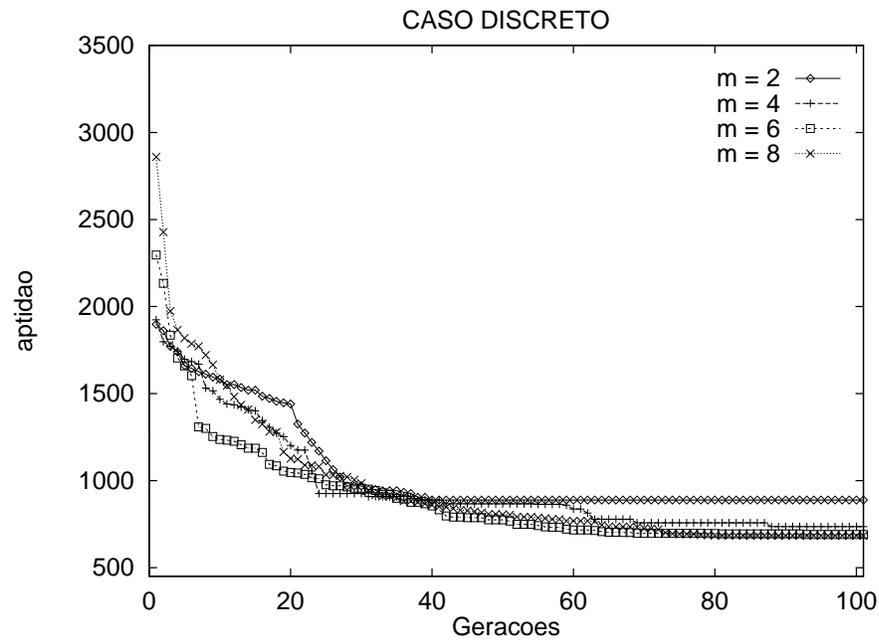


Tabela 5.118: Melhores execuções – caso discreto.

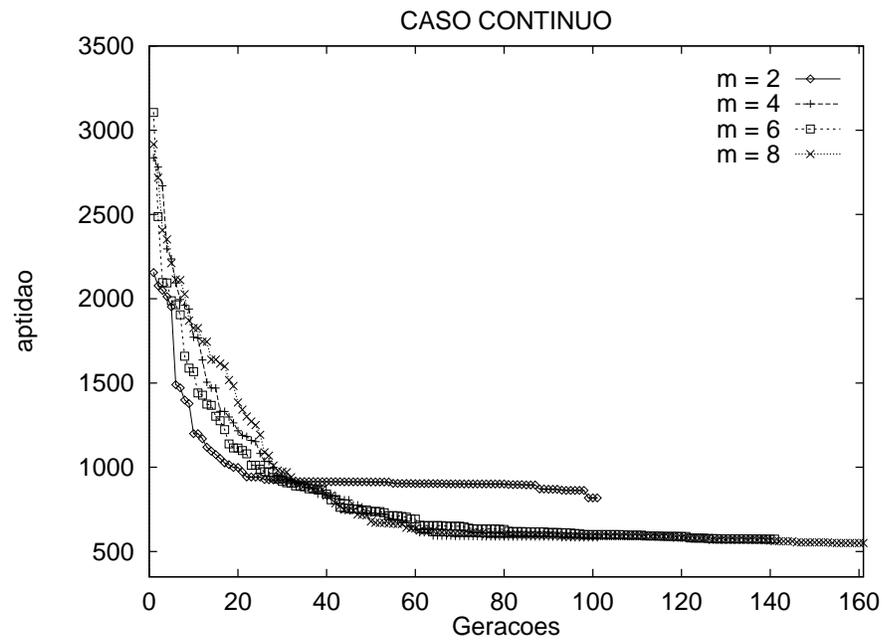


Tabela 5.119: Melhores execuções – caso contínuo.

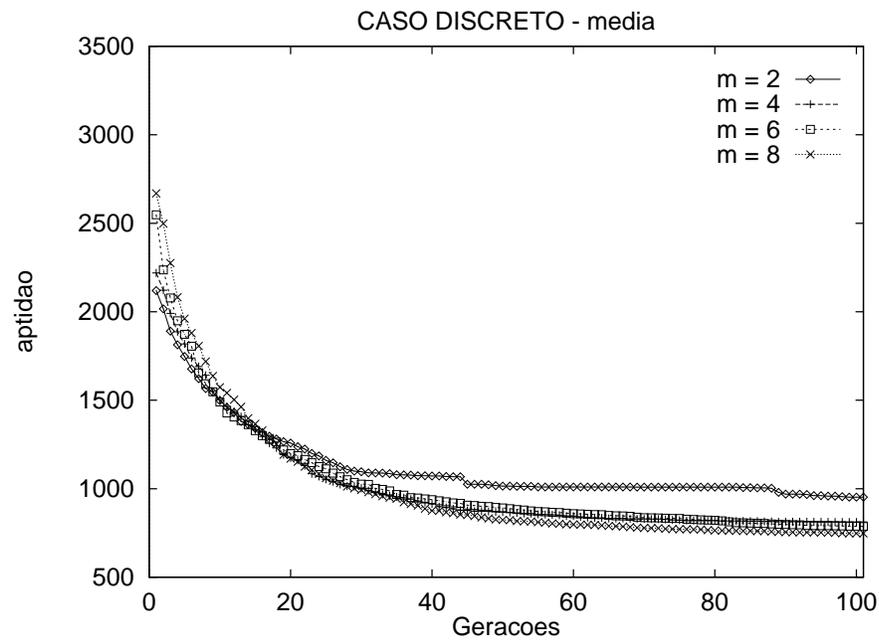


Tabela 5.120: Média – caso discreto.

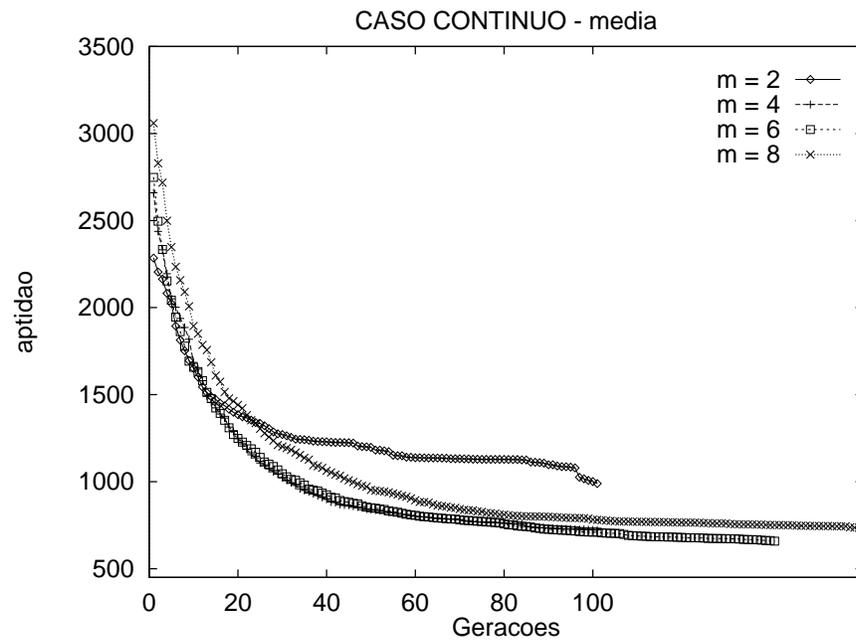


Tabela 5.121: Média – caso contínuo.

5.9.1 Comentários

O aspecto relevante deste exemplo foi a validação de uma técnica proposta para resolver um problema de otimização com variáveis discretas e contínuas através da busca de um conjunto de opções cardinalidade não superior “ m ” de opções a serem designadas para as variáveis de projeto. Isto foi realizado propondo-se uma nova codificação do cromossomo envolvendo apontadores.

Este tipo de problema é importante em engenharia, e mesmo em outras áreas, pois auxilia a produção, a armazenagem das barras, a confecção e a verificação, entre outros aspectos, das estruturas a serem fabricadas. Para este problema não foi encontrada na literatura qualquer proposta de solução.

Este tipo de análise proporciona possibilidades de escolhas, a partir da confecção de uma curva ($m \times X$ peso), pode-se dizer, de soluções “Pareto-ótimas”. De posse dessa curva, o projetista, deve ser capaz de optar entre ter uma estrutura com grande número de barras diferentes e um menor peso, podendo gerar uma confecção mais demorada e mais cara, ou, ter um menor número de barras e um peso maior, podendo gerar uma confecção mais rápida e mais barata.

Não foi objeto deste exemplo a busca exaustiva da melhor solução (menor peso da estrutura), e sim mostrar a potencialidade da técnica. Por isso, os resultados obtidos com o processo de otimização não podem ser dados como os melhores possíveis de serem alcançados. Entretanto, eles representam uma boa medida pelo aspecto apresentado pelas curvas que mostram os processos de evolução. Além disso, análises mais apuradas com esses objetivos podem ser realizadas e parametrizações mais elaboradas (tamanho da população e número de gerações), devem ser consideradas já que o tamanho dos cromossomos cresce, à medida que cresce o tamanho do subconjunto procurado.

5.10 Evoluindo a configuração de um projeto estrutural

5.10.1 Introdução

Pretende-se neste exemplo investigar a capacidade de se definir qual o melhor projeto para uma estrutura através de um processo de evolução.

O objetivo é projetar uma treliça espacial que servirá como cobertura de uma edificação.

A área que a treliça deve cobrir é circular e são permitidos pontos de apoio somente no perímetro dessa área. Várias são as opções para a concepção de projetos arquitetônicos e estruturais destinados a esse objetivo.

Aqui, optou-se por uma concepção da treliça imaginando-se que a mesma seja constituída por módulos – conhecidos como “tesouras” na prática do projeto estrutural – como mostra a Figura 5.74, dispostos radialmente sobre a área a ser coberta. Estes módulos têm em comum um elemento de ligação, entre os nós i e j como mostra a Figura 5.74, localizado no eixo vertical de simetria da estrutura (eixo Z da Figura 5.74).

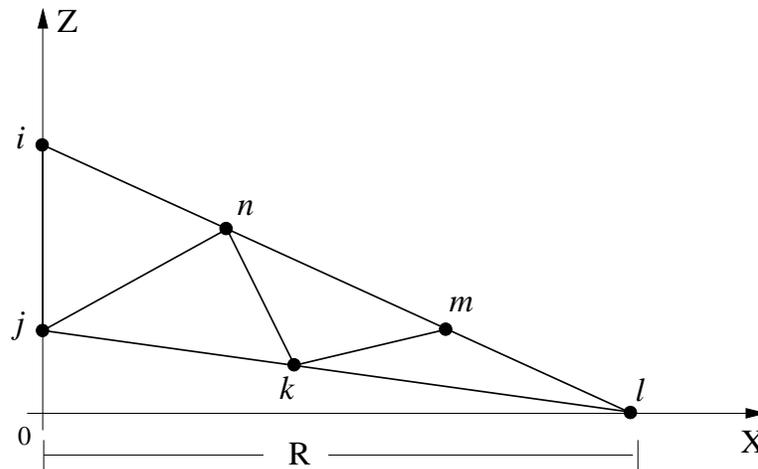


Figura 5.74: Módulo componente da treliça.

5.10.2 Características do projeto

As propriedades do material e limites admissíveis para as tensões normais são mostrados na Tabela 5.122.

propriedade	valor
Módulo de elasticidade	$2.047 \times 10^6 \text{ kg/cm}^2$
Tensões admissíveis	$\pm 1500.0 \text{ kg/cm}^2$
Massa específica	0.00785 kg/cm^3

Tabela 5.122: Propriedades do material.

As cargas que atuarão sobre a estrutura são provenientes do peso próprio da estrutura e da sobrecarga prevista para a cobertura. A Figura 5.75 mostra a definição das áreas de influências para a determinação de cargas nodais equivalentes provenientes da sobrecarga. Nesta Figura, mostra-se a projeção de uma possível configuração da estrutura com 8 módulos (N1 a N8) e as áreas A_m , A_n e A_i , que multiplicadas pelo valor da sobrecarga, definirão os valores de carregamentos para os nós m , n e i , respectivamente, e que serão válidas para o restante dos nós pertencentes aos outros módulos. Os limites entre essas áreas é a distância média, em projeção, dos nós. A sobrecarga para esse projeto é de 250 kg/m^2 .

5.10.3 Variáveis de projeto

O processo de otimização definirá qual a estrutura mais econômica, entendida como a de menor peso. Estão envolvidas na formulação do problema as seguintes variáveis de projeto:

- o número de módulos
- as coordenadas dos nós

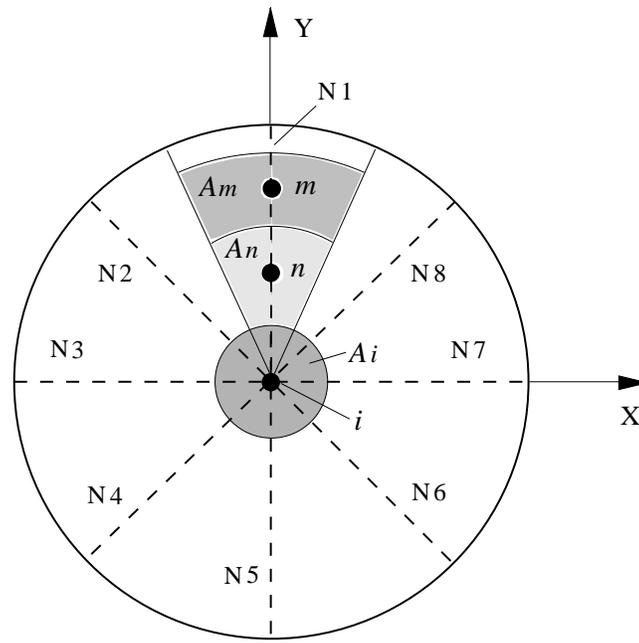


Figura 5.75: Áreas de influências para determinação de carregamento devido a sobrecarga.

- as áreas das barras

Na Figura 5.76 são mostradas as 5 variáveis de projeto associadas à forma. A variável X_2 refere-se à coordenada Z do nó j , X_3 à coordenada Z do nó i , X_4 à coordenada X do nó n , X_5 à coordenada X do nó k e finalmente X_6 à coordenada X do nó m . As coordenadas em relação ao eixo Z dos nós n , k e m são determinadas através de relações métricas do triângulo que compõe a geometria do módulo.

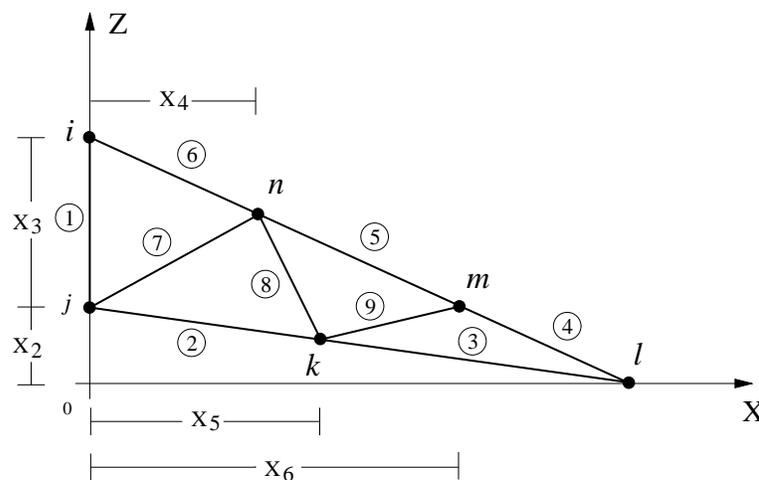


Figura 5.76: Variáveis de forma.

A numeração dos elementos indicada na Figura 5.76 corresponde a numeração do primeiro módulo e o agrupamento proposto para as barras que terão a mesma

seção transversal é dado por: $A_1, A_2 = A_3, A_4 = A_5 = A_6, A_7 = A_8 = A_9$ e, finalmente, o grupo formado pelos elementos horizontais que ligam os módulos entre si, representando 5 variáveis de projeto adicionais.

A escolha das áreas das barras é feita utilizando-se a estratégia apresentada no exemplo da torre de transmissão discutido anteriormente neste texto. O algoritmo busca simultaneamente, entre as seções disponíveis da Tabela 5.123, o sub-conjunto ótimo de cardinalidade não superior a 3, para este exemplo, a serem designados para as variáveis que representam as seções transversais das barras. A parcela do cromossomo que representa esta designação é dada por:

tipo(1) tipo(2) tipo(3) pt(1) pt(2) pt(3) pt(4) pt(5)

onde

tipo(1) ... tipo(3)

são os ponteiros para as 3 diferentes seções a serem escolhidas a partir da Tabela de 32 seções disponíveis na Tabela 5.123.

tab(1) tab(2) ... tab(32)

e

pt(1) pt(2) pt(4) pt(4) pt(5)

são os ponteiros de um dos 3 tipos listados no cromossomo para cada variável de projeto.

As áreas das seções transversais das barras são selecionadas a partir de opções mostradas nas Tabelas 5.123.

No.	cm ²						
1	5.0	9	9.0	17	13.0	25	17.0
2	5.5	10	9.5	18	13.5	26	17.5
3	6.0	11	10.0	19	14.0	27	18.0
4	6.5	12	10.5	20	14.5	28	18.5
5	7.0	13	11.0	21	15.0	29	19.0
6	7.5	14	11.5	22	15.5	30	19.5
7	8.0	15	12.0	23	16.0	31	20.0
8	8.5	16	12.5	24	16.5	32	20.5

Tabela 5.123: Seções transversais disponíveis – cm².

São listadas na Tabela 5.124 todas as variáveis de projeto e suas designações:

5.10.4 Codificação do cromossomo

Na Tabela 5.125 são mostrados para cada variável de projeto seus respectivos limites inferiores X^{INF} e superiores X^{SUP} e o número de bits (nb). Observa-se que o cromossomo tem um comprimento total de 89 bits. As variáveis X_1, X_2, X_3, X_4

Variável	designação
X_1	número de módulos
X_2	coordenada Z do nó j
X_3	coordenada Z do nó i
X_4	coordenada X do nó n
X_5	coordenada X do nó k
X_6	coordenada X do nó m
X_7	ponteiro que escolhe a seção a partir da Tabela 5.123
X_8	ponteiro que escolhe a seção a partir da Tabela 5.123
X_9	ponteiro que escolhe a seção a partir da Tabela 5.123
X_{10}	ponteiro que aponta para uma das variáveis X_7 , X_8 ou X_9 (área da barra A_1)
X_{11}	ponteiro que aponta para uma das variáveis X_7 , X_8 ou X_9 (área das barras A_2 e A_3)
X_{12}	ponteiro que aponta para uma das variáveis X_7 , X_8 ou X_9 (área das barras A_4 , A_5 e A_6)
X_{13}	ponteiro que aponta para uma das variáveis X_7 , X_8 ou X_9 (área das barras A_7 , A_8 e A_9)
X_{14}	ponteiro que aponta para uma das variáveis X_7 , X_8 ou X_9 (área dos elementos que ligam os módulos entre si)

Tabela 5.124: Variáveis de projeto e suas designações.

e X_5 têm seus valores dados em centímetros. A dimensão R mostrada na Figura 5.74 tem o valor de 1000.0 cm. Foram realizadas 10 execuções com população de 100 elementos evoluídas em 100 gerações. Os operadores de recombinação adotados foram o crossover de um ponto, dois pontos e uniforme, aplicados com as seguintes probabilidades relativas: $p_c^1 = 0.2$, $p_c^2 = 0.4$ e $p_c^u = 0.4$. A operação de recombinação foi definida com probabilidade $p_c = 0.9$ e para a mutação uma taxa $p_m = 0.01$ e foram realizadas 10 execuções. Utilizou-se um algoritmo genético do tipo geracional com codificação binária em todas as análises deste exemplo.

5.10.5 Resultados

A treliça espacial será submetida a um processo de otimização para a minimização do seu peso com restrições nas tensões normais admissíveis.

A função a ser otimizada é escrita da seguinte forma:

$$f(\mathbf{a}) = W(\mathbf{a}) + penal \sum_i \left(\left[\frac{|\sigma_j|}{\sigma_{max}} - 1 \right]^+ \right)^2$$

onde *penal* é o coeficiente de penalização definido com o valor de 200.

A Figura 5.77 mostra o valor médio da função aptidão bem como os valores de aptidões encontradas, respectivamente, para a melhor e pior execução. A Tabela 5.126 mostra os valores finais encontrados para as variáveis de projeto obtidas pela melhor execução. A variável X_1 indica 6 módulos que comporão a treliça espacial. As variáveis X_7 , X_8 e X_9 apontam respectivamente para os índices 25, 10 e 5 da

Variável	X^{INF}	X^{SUP}	nb
X_1	4	14	4
X_2	0.0	100.0	10
X_3	150.0	250.0	10
X_4	100.0	450.0	10
X_5	250.0	750.0	10
X_6	550.0	900.0	10
X_7	1	32	5
X_8	1	32	5
X_9	1	32	5
X_{10}	7	9	4
X_{11}	7	9	4
X_{12}	7	9	4
X_{13}	7	9	4
X_{14}	7	9	4

Tabela 5.125: Codificação e limites para as variáveis de projeto.

Tabela 5.123 de seções disponíveis e estes referem-se, respectivamente, aos valores de 17.0, 8.5 e 7.0, que encontram-se colocados na terceira coluna da Tabela 5.126, referindo-se a estas variáveis. As variáveis X_{10} , X_{11} , X_{12} , X_{13} e X_{14} , apontam, respectivamente, para as variáveis 7, 8, 7, 8 e 8, como mostra a segunda coluna da Tabela 5.126 e então assumem seus valores correspondentes de áreas, como mostra a terceira coluna da Tabela 5.126. O peso total da estrutura obtido após a otimização para a melhor execução é de 1561.787 kgs e de 1683.817 kgs para a média das execuções e de 1903.857 kgs para a pior execução. A Figura 5.78 mostra a forma final obtida para o módulo padrão.

5.10.6 Comentários

Mostrou-se neste exemplo a potencialidade de um algoritmo genético de evoluir a topologia e forma de uma estrutura. O algoritmo foi capaz de buscar qual a estrutura de menor peso destinada a atender requisitos arquitetônicos e estruturais.

Este é um aspecto atraente nas áreas de engenharia e arquitetura. Com esta ferramenta desenvolvida aqui, através de um algoritmo genético, é possível a busca de concepções estruturais e arquitetônicas propiciando novas soluções para os projetistas. Em geral a busca de novas concepções estruturais e arquitetônicas é limitada, por exemplo, pela falta de códigos computacionais específicos, impossibilidade de avaliação de um maior número de possíveis soluções e, até mesmo, pela “resistência” por parte de alguns projetistas em escolher projetos “diferentes” dos adotados tradicionalmente.

A evolução de forma como mostrada aqui poderia ser realizada através de cromossomos de diferentes tamanhos variando de acordo com o número de peças estruturais de cada estrutura concorrente no processo de evolução. Com certeza, a manipulação de cromossomos com tamanhos diferentes demanda uma certa sofisticação na implementação computacional. Entretanto, isto não ocorreu neste exemplo

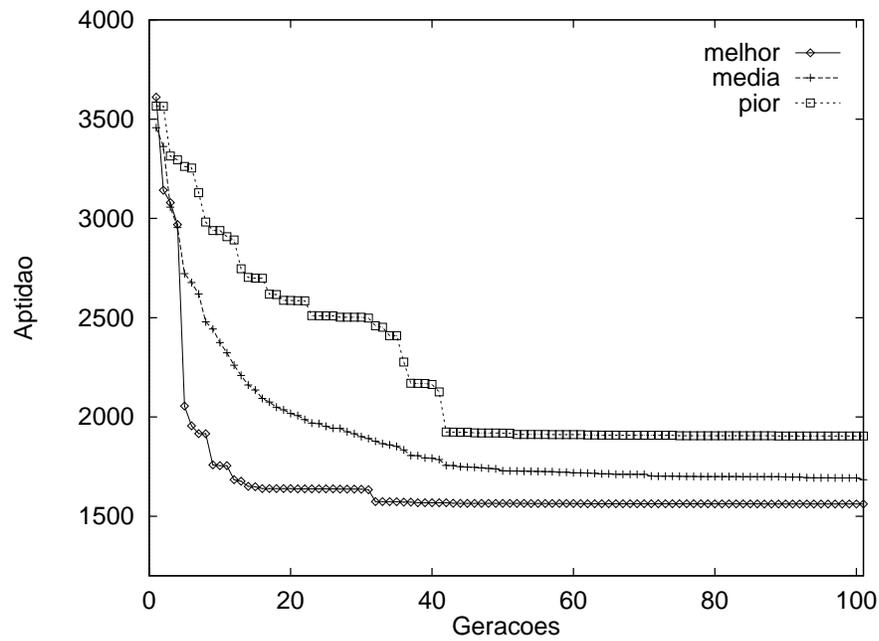


Figura 5.77: Função aptidão.

Variável	valor	áreas
X_1	6	–
X_2	90.61	–
X_3	244.13	–
X_4	257.03	–
X_5	359.74	–
X_6	550.0	–
X_7	25	17.0
X_8	10	8.5
X_9	5	7.0
X_{10}	7	17.0
X_{11}	8	8.5
X_{12}	7	17.0
X_{13}	8	8.5
X_{14}	8	8.5

Tabela 5.126: Variáveis de projeto obtidas na otimização.

e os cromossomos apresentaram o mesmo número de variáveis e o mesmo comprimento. Também, foram consideradas poucas variáveis mas com qualidade elevada de informações.

Como no caso da otimização da estrutura de uma ponte, mostrado anteriormente neste texto, também ocorreu um processo de morfogênese no mapeamento de genótipo no espaço de fenótipos, onde cromossomos de comprimentos iguais foram capazes de gerar estruturas diferentes competindo entre si em uma única população.

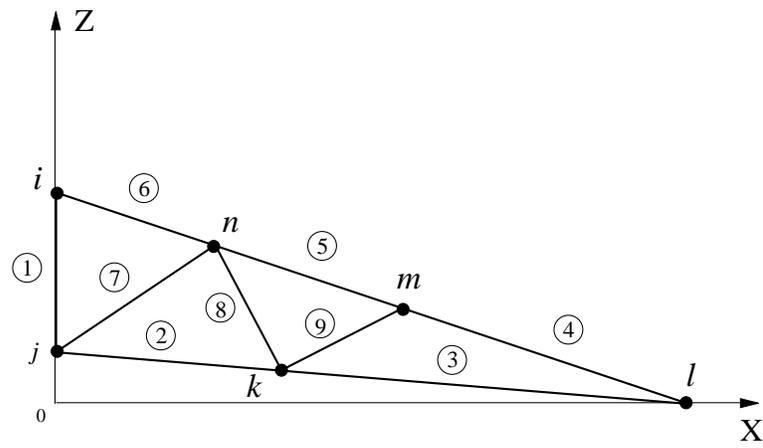


Figura 5.78: Forma final do módulo padrão.

Este tipo de técnica pode ser abstraída para vários problemas, estruturais ou não, sendo de grande interesse no sentido de evolução de formas.

Capítulo 6

Conclusões

Problemas de otimização estrutural foram abordados neste trabalho através de algoritmos genéticos.

Foram estudados problemas de minimização de peso, minimização de trabalho de deformação (*compliance*) e reação máxima de apoio, otimização de parâmetros e otimização de topologia na busca da melhor concepção de projetos estruturais.

Foram utilizados dois tipos de algoritmos genéticos: um do tipo “steady-state” com codificação real e outro do tipo “geracional” com codificação binária.

Mostrou-se que os algoritmos genéticos apresentam grande flexibilidade para avaliar problemas variados com diversos objetivos. Isto se reflete na facilidade de implementação, que resulta em pequenas alterações nos módulos, responsáveis pela obtenção da função aptidão das soluções candidatas.

Para grande parte dos problemas estudados aqui não houve necessidade de um código computacional exclusivo para cada um deles e sim mudanças específicas nos módulos responsáveis pela avaliação da função objetivo.

Exemplos clássicos foram discutidos neste texto, salientando-se o controle de violações das restrições. Mostrou-se pelos resultados obtidos nas análises a robustez dos algoritmos genéticos. Quase em sua totalidade, as soluções encontradas são melhores ou muito próximas das melhores encontradas na literatura. As funções de penalização, salvo em alguns casos, foram definidas através de parâmetros de penalização constantes durante toda a evolução. Todavia, foram testados para alguns problemas, mas não incluídas neste texto pelo fato de não serem representativas, simulações com vários tipos de estratégias, como as citadas no Capítulo 4. Estas estratégias mostraram-se extremamente sensíveis e de difícil definição da parametrização envolvida.

Técnicas Lamarckianas foram utilizadas, probabilisticamente, através de várias simulações, mostrando-se que nem sempre a recodificação total do cromossomo representa uma convergência mais rápida para a solução ótima.

Na grande maioria dos casos estudados nenhum tipo de informação sobre o problema foi introduzida no algoritmo e a busca baseou-se somente em valores da função aptidão. Para esta, nos casos onde haviam restrições, não houve para o algoritmo a identificação de soluções factíveis ou não. Isto porque os problemas com restrições foram transformados em problemas sem restrições pela introdução de funções de penalização. Entretanto, em alguns casos, o algoritmo foi informado sobre as características do problema, isto é, forneceram-se características do espaço de busca, o que certamente facilitou e acelerou a busca, pois soluções infactíveis ou anti-econômicas

não foram geradas nem avaliadas. Para isso, processos de morfogênese foram adotados para facilitar a codificação de soluções candidatas proporcionando cromossomos de tamanhos reduzidos e, conseqüentemente, facilitando a busca.

Comparativamente aos métodos clássicos nota-se que estes quando aplicáveis, apresentam em geral, um menor custo computacional no que tange ao número de avaliações. Entretanto, o que pôde ser observado nas análises comparativas mostradas neste texto é que, apesar de apresentarem um número elevado de avaliações da função objetivo, os algoritmos genéticos mostram-se robustos e eficazes na busca de soluções ótimas. Nota-se pelos gráficos que representam processos evolutivos em todos os exemplos dessa tese, que regiões com boas soluções foram encontradas rapidamente e foram gastos maiores tempos no refinamento das soluções. A incorporação de um outro tipo de algoritmo, por uma hibridização, pode diminuir esse tempo. Nenhuma medida de esforço computacional foi discutida pois, com os recursos computacionais disponíveis, as análises foram rápidas tornando-se o tempo de processamento secundário em relação à qualidade das soluções obtidas.

As principais contribuições deste trabalho podem ser resumidas em:

- Utilização de conhecimentos específicos do problema em questão, além das restrições naturais, como informação adicionada ao algoritmo genético para tornar a busca mais eficiente;
- Solução de problemas com variáveis contínuas e discretas tanto através de algoritmos genéticos com codificação binária quanto com codificação real;
- Foram propostas estratégias de co-evolução de parâmetros de solução de sistemas de equações lineares do algoritmo iterativo de Gauss-Seidel com relaxação. O parâmetro de relaxação do algoritmo é um dos elementos que participa do processo de evolução. Concluiu-se que o algoritmo genético destinado ao processo de co-evolução foi eficiente mostrando-se capaz de resolver o problema. Mostrou-se três estratégias para análise de aspectos referentes a espaço de busca com “cenário de aptidão móvel” e à perda de diversidade da população. As estratégias foram eficazes sendo extensíveis a problemas de diversas naturezas;
- Desenvolvimento e implementação de codificação para solução de um problema de grande interesse prático em engenharia e outras áreas. Isto é, dada a disponibilidade de peças estruturais pré-fabricadas no mercado, escolher, entre elas, o sub-conjunto que minimize o custo total do projeto.
- Apresentação da potencialidade de geração de topologias variáveis de uma estrutura, ou seja, capacidade de gerar formas que atendam a um determinado requisito de projeto, minimizando-se o seu custo.
- Discussão de exemplos clássicos de otimização estrutural mostrando a eficácia dos algoritmos genéticos na obtenção de melhores soluções, discutindo-se a factibilidade de soluções encontrados na literatura;

Como desenvolvimentos futuros podem ser considerados os seguintes aspectos:

- Implementação de algoritmos genéticos paralelos fazendo-se estudos comparativos com os problemas estudados aqui e análise de novos problemas;

- Implementação de técnicas de hibridização através de estratégias de busca local e/ou acoplamento de outros algoritmos ao algoritmo genético;
- Implementação de procedimentos para análise de otimização multi-objetivos;
- Estudo de problemas de otimização estrutural considerando-se estruturas planas (estado plano de deformações, estado plano de tensões, placas, etc);
- Estudo de estratégias alternativas para tratamento de problemas com restrições;
- Aprofundamento nas pesquisas sobre estratégias que objetivam evolução de formas em projetos estruturais;
- Estudo de convergência e critérios de encerramento do algoritmo genético.

Bibliografia

- [1] R. Dawkins. *Escalando o Monte Improvável*. Companhia das Letras, 1996.
- [2] L. Davis. *Handbook of Genetic Algorithms*. International Thomson Computer Press, 1996.
- [3] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [4] J.H. Holland and J.S. Reitman. Cognitive systems based on adaptive algorithms. In D.A. Waterman and F. Hayes-Roth, editors, *Pattern-Directed Inference Systems*. New York: Academic, 1978.
- [5] Goldberg D.E. Booker L.B. and Holland J.H. Classifier systems and genetic algorithms. In J.G. Carbonell, editor, *Machine Learning: Paradigms and Methods*, pages 235–282. MIT Press/Elsevier, 1989.
- [6] J.H. Holland. *How Adaptation builds Complexity*. Addison-Wesley Pub. Co., 1995.
- [7] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., 1989. Reading, Mass.
- [8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1996.
- [9] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary aerospace design optimization: Survey of recent developments. Technical Report 96-0711, American Institute of Aeronautics and Astronautics-AIAA, 1996.
- [10] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, 3(2):221–248, 1995.
- [11] D.E. Goldberg and M.P. Samtani. Engineering optimization via genetic algorithms. In *9th Conf. Electronic Computation*, pages 471–482, New York, N.Y., 1986.
- [12] K. Deb. Optimal design of a class of welded structures via genetic algorithms. In *31st AIAA/ASME, ASCE, ASH, ASC Structures, Structural Dynamics and Materials Conf.*, pages 444–453, 1990.

- [13] D.E. Goldberg; K. Zakrzewski; C. Chang; P. Gallego; B. Sutton; B.L. Miller and E. Cantú-Paz. Genetic algorithms: A bibliography. Technical Report IlliGAL No. 97002, Illinois Genetic Algorithms Laboratory - University of Illinois at Urbana-Champaign, 1997.
- [14] I.C. Parmee. The genetic algorithm and civil engineering design. In *Proceedings of Informing Technologies for Construction, Civil Engineering and Transport*, 1993.
- [15] I.C. Parmee. The implementation of adaptive search tools to promote global search in engineering design. In *Proceedings of 3rd IFIP Working Conference on Optimisation - Based Computer - Aided Modelling and Design*, 1994.
- [16] I.C. Parmee. Diverse evolutionary search for preliminary whole-system design. In *Proceedings of 4th Int. Conf. in Civil & Structural Engineering*. Civil-Comp. Press, 1995.
- [17] P. Husbands; G. Jermy; M. McIlhagga and R. Ives. Two applications of genetic algorithms to component design. In T. Fogarty, editor, *Evolutionary Computing Selected papers from AISB Workshop on Evolutionary Computing*, pages 50–61. Springer-Verlag, 1996.
- [18] I Husbands, P; Harvey and D. Cliff. Genetic convergence in a species of evolved robot control architectures. Technical Report CSRP 267, School of Cognitive and Computing Sciences , The University of Sussex, England, U.K., 1996.
- [19] J.S. Gero and J.C. Damski. A symbolic model for graphical emergence. *Environment and Planning B: Planning and Design*, (24):509–526, 1997.
- [20] W. Wang and J.S. Gero. Sequence-based prediction in the conceptual design of bridges. *Journal of Computing in Civil Engineering, ASCE*, 1(11):37–43, 1997.
- [21] J.H. Jo and J.S. Gero. Space layout planning using an evolutionary approach. *Artificial Intelligence in Engineering*, 3(12):149–162, 1998.
- [22] K.-J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, Inc., Eaglewood Cliffs, New Jersey, 1982.
- [23] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, Inc., Eaglewood Cliffs, New Jersey, 1987.
- [24] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. McGraw-Hill, New York, 1989. Vol. 1: Basic Formulation and Linear Problems.
- [25] R. Poli and B. Logan. On the relations between search and evolutionary algorithms. Technical Report CSRP-96-97, School of Computer Science, The University of Birmingham, UK, 1996.
- [26] H. Kargupta and D.E. Goldberg. Blackbox optimization: Implications of SEARCH. Technical Report LAUR-96-63, Los Alamos National Laboratory, 1996.

- [27] M.S. Eldred. Optimization strategies for complex engineering applications. Technical Report SAN98-0340, Sandia National Laboratories, 1998.
- [28] L. Spunt. *Optimum Structural Design*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
- [29] X. Liu and J.L. Susshil. Combining genetic algorithms and case-based reasoning for structures design. In *Proceedings of The Eleventh International Conference on Computers and Their Applications*, pages 103–106, Delft University of Technology, 1996. ISCA Press.
- [30] A.C.C. Lemonge and H.J.C. Barbosa. A genetic algorithm for optimal bridge pillar location. In *Fourth World Congress on Computational Mechanics*, Buenos Aires, Argentina, June 29 - July 2, 1998. In CD-ROM.
- [31] R.M. Stark and R.L. Nicholls. *Mathematical Foundations for Design in Civil Engineering Systems*. McGraw-Hill Book Company, New York, 1972.
- [32] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1984. Reading, Massachusetts.
- [33] Y.M. Xie and G.P. Steven. *Evolutionary Structural Optimization*. Springer-Verlag, 1997.
- [34] J. Heyman. Plastic design of beams and frames for minimum material consumption. *Q. Appl. Math.*, 8:373–381, 1956.
- [35] L.A. Schimidt. Structural design by systematic synthesis. In *Proceedings of the 2nd ASCE Conference on Electronic Computation*, pages 105–132, New York, N.Y., 1960.
- [36] O.C. Zienkiewicz and R.H. Gallagher. *Optimal Structural Design-Theory and Applications*. John Wiley and Sons, London, 1973.
- [37] J.J. Jarvis M.S. Bazaraa and H.D. Sherali. *Linear Programming and Network Flows*. John Wiley and Sons, 1990.
- [38] H.D. Sheradi M.S. Bazaraa and C.M. Shetty. *Nonlinear Programming - Theory and Algorithms*. John Wiley and Sons, Inc., 1993.
- [39] P.R. Adby and M.A.H. Dempster. *Introduction to Optimization Methods*. Chapman and Hall-England, 1974.
- [40] R.L. Fox. *Optimization Methods for Engineering Design*. Addison-Wesley Publishing Company, 1973.
- [41] D.M. Himmelblau. *Applied Nonlinear Programming*. McGraw-Hill Book Company, 1972.
- [42] D.G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, 1972. Reading, Massachusetts.
- [43] K.I. Majid. *Optimum Design of Structures*. Newnes-Butterworths, 1974.

- [44] B.H.V. Topping and M. Papadrakakis, editors. *Advances in Structural Optimization*, Athens, Greece, 1994. Civil-Comp Press. The Second International Conference on Computational Structures Technology.
- [45] D.M. Frangopol and F.Y. Cheng, editors. *Advances in Structural Optimization – Proceedings of The First U.S. – Japan Joint Seminar on Structural Optimization held in conjunction with the ASCE Technical Committee on Optimal Structural Design Meeting at the Structures Congress XIV*, Chicago, Illinois, 1996. ASCE American Society of Civil Engineers.
- [46] Artificial Intelligence Site Genetic Algorithms. Brunel University - UK. <http://http1.brunel.ac.uk:8080/depts/AI/alife/ga.htm>.
- [47] U. Hammel T. Bäck and H.-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.
- [48] I.C. Parmee and G.N. Bullock. Evolutionary techniques and their application to engineering design. In *Proceedings of EUROPIA93*, Delft University of Technology, 21-42 June 1993.
- [49] I. Rechenberg. Cybernetic solution path of an experimental problem. Technical Report 1122, Royal Aircraft Establishment, Farnborough, Hants, U.K., 1965.
- [50] H.-P. Schwefel. Projekt mhd-straustahlrohr: Experimentelle optimierung einer zweiphasendüse, teil i. Technical Report 11.034/68, AEG Forschungsinstitut, Berlin, Germany, 1968.
- [51] L. Fogel. Autonomous automata. *Ind. Res.*, 4:14–19, 1962.
- [52] K.A. De Jong. *Analysis of Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975. Dissertation Abstract International.
- [53] D.B. Fogel. An evolutionary approach to the traveling salesman problem. *Biological Cybern.*, 60:139–144, 1988.
- [54] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, Germany, 1973.
- [55] H.-P. Schwefel. *Evolutionsstrategie and Numerische Optimierung Dissertation*. PhD thesis, Technische Universität Berlin, Germany, 1973.
- [56] J.R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In N.S. Sridharan, editor, *Proc. 11th Int. Joint Conf. on Artificial Intelligence*, pages 768–774. Morgan Kaufmann, San Mateo, CA, 1989.
- [57] A.E. Eiben and Cees H.M. van Kemenade. Diagonal crossover in genetic algorithms for numerical optimization. *Journal of Control and Cybernetics*, 26(3):447–465, 1997.

- [58] M. Lozano F. Herrera and J.L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Technical Report DECSAI-95107, ETS de Ingeniería Informática, Universidad de Granada, 1995.
- [59] P.J. Angeline. Morfogenic evolutionary computations: Introduction, issues and examples. In J. McDonnell; B. Reynolds and D. Fogel, editors, *Fifth Annual Conference on Evolutionary Programming*. MIT Press, 1995.
- [60] M. Shoenauer and S. Xanthakis. Constrained GA optimization. In S. Forrest, editor, *Proceedings of The Fifth International Conference on Genetic Algorithms*, pages 573–580, Los Altos, CA, 1993. Morgan Kaufmann Publishers.
- [61] Z. Michalewicz. A survey of constraint handling techniques in evolutionary computation. In *Proceedings of The 4th International Conference on Evolutionary Programming*, pages 135–155, Cambridge, MA, 1995. MIT Press.
- [62] Z. Michalewicz and M. Shoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996.
- [63] Z. Michalewicz; D. Dasgupta; R.G. Le Riche and M. Shoenauer. Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 30(2):851–870, 1996.
- [64] H. Homaiifar; S.H.-Y. Lai and X. Qi. Constrained optimization via genetic algorithms. *Simulation*, 62(4):242–254, 1994.
- [65] J. Joines and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In Z. Michalewicz; J.D. Schaffer; H.-P. Schwefel; D.B. Fogel and H. Kitano, editors, *Proceedings of The First IEEE International Conference on Evolutionary Computation*, pages 579–584, June 19-23 1994.
- [66] J.C. Bean and A.B. Alouane. A dual genetic algorithm for bounded integer programs. Technical Report TR 92-53, Department of Industrial and Operations Engineering, The University of Michigan, 1992.
- [67] D.W. Coit; A.E. Smith and D.M. Tate. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing*, 6(2):173–182, 1996.
- [68] R.G. Le Riche; C. Knopf-Lenoir and R.T. Haftka. A segregated genetic algorithm for constrained structural optimization. In L.J. Eshelman, editor, *Proceedings of The Sixth International Conference on Genetic Algorithms*, pages 558–565, Pittsburgh, PA., July 1995.
- [69] M. Shoenauer and Z. Michalewicz. Boundary operators for constrained parameter optimization problems. In *Proceedings of The 7th International Conference on Genetic Algorithms*, pages 320–329, East Lansing, Michigan, July 19-23 1997.

- [70] R. Hinterding and Z. Michalewicz. Your brains and my beauty: Parent matching for constrained optimization. In *Proceedings of The Fifty International Conference on Evolutionary Computation*, pages 810–815, Alaska, May 4-9 1998.
- [71] S. Koziel and Z. Michalewicz. A decoder-based evolutionary algorithm for constrained optimization problems. In T. Bäck; A.E. Eiben; M. Shoenauer and H.-P. Schwefel, editors, *Proceedings of The Fifth Parallel Problem Solving from Nature*, Amsterdam, September 27-30 1998. Springer Verlag. Lecture Notes in Computer Science.
- [72] J.-H. Kim and H. Myung. Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 2(1):129–140, 1997.
- [73] R. Hinterding; Z. Michalewicz and A.E. Eiben. Adaptation in evolutionary computation: A survey. In *Proceedings of The 4th IEEE International Conference on Evolutionary Computation*, pages 65–69, Indianapolis, 1997. Morgan Kaufmann.
- [74] P.J. Angeline. Adaptive and self-adaptive evolutionary computation. In M. Palaniswami; Y. Attikiouzel; R.J.H. Marks; D. Fogel and T. Fukuda, editors, *Computational Intelligence, A Dynamic System Perspective*, pages 152–161. IEEE Press, 1995.
- [75] J.J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.
- [76] A.M. Sá e H.J.C. Barbosa. Adaptação das probabilidades dos operadores em algoritmos genéticos. In *Anais da I Jornada de Iniciação Científica do LNCC-Laboratório Nacional de Computação Científica*, pages 62–75, Petrópolis, Setembro 1998.
- [77] F.G. Lobo and D.E. Goldberg. Decision making in a hybrid genetic algorithm. Technical Report IlliGAL Report No. 96009, Department of General Engineering University of Illinois at Urbana-Champaign, 1996.
- [78] L. Davis. Adapting operator probabilities in genetic algorithms. In J.D. Schaffer., editor, *Proceedings of The Third International Conference on Genetic Algorithms and Their Applications*, pages 61–69, San Mateo, CA., June 1989. Morgan Kaufmann.
- [79] B.A. Julstron. What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In L.J. Eshelman., editor, *Proceedings of The Sixth International Conference on Genetic Algorithms and Their Applications*, pages 81–87, Pittsburgh, PA., July 1995.
- [80] B.A. Julstron. An inquiry into the behavior of adaptive operator probabilities in a steady-state genetic algorithms. In *Proceedings of The Second Nordic Workshop on Genetic Algorithms and Their Applications (2NWGA)*, pages 15–26, 1996. Available by ftp to `ftp.uwasa.fi`, directory `cs/2NWGA`.

- [81] B.A. Julstron. Adaptive operator probabilities in a genetic algorithm that applies three operators. In B. Bryant; J. Carrol; D. Oppenheim; J. Hightower and K.M. George., editors, *Applied Computing 1997: Proceedings of The 1997 ACM Symposium on Applied Computing*, pages 233–238, New York, 1997.
- [82] J.D. Schaffer and A. Morishima. An adaptive crossover distribution mechanism for genetic algorithms. In J.J. Grefenstette, editor, *Proceedings of The Second International Conference on Genetic Algorithms*, pages 36–40, New Jersey, 1987. Lawrence Erlbaum Associates.
- [83] J. Arabas; Z. Michalewicz and J. Mulawka. Gavaps - a genetic algorithm with varying population size. In *Proceedings of The First IEEE Conference on Evolutionary Computation*, pages 73–78. IEEE Press, 1994.
- [84] F. Herrera and M. Lozano. *Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers*, pages 95–125. Physica-Verlag, 1996.
- [85] S.J. Louis and G.J.E Rawlins. Syntatic analysis of convergence in genetic algorithms. In L.D. Whitley, editor, *Foundations of Genetic Algorithms-2*, pages 141–151, San Mateo, 1993. Morgan Kaufmann Publishers.
- [86] D. Whitley; K. Mathias and P. Fitzhorn. Delta coding: an iterative search strategy for genetic algorithms. In R. Belew and L.B. Booker, editors, *Proceedings of The Fourth International Conference on Genetic Algorithms*, pages 77–84, San Mateo, 1991. Morgan Kaufmann Publishers.
- [87] M.A. Lee and H. Takagi. Dynamic control of genetic algorithms using fuzzy logic techniques. In S. Forrest, editor, *Proceedings of The Fifth International Conference on Genetic Algorithms*, pages 76–83, San Mateo, 1993. Morgan Kaufmann Publishers.
- [88] M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 4(24):656–667, 1994.
- [89] W.D. Potter; C.N. Lapena, J.A. Miller and R.V. Gandhman. An evaluation of local improvement operators for genetic algorithms. *IEEE-Transactions on Systems, Man, and Cybernetics*, 23(5):1340–1351, 1993.
- [90] P. Hajela and J. Yoo. Constraint handling in genetic search using expression strategies. *AIAA Journal*, 34(12):2414–2420, 1996.
- [91] W.E. Hart; T.E. Kammeyer and R.K. Belew. The role of development in genetic algorithms. Technical Report CS94-394, Computer Science and Engineering, University of California, San Diego, 1994.
- [92] P. Turney. Myths and legends of the Baldwin effect. In *Proceedings of The Workshop on Evolutionary Computing and Machine Learning, at The 13th International Conference on Machine Learning (ICML-96)*, pages 135–142, Bari, Italy, July, 3-6 1996. Morgan Kaufmann Publishers.

- [93] E. Cantú-Paz. A survey of parallel genetic algorithms. Technical Report 97003, IlliGAL-Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign., 1997.
- [94] E. Cantú-Paz. Design efficient master-slave parallel genetic algorithms. Technical Report 97004, IlliGAL-Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign., 1997.
- [95] G. Harik; E. Cantú-Paz; D.E. Goldberg and B. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. In T. Bäck, editor, *Proceedings of the Fourth International Conference on Evolutionary Computation*, pages 7–12, New York, 1997. IEEE Press.
- [96] E. Cantú-Paz and D.E. Goldberg. Modeling idealized bounding cases of parallel genetic algorithms. In J. Koza; K. Deb; M. Dorigo; M. Fogel; M. Garzon; H. Iba and R. Riolo., editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 353–361, San Francisco, CA., 1997. Morgan Kaufmann.
- [97] E. Cantú-Paz and D.E. Goldberg. Modeling speedups of idealized bounding cases of parallel genetic algorithms. In T. Bäck., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms.*, San Mateo, CA., 1997. Morgan Kaufmann.
- [98] J.N.C. Guerreiro; H.J.C. Barbosa; E.L.M. Garcia; A.F.D. Loula and S.M.C. Malta. Identification of reservoir heterogeneities using tracer breakthrough profiles and genetic algorithms. In *V LACPEC (Fifth Latin American and Caribbean Petroleum Engineering Conference)*, Rio de Janeiro, 1997. August.
- [99] H. Adeli and O. Kamal. Concurrent optimization of large structures. ii: Applications. *Journal of Aerospace Engineering*, 5(1):91–110, 1992.
- [100] H. Adeli and N-T. Cheng. Concurrent genetic algorithms for optimization of large structures. *Journal of Aerospace Engineering*, 7(3):276–296, 1994.
- [101] H. Adeli and S. Kumar. Distributed genetic algorithm for structural optimization. *Journal of Aerospace Engineering*, 8(3):156–163, 1995.
- [102] S.L. Burgee; A.A. Giunta; V. Balabanov; B. Grossman; W.H. Mason; R. Narducci; R.T. Haftka and L. T. Watson. A coarse grained parallel variable-complexity multidisciplinary optimization paradigm. Technical Report TR 95-20, Department of Computer Science, Virginia Polytechnic Institute and State University, 1995.
- [103] H.J.C. Barbosa and A.C.C. Lemonge. A genetic algorithm for constrained discrete optimization of framed structures. In *XVIII Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia*, pages 1491–1498, October 1997. Brasília.
- [104] A.C.C. Lemonge and H.J.C. Barbosa. A genetic algorithm for shape and sizing optimization of truss structures. In *III Simpósio Mineiro de Mecânica Computacional*, pages 269–277, Ouro Preto, Minas Gerais, November 19-21, 1998.

- [105] A.C.C. Lemonge e H.J.C. Barbosa. Otimização estrutural via algoritmos genéticos. In E.M. Toledo; H.J.C. Barbosa; A.C.C. Lemonge e L.P.S. Barra, editor, *Anais do V Seminário sobre Elementos Finitos e Métodos Numéricos em Engenharia*, pages 55–77, Fac. Engenharia da Univ. Federal de Juiz de Fora, 3 e 4 Dezembro 1998.
- [106] R.A. Gellatly and L. Berke. Optimal structural design. Technical Report AFFDL-TR-70-165, Air Force Flight Dynamics Lab., AFFDL, 1971.
- [107] L.A. Schimit and H. Miura. Approximation concepts for efficient structural synthesis. Technical Report CR-2552, NASA, 1976.
- [108] L.A. Schimit and B. Farshi. Some approximation concepts in structural synthesis. *AIAA Journal*, 12:692–699, 1974.
- [109] V. B. Venkayya. Design of optimum structures. *Journal of Computers & Structures*, 1(1-2):265–309, 1971.
- [110] M.V. Dobbs and R.B. Nelson. Application of optimality criteria to automated structural design. *AIAA Journal*, 14:1436–1443, 1976.
- [111] P Rizzi. Optimization of multiconstrained structures based on optimality criteria. In *Proceedings of 17th Structures, Structural Dynamics, and Materials Conference*, King of Prussia, Pa., 1976.
- [112] K.D. Willmert M.R. Khan and W.A. Thornton. An optimality criterion method for large-scale structures. *AIAA Journal*, 17(7):753–761, 1979.
- [113] E.J. Haug and J.S. Arora. *Applied Optimal Design*. John Willey, New York, 1979.
- [114] R.T. Haftka and M.P. Kamat. *Elements of Structural Optimization*. Martinus Nijhoff, 1985.
- [115] A.M. Memari and A. Fuladgar. Minimum weight design of trusses by BEHSAZ program. In B.H.V. Topping and M. Papadrakakis, editors, *Advances in Structural Optimization*, pages 179–185, Athens, Greece, 1994. Civil-Comp Press. The Second International Conference on Computational Structures Technology.
- [116] H. Adeli and O. Kamal. Efficient optimization of plane trusses. *Adv. Eng. Software*, 13(3):116–122, 1991.
- [117] M.E. El-Sayed and T.S. Jang. Structural optimization using unconstrained nonlinear goal programming algorithm. *Computers & Structures*, 52(4):723–727, 1994.
- [118] M. Galante. Structures optimization by a simple genetic algorithm. In *Numerical Methods in Engineering and Applied Sciences*, pages 862–870, 1992.
- [119] L.A. Schimit and H. Miura. A new structural analysis/synthesis capability: Access 1. *AIAA Journal*, 14:661–671, 1976.

- [120] Ting-Yu Chen and Chung-Jei Chen. Improvements of simple genetic algorithm in structural design. *International Journal for Numerical Methods in Engineering*, 40:1323–1334, 1997.
- [121] C.S. Krishnamoorthy and S. Rajeev. Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*, 118(5), 1992.
- [122] M. Galante. Genetic algorithms as an approach to optimize real-world trusses. *International Journal for Numerical Methods in Engineering*, 39:361–382, 1996.
- [123] A.B. Templeman. Discrete optimum structural design. *Computer & Structures*, 30(3):511–518, 1988.
- [124] S.J. Wu and P.T. Chow. Steady-state genetic algorithms for discrete optimization of trusses. *Computers & Structures*, 56(6):979–991, 1995.
- [125] S.R Hansen and G.N. Vanderplaats. Approximation method for configuration optimization of trusses. *AIAA Journal*, 28(1):161–168, 1990.
- [126] E. Salajegheh and G.N. Vanderplaats. Optimum design of trusses with discrete sizing and shape variables. *Structural Optimization*, 6:79–85, 1993.
- [127] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization. In Stephanie Forrest, editor, *Proceedings of The Fifth International Conference On Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993. Morgan Kaufman.
- [128] ABNT Associação Brasileira de Normas Técnicas. Projeto e execução de obras de concreto armado. Technical report, ABNT, Rio de Janeiro - Brasil, 1982.
- [129] ABNT Associação Brasileira de Normas Técnicas. Cálculo e execução de pontes de concreto armado. Technical report, ABNT, Rio de Janeiro - Brasil, 1987.
- [130] H.J.C. Barbosa. A coevolutionary genetic algorithm for a game approach to structural optimization. In Th. Baeck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms and their Applications*, pages 545–552, East Lansing, MI, July 1997. Morgan Kaufman.