

OTIMIZAÇÃO DE ESTRUTURAS COM MULTI-OBJETIVOS
VIA ALGORITMOS GENÉTICOS

Rodrigo Evangelista de Castro

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 Gayoso de Azeredo Coutinho, D.Sc.

Prof. Helio José Corrêa Barbosa, D.Sc.

Prof. Ronaldo Carvalho Battista, Ph.D.

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. Elson Magalhães Toledo, D.Sc.

Prof. Afonso Celso de Castro Lemonge, D.Sc.

RIO DE JANEIRO, R.J. – BRASIL

AGOSTO DE 2001

CASTRO, RODRIGO EVANGELISTA DE

Otimização de Estruturas com Multi-objetivos
Via Algoritmos Genéticos [Rio de Janeiro] 2001

XX, 206 p. 29,7 cm (COPPE/UFRJ, D.Sc.,
Engenharia Civil, 2001)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1. Otimização Multi-objetivos
2. Algoritmos Genéticos
3. Otimização Estrutural

I. COPPE/UFRJ II. Título (série)

Aos meus queridos pais,
Sebastião Leme e Carmen Lucia,
em gratidão por tamanho e contínuo apoio no dia-a-dia
do meu viver e no conquistar de novos horizontes ...

Agradecimentos

Desejo expressar minha sincera e profunda gratidão para as seguintes pessoas e instituições que contribuíram para a realização deste trabalho:

- Professores Helio J. C. Barbosa e Alvaro L. G. A. Coutinho por suas valiosas orientações, pela dedicação e incentivos permanentes, tanto no âmbito profissional como pessoal.
- Professor Luis Paulo S. Barra pelos dois anos de orientação acadêmica em nível de iniciação científica junto ao Departamento de Estruturas da Universidade Federal de Juiz de Fora.
- Professores, funcionários e amigos da UFJF, COPPE/UFRJ e LNCC, pelo apoio oferecido e convívio gratificante.
- CNPq, COPPE/UFRJ e LNCC, que em conjunto possibilitaram a realização deste trabalho.
- Todos os meus familiares e amigos pessoais.
- Companheiros de turma de mestrado e doutorado pela sincera e duradoura amizade estabelecida.

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

OTIMIZAÇÃO DE ESTRUTURAS COM MULTI-OBJETIVOS VIA ALGORITMOS GENÉTICOS

Rodrigo Evangelista de Castro

Agosto/2001

Orientadores: Alvaro Luiz Gayoso de Azeredo Coutinho

Helio José Corrêa Barbosa

Programa: Engenharia Civil

Este trabalho apresenta uma revisão geral sobre a técnica dos algoritmos genéticos – fundamentos, operadores, parâmetros de configuração, vantagens e desvantagens – e de sua aplicação aos problemas de otimização multi-objetivos.

É desenvolvido um algoritmo genético para otimização de problemas multi-objetivos cuja finalidade é evoluir um conjunto uniformemente distribuído de soluções para determinar o conjunto ótimo de Pareto do problema tratado.

As principais características do algoritmo proposto são: (i) ordenamento da população de acordo com propriedades de dominância, (ii) utilização de operadores multi-objetivos adequadamente desenvolvidos: exclusão, adição e operador de otimização individual, (iii) definição de um filtro para reter as soluções de Pareto, (iv) elitismo deste filtro dentro das populações e (v) método de penalização por camadas.

Para ilustrar o desempenho do algoritmo desenvolvido são apresentados exemplos numéricos de problemas testes e estruturais multi-objetivos: viga soldada, treliça tri-objetiva, ponte treliçada, torre de 47 barras, dentre outros.

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

OPTIMIZATION OF STRUCTURES WITH MULTIOBJECTIVES
BY GENETIC ALGORITHMS

Rodrigo Evangelista de Castro

August/2001

Advisors: Alvaro Luiz Gayoso de Azeredo Coutinho

Helio José Corrêa Barbosa

Department: Civil Engineering

This work presents a general review on genetic algorithms – foundations, operators, configuration parameters, advantages and disadvantages – and their applications to multiobjective optimization problems.

A special genetic algorithm is developed for multiobjective optimization problems. Its purpose is to evolve an evenly distributed group of solutions to determine the optimum Pareto set for a given problem.

The main characteristics of the proposed algorithm are: (i) population sorting by its domination properties, (ii) use of multiobjective operators appropriately developed: exclusion, addition and individual optimization operator, (iii) definition of a filter to retain the Pareto solutions, (iv) elitism of this filter inside of the populations and (v) a method of penalization by layers.

To illustrate the use of the developed optimization algorithm some numeric examples of multiobjective test problems and multiobjective structural problems are presented: welded beam, three-objective truss, truss bridge, 47-bar truss, among others.

Índice

Agradecimentos	iv
Resumo.....	vii
Abstract.....	vii
Índice.....	vii
Índice de Figuras	xiii
Índice de Tabelas	xix

Capítulo I : Introdução.....	1
1. Motivação.....	2
2. Otimização	4
3. Programação Matemática.....	5
4. Computação Natural	6
4.1. Fractais	7
4.2. Recozimento Simulado.....	8
4.3. Lógica Nebulosa	8
4.4. Redes Neurais Artificiais.....	9
4.5. Computação Evolucionária.....	9
4.5.1. Algoritmos Genéticos (AGs)	12
4.5.2. Programação Evolucionária ou Evolutiva (PE)	12
4.5.3. Estratégias Evolucionárias ou de Evolução (EE).....	12
4.5.4. Programação Genética (PG)	13
4.5.5. Estratégia de Colônia.....	13
5. Introdução ao Código Genético.....	14
6. Organização da Tese.....	16

Capítulo II : Otimização.....	17
1. Formulação.....	18
2. Definições	18
3. Fraqueza dos Métodos Clássicos.....	20
4. Classificação dos Métodos Clássicos	20
 Capítulo III: Otimização Multi-objetivos.....	 22
1. Introdução	23
2. Definições	25
 Capítulo IV : Algoritmos Genéticos.....	 29
1. Introdução	30
2. Histórico.....	33
3. Definições	38
4. Diferenças entre os Algoritmos Genéticos e os Métodos Clássicos.....	39
5. Estrutura dos Algoritmos Genéticos.....	41
6. Aspectos Principais dos Algoritmos Genéticos.....	44
6.1. Representação e Codificação	44
6.2. Geração da População Inicial	45
6.3. Avaliação da População	45
6.4. Esquemas de Seleção.....	46
6.4.1. Seleção Proporcional à Aptidão via Método da Roleta.....	47
6.4.2. Seleção Estocástica Remanescente sem Substituição	49
6.4.3. Seleção por Ordenação: Linear e Exponencial	49
6.4.4. Seleção por Torneio.....	51
6.4.5. Seleção Elitista	52
6.5. Reprodução ou Cruzamento.....	53
6.6. Operadores Genéticos.....	53
6.6.1. Operadores de Recombinação.....	54
6.6.2. Operadores de Mutação	55

7. Parâmetros de Influência e Configurações.....	55
7.1. Tamanho da População - N	56
7.2. Taxa ou Probabilidade de Cruzamento - P_c	56
7.3. Taxa ou Probabilidade de Mutação - P_m	57
7.4. Recomendações de Autores.....	58
8. Vantagens e Desvantagens dos Algoritmos Genéticos.....	59
8.1. Vantagens dos Algoritmos Genéticos	59
8.2. Desvantagens dos Algoritmos Genéticos	59
9. Sumário e Discussão.....	60
 Capítulo V : Tópicos Adicionais em AGs	61
1. Tratamento das Restrições	62
2. Problemas com Múltiplos Objetivos	62
3. Hibridização	63
4. Computação Paralela	64
5. Time de Algoritmos.....	65
6. Adaptação.....	66
7. Distribuição Uniforme da População Inicial.....	66
 Capítulo VI : Algoritmos Genéticos Multi-objetivos.....	67
1. Introdução	68
2. Alguns Algoritmos Evolucionários Multi-objetivos	69
2.1. VEGA, Schaffer	69
2.2. Agregação dos objetivos por pesos variáveis, Hajela e Lin	71
2.3. MOGA, Fonseca e Fleming	71
2.4. nPGA, Horn e Nafpliotis	72
2.5. NSGA, Srinivas e Deb.....	73
2.6. SPEA, Zitzler e Theile.....	75
2.7. Outros algoritmos.....	76

Capítulo VII : PMOGA – Pareto Multiobjective GA.....	78
1. Introdução	79
2. Resumo e Evoluções do Algoritmo Evolucionista.....	79
2.1. Modificação do método de seleção	80
2.2. Modificações do operador de recombinação	80
2.3. Desenvolvimento e introdução dos operadores multi-objetivos.....	80
2.4. Introdução do elitismo do filtro nas populações	81
3. Algoritmo Final Detalhado	81
4. Filtro.....	82
5. Seleção	83
5.1. Ordenamento.....	84
6. Reprodução	84
6.1. Mutação	85
6.2. Recombinação.....	86
7. Operadores Multi-objetivos Desenvolvidos.....	86
7.1. Operador de Exclusão	86
7.2. Operador de Adição	87
7.3. Operador de Otimização Individual	88
8. Gerações Adicionais	88
9. Elitismo	89
10. Estratégia de Penalização	89
11. Resultados e Desempenho.....	90
 Capítulo VIII : Aplicações.....	 91
1. Introdução	92
2. Função f_2 de Schaffer	92
2.1. Introdução e Modelo	92
2.2. Características e Configurações do AG Utilizado	93
2.2.1. Atribuição de aptidão para as ordens	94
2.3. Resultados.....	95
2.4. Conclusões	96

3. Viga I	97
3.1. Introdução e Modelo	97
3.2. Características e Configurações do AG Utilizado	99
3.3. Resultados.....	100
3.4. Conclusões	102
4. Viga Soldada	103
4.1. Introdução e Modelo	103
4.2. Características e Configurações do AG Utilizado	105
4.3. Resultados.....	106
4.4. Conclusões	107
5. Trelça com Três Objetivos.....	108
5.1. Introdução e Modelo	108
5.2. Características e Configurações do AG Utilizado	109
5.3. Resultados.....	110
5.4. Conclusões	112
6. Ponte Trelçada.....	113
6.1. Introdução e Modelo	113
6.2. Características e Configurações do AG Utilizado	114
6.3. Resultados.....	115
6.4. Conclusões	118
7. Fronteira de Pareto Descontínua	119
7.1. Introdução e Modelo	119
7.2. Características e Configurações do AG Utilizado	119
7.3. Resultados.....	120
7.4. Conclusões	123
8. Funções-teste para Otimizadores Multi-objetivos.....	124
8.1. Introdução e Modelo	124
8.2. Metodologia de Comparação	129
8.3. Características e Configurações do AG Utilizado	129
8.4. Número de Avaliações	130
8.5. Resultados.....	131
8.6. Conclusões	151

9. Curva Tri-objetiva	153
9.1. Introdução e Modelo	153
9.2. Características e Configurações do AG Utilizado	154
9.3. Resultados.....	154
9.4. Conclusões	157
10. Esfera	158
10.1. Introdução e Modelo	158
10.2. Características e Configurações do AG Utilizado	159
10.3. Resultados.....	159
10.4. Conclusões	164
11. Cubo – 8 Funções Objetivo	165
11.1. Introdução e Modelo	165
11.2. Características e Configurações do AG Utilizado	166
11.3. Resultados.....	167
11.4. Conclusões	175
12. Trelça de 47 barras.....	176
12.1. Introdução e Modelo	176
12.2. Características e Configurações do AG Utilizado	179
12.3. Resultados.....	180
12.4. Conclusões	191
 Capítulo IX : Conclusões.....	193
1. Aspectos Gerais	194
2. Contribuições.....	194
3. Resultados	195
4. Experiências Adquiridas	195
5. Trabalhos Futuros	196
 Referências.....	197

Índice de Figuras

Capítulo I : Introdução.....	1
Figura 1 - Formigas em linha reta entre o ninho e a fonte de comida [23]	3
Figura 2 - Introdução de um obstáculo no percurso original [23].....	3
Figura 3 - Divisão mediana das formigas pelos dois novos percursos [23]	3
Figura 4 - O percurso menor (ótimo) é escolhido com o passar do tempo [23]	3
Figura 5 - Algumas Sub-áreas da Computação Natural	6
Figura 6 - Algoritmo Evolucionário conceitual	10
Figura 7 - Sub-áreas da Computação Evolucionária	11
 Capítulo II : Otimização.....	 17
Figura 8 - Classificação Geral dos Métodos Clássicos [53]	21
 Capítulo III : Otimização Multi-objetivos	 22
Figura 9 - Funções g e h de um problema multi-objetivo (f_2 - Schaffer [57])	27
Figura 10 - Identificação de uma fronteira de Pareto	28
 Capítulo IV : Algoritmos Genéticos.....	 29
Figura 11 - Aplicabilidade em Problemas x Eficiência de Resolução dos Métodos	32
Figura 12 - Algoritmo genético genérico.....	41
Figura 13 - Algoritmo genético geracional.....	42
Figura 14 - Algoritmo genético “em regime” – steady-state	43
Figura 15 - Exemplo do Método da Roleta.....	48
Figura 16 - Distribuição Linear para Seleção via Ordenamento.....	50
Figura 17 - Distribuição Exponencial para Seleção via Ordenamento.....	50
Figura 18 - Exemplo de Recombinação.....	54
Figura 19 - Exemplo de Mutação	55

Capítulo V : Tópicos Adicionais em AGs	61
Capítulo VI : Algoritmos Genéticos Multi-objetivos.....	67
Capítulo VII : PMOGA – Pareto Multiobjective GA.....	78
Figura 20 - Pseudocódigo do PMOGA.....	81
Capítulo VIII : Aplicações.....	91
Figura 21 - Funções objetivo e soluções no espaço das funções objetivo.....	93
Figura 22 - População inicial	95
Figura 23 - Fronteira de Pareto e soluções no espaço das variáveis de projeto....	95
Figura 24 - Fronteira de Pareto com novo método de seleção (torneio)	96
Figura 25 - Viga em seção I [15].....	97
Figura 26 - População inicial em diferentes escalas.....	100
Figura 27 - População relativa à geração 200 em diferentes escalas	100
Figura 28 - Soluções no filtro final (20 soluções).....	101
Figura 29 - Soluções no filtro final (ger. 500).	102
Figura 30 - Viga soldada [21].	103
Figura 31 - Soluções viáveis nas gerações inicial e centésima, respectivamente	106
Figura 32 - Soluções no filtro na geração 100: sem (esquerda) e com (direita) o uso dos operadores de adição e de otimização individual respectivamente.	106
Figura 33 - Soluções do filtro na geração 500 (100 + 400).	107
Figura 34 - Treliça com três objetivos.....	108
Figura 35 - Filtro no espaço $P \times u_1$	110
Figura 36 - Filtro no espaço $P \times u_2$.....	111
Figura 37 - Filtro no espaço $u_1 \times u_2$.....	111
Figura 38 - Filtro no espaço $P \times u_1 \times u_2$ (figura não disponível em [13]).....	112
Figura 39 - Ponte treliçada.....	113
Figura 40 - População inicial e da geração 100 respectivamente	116
Figura 41 - Soluções armazenadas no filtro final do processo de otimização	116
Figura 42 - Evolução das minimizações individuais dos objetivos	117
Figura 43 - Solução que apresentou o peso mínimo (f_1)	117

Figura 44 - Solução que apresentou o deslocamento vertical mínimo (f_2).....	117
Figura 45 - População inicial	120
Figura 46 - População na geração 20	121
Figura 47 - Filtro final na geração 100 (20+80).....	121
Figura 48 - População na geração 20	122
Figura 49 - Filtro final na geração 100 (20+80).....	122
Figura 50 - Exemplo de população inicial - τ_1	133
Figura 51 - Fronteiras de Pareto comparativas - τ_1 [72].....	133
Figura 52 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_1 [73].....	134
Figura 53 - Resultados de 5 execuções do algoritmo PMOGA - τ_1 [11].....	134
Figura 54 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_1 [11, 73].....	135
Figura 55 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_1 [11, 73].....	135
Figura 56 - Exemplo de população inicial - τ_2	136
Figura 57 - Fronteiras de Pareto comparativas - τ_2 [72].....	136
Figura 58 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_2 [73].....	137
Figura 59 - Resultados de 5 execuções do algoritmo PMOGA - τ_2 [11].....	137
Figura 60 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_2 [11, 73].....	138
Figura 61 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_2 [11, 73].....	138
Figura 62 - Exemplo de população inicial - τ_3	139
Figura 63 - Fronteiras de Pareto comparativas - τ_3 [72].....	139
Figura 64 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_3 [73].....	140
Figura 65 - Resultados de 5 execuções do algoritmo PMOGA - τ_3 [11].....	140
Figura 66 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_3 [11, 73].....	141
Figura 67 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_3 [11, 73].....	141
Figura 68 - Exemplo de população inicial - τ_4	142

Figura 69 - Fronteiras de Pareto comparativas - τ_4 [72].....	142
Figura 70 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_4 [73].....	143
Figura 71 - Resultados de 5 execuções do algoritmo PMOGA - τ_4 [11].....	143
Figura 72 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_4 [11, 73].....	144
Figura 73 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_4 [11, 73].....	144
Figura 74 - Exemplo de população inicial - τ_5	145
Figura 75 - Fronteiras de Pareto comparativas - τ_5 [72].....	145
Figura 76 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_5 [73].....	146
Figura 77 - Resultados de 5 execuções do algoritmo PMOGA - τ_5 [11].....	146
Figura 78 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_5 [11, 73].....	147
Figura 79 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_5 [11, 73].....	147
Figura 80 - Exemplo de população inicial - τ_6	148
Figura 81 - Fronteiras de Pareto comparativas - τ_6 [72].....	148
Figura 82 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_6 [73].....	149
Figura 83 - Resultados de 5 execuções do algoritmo PMOGA - τ_6 [11].....	149
Figura 84 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_6 [11, 73].....	150
Figura 85 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_6 [11, 73].....	150
Figura 86 - Fronteira ótima de Pareto completa [66]	155
Figura 87 - Configuração 1 - 50 soluções no filtro	155
Figura 88 - Configuração 2 - 80 soluções no filtro	156
Figura 89 - Configuração 3 - 100 soluções no filtro	156
Figura 90 - População inicial – vistas superior e lateral.....	160
Figura 91 - População inicial – vista frontal.....	160
Figura 92 - Fronteira 100 x 200 x 200 – vistas superior e lateral	161
Figura 93 - Fronteira 100 x 200 x 200 – vista frontal	161

Figura 94 - Fronteira 100 x 200 x 200, sem operadores multi-objetivos – vistas superior e lateral	162
Figura 95 - Fronteira 100 x 200 x 200, sem operadores multi-objetivos – vista frontal.....	162
Figura 96 - Fronteira 300 x 500 x 200 – vistas superior e lateral	163
Figura 97 - Fronteira 300 x 500 x 200 – vista frontal	163
Figura 98 - Modelo espacial – problema do cubo	166
Figura 99 - População inicial (1) – vistas espaciais	167
Figura 100 - População inicial (1) – projeções planas	167
Figura 101 - Soluções encontradas (1) – vistas espaciais.....	168
Figura 102 - Soluções encontradas (1) – projeções planas.....	168
Figura 103 - Disposições possíveis dos objetivos combinados 2 a 2.....	168
Figura 104 - População inicial (2) – vistas espaciais	169
Figura 105 - População inicial (2) – projeções planas	169
Figura 106 - Soluções encontradas (2) – vistas espaciais.....	170
Figura 107 - Soluções encontradas (2) – vista espacial inclinada.....	170
Figura 108 - Soluções encontradas (2) – projeções planas.....	170
Figura 109 - Funções objetivo combinadas (2).....	171
Figura 110 - Funções objetivo combinadas (2).....	172
Figura 111 - Funções objetivo combinadas (2).....	173
Figura 112 - População inicial (3) – vistas espaciais	173
Figura 113 - População inicial (3) – projeções planas	174
Figura 114 - Soluções encontradas (3) – vistas espaciais.....	174
Figura 115 - Soluções encontradas (3) – projeções planas.....	174
Figura 116 - Treliça de referência com 47 barras [44].....	177
Figura 117 - População inicial	180
Figura 118 - População na geração 5000.....	180
Figura 119 - Fronteiras obtidas das 3 configurações utilizadas	181
Figura 120 - Minimização do peso durante as 30000 gerações	183
Figura 121 - Minimização do peso, em escala reduzida, durante as 200 primeiras gerações	183

Figura 122 - Minimização do peso, em escala reduzida, durante as 30000 gerações	183
Figura 123 - Forma da solução ótima da referência [44] e da solução 1 do filtro, respectivamente.....	185
Figura 124 - Formas das soluções 5 e 10 do filtro, respectivamente	186
Figura 125 - Formas das soluções 15 e 20 do filtro, respectivamente	186
Figura 126 - Filtro final – geração 5000 + 25000	190
Figura 127 - Filtro final orientado no espaço de busca	190
Figura 128 - Detalhe do filtro final orientado	190
 Capítulo IX : Conclusões	 193

Índice de Tabelas

Capítulo I : Introdução.....	1
Capítulo II : Otimização.....	17
Capítulo III : Otimização Multi-objetivos	22
Capítulo IV : Algoritmos Genéticos.....	29
Capítulo V : Tópicos Adicionais em AGs	61
Capítulo VI : Algoritmos Genéticos Multi-objetivos.....	67
Capítulo VII : PMOGA – Pareto Multiobjective GA.....	78
Capítulo VIII : Aplicações.....	91
1. Introdução	92
2. Função f_2 de Schaffer.....	92
3. Viga I	97
Tabela 1 - Valores das variáveis de projeto e das funções objetivo.	102
4. Viga Soldada	103
Tabela 2 - Valores das variáveis de projeto e das funções objetivo do filtro final.	107
5. Treliça com Três Objetivos.....	108
Tabela 3 - Valores numéricos de algumas soluções do filtro final armazenado	112

6. Ponte Treliçada.....	113
Tabela 4 - Relação entre as variáveis de projeto e áreas dos elementos estruturais	113
Tabela 5 - Conjunto de áreas consideradas (in ²).....	114
Tabela 6 - Valores numéricos de algumas soluções do filtro final armazenado	118
7. Fronteira de Pareto Descontínua	119
Tabela 7 - Valores da variável de projeto e das funções objetivo do filtro final	123
8. Funções-teste para Otimizadores Multi-objetivos.....	124
9. Curva Tri-objetiva	153
Tabela 8 - Configurações utilizadas do AG.....	154
10. Esfera	158
Tabela 9 - Configurações utilizadas do AG.....	159
Tabela 10 - Resultados numéricos encontrados – avaliações e afastamento..	164
11. Cubo – 8 Funções Objetivo	165
Tabela 11 - Configurações utilizadas do AG.....	166
12. Treliça de 47 barras.....	176
Tabela 12 - Dados de carregamento – cargas em lb	176
Tabela 13 - Propriedades do material e restrições	176
Tabela 14 - Nós fixos e semi-fixos – coordenadas em in.....	178
Tabela 15 - Variáveis de área	178
Tabela 16 - Variáveis de forma.....	179
Tabela 17 - Configurações utilizadas do AG.....	179
Tabela 18 - Valores das funções objetivo do filtro final	182
Tabela 19 - Valores das variáveis de projeto de algumas soluções do problema.....	184
Tabela 20 - Tensões normais da solução 1 do filtro	187
Tabela 21 - Tensões normais da solução 20 do filtro.....	188
Capítulo IX : Conclusões.....	193

Capítulo I

Introdução

Neste capítulo é realizada uma introdução sobre o trabalho desenvolvido, sendo apresentada a motivação para a escolha do tema abordado, os principais conceitos e fundamentos nos campos da otimização, programação matemática e computação natural, além da organização do trabalho.

1. Motivação.....	2
2. Otimização	4
3. Programação Matemática.....	5
4. Computação Natural	6
5. Introdução ao Código Genético.....	14
6. Organização da Tese.....	16

1. Motivação

A motivação para trabalhar no estado da arte da otimização de estruturas com a utilização de algoritmos genéticos originou-se do conhecimento de fatos curiosos e interessantes que ocorrem na perfeição da natureza e que levaram, e ainda hoje o fazem por muitas vezes, os seres humanos neles se inspirarem. Uma pequena lista destas inspirações decorrentes da natureza é apontada abaixo:

- Pássaros → Aviões
- Morcegos → Sonares
- Peixes → Submarinos

Outro fato curioso na natureza foi a descoberta realizada por pesquisadores que ao estudarem o comportamento de colônias de formigas observaram que mesmo sendo elas seres tão simples e irracionais possuem mecanismos naturais de otimização. Isto é, são capazes de encontrar um caminho mais curto entre o ninho e uma fonte de comida sem usar sugestões visuais, mesmo que ocorram mudanças no ambiente original como a introdução de um obstáculo (Figuras 1 a 4).

A explicação para a descoberta do caminho ótimo a ser percorrido pelas formigas é que elas depositam durante suas caminhadas uma certa quantidade de feromônio, aproximadamente com taxa fixa através da urina, e preferem seguir uma trajetória probabilisticamente mais rica em tal substância.

Quando o caminho original é interrompido pela introdução de um obstáculo, as mesmas se dividem pelas alternativas possíveis e, com o decorrer do tempo, como pelo menor caminho terão passado mais formigas, ficando o mesmo com uma maior concentração de feromônio, este novo trajeto será então adotado. Nas figuras da página seguinte é ilustrado este processo de otimização natural.



Figura 1 - Formigas em linha reta entre o ninho e a fonte de comida [23]

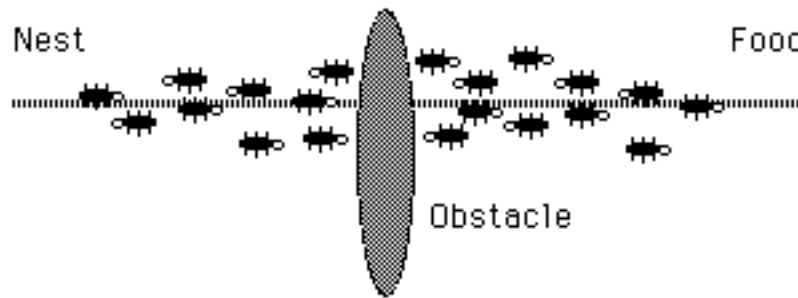


Figura 2 - Introdução de um obstáculo no percurso original [23]

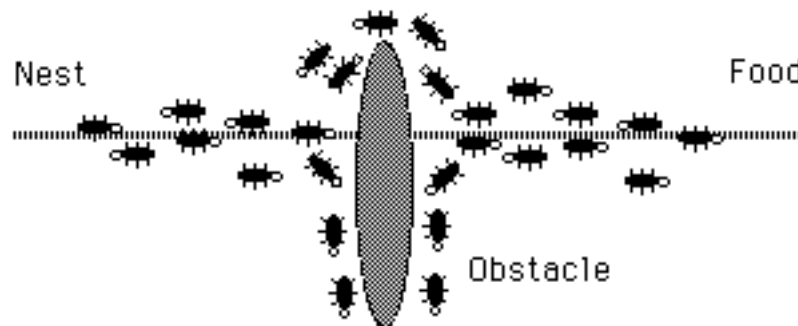


Figura 3 - Divisão mediana das formigas pelos dois novos percursos [23]

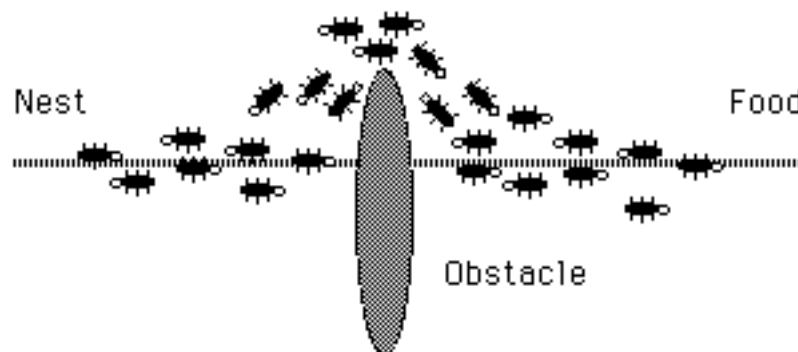


Figura 4 - O percurso menor (ótimo) é escolhido com o passar do tempo [23]

O emprego de algoritmos genéticos, que são fundamentados na teoria da Seleção Natural de Charles Darwin e nos mecanismos da genética, como ferramenta para a otimização pode ser filosoficamente defendida frente à utilização de outros métodos, simplesmente por ter sido este o processo adotado pela “quase” perfeita natureza. Há muitos séculos os homens estudam e pesquisam os fenômenos naturais, embora ainda tão longe a ciência se encontra das respostas e perfeições advindas da natureza.

A idéia e a necessidade de otimização, exemplificada até mesmo por simples formigas, em conjunto com a seleção natural dos mais aptos no processo de evolução torna-se deste modo um campo convidativo, interessante, prazeroso e pelos inúmeros resultados de sucesso apresentados nas mais diferentes áreas de aplicação, uma alternativa de grande força por suas características: robustez, flexibilidade, eficácia e, principalmente, simplicidade.

2. Otimização

O conceito de *otimização* de qualquer tarefa ou processo está diretamente ligado à sua realização, *do modo mais eficiente possível*. Esta eficiência pode ser avaliada de inúmeras maneiras, conforme o tipo de tarefa a ser realizada. Podendo ser, por exemplo:

- A minimização do tempo gasto na limpeza de um jardim.
- A maximização da velocidade de processamento de um chip eletrônico.
- A minimização do custo de execução de um projeto de barragem, dentre muitos outros.

A otimização é deste modo uma ciência que está sempre em demanda, uma vez que se encontra direta ou indiretamente relacionada com capital e é empregada em todos os campos de aplicações: engenharia civil, mecânica, automobilística, aérea, econômica, eletrônica, química, etc.

Na engenharia civil ela vem sendo muito aplicada por todas as suas sub-áreas, tais como nos problemas: otimização de redes de abastecimento, dosagens de materiais, agendamento de tarefas e no gerenciamento de itinerários de linhas de transporte, bem como no projeto de estruturas, tema deste trabalho.

3. Programação Matemática

A primeira linha de métodos desenvolvidos para tratamento dos problemas de otimização foi a Programação Matemática. Por esta razão, a grande maioria dos livros dedicados à otimização versa sobre este tema [04, 32, 42, 46, 53, 55].

A programação matemática trata o problema de forma iterativa e determinista, isto é, através de gradientes, funcionais, operações matriciais, etc. Por esta razão requerem normalmente muitas informações e condições do problema a ser resolvido: região viável bem definida, suavidade da função objetivo e convexidade do problema.

Um ponto fraco da programação matemática é ainda não ter desenvolvido um método que possibilite a busca de soluções ótimas globais, ou seja, seus algoritmos podem convergir e se prender a qualquer solução ótima local, o que dependerá basicamente da solução de partida adotada.

4. Computação Natural

Nas últimas décadas surgiu uma nova ciência computacional inspirada na natureza denominada Computação Natural. Algumas de suas principais áreas são apresentadas abaixo, junto com a descrição dos fenômenos ou processos naturais que as inspiraram e seus fundamentos básicos:

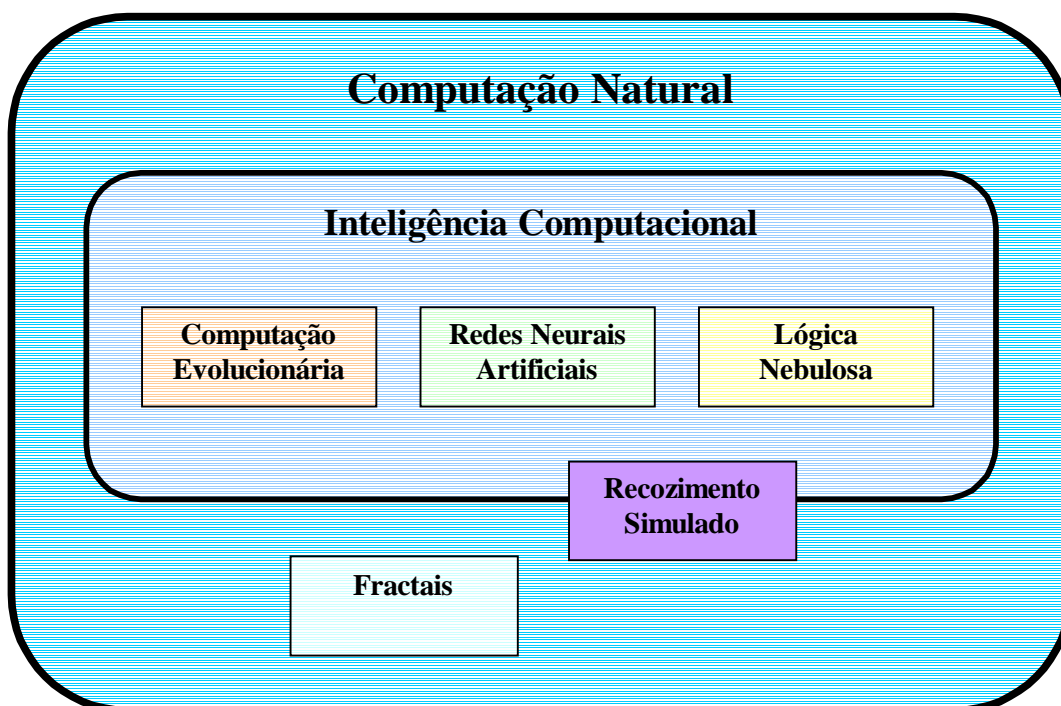


Figura 5 - Algumas Sub-áreas da Computação Natural

A Inteligência Computacional busca criar sistemas inteligentes que reproduzam aspectos do comportamento humano, tais como: percepção, raciocínio, adaptação e aprendizado.

A técnica de Recozimento Simulado não apresenta exatamente as características da inteligência computacional, sendo normalmente tomada numa classificação intermediária, como apresentado na ilustração anterior.

4.1. Fractais

A geometria fractal surgiu em 1975, quando o matemático polonês Benoit Mandelbrot publicou seus trabalhos no livro “Os objetos fractais: forma, acaso e dimensão”, no qual divulga novos objetos geométricos - os fractais [47].

- Fractal → palavra originada do adjetivo Latim fractus - quebrado; fragmentado; irregular. É uma figura que possui uma regularidade geométrica fundamental e invariante para diferentes escalas, ou seja, cada fragmento é estatisticamente semelhante ao todo, descrevendo em geral, objetos de dimensões fracionárias.

Ao contrário da geometria euclidiana que estuda cubos, esferas e muitas outras figuras que não existem na natureza, a geometria fractal descreve objetos reais, tais como: nuvens, montanhas, superfícies cerebrais, fronteiras entre países, etc. Essas figuras possuem dimensão fracionária e preservam sua identidade para todas as escalas nas quais sejam observadas.

Algumas décadas antes do surgimento da geometria fractal, George Cantor (1845-1918) já havia criado um conjunto fractal através da fragmentação do segmento de reta compreendido pelo intervalo $[0, 1]$. Trata-se de um procedimento onde, primeiramente, divide-se o segmento em três partes auto-similares, para depois se retirar a parte central. De cada intervalo restante retira-se a respectiva terça parte mediana, repetindo-se o processo indefinidamente. Define-se assim o Conjunto de Cantor, que é um tradicional exemplo de fractal.

A geometria fractal descreve bem as irregularidades e os processos turbulentos da natureza, sendo largamente utilizada em inúmeras áreas: meteorologia, biologia, hidrografia, medicina, geografia, dentre outras. Como um exemplo de sua aplicação pode-se citar o seu emprego recente como ferramenta de compactação de dados da enciclopédia Encarta desenvolvida pela empresa americana Microsoft.

4.2. Recozimento Simulado

No coração do método de Recozimento Simulado [41] está a analogia com a termodinâmica, mais especificamente no modo como os líquidos se esfriam e cristalizam ou no processo de recozimento de metais.

Em altas temperaturas as moléculas de um líquido movem-se livremente umas em relação às outras. Caso o líquido seja submetido a um resfriamento lento, esta mobilidade térmica é perdida. Os átomos alinham-se então de forma ordenada, formando um cristal puro, esse cristal é o estado de energia mínima do sistema. Por outro lado, caso o processo de resfriamento não seja lento e gradual, estados mais altos de energia são alcançados, não havendo a formação de cristal puro.

Estes princípios foram inseridos dentro do contexto da computação numérica por Metropolis, sendo posteriormente aplicados em problemas de otimização combinatória, tal como o do caixeiro viajante.

4.3. Lógica Nebulosa

A Lógica Nebulosa ou Difusa surgiu da necessidade do tratamento de problemas onde existe uma região cujas classificações tornam-se imprecisas [27].

A Lógica Nebulosa não trata as variáveis como tendo apenas um estado, mas sim n estados; cada qual com um certo grau de associação, ou seja, uma casa não é grande e sim 0.8 grande, 0.2 média e 0.0 pequena. Isto gera a criação de conjuntos para os quais os dados podem ser inseridos. Estas características lhe permitem um melhor tratamento de problemas onde existam fronteiras imprecisas ou não bem definidas.

A Lógica Nebulosa tem sido empregada em diversas áreas de modo isolado ou em associação com outras técnicas.

4.4. Redes Neurais Artificiais

As Redes Neurais [47, 75] são inspiradas na estrutura do cérebro humano, com o objetivo de apresentar características similares a ele: aprendizado, associação, generalização e abstração. Elas são compostas por diversos elementos, os processadores (neurônios artificiais), altamente interconectados, que efetuam um número pequeno de operações simples e transmitem seus resultados aos processadores vizinhos.

Devido à sua estrutura, as Redes Neurais são bastante efetivas no aprendizado de padrões a partir de dados não-lineares, incompletos, com ruídos e até compostos de exemplos contraditórios. Algumas aplicações são: reconhecimento de padrões (imagem, texto, voz, etc.), previsão de séries temporais e modelagem de problemas específicos.

4.5. Computação Evolucionária

A Computação Evolucionária ou Algoritmos Evolucionários [76] são técnicas estocásticas de busca e otimização, poderosas e largamente aplicáveis, inspiradas nos mecanismos da evolução e da genética. A idéia é a evolução de uma população de estruturas computacionais, de modo tal que melhore a adequação média dos indivíduos que formam esta população em relação ao ambiente a que ela está submetida.

O algoritmo básico da Computação Evolucionária pode ser definido como se segue:

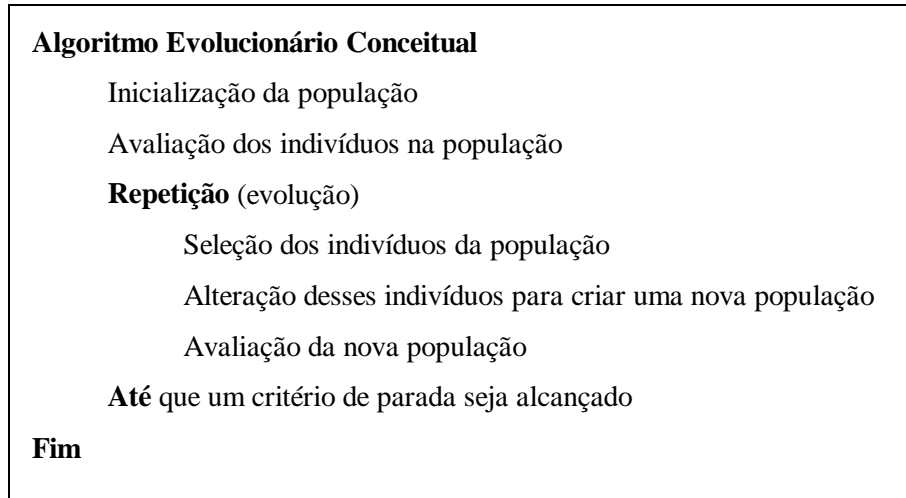


Figura 6 - Algoritmo Evolucionário conceitual

Uma pergunta intuitiva que surge em relação aos problemas de otimização é: “*Por que utilizar a Computação Evolucionária*”?

Ponto de vista filosófico

- É o método de otimização preferencialmente utilizado pela natureza, que é tida por muitos como o sistema mais perfeito.

Ponto de vista prático

- Resolve problemas com modelos matemáticos complexos de modo simples.
- Fácil acoplamento com outras técnicas (hibridização) e/ou aplicações.

Conforme a linha de aplicação e alguns detalhes da Computação Evolucionária formaram-se novas sub-áreas desta técnica que são descritas a seguir:

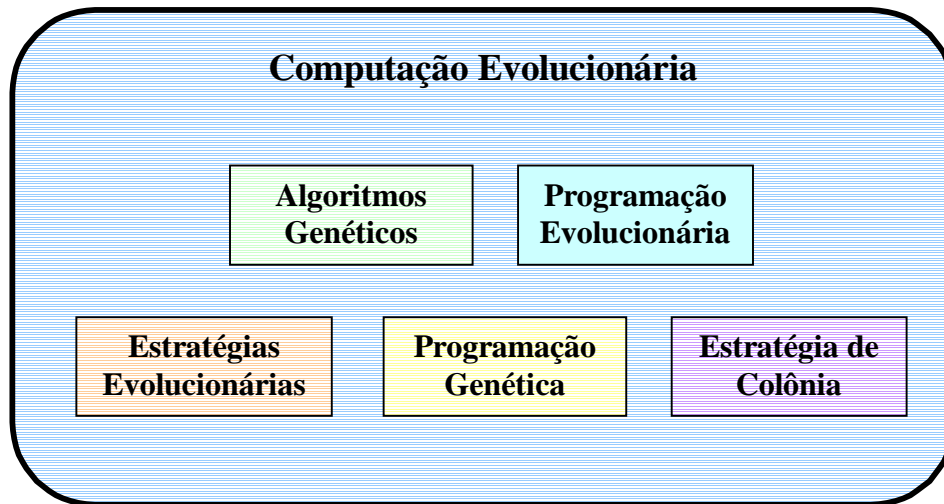


Figura 7 - Sub-áreas da Computação Evolucionária

Na seqüência serão apresentadas introduções rápidas sobre cada uma das sub-áreas destacadas na figura acima.

4.5.1. Algoritmos Genéticos (AGs)

São algoritmos que operam com populações de indivíduos representados por cromossomos, os quais durante o processo de evolução são submetidos ao procedimento de seleção e reprodução, onde são aplicados os operadores de recombinação e mutação, que são inspirados nos mecanismos naturais da genética [03, 30, 36, 52].

4.5.2. Programação Evolucionária ou Evolutiva (PE)

Originalmente foi concebida por Lawrence J. Fogel em 1960 como uma estratégia de otimização estocástica similar aos AGs. No entanto, enfatizam o relacionamento entre os genitores e seus descendentes ao invés de tentar emular operadores genéticos específicos observados na natureza [76].

A PE também opera com populações, mas apenas diferentes níveis de mutação são efetuados sobre os genitores na criação de novas soluções. O tamanho da população não necessita ser mantido constante, como também não é fixado um número de descendentes por genitor. Trabalham com representações mais flexíveis que as empregadas pelos AGs por não efetuarem recombinações.

4.5.3. Estratégias Evolucionárias ou de Evolução (EE)

Foram concebidas para tratarem problemas técnicos de otimização e quase que exclusivamente empregadas em engenharia como alternativa aos métodos convencionais. Operam com cromossomos na forma de vetores de números reais e originalmente na proporção (1+1), isto é, cada genitor gera um herdeiro por geração, normalmente por mutações distribuídas. Caso este descendente seja melhor que seu genitor ele lhe toma o lugar. Atualmente estas estratégias foram extendidas para as proporções (m+1) e (m+n), além de terem tido operadores de recombinação introduzidos no seu processo evolutivo [76].

4.5.4. Programação Genética (PG)

Opera sobre representações de trechos de programas na forma de árvores, de modo que possam ser combinados para gerarem novos trechos de programas mais complexos. Usualmente é utilizada a linguagem *Lisp* pela facilidade de sua representação [43, 76].

Os programas na população são compostos de elementos ou conjuntos fixos de símbolos, devidamente selecionados por serem apropriados ao domínio do problema a ser resolvido.

Normalmente, ênfase maior é dada às operações de recombinação, através de uma seleção aleatória de sub-árvores e posteriores trocas nos indivíduos. Os indivíduos são selecionados conforme suas aptidões.

4.5.5. Estratégia de Colônia

A Estratégia de Colônia é inspirada no comportamento de enxames, isto é, no comportamento coletivo de colônias de insetos, como por exemplo, colônia de formigas (*Ant Colony*), ou em sociedades de outros animais [23].

Operam com a idéia da comunicação indireta explorada pelas sociedades de insetos, que formam algoritmos distribuídos de multi-agentes.

Esta estratégia está sendo aplicada a vários problemas de otimização de cadeias de telecomunicações, distribuição de tarefas e principalmente em problemas de otimização combinatória.

5. Introdução ao Código Genético

Os seres humanos possuem uma sequência de aproximadamente 3,2 bilhões de letras distribuídas por 23 pares de cromossomos [31]. Estas letras devidamente sequenciadas formam palavras que se agrupam em frases. O conjunto destas frases é o código genético ou o genoma do indivíduo estudado, que é formado por moléculas estruturadas em dupla hélice – o DNA ou ácido desoxirribonucléico.

As referidas letras citadas anteriormente são: A, T, C e G, que representam, respectivamente, as bases químicas adenina, timina, citosina e guanina, constituintes do material genético. Como exemplo, pode-se citar o HIV-1, vírus da aids, que é formado por uma sequência de 9661 letras.

Cada uma de nossas células carrega no núcleo uma cópia de nosso próprio código genético e utiliza tais informações para produzir e regular a atividade de todas as proteínas que constituem nosso organismo.

Cada trecho do genoma que codifica uma proteína é chamado de gene ou gen. No caso dos seres humanos, os cientistas acreditam existir entre 30000 e 40000 genes, correspondentes a algo entre 3% e 5% das letras do genoma. O restante do material genético tem a função de regular a ação dos genes ou simplesmente não tem função, pelo menos pelo que se sabe até o presente momento. Assim, dois seres humanos têm sequências genéticas 99,9% idênticas.

O projeto genoma conseguiu recentemente escrever o primeiro rascunho do genoma humano, sendo seu próximo passo identificar onde estão e o que representam estas pequenas diferenças de 0,1% que fazem de cada pessoa um ser único. A partir do mapeamento do genoma humano espera-se encontrar a cura para doenças como o Câncer, o mal de Alzheimer, etc.

Um enorme banco de dados onde estão armazenados os genomas públicos é mantido na Internet pelo projeto genoma - chamado GenBank - e cresce a uma grande velocidade a cada dia. Na verdade, ele tem dobrado de tamanho a cada 14 meses e já conta com mais de 5 bilhões de letras, representando organismos como a mosca de frutas, o arroz, o HIV, os seres humanos, dentre muitos outros.

As seguintes etapas têm sido adotadas para a decodificação do genoma humano:

- 1) Quebra em pedaços menores de até 1000 letras
- 2) Seqüenciamento desses trechos
- 3) Ordenamento completo com o auxílio de computadores
- 4) Descobrimento e anotação dos genes na seqüência final

A empresa privada Celera Genomics, em Rockville no Estado de Maryland nos EUA, anunciou que já terminou a terceira etapa deste procedimento e que prossegue rumo ao mapeamento final de todos os genes humanos. Para isso, a Celera conta com 300 seqüenciadores de última geração e um supercomputador com 1200 processadores em paralelo, o qual funciona 24 horas por dia para colocar as letras em ordem e interpretar os resultados.

Segundo a Compaq, o supercomputador utilizado pela Celera é, no presente momento, o computador mais veloz do planeta e, para garantir toda esta estrutura tecnológica, a Celera conseguiu recentemente 1 bilhão de dólares em duas ofertas de ações na bolsa de Wall Street.

Acredita-se que a humanidade passa por uma nova revolução tecnológica, cujos benefícios e malefícios só serão descobertos com o desenrolar dos anos, dependendo principalmente do modo como todo este avanço científico será utilizado na prática.

6. Organização da Tese

O capítulo II introduz o assunto de otimização, apresentando a formulação e os conceitos básicos do tema, além das alternativas disponíveis para o seu tratamento.

O capítulo III estende os tópicos do capítulo II para o contexto dos problemas de otimização com multi-objetivos, destacando as principais diferenças conceituais e de tratamento entre estas duas linhas de otimização.

O capítulo IV descreve os fundamentos da técnica dos algoritmos genéticos, sua fonte de inspiração, fundamentos e particularidades. São detalhados os procedimentos do algoritmo, os operadores e as configurações necessárias para o emprego deles.

No capítulo V são discutidas as novas estratégias empregáveis aos algoritmos genéticos que os tornam atraentes e promissores na resolução de problemas.

No capítulo VI os principais algoritmos genéticos multi-objetivos são discutidos, suas idéias, inovações e limitações.

O capítulo VII apresenta o algoritmo genético multi-objetivo de Pareto desenvolvido, descrevendo suas características, particularidades, inovações e o conjunto de estratégias acopladas para sua formação.

No capítulo VIII várias aplicações são resolvidas com o intuito de atestarem o funcionamento do algoritmo proposto, bem como determinar o conjunto ótimo de Pareto de problemas estruturais diversos.

O capítulo IX discute as contribuições do trabalho realizado e apresenta as conclusões oriundas da experiência adquirida.

Capítulo II

Otimização

Neste capítulo é apresentada uma abordagem rápida e objetiva sobre a Programação Matemática. São descritos a formulação do problema de otimização e os seus conceitos principais. Fechando o capítulo são comentadas as sub-áreas da programação matemática, bem como apresentada uma possível classificação geral de seus principais métodos.

1. Formulação.....	18
2. Definições	18
3. Fraqueza dos Métodos Clássicos.....	20
4. Classificação dos Métodos Clássicos	20

1. Formulação

Os problemas de otimização apresentam alguns conceitos e definições largamente empregadas na literatura e de conhecimento imprescindível. Estas definições serão apresentadas na sequência da forma do problema de otimização.

Problema Clássico de Otimização

Max. ou Min. $f(x_1, x_2, \dots, x_N)$ (*função objetivo*)

sujeito a

$g_j(\mathbf{x}) \geq 0$ $j = 1, 2, \dots, J$ (*restrições de*)

$h_k(\mathbf{x}) = 0$ $k = 1, 2, \dots, K$ (*comportamento*)

$x_i^{(L)} \leq x_i \leq x_i^{(U)}$ $i = 1, 2, \dots, N$ (*restrições laterais*)

(*nas variáveis*)

(*de projeto x_i*)

2. Definições

- **Variável de Projeto** → as variáveis de projeto são aquelas que se alteram durante o processo de otimização. Elas podem ser contínuas (reais), inteiras ou discretas (valores compreendidos dentro de um certo conjunto fixo).

De um ponto de vista físico, as variáveis de projeto podem representar as seguintes informações sobre a estrutura:

- propriedades mecânicas ou físicas do material;
- a topologia da estrutura, isto é, o padrão de conexão dos elementos ou o seu número de elementos;
- a configuração ou a forma geométrica da estrutura;
- dimensões de seções transversais ou comprimento dos elementos.

- **Restrições** → as restrições são funções de igualdade ou desigualdade que descrevem situações indesejáveis de projeto. Dividem-se em dois grupos:
 - Restrições Laterais → efetuadas diretamente sobre as variáveis de projeto, limitando seus valores;
 - Restrições de Comportamento → condições desejáveis de limites de tensões, deslocamentos, frequências naturais de vibração, etc.
- **Espaço de Busca ou Região Viável** → É o conjunto, espaço ou região que compreende as soluções possíveis ou viáveis do problema a ser otimizado. Deve ser caracterizado pelas funções de restrição, que definem as soluções viáveis do problema a ser resolvido.
- **Função Objetivo ou de Avaliação** → É a função que se quer otimizar. Ela pode ser de uma ou mais variáveis, sendo estas duas opções classificadas como otimização *unidimensional* e *multidimensional* respectivamente.
- **Ponto Ótimo** → É o ponto, caracterizado pelo vetor $x^* = (x_1, x_2, \dots, x_N)$, formado pelas variáveis de projeto que extremizam a função objetivo e satisfazem as restrições.
- **Valor Ótimo** → É o valor da função objetivo $f(x^*)$ no ponto ótimo.
- **Solução Ótima** → É o par formado pelo ponto ótimo e o valor ótimo $[x^*, f(x^*)]$. A solução ótima pode ser:
 - local → quando o valor ótimo é localizado;
 - global → quando o valor ótimo é global na região viável;
 - restringida → quando há restrições satisfeitas como igualdade (ativas);
 - não-restringida → quando não há restrições ativas no ponto ótimo.

3. Fraqueza dos Métodos Clássicos

Os métodos matemáticos geralmente apresentam teoremas que lhes garantem a convergência para uma solução ótima. Entretanto, ela não será necessariamente a solução ótima global, o que pode, *eventualmente*, até ocorrer. Em geral, este fato dependerá do ponto de partida fornecido para o método utilizado. Este é um problema que tem sido abordado há muito tempo pelos pesquisadores mas que ainda não se encontra resolvido.

4. Classificação dos Métodos Clássicos

De acordo com as características da função objetivo e das restrições, classificam-se os problemas de otimização nas seguintes sub-áreas da Programação Matemática:

- **Programação Linear** → quando a função objetivo e as restrições são funções lineares das variáveis de projeto;
- **Programação Não-Linear** → quando a função objetivo ou pelo menos uma das restrições é função não-linear das variáveis de projeto.

Posteriormente, outras sub-áreas foram aparecendo com o objetivo de maior especialização nos problemas a serem resolvidos, deste modo tem-se:

- **Programação Quadrática** → quando a função objetivo é quadrática e as restrições são funções lineares das variáveis de projeto;

Os problemas de otimização não precisam necessariamente apresentar restrições, sendo este um fato que marca a distribuição dos problemas em duas grandes classes: Problemas de Otimização “*com restrições ou restringidos*” e “*sem restrições*”.

Inúmeros são os métodos criados para tratamento de problemas de otimização, bem como muitas são as classificações realizadas pelos diversos autores da literatura. A seguir é apresentada uma possível classificação geral de uma vasta gama de métodos matemáticos existentes [53]:

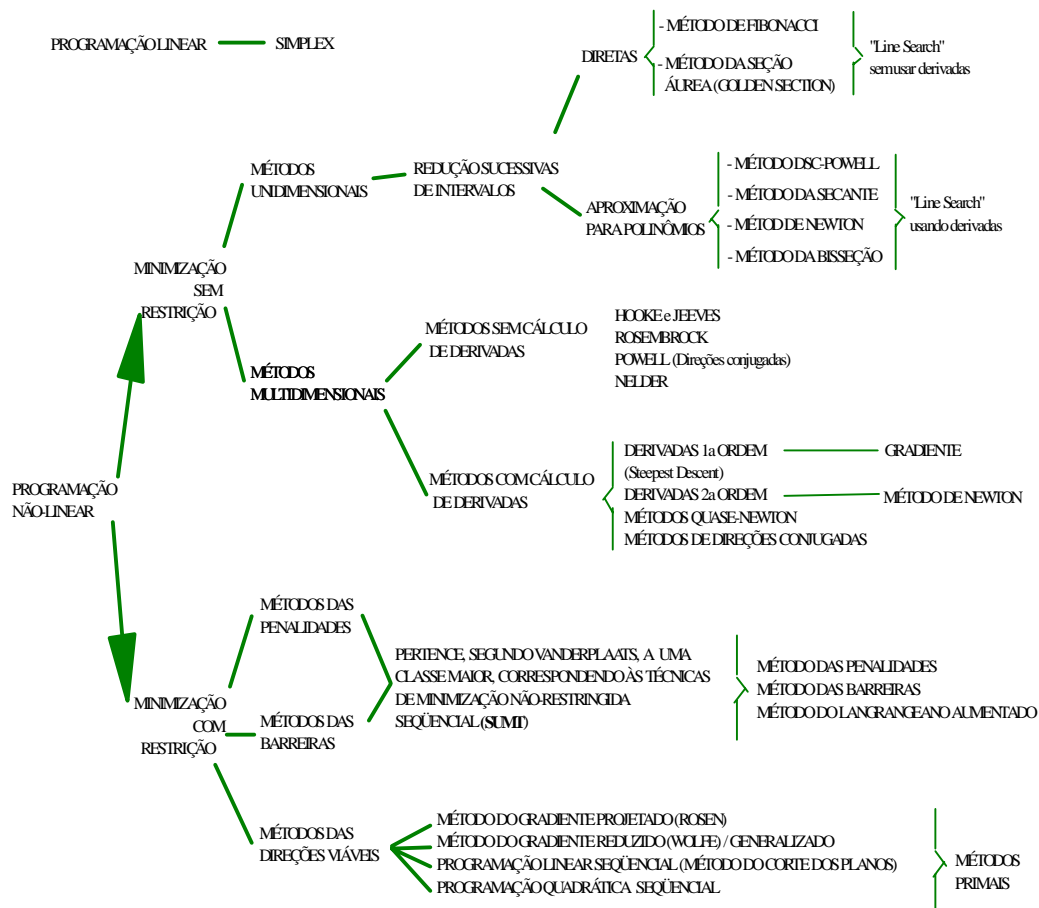


Figura 8 - Classificação Geral dos Métodos Clássicos [53]

Capítulo III

Otimização Multi-objetivos

As definições básicas dos problemas de otimização multi-objetivos são apresentadas neste capítulo. Além disso, são destacadas as principais diferenças entre a otimização com um só objetivo da otimização com multi-objetivos.

1. Introdução	23
2. Definições	25

1. Introdução

Muitos problemas do mundo real apresentam uma coleção de objetivos a serem extremizados que são na maioria das vezes conflitantes entre si, ou seja, a melhoria de algum(uns) objetivo(s) causa(m) conseqüentemente a deterioração de outro(s).

Um exemplo de problema com objetivos conflitantes seria o projeto de uma ponte onde se deseja minimizar o peso (custo) da estrutura e maximizar as frequências naturais de vibração (melhor desempenho dinâmico): à medida que se reduz o peso da ponte também se diminuem suas frequências naturais de vibração. Portanto, não existe uma solução ótima única e sim um conjunto de soluções. Tais soluções são ótimas porque não existem outras soluções no espaço de busca melhores do que elas, quando todos os objetivos são simultaneamente considerados, sendo conhecidas como *soluções ótimas de Pareto*.

Outro exemplo comum de problema multi-objetivo é a tarefa de comprar um computador. A aquisição ótima é aquela que fornece o custo mínimo enquanto maximiza o desempenho do equipamento. Estes objetivos são conflitantes entre si, uma vez que existirão desde computadores com elevado custo e desempenho até aqueles com baixo custo e desempenho. Um computador com o mais alto desempenho pelo menor custo, embora ideal, não existe no mundo real.

Assim, nenhuma solução que tenha menor custo e desempenho pode ser considerada como superior a outra com maior custo e desempenho. Contudo, dentre todas as configurações de equipamentos existem algumas que são superiores a outras, isto é, apresentam desempenho maior ou equivalente por um custo menor ou igual. Estas configurações (soluções) que superam outras são as já citadas *soluções não-dominadas*, enquanto que as configurações que são superadas por pelo menos uma outra são as *soluções dominadas*.

Deste modo, é muito interessante uma ferramenta que encontre o conjunto das soluções não-dominadas para que o projetista escolha dentre estas aquela que melhor atenda suas necessidades de projeto. Esta é a tarefa da *otimização multi-objetivos*.

Dentro da otimização com multi-objetivos existem duas correntes para tratamento do problema:

- i. definidas prioridades e/ou pesos entre os vários objetivos de interesse, encontra-se a solução ótima segundo estas informações fornecidas a priori;
- ii. sem nenhuma informação adicional, encontra-se o conjunto das soluções ótimas de Pareto para dentre estas se escolher uma a posteriori.

Exemplos de métodos da primeira linha seriam: Programação Objetiva, Recozimento Simulado ou métodos evolucionários onde se combinam as diversas funções objetivo dentro de uma única função, obtendo como resultado da otimização uma solução única. Por outro lado, existem outros métodos evolucionários que abordam o problema pela segunda linha de tratamento, possibilitando a obtenção de um grande número das soluções ótimas de Pareto para uma escolha pessoal posterior.

A grande questão da primeira corrente é: como definir todas as prioridades e/ou pesos de problemas que, na maioria das vezes, se conhece tão pouco. Ou seja, uma escolha errada de informações não permite uma maior avaliação das suas possibilidades, já que não se conhecem as demais soluções. Assim sendo, a aplicação destas duas correntes levam o tomador de decisões por analogia a duas possíveis provas:

- i. *prova aberta*: sem imaginar a resposta correta, o candidato precisa apresentar uma solução (prioridades e/ou pesos);
- ii. *prova de múltipla escolha*: dentre as alternativas apresentadas, o candidato precisa escolher aquela que mais lhe agrada. Sendo que neste caso, todas as alternativas são corretas (conjunto das soluções não-dominadas encontrado).

Levando-se em consideração os fatos apresentados e a realidade da falta de informações qualitativas e quantitativas dos problemas multi-objetivos do mundo real, conclui-se por uma abordagem mais interessante do problema pela segunda linha de solução apresentada: *obtenção dos conjuntos ótimos de Pareto*. Esta conclusão é que faz com que os métodos evolucionários sejam preferidos em relação aos métodos clássicos ou mesmo ao Recozimento Simulado, já que estas duas categorias não foram projetadas para trabalharem com múltiplas soluções (populações) como naturalmente acontece com os algoritmos genéticos.

2. Definições

Um problema de otimização multi-objetivos pode ser descrito por um vetor \mathbf{y} de n funções objetivo f_i que dependem de um vetor \mathbf{x} de m variáveis independentes x_i :

$$\text{minimize / maximize} \quad \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$$

$$\text{sujeito a} \quad \mathbf{x} = (x_1, x_2, \dots, x_m) \in X$$

$$\mathbf{y} = (y_1, y_2, \dots, y_m) \in Y$$

onde:

$\mathbf{y} \rightarrow$ vetor objetivo;

$f_i \rightarrow$ funções objetivo;

$Y \rightarrow$ espaço das funções objetivo;

$\mathbf{x} \rightarrow$ vetor de decisão;

$x_i \rightarrow$ variáveis de decisão;

$X \rightarrow$ espaço das variáveis de decisão.

O conjunto de soluções do problema anterior consiste em todos os vetores de decisão que não podem melhorar o valor de qualquer objetivo, sem a degradação de pelo menos algum dos outros. Estes vetores são os conhecidos ótimos de Pareto.

Sem perda de generalidade, pode-se descrever as soluções ótimas de Pareto formalmente para um problema de minimização puro, isto é, onde todas as funções objetivas devem ser minimizadas, pelas seguintes definições:

Definição 1: Dominância

Um ponto $\mathbf{A}=(a_1, a_2, \dots, a_m)$ domina um outro ponto $\mathbf{B}=(b_1, b_2, \dots, b_m)$ se:
 $\forall i \in \{ 1, 2, \dots, n \} : f_i (\mathbf{A}) \leq f_i (\mathbf{B})$ e $\exists j \in \{ 1, 2, \dots, n \} : f_j (\mathbf{A}) < f_j (\mathbf{B})$

Definição 2: Solução não-dominada ou ótima de Pareto

Um ponto \mathbf{A} é uma solução não-dominada e pertence à fronteira de Pareto se: não existe nenhum outro ponto \mathbf{B} , tal que \mathbf{B} domine \mathbf{A} . Em geral, para cada região viável existe um subconjunto S de soluções não-dominadas, também conhecidas como *ótimos de Pareto*.

Observação:

Existe uma diferença entre um *conjunto de soluções não-dominadas* e um *conjunto ótimo de Pareto*. Um conjunto de soluções não-dominadas é definido no contexto de uma amostra do espaço de busca, enquanto que o conjunto ótimo de Pareto é definido em relação a todo o espaço de busca.

i. **Exemplo do Emprego das Definições Apresentadas**

Um problema simples de otimização multi-objetivo que foi proposto e testado por Schaffer [57] (função f_2), consta de duas funções g e h que devem ser simultaneamente minimizadas:

$$f_2 = (g(x), h(x)) \quad \text{sendo} \quad g(x) = x^2 \quad \text{e} \quad h(x) = (x - 2)^2$$

A figura seguinte mostra a representação gráfica das funções objetivo g e h , sendo fácil identificar que as soluções ótimas de Pareto devem estar compreendidas no intervalo $[0, 2]$, já que fora desta região, tanto a função g como a h crescem.

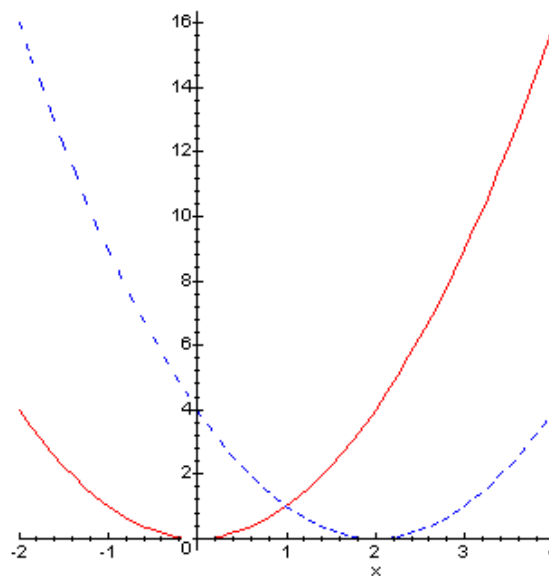


Figura 9 - Funções g e h de um problema multi-objetivo (f_2 - Schaffer [57])

Dentro do intervalo $[0, 2]$ acontece uma compensação entre as funções objetivo, isto é, enquanto uma cresce a outra diminui de valor, não sendo portanto tão trivial identificar as soluções ótimas de Pareto. Usualmente, recorre-se a um gráfico paramétrico no espaço das funções objetivo para tais conclusões, como pode ser observado na figura que se segue:

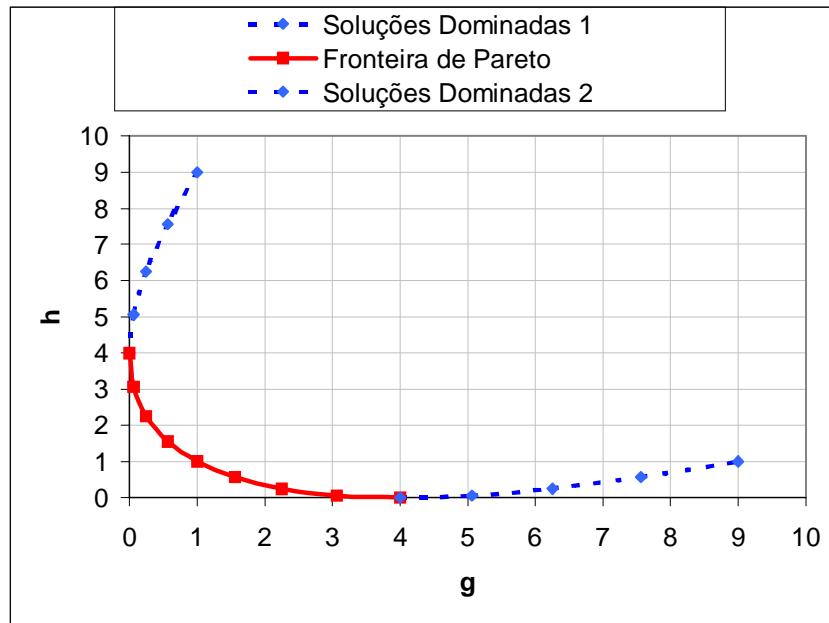


Figura 10 - Identificação de uma fronteira de Pareto

Pelo gráfico anterior é fácil identificar a fronteira de Pareto, todavia, para problemas maiores que o apresentado muitas vezes não é mais possível uma análise gráfica do problema estudado, entrando aí os métodos computacionais para a resolução do problema de otimização.

Capítulo IV

Algoritmos Genéticos

São apresentados neste capítulo os principais fundamentos da técnica dos Algoritmos Genéticos, além das origens do método, definições, procedimentos, vantagens, parâmetros de configuração e outros tópicos importantes relacionados ao tema.

1. Introdução	30
2. Histórico	33
3. Definições	38
4. Diferenças entre os Algoritmos Genéticos e os Métodos Clássicos.....	39
5. Estrutura dos Algoritmos Genéticos	41
6. Aspectos Principais dos Algoritmos Genéticos.....	44
7. Parâmetros de Influência e Configurações.....	55
8. Vantagens e Desvantagens dos Algoritmos Genéticos.....	59
9. Sumário e Discussão	60

1. Introdução

Há muito tempo, o homem tem se servido das características e princípios existentes na natureza para a criação de máquinas, métodos e técnicas que “melhorem” sua vivência neste planeta. Alguns exemplos típicos desta teoria foram as seguintes invenções: aviões baseados nas características dos pássaros, submarinos com sistemas de imersão semelhantes ao dos peixes, sonares baseados nos morcegos, dentre outros.

Recentemente novas técnicas têm sido inspiradas na natureza ou na biologia de um modo geral, como as Redes Neurais que se baseiam no funcionamento do cérebro humano para possibilitar aos computadores uma chamada “Inteligência Artificial”.

Dentro deste contexto, surgia em meados do século XIX um dos mais importantes princípios no campo da evolução da vida, A Seleção Natural de Darwin, que defende a idéia de que na natureza, aqueles seres vivos com melhores características ou “adaptabilidades”, tendem a sobreviver frente aos demais. Desde então, a medicina e suas ciências afins vêm numa busca acelerada tentando “mapear” toda a informação genética humana, relacionando deste modo cada gene de cada cromossomo às características que eles representam nos indivíduos: hereditárias, físicas e funcionais.

Há oito anos começou o projeto Genoma em diversos centros do mundo, liderados pelo instituto Nacional de Saúde dos Estados Unidos, que contou com investimento inicial de três bilhões de dólares. O principal objetivo do projeto é descobrir, até o ano 2005, todos os genes do corpo humano e identificar mais de três bilhões de seqüências genéticas de cromossomos.

Os cientistas querem conhecer, antes do nascimento, a possibilidade de cada ser humano desenvolver doenças com fatores hereditários, podendo assim evitar o desenvolvimento do mal, seja com terapias genéticas se a doença for

congenita ou com a interferência no estilo de vida do portador da mutação. Atualmente, apenas a função de cerca de 4% dos genes humanos são conhecidas.

Foi nestes últimos anos que a genética se alavancou com as primeiras clonagens realizadas no mundo, tendo para marco a famosa ovelha Dolly na Inglaterra.

As clonagens e possíveis objetivos e conseqüências destas descobertas provocaram inúmeras discordâncias e precauções por todo o planeta, afinal, quais seriam os rumos da aplicação destes conhecimentos. São indiscutíveis as vantagens oferecidas na prevenção de doenças, contudo, são assustadoras as possibilidades da criação de novas doenças, mutações, clonagens individuais, entre outras.

Cercadas por todas estas discussões, outras ciências, aliadas às vantagens computacionais hoje oferecidas, inspiraram-se mais uma vez nestes princípios para a resolução de outros problemas. Surgiu então, mais uma técnica de inteligência artificial denominada Algoritmos Genéticos [19, 30, 52].

Os Algoritmos Genéticos são algoritmos de busca fundamentados no processo da seleção natural proposto por Charles Darwin e nos mecanismos da genética. Foram inicialmente propostos por John Holland em seu livro publicado em 1975 [36].

Uma grande aplicação dos Algoritmos Genéticos é em problemas de busca: dado um conjunto de elementos ou indivíduos, deseja-se encontrar aquele ou aqueles que melhor atendam a certas condições especificadas [03, 30, 44].

Os Algoritmos Genéticos transformam uma população de indivíduos, cada um com um valor associado de adaptabilidade, chamado de *aptidão*, numa nova geração de indivíduos usando os princípios Darwianos de reprodução e

sobrevivência dos mais aptos, pela aplicação de operações genéticas tais como *recombinação* e *mutação*.

Cada indivíduo na população representa uma possível solução para um dado problema, o que o Algoritmo Genético faz é procurar aquela que seja muito boa ou a melhor para o problema analisado pela criação genética de populações de indivíduos cada vez mais aptos à extremização da função objetivo de interesse.

É impossível mapear todos os problemas existentes dentro de um eixo ordenado, entretanto, com uma representação abstrata desta forma e a adoção da eficiência com que os métodos resolvem cada problema individualmente para o eixo das ordenadas, seria possível determinar-se curvas indicativas da “*aplicabilidade x eficiência*” dos métodos disponíveis (Figura 11).

Utilizando-se deste gráfico encontram-se três tipos extremos de métodos, os quais foram ilustrados na figura seguinte:

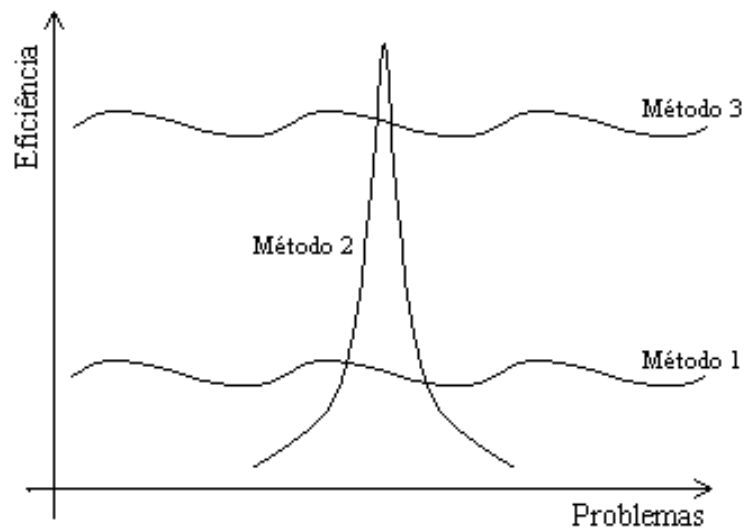


Figura 11 - Aplicabilidade em Problemas x Eficiência de Resolução dos Métodos

- **Método 1** → pouco eficiente para a totalidade dos problemas existentes.
- **Método 2** → altamente eficiente para uma pequena faixa de problemas, entretanto, pouco eficientes ou nem aplicáveis para a maior parte deles.
- **Método 3** → razoavelmente eficiente para a totalidade dos problemas existentes.

Dentro deste contexto, os Algoritmos Genéticos se aproximariam da terceira classe de métodos, não sendo mais eficientes que aqueles projetados especificamente para determinado problema tratado. Contudo, perturbações no problema original trariam quase nenhum prejuízo aos AGs, mas possivelmente, a inutilidade de outros métodos.

2. Histórico

Foram os gregos Anaximandro (séc. VI a.C.), Empédocles (V a.C.) e Aristóteles (IV a.C.) que deram os primeiros passos para a compreensão da história da vida. Entretanto, dentro de suas possibilidades, as contribuições achavam-se voltadas mais para a descrição da natureza de forma intuitiva que para a apreensão dos mecanismos de funcionamento [14].

Anaxágoras (500?-428 a.C.) chegou mesmo a supor que as espécies de animais domésticos derivavam de similares selvagens. Contudo, faltava aos gregos técnicas e materiais de pesquisa com que pudessem ultrapassar os limites da pura especulação.

Com o advento do cristianismo ganhou corpo a idéia da espécie fixa: cada ser vivo era teologicamente interpretado como um “padrão da Criação”. A forma era fruto de um ato divino, um reflexo do próprio poder do Criador, sendo imutável em si mesma.

A primeira objeção levantada contra essa teoria deveu-se a Santo Agostinho (354-430), que dizia ser a intervenção divina compatível com um mundo capaz de desenvolver-se, “da mesma maneira que no grão estão contidas invisivelmente todas as partes da futura árvore”. Todavia, essa idéia, nascida da observação do desenvolvimento individual, manteve-se confusa quanto ao fenômeno da evolução das espécies.

A partir do século XVII, o processo do conhecimento científico colocou a possibilidade da investigação basear-se em fatos concretos. Neste novo cenário, não podem ser esquecidos aqueles que foram pioneiros como Hooke (1635-1703), John Ray (1627-1705), Maupertuis (1698-1759), Buffon (1707-1788), Erasmus Darwin (1731-1802), avô de Charles Darwin, e Geoffrey de Saint-Hilaire (1772-1844), que elaborou uma tese onde afirmava terem as espécies mais evoluídas surgido ocasionalmente de monstros capazes de se desenvolver em ambiente propício. Esses homens consideraram a evolução como uma necessidade, mas foram incapazes de tirar dos ainda escassos fatos conhecidos, provas que abalassem a oposição unânime dos teólogos, cujas premissas nada tinham de científicas.

Apenas Jean Baptiste Lamarck formulou antes de Darwin uma hipótese colocando a necessidade da evolução para compreender o aparecimento das espécies atuais. Seu livro, *Philosophie Zoologique*, foi publicado em 1809 (ano em que nascia Charles Darwin).

As idéias de Lamarck argumentavam que uma grande mudança no meio ambiente provocaria em uma espécie animal a necessidade de modificar seus hábitos. Este era o fenômeno da adaptação, que mostrava que os seres vivos se encontram sempre adaptados em relação aos ambientes que vivem. Tal afirmação permanece válida até hoje e sobre ela construíram-se as mais modernas concepções sobre a evolução.

Lamarck fundamentou suas idéias em duas suposições: a lei do uso e desuso, e a transmissão de características adquiridas de pai para filho, ou a lei da herança dos caracteres adquiridos.

Quanto mais uma parte do corpo é usada, mais se desenvolve. Por outro lado, as partes não usadas se atrofiam e enfraquecem. Lamarck raciocinou então que se as características adquiridas individualmente pelo uso e desuso dos órgãos, fossem transmissíveis de pai para filho, encontrar-se-ia a chave para compreender a evolução dos seres vivos. Faltou a Lamarck o controle experimental, pois embora bem elaborada e dotada de conceitos novos e corretos, sua teoria esfacelou-se quando submetida a experiências.

Um pesquisador, Weissman, cortou sistematicamente as caudas de vinte gerações de camundongos, entretanto, na vigésima primeira as caudas continuavam aparecendo tão longas como na primeira. Todas as tentativas de comprovar a teoria de Lamarck fracassaram.

Atualmente, a genética está convicta de que os caracteres adquiridos não são transmissíveis, deste modo, uma pessoa pode passar vinte anos a se queimar ao sol, que seu filho terá as mesmas possibilidades de nascer moreno que teria se seu pai morasse num dos pólos.

A adaptação realmente ocorria, mas não como Lamarck supusera. Ainda assim, Lamarck pagou caro por sua intuição, uma vez que não pôde provar o que afirmava e foi por esta razão ridicularizado de maneira atroz e desonesta.

O ensaio sobre a progressão das populações, escrito por Malthus, deu a Darwin a idéia da competição pela sobrevivência. A observação do trabalho dos agricultores e pecuaristas, que aprimoravam suas criações através dos cruzamentos, trouxe-lhe o conceito da seleção das espécies.

“O grilo é verde porque vive na grama”. (Lamarckista)

“Porque o grilo é verde, vive na grama”. (Darwinista)

A grande quantidade de provas a favor da evolução dos organismos era realmente convincente, entretanto, ele não se satisfazia com isso e a partir das evidências formulou uma hipótese explicativa dos mecanismos da evolução.

Nascido em 12 de fevereiro de 1809, Charles Darwin depois de 28 anos de estudos e pesquisas publicou seu livro com o título “Da origem das Espécies por Via da Seleção Natural” ou “A Preservação das Raças Favorecidas na Luta pela Vida”. Os 1250 exemplares da primeira edição esgotaram-se num único dia. E, poucos dias depois, esgotaram-se mais 3000 exemplares de uma segunda tiragem.

O tempo passou e novos conhecimentos somaram-se aos anteriores, **“evoluiu a teoria da evolução”**. A ciência do século XX burilou as idéias de Darwin e outros estudiosos. Assim, para Darwin, seleção natural implicava na eliminação dos menos aptos e na sobrevivência dos mais aptos na luta pela existência. A verdadeira questão é saber se a seleção natural é ou não a causa da adaptação, mas sim como se produz a adaptação.

A seleção natural é apenas um dos elementos da adaptação que se reúne a outros, assim como a relação entre organismo e ambiente, e a hereditariedade. Além disso, é preciso distinguir adaptação de adaptabilidade. Enquanto adaptação é o preparo do organismo para sobreviver num ambiente, adaptabilidade é a capacidade de tirar partido do ambiente e controlá-lo. Nesse sentido de adaptabilidade, o homem é indubitavelmente o “rei da criação”. Mas, embora toda história da civilização humana nada mais seja que uma constante mudança no sentido de aumentar a amplitude e variedade do ajustamento ao ambiente, a evolução nos ensina que o homem não é fruto de uma criação especial, pois faz parte da natureza e é feito da mesma essência de tudo o que é vivo.

O desenvolvimento da genética estabeleceu que as características hereditárias são transmitidas através de genes (unidades químicas que se localizam no núcleo das células). Os genes são constituídos por uma substância química, o ADN, ou ácido desoxirribonucléico. Dispõe-se aos pares, dentro de filamentos visíveis ao microscópio, chamados cromossomos. A propriedade fundamental dos genes é que eles se auto reproduzem fielmente. Entretanto, não há uma precisão absoluta nessa auto reprodução, assim, pode ocorrer uma mutação gênica: o gene que até aquele momento produzia determinada característica passa a produzir outra.

Todo indivíduo nasce com seu lote herdado de gens (seu genótipo). A forma que ele terá em vida dependerá deste genótipo, bem como do tipo de vida que ele levar. Mas o tipo de vida não alterará em nada seus genes: o atleta pode ter músculos fortes, mas ele não transmite a seus filhos uma predisposição a músculos fortes. Os fatores que irão produzir mutações em seus gens, os agentes mutágenos, como a radiação, variações de temperatura (para certas espécies), fatores químicos e fatores desconhecidos, agem ao acaso, podendo produzir tanto mutações maléficas como benéficas.

Ocorre que, quando a mutação é adaptativa, os indivíduos tornam-se mais capazes de explorar as oportunidades oferecidas pelo meio, desenvolvem-se e produzem um número maior de descendentes. Este fenômeno chama-se reprodução diferencial e sua consequência é a disseminação até a generalização das características genéticas favoráveis dos melhores adaptados ao conjunto da população.

Modernamente, compreende-se o fenômeno da evolução como um processo universal de adaptação em relação ao ambiente. A mutação, seu mecanismo básico, faz-se acidentalmente, mas só persiste pela seleção natural, entendida como fenômeno de reprodução diferencial.

Estes princípios básicos de Genética Populacional que se baseiam na idéia que a variabilidade entre indivíduos de uma população de organismos que se reproduzem sexualmente se dá pela mutação e pela recombinação genética, foi desenvolvido durante as décadas de 30 e 40 por biólogos e matemáticos de importantes centros de pesquisa.

Nas décadas de 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos, entretanto, foi John Holland quem começou, seriamente, a refinar suas idéias sobre o tema, culminando em 1975 na publicação de seu livro, “Adaptation in Natural and Artificial Systems”, hoje considerado o marco dos Algoritmos Genéticos. Desde então, tais algoritmos vêm sendo aplicados com sucesso nas mais diversas áreas, dentre elas: otimização e aprendizado de máquinas.

3. Definições

As principais definições relacionadas com os AGs são:

- **Cromossomo** → Cadeia de caracteres representando alguma informação relativa às variáveis do problema. Cada cromossomo representa deste modo uma solução do problema.
- **Gen ou Gene** → É a unidade básica do cromossomo. Cada cromossomo tem um certo número de gens, cada um descrevendo uma certa variável do problema.
- **População** → Conjunto de cromossomos ou soluções.
- **Geração** → O número da iteração que o Algoritmo Genético executa.
- **Operações Genéticas** → Operações que o Algoritmo Genético realiza sobre cada um dos cromossomos.

- **Espaço de Busca ou Região Viável** → É o conjunto, espaço ou região que compreende as soluções possíveis ou viáveis do problema a ser otimizado. Deve ser caracterizado pelas funções de restrição, que definem as soluções viáveis do problema a ser resolvido.
- **Função Objetivo ou de Avaliação** → É a função que se quer otimizar. Ela contém a informação numérica do desempenho de cada cromossomo na população. Nela estão representadas as características do problema que o Algoritmo Genético necessita para realizar seu objetivo, sendo expressa normalmente como:

$$J = f(x_1, x_2, \dots, x_n)$$

onde x_1, x_2, \dots, x_n são as variáveis que o algoritmo procura determinar para otimizar J . Esta função objetivo é, em princípio, calculada para cada cromossomo da população.

4. Diferenças entre os Algoritmos Genéticos e os Métodos Clássicos

Os métodos clássicos de otimização iniciam-se com um único candidato, chamado de solução básica, e pelo cálculo de derivadas se determina para qual direção se deve caminhar na busca do próximo candidato. Por trabalharem com o cálculo de derivadas, são denominados algoritmos de ordem n , onde n é a maior derivada utilizada. Exemplos típicos são os métodos dos gradientes conjugados e de Newton, que por utilizarem derivadas primeiras e segundas, respectivamente, são caracterizados como algoritmos de primeira e segunda ordem. Todavia, classificações intermediárias são também possíveis, como é o caso do método Quasi-Newton, que se situaria entre os dois anteriores.

O maior problema destes algoritmos matemáticos é que não existe nenhuma garantia da obtenção de um ponto extremo global, ou seja, o algoritmo convergirá para o extremo local mais próximo da direção de busca determinada pelas derivadas. Por esta razão, tais algoritmos são mais convenientemente empregados para problemas *unimodais*, aqueles que apresentam apenas um extremo no intervalo considerado.

A aplicação destes algoritmos para problemas *multimodais* não é tão simples, já que a solução encontrada dependerá do ponto de partida inicial, podendo na maioria das vezes se encontrar uma solução extrema local muito “pior” que a solução ótima global desconhecida e procurada.

Os Algoritmos Genéticos representam uma classe de ferramentas muito versátil e robusta a ser empregada na solução de problemas de otimização, embora não devam ser considerados estritamente extremizadores de funções. Assim como outros métodos, por não empregarem o cálculo de derivadas, mas sim atuarem diretamente na busca das soluções no espaço viável, ele é classificado como método direto ou de ordem zero.

Quando utilizado no contexto de otimização, os Algoritmos Genéticos se distinguem dos métodos clássicos de Programação Matemática basicamente pelos seguintes aspectos:

- Emprega sempre uma população de indivíduos ou soluções.
- Opera com uma codificação das possíveis soluções (genótipos) e não com as soluções propriamente ditas (fenótipos).
- Trabalha com regras de transição probabilísticas.
- Não requer informações adicionais (derivadas, por exemplo) sobre a função a otimizar.

Deste modo, a busca por soluções pode se dar sobre conjuntos não-convexos e mesmo disjuntos, com funções objetivo também não-convexas e não-diferenciáveis, e simultaneamente com variáveis reais, lógicas e inteiras.

Outra grande diferença dos métodos clássicos para os Algoritmos Genéticos é que estes não se prendem tão facilmente a extremos locais, uma vez que se trabalha com uma população de indivíduos e se realiza a busca dentro de toda a região viável disponível.

5. Estrutura dos Algoritmos Genéticos

Existem inúmeras estruturas de Algoritmos Genéticos, com variações de procedimentos e operadores empregados, entretanto, as três estruturas mais encontradas dos algoritmos são [03, 54]:

- Algoritmo Genético “genérico”

Algoritmo Genético genérico

Inicialize a população de cromossomos (geração $i = 1$)

Avalie indivíduos na população (função objetivo e sobrevivência)

Repita (evolução)

 Selecione indivíduos para reprodução

 Aplique operadores de recombinação e/ou mutação

 Avalie indivíduos gerados na população

 Selecione indivíduos para sobreviver (geração $i = i + 1$)

Até objetivo final ou máximo de gerações

Fim

Figura 12 - Algoritmo genético genérico

Segundo a maneira pela qual os indivíduos criados são inseridos na população, são apresentados a seguir dois tipos extremos de Algoritmos Genéticos, os quais são conhecidos na literatura como: “*geracional*” e “em regime” (steady-state).

- Algoritmo Genético “*geracional*”

Neste extremo de algoritmo, “*toda*” a população é substituída por novos indivíduos gerados pelo processo de seleção e aplicação dos operadores genéticos:

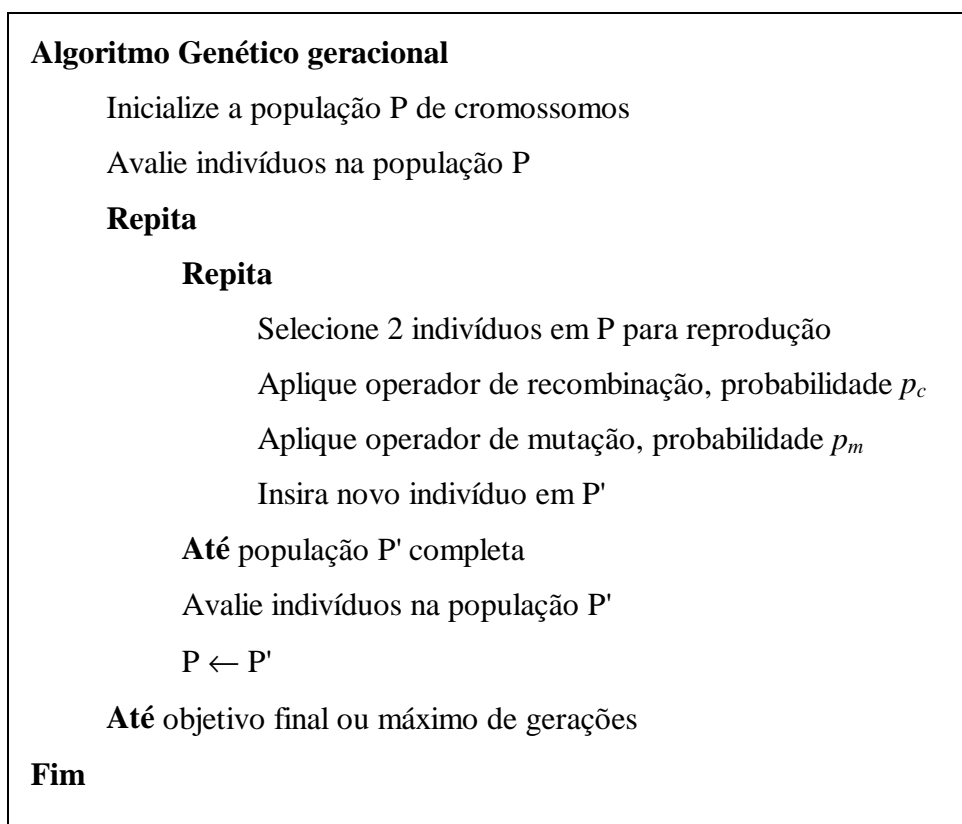


Figura 13 - Algoritmo genético geracional

Deve-se observar que como toda a geração (de “pais”) é integralmente substituída por outra mais nova (de “filhos”), não existe “convivência”, havendo assim a perda de bons indivíduos no processo. Por esta razão, especialmente em problemas de otimização, um procedimento freqüentemente empregado é o “*elitismo*”, ou seja, o(s) melhor(es) indivíduo(s) de uma geração é(são) preservado(s): uma cópia é passada diretamente para a geração seguinte.

- Algoritmo Genético “em regime”

Neste outro extremo de algoritmo, apenas “um” indivíduo é criado de cada vez e, depois de sua avaliação, ele será inserido na população em substituição a algum outro elemento, por exemplo, o pior de todos já existentes. Caso ele seja pior que todos os já existentes então nada é alterado e procede-se uma nova criação de indivíduo.

Com o intuito de facilitar a comparação do indivíduo gerado com os indivíduos já existentes na população, utiliza-se um ordenamento dentro da população, desta forma, o indivíduo gerado é comparado apenas com o último indivíduo do ordenamento, caso seja superior a ele, assumirá sua posição correspondente no ordenamento, sendo o último eliminado pela seleção natural.

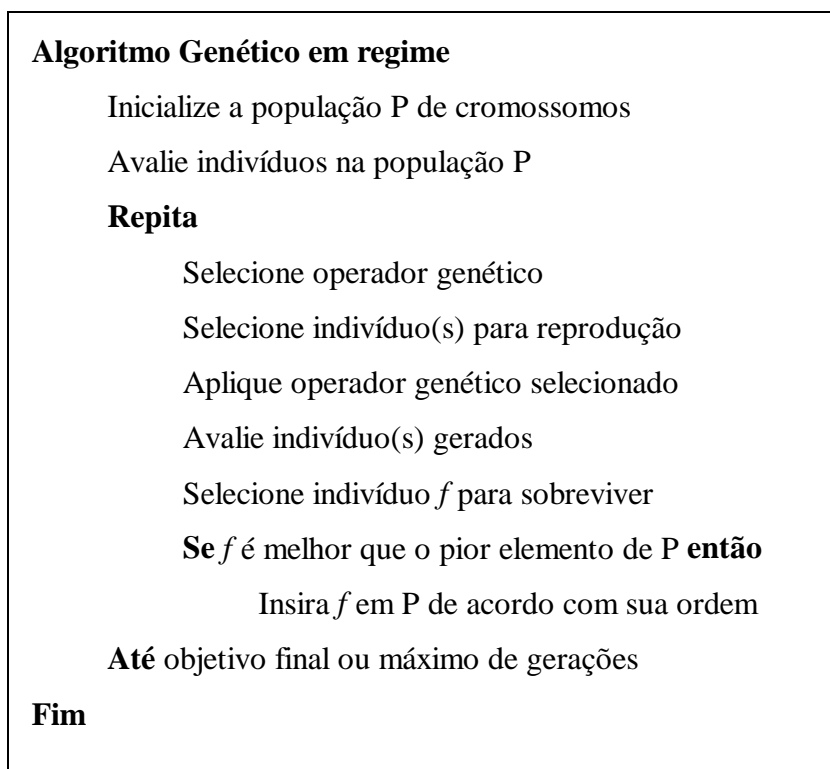


Figura 14 - Algoritmo genético “em regime” – steady-state

Entre estes dois extremos de algoritmos situam-se outros que somente uma parte da população é substituída: “pais” e “filhos” podem vir a coexistir. Outros esquemas de reprodução podem também ser encontrados na literatura.

6. Aspectos Principais dos Algoritmos Genéticos

6.1. Representação e Codificação

A representação ou codificação das variáveis de projeto a serem otimizadas proporciona um grande impacto no desempenho de busca, devendo ser o mais simples possível sem perder, no entanto, as características de representação do problema tratado.

Existem inúmeras formas de representação das variáveis, tais como: binária, números inteiros ou números reais.

A maioria dos trabalhos desenvolvidos utiliza a codificação binária, onde cada cromossomo é um vetor composto por zeros e uns, com cada bit representando um gene do mesmo.

Para problemas de otimização com variáveis reais, uma codificação binária pode ser introduzida pela conversão de valores numéricos de ponto flutuante para valores binários de comprimentos fixados, sempre se necessitando a posteriori realizar o caminho inverso para avaliação da função *de aptidão*. Existe assim, uma desvantagem no caso da representação binária, que seriam as constantes conversões entre valores de ponto flutuantes e binários.

Uma vez definidas as variáveis de projeto relevantes para a resolução do problema de otimização, bem como em qual forma tais variáveis serão representadas, deve-se concatenar as respectivas representações de maneira a formar o cromossomo.

Existe ainda a possibilidade de associar-se mais de um cromossomo a cada indivíduo, analogamente ao que ocorre na biologia, onde o Homo Sapiens, por exemplo, é diplóide, isto é, possui cromossomos aos pares.

6.2. Geração da População Inicial

A população inicial de indivíduos ou cromossomos é na maioria das vezes realizada de forma aleatória, embora existam ocasiões onde é mais apropriada uma seleção heurística da mesma, introduzindo logo de início, um ou mais indivíduos “interessantes”, como por exemplo, soluções aproximadas conhecidas contendo algum tipo de informação prévia. Diversos trabalhos realizados comprovam que a inicialização, em geral, não é crítica, desde que a população inicial contenha cromossomos suficientemente variados [07, 30].

6.3. Avaliação da População

A avaliação da população é realizada pela função *de aptidão*, que deve indicar a “qualidade” de cada indivíduo na população. Para problemas de otimização ela está intimamente ligada à função objetivo que se deseja extremizar. Para problemas de maximização, por exemplo, a função *de aptidão* pode ser tomada como uma modificação da função objetivo:

$$f(\mathbf{x}) = g(\mathbf{x}) + C$$

onde C é uma constante introduzida para os casos onde o processo de seleção exige que a *aptidão* seja positiva. Com o processo natural de evolução, os indivíduos passam a ter *aptidões* cada vez mais semelhantes entre si, podendo ser necessário aumentar a *pressão de seleção* pela adoção de alguma estratégia da seguinte forma:

$$f(\mathbf{x}) = h(g(\mathbf{x}))$$

com uma certa função h escolhida convenientemente.

Não existe a necessidade de se calcular as derivadas da função objetivo e, conseqüentemente, um algoritmo genético pode ser utilizado também sem nenhuma modificação para otimizar funções não diferenciáveis.

Outro fato interessante é que a *aptidão* pode variar ao longo do processo evolutivo, de maneira dinâmica e estocástica, imprecisa e mesmo subjetiva, no caso de algumas aplicações artísticas como música e pintura, por exemplo.

É preciso lembrar, entretanto, que a escolha da função *de aptidão* é para a maioria das aplicações a etapa crítica do processo, já que ela deverá ser avaliada para cada cromossomo de cada população dentro do processo evolutivo.

6.4. Esquemas de Seleção

O processo de seleção baseia-se no princípio da “sobrevivência dos melhores indivíduos”, onde os cromossomos com melhor aptidão ou adequabilidade recebem uma maior probabilidade de serem copiados para um novo conjunto, denominado população temporária, de onde serão aleatoriamente escolhidos para reprodução. Em contrapartida, os indivíduos com baixa aptidão serão descartados da população conforme a *pressão de seleção* do esquema utilizado.

Na prática computacional, esta população temporária não é literalmente formada, isto é, os indivíduos já são diretamente selecionados da população aos pares para a reprodução, sendo seus herdeiros implantados na próxima geração.

Inúmeros esquemas de seleção [05] já foram propostos e implementados na prática dos Algoritmos Genéticos, alguns não sendo biologicamente plausíveis. Na seqüência, alguns dos mais empregados são discutidos:

6.4.1. Seleção Proporcional à Aptidão via Método da Roleta

Este é o esquema original de seleção proposto por Holland [36], por esta razão, ele é ainda hoje muito utilizado nas implementações, apesar de apresentar algumas falhas relevantes de funcionamento que serão comentadas adiante.

Uma forma de quantificar a probabilidade p_i do i -ésimo indivíduo da população vir a ser selecionado para reprodução é o cálculo proporcional ao seu valor da função de *aptidão*, $f_i = f(x_i)$.

Uma possível implementação seria:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

onde f_i é assumida positiva e N é o tamanho da população.

Uma vez definida a forma de quantificação da probabilidade de sobrevivência de cada indivíduo da população, emprega-se um dos muitos métodos utilizados para a seleção dos indivíduos propriamente dita.

Um dos métodos largamente empregados para tal finalidade é o conhecido “método da roleta”, onde cada indivíduo da população é representado proporcionalmente ao seu índice de aptidão. Assim, os indivíduos com alta aptidão recebem uma porção maior da roleta, enquanto que os de baixa aptidão ocuparão uma porção relativamente menor. Deste modo, realiza-se o lançamento N vezes da roleta, dependendo do tamanho da população, e escolhe-se para a população temporária aqueles indivíduos por ela sorteados.

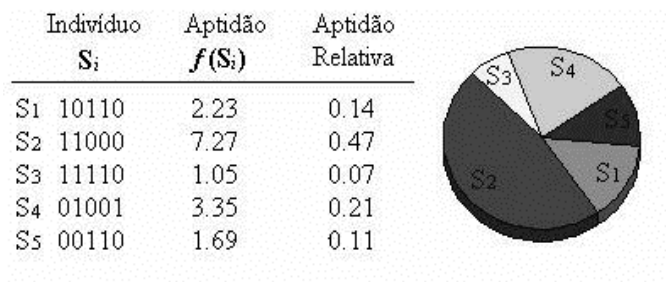


Figura 15 - Exemplo do Método da Roleta

O método da roleta tem a desvantagem de possuir uma alta variância, podendo levar a um grande número de cópias de um bom cromossomo, o que faz diminuir a diversidade da população. Esta falha pode ocasionar uma convergência prematura do algoritmo para uma solução não almejada. Por outro lado, quando a evolução está avançada, onde as aptidões não diferem muito entre si, observa-se uma estagnação do algoritmo, isto é, uma baixa pressão de seleção entre aptidões parecidas.

Este modelo também é fortemente dependente da escala da função de aptidão. Como exemplo, seja uma população de 10 indivíduos com valores de aptidão seqüenciais de 10 (melhor) a 1 (pior). A probabilidade de seleção do melhor indivíduo é $p_m \approx 16,6\%$ e a do pior $p_p \approx 1,5\%$. Caso ocorra uma translação (soma) de 100 unidades na função de aptidão de todos os indivíduos, as probabilidades serão agora: $p'_m \approx 10,4\%$ e $p'_p \approx 9,5\%$. Deste modo, observa-se que as probabilidades do melhor e do pior indivíduos se tornam quase idênticas.

Esta propriedade indesejável fez surgir o desenvolvimento de vários métodos de escalonamento, tais como: escalonamento estático linear, dinâmico linear, exponencial, logaritmo e truncamento sigma. Outro método utilizado para atenuar esta característica é realizar a seleção acima (*over selection*) de uma certa porcentagem dos melhores indivíduos, isto é, forçar que 80%, por exemplo, dos selecionados sejam tomados dos 20% melhores indivíduos da população.

6.4.2. Seleção Estocástica Remanescente sem Substituição

Este método parece realizar melhor o processo de reprodução e apresenta o seguinte procedimento:

Calcula-se diretamente o número esperado de cópias para cada cromossomo:

$$E_i = N \cdot p_i$$

onde N é o tamanho da população e a parte inteira de E_i , designada por $\text{int}(E_i)$, será o número esperado de cópias:

$$I_i = \text{int}(E_i)$$

deste modo, o cromossomo i será copiado I_i vezes na população temporária.

Calcula-se a seguir o resíduo:

$$R_i = E_i - I_i$$

que representa a probabilidade do cromossomo ser copiado novamente. Por exemplo, um cromossomo para o qual $E_i = 2.75$ será copiado duas vezes, e poderá ser copiado novamente com uma probabilidade de 0.75.

6.4.3. Seleção por Ordenação: Linear e Exponencial

Neste modelo os indivíduos são ordenados pelas suas aptidões, tal que: $f_i \geq f_j$ para $i < j$, ou seja, do melhor para o pior. Depois, a cada indivíduo é atribuída uma probabilidade de seleção p_i tomada de uma dada distribuição, sendo restringida para que: $\sum p_i = 1$.

As distribuições mais comuns são respectivamente a linear, $p_i = a.i + b$ ($a < 0$), e a exponencial, $p_i = a \exp(b.i+c)$, conforme os gráfico abaixo, correspondentes ao exemplo da página 46:

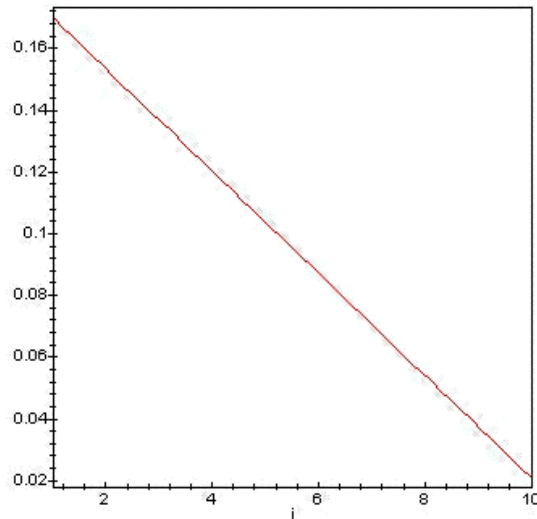


Figura 16 - Distribuição Linear para Seleção via Ordenamento

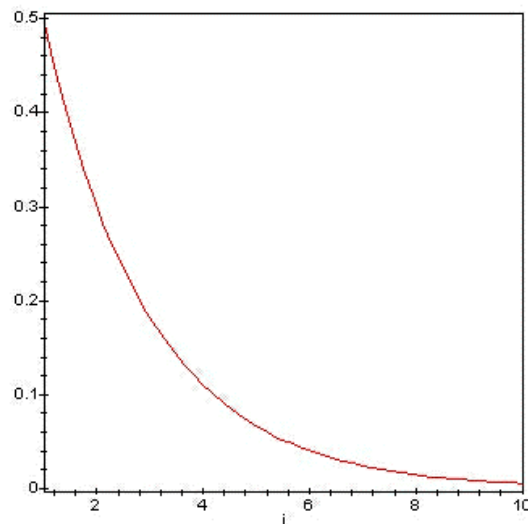


Figura 17 - Distribuição Exponencial para Seleção via Ordenamento

A distribuição exponencial equivale a selecionar o primeiro indivíduo com probabilidade p , o segundo com probabilidade p^2 , o terceiro com p^3 e assim por diante. Lembrando que $p < 1$ e $p^N < p^{N-1}$.

As principais vantagens do modelo de seleção via ordenamento são:

- Não causa convergência prematura, já que nenhuma solução tem $N_i \gg I$.
- Não permite estagnação, porque mesmo quando a evolução está bastante avançada $N_1 \neq N_2 \neq \dots$
- Não é necessária uma aptidão explícita, pois para ordenar os indivíduos apenas a habilidade de comparação dos mesmos é suficiente.

Como desvantagens destacam-se:

- Esforço computacional para ordenamentos constantes.
- Difícil análise teórica de convergência.

6.4.4. Seleção por Torneio

Este é um dos modelos mais simples para implementação computacional com bons resultados encontrados.

A idéia é promover um torneio entre um grupo de N ($N \geq 2$) indivíduos aleatoriamente tomados da população. Assim, o indivíduo com o maior valor de aptidão no grupo é selecionado, enquanto que os demais são descartados.

Dentre suas vantagens citam-se:

- Não acarreta convergência prematura.
- Combate à estagnação.
- Nenhum esforço computacional extra é necessário, tal como ordenamentos.
- Aptidão explícita é desnecessária.
- Inspiração biológica do processo.

6.4.5. Seleção Elitista

O modelo de seleção elitista normalmente é acoplado a outros métodos de seleção, na tentativa de se aumentar a velocidade de convergência do algoritmo, bem como em aplicações onde possa ser necessário o seu emprego.

Este processo simplesmente copia os N ($N \geq 1$) melhores indivíduos da população corrente para a próxima geração, garantindo que estes cromossomos não sejam destruídos nas etapas de recombinação e mutação.

Na maioria das implementações, normalmente, pelo menos o elitismo do melhor indivíduo é utilizado.

A principal vantagem do elitismo é garantir que não se perca o melhor indivíduo durante o processo evolutivo e, com isto, gerar uma sequência monotônica, onde o resultado não piora.

Sua desvantagem é a possibilidade de forçar a busca, pela presença de mais uma cópia do melhor indivíduo, na direção de algum ponto ótimo local que tenha sido descoberto antes do global, embora um algoritmo genético possa escapar de tais armadilhas.

Uma alternativa é guardar separadamente a melhor solução encontrada durante a evolução, para no final da execução designá-la como o indivíduo ótimo encontrado, mesmo que ele não esteja presente na última geração da execução.

6.5. Reprodução ou Cruzamento

O processo de seleção não introduz novos indivíduos na população temporária, apenas os chamados genitores, que servirão como pais para a nova geração, composta pelos filhos.

Na etapa de reprodução, o algoritmo tenta criar novas e melhores soluções (indivíduos mais aptos). Sendo a forma mais simples e natural de cruzamento a chamada “cruzamento em um ponto”, descrito a seguir.

Como passo inicial, toda a população temporária é agrupada aleatoriamente por pares para gerar um conjunto de $N / 2$ genitores potenciais.

O cruzamento é aprovado por um processo de decisão semelhante ao de um “lançamento de moeda”. Para isto, deve-se escolher uma probabilidade de cruzamento P_c igual para cada par. O lançamento de moeda corresponde aqui a gerar, para cada par, um valor aleatório entre 0 e 1. Logo, pela comparação do valor gerado de cada par com a probabilidade P_c , defini-se se o cruzamento é ou não permitido da seguinte forma: se o valor gerado pelo lançamento for inferior a P_c , o cruzamento é permitido, caso contrário, os genitores são mantidos inalterados.

6.6. Operadores Genéticos

O princípio básico dos operadores genéticos é transformar a população através de sucessivas gerações, para obter um resultado satisfatório no final do processo. Deste modo, eles são extremamente necessários para que a população se diversifique e mantenha as características de adaptação adquiridas pelas gerações anteriores.

6.6.1. Operadores de Recombinação

Ao cruzar pelo menos dois pais, uma ou mais novas soluções são criadas intercambiando-se a informação genética dos genitores em um ou mais pontos que também são selecionados aleatoriamente.

Este operador, também conhecido como *crossover*, pode ser utilizado de várias maneiras, onde as mais empregadas são:

- **um-ponto** → um ponto de cruzamento é escolhido e a partir dele as informações genéticas dos pais são trocadas, conforme exemplo abaixo.
- **multi-pontos** → é uma generalização da idéia de troca de material genético, onde muitos pontos de cruzamento podem ser utilizados.
- **uniforme** → não utiliza pontos de cruzamento, mas determina através de um parâmetro global, probabilidade de máscara, qual a probabilidade de cada variável ser herdada de cada pai.

A figura seguinte mostra um exemplo de dois cromossomos (escuro e claro) realizando o cruzamento no ponto 4. Assim, os novos cromossomos substituirão os genitores na nova população.

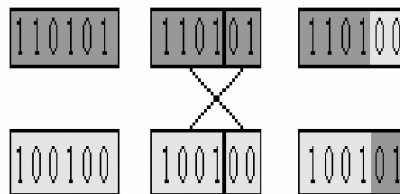


Figura 18 - Exemplo de Recombinação

6.6.2. Operadores de Mutação

O operador de mutação é necessário para a introdução e manutenção da diversidade genética na população, alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, como ilustrado na figura abaixo. Ele fornece assim, meios para a introdução de novos elementos na população, assegurando que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca seja zero, com o intuito de tentar contornar o problema de ótimos locais.

O operador de mutação é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação P_m . Geralmente se utiliza uma taxa de mutação pequena (como na genética natural), pois é um operador genético secundário.

Antes da Mutação: 1 1 1 0 0
Depois da Mutação: 1 1 0 0 0

Figura 19 - Exemplo de Mutação

7. Parâmetros de Influência e Configurações

A configuração correta dos parâmetros de influência é, sem dúvida, um dos aspectos mais relevantes dentro da estratégia dos Algoritmos Genéticos. Não existe muita literatura sobre este tema, uma vez que tais configurações irão depender entre outras coisas da aplicação resolvida, entretanto, é intuitivo que este passo é de muita importância para um bom desempenho do mecanismo de busca.

A eficiência e o funcionamento de um Algoritmo Genético é altamente dependente dos seus parâmetros de controle, sendo os básicos aqueles descritos a seguir.

7.1. Tamanho da População - N

O tamanho da população indica o número de cromossomos em cada população, normalmente constante durante a evolução.

Uma grande população apresentará uma maior diversidade de soluções, contudo, computacionalmente serão dispendiosas tantas avaliações de funções *de aptidão*. Assim, a principal influência deste parâmetro está relacionada com o desempenho global e a eficiência dos AGs.

Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Por outro lado, uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações são necessários maiores recursos computacionais ou que o algoritmo trabalhe por um período de tempo maior.

Uma idéia interessante é relacionar o tamanho da população com o tamanho do cromossomo, ou seja, quanto maior for o cromossomo maior deverá ser o tamanho da população para uma diversidade razoável.

Muitos pesquisadores sugerem a título de grandeza, tamanhos de população entre 20 e 100 cromossomos, podendo atingir valores bem mais altos no caso da programação genética.

7.2. Taxa ou Probabilidade de Cruzamento - P_c

Este parâmetro indica com qual taxa ou probabilidade irá ocorrer o cruzamento entre indivíduos selecionados na população.

Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Em contrapartida, se ela for muito alta, estruturas com boas aptidões poderão ser retiradas mais rapidamente da população, ocorrendo perda de estruturas de alta aptidão. Valores baixos podem ainda tornar a convergência do algoritmo muito lenta.

Geralmente, a taxa de cruzamento varia entre 0.5 e 0.95. Mas estes números indicam apenas uma ordem de grandeza, já que existem inúmeros tipos possíveis de cruzamentos, os quais são limitados apenas pela capacidade criativa de seus inventores.

7.3. Taxa ou Probabilidade de Mutação - P_m

A taxa de mutação indica a probabilidade ou taxa em que haverá a mutação de cromossomos nas populações ao longo da evolução.

A mutação é empregada para fornecer novas informações dentro das populações, prevenindo que as mesmas se tornem saturadas com cromossomos similares à medida que visa aumentar a diversidade populacional e possibilita ainda uma maior varredura do espaço de busca.

Há de se tomar cuidado, pois com uma taxa muito alta a busca pode se tornar essencialmente aleatória.

Alguns pesquisadores recomendam a escolha da taxa de mutação com base no tamanho dos cromossomos e das populações.

De Jong [22] sugere que a taxa de mutação deva ser inversamente proporcional ao tamanho da população.

Hesser e Manner [59] sugerem que uma taxa ótima de mutação pode ser achada pela expressão:

$$P_m = (N.L^{1/2})^{-1}$$

sendo N o tamanho da população e L o comprimento dos cromossomos.

Como os demais parâmetros, a taxa de mutação ideal dependerá da aplicação a ser resolvida, todavia, a maioria das taxas utilizada varia entre 0.001 e 0.1.

7.4. Recomendações de Autores

De Jong faz algumas recomendações com base na observação do desempenho de 5 exemplos de otimização via Algoritmos Genéticos, os quais incluem aspectos complicantes como descontinuidades, alta dimensão e ruído. Ele sugere para um desempenho satisfatório de uma vasta gama de problemas a seguinte configuração de parâmetros [22]:

$$(N, P_c, P_m) = (50, 0.6, 0.001)$$

Grefenstette [29] em simulações similares concluiu que para um melhor desempenho, quando a média da função objetivo de cada geração é usada como o índice a ser otimizado, devem ser adotados os seguintes números:

$$(N, P_c, P_m) = (30, 0.95, 0.01)$$

Já onde o índice a otimizar é a função objetivo do melhor cromossomo na população, normalmente o indicador mais usado para rotinas de otimização, ele recomenda os seguintes valores:

$$(N, P_c, P_m) = (80, 0.45, 0.01)$$

8. Vantagens e Desvantagens dos Algoritmos Genéticos

8.1. Vantagens dos Algoritmos Genéticos

- São robustos e aplicáveis a uma grande variedade de problemas.
- Não requerem conhecimentos ou informações dos gradientes da superfície definida pela função objetivo.
- Descontinuidades ou complexidades presentes na superfície acarretam pouco ou nenhum efeito no desempenho da busca.
- São mais resistentes a se prenderem a ótimos locais.
- Apresentam um bom desempenho para uma grande escala de problemas.
- São de fácil implementação e proporcionam maior flexibilidade no tratamento do problema a ser resolvido.

8.2. Desvantagens dos Algoritmos Genéticos

- Dificuldade para achar o ótimo global exato.
- Requerem um grande número de avaliações de funções *de aptidão*.
- Grandes possibilidades de configurações que podem complicar a resolução do problema tratado.

9. Sumário e Discussão

Embora no presente trabalho tenha-se dado uma maior abordagem da utilização dos Algoritmos Genéticos para a resolução de problemas de otimização, outras aplicações desta técnica são possíveis e tem sido utilizadas, tais como:

- gerenciamento de carteiras de ações, desenvolvimento de estratégias para leilões, detecção de fraudes em movimentos na bolsa de valores, dentre outras na área de economia e finanças;
- modelagem e estudo de problemas em Ecologia, genética populacional e na relação entre aprendizagem do indivíduo e evolução da espécie;
- construção automática de programas para a realização de tarefas específicas e outras estruturas computacionais como redes neurais;
- criação de imagens, texturas, músicas e estruturas complexas;
- mineração de dados (Data Mining);

Pelas suas características de robustez, flexibilidade e relativa facilidade de implementação, acredita-se que os Algoritmos Genéticos irão ganhar uma maior atenção com o decorrer do tempo, principalmente, pela rápida evolução dos computadores que deverão tornar as aplicações desta técnica cada vez mais viáveis e engenhosas.

Para os próximos anos um maior número de alternativas “mistas” da idéia genética com outras estratégias, que lhe agreguem novas capacidades, serão desenvolvidas, aprimoradas e difundidas no meio técnico, aumentando o leque de aplicações e trazendo novos benefícios na resolução dos problemas já tratados.

Para problemas de otimização as vantagens do emprego de Algoritmos Genéticos são consideráveis, principalmente pela sua versatilidade na obtenção de soluções ótimas globais, enquanto espera-se que suas desvantagens sejam sanadas por uma maior consolidação da técnica e pelo avanço das capacidades computacionais.

Capítulo V

Tópicos Adicionais em AGs

Neste capítulo são discutidos alguns temas e estratégias relacionados com a técnica dos algoritmos genéticos que podem facilitar, agilizar e melhorar o seu funcionamento.

1. Tratamento das Restrições	62
2. Problemas com Múltiplos Objetivos	62
3. Hibridização	63
4. Computação Paralela	64
5. Time de Algoritmos	65
6. Adaptação.....	66
7. Distribuição Uniforme da População Inicial.....	66

1. Tratamento das Restrições

Embora os Algoritmos Genéticos tenham sido inicialmente aplicados com sucesso em problemas de otimização sem restrições, foi uma questão de tempo para surgirem as primeiras utilizações da técnica na resolução de problemas com restrições, caracterizados pela seguinte formulação:

$$\begin{aligned} &\text{Extremize } f(x_1, x_2, \dots, x_N) \\ &\text{sujeito a } \quad g_j(\mathbf{x}) \geq 0 \quad j = 1, 2, \dots, J \\ &\quad \quad \quad h_k(\mathbf{x}) = 0 \quad k = 1, 2, \dots, K \\ &\quad \quad \quad x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, N \end{aligned}$$

Na maior parte das aplicações é empregada a técnica de penalização para introduzir as restrições do problema na própria função objetivo, técnica esta também utilizada por muitos métodos matemáticos de otimização. Todavia, já existem formas de abordagem para este problema sem o emprego de funções de penalização, bem como outras alternativas mais inovadoras tal como algoritmos co-evolucionários que preservam um maior número de populações interagindo entre si. Estes estudos são bem recentes e proporcionam uma vasta área de pesquisa a ser desenvolvida [50].

2. Problemas com Múltiplos Objetivos

Vários problemas reais são formulados por uma coleção de objetivos a serem extremizados, os quais são muitas vezes conflitantes entre si. Porém, em muitas destas aplicações são realizadas simplificações no problema original de forma a combinar vários objetivos numa única função. Ou ainda escolhendo-se aquele que dentre os objetivos do problema requer prioridade de extremização, normalmente o custo.

Nos últimos anos vêm aumentando os estudos na área da otimização multi-objetivos, trazendo como consequência o desenvolvimento de métodos matemáticos para tal tarefa.

Um exemplo [53] de aplicação multi-objetivos é o projeto de pontes, pois além da minimização do custo ou peso é de fundamental importância a maximização das frequências naturais de vibração da estrutura para melhorar o seu desempenho dinâmico. Observa-se que estes objetivos são conflitantes, já que para menores pesos, menores serão as frequências naturais de vibração e maiores os riscos de colapso estrutural por efeito de ressonância.

Um método matemático utilizado para a otimização multi-objetivos é a Programação Objetiva [38, 39, 53], que é um procedimento iterativo para o usuário e que tem tido crescente utilização, embora, apesar de se tratar de um método matemático, não apresenta garantias da obtenção de extremos globais, além de ser um método hierárquico com solução única.

Como os Algoritmos Genéticos trabalham com uma população de soluções, eles são naturalmente adequados para a determinação simultânea de várias soluções Pareto-ótimas, isto é, soluções não-dominadas [37].

3. Híbridização

Uma alternativa para melhorar o funcionamento dos Algoritmos Genéticos é a híbridização [24, 68], estratégia que visa acoplar algoritmos distintos com a finalidade de tirar o melhor proveito de cada um deles. Assim, é possível acoplar-se aos Algoritmos Genéticos, por exemplo, qualquer outro método matemático para efetuar uma busca local mais agressiva. Estas alternativas híbridas resultantes apresentam a capacidade evolucionária de uma grande exploração global da região viável aliada a uma boa eficiência nas buscas locais.

4. Computação Paralela

Os Algoritmos Genéticos são por si só um método naturalmente paralelo, uma vez que a etapa de avaliação da *aptidão* de cada indivíduo da população é independente das demais. Por esta razão, empregar computação paralela com Algoritmos Genéticos é uma estratégia que deve sempre ser considerada, pois pode-se obter grandes ganhos no tempo de execução de um problema [07, 24].

Quando o paralelismo é natural como o descrito acima, o resultado encontrado é o mesmo obtido pelo Algoritmo Genético seqüencial que lhe deu origem.

Pode-se também implementar estratégias mais ousadas para se efetuar tanto a seleção quanto a criação de novos indivíduos localmente em cada processador, visando com isto a diminuição das comunicações (envio e recebimento de dados). Nestas alternativas não está garantida a obtenção das mesmas soluções encontradas seqüencialmente, tratando-se de um novo Algoritmo Genético.

Uma outra idéia é a subdivisão da população original de p indivíduos em n sub-populações “residentes” nos n processadores disponíveis, sendo cada uma submetida a um Algoritmo Genético independente. Caso não ocorra nenhuma comunicação entre os processadores tudo acontecerá como se o problema estivesse sendo resolvido n vezes.

Por outro lado, ao se permitirem algumas comunicações entre os processadores, como por exemplo a migração de alguns indivíduos de certa sub-população para outra(s), obter-se-á um novo Algoritmo Genético. Para estes casos, novos parâmetros de controle devem ser adicionados com o intuito de definir “quando, quais, quantos e para onde” os indivíduos irão migrar, bem como de que modo serão inseridos na nova população.

5. Time de Algoritmos

A estratégia de times de algoritmos é uma das mais recentes e, de certa forma, compreende parte das duas anteriores. A idéia é resolver o problema com mais de um algoritmo como na técnica de hibridização, diferindo desta pelo fato de que ao invés de cada algoritmo entrar em operação conforme o desenvolvimento do problema, todos tentam resolvê-lo separadamente em paralelo, podendo os algoritmos comunicarem-se entre si conforme finalidades pré-estabelecidas: a cada intervalo de tempo, a cada número de iterações, a cada critério de convergência, etc., de acordo com as características do problema e dos algoritmos empregados.

A comunicação entre os algoritmos pode ser feita de maneira síncrona ou assíncrona, sendo esta última de maior eficiência já que não seriam necessários balanceamentos de carga e não haveria ociosidade dos processadores.

Experimentos realizados com times de algoritmos levaram a conclusões surpreendentes e que mostram, por exemplo, que certo problema que não podia ser resolvido através de quaisquer dos algoritmos isolados o foi pelo time formado por eles. Observou-se ainda que em algumas simulações o tempo e/ou número de iterações foi menor que o obtido pelo mais rápido dos algoritmos trabalhando sozinho. Estas conclusões podem ser verificadas para a resolução de sistemas de equações não lineares utilizando o método de Newton e o método dos gradientes conjugados nas referências [01, 02].

A estratégia de times é bem geral e pode ser empregada com um número variado de métodos em uma vasta gama de aplicações, sendo mais uma alternativa para o emprego dos Algoritmos Genéticos na resolução de problemas.

6. Adaptação

A técnica de adaptação consiste em não tomar valores fixos para os parâmetros de configuração dos algoritmos genéticos. Ou seja, pode-se escolher funções mais elaboradas que façam estes parâmetros variarem ao longo do processo evolutivo e/ou de acordo com critérios pré-definidos.

As principais finalidades da adaptação são:

- melhorar a eficiência dos algoritmos;
- tornar os algoritmos mais robustos às aplicações a serem tratadas.

A adaptação usualmente é feita de maneira estática, isto é, são escolhidos parâmetros, funções e/ou operadores no início do algoritmo e levados até o fim da execução. Todavia, recentemente uma maior atenção vem sendo dada à adaptação dinâmica, que permite que dentro do processo evolutivo o próprio algoritmo selecione os parâmetros de configuração e/ou os operadores genéticos a serem utilizados, conforme seus desempenhos.

A própria função objetivo pode ser adaptada, principalmente em aplicações que envolvam o emprego de muitas restrições e/ou penalidades [34, 49].

7. Distribuição Uniforme da População Inicial

Esta estratégia distribui uniformemente a população inicial pelo espaço de busca, ganhando assim uma maior confiança na obtenção de extremos globais e até mesmo melhorando a velocidade de convergência do algoritmo [61].

Capítulo VI

Algoritmos Genéticos Multi-objetivos

Uma introdução sobre os principais algoritmos evolucionários multi-objetivos é apresentada neste capítulo, bem como suas principais características e inovações no cenário da otimização multi-objetivos.

1. Introdução	68
2. Alguns Algoritmos Evolucionários Multi-objetivos	69

1. Introdução

A primeira implementação prática para tratamento de problemas multi-objetivos foi apresentada por Schaffer em 1984 [57]. Após este trabalho nenhum estudo significativo foi apresentado por quase uma década, à exceção de um revolucionário esboço de 10 linhas de um novo procedimento de ordenamento de soluções não-dominadas apresentado no livro de Goldberg [30]. Este livro tornou-se um marco dos algoritmos evolucionários e alavancou o tratamento de problemas de otimização multi-objetivos. Muitos pesquisadores desenvolveram diferentes versões de algoritmos de otimização multi-objetivos baseados nas idéias conceituais apresentadas por Goldberg.

Os métodos evolucionários apresentam certas características que os tornam mais apropriados para a resolução dos problemas multi-objetivos, principalmente quando se deseja conhecer o conjunto das soluções ótimas de Pareto. Por este motivo, o restante deste trabalho enfoca principalmente esta linha de pesquisa.

Duas são as finalidades quando se deseja determinar o conjunto de Pareto de problemas multi-objetivos via métodos evolucionários:

1. guiar a busca na direção da região ou conjunto ótimo de Pareto;
2. manter a diversidade da população na fronteira de Pareto.

Estas duas tarefas são as medidas usualmente empregadas para avaliar a eficiência e desempenho dos algoritmos desenvolvidos. O papel da segunda meta é tão relevante quanto o da primeira, pois fornece ao projetista maior alternativa de escolha para o projeto, já que é possível encontrar um conjunto ótimo de Pareto com uma boa e contínua distribuição de soluções.

2. Alguns Algoritmos Evolucionários Multi-objetivos

Nos últimos anos muitos pesquisadores têm modificado as idéias iniciais propostas por Goldberg em seu livro [30] para tratamento de problemas multi-objetivos, bem como aplicado as implementações desenvolvidas em problemas mais complexos do mundo real. Assim, atualmente existem inúmeras implementações e, por este motivo, a enumeração de todas é uma tarefa impossível, logo, serão enumeradas e posteriormente apresentadas as características fundamentais apenas das principais correntes e/ou implementações no campo dos algoritmos evolucionários multi-objetivos [16, 20, 69, 70, 72]:

- **VEGA**, Schaffer;
- Agregação dos objetivos por pesos variáveis, Hajela e Lin;
- **MOGA**, Fonseca e Fleming;
- **nPGA**, Horn e Nafpliotis;
- **NSGA**, Srinivas e Deb;
- **SPEA**, Zitzler e Theile;
- Outros algoritmos.

2.1. VEGA, Schaffer

Tido como o pioneiro na implementação de algoritmos evolucionários para solução de problemas multi-objetivos, Schaffer desenvolveu o chamado “Vector Evaluated Genetic Algorithms” [57], mais conhecido como VEGA. Schaffer modificou o software de domínio público GENESIS pela criação de um laço no procedimento de seleção original que faz com que o procedimento seja repetido para cada objetivo separadamente até atingir-se um determinado número pré-definido de indivíduos para cada objetivo para reprodução. Em seguida, estes indivíduos são randomicamente sorteados para as etapas de recombinação e mutação. Schaffer implementou este método em combinação com o procedimento de seleção proporcional à aptidão dos indivíduos.

O algoritmo funcionou eficientemente por algumas gerações, porém em alguns casos deixou de pesquisar alguns indivíduos ou regiões. A seleção independente dos indivíduos provocou a especialização da população e como resultado deste fato a população inteira convergiu na direção da região das soluções ótimas individuais após um grande número de gerações. Esta característica de especialização não é interessante, já que não adianta uma solução apresentar alta qualidade em um objetivo se conseguida às custas de valores ruins ou inaceitáveis para algum(uns) outro(s) objetivo(s). É importante haver e destacar o compromisso entre os objetivos.

Schaffer tentou minimizar os efeitos da especialização através do desenvolvimento de dois procedimentos heurísticos de seleção que foram denominados: seleção não-dominada e seleção cruzada ou de companheiro.

Na seleção heurística não-dominada os indivíduos dominados são penalizados pela subtração de uma pequena penalidade fixa sobre o número esperado de cópias durante a seleção. A penalidade total sobre estes indivíduos é então dividida entre os não-dominados por uma adição ao número esperado de cópias na seleção. Contudo, este algoritmo não funciona adequadamente quando a população tem poucos indivíduos não-dominados, resultando num grande valor de aptidão para os mesmos e conseqüentemente numa alta pressão de seleção.

A seleção heurística cruzada promove o cruzamento de indivíduos especializados de diferentes subgrupos. Este procedimento foi implementado pela seleção aleatória de um indivíduo para reproduzir com seu companheiro, isto é, o indivíduo que tem a máxima distância Euclidiana em relação ao indivíduo anteriormente selecionado. Este procedimento também não funciona satisfatoriamente por não prevenir a participação de indivíduos piores na primeira seleção randômica e pela possibilidade de haver uma grande distância Euclidiana entre um indivíduo campeão e um medíocre.

Depois destas duas tentativas Schaffer concluiu que os procedimentos tradicionais randômicos de seleção são largamente superiores aos seus procedimentos heurísticos.

2.2. Agregação dos objetivos por pesos variáveis, Hajela e Lin

Um outro algoritmo que é baseado na agregação dos objetivos foi introduzido por Hajela e Lin [33]. Neste procedimento é usado o método da soma dos produtos entre os objetivos e seus correspondentes pesos para o cálculo da função de aptidão: $(f = \sum w_i f_i)$. Normalmente os valores dos objetivos devem ser normalizados para o caso de magnitudes muito diferentes, embora neste estudo a normalização não tenha sido empregada devido à natureza dos problemas teste usados.

Os pesos variam no intervalo $]0, 1[$ tal que o somatório de todos resulte na unidade. Para buscar por múltiplas soluções em paralelo os pesos não são fixos e sim codificados no próprio genótipo. Os autores enfatizam a necessidade de restrições na reprodução para promover: velocidade de convergência e estabilidade para a busca genética.

2.3. MOGA, Fonseca e Fleming

Fonseca e Fleming [26] implementaram as sugestões de Goldberg [30] de um modo diferente, o “Multi-objective Optimization Genetic Algorithm” ou MOGA usa um procedimento de ordenamento não-dominado. Toda a população é verificada e todos os indivíduos não-dominados recebem uma posição ou ordem 1. Outros indivíduos são posicionados segundo a não dominância deles em relação ao restante da população do seguinte modo: para cada indivíduo, o número de soluções que o dominam estritamente é primeiramente determinado na população, logo, a posição no ordenamento deste indivíduo será este número mais 1. Assim sendo, no final deste procedimento de ordenamento poderão existir muitos indivíduos compartilhando a mesma posição no ordenamento. A rotina de seleção usa este ordenamento para selecionar ou remover blocos de pontos até escolher os

indivíduos para reprodução. Este tipo de atribuição de aptidão por blocos leva a produzir uma grande pressão de seleção que pode causar convergência prematura.

A implementação também usa o método de formação de nichos para distribuir a população através da região ótima de Pareto, além de compartilhar os valores da função de aptidão. Embora isto mantenha a diversidade nos valores da função de aptidão, pode não manter a diversidade no conjunto das variáveis, assim, o MOGA pode não estar apto a achar as múltiplas soluções em problemas onde diferentes pontos ótimos de Pareto correspondem para os mesmos valores de aptidões. O destaque relevante deste trabalho é o ordenamento dos indivíduos por critérios de dominância.

2.4. nPGA, Horn e Nafpliotis

Horn e Nafpliotis [37] implementaram um algoritmo genético geracional com sobreposição, isto é, nem todos os indivíduos são substituídos de uma geração para outra que foi denominado “Niche Pareto Genetic Algorithm” ou nPGA, que faz uso de um procedimento de ordenamento na etapa de seleção.

Um conjunto de comparação compreendido de um número específico de indivíduos (t_{dom}) é tomado randomicamente da população no início de cada processo de seleção. Em seguida dois indivíduos são retirados da população para a seleção de um vencedor conforme o seguinte procedimento: ambas soluções são comparadas com os membros deste conjunto de comparação para determinação da dominância segundo as funções objetivo. Se um deles é não-dominado e o outro é dominado, então o ponto não-dominado é selecionado, mas caso ambos sejam não-dominados ou dominados, um contador de nicho é criado para cada indivíduo na população inteira. O contador é baseado no número de soluções na população com uma certa distância (σ_{share}) do indivíduo. Assim, a solução que apresentar o menor contador de nicho é selecionada.

O sucesso deste algoritmo é altamente dependente do parâmetro t_{dom} . Se um tamanho apropriado for escolhido, pontos não-dominados (ótimos de Pareto) podem ser achados. Caso t_{dom} seja pequeno podem existir poucos pontos não-dominados na população e valores grandes de t_{dom} podem levar à convergência prematura do algoritmo.

O conceito de formação de nichos entre os pontos não-dominados é o aspecto mais relevante do trabalho, que tem como inconveniente a introdução de mais parâmetros a serem configurados para o funcionamento do algoritmo genético.

2.5. NSGA, Srinivas e Deb

“Nondominated Sorting Genetic Algorithm” [64] ou simplesmente NSGA é mais uma implementação baseada nas sugestões de Goldberg [30]. A idéia por trás do algoritmo é a utilização de um procedimento de seleção por ordenamento ou ordenamento para enfatizar as soluções não-dominadas correntes, juntamente com um método voltado para a criação de nichos para manter a diversidade da população.

A diferença desta implementação em relação a um algoritmo genético simples está apenas no modo com que o operador de seleção é empregado. Tanto o operador de recombinação quanto o operador de mutação são os usuais da técnica.

Antes do procedimento de seleção ser aplicado, a população é ordenada com base num nível de não-dominância dos indivíduos, isto é, todas as soluções não-dominadas da população corrente recebem valores altos de aptidão. Esta aptidão é a mesma para todos os indivíduos não-dominados, garantindo assim que todos possuam um mesmo potencial reprodutivo.

Para manter a diversidade na população as soluções não-dominadas compartilham os seus valores de aptidão segundo suas distâncias Euclidianas (também pode ser adotada a distância de Hamming - número de bits diferentes entre dois indivíduos). Finalmente, divide-se o valor da aptidão de cada indivíduo pelo contador de nichos que é proporcional ao número de vizinhos ao seu redor. Este procedimento proporciona a co-existência de pontos ótimos múltiplos na população. O pior valor de aptidão compartilhada na solução da primeira fronteira não-dominada é então guardada para uso posterior.

Depois que o compartilhamento é executado e que as aptidões são modificadas os indivíduos não-dominados são ignorados temporariamente para processar o resto dos membros da população.

O procedimento para determinar novas soluções não-dominadas (segundo nível) é novamente executado, sendo que agora eles recebem um valor de aptidão um pouco menor que o pior valor de aptidão compartilhada no nível anterior.

Uma vez mais o procedimento de compartilhamento é executado entre as soluções não-dominadas do segundo nível e as novas aptidões são calculadas como antes. Este processo é continuado até que todos os membros da população tenham um valor de aptidão compartilhada.

A reprodução da população é efetuada utilizando-se a aptidão compartilhada, ou seja, como o primeiro nível de soluções não-dominadas possui as mais altas aptidões, um maior número de cópias dos seus indivíduos será realizado e levará a busca para a fronteira ótima de Pareto.

A característica mais importante deste trabalho é que praticamente qualquer número de objetivos pode ser usado para os dois tipos de problemas: maximização ou minimização, bastando mudar o modo como os indivíduos não-dominados são identificados.

Como fraqueza este algoritmo exige alguns novos parâmetros de configuração que são sensíveis e provocam forte influência no desempenho da busca.

2.6. SPEA, Zitzler e Theile

Recentemente foi sugerido por Zitzler e Theile [69] o “Strength Pareto Evolutionary Algorithm” ou SPEA, um algoritmo evolucionário multi-objetivo elitista com conceitos de não dominância.

O algoritmo funciona com a manutenção de uma população externa a cada geração que armazena um conjunto de soluções não-dominadas determinado desde a população inicial e que participa nas operações genéticas.

A aptidão de cada indivíduo na população corrente e na população externa é decidida com base no número de soluções dominadas pelo seguinte procedimento. Inicialmente, uma população combinada pela população corrente e a externa é construída. A seguir, todas as soluções não-dominadas nesta população recebem um valor de aptidão baseado no número de soluções que elas dominam, mantendo a diversidade. Toma-se o cuidado de não atribuir para as soluções não-dominadas uma aptidão pior que o das melhores soluções dominadas, para garantir que a busca caminhe na direção da fronteira não-dominada e simultaneamente a diversidade entre os indivíduos dominados e não-dominados.

No problema da mochila os autores obtiveram melhores resultados que qualquer outro método utilizado nas comparações de seus estudos, o que reforça a importância do elitismo no estudo de problemas de otimização multi-objetivos.

Como destaque do método cita-se a ausência de quaisquer parâmetros de distância, tais como raio de nicho e compartilhamento e o fato da aptidão dos indivíduos ser determinada apenas pelas soluções armazenadas no conjunto de Pareto externo.

2.7. Outros algoritmos

Vários outros algoritmos têm sido desenvolvidos e publicados durante os últimos anos, tais como:

Ishibuchi e Murata [40] introduziram uma combinação de um algoritmo evolucionário que é baseado na soma dos objetivos ponderados por pesos com um algoritmo de busca local que previne a perda de soluções não-dominadas pelo armazenamento delas externamente. Depois de atualizar o conjunto de Pareto externo são escolhidos pares de indivíduos na fase de seleção de acordo com pesos randomicamente gerados, então, uma fração do conjunto externo de Pareto é injetada dentro da população. Finalmente, uma busca local é efetuada para cada indivíduo com o intuito de melhorar as soluções correntes.

Greenwood, Hu e D'Ambrosio [28] propuseram uma combinação entre métodos de nenhuma informação preferencial (caso de ordenamento puro de Pareto) e métodos de agregação como soma dos objetivos ponderados por pesos. Eles estenderam o conceito de dominância de Pareto por elementos de valor com vários atributos imprecisos para incorporar preferência no processo de busca.

Um algoritmo genético multi-sexual para otimização multi-objetivos foi proposto por Lis e Eiben [45] que, em contraste com os algoritmos evolucionários convencionais, atribui a cada indivíduo um sexo próprio que é relacionado com um objetivo particular. O número de sexos corresponde ao número de objetivos. Enquanto a seleção é executada para cada sexo distinto (objetivo), o operador de recombinação gera herdeiros de pais que pertençam a todos os sexos diferentes. O conjunto final de Pareto é obtido pelo monitoramento das soluções não-dominadas durante a execução do algoritmo.

Partindo do “Niche Pareto Genetic Algorithms”, nPGA, Valenzuela-Rendón e Uresti-Charre [65] desenvolveram um algoritmo evolucionário não geracional que incorpora seleção baseada em dominância e compartilhamento de aptidão. A aptidão de cada indivíduo é composta de duas contribuições: o contador de dominância e o contador de nichos em movimento. O contador de dominância reflete o número médio de indivíduos pelo qual o indivíduo é dominado e o segundo parâmetro representa o número de indivíduos que se encontram próximos de uma certa solução. Em cada passo evolucionário ambos os valores são ajustados de acordo com as mudanças na população.

Cunha, Oliveira e Covas [18] abordaram o problema de conjuntos extremamente grandes de ótimos de Pareto. Eles apresentaram um algoritmo evolucionário que aplica agrupamentos para reduzir e ordenar as soluções do conjunto não-dominado.

Outras aproximações baseadas nos conceitos de alvos e desvios são também encontradas, contudo, elas utilizam na maioria das vezes fundamentos matemáticos, herdando suas vantagens e desvantagens frente aos algoritmos evolucionários. As técnicas mais populares desta corrente são: Programação Objetiva, Obtenção Objetiva e os algoritmos de min-max. A maior desvantagem destas estratégias é a forte possibilidade de prenderem-se a ótimos locais e a usual obtenção de uma solução única, já que operam iterativamente com um candidato por vez.

Capítulo VII

PMOGA – Pareto Multiobjective GA

Este capítulo apresenta todos os aspectos e características do algoritmo genético multi-objetivo de Pareto desenvolvido – PMOGA. Todos os procedimentos desenvolvidos e empregados são detalhados e discutidos.

1. Introdução	79
2. Resumo e Evoluções do Algoritmo Evolucionista.....	79
3. Algoritmo Final Detalhado	81
4. Filtro.....	82
5. Seleção	83
6. Reprodução	84
7. Operadores Multi-objetivos Desenvolvidos.....	86
8. Gerações Adicionais	88
9. Elitismo	89
10. Estratégia de Penalização	89
11. Resultados e Desempenho.....	90

1. Introdução

Neste capítulo será descrita a forma final do algoritmo proposto, contudo, com algumas ressalvas para as etapas e evoluções gradativas decorrentes de exaustivos experimentos aqui realizados: as evoluções do algoritmo evolucionista. Estes experimentos são apresentados no capítulo referente às aplicações, todavia, para cada uma delas são descritas as características e configurações presentes no algoritmo no momento de sua utilização.

2. Resumo e Evoluções do Algoritmo Evolucionista

O algoritmo genético para problemas de otimização multi-objetivos tem por finalidade evoluir um conjunto uniformemente distribuído de soluções pertencentes ao conjunto ótimo de Pareto. Suas principais características são: (i) ordenamento da população de acordo com propriedades de dominância, (ii) utilização de operadores multi-objetivos adequadamente desenvolvidos: exclusão, adição e operador de otimização individual, (iii) definição de um filtro para reter as soluções de Pareto, (iv) elitismo deste filtro dentro das populações e (v) método de penalização por camadas.

As principais evoluções ou melhorias que levaram o algoritmo inicialmente desenvolvido (utilizado e descrito na primeira aplicação – função f_2 de Schaffer) até o seu estado atual podem ser cronologicamente apontadas como:

- Modificação do método de seleção
- Modificações do operador de recombinação
- Desenvolvimento e introdução dos operadores multi-objetivos
- Introdução do elitismo do filtro nas populações

2.1. Modificação do método de seleção

O método de seleção inicialmente implementado foi o da roleta, originalmente proposto por Holland [36] e descrito nos capítulos antecedentes. Este método de seleção provoca inúmeros problemas já conhecidos – *estagnação e convergência prematura* – que são altamente prejudiciais ao processo de otimização.

A segunda implementação foi baseada no torneio entre soluções com base nas aptidões calculadas para cada ordem como apresentado em [12, 13], contudo, novos estudos foram realizados e modificou-se o método utilizado para que o mesmo fosse baseado diretamente no ordenamento de cada solução, o que dispensou inúmeros cálculos de aptidões de ordens e tornou o procedimento mais intuitivo e eficiente. Maiores detalhes serão descritos a seguir no item referente ao método de seleção.

2.2. Modificações do operador de recombinação

O operador de recombinação percorreu um longo caminho, desde os tradicionais operadores de um ponto e dois pontos até o operador de recombinação uniforme, onde se preferiu adotar duas versões deste operador com diferentes probabilidades de máscara como será explicado adiante.

2.3. Desenvolvimento e introdução dos operadores multi-objetivos

Com os procedimentos anteriormente apresentados o algoritmo já cumpria sua função de otimizador, todavia, as soluções não eram aquelas esperadas. Por isso, foram desenvolvidos novos operadores, chamados de multi-objetivos por terem sido desenvolvidos especialmente para este contexto. Estes operadores foram nomeados de: operador de exclusão, operador de adição e operador de otimização individual. Eles foram desenvolvidos cronologicamente nesta ordem e serão abordados adiante.

2.4. Introdução do elitismo do filtro nas populações

Posteriormente foi implementado o elitismo do filtro para as populações, onde conseguiu-se melhorias na solução final armazenada com uma convergência mais rápida.

3. Algoritmo Final Detalhado

Algoritmo PMOGA

```
Geração da população inicial aleatória
Avaliação da população inicial
Ordenamento da população inicial
Cópia dos indivíduos de ordem 1 da população para o filtro
Operador de exclusão no filtro até reduzi-lo ao tamanho desejado
Inserção das soluções do filtro na próxima geração (como filhos)
Repita da primeira geração
    Se a geração corrente  $\leq$  ao número de gerações padrões então
        Repita
            Seleção (torneio pela menor ordem)
            Recombinação (uniforme 0.5 e 0.2)
            Mutação
            Avaliação
        Até completar o tamanho da população
        Atualização da geração (filho  $\rightarrow$  pai)
        Ordenamento da população
        Cópia dos indivíduos de ordem 1 da população para o filtro
    Fim se
    Operador de otimização individual no filtro (recombinação e mutação)
    Operador de adição no filtro
    Ordenamento do filtro
    Eliminação das soluções dominadas do filtro
    Operador de exclusão no filtro até reduzi-lo ao tamanho desejado
    Inserção das soluções do filtro na próxima geração (como filhos)
Até o número de gerações totais (padrões+adicionais)
Fim
```

Figura 20 - Pseudocódigo do PMOGA

4. Filtro

O algoritmo desenvolvido possui um filtro para armazenar as soluções não-dominadas que vão surgindo ao longo do processo evolutivo. Este procedimento é fundamental na resolução de problemas multi-objetivos, uma vez que a solução destes problemas é, na verdade, um conjunto de soluções – fronteira ótima de Pareto.

Ao final de cada geração as soluções não-dominadas da população corrente são adicionadas ao filtro. Logo, um novo teste de dominância – ordenamento – é efetuado para eliminar deste novo conjunto – formado pelas soluções adicionadas e pelas anteriormente presentes – as soluções que se tornaram dominadas. Depois disso são empregados os operadores multi-objetivos de otimização individual e de adição que serão descritos adiante, buscando a criação de soluções “interessantes” para o problema, bem como a aplicação do operador de exclusão para eliminar soluções do filtro até que ele alcance o número de soluções pré-estabelecido na configuração do algoritmo.

A última etapa realizada com o filtro é a inserção dele na população da próxima geração – elitismo do filtro na população – para aumentar a pressão de seleção e também a convergência das soluções para a fronteira de Pareto.

Os procedimentos que são empregados no filtro a cada nova geração podem ser resumidos da seguinte maneira:

- Adição das soluções não-dominadas da população corrente.
- Aplicação dos operadores multi-objetivos de otimização individual e adição.
- Ordenamento do novo filtro e eliminação das soluções agora dominadas.
- Aplicação do operador multi-objetivo de exclusão.
- Inserção do filtro na população da próxima geração.

5. Seleção

A seleção é um dos procedimentos mais importantes de qualquer algoritmo genético, uma vez que ele é o foco central da técnica dos algoritmos genéticos para incorporar ao algoritmo os conceitos da Seleção Natural de Darwin. Ou seja, deve-se assegurar que os melhores indivíduos tenham uma maior probabilidade de seleção para assim se obter um aumento progressivo da aptidão média das populações durante o processo evolutivo. Convergindo estes indivíduos para a resposta desejada do problema.

O procedimento de seleção adotado é o de torneio, onde algumas soluções são aleatoriamente tomadas da população e, com base em algum critério, a solução vencedora é então selecionada. Normalmente, o critério utilizado pela maioria dos algoritmos genéticos com um objetivo é o valor da função de aptidão, já para algoritmos genéticos multi-objetivos alguma estratégia de nicho é empregada de forma a modificar as aptidões reais dos indivíduos conforme a densidade de vizinhos no seu entorno.

O número de soluções tomadas para o torneio é proporcional à pressão de seleção aplicada na população e normalmente varia de duas a dez soluções.

Como o algoritmo proposto emprega operadores multi-objetivos ao invés de estratégias de nicho para melhorar a distribuição da fronteira de Pareto, nenhuma modificação é realizada nas aptidões reais dos indivíduos. Esta característica habilita este algoritmo a resolver alguns problemas que os demais algoritmos baseados em estratégias de nicho não resolveriam de maneira satisfatória, como vem acontecendo na função-teste τ_6 - formulada no conjunto de testes para algoritmos multi-objetivos da referência [71].

O procedimento de torneio empregado é efetuado diretamente sobre as ordens recebidas pelos indivíduos, deste modo, os indivíduos são selecionados não só pelas suas aptidões, mas sim pelas suas aptidões dentro do contexto multi-objetivo de dominância. Ou seja, dispensa cálculos adicionais de aptidões novas ou modificadas.

5.1. Ordenamento

O procedimento de ordenamento das soluções consiste em realizar um teste de dominância sobre todo o conjunto ao qual se quer ordenar. Assim, as soluções não-dominadas deste conjunto recebem a primeira ordem e são retiradas da população, logo, um novo teste é efetuado sobre o conjunto remanescente e as novas soluções não-dominadas adquirem a segunda ordem. Deste modo, este procedimento continua até que todas as soluções do conjunto especificado tenham sido ordenadas.

No final do ordenamento têm-se várias ordens que correspondem às várias fronteiras ou camadas de dominância existentes no conjunto ordenado. Quanto menor for a ordem de uma solução significa que ela está mais próxima da fronteira ótima.

6. Reprodução

A reprodução é uma etapa inspirada na natureza e tem por objetivo criar novas soluções na população. São utilizados operadores advindos da reprodução humana, tais como os operadores de recombinação e mutação. Entretanto, nenhuma barreira deve ser construída quanto ao desenvolvimento de novos operadores ou quanto ao emprego diferenciado dos mesmos, desde que tais mudanças sejam benéficas para a resolução dos problemas tratados.

O operador de mutação não tem sofrido grandes mudanças dentro da técnica em relação ao que ocorre na natureza, a não ser quanto sua probabilidade de aplicação, que tem variado bastante de algoritmo para algoritmo e de problema para problema.

O operador de recombinação tem sido bastante modificado ao longo do tempo e existem inúmeras possibilidades – um ponto, dois pontos, uniforme com várias probabilidades de máscaras e muitos outros.

O operador de recombinação uniforme apresenta a vantagem de ter sua eficiência independente do tamanho do cromossomo, isto é, se um cromossomo tem dez posições de comprimento, um operador de dois pontos provocará uma recombinação considerável para gerar as novas soluções, já se o tamanho do cromossomo for de novecentas posições, esta recombinação passará a ser pequena para este tamanho de codificação, ou seja, seria necessário um maior número de pontos para se ter proporcionalmente um mesmo desempenho. Entretanto, o operador de recombinação uniforme dispõe de um mesmo desempenho de recombinação independente deste tamanho, uma vez que ele não depende do número de pontos de corte para troca de material genético e sim da probabilidade de cada alelo do gene ser herdado do pai ou da mãe. Assim sendo, seu emprego torna o algoritmo menos dependente do tamanho do problema a ser resolvido [62].

Neste algoritmo finalizou-se com o operador de recombinação uniforme com diferentes probabilidades de máscara – 0.2 e 0.5 como será descrito e justificado adiante.

6.1. Mutação

O operador de mutação empregado não foge às regras do que é observado na natureza, ou seja, para cada gene re combinado existe uma probabilidade de mutação que uma vez ativada modifica o gene herdado, isto é, de zero para um ou vice-versa, conforme apresentado nos capítulos anteriores.

Devido à existência de mais de um objetivo, o espaço de busca é mais complexo nos problemas multi-objetivos, portanto, uma taxa de mutação mais alta pode ser usada para procurar por soluções mais dispersas, bem como melhor explorar o espaço de busca.

6.2. Recombinação

Foi escolhido o operador de recombinação uniforme que já foi apresentado nos capítulos anteriores, podendo o mesmo apresentar diferentes probabilidades de máscara. Ou seja, se a probabilidade de máscara é de 0.5, isto quer dizer que o gene que será herdado tem exatamente a mesma probabilidade de vir do pai ou da mãe. Por outro lado, se esta probabilidade é diferenciada para 0.2, por exemplo, então o gene herdado terá 20% de chances de vir de um dos pais e os restantes 80% de vir do outro genitor. Com isso o herdeiro será forçado a ficar “mais parecido” com um dos seus pais. Esta característica é interessante para garantir modificações mais suaves ou mais agressivas nos herdeiros conforme sejam os seus interesses.

No algoritmo final foram utilizados dois operadores de recombinação uniforme, um com probabilidade de máscara de 0.5 e outro com 0.2, ambos com a mesma probabilidade de utilização, 0.5.

7. Operadores Multi-objetivos Desenvolvidos

Os operadores multi-objetivos foram desenvolvidos para melhorar a distribuição das soluções pela fronteira de Pareto, podendo desta forma substituir o emprego de uma estratégia de nicho que tem a mesma finalidade. Dentre as vantagens estão um menor esforço computacional e a não modificação das aptidões reais dos indivíduos, que em certos problemas pode ocasionar dificuldades de resolução.

7.1. Operador de Exclusão

O operador de exclusão determina a solução mais próxima de outras soluções no filtro atual para ser removida, com o intuito de propiciar uma melhor distribuição das soluções ao longo do conjunto de Pareto. Este procedimento de

exclusão é repetido até que o filtro corrente alcance o tamanho especificado nas configurações do algoritmo.

A métrica utilizada para avaliar as distâncias entre os cromossomos é dada pela expressão seguinte, que resulta em um processo de soma das porcentagens das distâncias de cada objetivo entre as duas soluções j e k pesquisadas:

$$d_{j,k} = \sum_{i=1}^{N_{fo}} \frac{100 \cdot |fo_i(j) - fo_i(k)|}{0,5 \cdot [fo_i(j) + fo_i(k)]}$$

Esta métrica tem se comportado muito bem para uma vasta gama de aplicações, já que a distância entre soluções independe da magnitude das aptidões e sim da distância relativa entre elas.

7.2. Operador de Adição

O operador de adição acha as duas soluções mais distantes (no espaço de cada função objetivo) e as recombina n vezes, onde o valor de n normalmente varia de 1 a 5. A idéia do operador é gerar soluções que ocupem o espaço vazio encontrado entre tais soluções, melhorando deste modo a distribuição da fronteira ótima.

O número de avaliações adicionais por geração será $2 \times n \times n_{fo}$ (vezes 2 porque cada par de genitores gera duas novas soluções pela recombinação uniforme adotada).

Em problemas onde exista descontinuidade da fronteira de Pareto, recomenda-se o desligamento deste operador, uma vez que as soluções mais distantes provavelmente pertencerão à duas fronteiras não interligadas e a recombinação destas soluções tende a gerar indivíduos intermediários aos mesmos

que estariam fora das regiões da fronteira de Pareto. Contudo, como foi observado em um dos problemas analisados adiante, cuja fronteira de Pareto apresenta esta descontinuidade, observou-se que o emprego deste operador, embora não recomendado, não compromete o funcionamento do algoritmo.

7.3. Operador de Otimização Individual

Foram criadas duas formas deste operador:

- i) São determinadas as duas soluções mínimas de cada objetivo para serem re combinadas n vezes.
- ii) A solução mínima de cada objetivo é copiada n vezes com mutação.

Estas novas soluções criadas são inseridas no próprio filtro corrente. O objetivo deste operador é melhorar o ótimo de cada objetivo individualmente, permitindo que o algoritmo seja utilizado, em caráter emergencial, para otimizar problemas com um único objetivo. O número de avaliações é $2 \times n \times n_{fo}$ e $n \times n_{fo}$ respectivamente.

Nenhum outro algoritmo apresenta alguma estratégia diretamente desenvolvida para este fim, que trás enorme benefício para a solução final, pois abre as extremidades da fronteira ótima, mostrando ao tomador de decisões quais seriam os valores ótimos individuais de cada objetivo.

8. Gerações Adicionais

Os três operadores multi-objetivos anteriores trabalham ao término de cada geração, contudo, é possível e recomendado que: depois de um certo número de gerações processadas e que um número significativo de soluções ótimas estejam

retidas no filtro, sejam processadas gerações adicionais utilizando apenas estes três operadores. Estas gerações adicionais são mais rápidas e melhoram significativamente a qualidade do conjunto final de Pareto.

A idéia fundamental destas gerações adicionais é apenas melhorar a distribuição da fronteira de Pareto, pela aplicação sucessiva dos operadores de otimização individual, adição e exclusão diretamente sobre as soluções até então armazenadas no filtro.

9. Elitismo

Nos algoritmos genéticos desenvolvidos para otimização de um objetivo é comum o emprego de elitismo da melhor solução, ou seja, a melhor solução de cada geração é copiada para a geração seguinte para que a mesma não seja possivelmente perdida. Com o tempo vários autores passaram a adotar o elitismo de mais de uma solução, por exemplo, as cinco ou dez melhores.

No contexto multi-objetivo onde a resposta é composta por um conjunto de soluções este elitismo deve ser estendido para o conjunto ótimo até então armazenado, ou seja, o filtro de soluções não-dominadas. Assim, ao final de cada geração as soluções armazenadas no filtro são inseridas na próxima população com o intuito de aumentar a pressão de seleção e ao mesmo tempo acelerar a convergência do algoritmo.

10. Estratégia de Penalização

Métodos de penalização sempre são um entrave para a resolução de problemas de otimização, pois normalmente exigem a escolha de parâmetros de penalização que necessitam de uma definição iterativa e não trivial, variando ainda de problema para problema.

Depois dos inúmeros problemas resolvidos convergiu-se para uma estratégia de penalização bastante simples e de fácil configuração, pois se recomenda a atribuição de valores limites de cada objetivo para tais parâmetros, deste modo, garante-se que qualquer solução não-dominada domine qualquer solução que viole pelo menos uma restrição, do mesmo modo qualquer solução que viole uma restrição dominará qualquer solução que apresente duas violações de restrições e assim por diante. Com isso obtêm-se camadas de soluções, e conseqüentemente ordens, com um mesmo número de violações.

A penalização é aplicada pela seguinte expressão:

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) + c_i \cdot nviol$$

onde *nviol* é o número de restrições violadas e os valores das constantes c_i são escolhidos para cada problema em particular com limites extremos para cada objetivo. Isto é, não são necessários muitos estudos e testes para a determinação destas constantes, pois basta escolher-se valores altos que sejam superiores aos limites máximos possíveis para cada objetivo.

11. Resultados e Desempenho

Os resultados encontrados com o algoritmo proposto têm sido, em geral, equivalentes ou superiores aos apresentados na literatura com um bom desempenho computacional. Comprovando a boa integração entre as várias estratégias escolhidas e desenvolvidas para formarem o corpo do algoritmo multi-objetivo proposto.

O algoritmo tem se mostrado capaz inclusive de resolver problemas onde outros inúmeros algoritmos falharam, como pode ser observado na aplicação referente ao conjunto de funções-teste desenvolvidas por Deb [71] para otimizadores multi-objetivos.

Capítulo VIII

Aplicações

Neste capítulo vários problemas testes e/ou estruturais são analisados com o algoritmo multi-objetivo de Pareto proposto – PMOGA. Todas as configurações utilizadas pelo algoritmo em cada problema são detalhadas.

1. Introdução	92
2. Função f_2 de Schaffer.....	92
3. Viga I	97
4. Viga Soldada	103
5. Treliza com Três Objetivos.....	108
6. Ponte Trelizada.....	113
7. Fronteira de Pareto Descontínua	119
8. Funções-teste para Otimizadores Multi-objetivos.....	124
9. Curva Tri-objetiva	153
10. Esfera	158
11. Cubo – 8 Funções Objetivo.....	165
12. Treliza de 47 barras.....	176

1. Introdução

As aplicações são apresentadas segundo a ordem cronológica em que foram abordadas. Com o aumento gradativo das dificuldades dos problemas o algoritmo genético utilizado foi também sendo aperfeiçoado, deste modo, para cada aplicação existe um tópico que descreve as características e configurações empregadas naquele problema em particular.

O amadurecimento e a robustez do algoritmo desenvolvido são frutos de exaustivos testes e progressivas complicações a ele impostas, que foram obtidos após inúmeros insucessos e conseqüentes tentativas de novas estratégias e experiências.

2. Função f_2 de Schaffer

2.1. Introdução e Modelo

Um problema bastante simples e largamente utilizado para testar otimizadores multi-objetivos é o problema f_2 formulado por Schaffer [69]. O problema consiste na minimização simultânea de duas funções objetivo e embora seja um problema fácil para um algoritmo evolucionário, ele é comumente resolvido pela maioria dos autores como um ponto inicial de referência, principalmente para a investigação da distribuição da população ao longo da fronteira ótima de Pareto.

O problema é definido pelas seguintes funções:

- $f_{o1}(x) = x^2$ (linha contínua)
- $f_{o2}(x) = (x - 2)^2$ (linha tracejada)

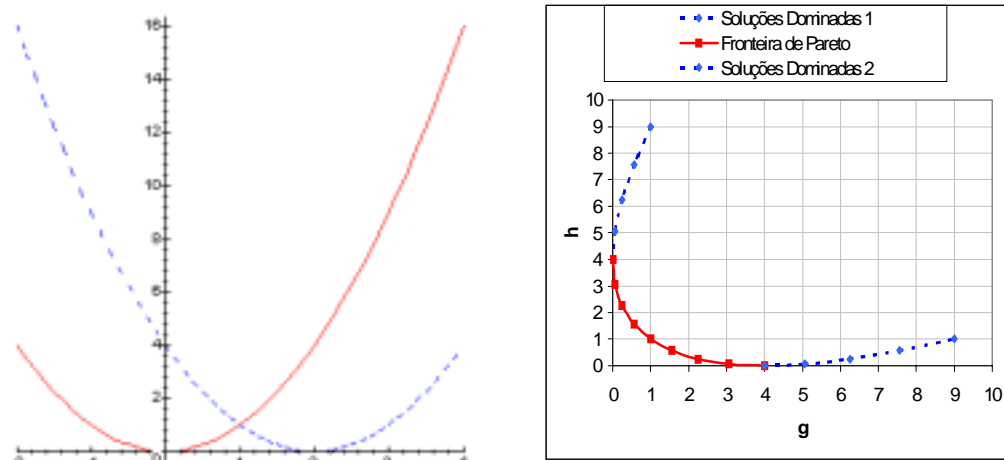


Figura 21 - Funções objetivo e soluções no espaço das funções objetivo

Observa-se pelas figuras que as soluções não-dominadas que compõem a fronteira ótima de Pareto situam-se no intervalo $[0, 2]$, deste modo, as soluções compreendidas neste intervalo não podem ser ditas melhores ou piores entre si, devendo a tarefa de escolha da solução ótima (projeto ótimo) ser definida posteriormente por um tomador de decisão (projetista).

2.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Ordenamento das soluções segundo critérios de não dominância.
- Mesma aptidão para cada ordem, dependendo da ordem e de seu tamanho.
- Método de seleção da roleta sobre as aptidões das ordens.
- Operador de recombinação em 2 pontos e mutação.
- Operador de exclusão.
- População inicial aleatória.
- Probabilidade de Recombinação = 85 %.
- Probabilidade de Mutação = 5%.
- Tamanho da população = 70.
- Tamanho do filtro = 30.
- Número de gerações = 100.

2.2.1. Atribuição de aptidão para as ordens

A atribuição de aptidão para as ordens utiliza o método proposto na referência [12] pelas seguintes expressões:

$$F_i = (M_r - i + 1) / SS$$

$$SS = \left[\sum_{i=1}^{M_r} (M_r - i + 1) T_{ri} \right] / T_{pop}$$

onde:

- T_{pop} é o tamanho da população;
- M_r é a mais alta ordem da população (ordem máxima);
- i varia de acordo com cada ordem;
- T_{ri} é o tamanho da população da ordem i ;
- F_i é a aptidão de cada cromossomo da ordem i .

Pode-se observar que quanto mais populosa for uma ordem, menor será sua aptidão proporcional, embora as ordens tenham suas aptidões inversamente proporcionais aos seus ordenamentos, isto é, a primeira ordem tem aptidão superior ao do segundo, que por sua vez tem maior aptidão que o terceiro e assim por diante.

Utilizou-se uma codificação binária em cromossomos de quatorze unidades de comprimento (bits) que são decodificados para um número real no intervalo $[-6, 6]$. Assim, o cromossomo foi interpretado como um valor inteiro i , sendo posteriormente convertido para um valor real x pela seguinte equação:

$$x = -6 + i \cdot \frac{|-6 - 6|}{2^{14} - 1}$$

Consegue-se por este modo a seguinte representação:

Cromossomo	Valor Inteiro (i)	Valor Real (x)
00000000000000	0	-6.00
...
11111111111111	$2^{14}-1$	6.00

2.3. Resultados

Nas figuras seguintes são mostradas a população inicial e os resultados obtidos, isto é, a fronteira ótima de Pareto e a distribuição das soluções no espaço das variáveis de projeto:

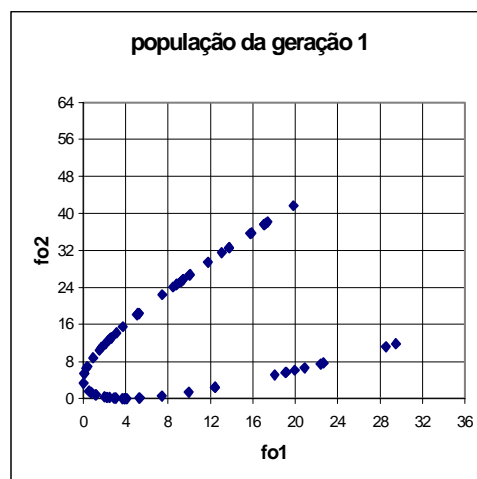


Figura 22 - População inicial

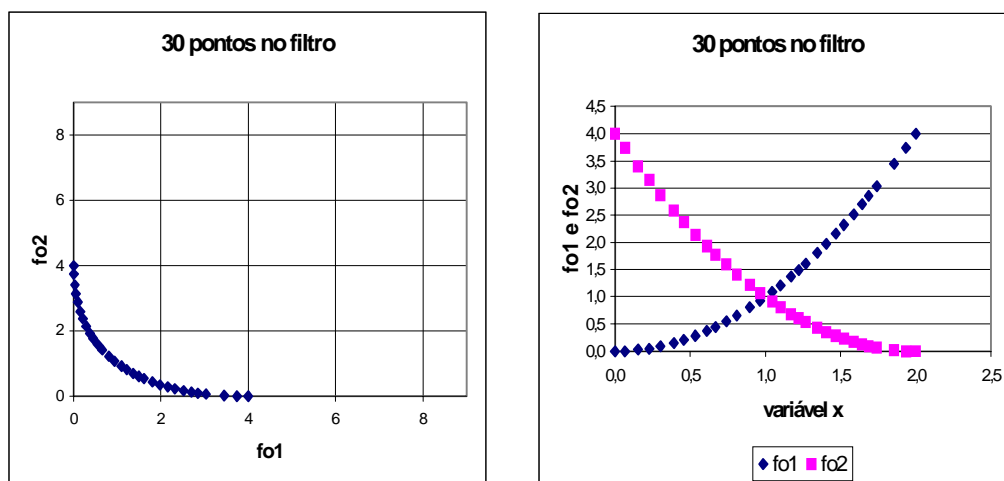


Figura 23 - Fronteira de Pareto e soluções no espaço das variáveis de projeto.

Uma nova alternativa para o método de seleção foi aqui testada. Este método proposto não faz uso de qualquer cálculo de aptidão para as várias ordens encontradas durante a evolução das gerações, mas sim, a utilização do método convencional de torneio entre indivíduos, comparando simplesmente a ordem entre eles, ou seja, o cromossomo que apresentar a menor ordem entre os

sorteados é selecionado. Caso as soluções sorteadas tenham uma mesma ordem, seleciona-se aleatoriamente uma delas.

A utilização deste método de seleção mostrou-se bastante interessante não só pela diminuição do esforço computacional, já que inúmeros cálculos são descartados, mas também pelos resultados que foram melhorados, conforme figura seguinte:

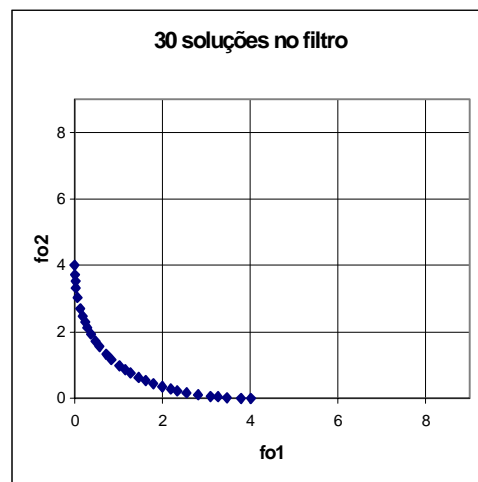


Figura 24 - Fronteira de Pareto com novo método de seleção (torneio)

2.4. Conclusões

Os resultados foram equivalentes ou superiores aos encontrados na literatura [57, 69], mesmo sem o emprego de estratégias de nicho que aumentam o número de parâmetros de configuração e tornam a resolução do problema sensível a estas configurações.

A alternativa pelo método de seleção por torneio atuando diretamente sobre as ordens mostrou-se uma boa alternativa, já que diminuiu o esforço computacional e eliminou as deficiências oriundas do método de seleção da roleta.

3. Viga I

3.1. Introdução e Modelo

O objetivo do problema é determinar o conjunto ótimo de Pareto decorrente da otimização multi-objetivos da viga em seção I apresentada abaixo [15].

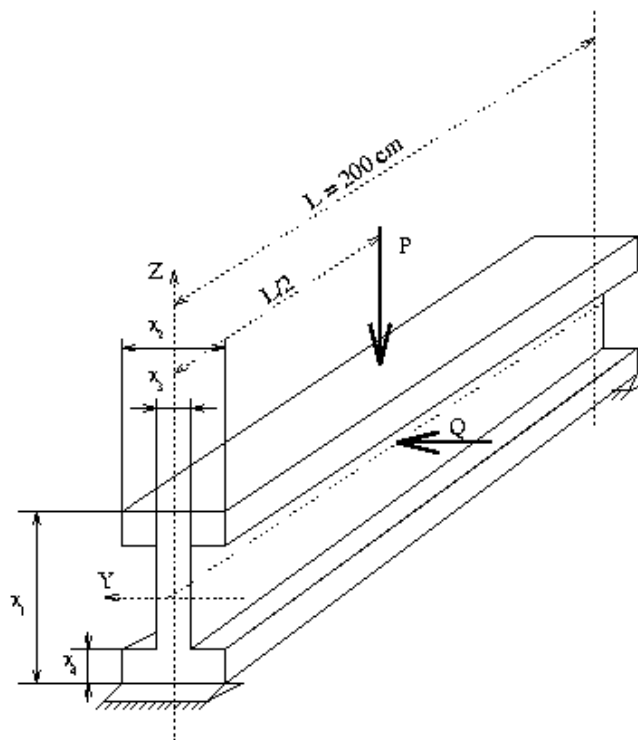


Figura 25 - Viga em seção I [15].

As propriedades da viga e os valores das cargas são:

- $E = 2 \times 10^4 \text{ kN/cm}^2$ (módulo de Young).
- $k_g = 16 \text{ kN/cm}^2$ (tensão de projeto da viga).
- $P = 600 \text{ kN}$ (carga vertical em $L/2$).
- $Q = 50 \text{ kN}$ (carga horizontal em $L/2$).

O vetor das variáveis de projeto são as dimensões da viga dadas em centímetros, é $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$, conforme mostrado na figura anterior.

Os objetivos a serem minimizados são:

- Área da seção transversal.
- Deslocamento estático máximo (em $L/2$) decorrente de P.

Como na maioria dos problemas multi-objetivo os critérios são conflitantes entre si, neste caso, a diminuição da área da seção transversal provoca o aumento do deslocamento máximo e vice e versa.

As restrições que devem ser atendidas pela solução do problema são:

1. Restrições Geométricas (valores em centímetros):

$$10 \leq x_1 \leq 80$$

$$10 \leq x_2 \leq 50$$

$$0,9 \leq x_3 \leq 5$$

$$0,9 \leq x_4 \leq 5$$

2. Restrição de projeto:

$$\frac{M_y}{W_y} + \frac{M_z}{W_z} \leq k_g$$

sendo:

- M_y e M_z os momentos máximos nas direções y e z.
- W_y e W_z os módulos resistentes da seção nas direções y e z.
- k_g a tensão limite estabelecida.

Para os valores das cargas atuantes P e Q tem-se:

- $M_y = 30000 \text{ kN.cm}$
- $M_z = 2500 \text{ kN.cm}$

Os módulos resistentes são calculados pelas seguintes expressões:

$$W_y = \frac{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]}{6x_1}$$

$$W_z = \frac{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3}{6x_2}$$

Substituindo estes valores na restrição de projeto obtêm-se:

$$16 - \frac{180000x_1}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} - \frac{15000x_2}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3} \geq 0$$

As funções objetivo podem ser expressas como se segue:

1. Área da seção transversal:

$$f_1(\mathbf{x}) = 2x_2x_4 + x_3(x_1 - 2x_4) \text{ cm}^2$$

2. Deslocamento estático máximo:

$$f_2(x) = \frac{PL^3}{48EI} \text{ cm}$$

onde I é o momento de inércia que pode ser calculado pela fórmula:

$$I = \frac{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]}{12}$$

resultando na seguinte expressão para o segundo objetivo:

$$f_2(x) = \frac{60000}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} \text{ cm}$$

3.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operadores convencionais de mutação e recombinação uniforme.
- Operador de exclusão.
- População inicial aleatória.
- Aplicação da penalização: $f_i(\mathbf{x}) = f_i(\mathbf{x}) + c_i$, onde $c_1=1000$ e $c_2=10$.
- Probabilidade de Recombinação = 85 %.
- Probabilidade de Mutação = 5%.
- Tamanho da população = 200.
- Tamanho do filtro = 20.
- Número de gerações = 200.

As variáveis de projeto foram codificadas numa cadeia binária com quarenta e três posições de comprimento, sendo treze posições destinadas para x_1 , doze para x_2 e nove tanto para x_3 como para x_4 . Cada cadeia binária é desmembrada e convertida para os valores reais de cada variável pela expressão:

$$x_i = x_{\min.} + B_i \frac{|x_{\min.} - x_{\max.}|}{2^{T_i} - 1}$$

sendo:

- i – índice da variável considerada
- $x_{\min.}$ e $x_{\max.}$ os valores limites dados pelas restrições geométricas
- B_i é o valor inteiro correspondente à cadeia binária de cada variável
- T_i é o número de posições destinadas à cada variável

3.3. Resultados

Nas duas figuras seguintes são mostradas respectivamente a população inicial e a população relativa à geração 200, verificando-se a nítida separação entre o conjunto das soluções factíveis e não-factíveis.

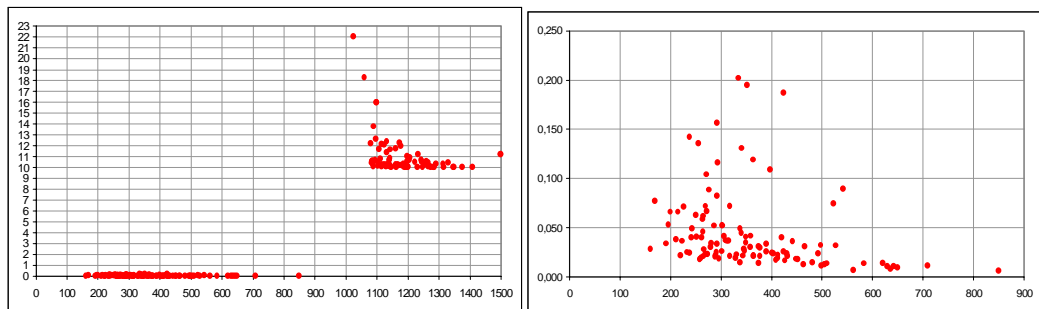


Figura 26 - População inicial em diferentes escalas

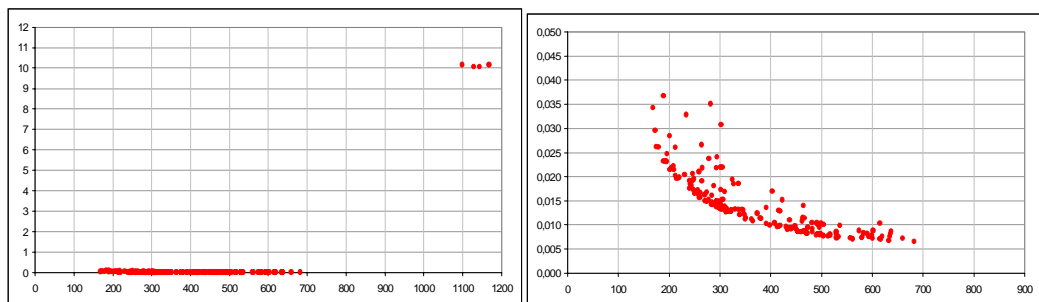


Figura 27 - População relativa à geração 200 em diferentes escalas

Observa-se com o decorrer do processo evolutivo a diminuição do número de soluções inviáveis (penalizadas) e a convergência para a região da fronteira ótima de Pareto. Na próxima figura é apresentada a solução final do filtro.

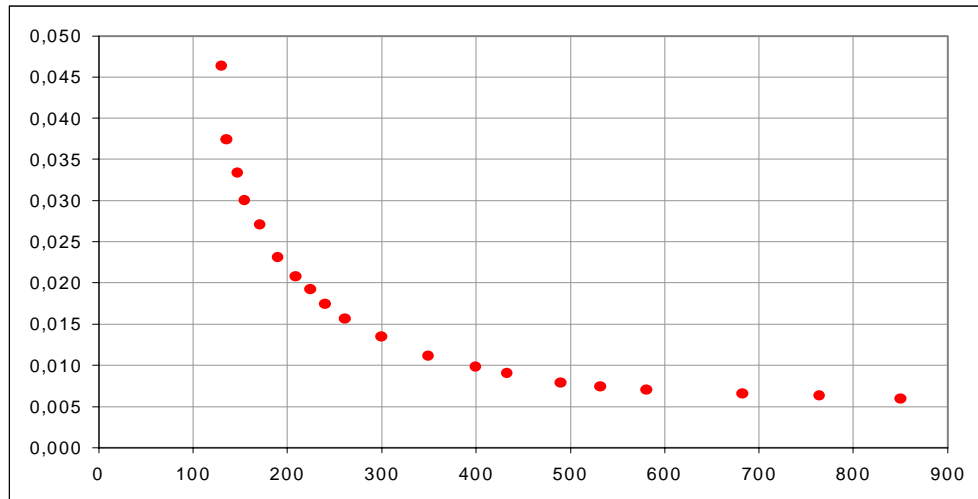


Figura 28 - Soluções no filtro final (20 soluções)

Durante o tratamento deste problema foram desenvolvidos os operadores de adição e de otimização individual que apresentam baixo custo computacional e devem ser aplicados diretamente no filtro. Na figura e tabela seguintes, oriundas de uma nova otimização pode-se perceber as vantagens conseguidas com a sua utilização: melhoramento dos extremos individuais e da distribuição da fronteira ótima.

Modificações realizadas:

- Introdução dos operadores de adição e de otimização individual.
- Tamanho da população = 50
- Número de gerações = 50 + 450 (filtro)

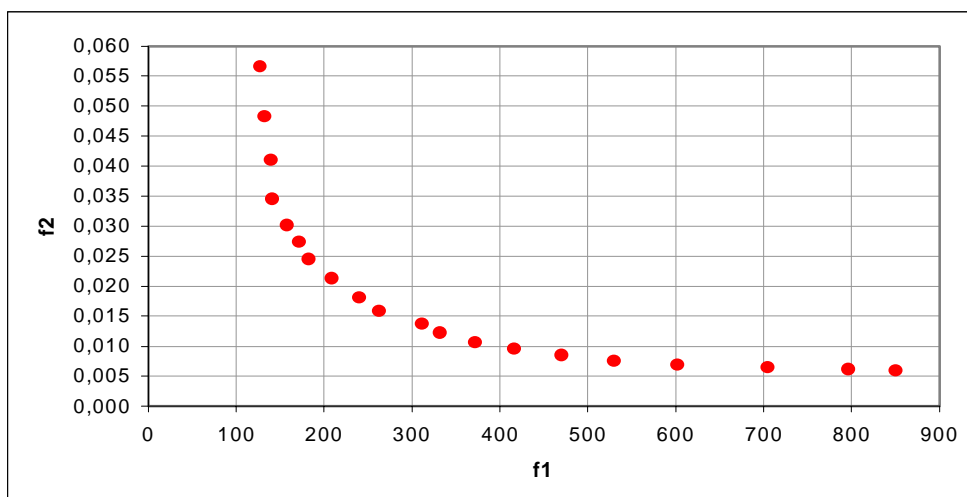


Figura 29 - Soluções no filtro final (ger. 500).

x1	x2	x3	x4	fo1	fo2
63,60	40,01	0,90	0,90	127,60	0,0565
68,05	40,34	0,90	0,90	132,20	0,0482
72,35	41,26	0,91	0,91	139,00	0,0409
79,12	40,05	0,90	0,90	141,70	0,0345
78,32	45,70	0,92	0,96	158,10	0,0301
79,65	23,03	0,96	2,15	171,80	0,0273
78,91	40,09	0,92	1,41	183,00	0,0245
78,89	43,89	1,00	1,51	208,70	0,0213
78,56	34,83	0,96	2,42	240,00	0,0181
79,66	46,58	1,02	1,99	262,70	0,0158
77,95	42,21	1,04	2,80	311,50	0,0136
78,62	45,11	0,93	2,92	331,40	0,0123
79,79	40,89	0,93	3,72	372,00	0,0106
79,92	40,04	1,01	4,30	416,70	0,0095
79,99	47,49	1,29	3,97	470,50	0,0085
79,58	50,00	1,15	4,49	529,80	0,0075
80,00	48,67	1,66	4,99	602,30	0,0068
79,59	50,00	2,95	5,00	705,00	0,0064
79,72	50,00	4,42	4,87	796,60	0,0062
80,00	50,00	5,00	5,00	850,00	0,0059

Tabela 1 - Valores das variáveis de projeto e das funções objetivo.

3.4. Conclusões

Os operadores de adição e de otimização individual foram desenvolvidos para serem aplicados diretamente sobre o filtro com o objetivo de preencher os espaços vazios encontrados entre as soluções mais espaçadas, bem como pesquisar por melhores ótimos individuais. Os resultados foram satisfatórios e melhores que os anteriores.

4. Viga Soldada

4.1. Introdução e Modelo

O problema consiste de uma viga submetida a uma força F em sua extremidade que precisa ser soldada a um outro componente estrutural satisfazendo às condições de estabilidade e as limitações de projeto. As quatro variáveis de projeto – espessura da solda (h); comprimento da solda (l); largura da viga (t) e espessura da viga (b) – são indicadas na figura seguinte.

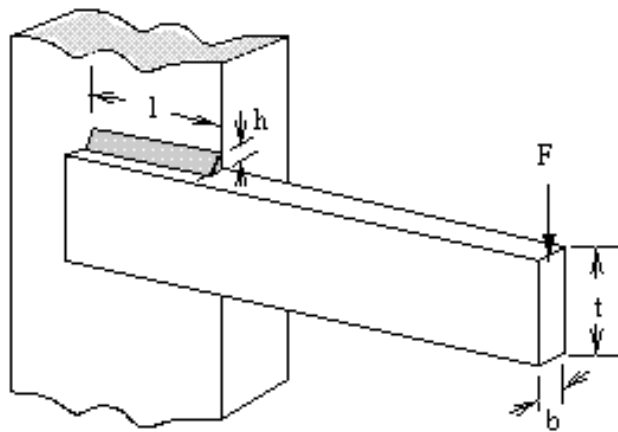


Figura 30 - Viga soldada [21].

Os dois objetivos do problema são:

- o custo da viga;
- o deslocamento na extremidade livre da viga.

Ambos os objetivos devem ser minimizados, embora eles sejam incomensuráveis e contraditórios, isto é, a redução do custo conduz a um maior deslocamento, enquanto que para diminuir o deslocamento é necessário aumentar o custo.

Existem cinco restrições intrínsecas do problema. As duas primeiras restrições garantem que a tensão de cisalhamento e a tensão normal desenvolvida ao longo do apoio da viga sejam respectivamente menores que as tensões permissíveis de cisalhamento (τ_{max}) e normal (σ_{max}) do material. A terceira restrição garante que o esforço resistente (ao longo da direção de t) da viga seja maior que a carga aplicada F . A quarta restrição é um limite máximo (u_{max}) para o deslocamento na extremidade da viga. A quinta restrição garante que a espessura da viga não seja menor que a espessura da solda. Existem ainda outras restrições geométricas nas variáveis de decisão de acordo com o modelo seguinte:

$$\text{Min. } f_1(\mathbf{x}) = 1.10471h^2l + 0.04811tb(14+l)$$

$$\text{Min. } f_2(\mathbf{x}) = 2.1952 / (t^3b)$$

Sujeitas a

$$g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{max} \leq 0$$

$$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{max} \leq 0$$

$$g_3(\mathbf{x}) = F - P_c(\mathbf{x}) \leq 0$$

$$g_4(\mathbf{x}) = 2.1952 / (t^3b) - u_{max} \leq 0$$

$$g_5(\mathbf{x}) = h - b \leq 0$$

$$0.125 \leq h, b \leq 5.0$$

$$0.1 \leq l, t \leq 10.0$$

As tensões e os termos das expressões anteriores são dados por [21]:

$$\tau(x) = \sqrt{\tau'^2 + \tau''^2 + l\tau'\tau'' / \sqrt{0.25[l^2 + (h+t)^2]}}$$

$$\tau' = \frac{6000}{\sqrt{2hl}}$$

$$\tau'' = \frac{6000(14 + 0.5l)\sqrt{0.25[l^2 + (h+t)^2]}}{2\{0.707hl[l^2 / 12 + 0.25(h+t)^2]\}}$$

$$\sigma(x) = \frac{504000}{t^2b}$$

$$P_c(x) = 64746.022(1 - 0.0282346t)tb^3$$

Os dados adotados para o problema são [17]:

$$\begin{array}{ll} F = 6000 \text{ lb} & \tau_{\max} = 13600 \text{ psi} \\ E = 30 \times 10^6 \text{ psi} & \sigma_{\max} = 30000 \text{ psi} \\ G = 12 \times 10^6 \text{ psi} & u_{\max} = 0.25 \text{ in} \\ L = 14 \text{ in (comp. livre)} & \end{array}$$

4.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operadores convencionais de mutação e recombinação uniforme.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).
- Aplicação de uma penalização conforme apresentado adiante.
- Probabilidade de recombinação = 85 %.
- Probabilidade de mutação = 5%.
- Tamanho da população = 200.
- Tamanho do filtro = 20.
- Número de gerações = 100.
- Número de gerações adicionais (filtro) = 0 e 400.

O cromossomo tem cinquenta e quatro bits de comprimento e um método simples de penalidade é proposto e utilizado:

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) + c_i \cdot n_{viol}$$

onde $c_1=100$; $c_2=0,01$ e n_{viol} é o número de restrições violadas. Os valores das constantes c_i garantem que uma solução que viole uma restrição seja pior que uma solução viável, da mesma forma que, uma solução que viole duas restrições seja pior que uma outra que viole uma única restrição e assim por diante.

4.3. Resultados

Na figura seguinte, tanto as soluções viáveis da geração inicial como as da geração 100 são mostradas. A eficiência dos novos operadores é comprovada pela figura subsequente, que mostra as soluções no filtro na geração 100, com (direito) e sem (esquerda) o uso destes operadores respectivamente.

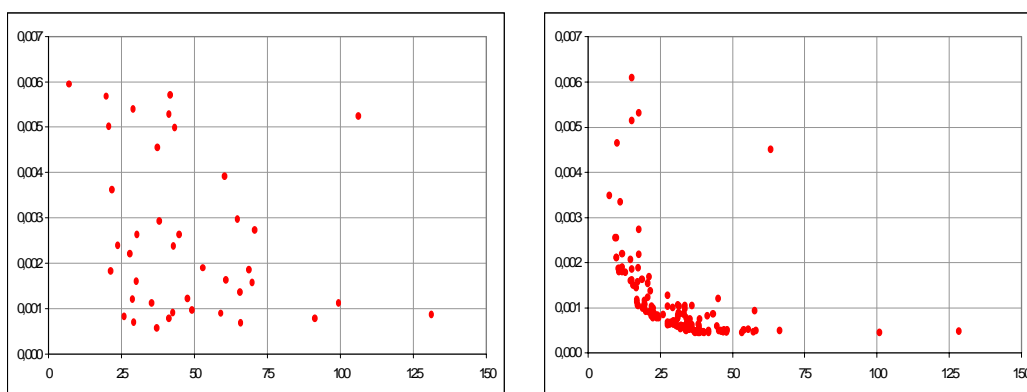


Figura 31 - Soluções viáveis nas gerações inicial e centésima, respectivamente.

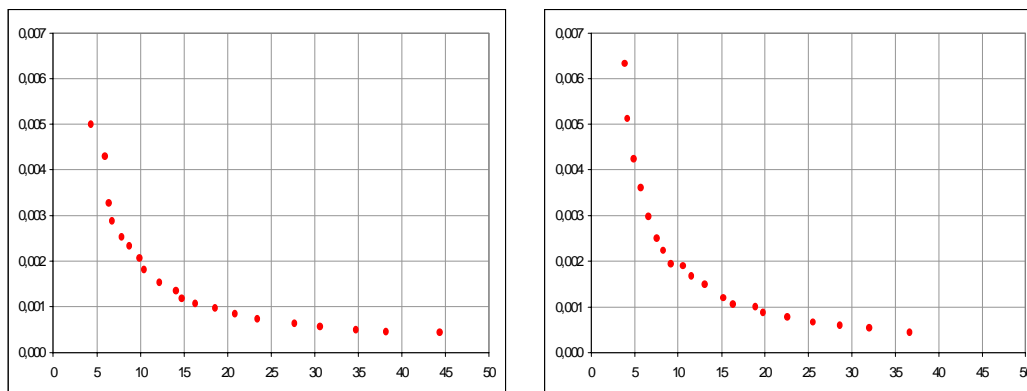


Figura 32 - Soluções no filtro na geração 100: sem (esquerda) e com (direita) o uso dos operadores de adição e de otimização individual respectivamente.

Os resultados podem ser melhorados caso se aumente o número de gerações adicionais que atuam no filtro para 400, como pode ser observado na figura e tabela seguintes.

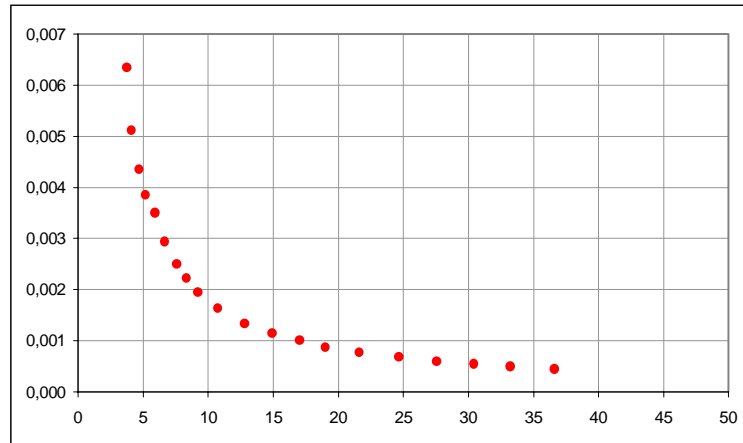


Figura 33 - Soluções do filtro na geração 500 (100 + 400).

h	l	t	b	fo1	fo2
0,393	2,888	9,304	0,430	3,742	0,00634
0,317	3,662	9,926	0,439	4,112	0,00511
0,396	3,091	9,981	0,508	4,707	0,00434
0,393	2,927	9,933	0,582	5,209	0,00385
0,413	3,004	9,810	0,666	5,909	0,00349
0,393	2,898	9,926	0,765	6,672	0,00293
0,454	2,471	9,948	0,893	7,600	0,00250
0,739	1,498	9,937	1,005	8,348	0,00223
0,798	1,246	9,966	1,140	9,213	0,00195
1,182	0,787	10,000	1,343	10,770	0,00163
1,136	0,782	10,000	1,648	12,830	0,00133
1,184	0,748	9,942	1,953	14,930	0,00114
1,182	0,903	10,000	2,181	17,030	0,00101
1,182	0,753	10,000	2,519	19,040	0,00087
0,870	1,057	10,000	2,867	21,650	0,00077
1,184	0,787	9,922	3,324	24,680	0,00068
1,182	0,754	9,923	3,753	27,590	0,00060
1,334	0,750	10,000	4,086	30,470	0,00054
1,182	0,748	10,000	4,524	33,250	0,00049
1,182	0,748	10,000	5,000	36,630	0,00044

Tabela 2 - Valores das variáveis de projeto e das funções objetivo do filtro final.

4.4. Conclusões

Neste problema foram obtidos resultados equivalentes ou superiores aos apresentados pela literatura [17, 21], comprovando a eficiência e aplicabilidade do algoritmo desenvolvido. Conclui-se também pela grande importância dos novos operadores desenvolvidos no âmbito da otimização multi-objetivos, pois quando empregados melhoram significativamente os resultados encontrados.

5. Treliça com Três Objetivos

5.1. Introdução e Modelo

A treliça plana da figura abaixo está sujeita a dois tipos de carregamento e a três objetivos de interesse. O primeiro objetivo é o peso estrutural, o segundo é o deslocamento vertical u_1 ocorrido pela aplicação do primeiro tipo de carregamento e o terceiro objetivo é o deslocamento vertical u_2 oriundo do segundo tipo de carregamento aplicado. Os deslocamentos e tipos de carregamento estão também na figura abaixo.

Este problema foi abordado na referência [13], diferenciando-se dos problemas multi-objetivo correntemente estudados não só pela presença de três objetivos ao invés de dois, como também pela consideração de diferentes tipos de carregamento para cada um dos deslocamentos verticais a serem minimizados, u_1 e u_2 .

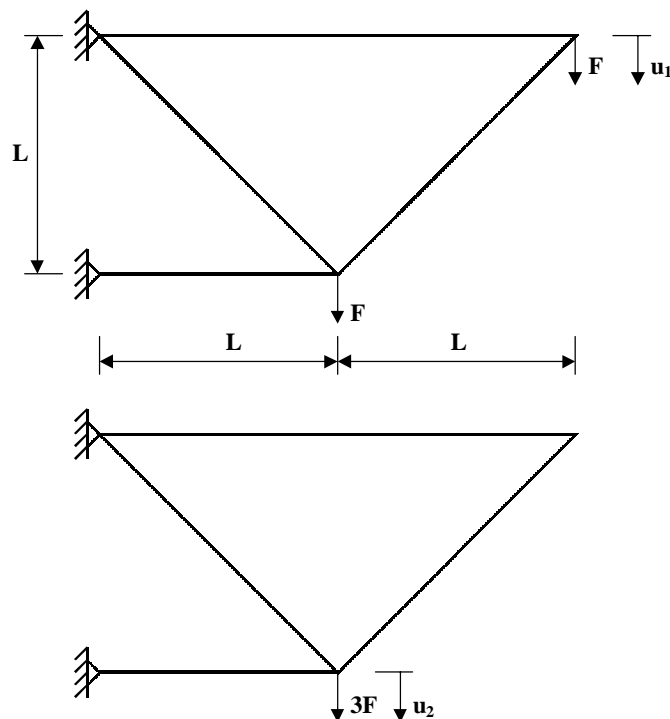


Figura 34 - Treliça com três objetivos

Os objetivos são conflitantes entre si, já que quando o peso diminui os deslocamentos tendem a aumentar e vice-versa.

As variáveis de projeto são as áreas das quatro barras que foram consideradas contínuas e variando dentro do seguinte intervalo:

$$0.1 \leq A_i \leq 8.0$$

Os demais dados do problema são:

$$L = 200; \quad E = 20000; \quad F = 10$$

As restrições do problema são as tensões normais desenvolvidas nas barras:

$$| \sigma_i | \leq 10$$

5.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operadores convencionais de mutação e recombinação uniforme.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).
- Aplicação de uma penalização conforme apresentado adiante.
- Probabilidade de Recombinação = 85 %.
- Probabilidade de Mutação = 5%.
- Tamanho da população = 100.
- Tamanho do filtro = 400.
- Número de gerações = 50 + 50 (filtro).

O tamanho do filtro foi tomado o mesmo adotado na referência [13], quatrocentos, para facilitar as comparações e conclusões.

As variáveis de projeto foram codificadas numa cadeia binária com quarenta posições de comprimento, sendo dez posições para cada uma delas.

O método de penalidade aplicado é descrito pela expressão seguinte:

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) + c_i \cdot nviol$$

onde $c_1=10000$; $c_2=20$ e $c_3=20$ e $nviol$ é o número de restrições violadas. Os valores das constantes c_i garantem que uma solução que viole uma restrição seja pior que uma solução viável, da mesma forma que uma solução que viole duas restrições seja pior que uma que viole uma única restrição e assim por diante.

5.3. Resultados

Nas figuras seguintes são apresentados os resultados obtidos para os diversos espaços disponíveis, sendo que os gráficos foram elaborados com as mesmas escalas que aquelas apresentadas na referência [13] para facilidade de comparações.

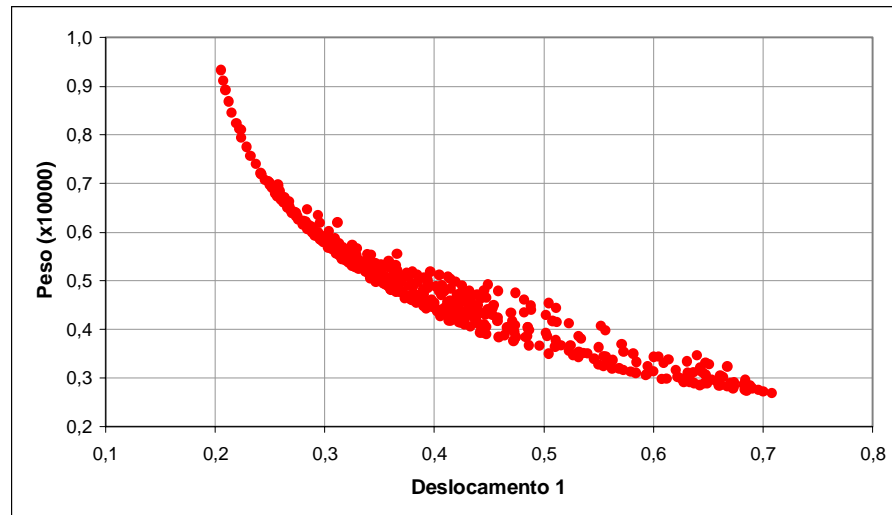


Figura 35 - Filtro no espaço $P \times u_1$

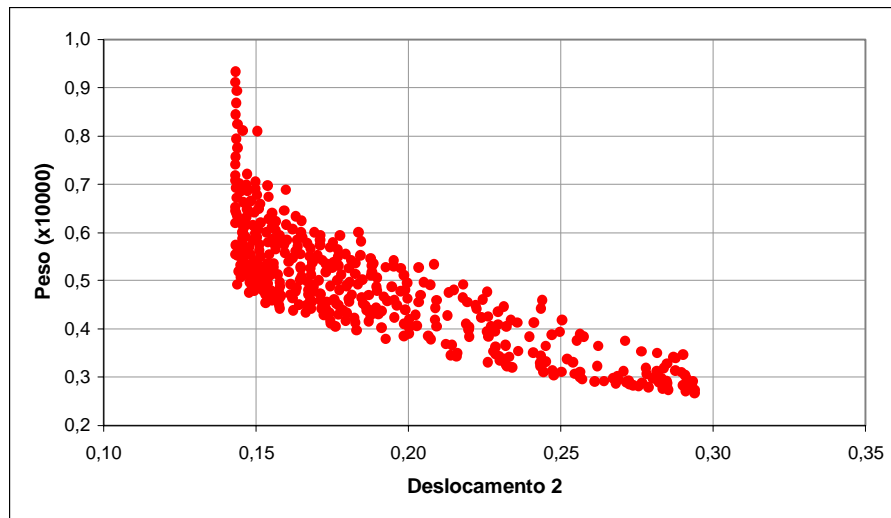


Figura 36 - Filtro no espaço $P \times u_2$

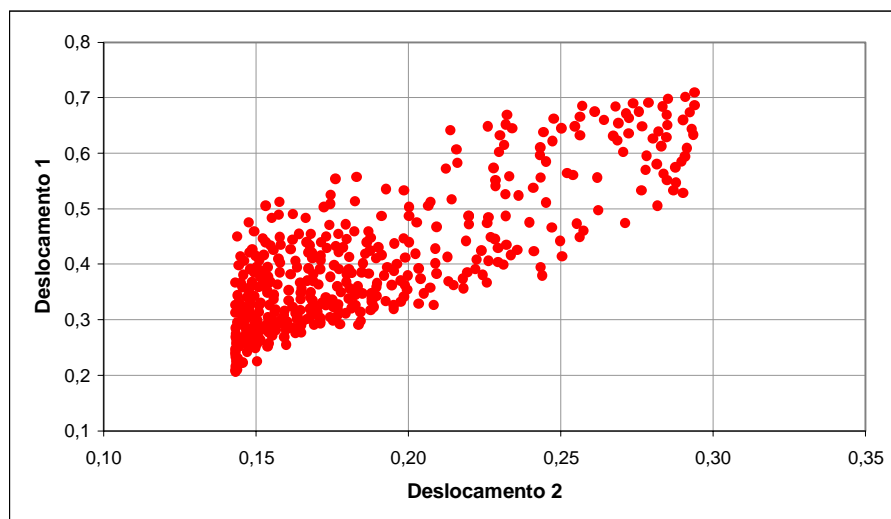


Figura 37 - Filtro no espaço $u_1 \times u_2$

A próxima figura mostra a distribuição das soluções armazenadas no filtro no espaço das três funções objetivo do problema. Este gráfico não é apresentado na referência [13].

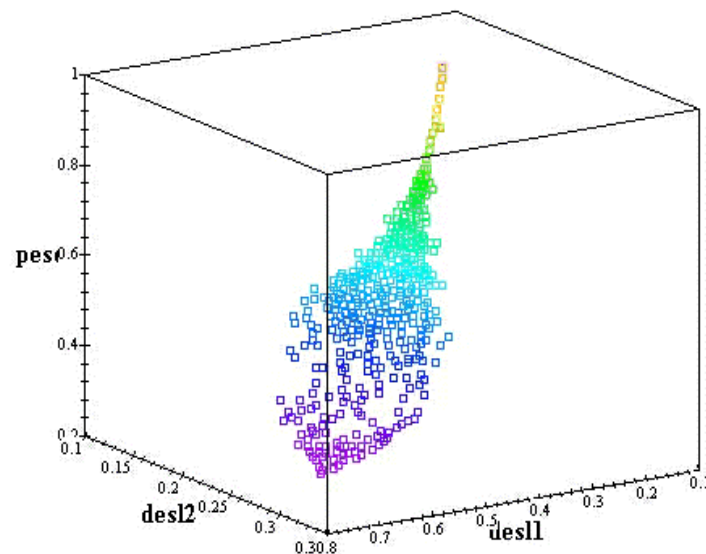


Figura 38 - Filtro no espaço $P \times u_1 \times u_2$ (figura não disponível em [13])

Os valores numéricos mínimos e máximos dos objetivos são mostrados na próxima tabela, bem como algumas das soluções armazenadas no filtro final:

Solução	A1	A2	A3	A4	Peso	u1	u2
1	1,09	1,42	4,32	3,08	2674,00	0,709	0,294
100	1,95	1,89	6,77	4,00	4032,00	0,486	0,200
200	1,48	1,64	7,61	7,92	4787,00	0,459	0,150
300	2,53	4,15	7,06	6,76	5536,00	0,316	0,165
400	8,00	8,00	8,00	8,00	9325,00	0,206	0,144

Tabela 3 - Valores numéricos de algumas soluções do filtro final armazenado

5.4. Conclusões

Comparando o tratamento aqui empregado ao problema em relação à referência [13] podem-se destacar as seguintes diferenças: menor tamanho de cromossomo; quatro vezes menos indivíduos por população; mesmo número final de gerações, embora com melhorias que diminuem o esforço computacional; nenhum emprego de tratamento de restrições sofisticado; além da ausência de estratégia de nicho ou de parâmetros que tornam a resolução dos problemas sensíveis às suas configurações. Ainda assim, os resultados apresentados foram superiores àqueles encontrados em [13], pois apresentaram as soluções melhor distribuídas pela região de Pareto.

6. Ponte Trelaçada

6.1. Introdução e Modelo

O problema ilustrado na figura seguinte foi extraído, com modificações, da referência [42]. Os objetivos são a minimização do peso estrutural (f_1) e do deslocamento vertical u (f_2) do nó central inferior.

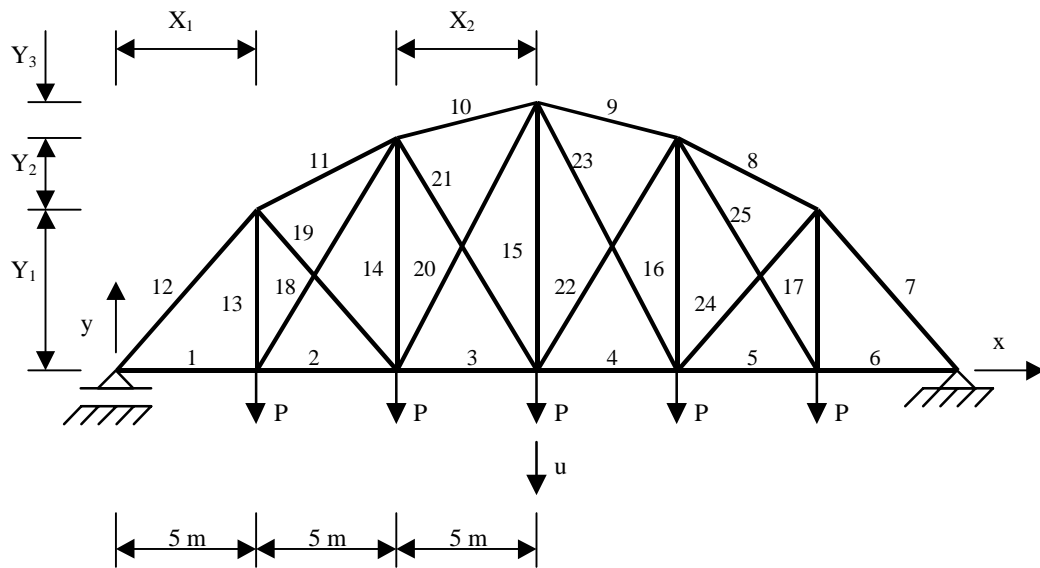


Figura 39 - Ponte treliçada

As variáveis de projeto são as áreas das vinte e cinco barras além das coordenadas dos nós superiores. São também impostas condições de simetria, deste modo, são suficientes treze variáveis (A_1 até A_{13}) de projeto para a representação das vinte e cinco áreas de barras, além de cinco variáveis (Y_1 , Y_2 , Y_3 , X_1 e X_2) para a localização dos nós superiores, conforme Figura 10. Todos os nós inferiores são fixos e as treze variáveis de projeto referentes às áreas estruturais dos elementos são indicadas na seguinte tabela:

Variável	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
Elementos	1, 6	2, 5	3, 4	7, 12	8, 11	9, 10	13, 17	14, 16	15	18, 25	19, 24	20, 23	21, 22

Tabela 4 - Relação entre as variáveis de projeto e áreas dos elementos estruturais

As áreas são discretas e retiradas de um conjunto de trinta e duas áreas possíveis que foram escolhidas das quarenta e duas indicadas pelo American Institute of Steel Construction Manual, estando apresentadas na tabela abaixo.

1.62	1.80	1.99	2.13	2.38	2.62	2.88	3.13
3.38	3.55	3.87	4.18	4.49	4.59	4.80	4.97
5.12	5.74	7.22	7.97	11.5	13.5	13.9	14.2
15.5	16.0	16.9	18.8	19.9	22.0	22.9	26.5

Tabela 5 - Conjunto de áreas consideradas (in²)

As restrições laterais sobre as variáveis de projeto correspondentes ao posicionamento dos nós superiores são:

$$0,25 \text{ m} \leq Y_1, Y_2 \text{ e } Y_3 \leq 3,0 \text{ m}$$

$$3,0 \text{ m} \leq X_1, \text{ e } X_2 \leq 7,0 \text{ m}$$

Os dados considerados para o problema são:

- $P = 10^5 \text{ N}$
- $E = 2,1 \times 10^{11} \text{ N/m}^2$
- $\rho = 7850 \text{ Kg/m}^3$
- $u_{\text{máx.}} = 0,03 \text{ m}$
- $|\sigma_{\text{máx.}}| = 1,3 \times 10^8 \text{ N/m}^2$

6.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.

- Execução única (uma só semente).
- Aplicação de uma penalização conforme apresentado adiante.
- Probabilidade de Recombinação = 90 %.
- Probabilidade de Mutação = 2%.
- Tamanho da população = 300.
- Tamanho do filtro = 20.
- Número de gerações = 100 + 400 (filtro).

O problema foi codificado em cromossomos binários de cento e vinte posições de comprimento, sendo cinco posições para cada variável representativa de área e nove posições para as demais variáveis.

Foi utilizado um método simples de penalidade dado pela expressão seguinte:

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) + c_i \cdot nviol$$

onde:

- i é o índice da função objetivo considerada (1 ou 2);
- os valores das constantes c_i são: $c_1=15000$ e $c_2=0,03$;
- $nviol$ é o número de violações de uma dada solução.

6.3. Resultados

Na figura posterior pode ser vista parte da população inicial e da geração 100. Observa-se que na população inicial existem apenas três soluções viáveis, algumas outras, duas vezes penalizadas, e o restante que não aparece na figura pelo número maior de penalizações. Já na população correspondente à geração 100 observa-se grande número de soluções viáveis, duas soluções uma vez penalizadas e outras duas vezes penalizadas.

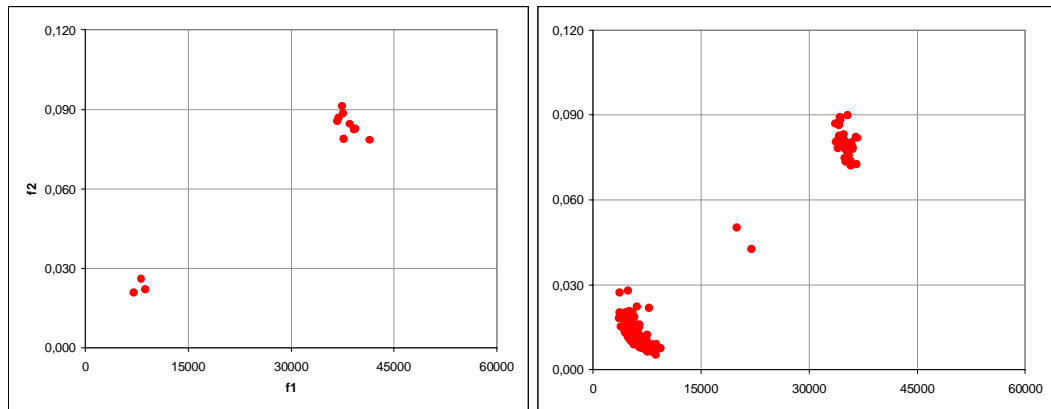


Figura 40 - População inicial e da geração 100 respectivamente

Na figura abaixo é apresentado o filtro final do processo de otimização.

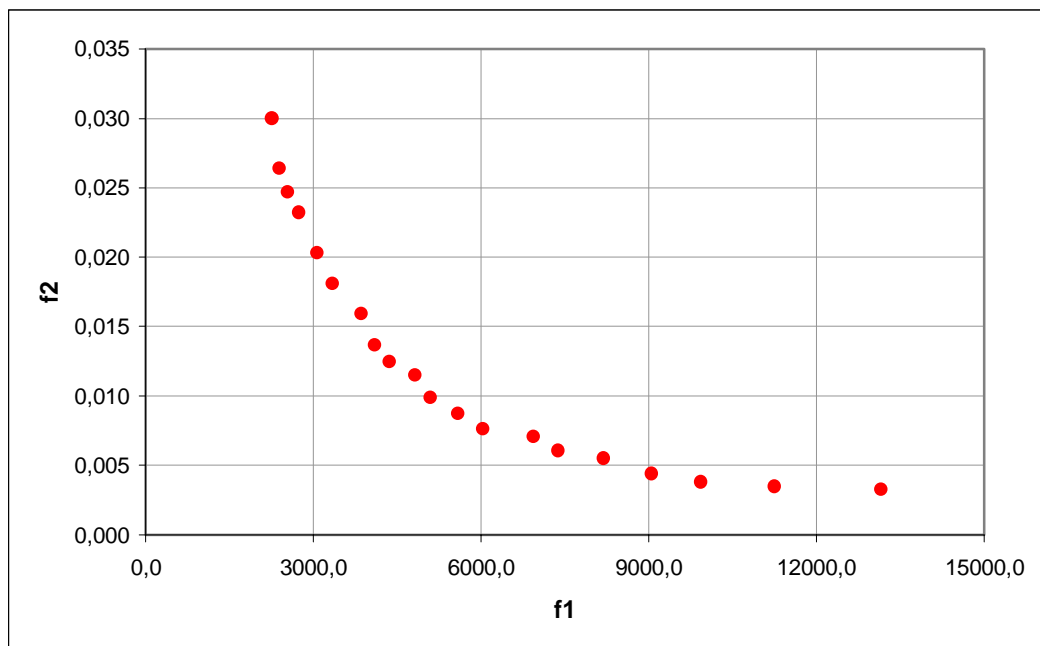


Figura 41 - Soluções armazenadas no filtro final do processo de otimização

Na próxima figura é mostrada a evolução do processo de minimização dos objetivos 1 e 2 respectivamente. Embora pelos gráficos apresentados tenha-se a impressão de que o processo de otimização possa ser interrompido com um número bem inferior de gerações, é preciso destacar que o número maior de gerações é importante para a melhor distribuição e definição do conjunto ótimo de Pareto.

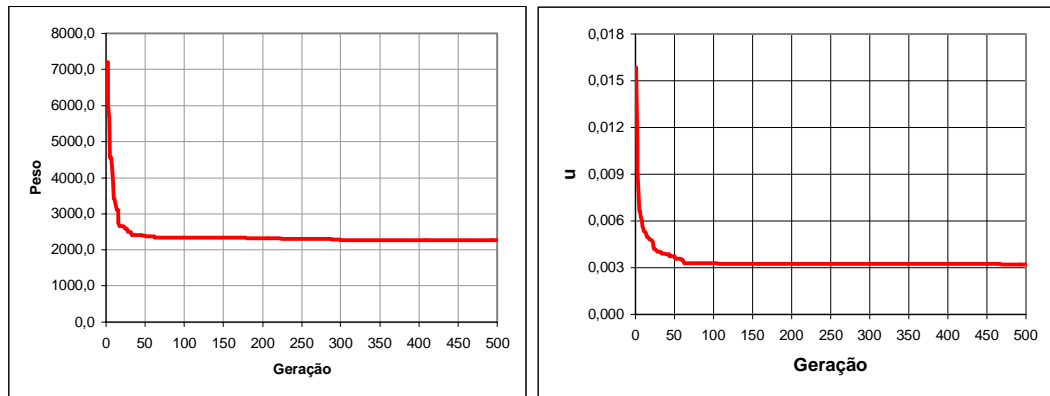


Figura 42 - Evolução das minimizações individuais dos objetivos

Nas duas figuras seguintes são apresentadas as formas das soluções que alcançaram os valores mínimos para os objetivos 1 e 2 respectivamente.

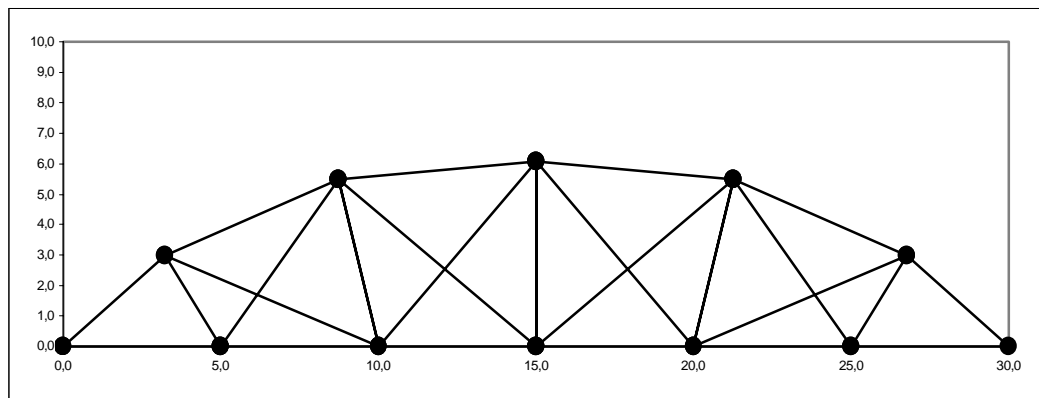


Figura 43 - Solução que apresentou o peso mínimo (f_1)

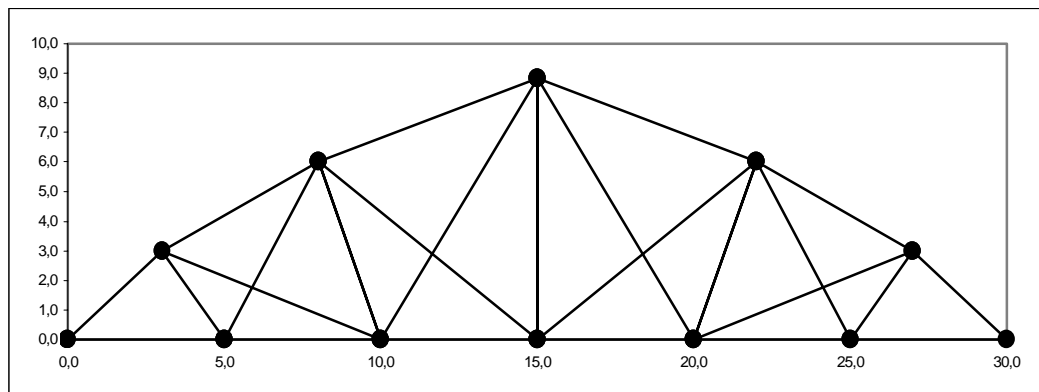


Figura 44 - Solução que apresentou o deslocamento vertical mínimo (f_2)

Os valores numéricos bem como as áreas das seções transversais podem ser observadas na tabela abaixo.

Solução	1	5	10	15	20
A1	5,12	5,12	7,22	13,90	26,50
A2	3,55	4,97	18,80	13,90	26,50
A3	5,12	5,12	7,22	14,20	26,50
A4	5,12	11,50	7,22	16,00	26,50
A5	5,12	11,50	13,90	22,00	26,50
A6	5,12	5,12	14,20	22,00	26,50
A7	1,62	1,62	1,62	2,62	26,50
A8	1,62	1,62	2,62	2,88	26,50
A9	1,62	1,62	3,55	16,90	26,50
A10	1,62	1,62	2,38	1,62	1,62
A11	1,62	1,62	1,80	3,13	1,62
A12	1,62	1,62	1,62	1,80	1,62
A13	1,62	1,62	1,99	1,62	1,62
Y1	2,984	2,984	2,989	2,909	3,000
Y2	2,510	2,941	2,822	2,919	3,000
Y3	0,594	1,283	1,924	2,360	2,833
X1	3,243	3,227	3,000	3,055	3,000
X2	6,256	6,256	6,890	6,914	7,000
f1	2262,0	3079,0	4835,0	7378,0	13160,0
f2	0,02997	0,02027	0,01148	0,00610	0,00322

Tabela 6 - Valores numéricos de algumas soluções do filtro final armazenado

6.4. Conclusões

Este problema só existe na literatura em forma modificada na referência [42] e com otimização simples. A fronteira ótima de Pareto encontrada ficou bem distribuída, facilitando a tarefa de projeto e destacando a importância de uma otimização estrutural, pois são possíveis grandes variações nos objetivos estudados: peso estrutural e deslocamento central, facilitando sobremaneira a etapa de projeto.

7. Fronteira de Pareto Descontínua

7.1. Introdução e Modelo

Este problema multi-objetivo foi um dos apresentados por Schaffer na referência [58]. Posteriormente, ele foi também resolvido em [06, 25] por outros algoritmos genéticos e autores. O objetivo é minimizar as duas funções objetivo definidas por:

$$f_1 = \begin{cases} -x & \rightarrow x \leq 1 \\ -2 + x & \rightarrow 1 < x \leq 3 \\ 4 - x & \rightarrow 3 < x \leq 4 \\ -4 + x & \rightarrow x > 4 \end{cases}$$

$$f_2 = (x-5)^2$$

com $-10 \leq x \leq 10$

Este problema é muito utilizado para testar algoritmos por suas características particulares, simplicidade e regiões não-dominadas disjuntas ou descontínuas. O conjunto de Pareto é determinado pelos intervalos $1 \leq x \leq 2$ e $4 \leq x \leq 5$.

7.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.

- População inicial aleatória.
- Execução única (uma só semente).
- Número de cromossomos para torneio = 3.
- Probabilidade de Recombinação = 90 %.
- Probabilidade de Mutação = 5%.
- Tamanho da população = 50.
- Tamanho do filtro = 20.
- Número de gerações = 20 + 80 (filtro).

O problema foi codificado de maneira binária em cromossomos de trinta e duas posições de comprimento como utilizado em [58].

7.3. Resultados

Nas próximas três figuras são apresentadas respectivamente a população inicial, a população da geração vinte e o filtro final encontrado.

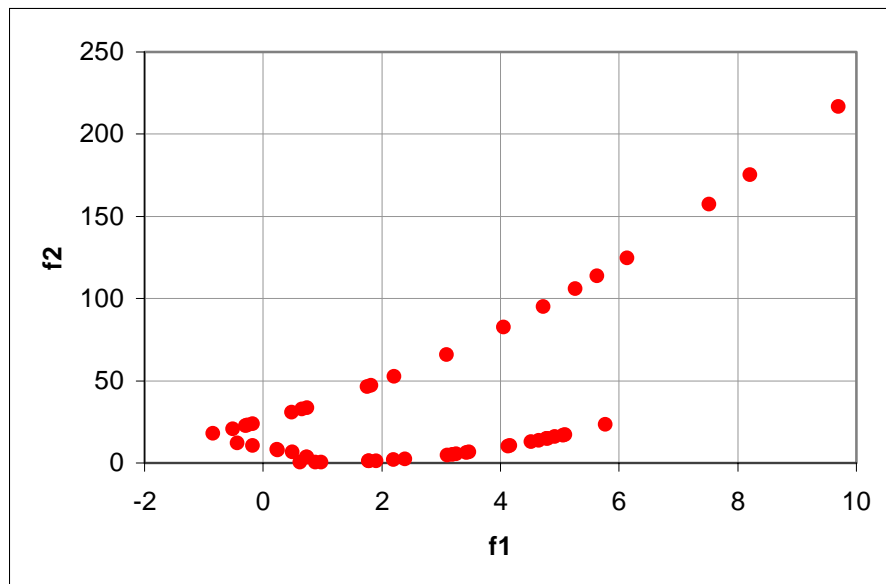


Figura 45 - População inicial

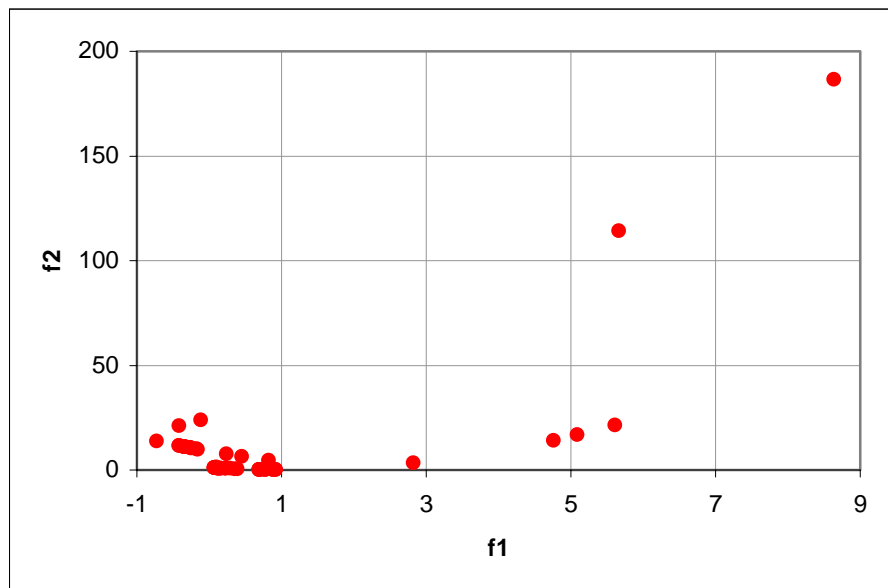


Figura 46 - População na geração 20

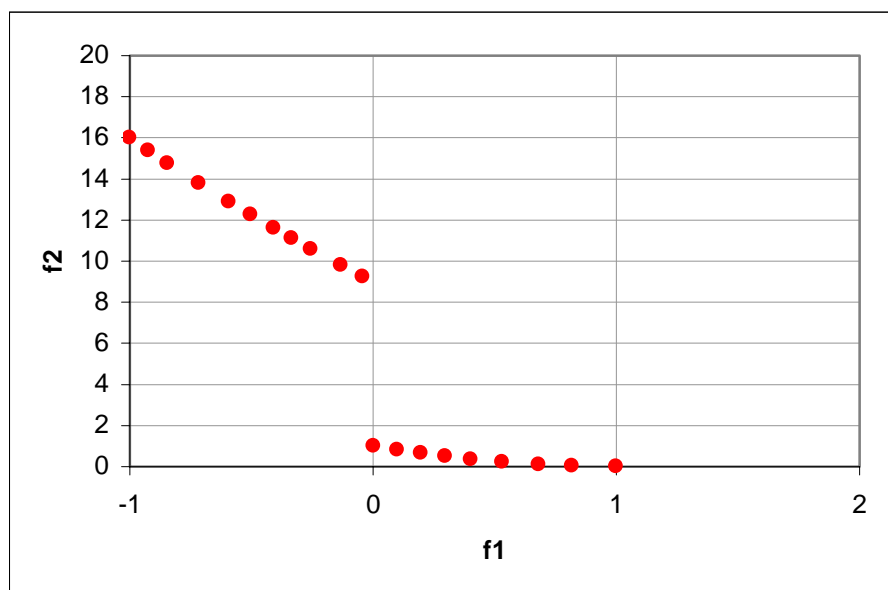


Figura 47 - Filtro final na geração 100 (20+80)

Durante a resolução deste problema uma nova melhoria foi desenvolvida para o algoritmo utilizado, que consta da possibilidade do mesmo funcionar como um algoritmo do tipo em regime. Isto é possível com a recolocação das soluções armazenadas no filtro corrente na próxima população. Em geral, este procedimento deve acelerar o processo de busca durante as gerações e foi chamado de **elitismo do filtro**.

O mesmo problema foi resolvido com esta nova metodologia e utilizou a mesma população inicial anterior e configurações, o que permite a comparação de resultados e a comprovação do melhor desempenho da modificação proposta.

- Introdução do elitismo do filtro.

Os novos resultados encontrados são mostrados nas figuras seguintes:

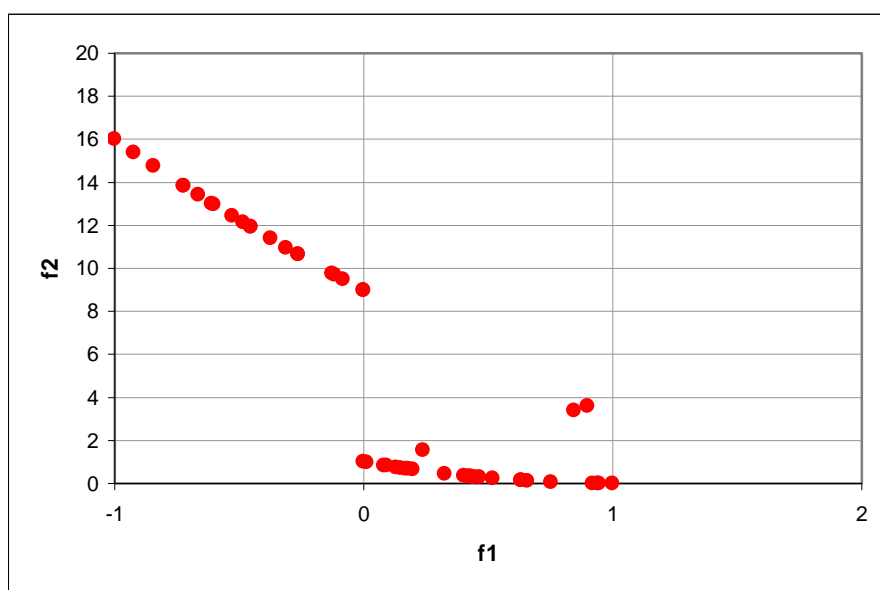


Figura 48 - População na geração 20

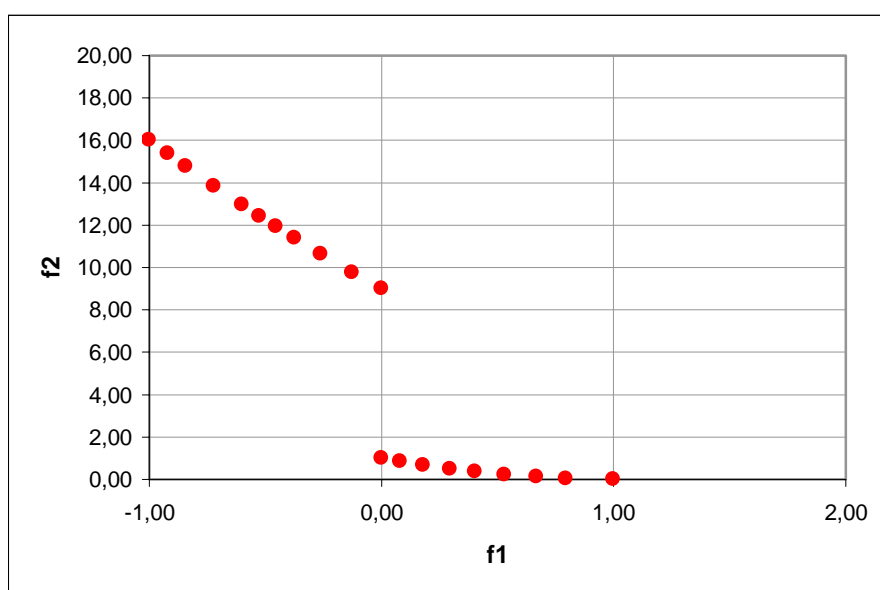


Figura 49 - Filtro final na geração 100 (20+80)

Na tabela abaixo são relacionados os valores da variável de projeto com os das funções objetivo das soluções finais armazenadas no filtro:

x	f1	f2
1,00	-1,00	16,00
1,08	-0,92	15,38
1,16	-0,84	14,77
1,28	-0,72	13,85
1,40	-0,60	12,97
1,47	-0,53	12,43
1,55	-0,45	11,92
1,63	-0,37	11,39
1,74	-0,26	10,64
1,87	-0,13	9,77
2,00	0,00	9,00
4,00	0,00	1,00
4,08	0,08	0,84
4,18	0,18	0,67
4,30	0,30	0,50
4,40	0,40	0,36
4,53	0,53	0,22
4,67	0,67	0,11
4,79	0,79	0,04
5,00	1,00	0,00

Tabela 7 - Valores da variável de projeto e das funções objetivo do filtro final

7.4. Conclusões

A descontinuidade do conjunto de Pareto, característica que certamente ocasiona problemas diversos em métodos da programação matemática, foi resolvida normalmente pelo algoritmo desenvolvido, comprovando a robustez da técnica.

Os resultados encontrados foram melhores que aqueles apresentados nas referências [06, 25, 58], mesmo com a utilização de uma configuração mais modesta do algoritmo genético – menor número de gerações e tamanho da população. Foi utilizado também um tamanho de filtro inferior, vinte soluções, para deixar claro a boa distribuição das soluções pela fronteira de Pareto.

Uma nova metodologia foi acoplada ao algoritmo durante a resolução deste problema – o elitismo do filtro – que aumenta a pressão de seleção e aumenta a velocidade de convergência do algoritmo.

8. Funções-teste para Otimizadores Multi-objetivos

8.1. Introdução e Modelo

Um conjunto de seis funções-teste para problemas multi-objetivo foi desenvolvido por Deb [71, 72], sendo desde então o mais elaborado e diversificado pacote de funções-teste para avaliação e comparação de algoritmos multi-objetivos que aproximam o conjunto de Pareto.

Nas referências [71, 72] são comparados oito algoritmos otimizadores:

- seis algoritmos genéticos multi-objetivos bem conhecidos (FFGA, NPGA, HLGA, VEGA, NSGA e SPEA);
- um algoritmo evolucionário de otimização simples (SOEA) utilizando o método da agregação de pesos somados;
- um algoritmo de busca randômica (RAND).

Este trabalho é na atualidade a melhor comparação realizada entre algoritmos otimizadores aplicados à problemas multi-objetivos, tanto pelos muitos algoritmos considerados como pelo conjunto desenvolvido de funções-teste, o qual aborda uma grande variedade de características que trazem dificuldades na resolução do problema.

Os principais resultados e conclusões alcançados com o trabalho [71, 72] podem ser resumidos pelos itens seguintes, que foram livremente traduzidos das referências apontadas:

- As funções-teste sugeridas provêm complexidade suficiente para comparar diferentes algoritmos otimizadores multi-objetivo. Funções multimodais e/ou enganosas [71, 72] são as que causam maiores dificuldades para aproximações evolucionárias. Entretanto, não-convexidade é também uma característica na qual algoritmos baseados na soma de pesos tendem a apresentar dificuldades.

- Uma hierarquia clara surgiu da utilização dos algoritmos com respeito à distância até o conjunto de Pareto, estando tal hierarquia apresentada de forma descendente na lista seguinte:

- **SPEA** (Zitzler and Thiele 1998/1999)
- **NSGA** (Srinivas and Deb 1994)
- **VEGA** (Schaffer 1985)
- **HLGA** (Hajela and Lin 1992)
- **NPGA** (Horn, Nafpliotis and Goldberg 1994)
- **FFGA** (Fonseca and Fleming 1993)

Existe um claro salto de desempenho entre os algoritmos SPEA e NSGA, assim como entre o algoritmo NSGA e os demais. As fronteiras encontradas pelos algoritmos VEGA, HLGA, NPGA e FFGA estão muito próximas entre si. Contudo, os resultados indicam que o VEGA é sensivelmente superior a seus concorrentes diretos, enquanto que o NPGA atinge fronteiras mais próximas do ótimo global que o FFGA. Além disso, é notório que tanto o VEGA como HLGA apresentam dificuldades na obtenção de fronteiras bem distribuídas na função não-convexa.

- Elitismo é um fator importante no processo de otimização evolucionária multi-objetivos. Fato comprovado pelos seguintes fatores: o algoritmo SPEA: i) claramente supera todos os demais algoritmos em cinco das seis funções-teste e ii) é o único algoritmo que incorpora o elitismo como foco central do processo evolutivo. Por outro lado, o desempenho dos demais algoritmos melhorou significativamente quando a estratégia de elitismo presente no SPEA foi a eles incorporada. Preliminarmente, os resultados mostram que o NSGA com elitismo tem desempenho igual ao SPEA.

As funções-teste desenvolvidas em [71, 72] cobrem as diferentes e difíceis características presentes em problemas multi-objetivos – *convexidade, não-convexidade, fronteiras discretas ou descontínuas, multimodalidade, problemas enganosos e espaços de busca tendenciosos*. Deste modo, pode-se efetuar uma comparação sistemática que considera diferentes tipos de complexidade e permite conclusões sobre quais técnicas e estratégias levam vantagens e/ou desvantagens sobre as demais.

Cada uma das funções-teste compreende um problema de minimização com dois objetivos e são estruturadas de uma forma única, isto é, definidas como se segue:

$$\begin{aligned} \text{Minimizar} \quad & \tau(\mathbf{x}) = (f_1(x_1), f_2(\mathbf{x})) \\ \text{Sujeito a} \quad & f_2(\mathbf{x}) = g(x_2, \dots, x_m) \cdot h(f_1(x_1), g(x_2, \dots, x_m)) \\ \text{Onde} \quad & \mathbf{x} = (x_1, \dots, x_m) \end{aligned}$$

A função f_1 é uma função apenas da primeira variável de projeto ou decisão, g é uma função das $(m-1)$ variáveis de projeto restantes e os parâmetros de h são os valores das funções de f_1 e g . As funções-teste diferem-se: i) por estas três funções; ii) pelo número de variáveis m e iii) pelo domínio das variáveis de projeto. Assim, as seis funções τ_1, \dots, τ_6 podem ser introduzidas pelas definições que se seguem:

- Função-teste τ_1 que tem uma fronteira ótima de Pareto convexa:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2, m} x_i / (m-1) \\ h(f_1, g) &= 1 - (f_1/g)^{0.5} \end{aligned}$$

onde $m = 30$ e $x_i \in [0, 1]$. A fronteira ótima de Pareto é formada com $g(\mathbf{x}) = 1$.

- Função-teste τ_2 que tem uma fronteira ótima de Pareto não-convexa:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2, m} x_i / (m-1) \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned}$$

onde $m = 30$ e $x_i \in [0, 1]$. A fronteira ótima de Pareto é formada com $g(\mathbf{x}) = 1$.

- Função-teste τ_3 que tem uma fronteira ótima de Pareto discreta, ou seja, composta por algumas partes descontínuas:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2, m} x_i / (m-1) \\ h(f_1, g) &= 1 - (f_1/g)^{0.5} - (f_1/g) \cdot \sin(10\pi f_1) \end{aligned}$$

onde $m = 30$ e $x_i \in [0, 1]$. A fronteira ótima de Pareto é formada com $g(\mathbf{x}) = 1$. A introdução da função seno em h provoca a descontinuidade da fronteira de Pareto. Contudo, não existe descontinuidade no espaço das variáveis de decisão ou projeto.

- Função-teste τ_4 que contem 21^9 fronteiras ótimas de Pareto e por esta razão verifica a habilidade do algoritmo de abordar multimodalidade:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 10 \cdot (m-1) + \sum_{i=2, m} (x_i^2 - 10 \cdot \cos(4\pi x_i)) \\ h(f_1, g) &= 1 - (f_1/g)^{0.5} \end{aligned}$$

onde $m = 10$, $x_1 \in [0, 1]$ e $x_2, \dots, x_m \in [-5, 5]$. A fronteira ótima de Pareto global é formada com $g(\mathbf{x}) = 1$ e a fronteira local com $g(\mathbf{x}) = 1.25$. Nem todos os conjuntos ótimos de Pareto são distinguíveis no espaço das funções objetivo.

- Função-teste τ_5 que descreve um problema enganoso, distinguindo-se da função-teste τ_4 pelo fato de x_1 representar uma cadeia binária:

$$\begin{aligned}f_1(x_1) &= 1 + u(x_1) \\g(x_2, \dots, x_m) &= \sum_{i=2,m} v(u(x_i)) \\h(f_1, g) &= 1/f_1\end{aligned}$$

onde $u(x_i)$ retorna o número de uns no vetor binário x_i ,

$$v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{se } u(x_i) < 5 \text{ ou} \\ 1 & \text{se } u(x_i) = 5 \end{cases}$$

onde $m = 11$, $x_1 \in \{0, 1\}^{30}$ e $x_2, \dots, x_m \in \{0, 1\}^5$. A verdadeira fronteira ótima de Pareto é formada com $g(\mathbf{x}) = 10$, enquanto que a melhor fronteira ótima de Pareto enganosa é representada pelas soluções em que $g(\mathbf{x}) = 11$. Tanto a fronteira global ótima de Pareto quanto a local são convexas.

- Função-teste τ_6 que inclui duas dificuldades causadas pela não uniformidade do espaço de busca: primeiramente, as soluções ótimas de Pareto não estão uniformemente distribuídas ao longo da fronteira global de Pareto (a fronteira é tendenciosa para soluções onde $f_1(\mathbf{x})$ está próximo de um); em segundo lugar, a densidade das soluções é menor perto da fronteira ótima de Pareto e maior na região afastada dela:

$$\begin{aligned}f_1(x_1) &= 1 - \exp(-4x_1) \cdot \sin^6(6\pi x_1) \\g(x_2, \dots, x_m) &= 1 + 9 \cdot \left(\sum_{i=2,m} x_i / (m-1) \right)^{0.25} \\h(f_1, g) &= 1 - (f_1/g)^2\end{aligned}$$

onde $m = 10$ e $x_i \in [0, 1]$. A fronteira ótima de Pareto é formada com $g(\mathbf{x}) = 1$ e é não-convexa.

8.2. Metodologia de Comparação

Para testar o algoritmo proposto ele foi utilizado na otimização das seis funções-teste apresentadas, através de cinco execuções aleatórias para cada problema, para que seus resultados fossem defrontados com os apresentados em [71, 72, 73], que são decorrentes das cinco primeiras execuções do algoritmo SPEA, o qual se mostrou superior a todos os demais algoritmos utilizados nas referências citadas.

8.3. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual ($n=2$ e $n2=2$).
- População inicial aleatória.
- Execução única (uma só semente para cada execução independente).
- Elitismo do filtro.

As configurações utilizadas pelos dois algoritmos foram:

- Número de gerações = 250.
- Tamanho da população = 80 + 20 (filtro).
- Probabilidade de recombinação = 80%.
- Probabilidade de mutação = 1%.
- Parâmetro de Nicho $\sigma_{share} = 0.48862$ (SPEA apenas).
- Pressão de dominância $t_{dom} = 10$.

Os dois algoritmos utilizam o armazenamento das melhores soluções durante o processo evolutivo – elitismo. Assim, a população de ambos foi dividida em oitenta cromossomos para a população corrente e vinte para o armazenamento das soluções elitistas.

Pelos resultados apresentados na referência [73], dá-se a entender que das cem soluções finais foram retirados os indivíduos dominados para a formação da fronteira de Pareto, pois o número de soluções apresentadas ultrapassa vinte - tamanho do filtro. Já o resultado do algoritmo desenvolvido foi restringido àquele armazenado no filtro, o que facilita e intensifica a avaliação da uniformidade das soluções encontradas pela fronteira ótima de Pareto.

Em todos os problemas cada variável foi codificada por trinta bits, com exceção do quinto problema, cujas variáveis x_2, \dots, x_m foram codificadas em apenas cinco bits cada. Estes números indicam que na primeira função-teste, por exemplo, são necessários cromossomos com novecentas posições para codificar as trinta variáveis de projeto do problema.

8.4. Número de Avaliações

Embora os dois algoritmos utilizem elitismo, os mesmos operam de maneira diferenciada, isto é, enquanto o algoritmo SPEA trabalha com uma população “independente” do filtro, o algoritmo PMOGA copia o filtro para a população da próxima geração. Assim, um menor número de avaliações são realizadas por geração.

Nas comparações realizadas nas referências [71, 72] todos os algoritmos trabalharam com cem avaliações por geração (população de cem indivíduos). Todavia, como o algoritmo SPEA emprega elitismo ele foi configurado para trabalhar com oitenta indivíduos na população e vinte no filtro. Não ficou claro, entretanto, qual o número de avaliações realizadas pelo algoritmo: se oitenta ou cem por geração. A seguir é apresentada uma descrição do número de avaliações para cada AG:

i) SPEA

- Todas as gerações
 - 80 ou 100 avaliações por geração. Provavelmente 100, já que nada foi escrito a respeito nas referências [71, 72]

ii) PMOGA

- Primeira geração (população inicial):
 - 80 avaliações
- Demais gerações:
 - 60 avaliações
 - 8 avaliações decorrentes do operador de adição
 - 12 avaliações decorrentes do operador de otimização individual
 - Totalizando 80 avaliações

8.5. Resultados

Os resultados são oriundos de cinco execuções aleatórias, destacando que nas referências [71, 72] as fronteiras exibidas são resultantes da união das fronteiras de cinco execuções, depois de retiradas as soluções dominadas. Este procedimento não é o ideal, uma vez que a união de resultados de diferentes execuções pode formar uma boa fronteira de Pareto a partir de duas ou mais fronteiras de má qualidade. Deste modo, as fronteiras apresentadas nestas referências são aqui reapresentadas juntamente com as fronteiras individualizadas de cada execução para possibilitar melhores conclusões.

São apresentados exemplos de populações iniciais para cada função-teste para ilustrar suas particularidades e dificuldades. Os resultados oriundos da utilização do algoritmo proposto estão disponíveis na referência [11].

No problema τ_3 o algoritmo desenvolvido foi empregado com o operador de adição ligado, mesmo com a característica de descontinuidade da fronteira de

Pareto do problema, o que poderia provocar dificuldades pelo fato do algoritmo sempre selecionar duas soluções em regiões descontínuas como as mais distantes. Contudo, ainda assim o desempenho do algoritmo foi excelente e os resultados não foram comprometidos.

Nas figuras das próximas páginas são apresentados os gráficos das análises realizadas, bem como alguns dos resultados das referências [71, 72]. Lembrando que as funções-teste apresentam as seguintes características e/ou dificuldades:

- 1 – função *convexa*;
- 2 – função *não-convexa*;
- 3 – função *discreta* ou *descontínua*;
- 4 – função *multimodal*;
- 5 – função *enganosa*;
- 6 – função *não uniforme* ou *tendenciosa*.

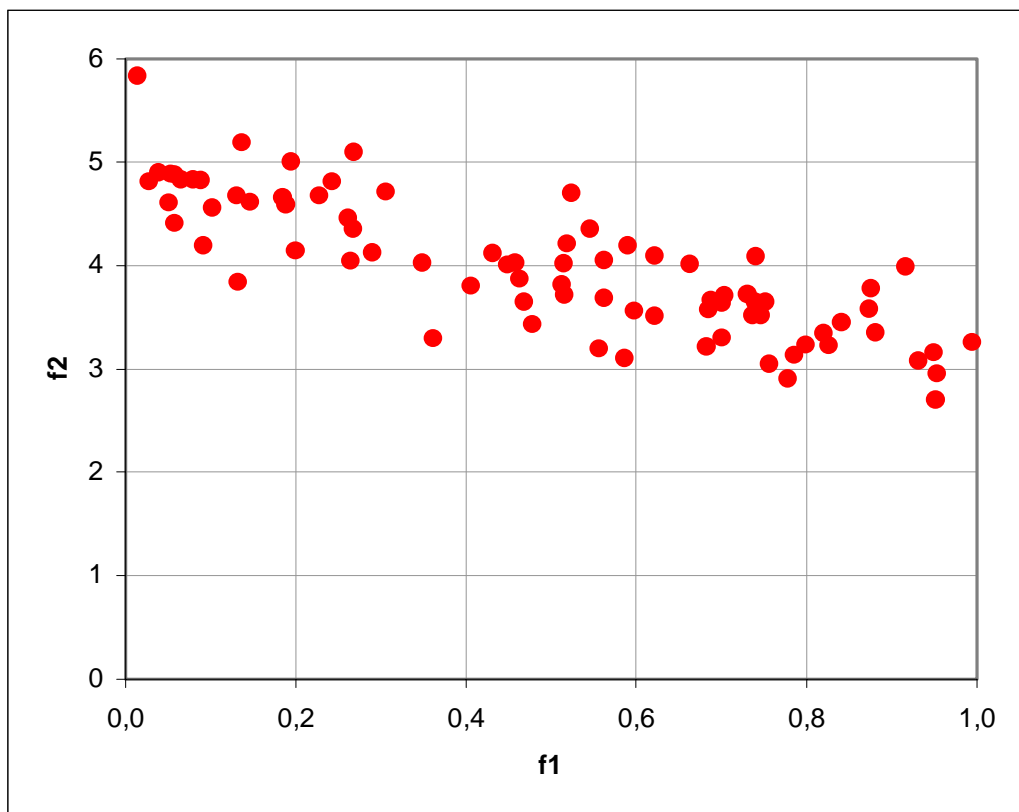


Figura 50 - Exemplo de população inicial - τ_1

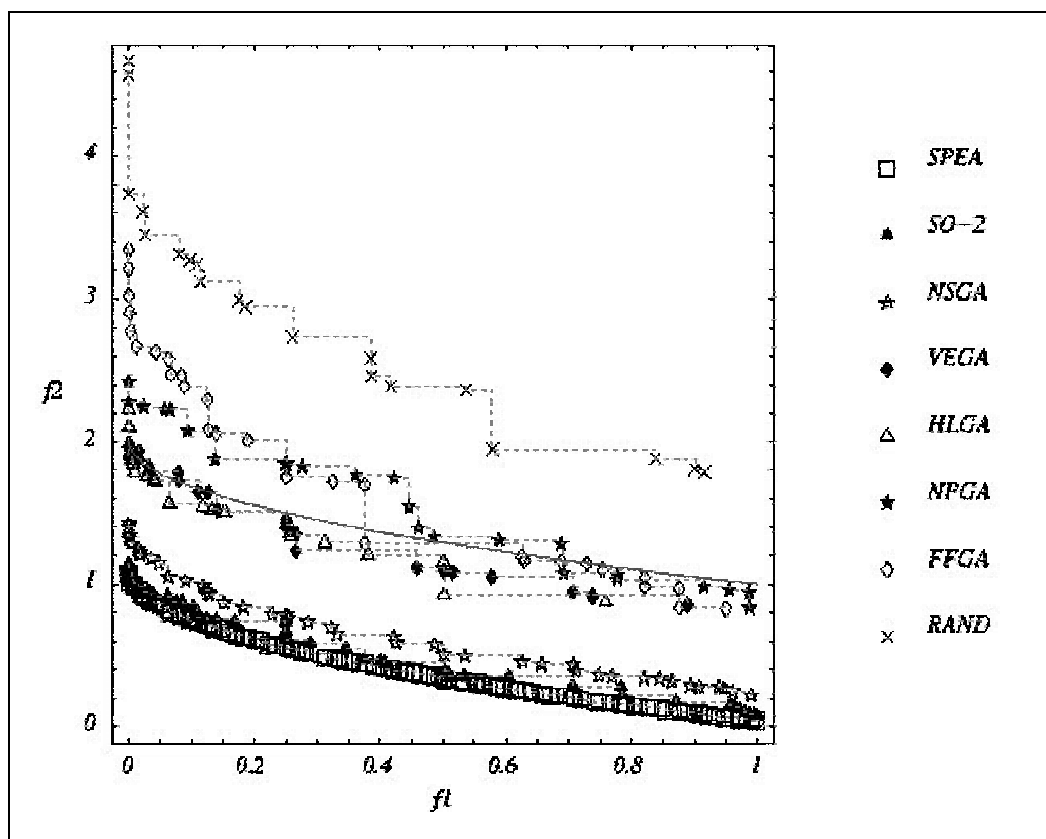


Figura 51 - Fronteiras de Pareto comparativas - τ_1 [72]

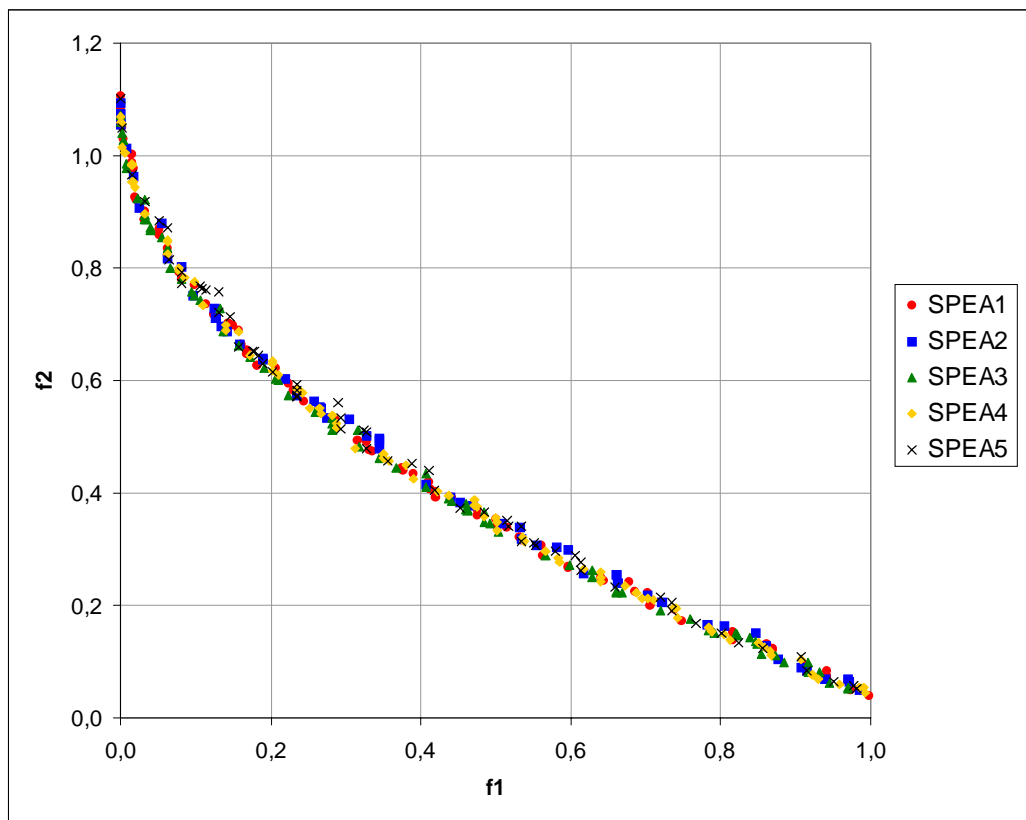


Figura 52 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_1 [73]

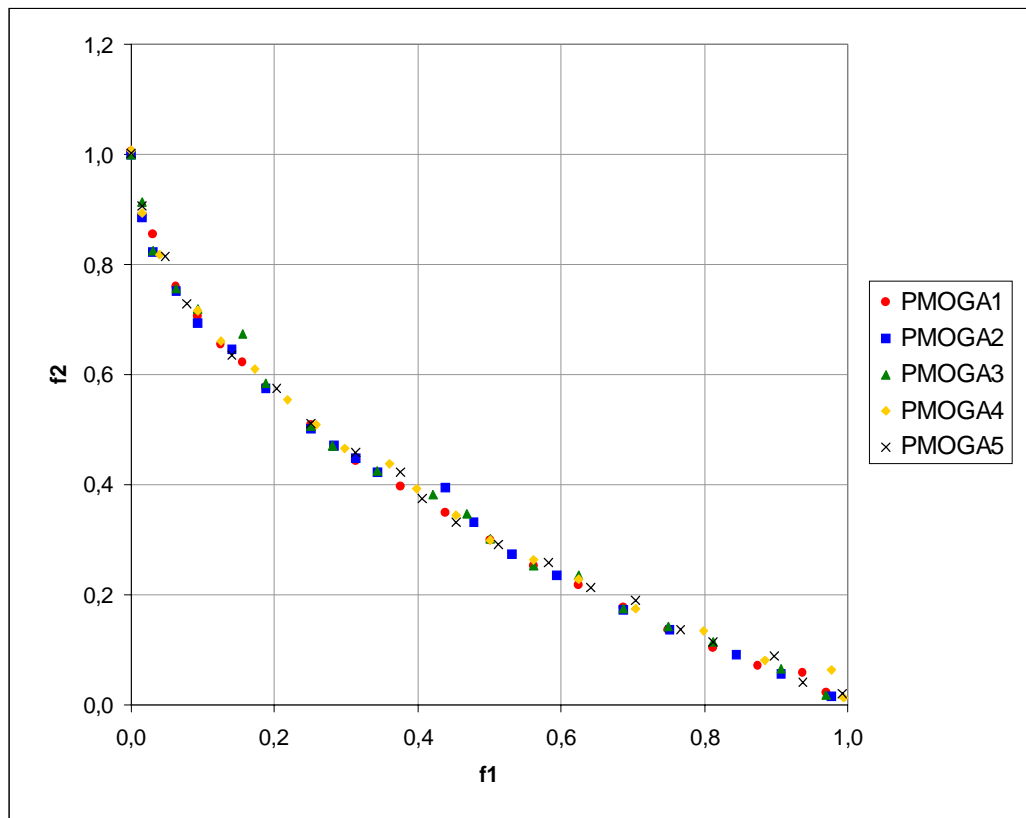


Figura 53 - Resultados de 5 execuções do algoritmo PMOGA - τ_1 [11]

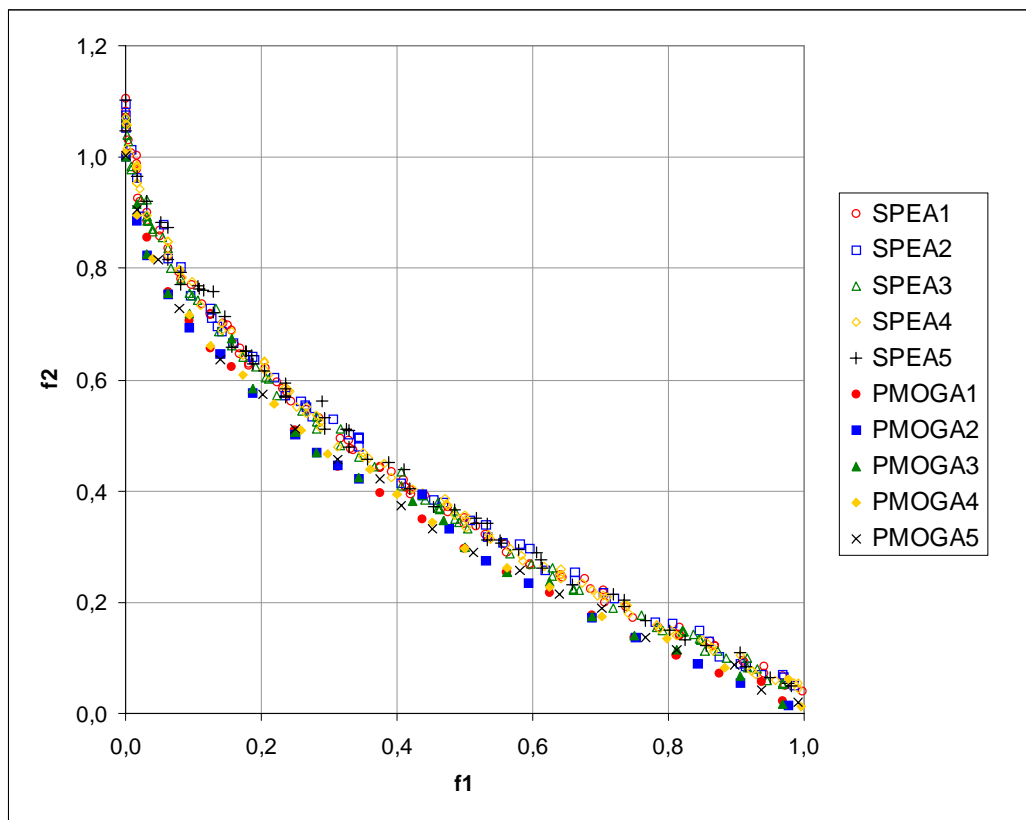


Figura 54 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_1 [11, 73]

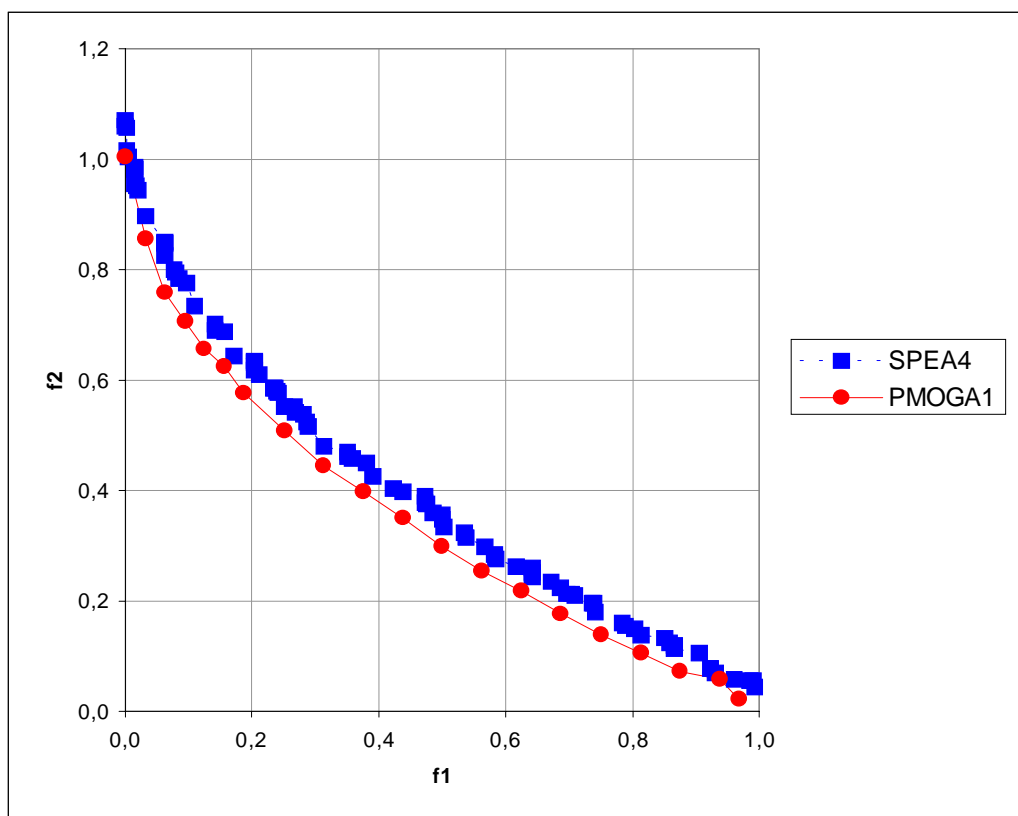


Figura 55 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_1 [11, 73]

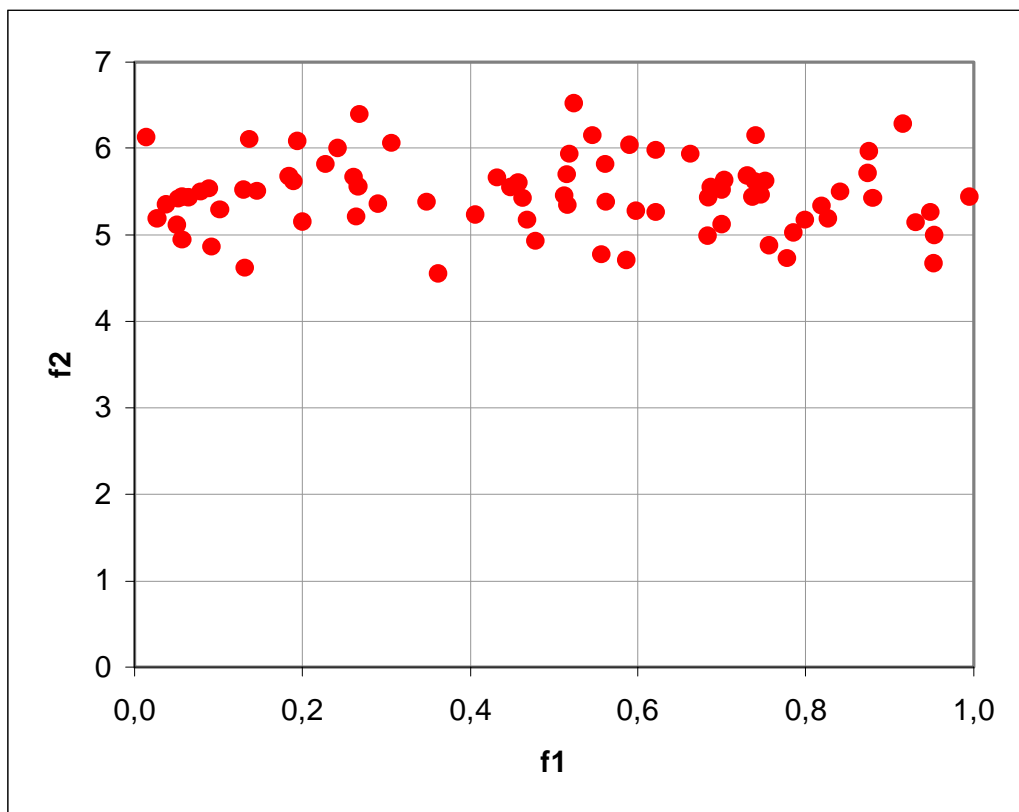


Figura 56 - Exemplo de população inicial - τ_2

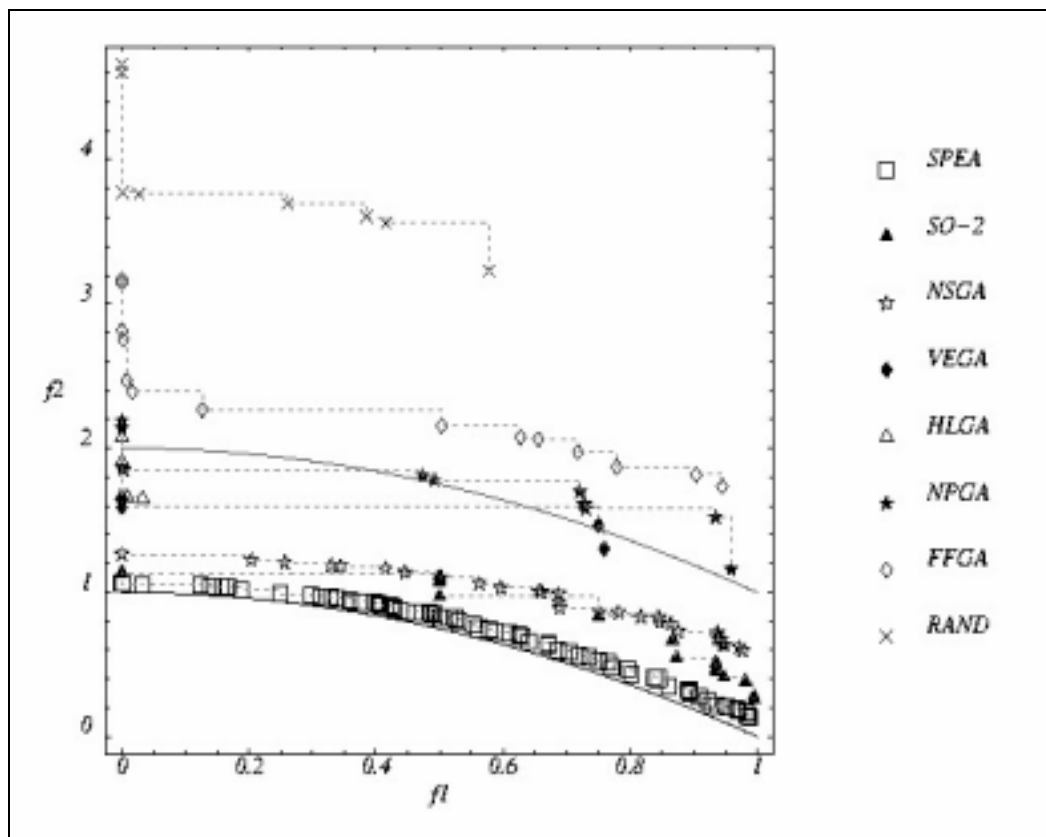


Figura 57 - Fronteiras de Pareto comparativas - τ_2 [72]

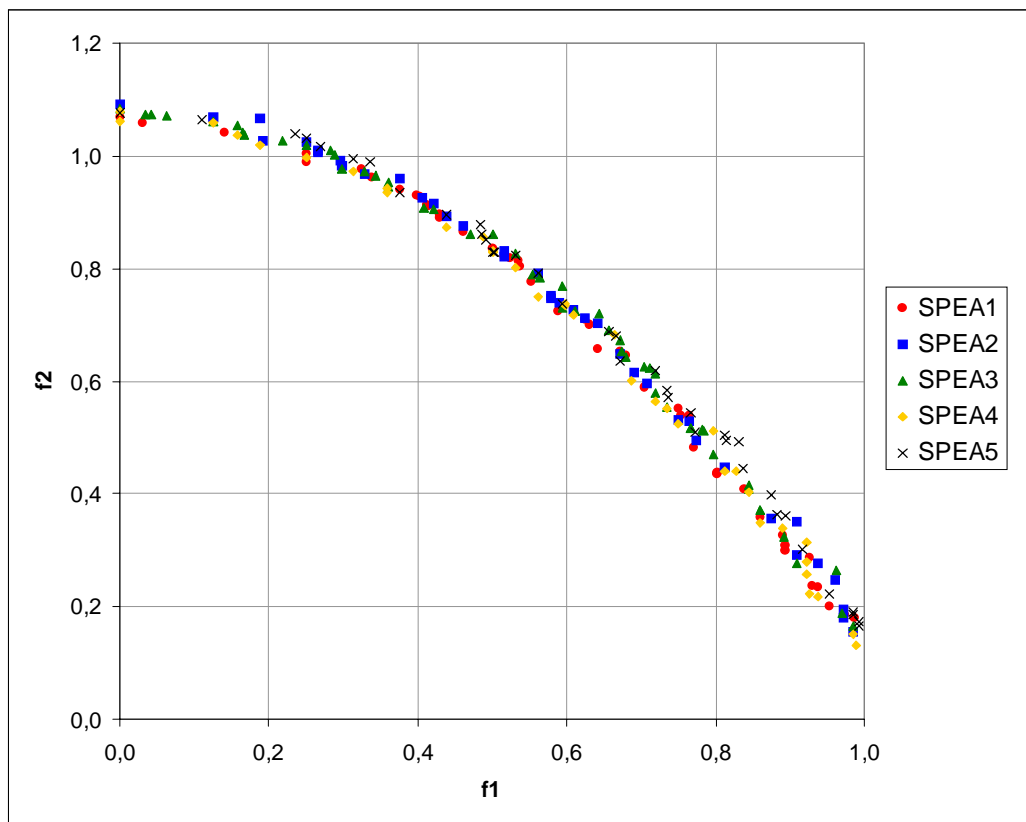


Figura 58 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_2 [73]

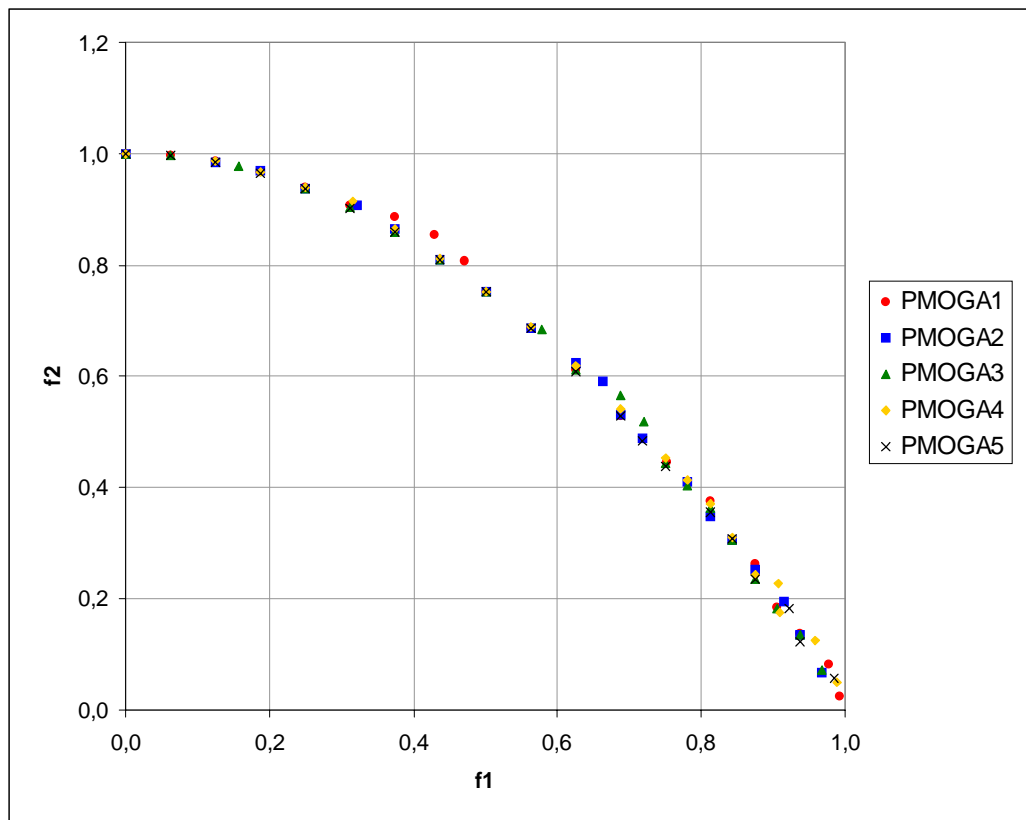


Figura 59 - Resultados de 5 execuções do algoritmo PMOGA - τ_2 [11]

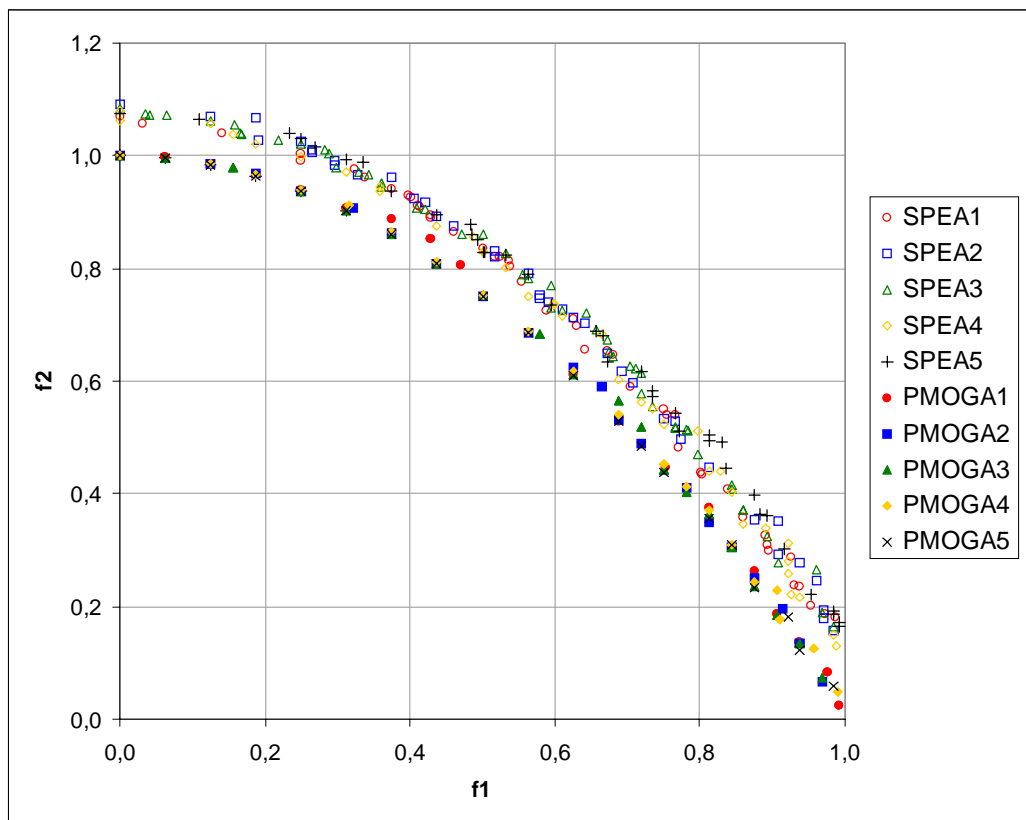


Figura 60 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_2 [11, 73]

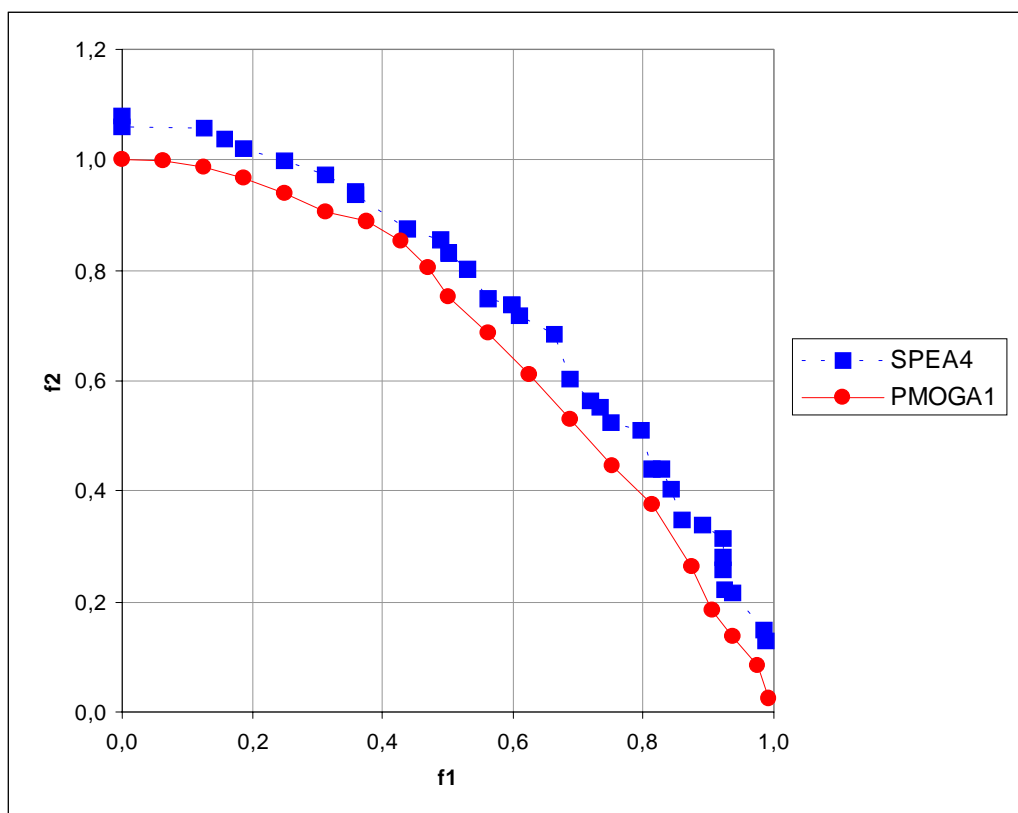


Figura 61 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_2 [11, 73]

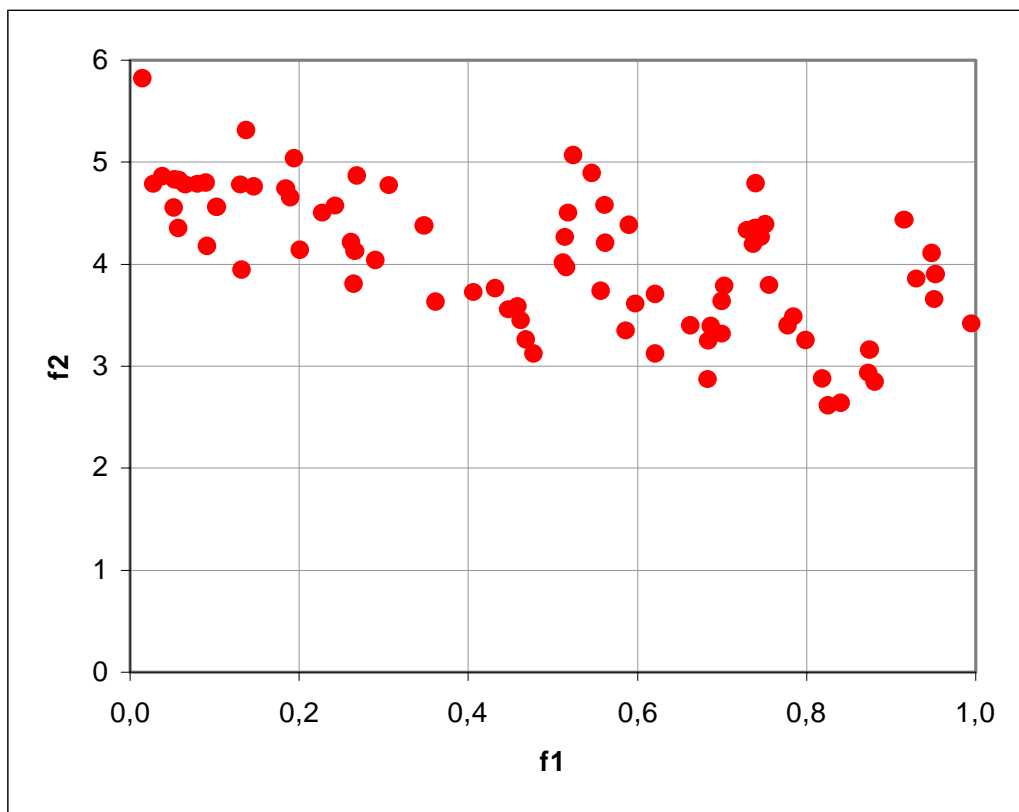


Figura 62 - Exemplo de população inicial - τ_3

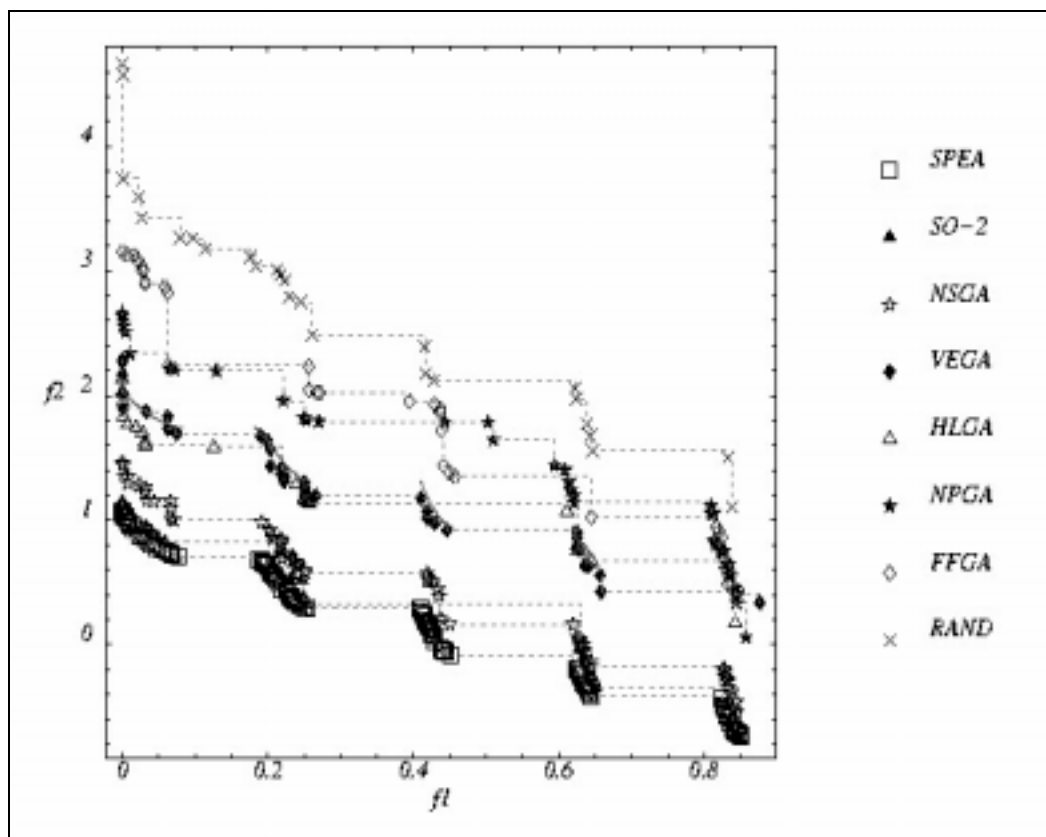


Figura 63 - Fronteiras de Pareto comparativas - τ_3 [72]

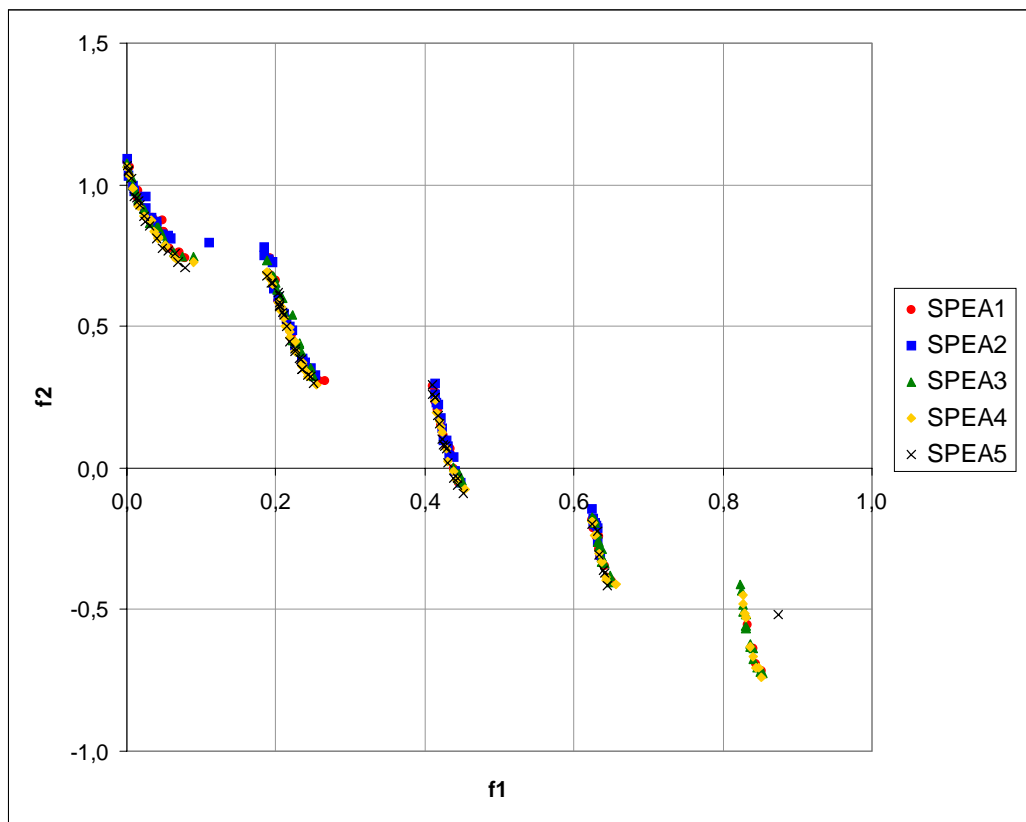


Figura 64 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_3 [73]

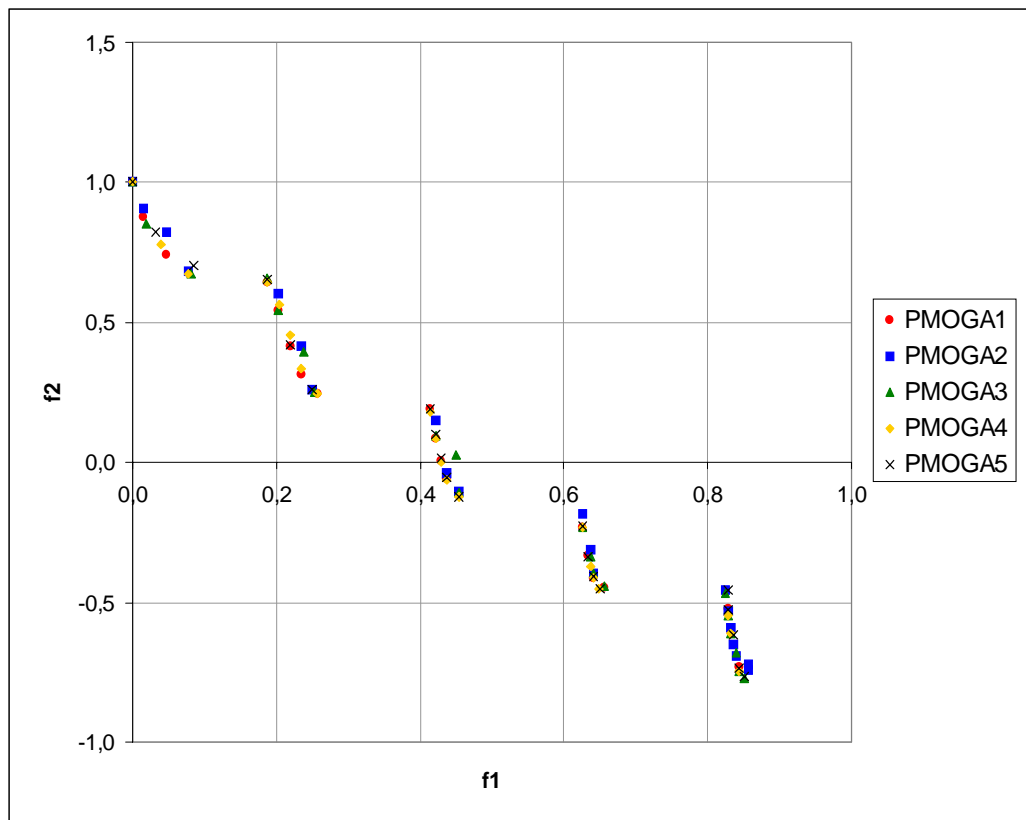


Figura 65 - Resultados de 5 execuções do algoritmo PMOGA - τ_3 [11]

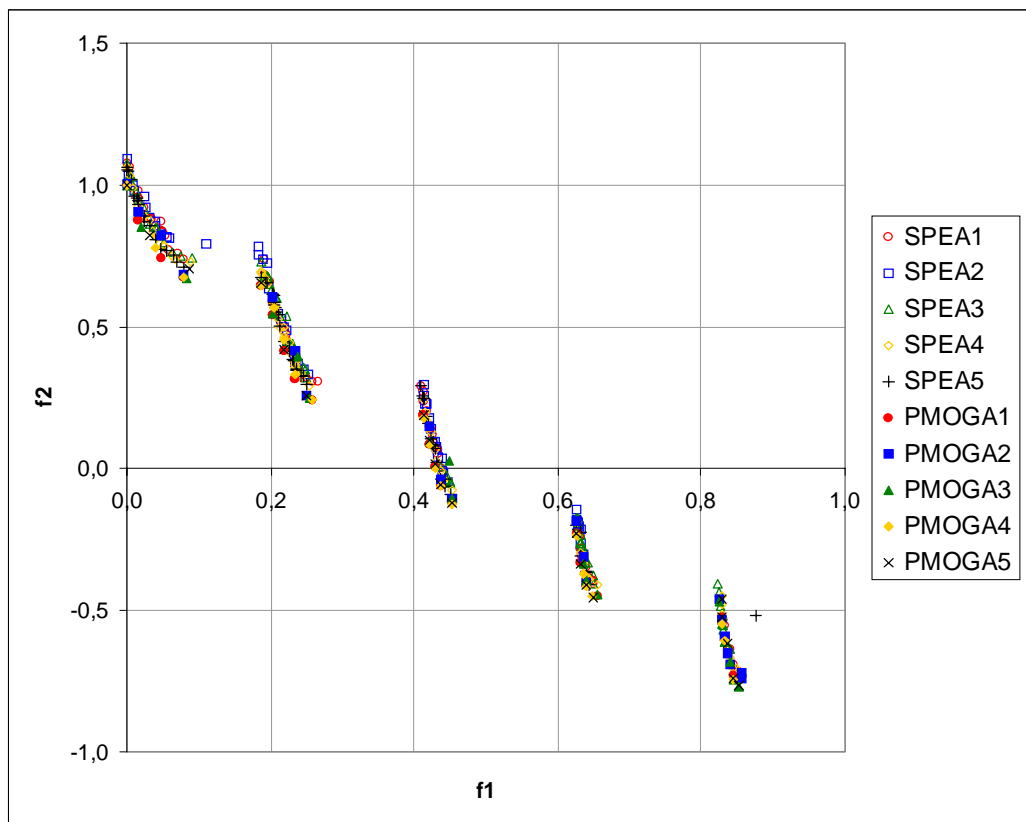


Figura 66 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_3 [11, 73]

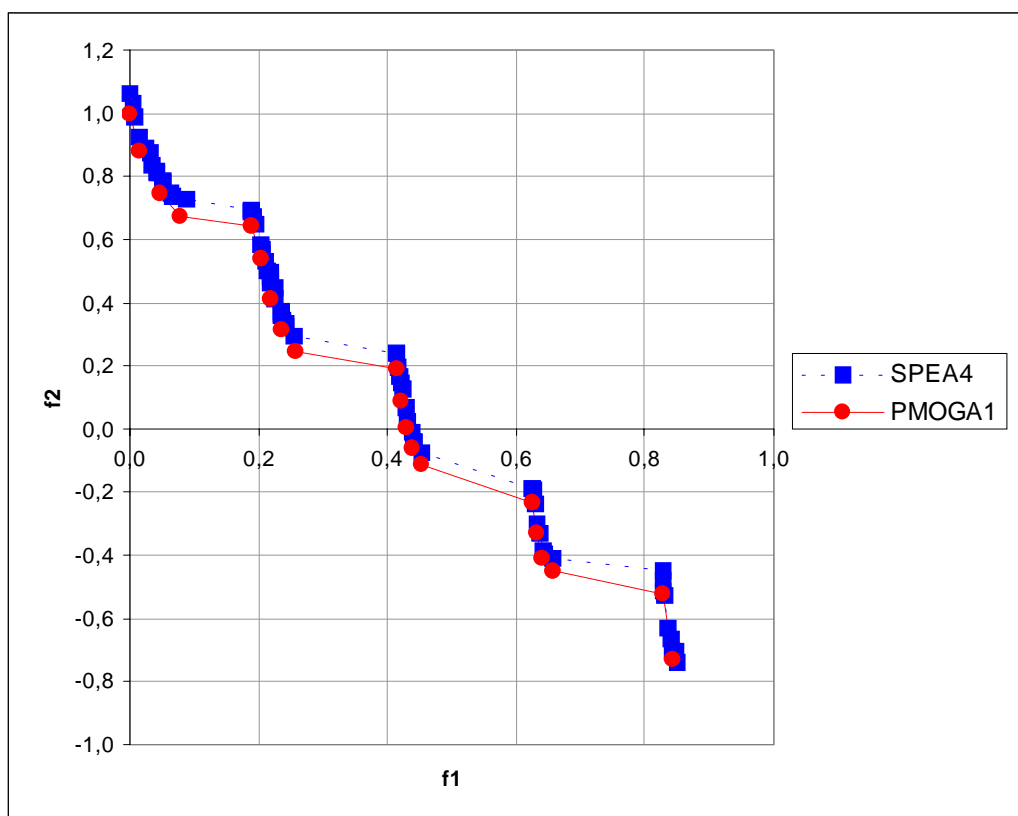


Figura 67 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_3 [11, 73]

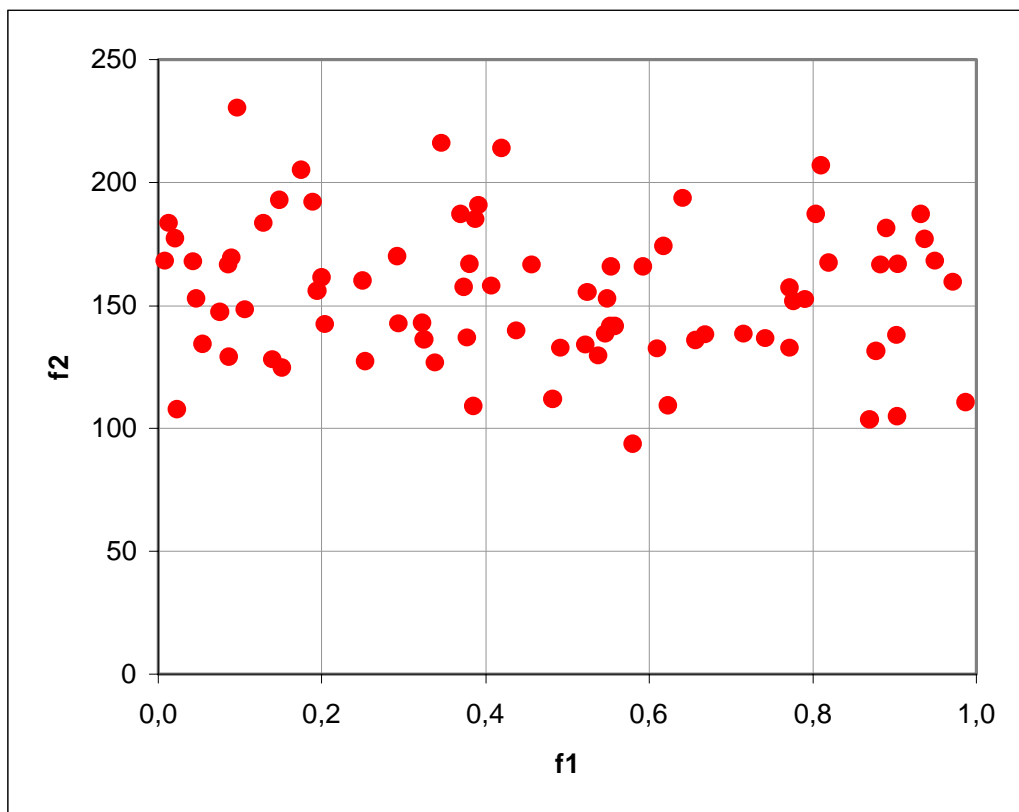


Figura 68 - Exemplo de população inicial - τ_4

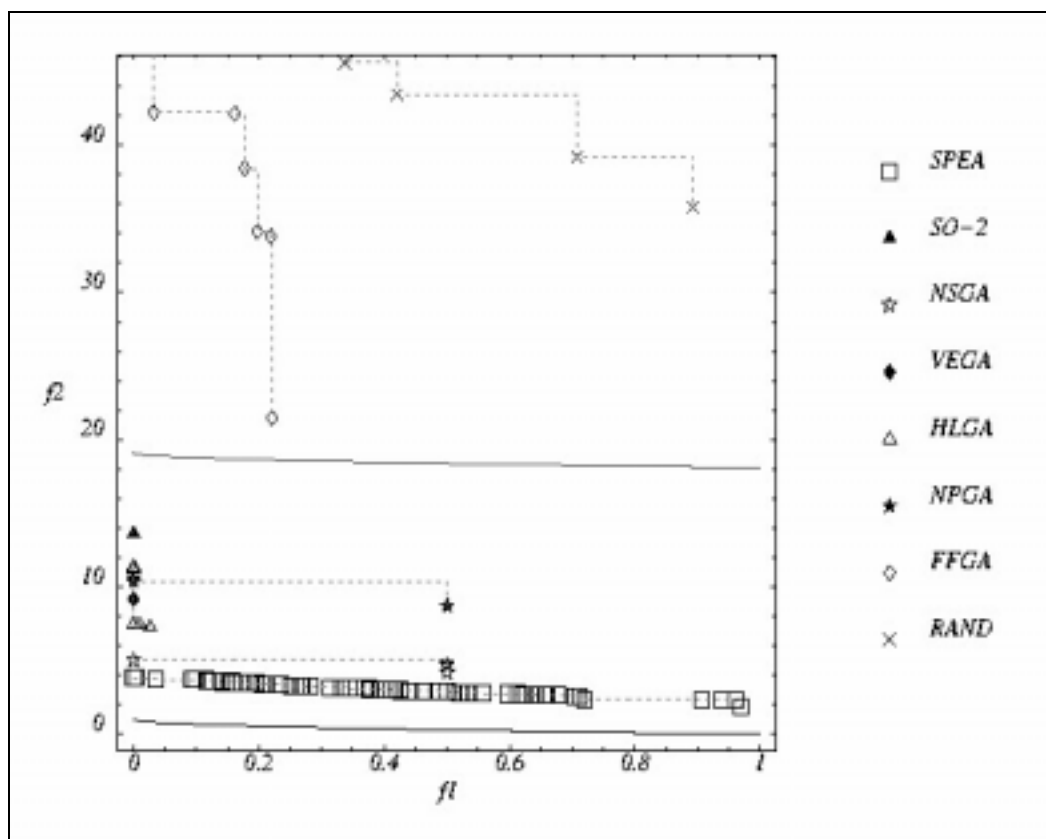


Figura 69 - Fronteiras de Pareto comparativas - τ_4 [72]

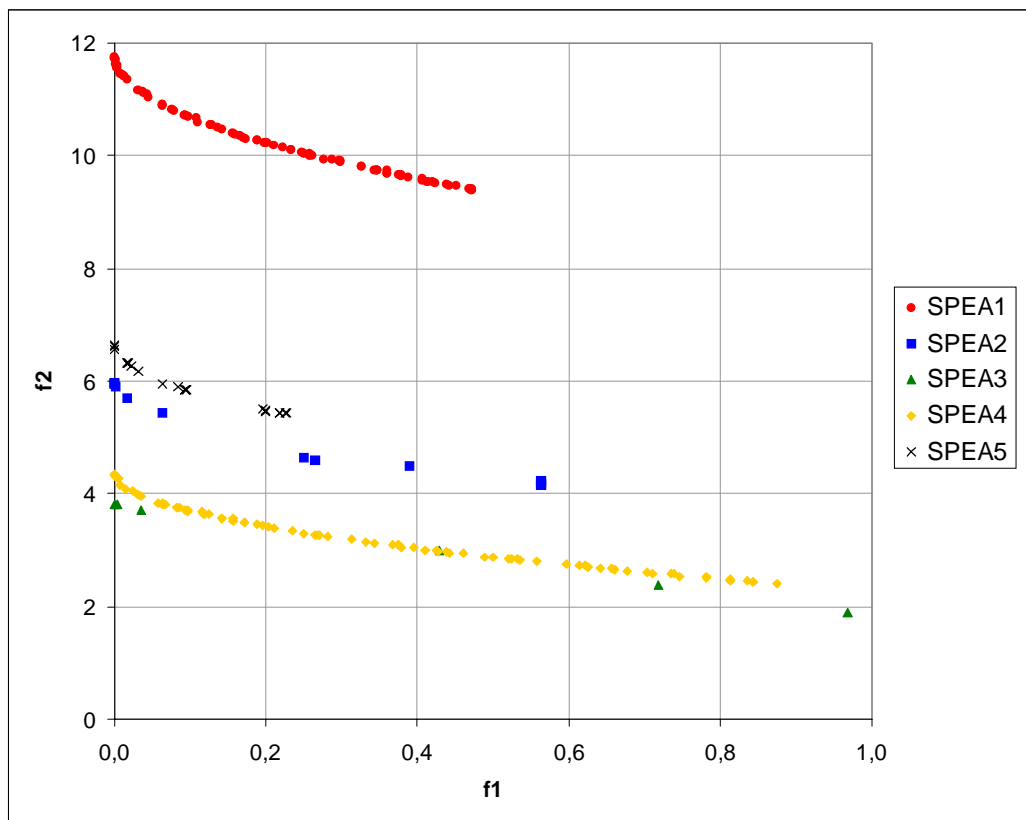


Figura 70 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_4 [73]

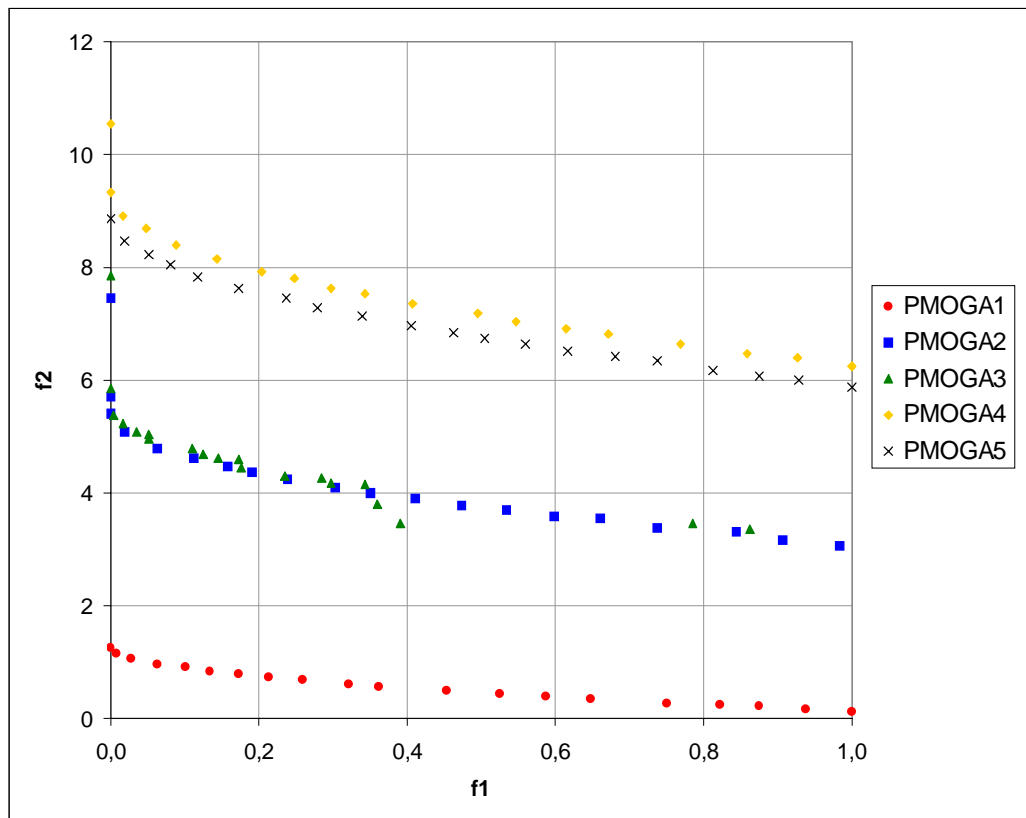


Figura 71 - Resultados de 5 execuções do algoritmo PMOGA - τ_4 [11]

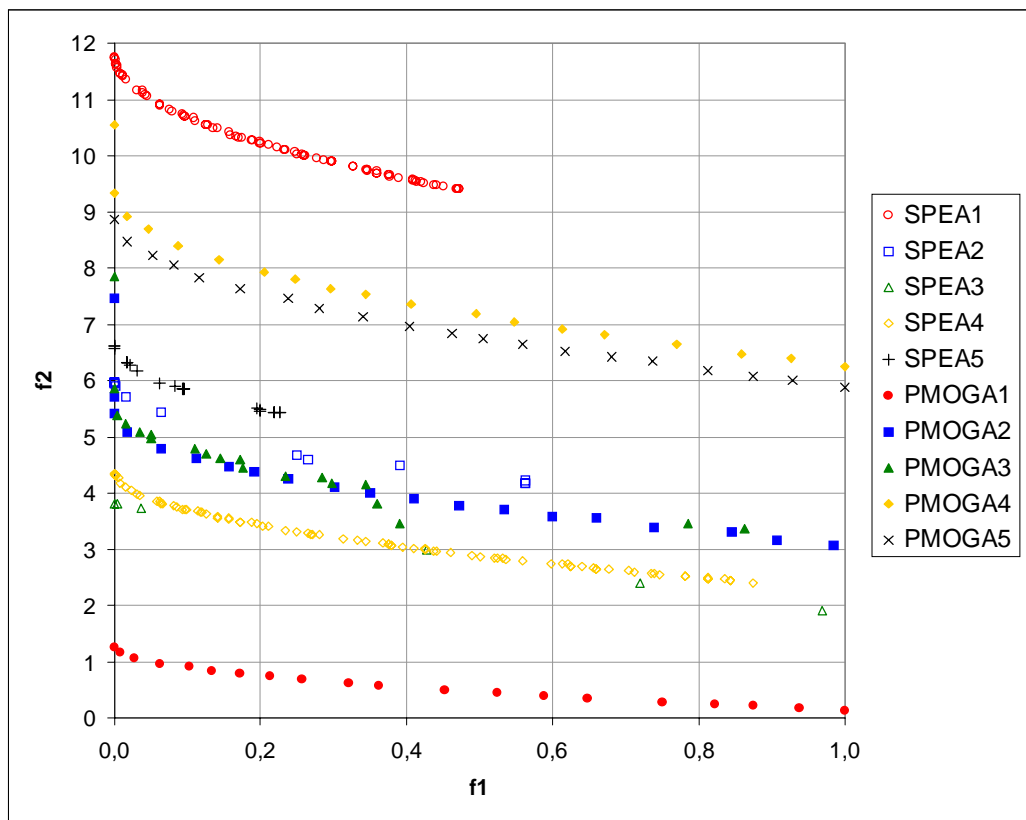


Figura 72 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_4 [11, 73]

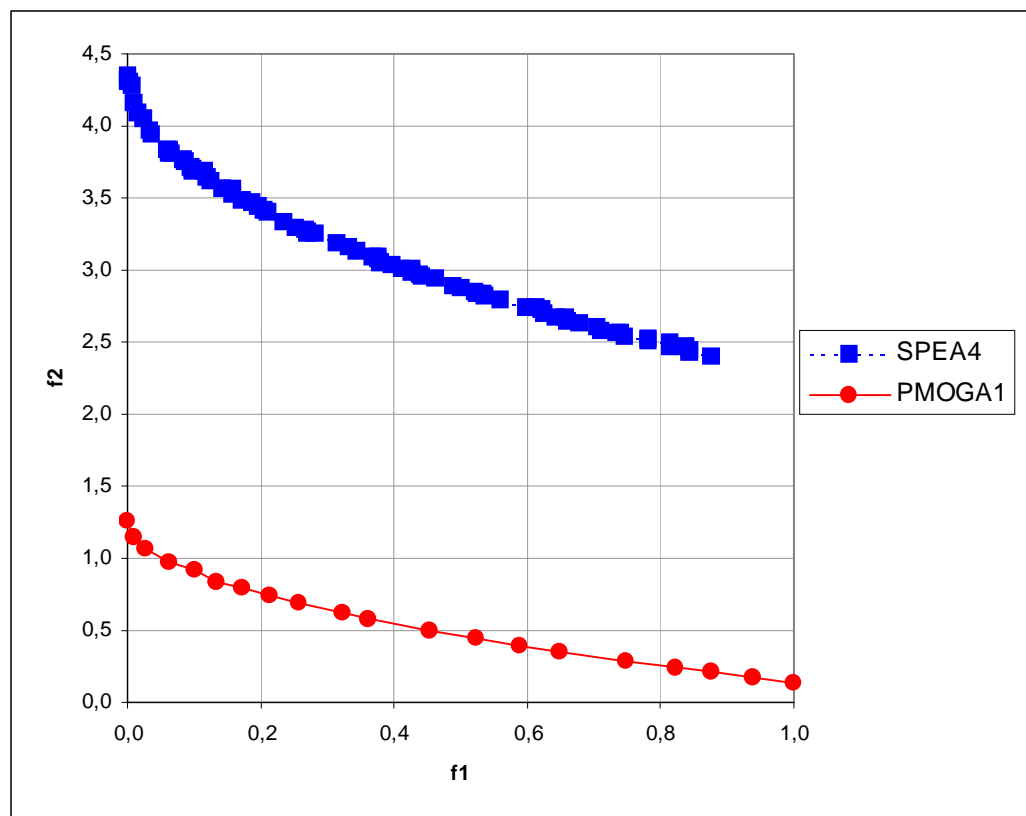


Figura 73 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_4 [11, 73]

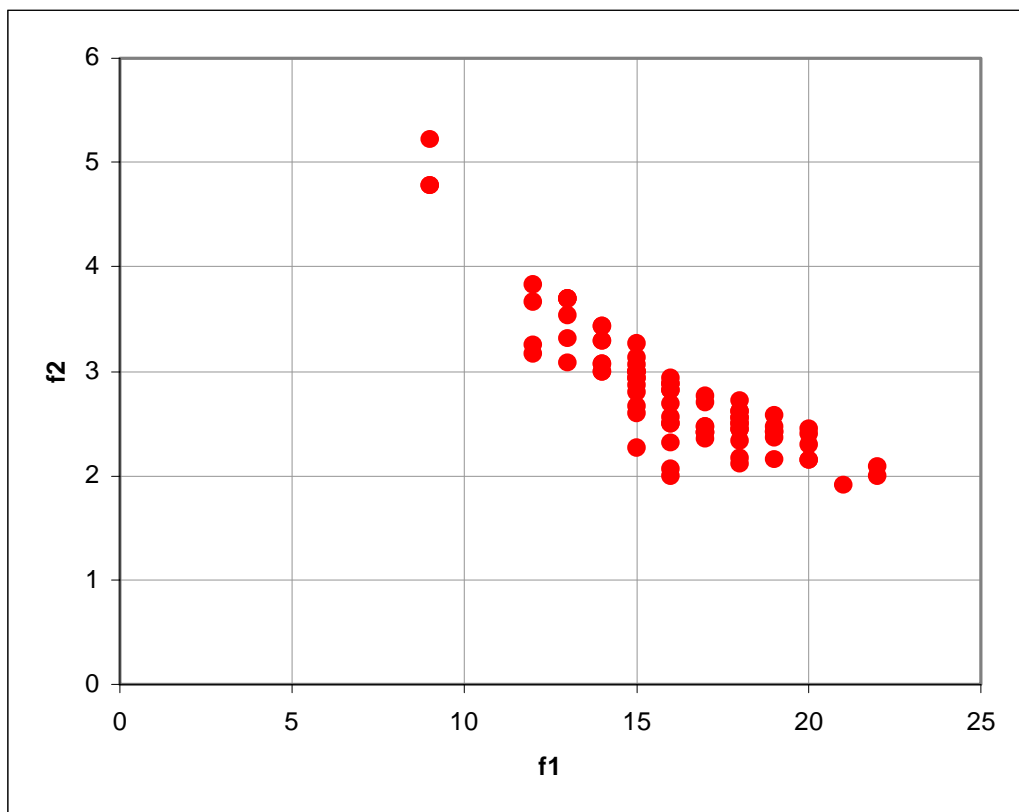


Figura 74 - Exemplo de população inicial - τ_5

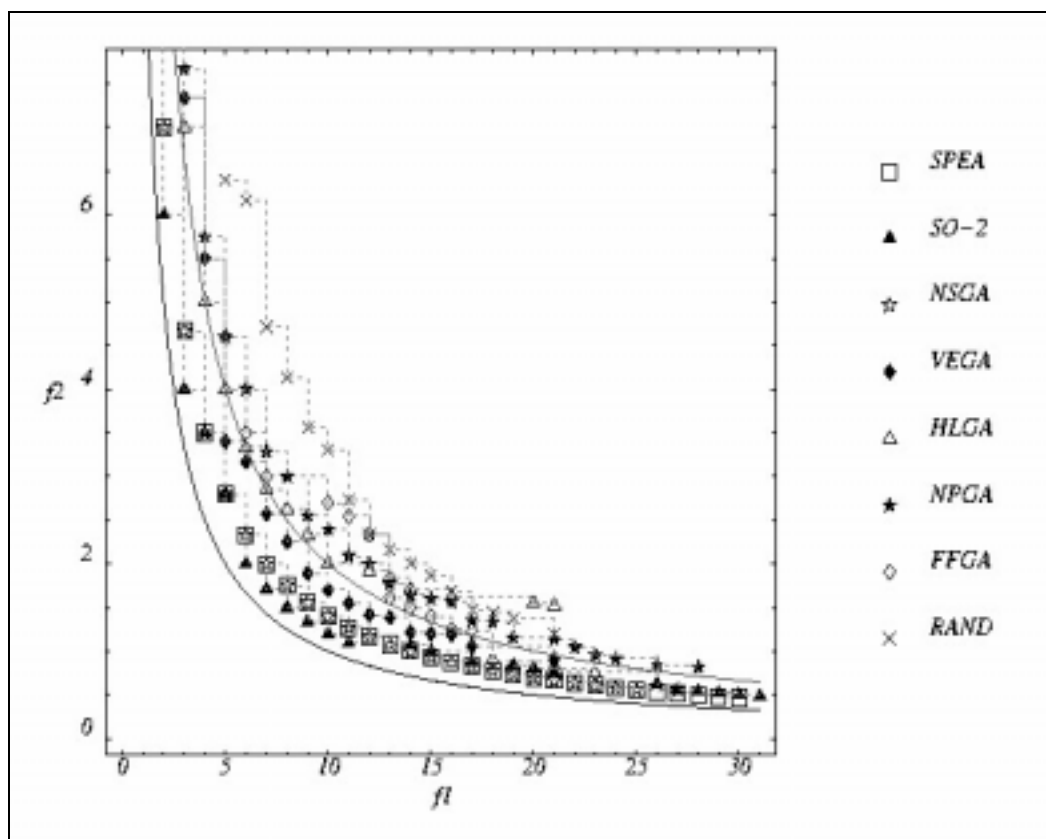


Figura 75 - Fronteiras de Pareto comparativas - τ_5 [72]

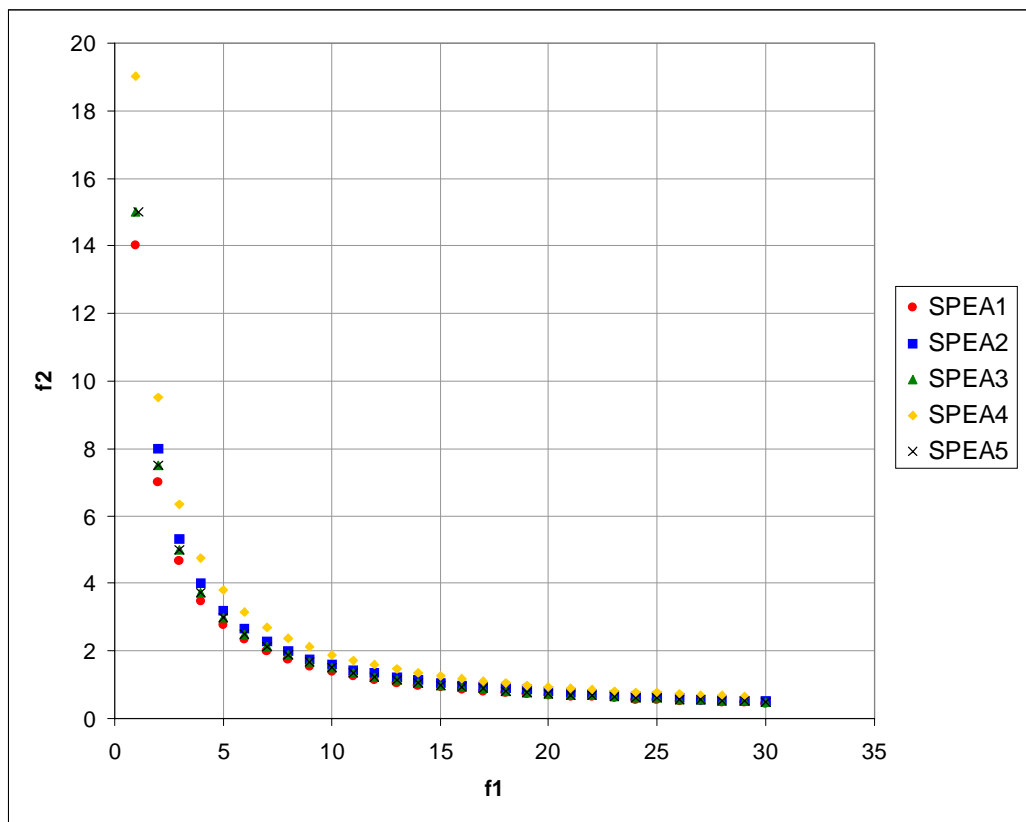


Figura 76 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_5 [73]

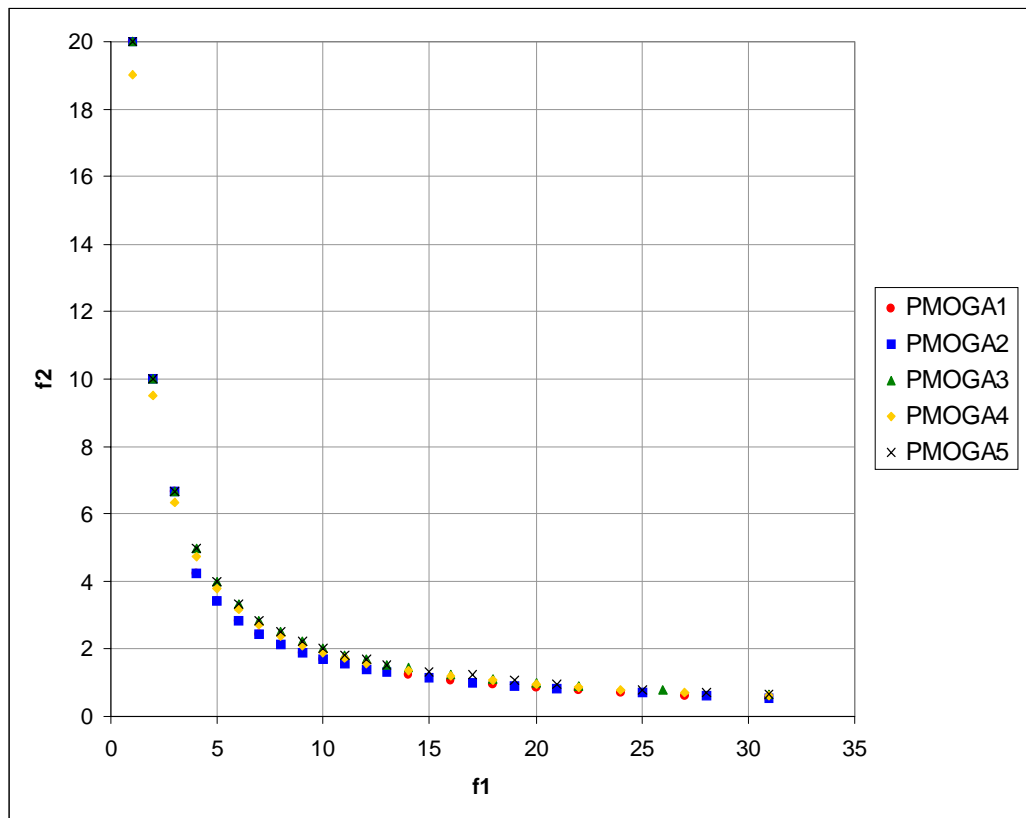


Figura 77 - Resultados de 5 execuções do algoritmo PMOGA - τ_5 [11]

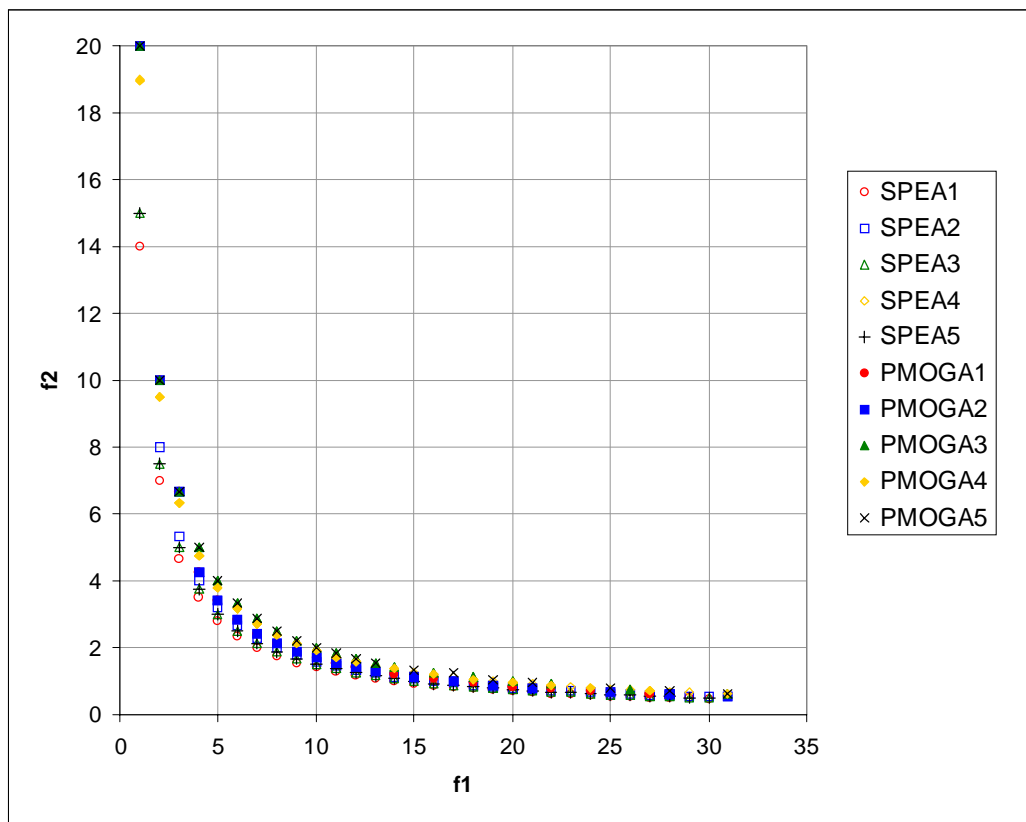


Figura 78 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_5 [11, 73]

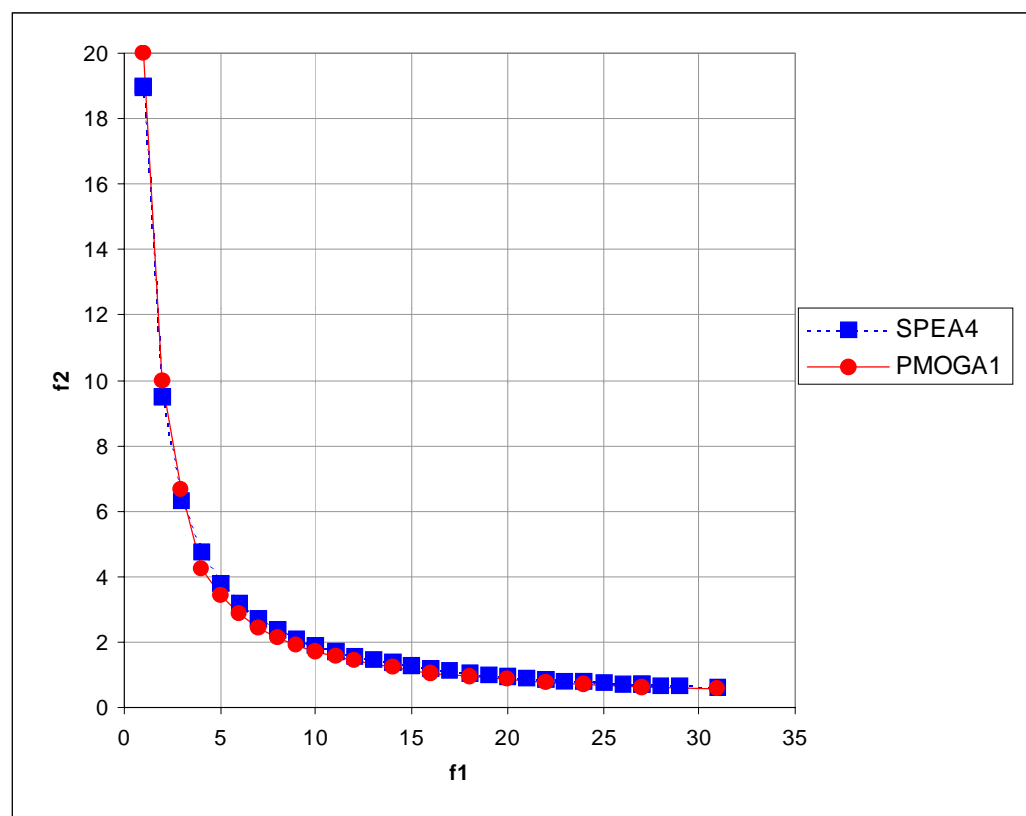


Figura 79 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_5 [11, 73]

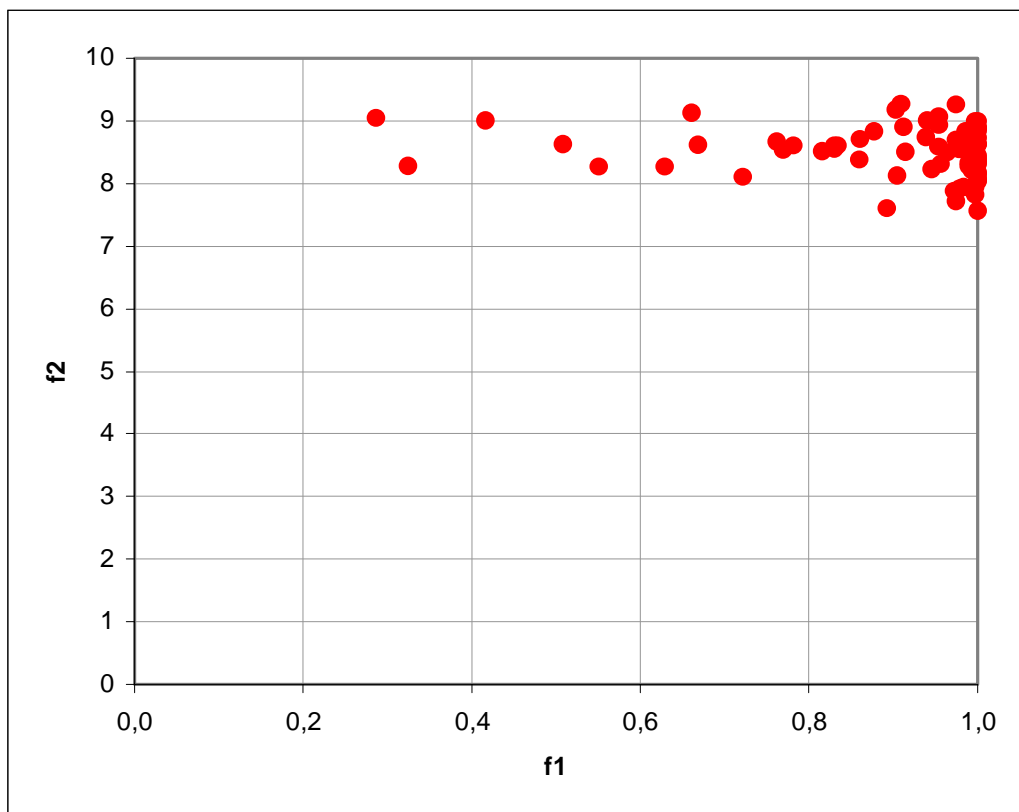


Figura 80 - Exemplo de população inicial - τ_6

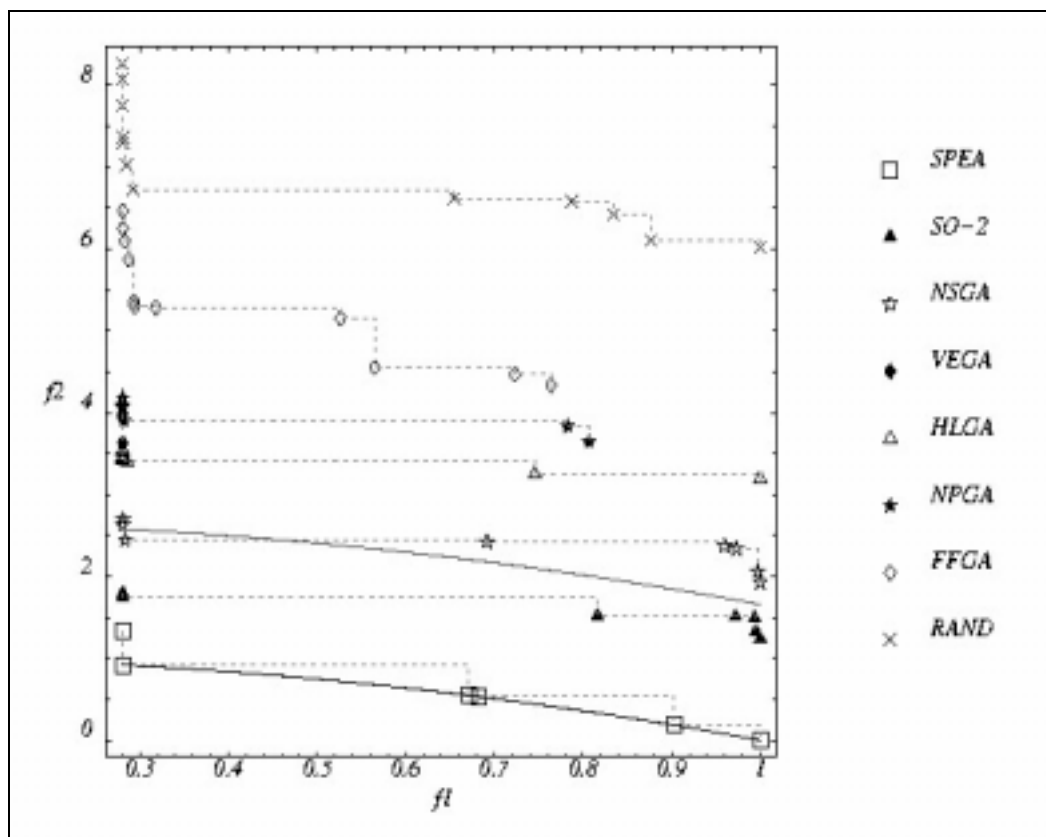


Figura 81 - Fronteiras de Pareto comparativas - τ_6 [72]

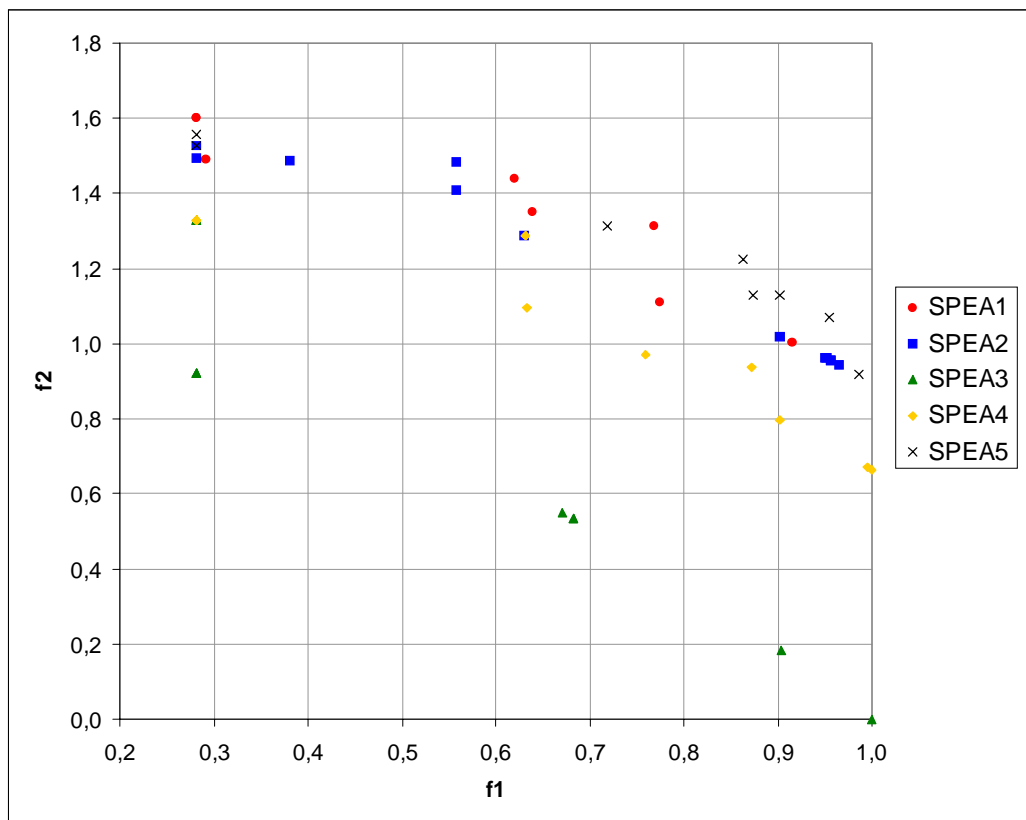


Figura 82 - Resultados das 5 primeiras execuções do algoritmo SPEA - τ_6 [73]

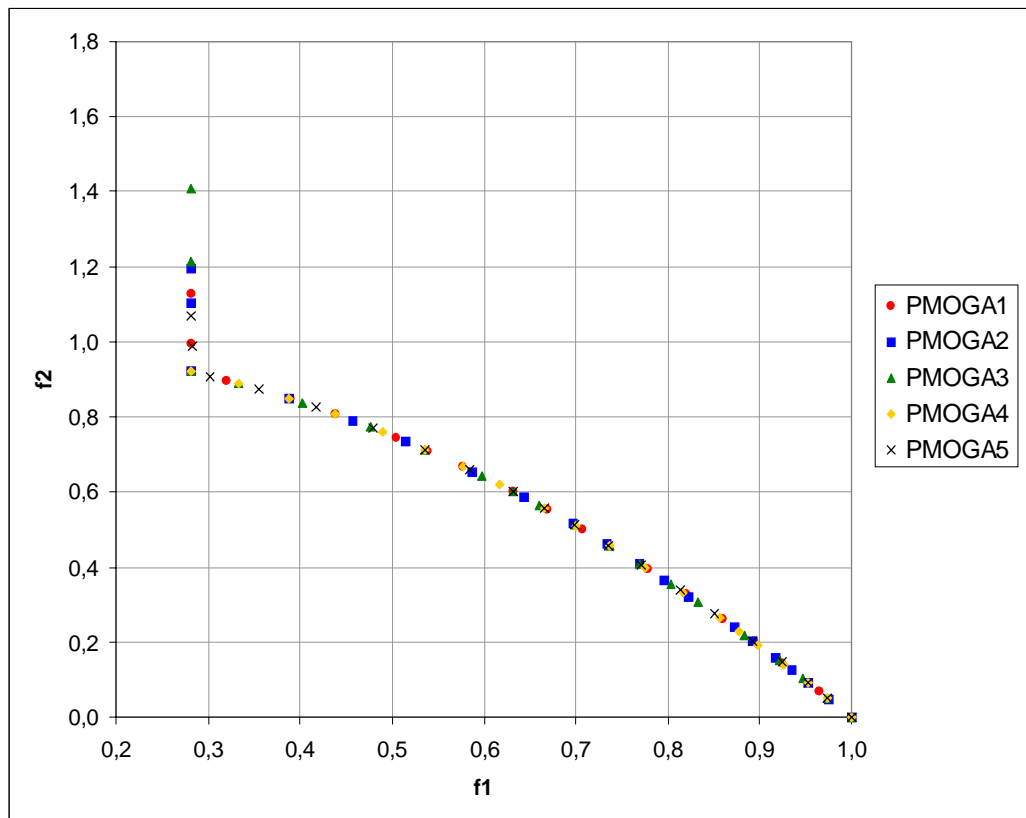


Figura 83 - Resultados de 5 execuções do algoritmo PMOGA - τ_6 [11]

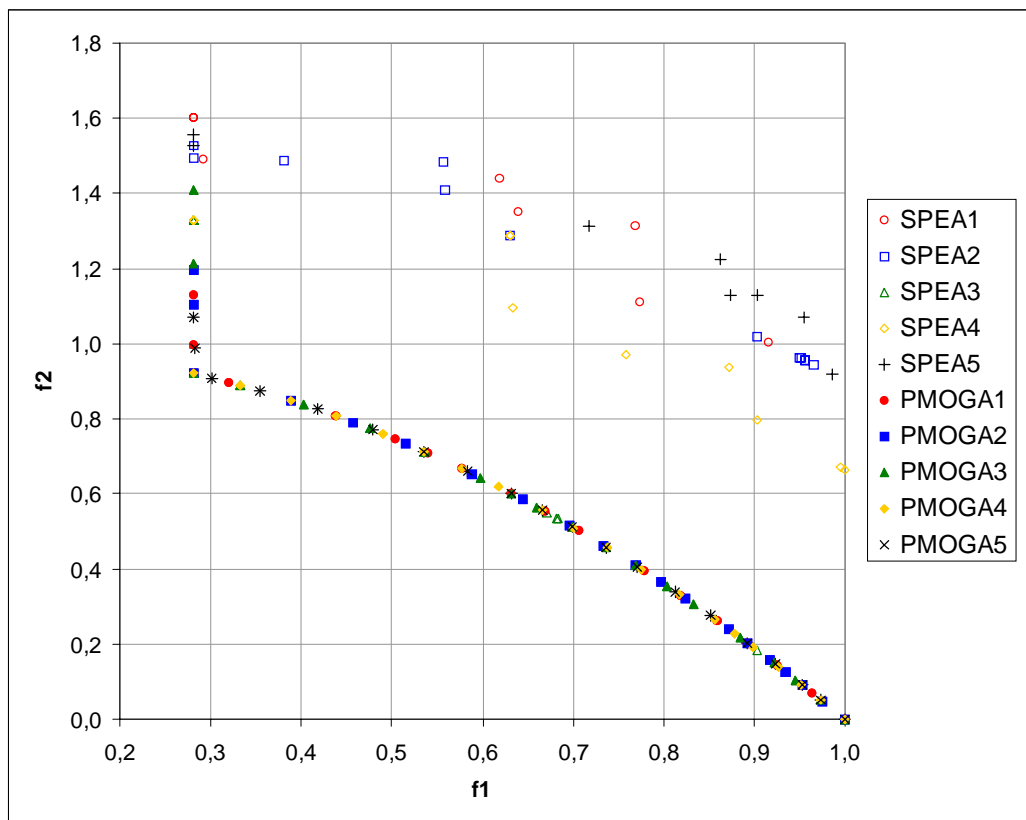


Figura 84 - Resultados de 5 execuções dos algoritmos SPEA e PMOGA - τ_6 [11, 73]

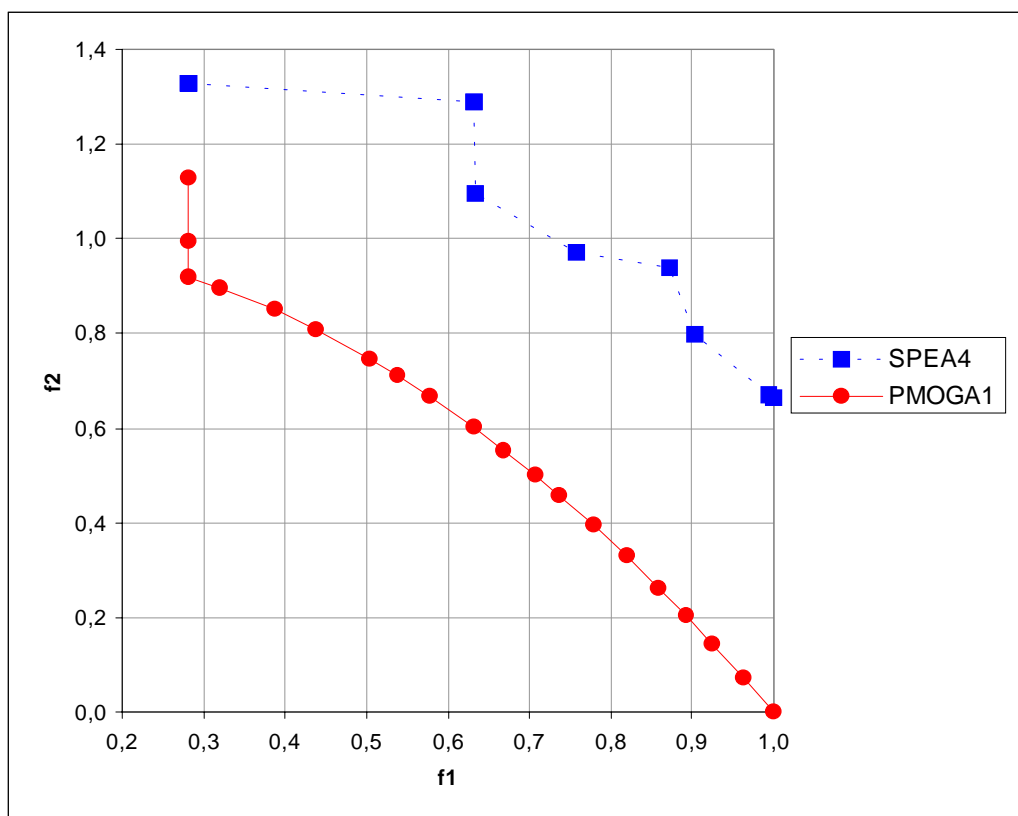


Figura 85 - Resultados das melhores execuções dos algoritmos SPEA e PMOGA - τ_6 [11, 73]

8.6. Conclusões

As funções-teste propostas em [71, 72] foram utilizadas em uma vasta gama de complexidades e características que tornam de difícil resolução o problema de otimização. Entretanto, os algoritmos apresentaram soluções satisfatórias, que ainda podem ser melhoradas com o ajuste dos parâmetros de configuração para cada problema e algoritmo em particular.

Os problemas além de complexos apresentam tamanhos consideráveis, uma vez que em alguns deles foram utilizados cromossomos com novecentas posições binárias. Este valor poderia ser em muito reduzido nestes problemas, contudo, sua utilização vem comprovar a aplicabilidade da técnica para problemas de grande porte.

Os resultados encontrados com o algoritmo proposto – PMOGA – foram na média superiores aos obtidos pelo algoritmo SPEA. Um destaque maior sobre esta observação é verificado no problema 6 onde nenhum dos oito algoritmos utilizados nas comparações das referências [71, 72], inclusive o SPEA, resolveram satisfatoriamente o problema tratado, enquanto que o algoritmo PMOGA obteve uma boa solução. Tal fato deve estar associado, principalmente, à utilização dos operadores multi-objetivos no lugar de uma estratégia de nicho, que modifica os valores originais das aptidões dos indivíduos.

Os mínimos individuais de cada objetivo foram melhor aproximados com o algoritmo PMOGA em função do operador de otimização individual, como pode ser observado no problema teste 5.

Um menor número de avaliações por geração foi utilizado pelo algoritmo proposto – 80 frente às 100 avaliações utilizadas pelos algoritmos citados. A dúvida fica por conta do SPEA que poderia, embora pouco provável, ter utilizado o mesmo número de 80 avaliações por geração.

Dentre as principais vantagens do algoritmo proposto frente aos algoritmos utilizados nas comparações realizadas em [71, 72] estão:

- Não utilização de técnicas de nicho ou compartilhamento, garantindo que os valores das funções objetivo são reais, isto é, não dependentes da proximidade ou não de outras soluções.
- Como uma consequência da vantagem anterior, espera-se uma eficiência maior, ou seja, menor esforço computacional já que alguns cálculos adicionais são dispensados.
- Não necessidade de se ajustar um parâmetro sensível à resolução do problema, que é o raio de vizinhança - σ_{share} – utilizado nas estratégias de nicho.
- Estrutura simples e intuitiva, contudo, capaz de apresentar soluções bem distribuídas pela fronteira ótima de Pareto.
- Comportamento regular nos mais variados tipos de problemas, mesmo naqueles com características que propiciariam o surgimento de enormes dificuldades, tais como: convexidade, não-convexidade, fronteiras discretas ou descontínuas, multimodalidade, problemas enganosos e espaços de busca tendenciosa.

9. Curva Tri-objetiva

9.1. Introdução e Modelo

O problema consiste de 3 funções objetivo de duas variáveis que devem ser simultaneamente minimizadas. A fronteira ótima de Pareto do problema é uma curva em 3 dimensões. Este problema é também apresentado, resolvido e discutido nas referências [66, 67].

As funções objetivo do problema são dadas pelas expressões:

$$f_1(x, y) = 0.5 * (x^2 + y^2) + \sin(x^2 + y^2)$$

$$f_2(x, y) = \frac{(3x - 2y + 4)^2}{8} + \frac{(x - y + 1)^2}{27} + 15$$

$$f_3(x, y) = \frac{1}{(x^2 + y^2 + 1)} - 1.1e^{(-x^2 - y^2)}$$

onde $-3 \leq x, y \leq 3$

Foi utilizado um cromossomo binário com 20 posições de comprimento.

9.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).

O algoritmo foi executado três vezes com diferentes configurações, as quais são descritas na tabela que se segue:

Configuração	1	2	3
Tamanho população	80	100	150
Tamanho do filtro	50	80	100
Número de gerações*	50 + 10	90 + 10	100 + 50
Número de sol. Torneio	5	5	5
Prob. de recombinação	90%	90%	90%
Prob. de mutação	5%	5%	1%
Total de avaliações **	3967	5914	7973

* número de gerações padrões + adicionais (filtro)

** para cada função objetivo

Tabela 8 - Configurações utilizadas do AG

9.3. Resultados

Nas próximas figuras são apresentadas as fronteiras encontradas com o AG proposto para as 3 configurações empregadas, além da fronteira ótima de Pareto apresentada na referência [66], como resultado de uma otimização completa por todo o espaço de busca do problema.

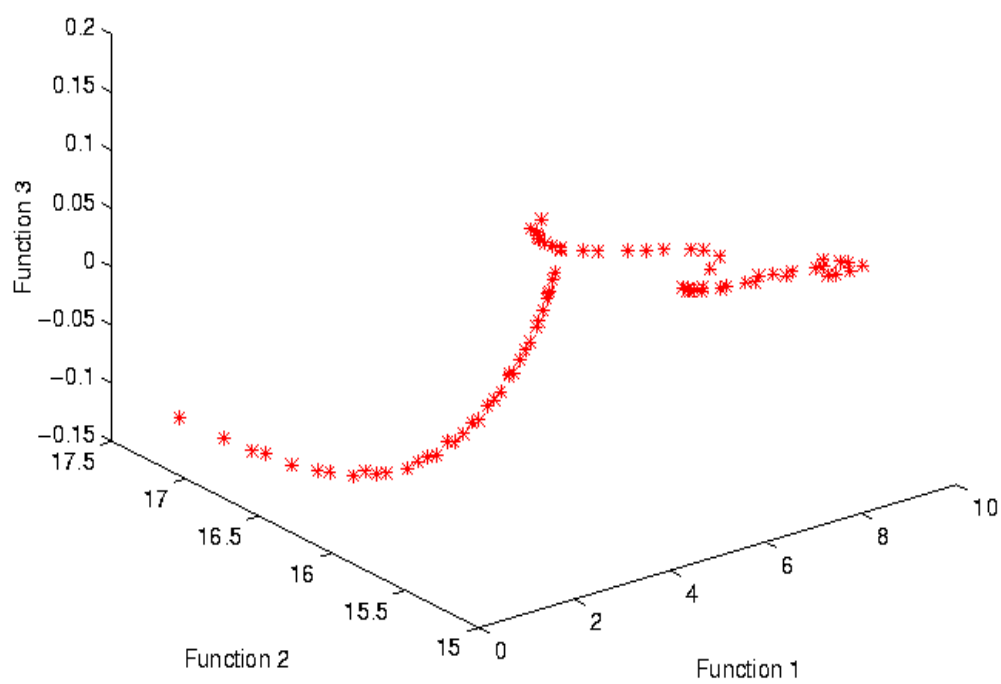


Figura 86 - Fronteira ótima de Pareto completa [66]

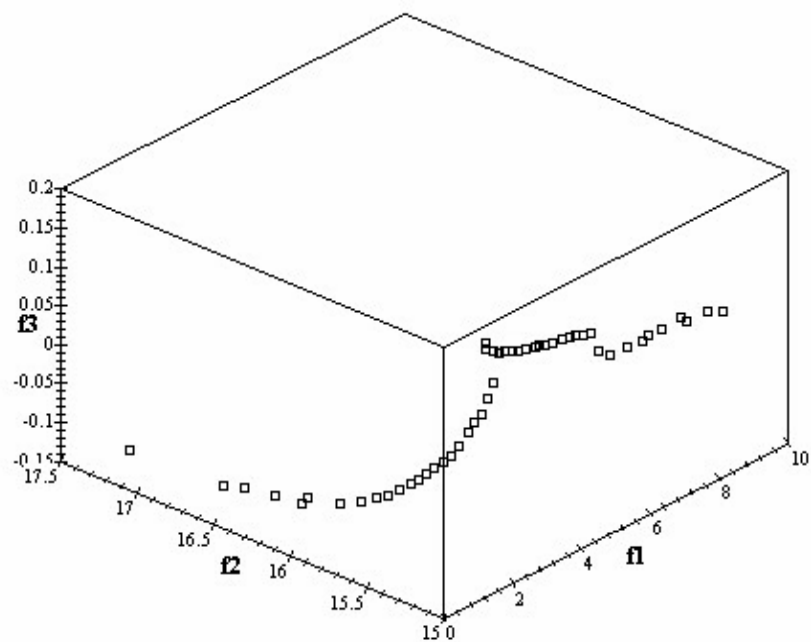


Figura 87 - Configuração 1 - 50 soluções no filtro

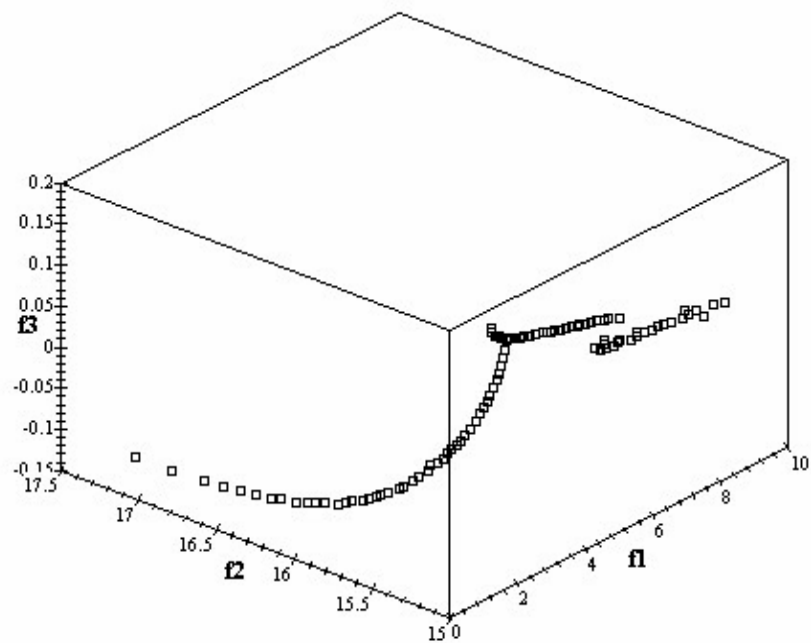


Figura 88 - Configuração 2 - 80 soluções no filtro

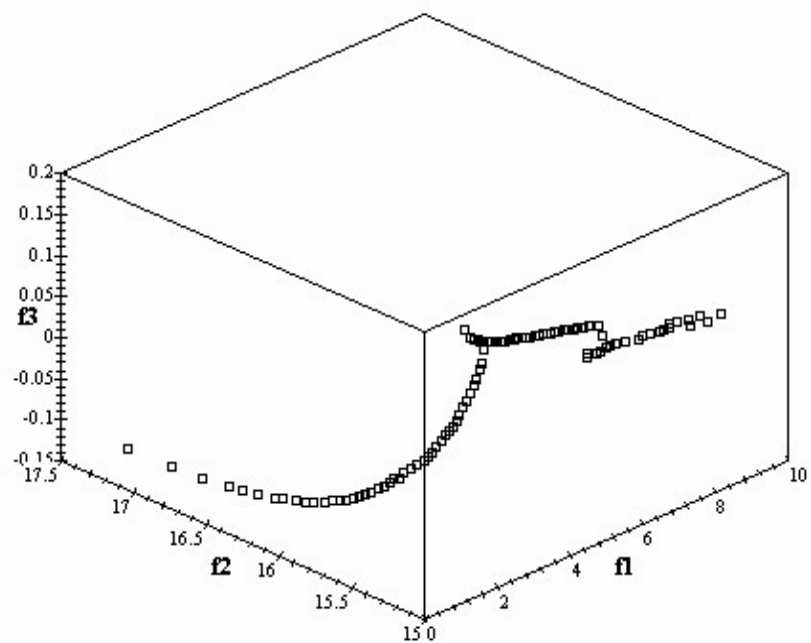


Figura 89 - Configuração 3 - 100 soluções no filtro

9.4. Conclusões

A fronteira de Pareto ficou bem identificada em todas as análises realizadas, comprovando a robustez do algoritmo e sua escalabilidade para um maior número de funções objetivo.

Comparando-se os resultados encontrados com a fronteira de Pareto original do problema pode-se concluir que o algoritmo genético desenvolvido possibilita a obtenção de boas aproximações da solução do problema com um esforço computacional bem inferior, já que uma busca completa por todo o espaço de definição do problema é uma tarefa bastante onerosa, sendo para uma vasta gama de aplicações uma impossibilidade computacional.

O número de avaliações foi informado para cada análise com o intuito de facilitar a realização de futuras comparações, pois esta é, sem dúvida, a melhor maneira de comparar o desempenho entre algoritmos evolucionários.

10. Esfera

10.1. Introdução e Modelo

Este é um problema muito interessante ilustrado nas referências [66, 67] em conjunto com outras aplicações elaboradas para otimização multi-objetivo. Os principais destaques observados neste problema podem ser enumerados nas seguintes características:

- i) presença de 3 objetivos;
- ii) um modo simples de quantificar o afastamento da solução encontrada da fronteira ótima de Pareto (definida pela superfície da esfera de raio unitário no quadrante positivo);
- iii) possibilidade de uma fácil visualização da distribuição (uniformidade) das soluções pela fronteira encontrada;
- iv) escalabilidade para um maior número de funções objetivo.

O problema é formulado segundo a maximização de três objetivos associados à três variáveis, tendo como restrição superior a superfície de uma esfera de raio unitário:

Maximize	$F = (f_1, f_2, f_3)$
	$f_1 = x$
	$f_2 = y$
	$f_3 = z$
Sujeita a	$x^2 + y^2 + z^2 \leq 1$
Onde	$x, y, z \in [0, 1]$

As variáveis foram codificadas num cromossomo de 30 posições de comprimento com 3 casas de precisão para cada uma delas.

10.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).

O algoritmo foi executado com duas configurações diferentes segundo a tabela que se segue:

Configuração	100 x 200 x 200	300 x 500 x 200
Tamanho população	200	500
Tamanho do filtro	100	300
Número de gerações	200	200
Número de sol. Torneio	10	10
Prob. de recombinação	90%	90%
Prob. de mutação	1%	1%

Tabela 9 - Configurações utilizadas do AG

10.3. Resultados

Nas próximas figuras são apresentadas graficamente os resultados encontrados com o algoritmo desenvolvido para as configurações empregadas.

Nas figuras abaixo são apresentadas três vistas (superior, lateral e frontal) da população inicial utilizada para a primeira configuração do algoritmo – 100 x 200 x 200. Assim, pode-se visualizar inclusive as soluções infactíveis da população – situadas acima da superfície descrita por uma esfera de raio unitário.

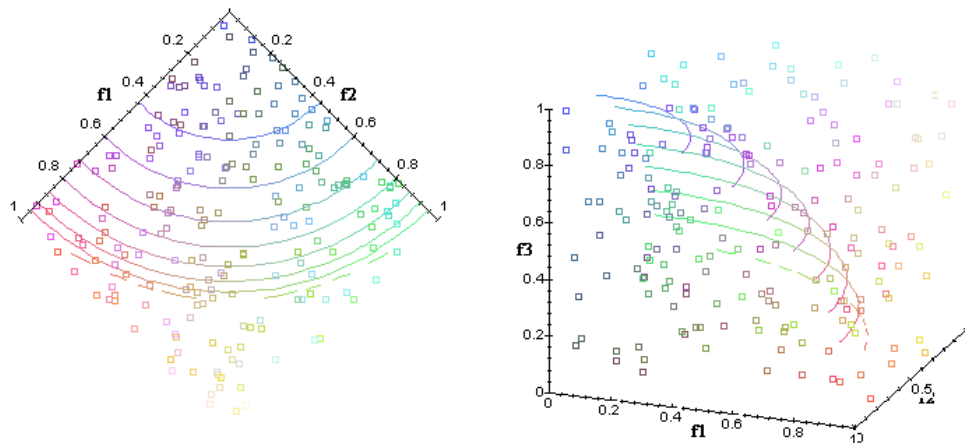


Figura 90 - População inicial – vistas superior e lateral

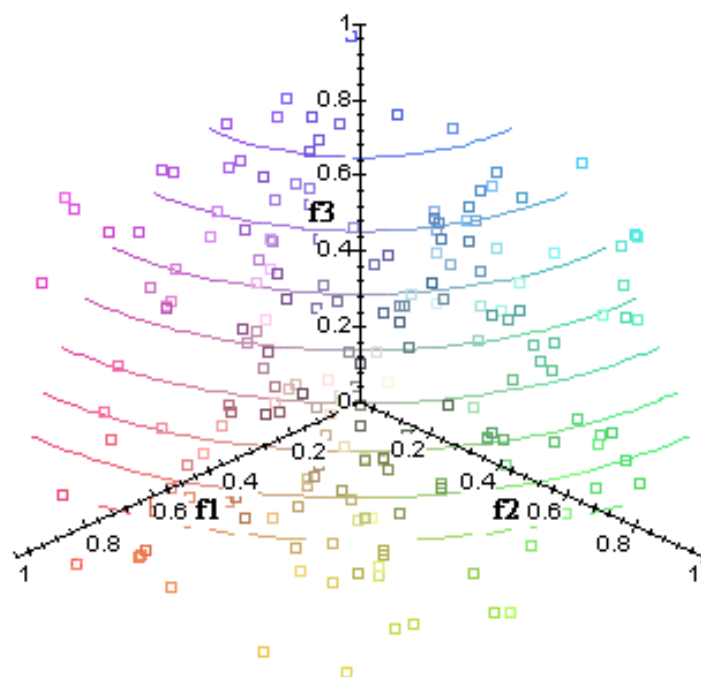


Figura 91 - População inicial – vista frontal

Nas figuras abaixo são apresentadas as fronteiras encontradas para a primeira configuração do algoritmo – 100 x 200 x 200. Observe que não existem soluções infactíveis no conjunto e que elas estão uniformemente distribuídas pela fronteira.

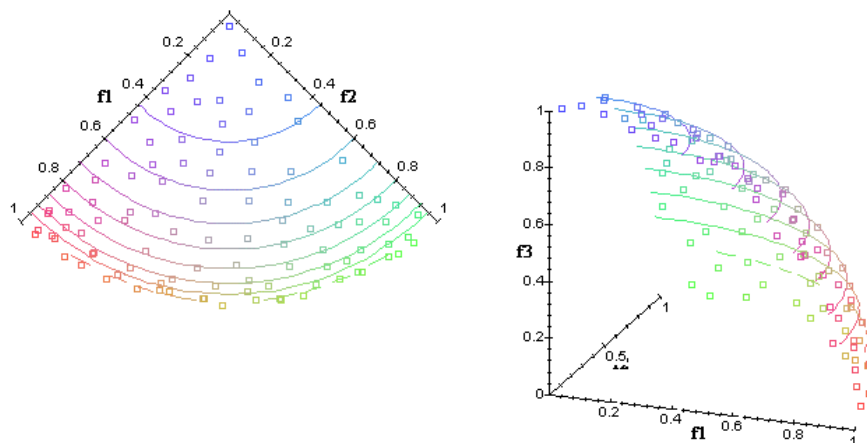


Figura 92 - Fronteira 100 x 200 x 200 – vistas superior e lateral

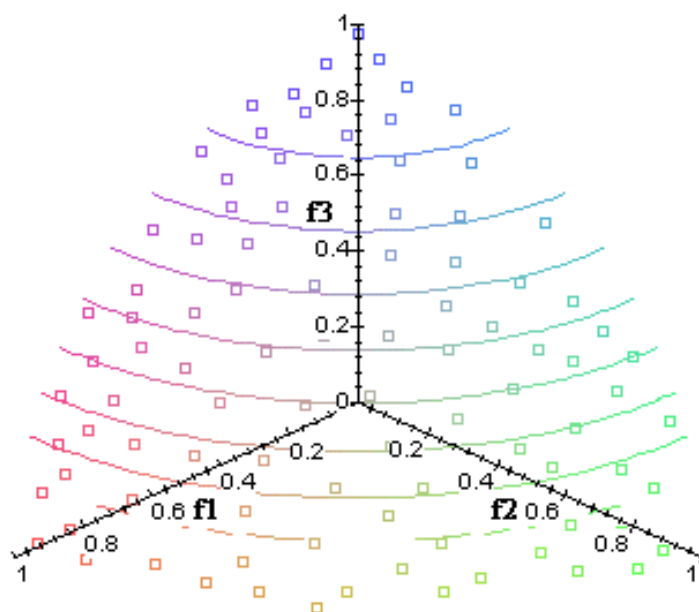


Figura 93 - Fronteira 100 x 200 x 200 – vista frontal

Nas figuras abaixo são apresentadas as fronteiras encontradas para a primeira configuração do algoritmo – 100 x 200 x 200, sem o emprego dos operadores multi-objetivos de adição e de otimização individual. Entretanto, o operador de exclusão é obrigatoriamente empregado e garante uma boa solução para o problema.

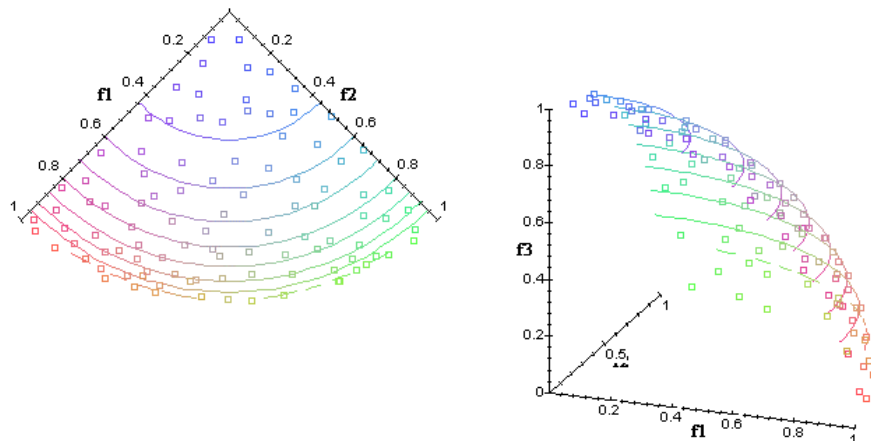


Figura 94 - Fronteira 100 x 200 x 200, sem operadores multi-objetivos – vistas superior e lateral

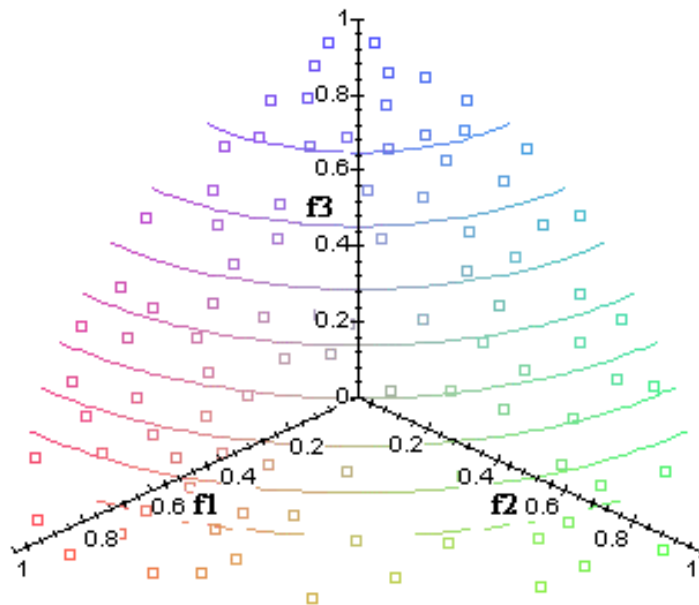


Figura 95 - Fronteira 100 x 200 x 200, sem operadores multi-objetivos – vista frontal

Nas figuras abaixo são apresentadas as fronteiras encontradas para a segunda configuração do algoritmo – 300 x 500 x 200. Obtêm-se uma maior cobertura da fronteira pelo fato de se adotar um filtro maior.

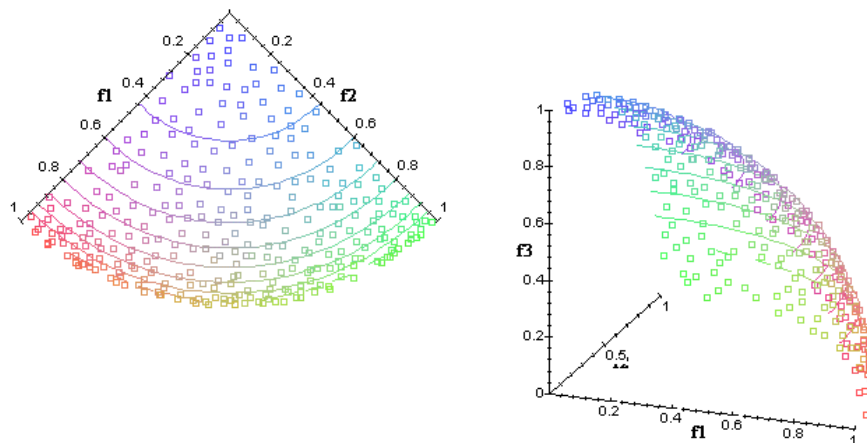


Figura 96 - Fronteira 300 x 500 x 200 – vistas superior e lateral

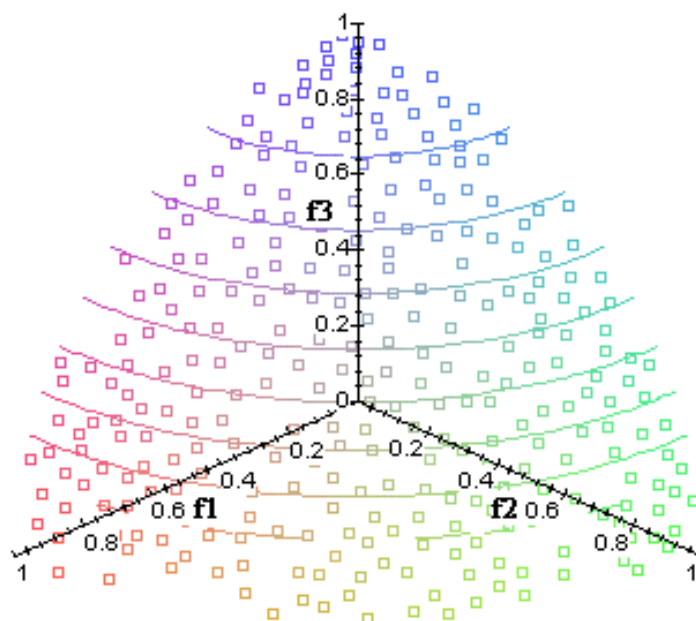


Figura 97 - Fronteira 300 x 500 x 200 – vista frontal

10.4. Conclusões

A fronteira de Pareto ficou bem definida em todas as análises realizadas e com suas soluções uniformemente distribuídas, conforme pode ser visualmente observado, principalmente, pelas vistas frontais das fronteiras apresentadas.

Na Tabela seguinte são apresentados os resultados numéricos das execuções realizadas, destacando-se a medida de afastamento das soluções presentes no filtro final encontrado com relação à fronteira de Pareto do problema (superfície da esfera). Deste modo, pode-se através do afastamento médio, que leva em consideração o número de soluções adotadas no filtro, estabelecer-se uma quantificação deste afastamento que poderá servir como medida de comparação entre os resultados aqui apresentados, bem como os futuros resultados encontrados por outros algoritmos.

Configuração*	100 x 200 x 200			300 x 500 x 200
	Pop. Inicial	Normal	Sem Oper. MO**	Normal
Núm. de Avaliações	200	26339	20359	47446
Núm. de Soluções	200	100	100	300
Afast. Total***	87,588	2,440	2,869	4,785
Afast. Médio****	0,438	0,024	0,029	0,016
Afast. Médio %	43,80%	2,40%	2,90%	1,60%

* tamanho do filtro x tamanho da população x número de gerações

** com emprego apenas do operador de exclusão

$$*** \text{ Afastamento Total} = \sum_{i=1}^{NSoluções} [1 - (x_i^2 + y_i^2 + z_i^2)]$$

**** Afastamento Médio = Afastamento Total / Número de Soluções

Tabela 10 - Resultados numéricos encontrados – avaliações e afastamento

Como era de se esperar, observa-se que os operadores multi-objetivos melhoram além da distribuição das soluções pela fronteira, também o afastamento à fronteira de Pareto do problema, como uma consequência do maior número de avaliações, o que caracteriza o processo evolutivo implementado.

11. Cubo – 8 Funções Objetivo

11.1. Introdução e Modelo

Este problema inédito tem como principal objetivo avaliar a escalabilidade dos otimizadores multi-objetivos com relação ao aumento do número de funções. Por isso, ele é caracterizado pela minimização de oito funções objetivo que avaliam a distância das soluções, pontos no espaço tridimensional, em relação aos vértices de um cubo de lado meio, inserido dentro de um outro cubo de lado unitário que define o espaço de busca do problema, conforme os modelos seguintes:

$$\begin{array}{ll}\text{Minimize} & f_i = |x - x_i| + |y - y_i| + |z - z_i| \\ & i = 1, 2, \dots, 8 \text{ (vértices)} \\ \text{Onde} & x, y, z \in [0, 1]\end{array}$$

Coordenadas dos vértices:

$$\begin{array}{l}\text{vértice 1} \rightarrow (0.25, 0.25, 0.25) \\ \text{vértice 2} \rightarrow (0.75, 0.25, 0.25) \\ \text{vértice 3} \rightarrow (0.25, 0.75, 0.25) \\ \text{vértice 4} \rightarrow (0.25, 0.25, 0.75) \\ \text{vértice 5} \rightarrow (0.75, 0.75, 0.25) \\ \text{vértice 6} \rightarrow (0.75, 0.25, 0.75) \\ \text{vértice 7} \rightarrow (0.25, 0.75, 0.75) \\ \text{vértice 8} \rightarrow (0.75, 0.75, 0.75)\end{array}$$

As variáveis foram codificadas num cromossomo de 30 posições de comprimento com 3 casas de precisão para cada uma delas.

A visualização espacial do problema pode ser vista pela Figura seguinte:

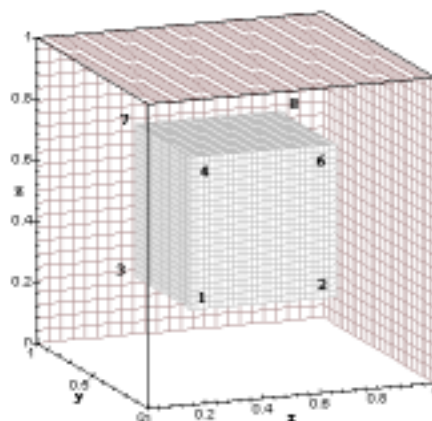


Figura 98 - Modelo espacial – problema do cubo

11.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).

Foram empregadas três configurações distintas segundo a tabela abaixo:

Configuração	1	2	3
Tamanho população	20	100	200
Tamanho do filtro	8	50	100
Número de gerações	20+880	500+500	200+800
Número de sol. Torneio	10	10	10
Prob. de recombinação	90%	90%	90%
Prob. de mutação	2%	2%	2%
Número de Avaliações	24260	73102	60220

Tabela 11 - Configurações utilizadas do AG

11.3. Resultados

Nas Figuras seguintes são apresentadas as populações iniciais utilizadas pela primeira configuração do algoritmo – 8 x 20 x 20+880.

A finalidade desta configuração é avaliar a capacidade do algoritmo em encontrar os ótimos individuais de cada função objetivo, o que possibilita ao tomador de decisões uma primeira impressão do problema a ser resolvido e o conhecimento preliminar do espaço viável e de busca do mesmo.

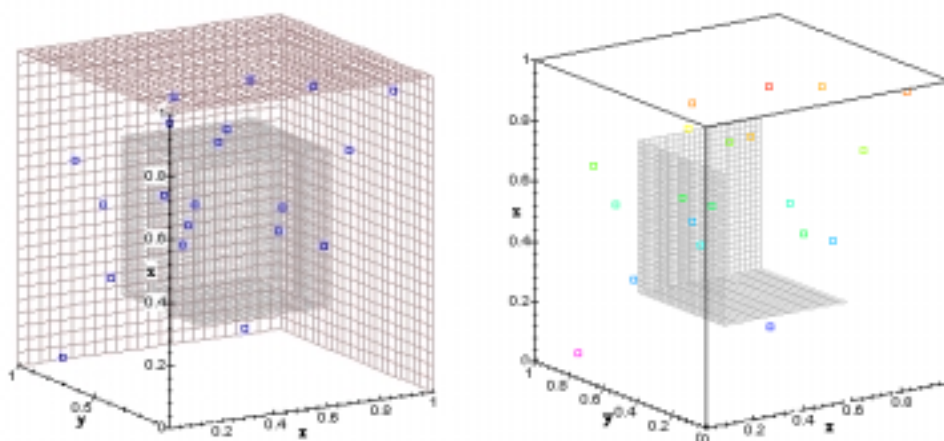


Figura 99 - População inicial (1) – vistas espaciais

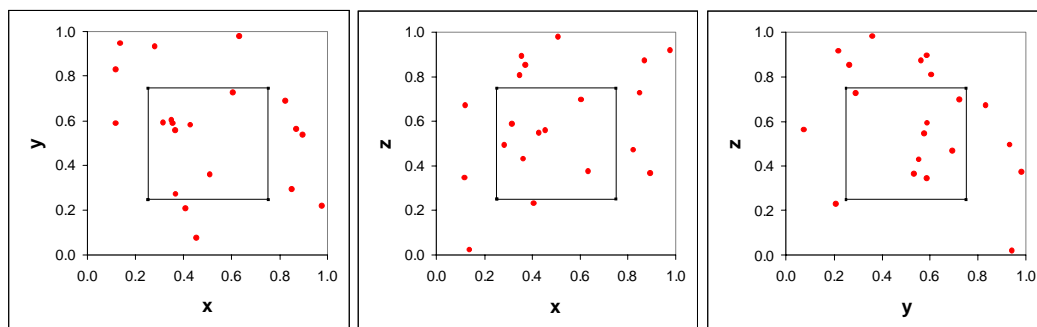


Figura 100 - População inicial (1) – projeções planas

Nas próximas Figuras são apresentadas as soluções da primeira configuração do algoritmo, que são os ótimos individuais de cada função objetivo do problema.

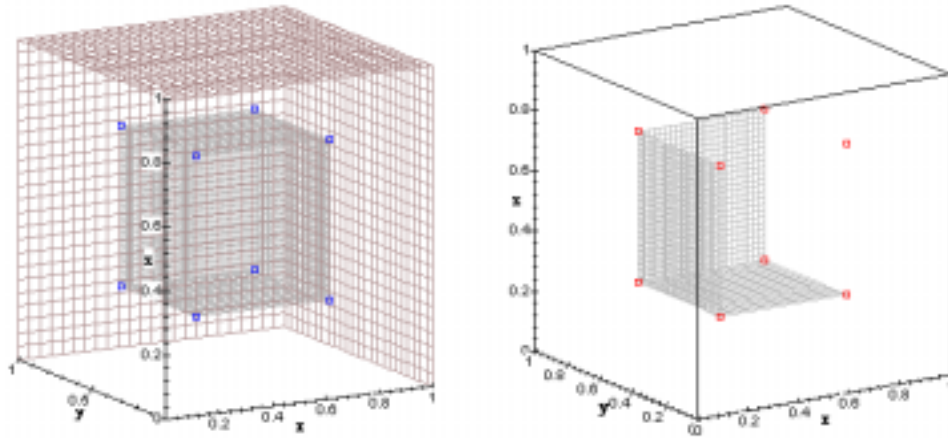


Figura 101 - Soluções encontradas (1) – vistas espaciais

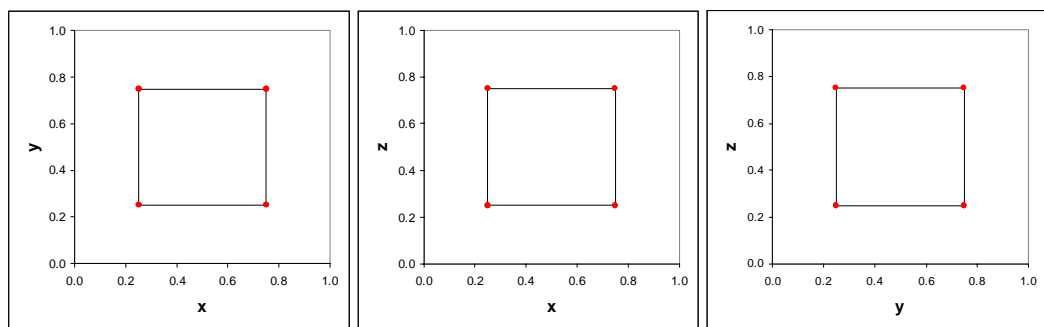


Figura 102 - Soluções encontradas (1) – projeções planas

Quando as funções objetivo são combinadas duas a duas ocorrem três disposições: retangular (inclinada) crescente, retangular decrescente e reta decrescente, conforme a Figura que se segue:

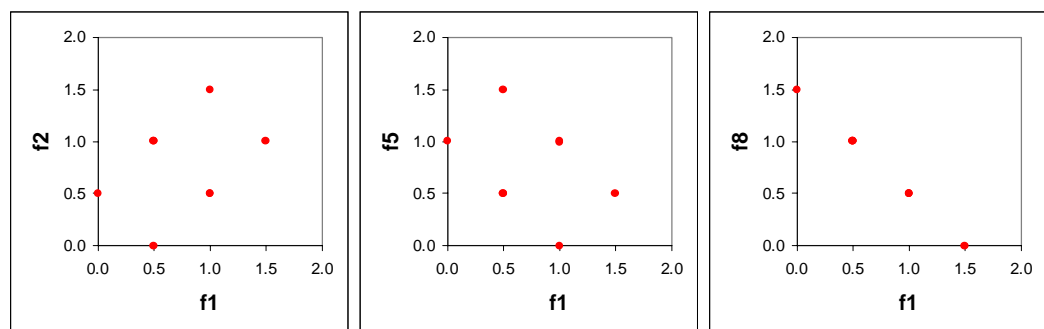


Figura 103 - Disposições possíveis dos objetivos combinados 2 a 2

Estas disposições acontecem segundo as regras definidas abaixo:

- Retangular crescente → Quando os vértices relativos às funções objetivo estão separados por apenas uma aresta do cubo – são vértices extremos de uma mesma aresta.
- Retangular decrescente → Quando os vértices relativos às funções objetivo estão separados por duas arestas do cubo – são vértices diagonais de uma mesma face.
- Reta decrescente → Quando os vértices relativos às funções objetivo estão separados por três arestas do cubo – são vértices diagonais dele.

Nas Figuras seguintes são apresentadas as populações iniciais utilizadas pela segunda configuração do algoritmo – 50 x 100 x 500+500.

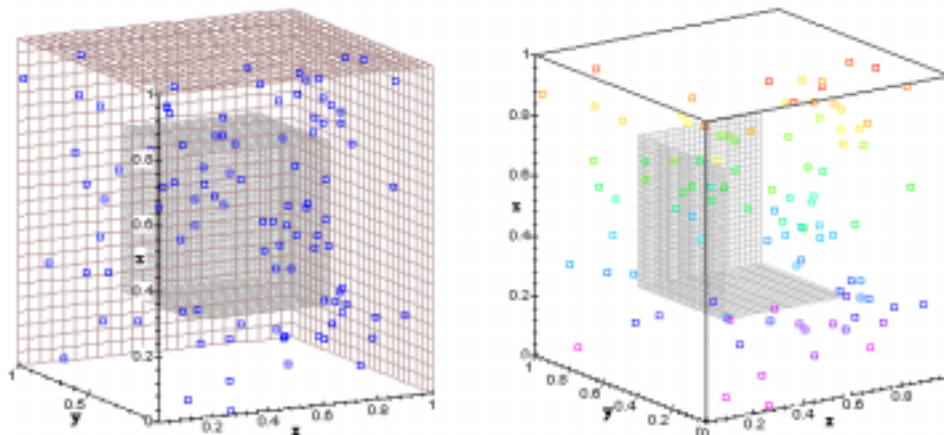


Figura 104 - População inicial (2) – vistas espaciais

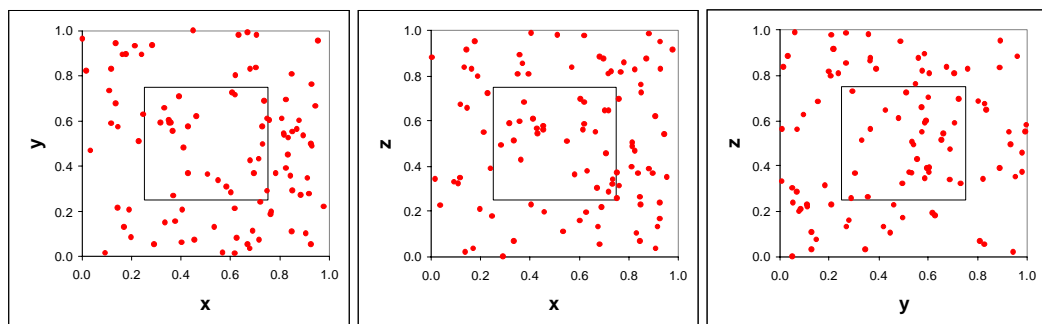


Figura 105 - População inicial (2) – projeções planas

Nas Figuras abaixo são mostradas as soluções encontradas pela segunda configuração do algoritmo.

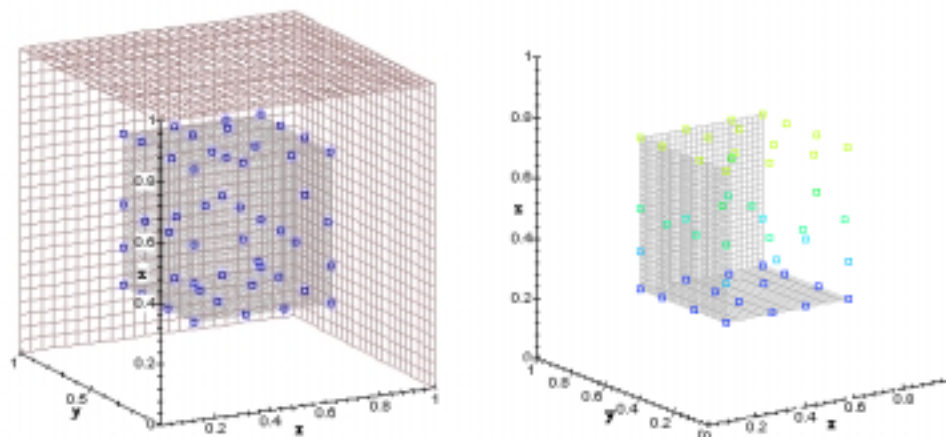


Figura 106 - Soluções encontradas (2) – vistas espaciais

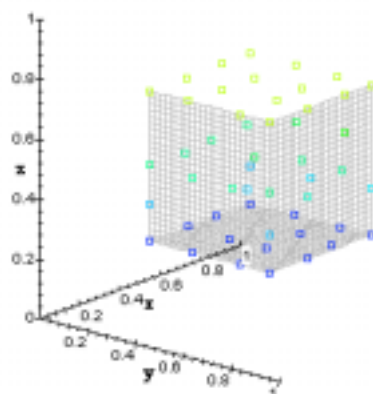


Figura 107 - Soluções encontradas (2) – vista espacial inclinada

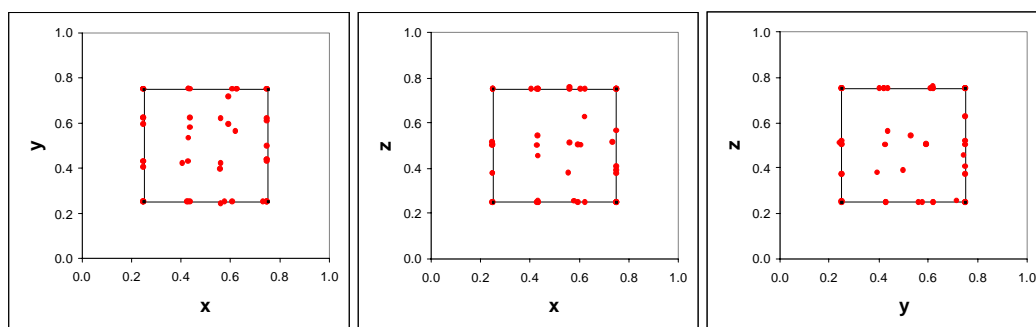


Figura 108 - Soluções encontradas (2) – projeções planas

No conjunto de vinte e oito Figuras seguintes são apresentados os gráficos das oito funções objetivo combinadas duas a duas.

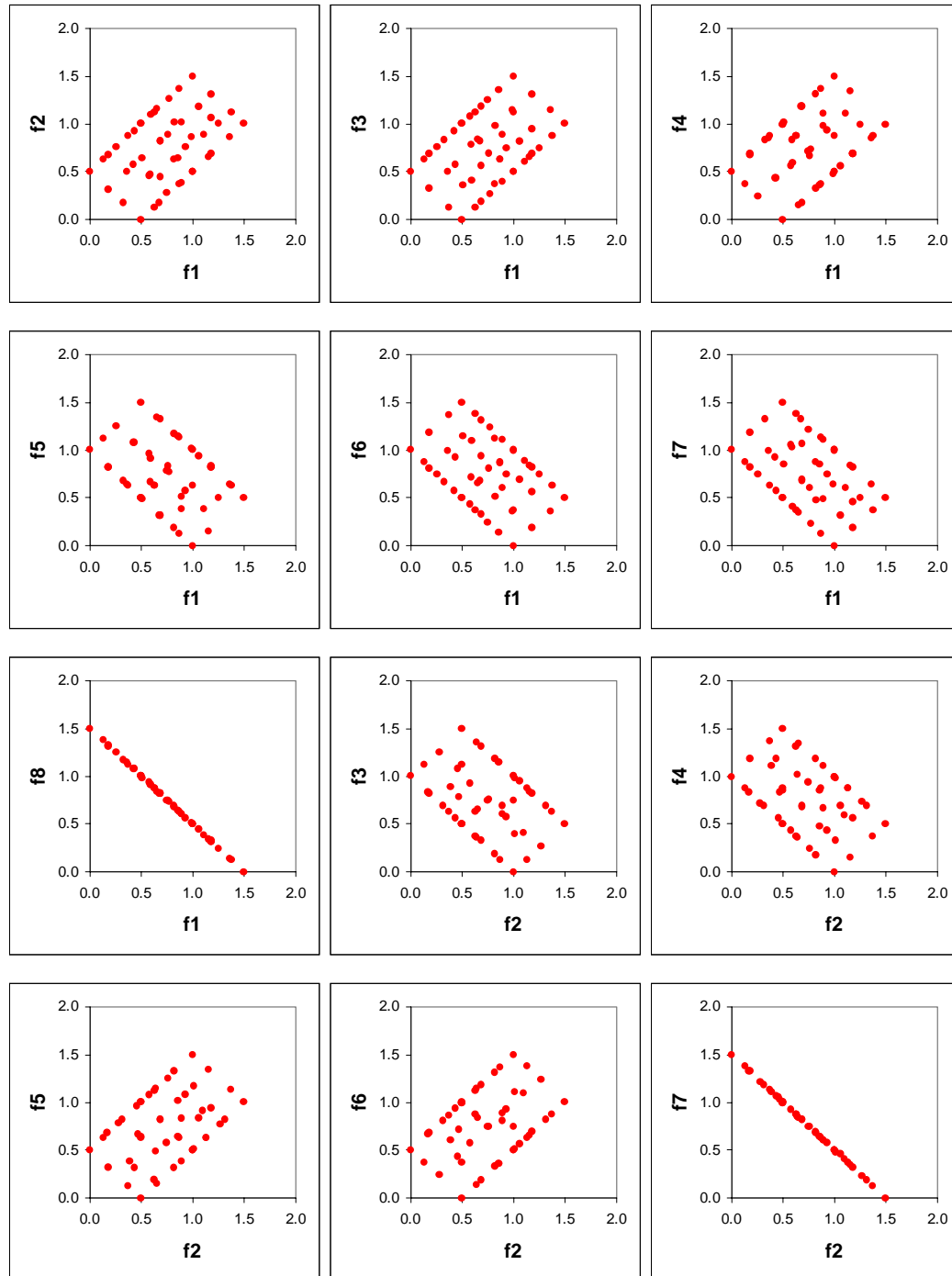


Figura 109 - Funções objetivo combinadas (2)

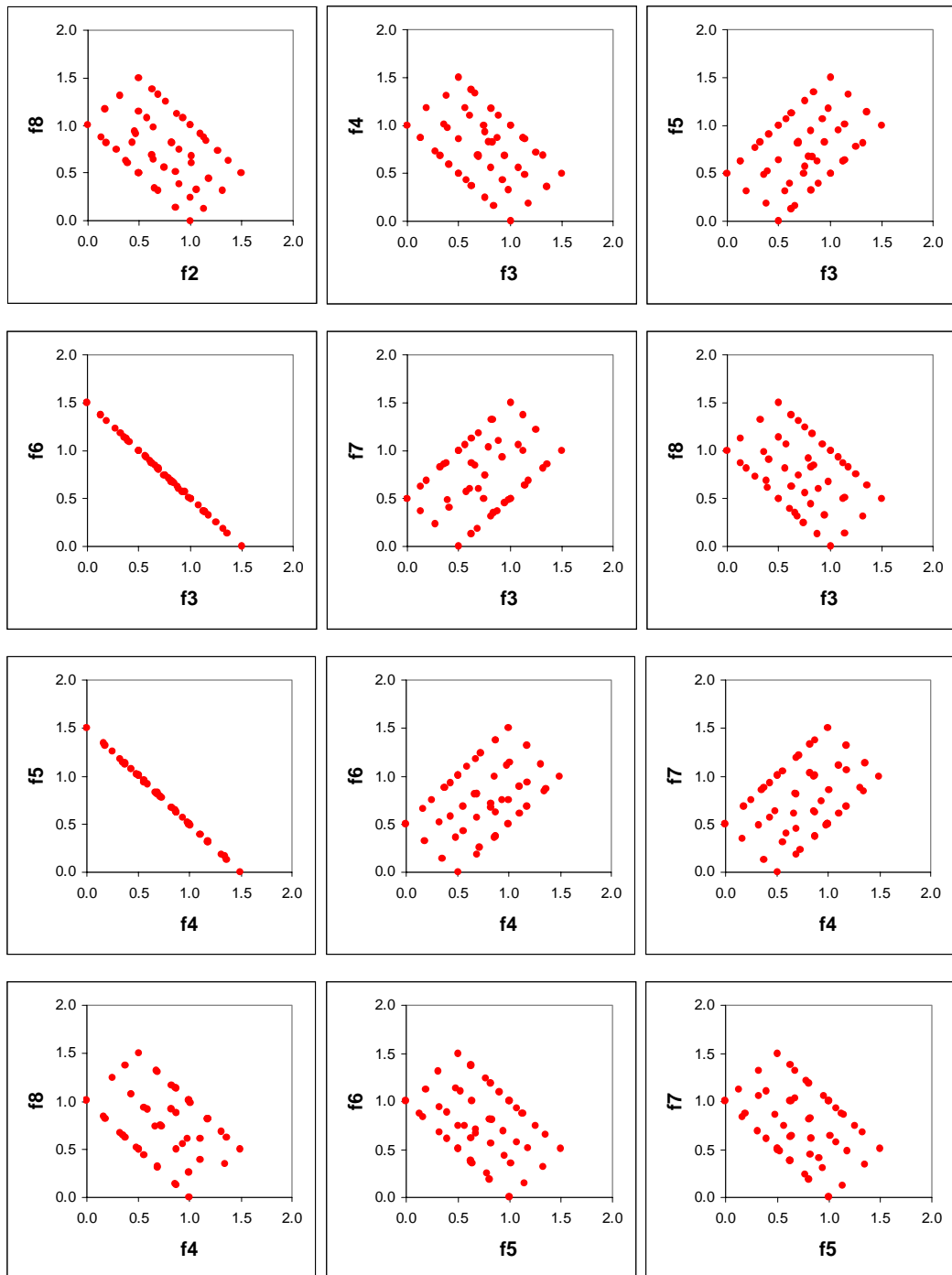


Figura 110 - Funções objetivo combinadas (2)

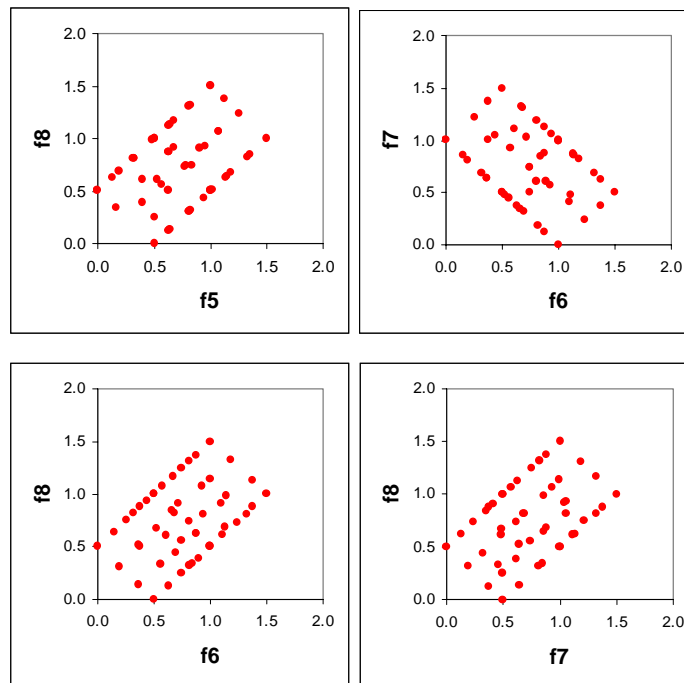


Figura 111 - Funções objetivo combinadas (2)

Nas Figuras seguintes são apresentadas as populações iniciais utilizadas pela terceira configuração do algoritmo – 100 x 200 x 200+800.

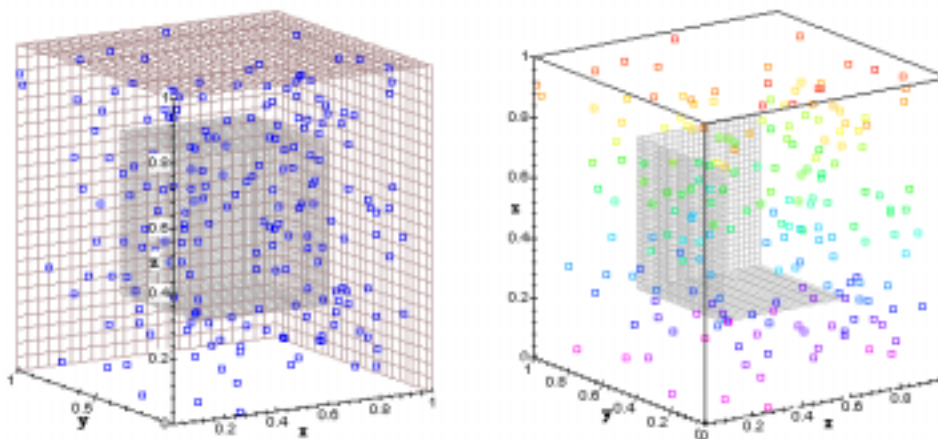


Figura 112 - População inicial (3) – vistas espaciais

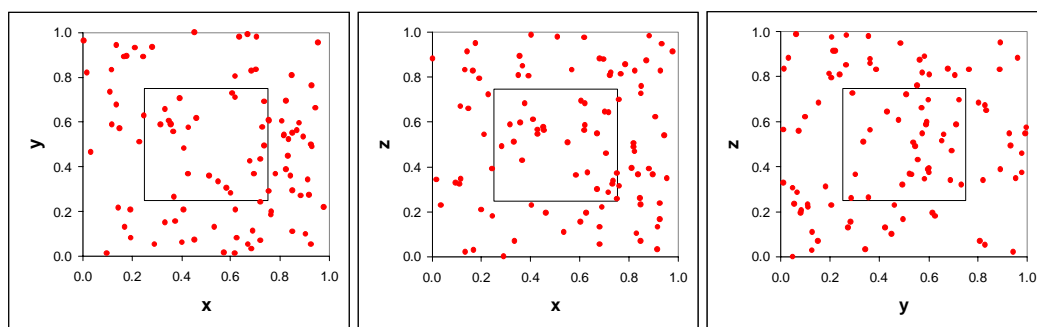


Figura 113 - População inicial (3) – projeções planas

Nas Figuras abaixo são mostradas as soluções encontradas pela terceira configuração do algoritmo.

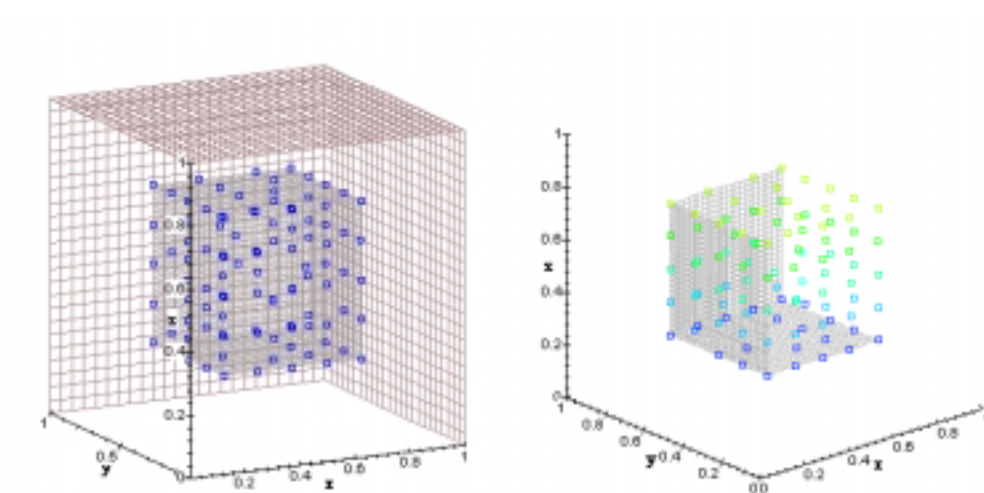


Figura 114 - Soluções encontradas (3) – vistas espaciais

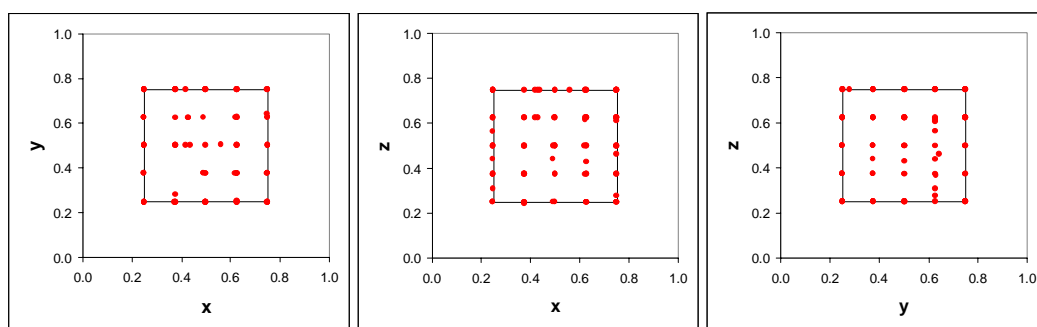


Figura 115 - Soluções encontradas (3) – projeções planas

11.4. Conclusões

Este problema mostrou-se interessante por vários aspectos. O primeiro deles por comprovar a escalabilidade do algoritmo proposto quando do aumento do número de funções objetivo, sendo uma boa contribuição para o meio técnico que era até então carente de problemas desta natureza.

Um segundo ponto relevante é o emprego do operador de otimização individual, o qual permite numa análise preliminar (como na primeira análise realizada) uma clara visualização do espaço de busca e/ou do espaço viável do problema, mesmo que seja pela combinação das funções objetivo como apresentado nesta aplicação.

Uma terceira característica que torna este problema interessante é o fato de que mesmo tendo oito objetivos ele permite a visualização tridimensional das soluções encontradas pelo algoritmo multi-objetivo estudado, facilitando sobremaneira as análises a serem feitas.

Finalmente, pelos resultados apresentados conclui-se que o algoritmo desenvolvido conseguiu bons resultados, já que as soluções encontradas não só otimizaram bem os objetivos individualmente mas foram também bem distribuídas pela região ótima do problema – cubo interno menor.

12. Treliça de 47 barras

12.1. Introdução e Modelo

A treliça utilizada nesta aplicação foi também descrita e analisada em [44] com um só objetivo – o peso estrutural. Na página seguinte é apresentada uma treliça de referência extraída de [44] para uma melhor compreensão do problema.

Foi considerado apenas o caso misto [44], isto é, as variáveis de forma são contínuas no intervalo definido pela tabela das variáveis de forma, enquanto que as variáveis de área são discretas e obtidas no conjunto $T=\{0.1, 0.2, 0.3, \dots, 50.0\}$. Além disso, somente as tensões normais foram tomadas como restrições do problema.

Tanto a nomenclatura das variáveis de área e de forma, quanto a numeração dos nós e elementos estruturais foram mantidos iguais aos utilizados em [44].

A treliça está sujeita a 3 carregamentos conforme mostra a tabela abaixo:

Carregamento	Nó	Fx	Fy
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 12 - Dados de carregamento – cargas em lb

As propriedades do material e as restrições são apresentadas na tabela seguinte:

Propriedade	Valor
Módulo de Elasticidade	30000000.0 psi
Tensões Admissíveis de Tração	20000.0 psi
Tensões Admissíveis de Compressão	-15000.0 psi
Massa Específica	0.3 lb/in ³

Tabela 13 - Propriedades do material e restrições

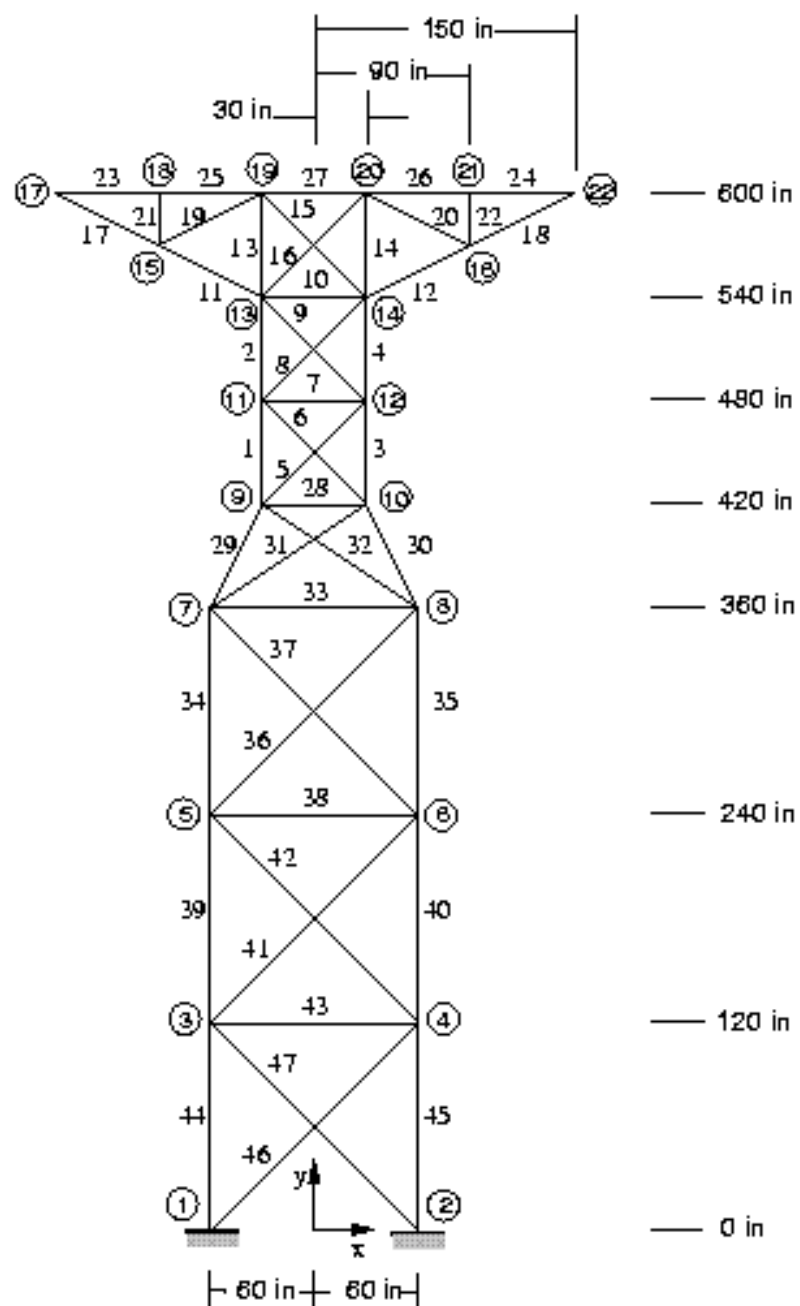


Figura 116 - Treliza de referência com 47 barras [44]

Os nós 15, 16, 17 e 22 são fixos, enquanto que os nós 1 e 2 são semi-fixos, pois podem ter suas coordenadas alteradas em relação ao eixo-x, conforme a tabela abaixo:

Nó	Coord. X	Coord. Y
1	variável	0.0
2	variável	0.0
15	-90.0	570.0
16	90.0	570.0
17	-150.0	600.0
22	150.0	600.0

Tabela 14 - Nós fixos e semi-fixos – coordenadas em in

Todos os demais nós são livres para terem suas coordenadas alteradas em relação aos eixos x e y. Deste modo, foram utilizadas 44 variáveis – 27 de área e 17 de forma que foram codificadas em um cromossomo binário de 373 posições de comprimento.

Pela imposição da simetria vertical (eixo-y) à treliça, conseguiu-se reduzir o número de variáveis de área para 27. Assim, na tabela abaixo são relacionadas as variáveis de área com as áreas dos elementos por elas representadas:

Variável	Área/Elemento		Variável	Área/Elemento
A3	A1, A3		A27	A27
A4	A2, A4		A28	A28
A5	A5, A6		A30	A29, A30
A7	A7		A31	A31, A32
A8	A8, A9		A33	A33
A10	A10		A35	A34, A35
A12	A11, A12		A36	A36, A37
A14	A13, A14		A38	A38
A15	A15, A16		A40	A39, A40
A18	A17, A18		A41	A41, A42
A20	A19, A20		A43	A43
A22	A21, A22		A45	A44, A45
A24	A23, A24		A46	A46, A47
A26	A25, A26			

Tabela 15 - Variáveis de área

A nomenclatura das 17 variáveis de forma e seus respectivos limites, inferior e superior (restrições laterais), são apresentadas na tabela seguinte:

Variável	Lim. Inf.	Lim. Sup.	Variável	Lim. Inf.	Lim. Sup.
X2	60.0	120.0	X12	30.0	60.0
X4	60.0	120.0	Y12	460.0	490.0
Y4	100.0	140.0	X14	30.0	60.0
X6	60.0	120.0	Y14	500.0	540.0
Y6	230.0	270.0	X20	10.0	40.0
X8	60.0	120.0	Y20	570.0	630.0
Y8	330.0	370.0	X21	70.0	110.0
X10	30.0	60.0	Y21	600.0	650.0
Y10	380.0	440.0			

Tabela 16 - Variáveis de forma

12.2. Características e Configurações do AG Utilizado

- Utilização de um filtro para armazenar as soluções não-dominadas.
- Método de seleção por torneio segundo critério de não-dominância.
- Operador convencional de mutação.
- Operadores de recombinação uniforme com probabilidades para geração das máscaras de 0.5 e 0.2, ambos com probabilidades de 50% de utilização.
- Operadores de exclusão, adição e otimização individual.
- População inicial aleatória.
- Execução única (uma só semente).
- Constantes de penalização: $c_1 = 3000.0$ e $c_2 = 3.0$

O algoritmo foi executado três vezes alterando-se o número de gerações, conforme é mostrado na próxima tabela:

Configuração	1	2	3
Tamanho população	50	50	50
Tamanho do filtro	20	20	20
Número de gerações*	1000	5000	5000+25000
Número de sol. Torneio	10	10	10
Prob. de recombinação	90%	90%	90%
Prob. de mutação	1%	1%	1%
Total de avaliações **	52034	260167	810167

* número de gerações padrões + adicionais (filtro)

** para cada função objetivo

Tabela 17 - Configurações utilizadas do AG

12.3. Resultados

Nas figuras que se seguem são mostradas respectivamente a população inicial e a população correspondente à geração 5000. Pode-se observar que não existem soluções viáveis na população inicial, fato que não compromete, entretanto, a resolução do problema, já que tais soluções viáveis são encontradas naturalmente durante o processo evolutivo que possibilita a convergência das populações para a região ótima.

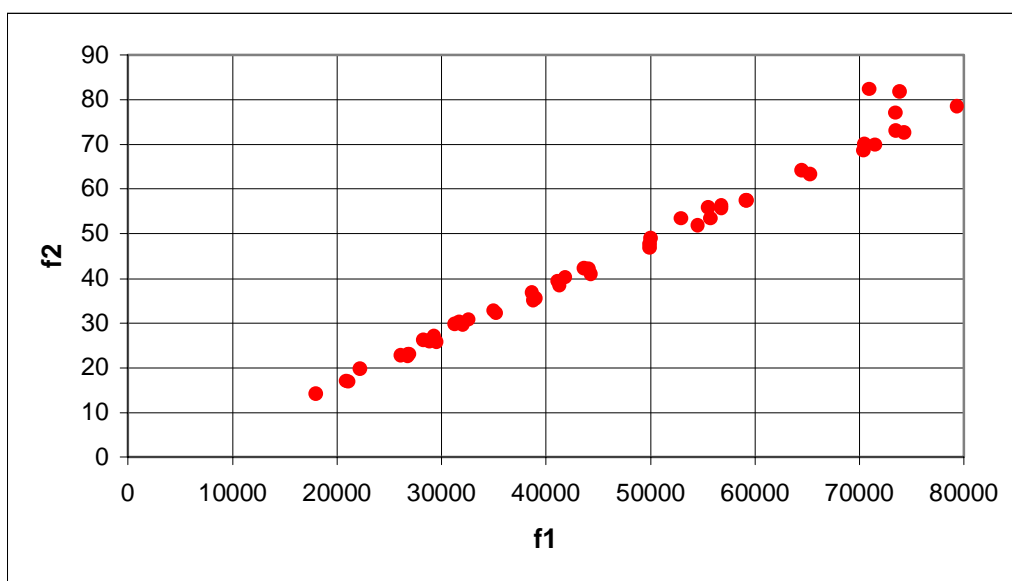


Figura 117 - População inicial

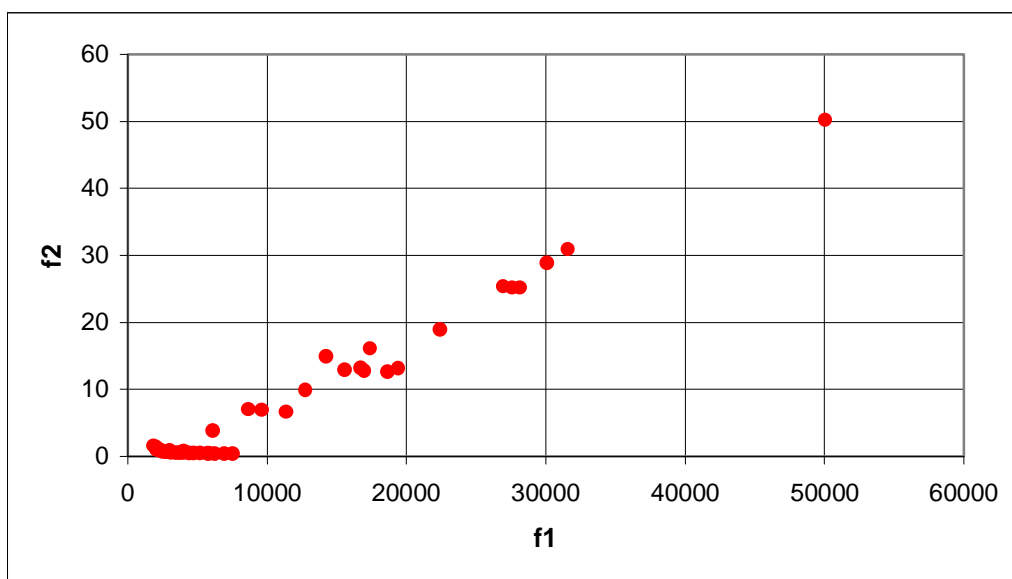


Figura 118 - População na geração 5000

Na figura seguinte são mostradas as fronteiras de Pareto encontradas por cada uma das três configurações utilizadas. Deve-se observar que foi empregada uma mesma semente inicial para o procedimento de geração randômica de números, garantindo-se deste modo que a diferença observada nestas fronteiras seja fruto, único e exclusivamente do acréscimo de gerações.

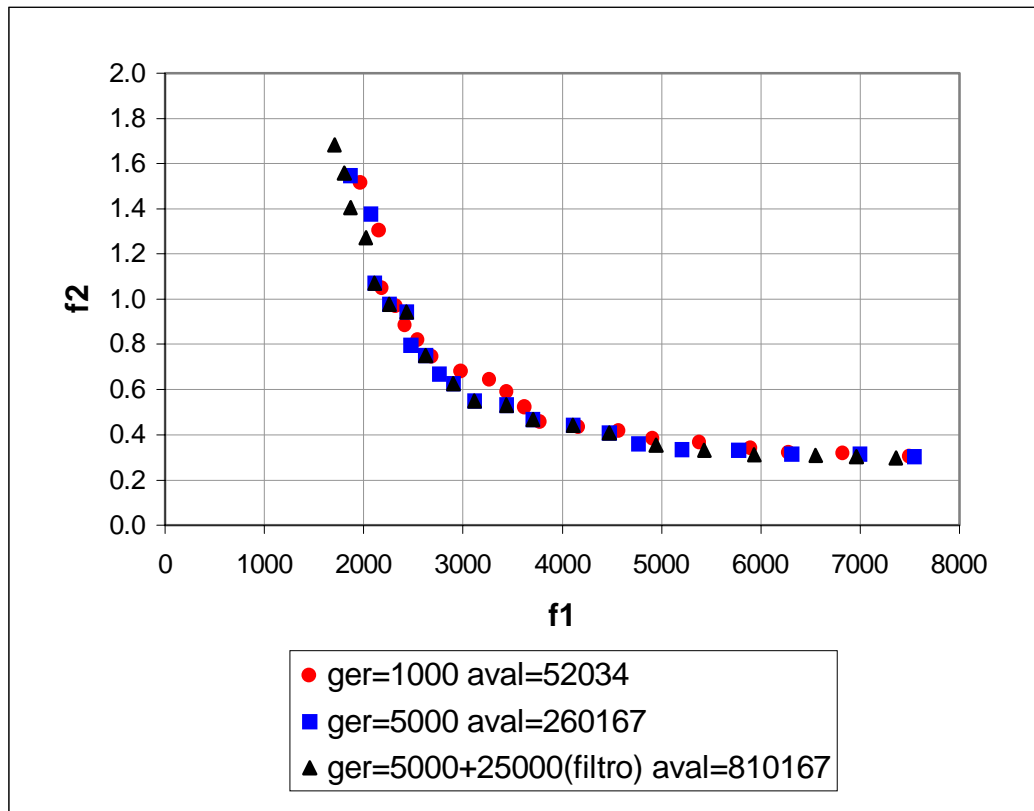


Figura 119 - Fronteiras obtidas das 3 configurações utilizadas

Todas as fronteiras ficaram bem definidas e razoavelmente uniformes, sendo uma excelente orientação para o tomador de decisões, que deve ser um especialista no problema tratado.

Com um número relativamente pequeno de gerações, e conseqüentemente de avaliações, já foi possível obter-se uma boa fronteira de Pareto, que fornece muita informação para o especialista do problema. Assim, segundo suas escolhas particulares, que irão variar caso a caso, pode-se empregar um otimizador simples (para uma única função objetivo) para encontrar uma solução melhor qualificada para o problema.

Na tabela abaixo são mostrados os valores numéricos das funções objetivo do problema abordado para a configuração 3 (5000 gerações convencionais mais 25000 gerações adicionais atuantes no filtro). Deste ponto em diante do trabalho, todos os resultados apresentados serão relativos a esta configuração.

Solução	f1	f2
1	1707.13	1.682
2	1808.61	1.558
3	1868.62	1.404
4	2026.17	1.272
5	2112.44	1.069
6	2261.26	0.975
7	2434.61	0.941
8	2627.21	0.749
9	2907.54	0.624
10	3113.25	0.548
11	3435.34	0.532
12	3707.83	0.467
13	4106.12	0.441
14	4474.96	0.407
15	4944.66	0.355
16	5431.15	0.333
17	5932.06	0.311
18	6556.07	0.308
19	6961.31	0.303
20	7361.39	0.299

Tabela 18 - Valores das funções objetivo do filtro final

Na página seguinte são mostradas as figuras que mostram em diferentes escalas o processo evolutivo da minimização do peso da treliça ao longo das gerações.

Na Tabela 16 são mostrados os valores numéricos das variáveis de projeto de algumas soluções do problema abordado. Cabendo ressaltar que, embora seja mostrada a solução obtida pela referência [44] a título de orientação, nenhuma comparação direta pode ser realizada, já que nesta referência o problema foi resolvido com um único objetivo, o peso, e por um algoritmo de codificação real. Inclusive, esta é uma recomendação para futuros trabalhos, isto é, modificar a metodologia apresentada neste trabalho para operar sobre codificação real que parece ser mais eficiente e de maiores possibilidades de expansão a problemas de grande porte.

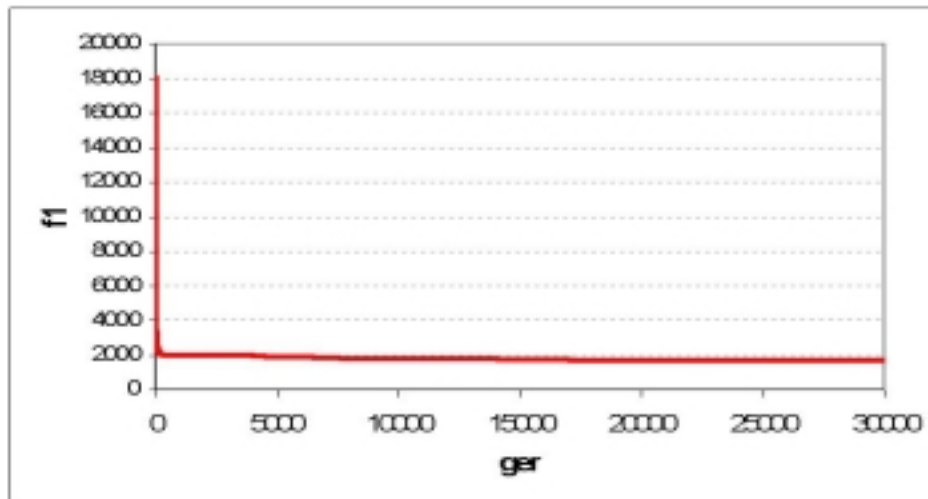


Figura 120 - Minimização do peso durante as 30000 gerações

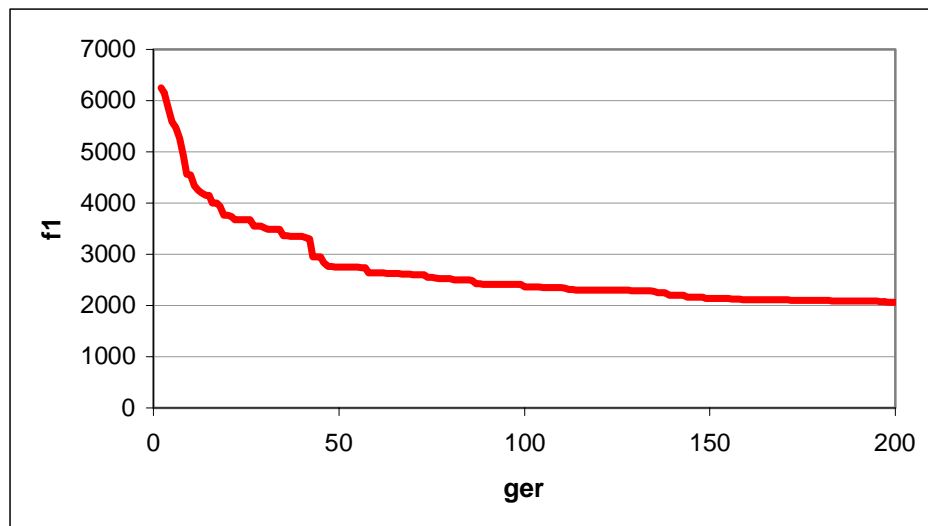


Figura 121 - Minimização do peso, em escala reduzida, durante as 200 primeiras gerações

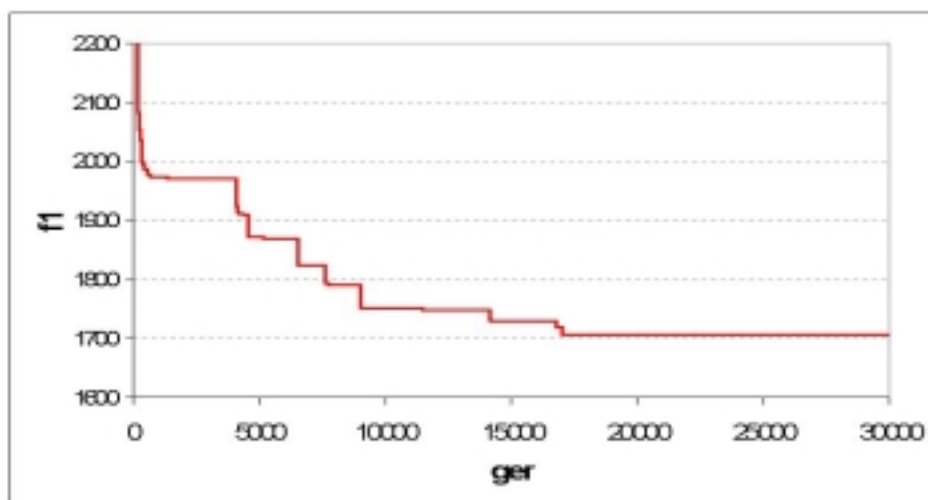


Figura 122 - Minimização do peso, em escala reduzida, durante as 30000 gerações

Variável	Soluções do Filtro Final Encontrado					Ref. **	Lemonge*
	1	5	10	15	20		
A3	3.2	3.2	5.2	6.3	6.3	3.8	2.4
A4	2.4	2.6	6.0	4.7	6.3	3.4	2.3
A5	0.4	0.4	0.4	0.4	0.4	0.8	0.4
A7	0.1	0.2	0.4	1.5	6.3	0.9	0.1
A8	0.8	0.8	0.8	1.5	6.3	0.9	0.6
A10	1.6	1.9	2.1	4.7	6.3	1.8	0.9
A12	3.2	3.3	4.3	6.3	6.3	2.1	1.6
A14	0.5	0.5	1.4	2.9	6.3	1.2	0.5
A15	0.8	1.6	2.1	4.7	6.3	1.6	0.7
A18	1.6	1.6	2.9	5.7	6.3	2.1	1.3
A20	0.4	0.4	0.6	0.5	0.5	0.7	0.3
A22	0.8	2.0	4.3	6.3	6.3	0.9	0.9
A24	1.6	3.2	3.4	4.7	6.3	1.7	1.0
A26	1.6	1.7	2.8	6.3	6.3	1.7	0.9
A27	1.6	1.6	4.3	6.3	6.3	1.4	0.6
A28	0.1	0.3	0.6	1.8	4.8	0.9	0.1
A30	3.2	3.7	6.1	6.3	6.3	3.7	2.4
A31	0.4	0.8	1.2	2.8	6.4	1.5	0.3
A33	0.1	0.1	0.1	1.4	6.3	0.7	0.1
A35	3.2	3.3	4.0	6.4	6.4	2.9	2.7
A36	0.4	0.8	0.4	0.4	0.4	0.7	0.3
A38	0.1	0.1	0.1	0.7	6.3	1.6	0.1
A40	3.2	3.2	4.5	5.5	6.3	3.7	3.0
A41	0.4	0.8	0.9	0.4	0.4	1.6	0.3
A43	0.1	0.1	0.5	1.3	1.7	0.7	0.1
A45	3.2	3.3	3.7	6.0	6.3	4.5	3.3
A46	0.2	0.2	0.4	0.3	0.3	1.6	0.2
X2	104.10	104.70	116.40	119.20	119.10	60.0	97.76432
X4	90.00	90.36	105.90	108.70	108.70	60.0	85.10024
Y4	140.00	117.40	129.60	130.00	130.00	120.0	121.5425
X6	67.50	67.56	88.93	100.70	100.50	60.0	71.52607
Y6	260.50	256.10	269.80	270.00	269.70	240.0	261.4962
X8	60.00	64.86	77.27	94.19	94.22	60.0	65.94714
Y8	336.00	347.10	351.50	338.80	339.10	360.0	334.0145
X10	48.75	56.56	59.79	59.65	60.00	30.0	57.47306
Y10	410.00	427.30	438.90	440.00	440.00	420.0	414.883
X12	45.00	54.92	58.13	58.95	58.95	30.0	50.13195
Y12	460.00	464.80	460.90	460.20	460.20	480.0	466.5514
X14	48.75	58.37	59.71	59.53	60.00	30.0	48.77601
Y14	530.00	524.20	523.60	500.10	500.00	540.0	508.6798
X20	17.50	11.90	14.19	10.06	10.06	30.0	16.88877
Y20	592.50	599.00	628.00	622.40	630.00	600.0	606.7329
X21	105.60	108.20	97.01	90.84	92.16	90.0	87.71766
Y21	609.40	635.50	635.20	650.00	650.00	600.0	628.7389
f1 (lb)	1707.13	2112.44	3113.25	4944.66	7361.39	2446.8	1432.993
f2 (in)	1.682	1.069	0.548	0.355	0.299	2.745	1.583

* Otimização simples de f1 – referência [44]

** Treliça de referência – Figura 89

Tabela 19 - Valores das variáveis de projeto de algumas soluções do problema

Nas figuras que se seguem são mostradas as formas estruturais encontradas para algumas das soluções armazenadas no filtro final, bem como a forma da solução apresentada na referência [44].

Nas Tabelas 17 e 18 que vêm na sequência são apresentadas as tensões normais decorrentes da aplicação dos três casos de carregamento para as soluções 1 e 20 do filtro final, respectivamente as soluções de menor e maior peso do filtro encontrado.

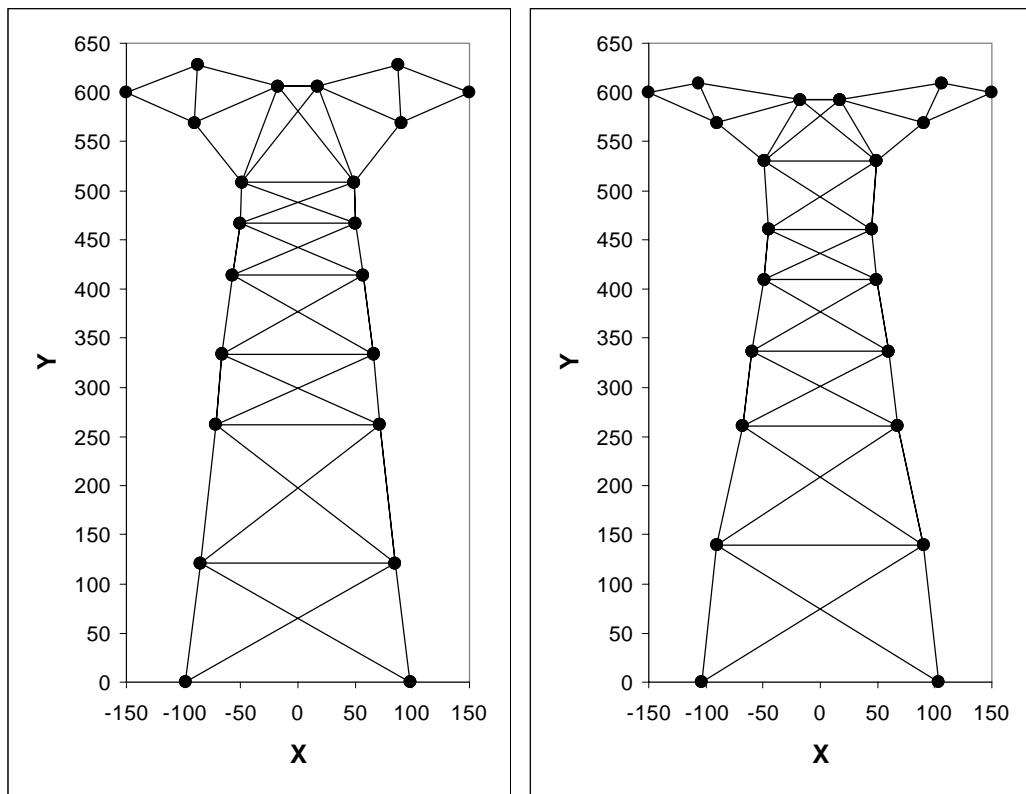


Figura 123 - Forma da solução ótima da referência [44] e da solução 1 do filtro, respectivamente

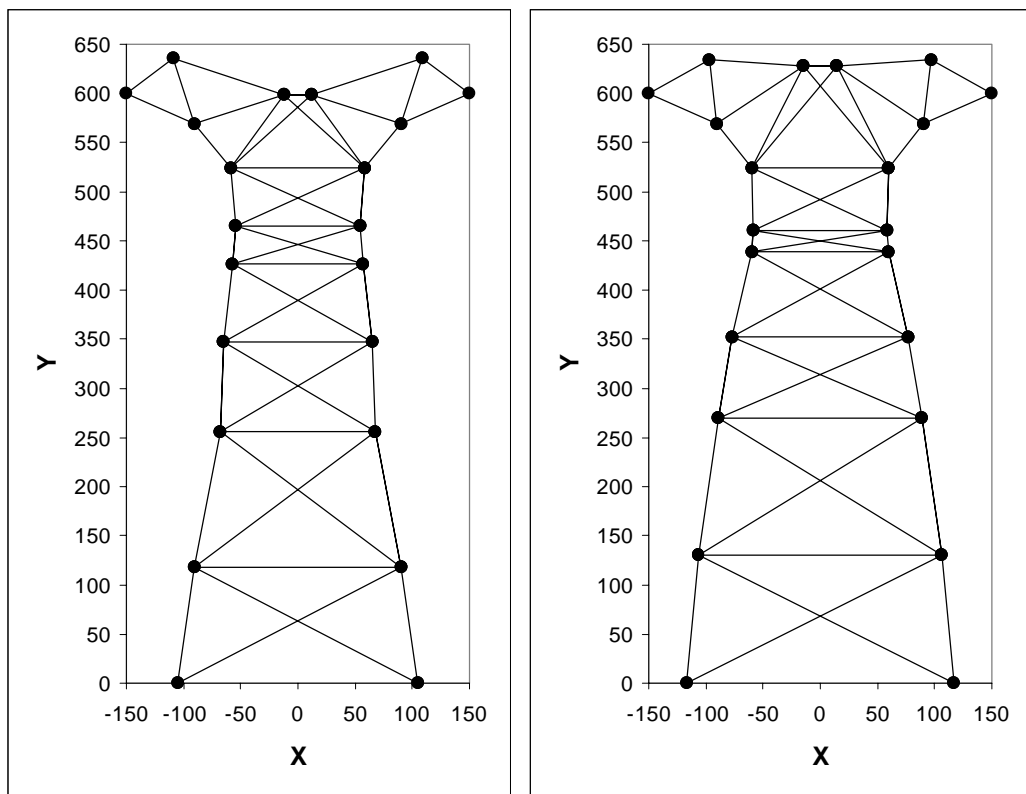


Figura 124 - Formas das soluções 5 e 10 do filtro, respectivamente

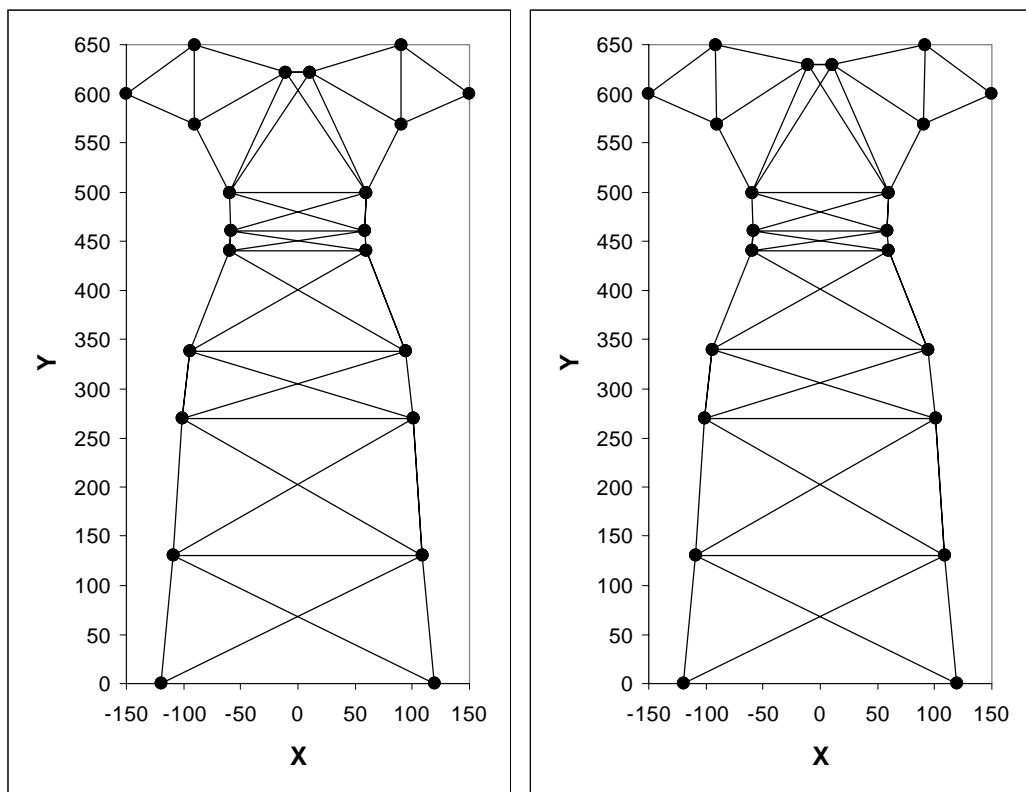


Figura 125 - Formas das soluções 15 e 20 do filtro, respectivamente

Tensões normais - Sol. 1 - Carregamentos 1, 2 e 3			
Barra	sigma - C1	sigma - C2	sigma - C3
1	2239.85	-5908.97	8148.81
2	553.69	-9092.79	9646.47
3	-10937.60	1563.94	-12501.54
4	-10829.74	3938.75	-14768.49
5	11880.34	10675.43	1204.94
6	-13182.59	-11391.81	-1790.76
7	7878.62	3792.04	4086.62
8	6976.29	1653.39	5322.88
9	-14022.91	-5096.58	-8926.34
10	-13116.75	-6277.13	-6839.68
11	-8489.93	-8489.91	0.03
12	-8912.03	0.01	-8912.03
13	15725.66	-2441.20	18166.84
14	-1435.59	9450.52	-10886.09
15	-3861.82	10937.45	-14799.25
16	11831.26	-7698.69	19529.91
17	-14992.57	-14992.62	0.07
18	-12498.30	-0.01	-12498.29
19	-12105.01	-12105.09	0.03
20	-4038.58	-0.02	-4038.51
21	-9074.27	-9074.32	0.11
22	-14023.89	0.02	-14023.88
23	9874.00	9873.91	-0.03
24	15259.63	0.00	15259.60
25	11536.68	11536.67	0.01
26	17829.27	-0.01	17829.28
27	12042.29	4120.74	7921.52
28	-7062.04	-3468.80	-3593.23
29	3472.20	-4434.57	7906.76
30	-12263.64	36.96	-12300.60
31	6284.29	12128.84	-5844.54
32	-7115.00	-12517.64	5402.65
33	10043.35	4999.63	5043.73
34	4512.11	-2933.84	7445.95
35	-13254.74	-1436.87	-11817.88
36	8873.54	9202.47	-328.93
37	-9661.31	-9605.85	-55.46
38	-8004.45	-3993.26	-4011.19
39	5283.09	-1685.41	6968.49
40	-14058.59	-2702.62	-11355.97
41	192.16	8097.93	-7905.77
42	-1819.05	-8907.84	7088.79
43	10798.08	5395.96	5402.12
44	5776.10	-527.85	6303.94
45	-14716.41	-3942.24	-10774.17
46	18719.75	16112.44	2607.31
47	-14744.86	-14126.98	-617.88

Tabela 20 - Tensões normais da solução 1 do filtro

Tensões normais - Sol. 20 - Carregamentos 1, 2 e 3			
Barra	sigma - C1	sigma - C2	sigma - C3
1	177.31	-2717.80	2895.12
2	-132.34	-2860.75	2728.41
3	-4627.86	492.28	-5120.14
4	-3979.91	822.52	-4802.43
5	13225.96	8952.44	4273.53
6	-13220.48	-8924.99	-4295.48
7	328.70	190.54	138.16
8	532.07	159.88	372.19
9	-1583.63	-742.03	-841.60
10	-1037.33	-609.55	-427.79
11	-3496.50	-3496.50	-0.01
12	-2908.09	-0.01	-2908.08
13	1554.66	363.89	1190.77
14	-725.28	236.85	-962.14
15	-522.20	740.51	-1262.71
16	1281.11	-251.15	1532.27
17	-2495.48	-2495.50	0.00
18	-1146.29	0.00	-1146.28
19	-14072.60	-14072.49	-0.10
20	956.68	0.05	956.61
21	-1427.85	-1427.85	0.00
22	-2206.71	-0.01	-2206.69
23	1691.51	1691.51	0.00
24	2614.19	0.00	2614.18
25	1356.72	1356.75	0.01
26	2096.85	0.02	2096.83
27	1230.19	204.15	1026.08
28	-194.35	-97.93	-96.41
29	664.10	-2068.41	2732.51
30	-4802.87	-0.75	-4802.12
31	-384.34	714.61	-1098.96
32	-559.61	-1186.95	627.33
33	854.23	427.22	427.00
34	650.32	-1368.64	2018.95
35	-5050.45	-831.42	-4219.03
36	11732.88	8459.18	3273.70
37	-11390.83	-8288.06	-3102.77
38	95.77	47.88	47.89
39	1426.96	-885.13	2312.09
40	-5834.02	-1318.38	-4515.64
41	13370.84	8459.30	4911.54
42	-14647.44	-9097.60	-5549.85
43	30.56	15.28	15.28
44	2255.81	-343.16	2598.97
45	-6689.10	-1873.47	-4815.63
46	13868.76	9770.59	4098.17
47	-14941.84	-10307.12	-4634.71

Tabela 21 - Tensões normais da solução 20 do filtro

Como foi sugerido anteriormente, a otimização multi-objetivos deve sempre que possível ser empregada, pois ela fornece informações importantes ao especialista do problema tratado que podem auxiliá-lo na escolha adequada de suas soluções.

Além disso, a otimização multi-objetivos deverá cada vez mais se tornar um procedimento padrão, já que os problemas se tornam sempre maiores e mais complexos, não permitindo mais a estes especialistas a obtenção de soluções oriundas de sua prévia experiência com rigorosos ganhos de custo e/ou segurança, como pode ser disponibilizado com o emprego da otimização.

Como consequência destes fatores, espera-se diminuir o tempo de projeto e com isso o tempo utilizado pelos especialistas, que terão ao longo dos anos os seus custos de trabalho aumentados, enquanto que o custo computacional tende a diminuir com o avanço da capacidade computacional e da tecnologia de uma forma geral.

Para facilitar a decisão destes especialistas podem e devem ser utilizadas etapas intermediárias da otimização multi-objetivos com o intuito de diminuir o leque de soluções para o problema específico a ser resolvido. Assim, o tomador de decisões, conforme suas necessidades, pode definir novos limites para as funções objetivo, o que é realizado numericamente pela introdução de novas restrições ao problema original.

Para exemplificar este procedimento, foi estabelecido como condição para o corrente problema analisado, o intervalo de interesse dos pesos estruturais: $\text{peso} \in [2750, 3250]$, informação que pode e deve ser introduzida pelo especialista do problema a ser resolvido na vida prática.

Com a introdução desta nova informação, o problema foi mais uma vez resolvido com a mesma configuração (3) e tem seus resultados apresentados nas próximas figuras. Este procedimento poderia se repetir quantas vezes fossem necessárias, até mesmo o emprego de uma otimização simples (um só objetivo) com restrições que garantissem as pré-escolhas do especialista poderia ser utilizada para a certeza de uma otimização rigorosa.

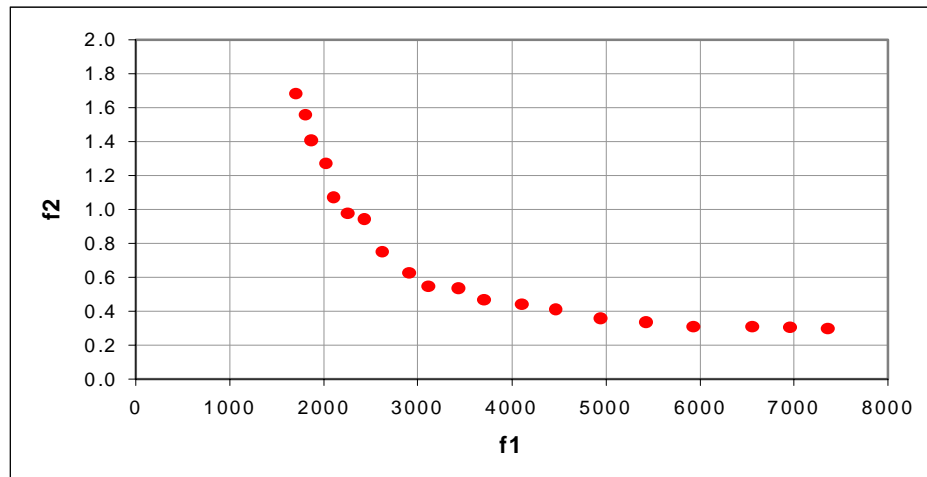


Figura 126 - Filtro final – geração 5000 + 25000

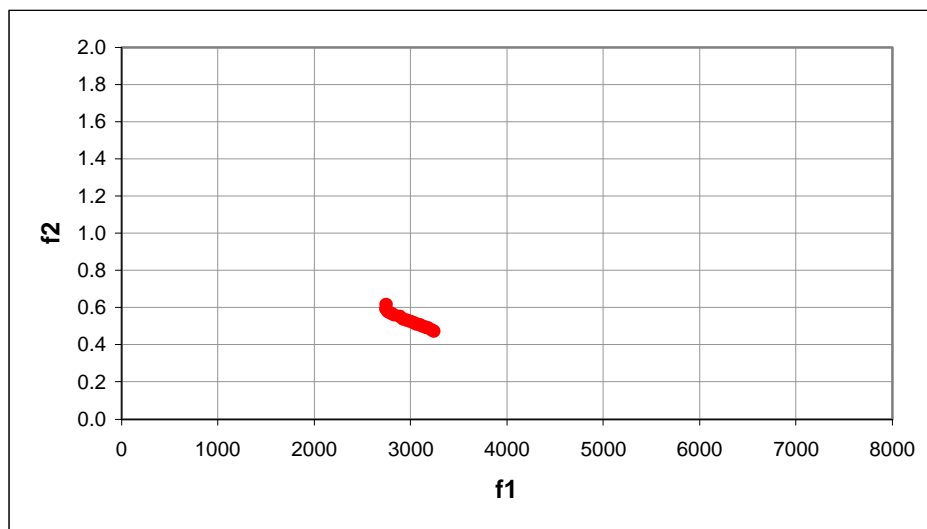


Figura 127 - Filtro final orientado no espaço de busca

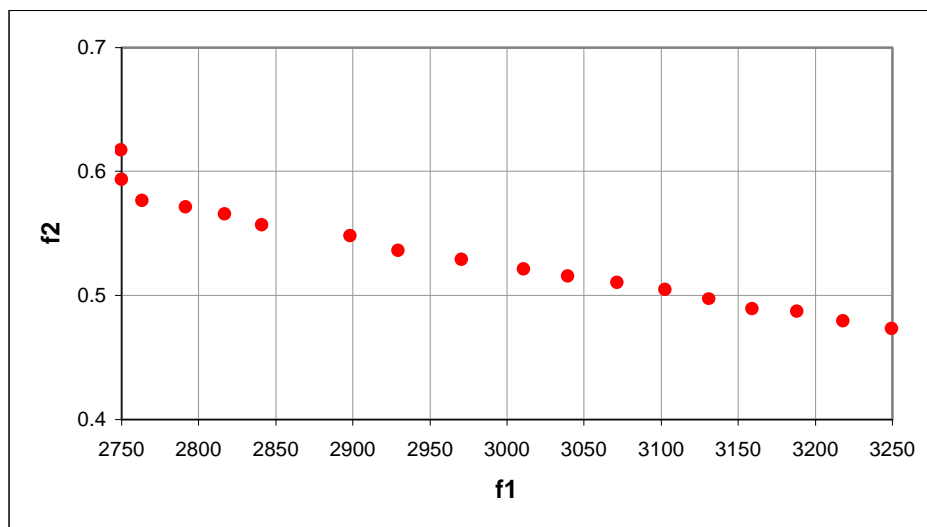


Figura 128 - Detalhe do filtro final orientado

12.4. Conclusões

O problema descrito é relativamente difícil pela grande taxa de soluções inviáveis existente, já que as restrições sobre tensões impostas são bastante restritivas, além disto, há um grande e complexo espaço de busca a ser pesquisado, pois enquanto uma melhor barra é encontrada uma outra coordenada pode simultaneamente tornar a solução infactível. Ainda assim, todas as soluções apresentadas respeitam rigorosamente as restrições impostas no problema.

O algoritmo empregado em [44] é do tipo em regime (steady-state) com codificação real. Além de ter utilizado uma população de 1200 indivíduos que totalizaram 480000 avaliações para cada uma das 10 execuções realizadas com o algoritmo lá apresentado.

Em análises preliminares realizadas, foi verificado que o algoritmo desenvolvido alcançou melhores resultados trabalhando com populações menores – cerca de 50 indivíduos – por um maior número de gerações, embora configurações semelhantes às utilizadas em [44] também resolvessem o problema. Além disso, populações menores proporcionam redução computacional na etapa do teste de dominância das soluções da população.

Os resultados podem ser melhorados com o aumento do número de gerações, inclusive no que diz respeito aos ótimos individuais de cada função objetivo. Deste modo, poder-se-ia encontrar soluções mais próximas da apresentada em [44], mesmo não sendo esta a filosofia prioritária da otimização multi-objetivos.

Os resultados de [44] não podem ser comparados com os resultados encontrados pelo algoritmo proposto, uma vez que o mesmo tem caráter multi-objetivo. Entretanto, eles são aqui apresentados para servirem como orientação de análise em relação aos resultados multi-objetivos encontrados.

É preciso destacar que pelo fato do problema multi-objetivo proposto apresentar um espaço de busca muitas vezes mais abrangente e complexo que o seu correspondente problema de um só objetivo (peso), é razoável esperar um número bastante superior de avaliações para a solução do problema, já que a cada geração existem variadas direções de busca para serem pesquisadas.

Outro fator importante é o fato do algoritmo desenvolvido operar sobre cromossomos binários, característica que lhe permitiu ser empregado na análise de alguns problemas que o requeriam, como o próprio conjunto das funções-teste para otimizadores multi-objetivos.

Apesar destes fatores, os resultados encontrados com o algoritmo PMOGA foram razoáveis e possibilitaram a obtenção de fronteiras bem definidas e uniformes.

Uma sugestão para estudos futuros é o desenvolvimento de um algoritmo semelhante ao PMOGA mas operando sobre uma codificação real, o que deverá prover uma maior eficiência computacional, bem como permitir a análise de problemas maiores e mais complexos. Uma referência bastante completa sobre a codificação real pode ser encontrada em [44].

Capítulo IX

Conclusões

Este capítulo apresenta as principais conclusões resultantes do trabalho realizado, destacando as contribuições, os resultados, as experiências adquiridas e as áreas para desenvolvimento em trabalhos futuros.

1. Aspectos Gerais	194
2. Contribuições.....	194
3. Resultados	195
4. Experiências Adquiridas	195
5. Trabalhos Futuros	196

1. Aspectos Gerais

Com a grande capacidade computacional que está a serviço do homem, abrir mão de processos de otimização é, sem dúvida, desperdício de dinheiro. Além de outros fatores, não menos importantes, que deixam de ser estudados e melhorados como, por exemplo: confiabilidade, segurança, viabilidade prática, etc.

2. Contribuições

Foi implementado um algoritmo genético multi-objetivo binário que aproxima a fronteira ótima de Pareto de problemas de otimização multi-objetivos, restringidos ou não. Por suas características e estratégias empregadas o algoritmo mostrou-se robusto e eficiente para uma ampla gama de aplicações.

Inúmeros problemas multi-objetivos testes e/ou estruturais foram analisados para comprovarem o funcionamento e a eficiência do algoritmo proposto.

Um método prático de penalização foi desenvolvido e largamente utilizado com sucesso e relativa facilidade.

Novos operadores genéticos multi-objetivos foram elaborados com o intuito de melhorar a distribuição das soluções pela fronteira de Pareto, substituindo as estratégias de nicho normalmente utilizadas que introduzem parâmetros sensíveis à resolução dos problemas.

Pelo bom desempenho obtido com o algoritmo verifica-se a grande importância do emprego da estratégia de elitismo e/ou de um filtro para armazenar as soluções não-dominadas correntes do processo evolutivo de otimização utilizado.

3. Resultados

Os resultados dos problemas resolvidos foram em sua maioria equivalentes ou superiores aos encontrados na literatura.

Pode-se dizer que o desempenho do algoritmo foi excelente, pois pelas comparações realizadas ele superou o algoritmo SPEA, o qual já comprovou seu valor por diversas comparações e análises realizadas [71].

4. Experiências Adquiridas

O método de seleção por torneio mostrou ser muito simples e robusto, superando todos os problemas oriundos do método da roleta – estagnação e convergência prematura.

O método de penalização utilizado mostrou-se bastante prático, pois não exigiu exaustivas configurações de parâmetros de penalidade para o tratamento das restrições dos problemas analisados e forneceu bons resultados.

O operador de recombinação uniforme incorpora a facilidade da independência quanto ao tamanho do problema (cromossomo) tratado, podendo acelerar ou não a capacidade de pesquisa pelo espaço de busca conforme a escolha das probabilidades de máscara adotadas.

A codificação binária torna-se desaconselhável para problemas de grande porte, já que exige esforço computacional para a decodificação, bem como diminui a eficiência da busca por operar com cromossomos excessivamente longos. Um algoritmo semelhante ao desenvolvido com codificação real é uma sugestão para desenvolvimentos futuros.

5. Trabalhos Futuros

Falar em trabalhos futuros no campo da inteligência artificial é uma tarefa difícil, pois muito há para ser feito. Entretanto, algumas idéias que podem ser abordadas em trabalhos futuros são:

- Na área da otimização multi-objetivo:
 - Resolução de novas aplicações maiores e mais complexas devido ao amadurecimento dos métodos e do avanço computacional.
 - Tratamento de problemas dinâmicos – peso x frequências naturais de vibração, que é um problema “fortemente” multi-objetivo.
 - Desenvolvimento e teste de novas estratégias de penalização, seleção, recombinação, mutação, elitismo, operadores, adaptação, etc.
- Em outras áreas de otimização:
 - Acoplamento da técnica de Redes Neurais com os Algoritmos Genéticos de diferentes formas, principalmente no cálculo aproximado das aptidões.
 - Resolução de problemas topológicos de barras – número livre de elementos e posições (conectividade).
 - Otimização de vigas de concreto armado, onde em geral, o custo não é proporcional ao peso. Confrontando os resultados com os obtidos pelas técnicas da programação matemática

Referências

- [1] BARÁN, B., KASZKUREWICS, E., BHAYA, A., “Parallel Asynchronous Team Algorithms Convergence and Performance Analysis”, *IEEE Transactions on Parallel and Distributed Systems*, v. 7, n. 7, Jul. 1996.
- [2] BARÁN, B., *Estudo de Algoritmos Combinados Paralelos Assíncronos*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1993.
- [3] BARBOSA, H. J. C., “Algoritmos Genéticos para Otimização em Engenharia: Uma Introdução”. Em: *IV Seminário sobre Elementos Finitos e Métodos Numéricos em Engenharia*, Juiz de Fora, 1996.
- [4] BAZARAA, M. S., SHETTY, C. M., *Nonlinear Programming - Theory and Algorithms*. New York, John Wiley & Sons, 1979.
- [5] BLICKLE, T., THIELE, L., *A Comparison of Selection Schemes used in Genetic Algorithms*. Report No. 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 1995.
- [6] BORGES, C. C. H., *Algoritmos Genéticos para Otimização em Dinâmica de Estruturas*. Tese de D.Sc., Programa de Engenharia Civil, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Dezembro de 1999.
- [7] CANTÚ-PAZ, E., *A Summary of Research on Parallel Genetic Algorithms*. IlliGal Report No. 95007, Jul. 1995.

- [8] CASTRO, R. E., BARBOSA, H. J. C., “A Genetic Algorithm for Multiobjective Structural Optimization”. Em: *Anais do IV SIMMEC – Simpósio Mineiro de Mecânica Computacional*, Uberlândia, Maio de 2000.
- [9] CASTRO, R. E., BARBOSA, Helio J. C., “Otimização Estrutural com Multi-objetivos Via Algoritmos Genéticos”. Em: *Anais do IV Congresso de Engenharia Civil*, Volume 1, pp. 275-286, Juiz de Fora, Agosto de 2000.
- [10] CASTRO, R. E., *Otimização: Programação Matemática e Algoritmos Genéticos*. Monografia para inscrição direta de D.Sc., Programa de Engenharia Civil, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Fevereiro de 1999.
- [11] CASTRO, R. E., *PMOGA – Pareto Multiobjective Genetic Algorithms Home Page*. Em: <http://www.sdnnet.com.br/~bicofino/rodrigo.htm>
- [12] CHENG, F. Y., LI, D., “Genetic Algorithm Development for Multiobjective Optimization of Structures”, *AIAA Journal*, v. 36, n. 6, Jun. 1998.
- [13] CHENG, F. Y., LI, D., “Multiobjective Optimization Design with Pareto Genetic Algorithm”, *Journal of Structural Engineering*, pp. 1252-1261, Sep. 1997.
- [14] CIVITA, V. (Editor), Em: *Enciclopédia Conhecer*, volume VII, Editoras Abril S. A. – Cultural e Industrial, pp. 1646-1648, 1659-1661.
- [15] COELLO, C. A., CHRISTIANSEN, A. D., *Two New GA-based Methods for Multiobjective Optimization*. Tulane University. New Orleans, LA 70118, USA.

- [16] COELLO, C. A. C., *An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends*. Laboratorio Nacional de Informática Avanzada. Veracruz, México.
- [17] COELLO, C. A. C., *Self-Adaptive Penalties for GA-based Optimization*. Laboratorio Nacional de Informática Avanzada, Rébsamen 80, Xalapa, Veracruz 91090, México.
- [18] CUNHA, A. G., OLIVEIRA, P., COVAS, J., “Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems”. In: *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 682-688, San Francisco, California, 1997.
- [19] DAVIS, L., *Handbook of Genetic Algorithms*. New York, Van Nostrand Reinhold, 1991.
- [20] DEB, K., *Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design*. Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, Kanpur.
- [21] DEB, K., *Non-linear Goal Programming Using Multi-Objective Genetic Algorithms*. In: Technical Report No. CI-60/98. Department of Computer Science/XI, University of Dortmund, Germany.
- [22] DE JONG, K. A., *An Analysis of the Behavior of a class of Genetic Adaptive System*. Ph.D. dissertation, University of Michigan, ANN Arbor, MI., 1975.
- [23] DORIGO, M., *Ant Colony Optimization*. Iridia. Université Libre de Bruxelles, Belgium. <http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

- [24] ELDRED, M. S., *Optimization Strategies for Complex Engineering Applications*. Sandia Report, Sandia National Laboratories, 1998.
- [25] ESQUIVEL, S. C., LEIVA, H. A., GALLARD, R. H., *Multiplicity in Genetic Algorithms to Face Multicriteria Optimization*. Departamento de Informática, Universidade de San Luis, Argentina.
- [26] FONSECA, C. M., FLEMING, P. J., “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization”. In: *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416-423, 1993.
- [27] *Fuzzy Logic Tutorial*. www.cacs.usl.edu/~manaris/ai-education-repository/fuzzy-tutorial.html
- [28] GREENWOOD, G. W., HU, X. S., D'AMBROSIO, J. G., “Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings”. In: *Foundations of Genetic Algorithms 4 (FOGA-96)*, pp. 437-455, Morgan Kaufmann, 1996.
- [29] GREFENSTETTE, J. J., “Optimization of Control Parameters for Genetic Algorithms”. *IEE Trans, Syst. Man. Cyber*, pp. 122-128, 1986.
- [30] GOLDBERG, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Adison-Wesley, USA, 1989.
- [31] GUROVITZ, H., “Genética: O Negócio da Vida”. *Revista Exame*, n. 11, pp. 40-52, Maio de 2000.
- [32] HAFTKA, R. T., KAMAT, M. P., *Elements of Structural Optimization*. Dordrecht, Martinus Nijhoff Publishers, 1985.

- [33] HAJELA, P., LIN, C. Y., “Genetic Search Strategies in Multicriterion Optimal Design”. In: *Structural Optimization*, v. 4, pp. 99-107, 1992.
- [34] HINTERDING, R., MICHALEWICZ, Z., EIBEN, A. E., *Adaptation in Evolutionary Computation: A Survey*. In: <http://www.coe.uncc.edu/~zbyszek/papers.html>
- [35] HIROYASU, T., MIKI, M., WATANABE, S., “Divided Range Genetic Algorithms in Multiobjective Optimization Problems”. *International Workshop on Emergent Synthesis – IWES '99*, Kobe, Dec. 1999.
- [36] HOLLAND, J. H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [37] HORN, J., NAFPLIOTIS, N., *Multiobjective Optimization Using the Niche Pareto Genetic Algorithm*. IlliGAL Report No. 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign, Jul. 1993.
- [38] IGNIZIO, J. P., *Goal Programming and Extensions*. Lexington Books, London, Massachusetts, 1976.
- [39] IGNIZIO, J. P., *Linear Programming for Decision Analysis*. Averbach Publishers, Philadelphia, 1982.
- [40] ISHIBUCHI, H., MURATA, T., “Multiobjective Genetic Local Search Algorithm”. In: *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pp. 119-124, 1996.
- [41] KIRKPATRICK, S., GELATT Jr., C. D., VECCHI, M. P., “Optimization by Simulating Annealing”. In: *Science*, 220, No. 4598, pp. 671-680.

- [42] KIRSCH, U., *Optimum Structural Design*. McGraw-Hill, 1981.
- [43] KOZA, J. R., In: www.genetic-programming.com/johnkoza.html
- [44] LEMONGE, A. C. C., *Aplicação de Algoritmos Genéticos em Otimização Estrutural*. Tese de D.Sc., Programa de Engenharia Civil, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Junho de 1999.
- [45] LIS, J., EIBEN, A. E., “A Multi-Sexual Genetic Algorithm for Multiobjective Optimization”. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pp. 59-64, 1997.
- [46] LUENBERGER, D. G., *Introduction to Nonlinear Programming*. Addison-Wesley Publishing Company, London, 1984.
- [47] MANDELBROT, B., *The fractal geometry of nature*. W. H. Freeman, New York, 1983.
- [48] MARIANO, C. E., MORALES, E., *A New Distributed Reinforcement Learning Algorithm for Multiple Objective Optimization Problems*. Morelos, México.
- [49] MEDEIROS, A., BARBOSA, H. J. C., “Adaptação das Probabilidades dos Operadores Genéticos: Um Problema de Decisão”. Em: *V Seminário sobre Elementos Finitos e Métodos Numéricos em Engenharia*, UFJF, Juiz de Fora, 1998.
- [50] MICHALEWICZ, Z., SCHOENAUER, M., *Evolutionary Algorithms for Constrained Parameter Optimization Problems*. In: <http://www.coe.uncc.edu/~zbyszek/papers.html>

- [51] MICHALEWICZ, Z., *Genetic Algorithms, Numerical Optimization, and Constraints*. Department of Computer Science, University of North Carolina. Charlotte, NC 28223.
- [52] MICHELL, M., “*An Introduction to Genetic Algorithms (Complex Adaptive Systems)*”. 1996
- [53] NEVES, F. A., *Programação com Multi-Objetivos Aplicada à Otimização do Projeto de Pontes Estaiadas*. Tese de D.Sc., Programa de Engenharia Civil, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997.
- [54] PENADILLOS, E. M. M., *Auto-sintonia Robusta de Controladores PID Usando Algoritmos Genéticos*. Dissertação de M.Sc., Programa de Engenharia Elétrica, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997.
- [55] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., FLANNERY, B. P., *Numerical Recipes in Fortran*. University of Cambridge, 1992.
- [56] PUCCINI, A. L., *Introdução à Programação Linear*. Livros Técnicos e Científicos Editora, Rio de Janeiro, 1978.
- [57] SCHAFFER, J. D., *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Ph.D. dissertation, Vanderbilt University, 1984.
- [58] SCHAFFER, J. D., *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. D.Sc. Thesis, Department of Electrical Engineering, Vanderbilt University, 1984.
- [59] SHAFFER, R., *Practical Guide to Genetic Algorithms*. Naval Research Laboratory, Chemistry Division. In: <http://chem1.nrl.navy.mil/~shaffer/practga.html>

- [60] SHEWCHUK, J. R., *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Carnegie Mellon University, Pittsburgh, 1994.
- [61] SILVA, T. T., BARBOSA, H. J. C., “Inicialização da População em Algoritmos Genéticos com Codificação Real”. Em: *V Seminário sobre Elementos Finitos e Métodos Numéricos em Engenharia*, UFJF, Juiz de Fora, 1998.
- [62] SPEARS, W. M., DE JONG, K. A., *On the Virtues of Parameterized Uniform Crossover*.
- [63] SPEARS, W. M., *Adapting Crossover in a Genetic Algorithm*. Naval Research Laboratory. Washington, D. C. 20375 USA.
- [64] SRINIVAS, N., DEB, K., “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. In: *Evolutionary Computation*, 2(3): pp. 221-248, 1994.
- [65] VALENZUELA-RENDÓN, M., URESTI-CHARRE, E., “A Non-generational Genetic Algorithm for Multiobjective Optimization”. In: *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 658-665, San Francisco, 1997.
- [66] VELDHUIZEN, D. A., LAMONT, G. B., *Multiobjective Evolutionary Algorithm Test Suites*. Department of Electrical and Computer Engineering. Graduate School of Engineering, Air Force Institute of Technology.

- [67] VELDHUIZEN, D. A. V., *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Dissertation presented to the Faculty of Graduate School of Engineering of the Air Force Institute of Technology, Air University, June 1999.
- [68] YANG, R., DOUGLAS, I., “Simple Genetic Algorithm with Local Tuning: Efficient Global Optimizing Technique”. In: *Journal of Optimization Theory and Applications*, v. 98, n. 2, pp. 449-465, August 1998.
- [69] ZITZLER, E., THIELE, L., *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, TIK-Report No. 43, May 1998.
- [70] ZITZLER, E., THIELE, L., *Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study*. Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich.
- [71] ZITZLER, E., DEB, K., THIELE, L., *Comparison of Evolutionary Algorithms: Empirical Results (Revised Version)*. TIK-Report No. 70. Institut für Technische Informatik und Kommunikationsnetze. Zürich. December 22, 1999.
- [72] ZITZLER, E., *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Dissertation for the degree of Doctor of Technical Sciences. Swiss Federal Institute of Technology Zurich. November 11, 1999.
- [73] ZITZLER, E., In: <http://www.tik.ee.ethz.ch/~zitzler/testdata.html>
- [74] HAYKIN, S., *Neural Networks*. Macmillan, 1994.

[75] WASSERMAN, P., *Neural Computing*. Ven Nostrand, cap. 1-6, 1989.

[76] www.kneehighs.com/intro.html