



**Universidade Estadual de Campinas**  
**Faculdade de Engenharia Elétrica e de Computação**  
**Departamento de Telemática**

# **Sistema Imunológico Artificial para Otimização Multiobjetivo**

**Priscila Cristina Berbert**

**Orientador: Prof. Dr. Akebo Yamakami**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

## **Banca Examinadora**

Prof. Dr. Fernando José Von Zuben	DCA/FEEC/UNICAMP
Prof. Dr. Antônio Carlos Moretti	DMA/IMECC/UNICAMP
Prof. Dr. Leandro Nunes de Castro	UNISANTOS/Santos

Campinas, São Paulo.  
Março de 2008.

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE -  
UNICAMP

B45s Berbert, Priscila Cristina  
Sistema imunológico artificial para otimização  
multiobjetivo / Priscila Cristina Berbert. --Campinas, SP:  
[s.n.], 2008.

Orientador: Akebo Yamakami  
Dissertação (Mestrado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Elétrica e de  
Computação.

1. Sistema imune. 2. Algoritmos evolutivos. 3.  
Otimização matemática. 4. Inteligência artificial. I.  
Yamakami, Akebo. II. Universidade Estadual de Campinas.  
Faculdade de Engenharia Elétrica e de Computação. III.  
Título.

Título em Inglês: Artificial immune system for multiobjective optimization

Palavras-chave em Inglês: Multiobjective optimization, Bio-inspired algorithms,  
Artificial immune systems, Clonal selection

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Fernando José Von Zuben, Antônio Carlos Moretti, Leandro  
Nunes de Castro

Data da defesa: 10/03/2008

Programa de Pós-Graduação: Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE MESTRADO

**Candidata:** Priscila Cristina Berbert

**Data da Defesa:** 10 de março de 2008

**Título da Tese:** "Sistema Imunológico Artificial para Otimização Multiobjetivo"

Prof. Dr. Akebo Yamakami (Presidente):



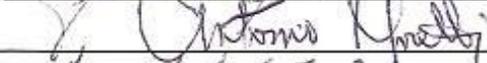
\_\_\_\_\_

Prof. Dr. Leandro Nunes de Castro Silva:



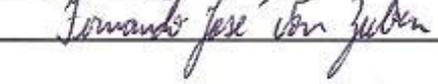
\_\_\_\_\_

Prof. Dr. Antônio Carlos Moretti:



\_\_\_\_\_

Prof. Dr. Fernando José Von Zuben:



\_\_\_\_\_

# Resumo

O objetivo desta dissertação é explorar a utilização de um Sistema Imunológico Artificial, baseado no princípio de Seleção Clonal, na resolução de problemas de Otimização Multiobjetivo.

Os Sistemas Imunológicos Artificiais apresentam, em sua estrutura elementar, as principais características requeridas para a resolução de problemas de Otimização Multiobjetivo: exploração, exploração, paralelismo, elitismo, memória, diversidade, mutação e clonagem proporcionais à afinidade e população dinâmica.

A abordagem proposta utiliza o conceito de Pareto dominância e factibilidade para identificar os anticorpos (soluções) que devem ser clonados. Nos experimentos, foram consideradas algumas situações importantes que podem aparecer nos problemas reais: presença de restrições (lineares e não-lineares) e formato da Fronteira de Pareto (convexa, côncava, contínua, descontínua, discreta, não-uniforme). Na maioria dos problemas, o algoritmo obteve resultados bons e competitivos quando comparados com as propostas da literatura.

**Palavras-chave:** Otimização Multiobjetivo, Algoritmos Bio-inspirados, Sistemas Imunológicos Artificiais, Seleção Clonal.

# Abstract

The aim of this work is to explore an Artificial Immune System, based on the Clonal Selection principle, in the solution of Multiobjective Optimization problems.

Artificial Immune Systems have, in their elementary structure, the main characteristics required to solve Multiobjective Optimization problems: exploration, exploitation, parallelism, elitism, memory, diversity, mutation and proliferation proportional to the affinity, and dynamic repertoire.

The proposed algorithm uses the Pareto dominance concept and feasibility to identify the antibodies (solutions) that must be cloned. In the experiments, some important situations that occurs in real problems were considered: the presence of constraints (linear and non-linear) and Pareto Front format (convex, concave, continuous, discontinuous, discrete, non-uniforme). In the major part of the problems, the algorithm obtains good and competitive results when compared with approaches from the literature.

**Keywords:** Multiobjective Optimization, Bio-inspired Algorithms, Artificial Immune Systems, Clonal Selection.

Aos meus pais...  
Ao meu namorado, Willian...

“Feliz é o homem que acha sabedoria, e o homem que adquire entendimento; pois melhor é o lucro que ela dá do que o lucro da prata, e a sua renda do que o ouro. Mais preciosa é do que as jóias, e nada do que possas desejar é comparável a ela.”  
Provérbios 3:13-15

# Agradecimentos

Desejo expressar minha sincera gratidão...

Ao meu orientador, Professor Akebo Yamakami.

Obrigada pela oportunidade, pela orientação, pelo incentivo e pela motivação.

Aos meus pais, Araci e Alcenir.

Obrigada pelo carinho, pelo amor, pela doação. Sei que isso se tornou uma realização para vocês também.

Ao meu namorado, Willian.

Obrigada pelo amor, pela paciência, pelo carinho, pelas sugestões, pela ajuda e por todos os incentivos. Obrigada por sempre estar ao meu lado... juntos, não há sonhos que não possamos realizar.

À minha avó, Natalina.

Obrigada pelo amor, pela compreensão, pelo incentivo e pelo carinho.

Ao pessoal da FEEC: Márcia, Laura, Tiago, Fabrício, Ricardo e Leonardo.

Obrigada pela ajuda e pelas sugestões.

Aos professores da banca: Professor Antônio Carlos Moretti, Professor Fernando José Von Zuben e Professor Leandro Nunes de Castro.

Obrigada pela atenção e pelas sugestões.

À Unicamp, pela infra-estrutura.

À CAPES, pelo apoio financeiro.

A Deus, sempre e por tudo.

# Índice

<b>Resumo</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Agradecimentos</b> .....	<b>v</b>
<b>Lista de Figuras</b> .....	<b>vii</b>
<b>Lista de Tabelas</b> .....	<b>ix</b>
<b>1 Introdução</b> .....	<b>1</b>
<b>2 Otimização Multiobjetivo</b> .....	<b>5</b>
2.1 Formulação e Conceitos Básicos .....	8
2.2 Métodos de Resolução .....	12
2.2.1 Métodos de Programação Matemática .....	13
2.2.2 Métodos Bio-inspirados .....	18
2.3 Aplicações .....	32
<b>3 Sistemas Imunológicos Artificiais e a Otimização Multiobjetivo</b> .....	<b>35</b>
3.1 Princípios Fundamentais do Sistema Imunológico .....	35
3.1.1 Sistema imune adaptativo .....	36
3.2 Sistemas Imunológicos Artificiais .....	38
3.2.1 Considerações em Otimização Multiobjetivo .....	41
<b>4 O Algoritmo Proposto</b> .....	<b>43</b>
4.1 Apresentação dos componentes e funcionalidades .....	44
4.1.1 Representação .....	44
4.1.2 Inicialização do Repertório .....	45
4.1.3 Medida de afinidade .....	45
4.1.4 Clonagem .....	46
4.1.5 Maturação .....	48
4.1.6 Infactibilidade .....	52
4.1.7 Seleção .....	52
4.1.8 Deleção .....	53
4.1.9 Diversidade .....	55
4.1.10 Memória .....	55
4.1.11 Supressão .....	56
4.1.12 Repertório com tamanho dinâmico .....	57
4.1.13 Maturação Final .....	57
4.2 Pseudocódigo .....	58
<b>5 Experimentos e Resultados</b> .....	<b>61</b>
5.1 Variação dos parâmetros .....	63
5.2 Primeira Etapa de Testes .....	65
5.3 Segunda Etapa de Testes .....	78
5.4 Terceira Etapa de Testes .....	95
5.5 Quarta Etapa de Testes: Espaços de Busca fortemente restritos .....	100
<b>6 Conclusões e Perspectivas Futuras</b> .....	<b>102</b>
<b>Bibliografia</b> .....	<b>104</b>

# Lista de Figuras

<b>Figura 2.1.</b> Exemplo de problema: planejamento de uma estrutura.....	6
<b>Figura 2.2.</b> Fronteira de Pareto: planejamento de uma estrutura.....	7
<b>Figura 2.3.</b> Mapeamento do espaço das soluções no espaço das funções-objetivo.....	8
<b>Figura 2.4.</b> Conjunto das soluções Pareto-ótimas.....	9
<b>Figura 2.5.</b> Dominância de Pareto no espaço objetivo.....	10
<b>Figura 2.6.</b> Exemplo de Fronteiras global e local.....	12
<b>Figura 2.7.</b> Exemplo bidimensional do método da realização das metas.....	15
<b>Figura 2.8.</b> Esquema do VEGA.....	20
<b>Figura 2.9.</b> Ranking por grau de Pareto-dominância do NPGA2.....	22
<b>Figura 2.10.</b> Representação de uma distribuição por hiperplano para $k = 5$ .....	31
<b>Figura 2.11.</b> Distribuição das Aplicações.....	32
<b>Figura 3.1.</b> Estrutura multicamadas do Sistema Imunológico.....	36
<b>Figura 3.2.</b> O Sistema imune adaptativo.....	37
<b>Figura 3.3.</b> SIA: uma ramificação da Inteligência Computacional.....	39
<b>Figura 4.1.</b> Exemplo de seleção de anticorpos para clonagem.....	47
<b>Figura 4.2.</b> Número de clones gerados por anticorpo factível, em função de $\text{rank}(\mathbf{Ab}_i)$ .....	48
<b>Figura 4.3.</b> A direção $\mathbf{d}_A$ , no caso em que o clone $\mathbf{C}_j$ substitui o anticorpo-pai $\mathbf{Ab}_i$ .....	50
<b>Figura 4.4.</b> A direção $\mathbf{d}_B$ , considerando uma solução não-dominada qualquer como líder.....	51
<b>Figura 4.5.</b> Exemplo de clonagem e maturação, para anticorpos factíveis.....	54
<b>Figura 4.6.</b> Exemplo de clonagem e maturação, para anticorpos infactíveis.....	54
<b>Figura 4.7.</b> Continuação do exemplo, resultado da Seleção Clonal.....	55
<b>Figura 4.8.</b> Exemplo bidimensional, cálculo do Crowding Distance do anticorpo $i$ .....	56
<b>Figura 5.1.</b> Verdadeira Fronteira de Pareto - Exemplo 1.....	66
<b>Figura 5.2.</b> Conjunto de soluções não-dominadas obtido - Exemplo 1.....	67
<b>Figura 5.3.</b> Fronteira de Pareto obtida - Exemplo 1.....	67
<b>Figura 5.4.</b> Verdadeira Fronteira de Pareto - Exemplo 2.....	68
<b>Figura 5.5.</b> Fronteira de Pareto obtida - Exemplo 2.....	69
<b>Figura 5.6.</b> Conjunto de soluções não-dominadas obtido - Exemplo 2.....	69
<b>Figura 5.7.</b> Verdadeira Fronteira de Pareto - Exemplo 3.....	70
<b>Figura 5.8.</b> Fronteira de Pareto obtida - Exemplo 3.....	71
<b>Figura 5.9.</b> Verdadeira Fronteira de Pareto - Exemplo 4.....	72

<b>Figura 5.10.</b> Conjunto de soluções não-dominadas obtido - Exemplo 4.....	72
<b>Figura 5.11.</b> Fronteira de Pareto obtida - Exemplo 4.....	73
<b>Figura 5.12.</b> Verdadeira Fronteira de Pareto - Exemplo 5.....	74
<b>Figura 5.13.</b> Conjunto de soluções não-dominadas obtido - Exemplo 5.....	75
<b>Figura 5.14.</b> Fronteira de Pareto obtida - Exemplo 5.....	75
<b>Figura 5.15.</b> Verdadeira Fronteira de Pareto - Exemplo 6.....	76
<b>Figura 5.16.</b> Fronteira de Pareto e soluções não-dominadas obtidos - Exemplo 6.....	77
<b>Figura 5.17.</b> Verdadeira Fronteira de Pareto – CTP1.....	79
<b>Figura 5.18.</b> Fronteira de Pareto obtida - CTP1.....	80
<b>Figura 5.19.</b> Conjunto de soluções não-dominadas obtido - CTP1.....	80
<b>Figura 5.20.</b> Verdadeira Fronteira de Pareto – CTP2.....	81
<b>Figura 5.21.</b> Fronteira de Pareto obtida – CTP2.....	82
<b>Figura 5.22.</b> Conjunto de soluções não-dominadas obtido – CTP2.....	82
<b>Figura 5.23.</b> Verdadeira Fronteira de Pareto – CTP3.....	83
<b>Figura 5.24.</b> Fronteira de Pareto obtida – CTP3.....	84
<b>Figura 5.25.</b> Conjunto de soluções não-dominadas obtido – CTP3.....	85
<b>Figura 5.26.</b> Verdadeira Fronteira de Pareto – CTP4.....	86
<b>Figura 5.27.</b> Fronteira de Pareto obtida – CTP4.....	86
<b>Figura 5.28.</b> Conjunto de soluções não-dominadas obtido – CTP4.....	87
<b>Figura 5.29.</b> Verdadeira Fronteira de Pareto – CTP5.....	88
<b>Figura 5.30.</b> Fronteira de Pareto obtida – CTP5.....	88
<b>Figura 5.31.</b> Conjunto de soluções não-dominadas obtido – CTP5.....	89
<b>Figura 5.32.</b> Verdadeira Fronteira de Pareto – CTP6.....	90
<b>Figura 5.33.</b> Fronteira de Pareto obtida – CTP6.....	90
<b>Figura 5.34.</b> Conjunto de soluções não-dominadas obtido – CTP6.....	91
<b>Figura 5.35.</b> Verdadeira Fronteira de Pareto – CTP7.....	91
<b>Figura 5.36.</b> Fronteira de Pareto obtida – CTP7.....	92
<b>Figura 5.37.</b> Conjunto de soluções não-dominadas obtido – CTP7.....	93
<b>Figura 5.38.</b> Verdadeira Fronteira de Pareto – CTP8.....	94
<b>Figura 5.39.</b> Fronteira de Pareto obtida – CTP8.....	94
<b>Figura 5.40.</b> Conjunto de soluções não-dominadas obtido – CTP8.....	95
<b>Figura 5.41.</b> Fronteira de Pareto obtida – VNT.....	97
<b>Figura 5.42.</b> Fronteiras de Pareto obtidas pelo algoritmo NSGA2 e pelo algoritmo proposto.	99

# Lista de Tabelas

<b>Tabela 3.1.</b> Resumo dos modelos de SIA mais estudados (Dasgupta, 2006).....	39
<b>Tabela 4.1.</b> Parâmetros de Entrada.....	58
<b>Tabela 5.1.</b> Parâmetros comuns da primeira etapa de testes.....	65
<b>Tabela 5.2.</b> Parâmetros específicos - Exemplo 1.....	66
<b>Tabela 5.3.</b> Parâmetros específicos - Exemplo 2.....	68
<b>Tabela 5.4.</b> Parâmetros específicos - Exemplo 3.....	70
<b>Tabela 5.5.</b> Parâmetros específicos - Exemplo 4.....	71
<b>Tabela 5.6.</b> Parâmetros específicos - Exemplo 5.....	74
<b>Tabela 5.7.</b> Parâmetros específicos - Exemplo 6.....	76
<b>Tabela 5.8.</b> Métrica de Dispersão - Algoritmo Proposto.....	78
<b>Tabela 5.9.</b> Métrica de Dispersão - Algoritmo NSGA2.....	78
<b>Tabela 5.10.</b> Parâmetros comuns da segunda etapa de testes.....	78
<b>Tabela 5.11.</b> Parâmetros específicos – CTP1.....	79
<b>Tabela 5.12.</b> Parâmetros específicos – CTP2.....	82
<b>Tabela 5.13.</b> Parâmetros específicos – CTP3.....	84
<b>Tabela 5.14.</b> Parâmetros específicos – CTP4.....	85
<b>Tabela 5.15.</b> Parâmetros específicos – CTP5.....	87
<b>Tabela 5.16.</b> Parâmetros específicos – CTP6.....	89
<b>Tabela 5.17.</b> Parâmetros específicos – CTP7.....	91
<b>Tabela 5.18.</b> Parâmetros específicos – CTP8.....	93
<b>Tabela 5.19.</b> Parâmetros – VNT.....	96
<b>Tabela 5.20.</b> Métrica de Dispersão – VNT.....	96
<b>Tabela 5.21.</b> Parâmetros – WATER.....	97
<b>Tabela 5.22.</b> Parâmetros – Problema fortemente restrito.....	100



# Capítulo 1

## Introdução

Fundamentada em uma vasta área de conhecimento, a otimização se dedica a solucionar problemas independentemente do contexto onde surgem. Problemas nos quais o objetivo é determinar, segundo um critério, a melhor alternativa dentro de um universo de escolhas.

Existem várias classificações possíveis para um problema de otimização:

1. quanto à relação entre as variáveis de decisão (linear ou não-linear);
2. quanto ao valor que as variáveis podem assumir (real, inteira, binária ou mista);
3. quanto à natureza da função-objetivo (convexa ou côncava);
4. quanto à presença de restrições (restrito ou irrestrito);
5. quanto ao número de funções-objetivo (monobjetivo ou multiobjetivo).

Algumas destas classificações apresentam métodos exatos e eficientes de resolução. Outras levarão à necessidade de métodos não-exatos (métodos heurísticos), uma vez que a resolução exata levaria a uma complexidade intratável.

A definição do problema (universo de escolhas, objetivo, limites e restrições), a construção e validação do modelo (representado pelas características mais importantes do problema) e a escolha do método de resolução do modelo são etapas importantes na resolução de problemas de otimização.

Inicialmente restritos a problemas que envolviam um único objetivo, os modelos de otimização e as técnicas associadas evoluíram com o intuito de contemplarem a situação mais realista em que vários objetivos, em geral conflitantes, concorrem para a solução de um determinado problema (Ferreira, 1999b).

Um problema de Otimização Multiobjetivo pode ser definido como a otimização de uma função vetorial, cujos elementos representam cada uma das funções-objetivo. Entretanto, sabe-se que o conjunto das funções vetoriais é um conjunto parcialmente

ordenado. Portanto, podem existir alternativas factíveis que não satisfazem nenhuma relação de ordem, do tipo “ $\leq$ ”, inviabilizando desta forma a utilização do conceito usual de solução ótima adotado em problemas monobjetivos. Assim sendo, a solução do problema dependerá da noção de equilíbrio utilizada para resolver os conflitos que surgem da consideração simultânea de vários objetivos (Ferreira, 1999b).

Na área de Pesquisa Operacional existem várias técnicas de Programação Matemática que foram propostas para a resolução dos problemas de Otimização Multiobjetivo. Porém, muitas dessas técnicas têm alta complexidade e dificuldades no tratamento de certos tipos de problemas. Estas técnicas são fundamentadas em informações sobre o problema, restringindo ou dificultando uma aplicação geral.

Atualmente, existem vários métodos heurísticos que são utilizados para resolver diversos tipos de problemas, como, por exemplo, os Algoritmos Bio-inspirados. Inspirados em fenômenos específicos que ocorrem na natureza, estes algoritmos vêm sendo amplamente utilizados em diversas aplicações, principalmente na área de otimização.

Alguns pesquisadores têm sugerido que a Otimização Multiobjetivo deve ser uma área em que os Algoritmos Bio-inspirados têm performance melhor do que outras estratégias de busca (Toro *et. al.*, 2006).

Dentre os Algoritmos Bio-inspirados utilizados na resolução de problemas de Otimização Multiobjetivo, destacam-se: Algoritmos Genéticos, Sistemas Imunológicos Artificiais e Enxame de Partículas (Seção 2.2.2). Algumas características destes métodos merecem ser destacadas: flexibilidade, exploração, exploração, paralelismo, capacidade de escapar de ótimos locais, capacidade de lidar com problemas complexos para os quais não é possível (ou é difícil) obter uma descrição detalhada, além de serem menos susceptíveis à forma ou continuidade das funções.

Mais especificamente, o campo de pesquisa em Sistemas Imunológicos Artificiais vem crescendo e se tornando um atrativo para uma grande variedade de modelos, técnicas e aplicações (Dasgupta, 2006); surgiu a partir de tentativas de modelar e aplicar princípios imunológicos no desenvolvimento de novas ferramentas computacionais: reconhecimento de padrões, detecção de faltas e anomalias, segurança computacional, otimização, controle, robótica, scheduling, análise de dados, aprendizagem de máquina, processamento de informações e resolução de problemas.

Os Sistemas Imunológicos Artificiais apresentam, em sua estrutura elementar, as principais características requeridas para a resolução de problemas de Otimização Multiobjetivo: elitismo, memória, diversidade, mutação e clonagem proporcionais à afinidade e repertório dinâmico. A importância e o papel destas características na resolução de problemas de Otimização Multiobjetivo serão evidenciados nos Capítulos 3 e 4.

## Objetivo

O objetivo desta dissertação é explorar a utilização de um Sistema Imunológico Artificial na resolução de problemas de Otimização Multiobjetivo, levando em consideração algumas situações importantes que podem aparecer nos problemas reais: presença de restrições (lineares e não-lineares) e formato da fronteira de Pareto (convexa, côncava, contínua, descontínua, discreta, multimodal, deceptiva, não-uniforme).

Serão realizados experimentos com diferentes funções de teste, comparando-os com os resultados obtidos por outros algoritmos existentes.

## Organização da dissertação

Este trabalho foi organizado da seguinte maneira:

- No Capítulo 2 são apresentados os conceitos de Otimização Multiobjetivo utilizados ao longo do estudo e algumas técnicas de resolução (tradicional e alternativas).
- No Capítulo 3, uma breve apresentação do Sistema Imunológico Biológico é realizada, discute-se sobre os Sistemas Imunológicos Artificiais e suas aplicações na engenharia, com ênfase em Otimização Multiobjetivo.
- No Capítulo 4, propõe-se a implementação de um Sistema Imunológico Artificial para Otimização Multiobjetivo.

- O Capítulo 5 apresenta alguns problemas de Otimização Multiobjetivo, a resolução destes problemas através do algoritmo proposto, juntamente com as comparações e discussões dos resultados.
- Finalmente, as conclusões do trabalho e as perspectivas de estudos complementares são apresentadas no Capítulo 6.

## Capítulo 2

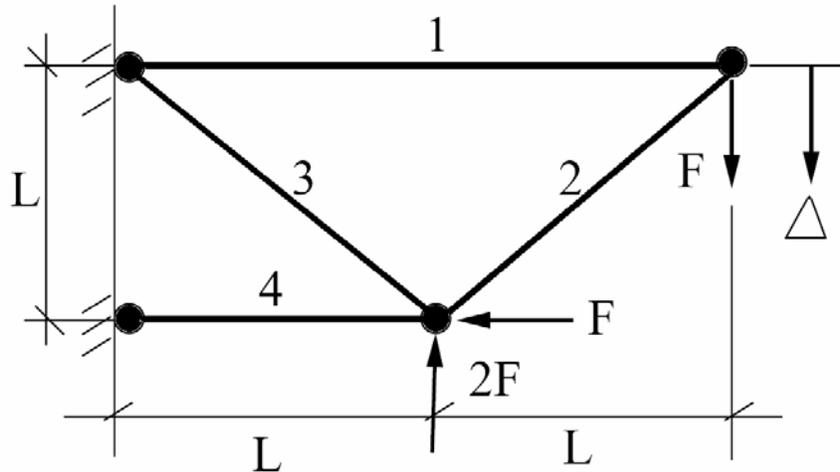
# Otimização Multiobjetivo

Muitos problemas do mundo real apresentam um conjunto de objetivos a serem otimizados. Na maioria das vezes estes objetivos são conflitantes, ou seja, a melhoria em algum(uns) objetivo(s) causa(m) deterioração em outro(s).

Atualmente, o enfoque Multiobjetivo encontra aplicações em qualquer área: problemas de planejamento de produção, transporte, alocação, gerenciamento de recursos (elétricos, hídricos, etc.), dentre outros.

Um exemplo de aplicação multiobjetivo é o problema de despacho ótimo de potências ativa e reativa (Amorim *et. al.*, 2006). Além da minimização do custo operacional da geração de potência ativa, é de fundamental importância a maximização da eficiência operacional, da qualidade e da confiabilidade dos serviços oferecidos pelas concessionárias de energia elétrica. Estes objetivos são conflitantes. A operação dos sistemas de potência a custo mínimo, força-os a operar em seus limites físicos e operacionais. Uma solução que apresente menor custo operacional, mas com limites físicos violados e baixa confiabilidade, não pode ser considerada como superior (ou inferior) a outra solução com maior custo operacional e que atenda as restrições físicas e possua alta confiabilidade. No entanto, dentre todas as possíveis soluções existem algumas que são superiores a outras, ou seja, apresentam custo menor ou igual por uma confiabilidade maior ou equivalente. Assim, as *soluções não-dominadas* superam outras, enquanto que as *soluções dominadas* são superadas por pelo menos uma outra solução.

Outro exemplo de aplicação apresentado em Coello (2001) é o planejamento de uma estrutura formada por quatro vigas, ilustrada na Figura 2.1. Duas funções-objetivo foram identificadas: o volume e o deslocamento da viga devem ser minimizados.



**Figura 2.1.** Exemplo de problema: planejamento de uma estrutura.

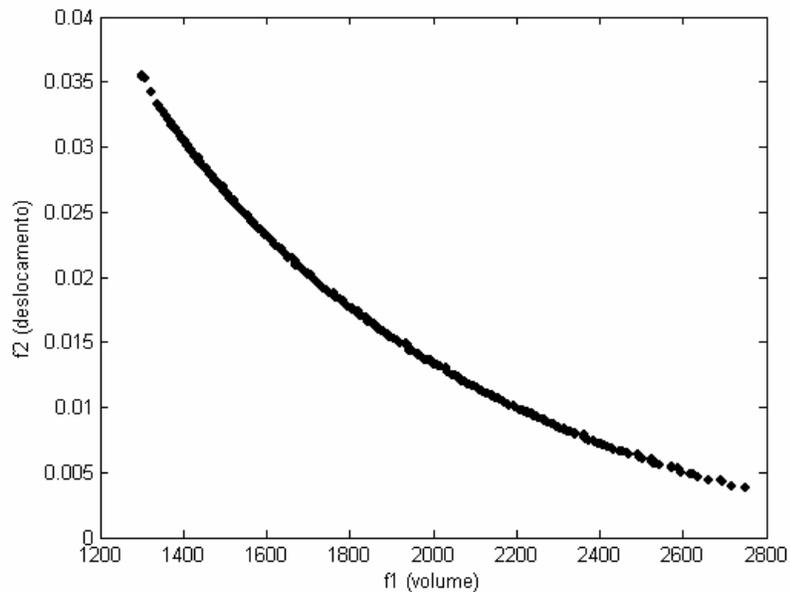
Considerando:

- $x_i$  a área da viga  $i$  (em  $\text{cm}^2$ , seção transversal, perpendicular ao comprimento).
- $f_1(\mathbf{x})$  o volume total da estrutura (em  $\text{cm}^3$ );
- $f_2(\mathbf{x})$  o deslocamento total da estrutura (em cm);

Matematicamente, o problema pode ser formulado da seguinte maneira:

minimizar	{	$f_1(\mathbf{x}) = L(2x_1 + \sqrt{2}x_2 + \sqrt{2}x_3 + x_4)$ $f_2(\mathbf{x}) = \frac{FL}{E} \left( \frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4} \right)$	
s.a.		$\left( \frac{F}{\rho} \right) \leq x_1 \leq 3 \left( \frac{F}{\rho} \right)$ $\sqrt{2} \left( \frac{F}{\rho} \right) \leq x_2 \leq 3 \left( \frac{F}{\rho} \right)$ $\sqrt{2} \left( \frac{F}{\rho} \right) \leq x_3 \leq 3 \left( \frac{F}{\rho} \right)$ $\left( \frac{F}{\rho} \right) \leq x_4 \leq 3 \left( \frac{F}{\rho} \right)$	$\mathbf{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]$ $F = 10 \text{ kN}$ $E = 2 \cdot 10^5 \text{ kN/cm}^2$ $L = 200 \text{ cm}$ $\rho = 10 \text{ kN/cm}^2$

A Figura 2.2 apresenta a Fronteira de Pareto<sup>1</sup> global (solução do problema apresentada no espaço das funções):



**Figura 2.2.** Fronteira de Pareto: planejamento de uma estrutura.

Quanto menor o volume das vigas (minimização de  $f_1$ ), maior a ductibilidade da estrutura e maior o deslocamento da estrutura. Por outro lado, um menor deslocamento (minimização de  $f_2$ ) exige uma estrutura mais rígida e, conseqüentemente, maior o volume das vigas utilizadas. Portanto, os objetivos são conflitantes.

Desta maneira, existe a necessidade do desenvolvimento de técnicas e procedimentos que permitam o tratamento simultâneo de todos os objetivos do problema. Esta é a tarefa da Otimização Multiobjetivo.

Dado que um problema de Otimização Multiobjetivo pode apresentar infinitas soluções não comparáveis entre si, torna-se evidente a necessidade de critérios adicionais para se chegar a uma solução final. Estes critérios são fornecidos por um decisor, que busca selecionar uma solução que proporcione uma relação de compromisso adequada entre os objetivos do problema (Ferreira, 1999b).

---

<sup>1</sup> A Fronteira da Pareto deste problema foi obtida através do algoritmo proposto nesta dissertação.

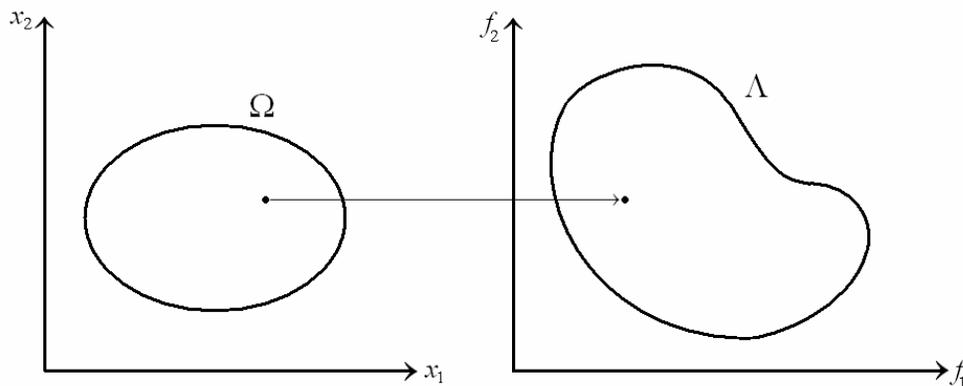
## 2.1 Formulação e Conceitos Básicos

Matematicamente, um problema Otimização Multiobjetivo pode ser definido como o problema de encontrar um vetor de variáveis de decisão  $\mathbf{x}$  que satisfaça as restrições  $\Omega = \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{G}(\mathbf{x}) \leq 0, \mathbf{H}(\mathbf{x}) = 0, \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}$  e otimize<sup>2</sup> a função vetorial  $\mathbf{F}(\mathbf{x})$ , cujos elementos representam as funções-objetivo (Osyczka, 1985).

$$\begin{array}{ll} \text{minimizar} & \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \quad m \geq 2 \\ \text{s.a.} & \mathbf{x} \in \Omega. \end{array}$$

Também,  $\Lambda = \{\mathbf{y} \in \mathcal{R}^m \mid \mathbf{y} = \mathbf{F}(\mathbf{x})_{\mathbf{x} \in \Omega}\}$  é definida como a região factível correspondente para o espaço das funções-objetivo.

A Figura 2.3 ilustra um exemplo de mapeamento do espaço das soluções no espaço das funções-objetivo no caso bidimensional.



**Figura 2.3.** Mapeamento do espaço das soluções no espaço das funções-objetivo.

Note que, por  $\mathbf{F}(\mathbf{x})$  ser um vetor, se quaisquer componentes de  $\mathbf{F}(\mathbf{x})$  competirem, não existirá uma solução única para o problema e sim um conjunto de soluções de

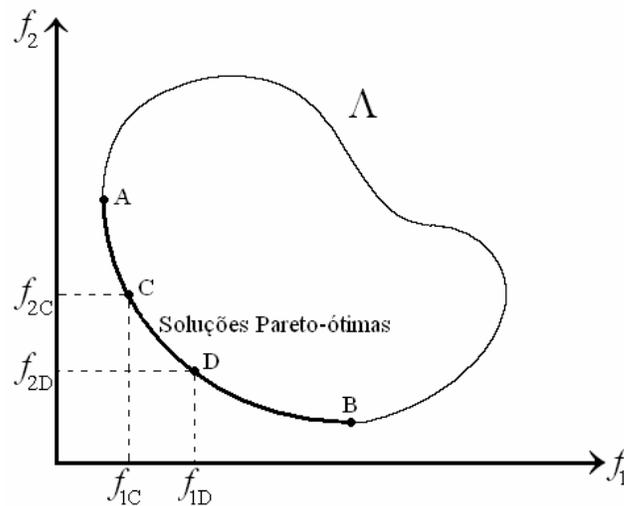
<sup>2</sup> Serão apresentadas definições para o problema de minimização; definições para o problema de maximização são análogas.

compromisso (trade-offs). Desta maneira, o conceito de *Pareto<sup>3</sup>-otimalidade* deve ser utilizado para a caracterização e obtenção das soluções.

Uma solução Pareto-ótima (ou *não-dominada, não-inferior, eficiente*) é aquela que um acréscimo em um dos objetivos resulta em degradação de outro(s) (Ferreira, 1999a).

No exemplo bidimensional da Figura 2.4 o conjunto de soluções Pareto-ótimas está representado entre os pontos A e B.

Exemplo: C e D são soluções Pareto-ótimas. Uma melhora no objetivo  $f_1$  resulta em degradação do objetivo  $f_2$ , isto é,  $f_{1C} < f_{1D}$  e  $f_{2C} > f_{2D}$ .



**Figura 2.4.** Conjunto das soluções Pareto-ótimas.

Quaisquer outros pontos de  $\Omega$ , que não sejam Pareto-ótimos, representam soluções em que uma melhora pode ser obtida em todos os objetivos; é claro que estes pontos não têm nenhum valor. A Otimização Multiobjetivo visa, portanto, a obtenção e seleção das soluções Pareto-ótimas. As técnicas de Otimização Multiobjetivo são várias e na seção subsequente algumas delas serão discutidas.

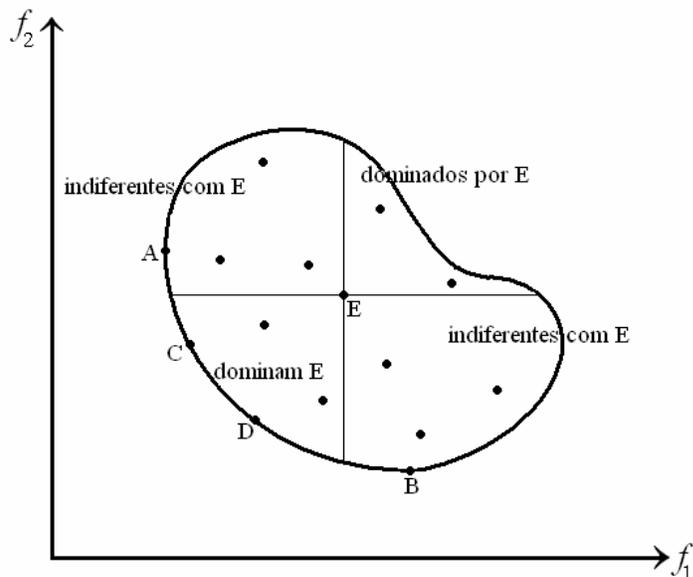
Nos problemas de Otimização Multiobjetivo, otimizar significa encontrar todos os valores aceitáveis para as funções-objetivo para uma tomada de decisão. O espaço de busca

<sup>3</sup> A noção de ‘ótimo’ foi originalmente proposta por Francis Ysidro Edgeworth em 1881. Esta noção foi generalizada por Vilfredo Pareto em 1896. Embora alguns autores utilizem a denominação de *Edgeworth-Pareto-otimalidade*, nesta dissertação, será utilizado o termo *Pareto-otimalidade*.

$\Omega$  é parcialmente ordenado no sentido de que duas soluções arbitrárias são relacionadas de duas possíveis maneiras: ou uma domina a outra ou nenhuma delas domina (Figura 2.5).

Exemplo: Se  $\mathbf{F} : \Omega \rightarrow \mathfrak{R}^2$  é a função vetorial de um problema de minimização, existem três possibilidades:

- 1) Sejam  $\mathbf{a}$  e  $\mathbf{b} \in \Omega$ , tais que,  $\mathbf{F}(\mathbf{a}) = (2, 7)$  e  $\mathbf{F}(\mathbf{b}) = (4, 9)$ .  
Então  $\mathbf{F}(\mathbf{a}) \preceq \mathbf{F}(\mathbf{b})$  ( $\mathbf{a}$  domina  $\mathbf{b}$ ).
- 2) Sejam  $\mathbf{c}$  e  $\mathbf{d} \in \Omega$ , tais que,  $\mathbf{F}(\mathbf{c}) = (5, 8)$  e  $\mathbf{F}(\mathbf{d}) = (1, 6)$ .  
Então  $\mathbf{F}(\mathbf{c}) \succeq \mathbf{F}(\mathbf{d})$  ( $\mathbf{c}$  dominado por  $\mathbf{d}$ ).
- 3) Sejam  $\mathbf{i}$  e  $\mathbf{j} \in \Omega$ , tais que,  $\mathbf{F}(\mathbf{i}) = (3, 9)$  e  $\mathbf{F}(\mathbf{j}) = (5, 7)$ .  
Então  $\mathbf{F}(\mathbf{i}) \not\preceq \mathbf{F}(\mathbf{j})$  e  $\mathbf{F}(\mathbf{j}) \not\preceq \mathbf{F}(\mathbf{i})$  ( $\mathbf{i}$  e  $\mathbf{j}$  são indiferentes).



**Figura 2.5.** Dominância de Pareto no espaço objetivo.

Este conceito pode ser formalizado da seguinte forma:

**Definição 2.1 (Dominância de Pareto):** Sejam  $\mathbf{a}$  e  $\mathbf{b} \in \Omega$ .  $\mathbf{a}$  domina  $\mathbf{b}$ ,  $\mathbf{F}(\mathbf{a}) \preceq \mathbf{F}(\mathbf{b})$  (problema de minimização), se e somente se:

$$\forall i \in \{1, 2, \dots, m\}, f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \quad \text{e} \quad \exists j \in \{1, 2, \dots, m\}, f_j(\mathbf{a}) < f_j(\mathbf{b}).$$

Em outras palavras, “**a** não é pior que **b** em nenhum dos objetivos e é melhor em pelo menos um”.

**Definição 2.2 (Otimalidade de Pareto):** Uma solução  $\mathbf{x}^*$  é Pareto-ótima se e somente se  $\mathbf{x}^*$  é não-dominada em relação à  $\Omega$ , ou seja, nenhum vetor do espaço de busca domina  $\mathbf{x}^*$ .

A solução esperada é composta por um conjunto de pontos de equilíbrio, isto é, uma família de soluções consideradas iguais entre si e superiores em relação ao restante das soluções.

**Definição 2.3 (Conjunto Pareto-ótimo):** O Conjunto Pareto-ótimo ( $P^*$ ) é definido como:

$$P^* = \{ \mathbf{x} \in \Omega \mid \neg \exists \mathbf{y} \in \Omega, \mathbf{F}(\mathbf{y}) \preceq \mathbf{F}(\mathbf{x}) \}.$$

**Definição 2.4 (Fronteira de Pareto):** Dado um Conjunto Pareto-ótimo ( $P^*$ ), a Fronteira de Pareto ( $FP^*$ ) pode ser definida da seguinte forma:

$$FP^* = \{ \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \mid \mathbf{x} \in P^* \}.$$

Da mesma maneira que nos problemas de otimização monobjetivos, existe a possibilidade de problemas de Otimização Multiobjetivo apresentarem soluções Pareto-ótimas locais e globais. Em alguns casos, grande parte das soluções é atraída para as Fronteiras de Pareto locais. A Figura 2.6 ilustra a diferença entre uma Fronteira de Pareto global e uma local.

**Definição 2.5 (Conjunto Pareto-ótimo local):** Se para todo  $\mathbf{x}$ , membro de um conjunto  $\mathbf{P}$ , não existir solução  $\mathbf{y}$ ,  $\|\mathbf{y} - \mathbf{x}\|_\infty \leq \varepsilon$ , que domine qualquer solução pertencente a  $\mathbf{P}$ , então as soluções pertencentes ao conjunto  $\mathbf{P}$  constituem um Conjunto Pareto-ótimo local.  $\varepsilon$  é número positivo pequeno e  $\mathbf{y}$  é obtido através de uma perturbação numa pequena vizinhança de  $\mathbf{x}$  (Deb, 1999).

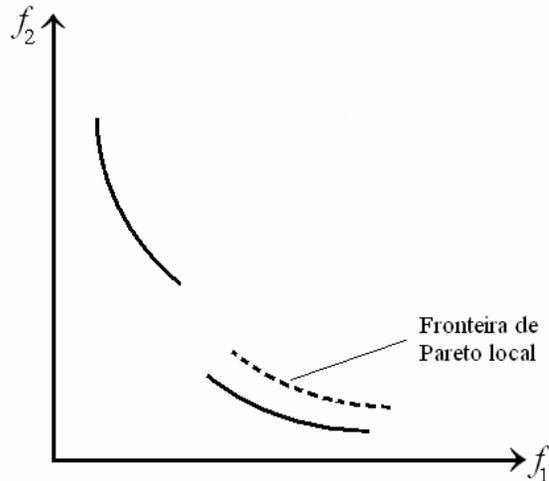


Figura 2.6. Exemplo de Fronteiras global e local.

**Definição 2.6 (Conjunto Pareto-ótimo global):** Se não existir solução no espaço de busca que domine qualquer membro de um conjunto  $\mathbf{P}$ , então as soluções pertencentes a  $\mathbf{P}$  constituem o Conjunto Pareto-ótimo global (Deb, 1999).

O tamanho e formato da Fronteira de Pareto geralmente dependem do número de funções-objetivo e de suas interações. Se as funções-objetivo são conflitantes, a Fronteira de Pareto pode ser maior e mais espalhada do que se as funções-objetivo fossem cooperativas<sup>4</sup>. Entretanto, na maioria dos problemas de otimização multiobjetivo (os de aplicações reais e os estudados na literatura), as funções-objetivo são conflitantes e geralmente resultam em uma Fronteira de Pareto (local ou global) com várias soluções (Deb, 1999).

## 2.2 Métodos de Resolução

Desde 1896, quando o conceito de Pareto-ótimo foi introduzido, várias técnicas para resolução de problemas de Otimização Multiobjetivo, tradicionais (Programação Matemática) ou alternativas (Algoritmos Genéticos, Sistemas Imunológicos Artificiais,

---

<sup>4</sup> Analisando individualmente, se duas funções-objetivo têm soluções ótimas similares e valores das funções também similares, elas estão cooperando. Soluções conflitantes são aquelas onde as funções-objetivo têm soluções ótimas e valores de funções o mais diferente possível.

Enxame de Partículas), têm sido desenvolvidas. O interesse é que os métodos apresentem as soluções não-dominadas bem distribuídas pela Fronteira de Pareto (global), facilitando o conhecimento do problema e a escolha da solução mais adequada pelo decisor.

A seguir, uma breve revisão dos métodos de Programação Matemática e dos Métodos Bio-inspirados é apresentada.

## 2.2.1 Métodos de Programação Matemática

Os métodos de Programação Matemática podem ser classificados de acordo com o momento em que o decisor aplica seus critérios (Ferreira, 1999a): *Métodos a-Priori*, *Métodos a-Posteriori* e *Métodos Interativos*. Com a utilização desses métodos são necessárias atribuições de pesos ou constantes que apontam para uma solução única, sem a certeza de que seria a mais indicada.

### 2.2.1.1 Métodos a-Priori

Nos Métodos a-Priori, as preferências do decisor são definidas antes da resolução. O problema multiobjetivo original é transformado em problemas mais simples, que podem ser resolvidos através de métodos conhecidos. A desvantagem desses métodos é que características desejáveis a uma solução são raramente conhecidas antecipadamente.

- **Método Lexicográfico:** o decisor deve listar os objetivos em ordem decrescente de importância  $\{f_1, f_2, \dots, f_m\}$ . A primeira função-objetivo da lista é otimizada no espaço de busca e este resultado é considerado como uma restrição na otimização da segunda função; e assim por diante. Generalizando:

minimizar	$f_j(\mathbf{x})$
s.a.	$\mathbf{x} \in \Omega_j$

onde  $\Omega_j = \{\mathbf{x} \mid \mathbf{x} \in \Omega_{j-1}, \mathbf{x} = \operatorname{argmin}(f_{j-1})\}$  e  $\Omega_0 = \Omega$ .

Uma desvantagem é que a solução final é muito sensível à ordenação dos objetivos.

- **Método baseado em Limitantes:** o decisor deve seleccionar um objetivo de preferência  $f_i$ ,  $i \in \{1, 2, \dots, m\}$ , e indicar limitantes para os demais objetivos  $f_j$ ,  $j \in \{1, 2, \dots, m\}$ ,  $i \neq j$ . Formalmente:

$$\begin{array}{ll} \text{minimizar} & f_i(\mathbf{x}) \\ \text{s.a.} & l_j \leq f_j \leq L_j \quad \forall j \neq i \\ & \mathbf{x} \in \Omega \end{array}$$

Especificar  $l_j$  e  $L_j$ , de modo que o problema continue factível e que torne a solução satisfatória ao decisor, pode ser difícil.

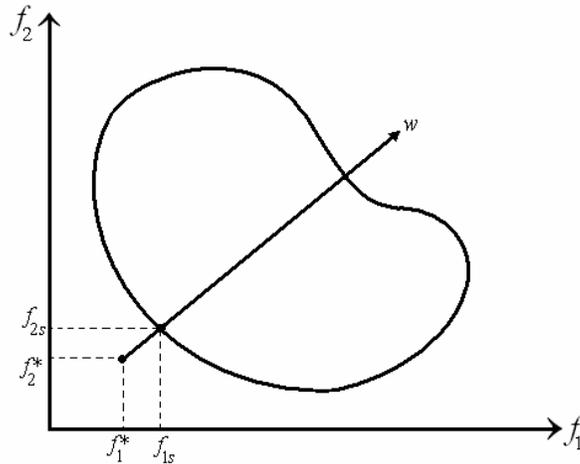
- **Método da realização das metas:** Associa-se o conjunto de objetivos  $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]$  a um conjunto de metas  $\mathbf{F}^* = [f_1^*, f_2^*, \dots, f_m^*]$ . A formulação do problema permite que os objetivos se encontrem acima ou abaixo das metas, esta imprecisão (ou folga) é controlada por um vetor de coeficientes de peso  $\mathbf{w} = [w_1, w_2, \dots, w_m]$ .

$$\begin{array}{ll} \text{minimizar} & \gamma \\ \text{s.a.} & f_i(\mathbf{x}) - w_i \gamma \leq f_i^* \\ & \mathbf{x} \in \Omega \\ & \gamma \in \mathfrak{R} \end{array}$$

O método da realização das metas é representado geometricamente em duas dimensões pela Figura 2.7.

$[f_1^*, f_2^*]$  define o ponto meta e o vetor  $\mathbf{w} = [w_1, w_2]$  define a direção de busca do ponto-meta ao espaço factível das funções ( $\Lambda$ ).

A solução (única) do problema,  $[f_{1s}, f_{2s}]$ , não é necessariamente Pareto-ótima. Soluções Pareto-ótimas são geralmente obtidas quando as metas não são atingidas.



**Figura 2.7.** Exemplo bidimensional do método da realização das metas.

### 2.2.1.2 Métodos a-Posteriori

Nos Métodos a-Posteriori, busca-se gerar o Conjunto Pareto-ótimo (ou parte dele) para que depois, o decisor escolha a solução de compromisso.

- **Método das ponderações:** Este método baseia-se na transformação do problema multiobjetivo em problemas monoobjetivo  $P(\mathbf{w})$ , com a variação do vetor  $\mathbf{w} \in W = \{\mathbf{w} : \mathbf{w} \in \mathbb{R}^m, w_i \geq 0 \text{ e } \sum_{i=1}^m w_i = 1\}$ . Se  $f_1, f_2, \dots, f_m$  são convexas<sup>5</sup>, dado um vetor de pesos  $\mathbf{w}$ , o método das ponderações consiste na resolução do problema:

$\begin{array}{ll} \text{minimizar} & \sum_{i=1}^m w_i f_i(\mathbf{x}) \\ \text{s.a.} & \mathbf{x} \in \Omega \end{array}$
--

O método é útil na geração de subconjuntos de soluções de acordo com as “preferências” indicadas pelo vetor de pesos  $\mathbf{w}$ ; a geração de todo o conjunto Pareto-ótimo é inviável.

<sup>5</sup> No caso de funções não-convexas, algumas porções do conjunto Pareto-ótimo podem não ser geradas.

- **Método das  $\varepsilon$ -restrições:** baseado na minimização do objetivo de maior prioridade, transformando os outros objetivos em restrições de desigualdade. O método consiste na resolução de problemas  $P(\varepsilon)$ ,  $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m]$ :

$$\begin{array}{ll} \text{minimizar} & f_i(\mathbf{x}) \\ \text{s.a.} & f_j \leq \varepsilon_j \quad \forall j \neq i \\ & \mathbf{x} \in \Omega \end{array}$$

Variando convenientemente os limitantes  $\varepsilon_i$ , é possível gerar o conjunto Pareto-ótimo mesmo para  $\Lambda$  não-convexo (caso em que o método das Ponderações pode falhar). Uma dificuldade encontrada é a definição de valores para  $\varepsilon_i$  que garanta uma solução factível.

- **Método Simplex Multiobjetivo:** este método é restrito à resolução de problemas lineares do tipo:

$$\begin{array}{ll} \text{minimizar} & \mathbf{C}\mathbf{x} \\ \text{s.a.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \quad \mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_m \end{bmatrix} \in \mathfrak{R}^{m \times n} \quad \begin{array}{l} \mathbf{c}_i \in \mathfrak{R}^{1 \times n} \\ \mathbf{x} \in \mathfrak{R}^n \\ \mathbf{A} \in \mathfrak{R}^{l \times n} \\ \mathbf{b} \in \mathfrak{R}^l \end{array}$$

Através deste método, os objetivos podem ser tratados simultaneamente, sem a necessidade de utilização de técnicas de escalarização. A região factível  $\Omega$  tem a forma poliédrica, portanto, um número finito de ponto extremos. Assim o método Simplex Multiobjetivo “caminha” de um ponto extremo Pareto-ótimo a outro, até que todo o Conjunto de Pareto (formado por pontos extremos Pareto-ótimo e faces Pareto-ótimas) esteja caracterizado.

### 2.2.1.3 Métodos Interativos

Estes métodos são mais elaborados do que os apresentados anteriormente. Nos métodos interativos, informações sobre preferências são passadas para o decisor durante a

resolução do problema. O decisor especifica e ajusta suas preferências ao mesmo tempo em que aprende (ele não é capaz de indicar preferências a-priori devido à complexidade do problema).

Algumas vantagens são: Não existe a necessidade de conhecimento a-priori sobre o problema; por ser um processo de aprendizagem, o decisor pode entender melhor o problema; o decisor faz parte da busca logo, a solução pode ser aceita com maior possibilidade. Por outro lado, a solução depende do comportamento do decisor e da maneira que ele gerencia suas preferências.

- **Método de Geoffrion, Dyer e Feinberg:** Este método é destinado a resolução de problemas convexos, do tipo:

$$\begin{array}{ll} \text{minimizar} & U[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \\ \text{s.a.} & \mathbf{x} \in \Omega \end{array}$$

Considerando:  $\Omega$  convexo;

$[f_1, f_2, \dots, f_m]$  funções convexas sobre  $\Omega$ ;

$U : \Lambda \rightarrow \mathfrak{R}$ , função convexa crescente sobre  $\Lambda$ , que reflete as preferências do decisor em relação aos  $m$  objetivos. Como  $U$  não é totalmente conhecida, algumas informações devem ser fornecidas pelo decisor durante o processo.

- **Método STEM:** Este método é um procedimento iterativo destinado à resolução de problemas lineares:

$$\begin{array}{ll} \text{minimizar} & \mathbf{C}\mathbf{x} \\ \text{s.a.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \quad \mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_m \end{bmatrix} \in \mathfrak{R}^{m \times n} \quad \begin{array}{l} \mathbf{c}_i \in \mathfrak{R}^{1 \times n} \\ \mathbf{x} \in \mathfrak{R}^n \\ \mathbf{A} \in \mathfrak{R}^{l \times n} \\ \mathbf{b} \in \mathfrak{R}^l \end{array}$$

Inicia-se, com a resolução de  $m$  problemas lineares, formados por cada um dos objetivos separadamente.

O decisor deve classificar cada solução encontrada (valores das funções-objetivo) como satisfatória ou insatisfatória, quando comparada com uma solução utópica. Se todos os objetivos forem satisfatórios, o algoritmo é encerrado. Caso contrário, um ou mais objetivos, classificados como satisfatórios, devem ser relaxados, com o intuito de melhorar os valores dos outros objetivos.

## 2.2.2 Métodos Bio-inspirados

Atualmente, existem mais de 30 técnicas de Programação Matemática para resolução de problemas de Otimização Multiobjetivo. Entretanto, estas técnicas tendem a gerar somente um elemento do Conjunto Pareto-ótimo a cada rodada (Coello, 2002a).

Considerando que a solução de um problema de Otimização Multiobjetivo é constituída por um conjunto de pontos, a utilização de Algoritmos Bio-inspirados parece imediata, pois considera simultaneamente um conjunto de soluções. Isto permite encontrar vários pontos do Conjunto Pareto-ótimo em uma única rodada do algoritmo (paralelismo).

Outro grande potencial desses algoritmos é a integração da ampla exploração do espaço de busca com um processo de busca mais localizada (exploração) resultando em um alto grau de robustez, que permite sua aplicação em vários problemas práticos, junto aos quais outras estratégias de solução se mostram inócuas (Bäck *et. al.*, 2000a; Bäck *et. al.*, 2000b).

Além disso, os problemas reais têm se tornado cada vez mais complexos, sem funções definidas ou bem comportadas e na maioria das vezes descontínuas e com domínios não-convexos, dificultando a utilização de métodos exatos na resolução. Adicionalmente, os Algoritmos Bio-inspirados são menos susceptíveis à forma ou à continuidade da Fronteira de Pareto (Coello e Cortés, 2005).

As técnicas tradicionais exigem o conhecimento prévio do problema, uma especificação detalhada ou a indicação de preferências, o que quase sempre não é possível. Já os Algoritmos Bio-inspirados requerem apenas a descrição aproximada das características que representem o comportamento global, como uma função (ou medida) de afinidade, adaptabilidade ou desempenho.

Estas são algumas das razões que têm impulsionado cada vez mais a utilização dos Métodos Bio-inspirados nas aplicações, somando-se sua flexibilidade, generalidade, robustez e a principal característica: a possibilidade de escapar de ótimos locais para encontrar as soluções globais desejadas.

Os Métodos Bio-inspirados podem ser classificados como métodos a-posteriori, visando a geração de pontos uniformemente distribuídos que representem o Conjunto Pareto-ótimo.

Nas próximas subseções serão discutidos alguns dos principais algoritmos propostos na literatura. É importante ressaltar que todos os algoritmos representam boas metodologias de resolução, com diferentes características. As vantagens e desvantagens da utilização de cada abordagem serão brevemente apresentadas.

### 2.2.2.1 Algoritmos Genéticos

Os algoritmos Genéticos têm demonstrado bom desempenho na resolução de problemas de Otimização Multiobjetivo e nos últimos anos várias abordagens foram apresentadas:

- **VEGA (Vector Evaluated Genetic Algorithm):** O primeiro trabalho nesta área foi proposto por Schaffer (1984) na apresentação do algoritmo VEGA. VEGA se diferencia de um simples algoritmo genético somente pelo modo como a seleção é realizada. O desempenho dos  $n$  indivíduos da população é avaliado em cada função-objetivo (separadamente). Um indivíduo é formado pelo código genético (binário) e pelo seu desempenho em cada uma das funções. Através de seleção proporcional ao fitness (*Roulette Wheel*),  $m$  sub-populações (uma para cada função) são formadas. A nova população é constituída pela junção das sub-populações e aplicação dos operadores de crossover e mutação (Figura 2.8). VEGA não incorpora diretamente o conceito de Pareto-ótimo. Algumas desvantagens do algoritmo são: tendência em selecionar indivíduos com grande desempenho em apenas um dos objetivos, ignorando indivíduos com

desempenho médio em todos os objetivos (isso impede uma boa caracterização do conjunto Pareto-ótimo); é incapaz de produzir certas porções da Fronteira de Pareto; não tem mecanismo explícito para manutenção de diversidade; não necessariamente produz indivíduos globalmente não-dominados.

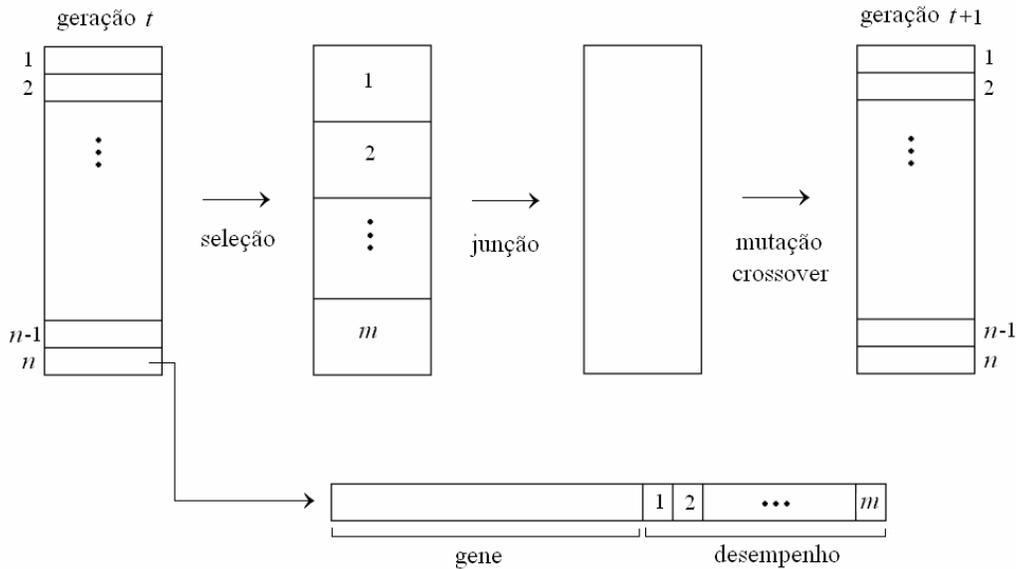


Figura 2.8. Esquema do VEGA.

- **MOGA (Multi-Objective Genetic Algorithm):** Esta abordagem, proposta por Fonseca e Fleming (1993), foi a primeira a considerar o conceito de dominância na classificação das soluções.

A classificação de um determinado indivíduo é igual a um mais o número de indivíduos da população corrente pelos quais ele é dominado. Dessa forma, todos os indivíduos não-dominados recebem classificação (*rank*) igual a um e a classificação dos indivíduos dominados não excede o tamanho da população.

A seguir, a população é ordenada de acordo com a classificação recebida. Para cada solução é atribuído um *fitness* preliminar (*raw fitness*) usando uma função linear ou exponencial. O *fitness* médio das soluções que possuem a mesma classificação é calculado. Isto garante que soluções com melhor classificação possuam *fitness* melhores.

Para manter a diversidade das soluções utilizou-se, para cada classificação, o método de formação de nichos (*niching*). Após a obtenção dos nichos, calcula-se o *fitness* compartilhado (*fitness sharing*) de cada solução, de acordo com a ordem de classificação. Desta forma as soluções que residem em um nicho menos ocupado terão melhor *fitness* compartilhado. Este método permite que soluções pouco representadas em cada ranking se destaquem. Por outro lado o *fitness* de compartilhamento não garante que as soluções de maior classificação terão pior desempenho que soluções de menor classificação. Os operadores de seleção, cruzamento e mutação são os usuais.

- **NPGA (Niched Pareto Genetic Algorithm):** Este algoritmo, apresentado por Horn *et. al.* (1994), apresenta um esquema de seleção por torneio baseado em Pareto dominância (considerando que a relação de dominância estabelece uma ordem parcial) e emprega o compartilhamento de *fitness* (*fitness sharing*) para os casos em que o torneio encontra não-dominância.

Os torneios obedecem aos seguintes critérios: dois indivíduos e um conjunto de comparação (em torno de 10% da população) são selecionados aleatoriamente. O vencedor é o indivíduo que não for dominado pelo conjunto de comparação. Se houver empate (ambos dominados ou ambos não-dominados), a análise passa a focar o número de indivíduos presentes no nicho em que se encontra cada indivíduo candidato. Vence o indivíduo que pertencer ao nicho menos populoso.

Uma dificuldade é que o NPGA é muito sensível ao tamanho do conjunto de comparação do torneio: se o tamanho for muito grande, a população irá convergir prematuramente para apenas uma parte da Fronteira de Pareto; se for muito pequeno, a população final será formada por muitas soluções dominadas.

Erickson *et. al.* (2001) propuseram o NPGA2, com o intuito de melhorar a dependência do NPGA em relação ao tamanho do conjunto de comparação. No NPGA o rank é baseado na Pareto dominância, no NPGA2 a classificação (*rank*) passa a ser o grau de Pareto dominância, como ilustrado na Figura 2.9:

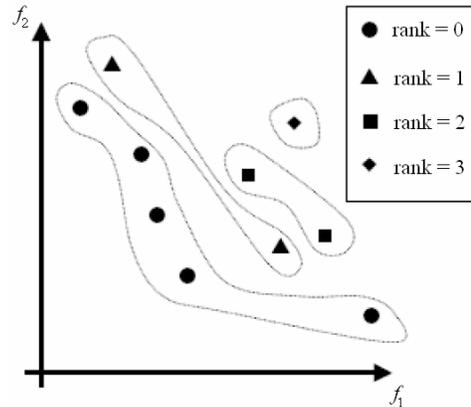


Figura 2.9. Ranking por grau de Pareto-dominância do NPGA2.

- **PAES (Pareto Archived Evolution Strategy):** Knowles e Corne (2000) afirmam a existência de aplicações reais nas quais os algoritmos de busca local (Busca Tabu, *Simulated Annealing*) superam os Algoritmos Genéticos. Considerando isso, os autores introduzem o algoritmo PAES, que emprega uma busca local, em três variantes:  $(1+1)$ ,  $(1+\lambda)$ ,  $(\mu+\lambda)$ .

O algoritmo  $(1+1)$ -PAES é composto de três partes: gerador de soluções candidatas (que mantém a solução e em cada geração produz um único candidato por mutação aleatória); a função de aceitação dos candidatos (usa um conjunto de comparação para ajudar a decidir entre a solução mutada e a atual) e o arquivo de soluções não-dominadas (que permite realizar comparações entre soluções não-dominadas).

No  $(1+\lambda)$ -PAES, a geração dos  $\lambda$  mutantes da solução atual aumenta o problema de decisão. No caso  $(\mu+\lambda)$ -PAES, os  $\lambda$  mutantes são gerados a partir das  $\mu$  soluções selecionadas por torneio binário.

Utiliza um *grid* adaptativo para a manutenção da diversidade.

- **NSGA (Nondominated Sorting Genetic Algorithm):** Proposto por Srinivas e Deb (1994). A principal característica do algoritmo é que antes da seleção, os indivíduos são classificados em diferentes níveis de dominância (fronteiras de classificação). Primeiro, os indivíduos não-dominados presentes na população corrente são identificados. Estes indivíduos recebem um valor de *fitness dummy* (o que permite que tenham o mesmo potencial de serem selecionados). Depois, o

*fitness* por compartilhamento (*fitness sharing*) é aplicado nos indivíduos da primeira fronteira, de maneira a penalizar seus respectivos *fitness*, baseado no número de indivíduos que compartilham a mesma vizinhança (o tamanho da vizinhança é controlado por um parâmetro, o raio de nicho). Então, os indivíduos da primeira fronteira são retirados da população para continuidade do processo de classificação dos indivíduos restantes nas respectivas fronteiras (os indivíduos da segunda fronteira recebem *fitness dummy* menor que os indivíduos da primeira fronteira, e assim por diante).

Como os indivíduos da primeira fronteira têm os maiores valores de *fitness*, eles serão selecionados mais vezes que o resto da população, o que permite a busca por indivíduos não-dominados. Além disso, a diversidade da população é mantida usando o método de *fitness* por compartilhamento.

O NSGA recebeu algumas críticas em relação à alta complexidade computacional de ordenação de não-dominância, à necessidade de especificar o parâmetro de compartilhamento e à falta de elitismo.

O NSGA2, proposto por Deb *et. al.* (2002) é computacionalmente mais eficiente do que o NSGA. Também deixou de utilizar as fronteiras na classificação das soluções, passando a utilizar somente o conceito de soluções dominadas e não-dominadas. Passou a utilizar a técnica de comparação por *crowding* (um operador de comparação de aglomeração que prioriza as melhores fronteiras ou pontos que estão menos aglomerados) o que eliminou a necessidade de especificar um parâmetro de compartilhamento. Também utilizou o conceito de elitismo.

Outro enfoque do NSGA2 está no estudo de problemas de Otimização Multiobjetivo restritos.

- **SPEA (Strength Pareto Evolutionary Algorithm):** Proposto por Zitzler e Thiele (1999), o algoritmo usa o critério de Pareto dominância para a atribuição de *fitness* e seleção dos indivíduos. A cada geração, indivíduos da população corrente classificados como não-dominados são copiados para a população externa (arquivo).

A atribuição de *fitness* é realizada em dois estágios: primeiro, um valor representando a força de Pareto (*strength*), é atribuído para cada indivíduo do arquivo (proporcional ao número de indivíduos da população corrente que por ele são dominados); depois, para cada indivíduo da população corrente é calculado um valor de *fitness* somando a força de cada indivíduo do arquivo que o domina. Esta estratégia reflete intuitivamente a idéia de preferir indivíduos próximos à Fronteira de Pareto e ao mesmo tempo, distribuídos ao longo da superfície de busca.

A diversidade é mantida através de uma técnica de agrupamento chamada *average linkage method*. A seleção é realizada através de torneio binário com reposição.

Alguns problemas encontrados no algoritmo são: indivíduos da população dominados pelos mesmos indivíduos do arquivo possuem *fitness* idênticos, portanto, se o arquivo contém um único elemento então todos os indivíduos terão o mesmo *fitness* (neste caso, o SPEA comporta-se como uma busca aleatória); pouca ou nenhuma informação pode ser extraída se muitos indivíduos são indiferentes (situação comum em problemas com mais de dois objetivos); o algoritmo de agrupamento pode destruir as soluções nas bordas da Fronteira de Pareto.

O SPEA2, proposto por Zitzler *et. al.* (2001), tenta contornar as dificuldades do SPEA. O novo algoritmo redefine a atribuição de *fitness*, o mecanismo de *crowding* e o mecanismo para a preservação da Fronteira de Pareto. As principais alterações são: a atribuição de *fitness* leva em consideração quantos indivíduos dominam e são dominados por cada indivíduo (considerando população corrente e arquivo, juntos); uma técnica baseada no *k*-ésimo vizinho mais próximo é usada para estimar a densidade, isto permite um método de busca mais preciso; truncamento dos indivíduos do arquivo para preservação dos indivíduos da Fronteira de Pareto.

### 2.2.2.2 Sistemas Imunológicos Artificiais

Apesar da quantidade considerável de pesquisas realizadas nos últimos anos houve poucas tentativas em estender a resolução de problemas de Otimização Multiobjetivo a certas heurísticas populacionais. Particularmente, os esforços de estender um Sistema

Imunológico Artificial (SIA) para lidar com problemas de Otimização Multiobjetivo (OM) eram praticamente inexistentes até recentemente.

A seguir, uma revisão de algumas propostas:

- **CLONALG (Clonal Selection Algorithm):** de Castro e Von Zuben (2002), propuseram um Sistema Imunológico Artificial baseado no princípio de Seleção Clonal. Apesar de tratar somente problemas de otimização monobjetivo multimodais, o CLONALG inspirou algumas propostas de Sistemas Imunológicos Artificiais para Otimização Multiobjetivo.

Originalmente proposto para resolver problemas de aprendizagem de máquina e reconhecimento de padrões, com uma população aleatória de anticorpos que devem aprender a reconhecer um conjunto de antígenos. Devido às características adaptativas, o CLONALG foi estendido para aplicações a problemas de otimização. A implementação do algoritmo representa simplificada o princípio de seleção clonal durante a resposta imune adaptativa. Na resolução de problemas de otimização, todos os anticorpos são selecionados, gerando uma população de clones. Durante o processo de clonagem, alguns clones sofrerão mutação genética (maturação) com taxas elevadas e inversamente proporcionais à afinidade. O processo de seleção é realizado na população de clones, após sua maturação. Indivíduos com baixa afinidade são substituídos por outros aleatoriamente gerados.

- **opt-aiNet (Artificial Immune Network for Optimization):** de Castro e Timmis (2002), propuseram o opt-aiNet para a resolução de problemas de otimização multimodal. Parte das considerações propostas pelos autores auxiliou na implementação de alguns algoritmos imuno-baseados implementados para a resolução de problemas de Otimização Multiobjetivo. É basicamente um procedimento de busca evolutivo baseado em mutação e população com tamanho dinâmico. Cada célula gera um certo número de clones que estão sujeitos a um número de mutações inversamente proporcional ao *fitness* (valor da função-objetivo) do anticorpo-pai. Quando o algoritmo estagna, ou seja, quando a média não melhora significativamente após uma iteração, a afinidade entre as células

(medida de quão similares elas são, baseada na distância Euclidiana) é calculada. As células com afinidade menor que um limiar pré-especificado são eliminadas da população e novas células aleatoriamente geradas são inseridas, evitando assim a redundância.

Algumas melhorias e modificações foram propostas por de França *et. al.* (2005), com a implementação do dopt-aiNet (opt-aiNet em ambientes dinâmicos). Dentre as modificações: a utilização de uma subpopulação (memória) que armazena as células que deixaram de evoluir, assumindo que elas convergiram para um ótimo local; novos operadores de mutação; um novo mecanismo de supressão; e limite do tamanho da população.

- **MISA (Multiobjective Immune System Algorithm):** A proposta de Coello e Cortés (2005) é um Sistema Imunológico Artificial, baseado no princípio de seleção clonal, para resolver problemas de Otimização Multiobjetivo. Os anticorpos foram representados por *strings* binárias que codificaram as variáveis de decisão do problema. Dominância e factibilidade das soluções (anticorpos) foram utilizadas como medida de afinidade, para identificar soluções que mereciam ser clonadas. A mutação uniforme foi aplicada aos clones e mutação não-uniforme aos anticorpos “não tão bons”. Uma população externa armazena as soluções não-dominadas encontradas ao longo do processo de busca. Esta população, além de ser o mecanismo elitista mais comumente adotado em Otimização Multiobjetivo, também está de acordo com a memória do Sistema Imunológico Artificial. Também existe interação com a população externa para decidir quantos clones produzir, baseado num critério de *crowding* (aglomeração).

O uso da medida de afinidade é uma alternativa interessante para manter a diversidade da população (ao invés do uso de *fitness sharing*).

O princípio de seleção clonal permite a busca local em diferentes direções ao longo da Fronteira de Pareto (exploração) através do processo de clonagem. A mutação aplicada aos piores anticorpos desempenha a exploração através de todo o espaço de busca. Este balanço entre exploração e exploração é muito importante para a Otimização Multiobjetivo.

- **ISPEA (Improvement for the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization with Immunity):** Entre os mais importantes algoritmos para Otimização Multiobjetivo está o SPEA (Seção 2.2.2.1). Hongyun e Sanyang (2003), propuseram o ISPEA, uma tentativa de melhoria do SPEA utilizando conceitos de Imunidade, visando conter a degeneração do processo de evolução (devido à aleatoriedade). Na prática, percebe-se também que, usando somente o Algoritmo Genético, torna-se difícil simular a habilidade com a qual os humanos tratam os problemas (como aprender, explorar e utilizar a inteligência humana). Os seguintes conceitos são utilizados:

1) Extração de vacinas: através da análise detalhada das soluções não-dominadas mantidas num conjunto externo (memória), extrair informações e características comuns e transformá-las em vacinas. A escolha da vacina afeta a eficácia da busca. Vacinas podem ser estimações para a solução ótima de algum componente ou relações entre componentes da solução. As vacinas são usadas para aumentar a diversidade da população e promover toda a população através de uma direção ótima.

2) Vacinação: escolher alguns indivíduos de acordo com uma proporção dada e com um conhecimento inicial modificar alguns componentes ou genes de acordo com a vacina, repetidamente. Se o indivíduo corrente é ótimo então ele se transforma nele mesmo com probabilidade um.

3) Seleção Imune: a seleção é necessária depois da vacinação para impedir degeneração e prematuridade. Se um indivíduo vacinado é pior que seu pai então é o pai que passa para a próxima geração, ou vice-versa.

Com vacinação, mais soluções Pareto-ótimas são encontradas numa extensão mais ampla da Fronteira de Pareto, sem muito tempo de computação. Isto mostra que esta abordagem também guiou a busca através da Fronteira de Pareto. Além disso, o ISPEA também manteve os méritos do SPEA, faz uso de informações e conhecimentos prévios do problema durante todo o processo evolutivo e contêve a degeneração.

- **VAIS (Vector Artificial Immune System):** Freschi e Repetto (2005), apresentaram um Sistema Imunológico Artificial para resolução de problemas de Otimização Multiobjetivo, baseado no princípio de seleção clonal. Os anticorpos foram codificados por variáveis reais. Dominância e factibilidade das soluções (anticorpos) foram utilizadas como medida de afinidade. Para cada solução não-dominada, a afinidade é igual a uma força (*strength*), como definida no SPEA2. Para as soluções dominadas, a afinidade é igual ao número de soluções que a dominam. Esta classificação resulta num ordenamento parcial porque todas as soluções não-dominadas possuem afinidade menor que um e as soluções dominadas possuem afinidade maior que um. Todos os anticorpos são clonados. Quanto maior a afinidade do anticorpo-pai, menor a amplitude de mutação sofrida pelos herdeiros (clones).

Os anticorpos não-dominados são armazenados na memória. Calcula-se a distância Euclidiana entre os anticorpos da memória, no espaço das funções-objetivo, com o intuito de obter uma Fronteira de Pareto uniformemente distribuída. Se o indivíduo gerado for inactível, ele é descartado. Se a inactibilidade ocorre após a aplicação do operador de mutação, reduz-se progressivamente a amplitude da mutação até que o clone se torne factível.

- **omni-aiNet (Artificial Immune Network for Omni-optimization):** Coelho e Von Zuben (2006) propuseram este algoritmo baseado principalmente no opt-aiNet (de Castro e Timmis, 2002), com a incorporação de alguns mecanismos introduzidos pelo dopt-aiNet (de França *et. al.*, 2005). Além disso, o algoritmo contempla a resolução de problemas de otimização monobjetivo ou multiobjetivo, e unimodais ou multimodais; a chamada omni-otimização, proposta anteriormente por Deb e Tiwari (2005) com a apresentação do algoritmo *Omni-Optimizer*.

O algoritmo possui um repertório de anticorpos capaz de ajustar seu tamanho durante a execução do algoritmo, de acordo com um limiar de supressão pré-definido e um mecanismo de *grid* que controla o espalhamento das soluções no espaço objetivo. Esta característica garante maior flexibilidade ao processo de

busca, uma vez que o algoritmo ajusta o tamanho do repertório automaticamente, propiciando uma melhor alocação dos recursos computacionais.

Inicia-se com a geração aleatória do repertório inicial, formado por anticorpos codificados por variáveis reais e representando uma solução do problema. Os principais passos do algoritmo são executados: clonagem, hipermutação polinomial, seleção e duplicação genética (mecanismo de introdução de variabilidade genética). A supressão (baseada na distância Euclidiana) e a inserção de novos anticorpos aleatoriamente gerados são realizadas segundo uma frequência definida pelo usuário.

Após a clonagem e a hipermutação, o repertório (formado por anticorpos-pai e clones) é ordenado em classes, de acordo com a qualidade de cada indivíduo. A qualidade é definida de acordo com o conceito de  $\varepsilon$ -dominância restrita, originalmente proposta por Deb e Tiwari (2005).

O *omni-aiNet* foi aplicado na resolução de vários problemas de otimização (monobjetivo e multiobjetivo).

### 2.2.2.3 Enxame de Partículas

A utilização da técnica de Enxame de Partículas (ou *Particle Swarm*) na resolução de problemas de Otimização Multiobjetivo é relativamente recente. Inspirada no movimento (ou coreografia) de grupos de pássaros, esta técnica já vinha obtendo bons resultados numa variedade de problemas de otimização monobjetivo. Alguns algoritmos de Enxame de Partículas aplicados a Otimização Multiobjetivo serão apresentados.

- **MOPSO (Multiobjective Particle Swarm Optimization):** Esta abordagem de Coello e Lechuga (2002) utiliza o conceito de Pareto dominância para determinar a direção de vôo da partícula e mantém as soluções não-dominadas encontradas nas iterações anteriores em um repositório (arquivo histórico). O repositório é utilizado de maneira a guiar o vôo de outras partículas. Através dele cada partícula deve escolher um líder para guiar a busca. O uso de mecanismos globais de atração,

combinados com este repositório motivaria a convergência para a Fronteira de Pareto global.

A atualização do repositório é realizada considerando um sistema geograficamente baseado definido em termos dos valores das funções-objetivo de cada partícula. Esta técnica é inspirada no arquivo externo do algoritmo PAES, discutido na Seção 2.2.2.1.

O número de partículas de um algoritmo de Enxame de Partículas é equivalente ao tamanho da população em um Algoritmo Genético. E o tamanho do repositório é utilizado para delimitar o número máximo de soluções não-dominadas que devem ser armazenadas; este parâmetro irá determinar a qualidade da Fronteira de Pareto produzida.

O algoritmo proposto resolve somente problemas irrestritos.

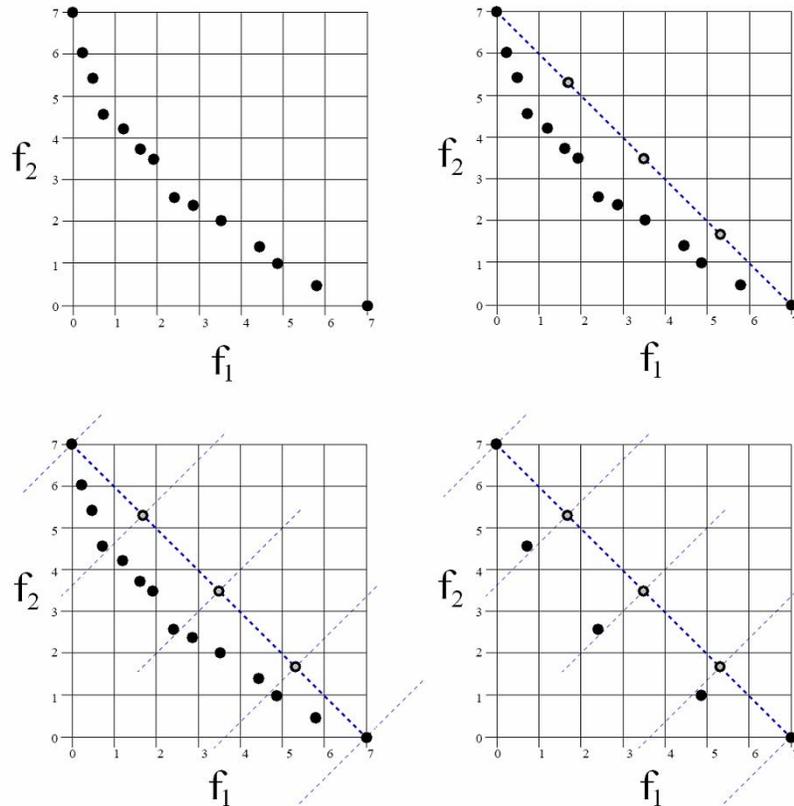
▪ **EMOPSO (Efficient Multi-Objective Particle Swarm Optimizer):**

Toscano-Pulido *et. al.* (2007) focaram o estudo deste trabalho no desenvolvimento de mecanismos para um algoritmo de enxame de partículas na resolução de problemas de Otimização Multiobjetivo.

Para a seleção de soluções não-dominadas bem distribuídas foram propostos três mecanismos: um *grid* adaptativo; uma forma relaxada de Pareto dominância; e uma distribuição por hiperplano. A abordagem que obteve mais sucessos nos testes foi a distribuição por hiperplano. A idéia é selecionar um conjunto representativo de soluções não-dominadas distribuídas no hiperplano definido pelos valores mínimos dos objetivos. Seja  $k$  o número desejado de soluções não-dominadas. Então, o algoritmo seleciona as soluções que representam o valor mínimo de cada objetivo. Um hiperplano entre todas as soluções mínimas é computado. Depois o espaço é dividido em  $k-1$  regiões e no vértice de cada região uma linha perpendicular ao hiperplano é traçada. Finalmente o algoritmo aceita as soluções que estão mais próximas de cada linha. A Figura 2.10 ilustra o funcionamento deste mecanismo.

Também foi proposto um operador de turbulência, que altera a velocidade do vôo da partícula, com o intuito de evitar a convergência prematura e controlar a pressão seletiva.

Em algoritmos de Enxame de Partículas a seleção apropriada do líder é fundamental para uma boa performance do algoritmo, caso contrário, as partículas não serão capazes de visitar as regiões promissoras do espaço de busca. Para evitar esse problema, os autores propuseram a utilização de vários enxames (e conseqüentemente, vários líderes). Esta abordagem obteve bons resultados principalmente na presença de Fronteiras de Pareto desconexas.



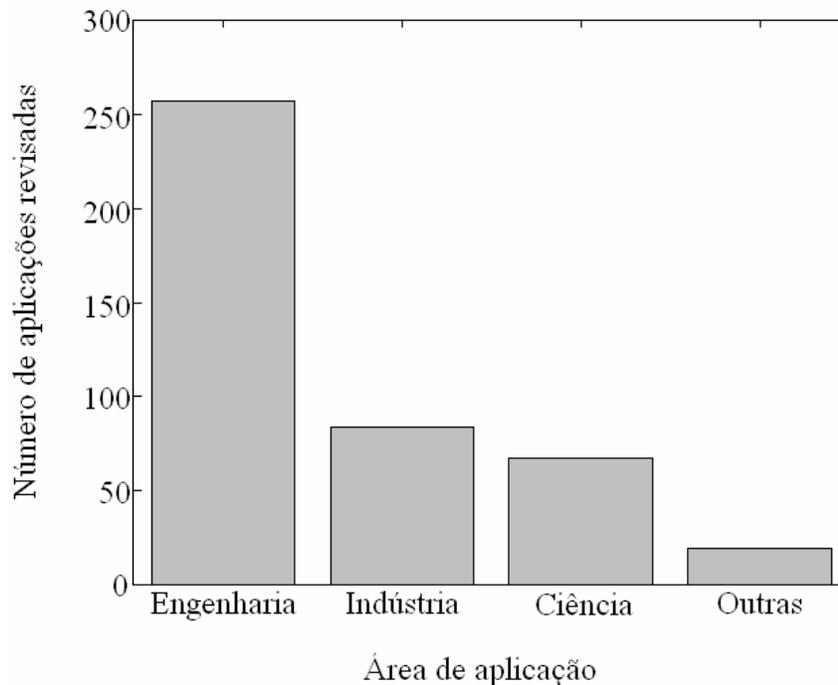
**Figura 2.10.** Representação de uma distribuição por hiperplano para  $k = 5$ .

O algoritmo MPSO proposto por Coello e Lechuga (2002) não utilizou nenhum mecanismo para lidar com espaços de busca restritos. Em EMPSO um mecanismo que trata as restrições foi proposto.

Além disso, um mecanismo de seleção proporcional tradicional foi implementado com o objetivo de selecionar a combinação apropriada a ser adotada para os valores dos parâmetros do algoritmo. Desta maneira, não é requerido nenhum ajuste manual dos parâmetros.

## 2.3 Aplicações

As aplicações de Algoritmos Bio-inspirados em problemas de Otimização Multiobjetivo têm crescido notavelmente e ocupam grande parte da bibliografia da área. A Figura 2.11 ilustra a distribuição da pesquisa nas áreas principais: Engenharia (Elétrica, Eletrônica, Civil, Aeronáutica, Robótica, Telecomunicações, etc.), Indústria (Manufatura, Gerenciamento, etc.), Ciências (Física, Medicina, Ecologia, Química, etc) e Outras (Finanças, Classificação, Predição, etc.).



**Figura 2.11.** Distribuição das Aplicações (Coello, 2002b).

A seguir, uma breve exemplificação de algumas aplicações. É importante ressaltar que todos os trabalhos propostos na literatura têm relevância nas respectivas áreas de pesquisa. Mas devido à imensa quantidade de trabalhos correlacionados, a escolha foi pela apresentação de uma aplicação representando cada área.

- **Despacho de Energia Elétrica:** Abido (2003) apresenta um algoritmo baseado no SPEA (Seção 2.2.2.1) para resolver o problema de despacho econômico-ambiental ótimo de potência. O problema é formulado com dois objetivos: minimização de custos de geração e minimização da emissão de poluentes pelos geradores. É adotado um procedimento para apresentar a solução de melhor compromisso ao decisor, dentre todas presentes no Conjunto Pareto-ótimo.
- **Reativos em Sistemas de Potência:** Barán *et. al.* (2001) apresentaram um Algoritmo Bio-inspirado Multiobjetivo para o Problema de compensação de reativos em sistemas de potência. O problema é formulado com a minimização de quatro objetivos: investimento em equipamentos de compensação, perdas de potência ativa, desvio médio de tensão e desvio máximo de tensão. Os autores propõem modificações no algoritmo SPEA para a resolução do problema.
- **Placa de Circuito Impresso:** Deb *et. al.* (2004) utiliza a técnica NSGA2 (Seção 2.2.2.1) para resolver um problema de otimização da localização de componentes eletrônicos numa placa de circuito. O problema apresenta dois objetivos: minimizar o comprimento total dos fios e minimizar a taxa de deficiência devido a acumulação total de temperatura.
- **Manufatura:** Dimopoulos (2004) realiza uma revisão das aplicações de Algoritmos Bio-inspirados em problemas de manufatura com múltiplos objetivos. Problemas de alocação, planejamento e controle de produção, manufatura celular, sistemas de manufatura flexíveis e otimização de linha de montagem foram considerados. Alguns desses problemas já utilizam os benefícios das últimas técnicas desenvolvidas: NSGA2 e SPEA2 (Seção 2.2.2.1).
- **Otimização Multiobjetivo Fuzzy:** Moura (2002) propõe um Algoritmo Genético para otimizar problemas de Otimização Multiobjetivo com incertezas no conjunto de restrições e nas funções-objetivo.

- **Diagnósticos Médicos:** Toro *et. al.* (2006) propuseram um algoritmo para encontrar um diagnóstico médico preciso (baseado na interpretação de bio-sinais) e satisfazer outros objetivos de interesse médico. A utilização de Algoritmos Bio-inspirados para Otimização Multiobjetivo e Multimodal resulta em um método médico-especialista com diferentes alternativas para configurar o esquema de diagnósticos.

## Capítulo 3

# Sistemas Imunológicos Artificiais e a Otimização Multiobjetivo

A primeira parte deste capítulo apresenta o funcionamento e os principais componentes do Sistema Imunológico Biológico utilizados neste trabalho. Na segunda parte, discute-se sobre os Sistemas Imunológicos Artificiais e considerações sobre problemas de engenharia, com destaque à otimização. A discussão geral deste capítulo foi inspirada em de Castro e Von Zuben (1999, 2000) e de Castro (2001).

### 3.1 Princípios Fundamentais do Sistema Imunológico

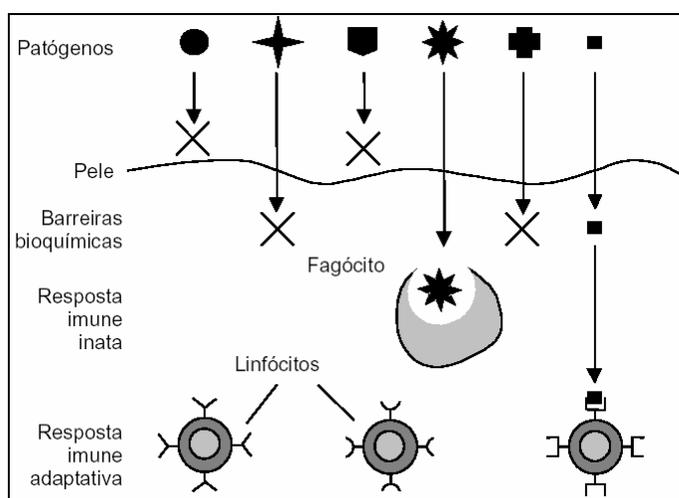
Na apresentação dos conceitos básicos do Sistema Imunológico Biológico serão evidenciadas somente as funcionalidades consideradas importantes na resolução do problema proposto, pois a complexidade do Sistema Imunológico é enorme. Maior ênfase será dada ao princípio de Seleção Clonal e aos mecanismos de Maturação de Afinidade.

Conceitualmente, o Sistema Imunológico é uma coleção de moléculas, células e órgãos cuja complexa interação num sistema eficiente é usualmente capaz de proteger um indivíduo de invasores ou de suas próprias células alteradas devido a doenças.

O Sistema Imunológico está estruturado em múltiplas camadas (Figura 3.1), com funções e mecanismos em diferentes níveis de defesa:

- Barreiras físicas: a pele e o sistema respiratório atuam como o primeiro nível de defesa

- Barreiras bioquímicas: fluidos como a saliva, lágrima, suor, ácidos estomacais, etc, eliminam ou neutralizam alguns microorganismos invasores.
- Sistema imune inato: formado por células fagocitárias, como os *macrófagos* e os *neutrófilos*, além de fatores solúveis. Este nível desempenha um papel importante na iniciação e posterior direcionamento das respostas imunes adaptativas. Atua no controle das infecções enquanto o sistema imune adaptativo é ativado por completo.
- Sistema imune adaptativo: formado por células (*linfócitos*) que evoluem para proporcionar meios de defesas mais versáteis e um maior nível de proteção face às novas infecções pelo mesmo agente. As características que fomentam a teoria dos Sistemas Imunológicos Artificiais são baseadas nas funcionalidades do Sistema imune adaptativo e serão discutidas a seguir.



**Figura 3.1.** Estrutura multicamadas do Sistema Imunológico. (de Castro e Von Zuben, 1999).

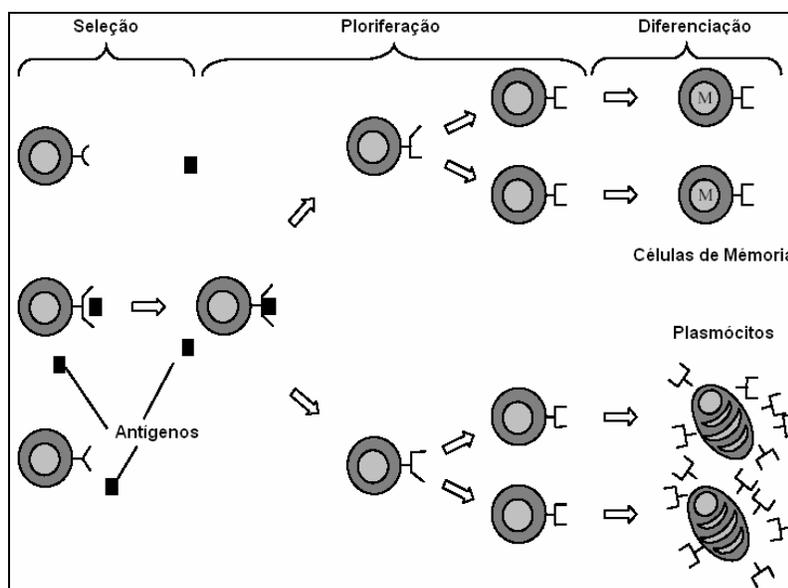
### 3.1.1 Sistema imune adaptativo

Os linfócitos (produzidos na medula óssea e no timo) são as principais células que compõem este sistema. Presentes apenas nos animais vertebrados, eles desempenham o principal papel no desencadeamento e posterior regulação das respostas imunes inatas.

Os linfócitos desempenham o principal papel no Sistema Imunológico. Cada “linfócito virgem” que penetra na corrente circulatória é portador de receptores de um antígeno com uma única especificidade.

Os linfócitos (ou anticorpos de superfície) sofrem um processo parecido com a seleção natural (Lederberg, 1988): somente aqueles que encontram um antígeno com o qual seu receptor pode interagir serão ativados para se proliferarem. Eles são capazes de se aderir ao antígeno com o objetivo de neutralizá-los e marcá-los para que outras células do sistema imunológico os eliminem.

Após a ligação do anticorpo de superfície ao antígeno, a célula é ativada para proliferar por clonagem. Estes clones irão se diferenciar em plasmócitos secretando os anticorpos com a mesma especificidade do receptor; ou células de memória: quando o sistema imunológico estiver novamente exposto a um determinado antígeno, estas células são ativadas a fim de obter uma resposta futura mais eficiente (Figura 3.2). Este princípio recebeu o nome de teoria da Seleção Clonal, e constitui a parte central da imunidade adaptativa.



**Figura 3.2.** O Sistema imune adaptativo. (de Castro e Von Zuben, 1999).

Após a clonagem das células de alta afinidade, a hipermutação somática é acionada para refinar, por trocas de bases nucleotídicas, a resposta imunológica ao antígeno

reconhecido. Isso permite a criação de moléculas de imunoglobulina (anticorpos) capazes de reconhecer aquele antígeno com maior eficiência. Estes mecanismos de geração e mutação proporcionam um reconhecimento poderoso de antígenos, uma vez que o Sistema Imunológico pode produzir um número praticamente infinito de receptores celulares a partir de um genoma finito.

Uma propriedade importante do Sistema Imunológico é a contínua produção de novos anticorpos, denominada de metadinâmica. Existe uma constante renovação do repertório linfocitário e, conseqüentemente, a morte de elementos não estimulados. A metadinâmica representa o potencial que o Sistema Imunológico possui de introduzir diversidade, garantindo sua capacidade de combater novos antígenos.

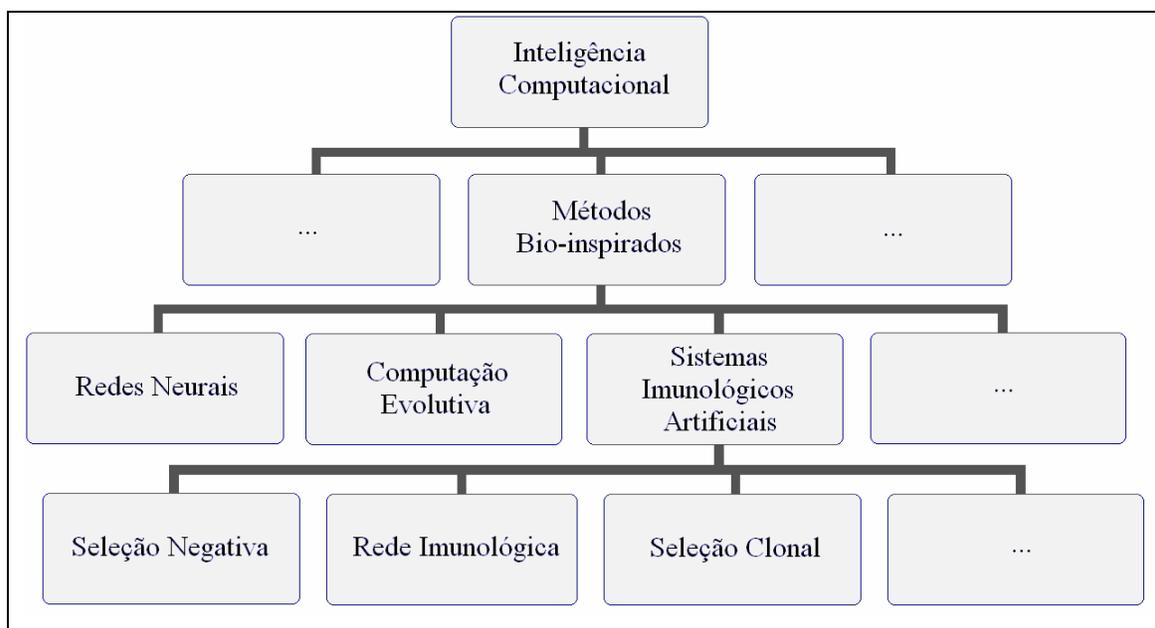
O Sistema Imunológico natural possui uma grande variedade de componentes e funcionalidades. Para desenvolver um sistema computacional inspirado no natural, faz-se necessário uma simplificação de maneira a enquadrar os conceitos biológicos, julgados mais relevantes, para aplicação no problema em questão.

## **3.2 Sistemas Imunológicos Artificiais**

Analogamente à Computação Evolutiva, os Sistemas Imunológicos Artificiais se inspiram na natureza, a partir da tentativa de simular mecanismos particulares e criar sistemas artificiais para a resolução de problemas complexos.

A área de pesquisa dos Sistemas Imunológicos Artificiais é extensa. A escolha dos melhores modelos (Seleção Negativa, Teoria de Rede Imunológica, Seleção Clonal (Figura 3.3)) depende do objetivo e das características do problema a ser estudado. As diferentes características, funcionalidades e propriedades dos Sistemas Imunológicos Artificiais permitem sua aplicação em diversos problemas.

Diante da quantidade de modelos e aplicações dos Sistemas Imunológicos Artificiais, não existe um esquema geral de quais elementos essenciais um algoritmo deve possuir, ao contrário de outras técnicas bio-inspiradas, que possuem um algoritmo geral único.



**Figura 3.3.** SIA: uma ramificação da Inteligência Computacional. (Dasgupta, 2006).

Segundo Dasgupta (1998), os Sistemas Imunológicos Artificiais são ferramentas computacionais compostos por metodologias inteligentes, inspirados no Sistema Imunológico Biológico, para a solução de problemas do mundo real.

Os Sistemas Imunológicos Artificiais surgiram a partir de tentativas de modelar e aplicar princípios imunológicos no desenvolvimento de novas ferramentas computacionais (Tabela 3.1): reconhecimento de padrões, detecção de faltas e anomalias, segurança computacional, otimização, controle, robótica, scheduling, análise de dados, aprendizagem de máquina, processamento de informações, resolução de problemas, etc.

<b>Característica do Sistema Imunológico Biológico</b>	<b>Problema Computacional</b>	<b>Aplicações Típicas</b>
Reconhecimento de próprio e não-próprio	Deteção de mudanças ou anomalias	· Segurança de computador · Deteção de Falta
Teoria de Rede Imunológica e Memória Imunológica	Aprendizagem (supervisionada ou não-supervisionada)	· Classificação · Clusterização · Análise de dados · Mineração de dados

<b>Característica do Sistema Imunológico Biológico</b>	<b>Problema Computacional</b>	<b>Aplicações Típicas</b>
Seleção Clonal	Busca, Otimização	· Otimização de funções
Mobilidade e Distribuição	Processamento Distribuído	· Arquiteturas de agente · Controle Robótico Descentralizado
Imunidade Inata	Teoria do Perigo	· Segurança de redes

**Tabela 3.1.** Resumo dos modelos de SIA mais estudados (Dasgupta, 2006).

Algumas características do Sistema Imunológico podem ser destacadas, sob a perspectiva da engenharia (de Castro, 2001):

- Unicidade: cada indivíduo possui seu próprio Sistema Imunológico, com suas capacidades e vulnerabilidades particulares.
- Reconhecimento de padrões internos e externos ao sistema: as células e moléculas que não pertencem ao organismo são reconhecidas e eliminadas pelo Sistema Imunológico.
- Detecção de anomalias: o Sistema Imunológico pode detectar e reagir a agentes patogênicos.
- Detecção imperfeita: uma detecção perfeita não é necessária (tolerância a ruídos).
- Detecção distribuída: as células do Sistema Imunológico estão distribuídas por todo o corpo e não estão sujeitas a um mecanismo de controle central.
- Diversidade: existe uma quantidade limitada de células e moléculas no Sistema Imunológico que são utilizadas para se obter o reconhecimento de um número praticamente infinito de elementos.
- Aprendizagem por reforço: a cada encontro com o mesmo patógeno o Sistema Imunológico melhora a qualidade de sua resposta.
- Memória: os componentes do Sistema Imunológico bem sucedidos no reconhecimento e combate às patologias são armazenados para uma resposta futura mais intensa e efetiva.

### 3.2.1 Considerações em Otimização Multiobjetivo

Os Sistemas Imunológicos Artificiais apresentam, em sua estrutura elementar, as principais características requeridas para a resolução de problemas de Otimização Multiobjetivo: elitismo, memória, diversidade, mutação e clonagem proporcionais à afinidade e repertório dinâmico. Estas características inerentes fazem com que o método tenha vantagens em relação a outras estratégias populacionais.

Na Seção 3.2 foram destacadas algumas características do Sistema Imunológico sob a perspectiva da engenharia. Uma análise paralela entre algumas destas características e os problemas de Otimização Multiobjetivo também pode ser realizada:

- **Detecção distribuída:** os anticorpos devem estar espalhados no espaço de busca, cada um responsável pela exploração de uma região. Evitando assim a aglomeração (centralização) das soluções. O objetivo do processo de otimização é determinar múltiplos ótimos (soluções não-dominadas) dentro do repertório de anticorpos. Neste caso, o interessante é que todo o repertório de anticorpos (ou a maior parte dele) seja selecionado para clonagem, sendo cada anticorpo analisado localmente, resultando em um algoritmo capaz de executar uma busca por várias soluções distintas. Além disso, soluções ótimas locais são simultaneamente preservadas.
- **Diversidade:** o espaço de busca tem infinitas possibilidades que devem ser exploradas. Além disso, um problema de Otimização Multiobjetivo deve apresentar um conjunto de soluções diversas, a fim de que toda a Fronteira de Pareto (ou a maior parte dela) possa ser obtida. Os Sistemas Imunológicos são inerentemente capazes de manter a diversidade da população. Em um Algoritmo Genético seria necessário algum mecanismo para a manutenção de diversidade; nos Sistemas Imunológicos, módulos com algumas semelhanças com *niching* (nichos) e *fitness sharing* (compartilhamento de fitness) são partes intrínsecas do algoritmo (Coelho e Von Zuben, 2006).

- **Memória:** Muitos Algoritmos Bio-inspirados para Otimização Multiobjetivo trabalham com uma memória externa elitista, para que não ocorra a perda do melhor indivíduo (solução não-dominada). Esta já é uma característica intrínseca do Sistema Imunológico, pois no processo de seleção clonal, é a melhor célula (entre anticorpo-pai e clones) que sobrevive para a próxima iteração. O processo de seleção clonal é elitista, logo o Sistema Imunológico Artificial não apresenta efeito retrógrado durante as iterações.

Outras características do Sistema Imunológico, interessantes na resolução de problemas de otimização podem ser ressaltadas:

- A amplitude da mutação sofrida pelos clones é inversamente proporcional à afinidade do anticorpo-pai, de maneira que, quanto maior a afinidade, menor será a mutação (Kepler e Perelson, 1993).
- Uma vez que os antígenos tenham sido eliminados do organismo, existirá uma quantidade excedente de anticorpos e o sistema deve voltar ao seu nível normal. É por isso que algumas células são eliminadas (autoregulação). Algumas, porém, se conservam, circulando pelo organismo como células de memória. O nível normal de anticorpos no repertório não necessariamente se manterá constante. Ao contrário, o tamanho do repertório é dinâmico. Em otimização, o interesse é que esse tamanho seja estabelecido de acordo com a necessidade do problema.

Mais considerações, de acordo com a Teoria Imunológica, serão propostas no algoritmo implementado no Capítulo 4.

# Capítulo 4

## O Algoritmo Proposto

O foco desta dissertação é a apresentação de um Sistema Imunológico Artificial elaborado com o intuito de ser genérico o bastante para tratar problemas de Otimização Multiobjetivo com características que geralmente dificultam a utilização de métodos exatos: não-linearidade, não-convexidade e presença de restrições.

Para a resolução dos problemas, admite-se que não são conhecidas informações de preferências entre os objetivos, de forma que o algoritmo deve gerar alternativas para a posterior escolha do decisor. Portanto, algumas metas importantes em Otimização Multiobjetivo devem ser atingidas:

- Encontrar um conjunto de soluções que esteja o mais próximo possível do Conjunto Pareto-ótimo.
- Encontrar um conjunto de soluções com a maior diversidade possível no espaço dos objetivos.
- Encontrar um conjunto de soluções com a maior diversidade possível no espaço das variáveis, principalmente nos casos multimodais.

Ou seja, o método de resolução deve guiar a busca pela Fronteira de Pareto global e manter a diversidade das soluções não-dominadas atuais (Deb, 1999).

As abordagens apresentadas no Capítulo 2 possuem características e mecanismos específicos responsáveis pelo desempenho de cada uma na resolução de problemas de Otimização Multiobjetivo. O algoritmo proposto nesta dissertação reuniu algumas das principais características propostas por estas abordagens (já conhecidas como importantes ou necessárias na resolução de problemas de Otimização Multiobjetivo).

Apesar do considerável número de pesquisas e abordagens a área ainda é promissora para novos trabalhos. O algoritmo, como inovação, propôs uma hibridização inspirada nos algoritmos de Otimização por Enxame de Partículas para o cálculo da direção de mutação dos clones. Além disso, a proposta indicou a importância do tratamento de problemas de otimização fortemente restritos (pouco estudados na literatura) e permitiu, para este tipo de problema, que algumas soluções infactíveis fossem consideradas no processo de busca.

É proposto um Sistema Imunológico Artificial inspirado nos conceitos apresentados no Capítulo 3. A razão da proposta deste trabalho é que o Sistema Imunológico Artificial possui em sua estrutura elementar (além de todas as características dos Algoritmos Bioinspirados populacionais citadas na Seção 2.2.2), mecanismos interessantes para a resolução de problemas de Otimização Multiobjetivo, tais como: manutenção de diversidade, supressão, elitismo, repertório dinâmico, clonagem e mutação proporcionais à afinidade.

## 4.1 Apresentação dos componentes e funcionalidades

O repertório é formado por anticorpos (possíveis soluções do problema de Otimização Multiobjetivo) que geram novos herdeiros (clones) com as mesmas características dos anticorpos-pai. Os clones evoluem por meio de operadores de maturação. Uma medida de afinidade (definida de acordo com critérios de dominância e factibilidade) é utilizada de forma a privilegiar a proliferação dos anticorpos mais adaptados ao ambiente (Seleção Clonal).

A seguir, uma apresentação detalhada de cada componente de funcionalidade do Sistema Artificial proposto é realizada.

### 4.1.1 Representação

O repertório  $\mathbf{Ab}$  é formado por  $n_T$  anticorpos de dimensão  $n$ .

Cada anticorpo  $\mathbf{Ab}_i$  codifica um vetor de argumentos das funções-objetivo  $f_1(\cdot), f_2(\cdot), \dots, f_m(\cdot)$ , ou seja, cada anticorpo representa um vetor de variáveis reais

$$\mathbf{Ab}_i = \mathbf{x} = [x_1, x_2, \dots, x_n].$$

### 4.1.2 Inicialização do Repertório

O repertório é inicializado de modo aleatório e quando possível, dentro da região factível.

Em alguns tipos de problemas, restringir o acesso a regiões infactíveis, que poderiam servir de acesso a regiões factíveis, poderia não dar bons resultados. Dessa forma, considerou-se também a representação de anticorpos infactíveis com a finalidade de melhorá-los ao invés de rejeitá-los.

### 4.1.3 Medida de afinidade

A medida de afinidade pode ser entendida como o desempenho de um anticorpo.

Os anticorpos são classificados de acordo com critérios de dominância e factibilidade. Quanto maior a afinidade de um anticorpo, menor será sua classificação. Ou seja, os melhores anticorpos (não-dominados) possuirão classificação igual a zero; e a classificação é crescente, de maneira que a maior classificação pertencerá ao anticorpo mais dominado.

Inicialmente, determina-se se um anticorpo é factível ou não.

- I. Para cada anticorpo factível  $\mathbf{Ab}_i$  determina-se, de acordo com os critérios de dominância definidos na Seção 2.1, o número de anticorpos que o dominam e o número de anticorpos dominados por ele. É importante ressaltar que a relação de Pareto-dominância é determinada somente entre anticorpos da mesma classe (ou seja, anticorpos factíveis serão comparados somente com outros anticorpos factíveis).

Cada anticorpo factível recebe uma classificação  $\text{rank}(\mathbf{Ab}_i)$ , definida da seguinte maneira:

$$\text{rank}(\mathbf{Ab}_i) = \sum_{j \text{ domina } i} S(j) \quad (4.1)$$

$$S(j) = \frac{\text{número de soluções dominadas por } j}{\text{total de soluções dominadas}} \quad (4.2)$$

Ou seja, quanto melhores (ou quanto mais fortes, mais dominantes) forem as soluções que dominam o anticorpo  $\mathbf{Ab}_i$ , maior será sua classificação.

Esta classificação é baseada na atribuição de *fitness* proposta por Zitzler et al. (2001) para o algoritmo SPEA2.

- II. Cada anticorpo inactível  $\mathbf{Ab}_i$  receberá uma classificação de acordo com a violação da factibilidade. Dessa maneira, quanto maior o  $\text{rank}(\mathbf{Ab}_i)$ , maior será a distância de  $\mathbf{Ab}_i$  à região factível.

$$\text{rank}(\mathbf{Ab}_i) = \max\{\mathbf{G}(\mathbf{Ab}_i), 0\} + |\mathbf{H}(\mathbf{Ab}_i)| + \max\{\mathbf{x}_l - \mathbf{Ab}_i, 0\} + \max\{\mathbf{Ab}_i - \mathbf{x}_u, 0\} \quad (4.3)$$

Ou seja quanto maior a violação das restrições  $\mathbf{G}(\mathbf{x}) \leq 0$ ,  $\mathbf{H}(\mathbf{x}) = 0$ ,  $\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$ , maior será a classificação do anticorpo  $\mathbf{Ab}_i$ .

#### 4.1.4 Clonagem

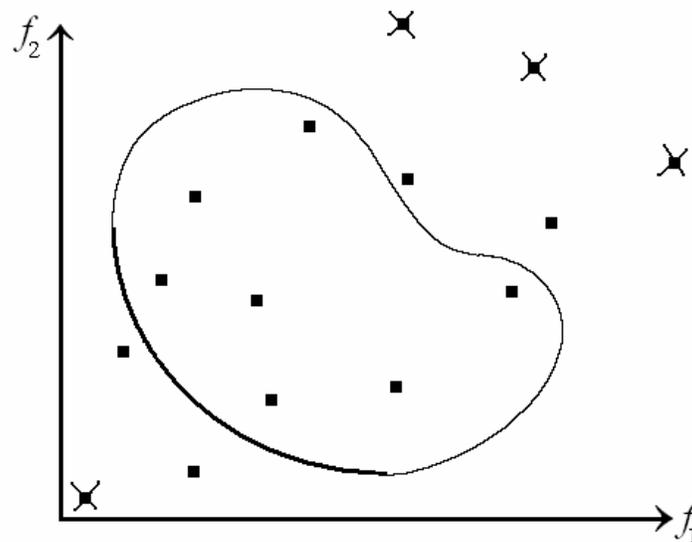
Como foi discutido anteriormente, múltiplos ótimos podem ser obtidos, logo, todos os anticorpos factíveis do repertório  $\mathbf{Ab}$  são selecionados para clonagem.

Em relação aos anticorpos inactíveis, somente aqueles que satisfizerem (4.4) serão selecionados para clonagem:

$$\frac{\text{rank}(\mathbf{Ab}_i)}{\sum_{j \text{ inactível}} \text{rank}(\mathbf{Ab}_j)} \leq \text{lim}_{\text{infac}} \quad (4.4)$$

Sendo que o limiar de inactibilidade,  $\text{lim}_{\text{infac}} \in [0,1]$ . Para problemas fortemente restritos, define-se  $\text{lim}_{\text{infac}}$  próximo de 1. Em problemas mais relaxados,  $\text{lim}_{\text{infac}}$  é definido próximo de 0.

A Figura 4.1 (criada a partir do espaço das funções e da Fronteira de Pareto apresentados nas Figuras 2.3 e 2.4) ilustra um exemplo considerando anticorpos factíveis e infactíveis seleccionados para clonagem e anticorpos infactíveis não seleccionados.



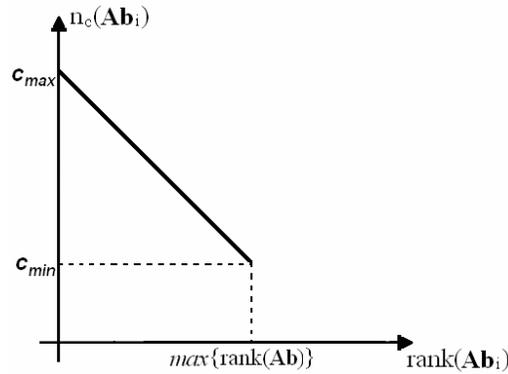
**Figura 4.1.** Exemplo de seleção de anticorpos para clonagem.

Para cada anticorpo  $\mathbf{Ab}_i$  seleccionado do repertório são gerados  $n_c(\mathbf{Ab}_i)$  clones idênticos.

- I. Para os anticorpos infactíveis  $n_c(\mathbf{Ab}_i) = c_{min}$ , ou seja o número de clones que cada anticorpo infactível gera é  $c_{min}$  (limiar mínimo). Com o intuito de não gastar muito recurso computacional na tentativa de melhora dos clones, mas ao mesmo tempo permitir que uma melhora ocorra (o que é interessante em problemas fortemente restritos). Os anticorpos que forem gerados infactíveis (Seção 4.1.11) e, após o processo de Maturação e Seleção Clonal, não obtiverem melhora, serão excluídos do repertório (Seção 4.1.8).
- II. Para os anticorpos factíveis, quanto maior a afinidade do anticorpo-pai (ou seja, quanto menor a classificação de  $\mathbf{Ab}_i$ ) mais clones serão gerados, segundo a Figura 4.2:

$$n_c(\mathbf{Ab}_i) = c_{max} - \left\lfloor \left( \frac{c_{max} - c_{min}}{\max\{\text{rank}(\mathbf{Ab}_i)\}} \right) (\text{rank}(\mathbf{Ab}_i)) \right\rfloor \quad (4.5)$$

Sendo:  $c_{max}$  = número máximo de clones,  $c_{min}$  = número mínimo de clones,  $\max\{\text{rank}(\mathbf{Ab}_i)\}$  = maior valor de classificação dentre os anticorpos factíveis,  $\text{rank}(\mathbf{Ab}_i)$  = classificação do anticorpo factível  $\mathbf{Ab}_i$ .



**Figura 4.2.** Número de clones gerados por anticorpo factível, em função de  $\text{rank}(\mathbf{Ab}_i)$ .

### 4.1.5 Maturação

Cada clone  $\mathbf{C}_j$  é localmente mutado por uma perturbação aleatória  $(\alpha_i \cdot \mathbf{d}_j)$ .

Onde  $\alpha_i$  é o tamanho do passo a partir da posição de  $\mathbf{Ab}_i$ , e  $\mathbf{d}_j$  é a direção de deslocamento do clone  $\mathbf{C}_j$ .

$$\mathbf{C}_j = \mathbf{Ab}_i + \alpha_i \cdot \mathbf{d}_j \quad (4.6)$$

A medida de afinidade é utilizada para calcular o número de clones gerados por anticorpo e a amplitude das mutações realizadas nos clones. Assim, anticorpos de alta afinidade sofrem mutações menores e geram mais clones, pois neste caso, o interesse é *intensificar* a busca localmente. E anticorpos de baixa afinidade sofrem mutações maiores e geram menos clones, pois um número maior de mutações permite uma maior *diversificação* das soluções promovendo assim uma melhor exploração deste espaço (Berbert *et. al.*, 2007). Este é um aspecto inerente às abordagens imuno-baseadas.

#### 4.1.5.1 Cálculo de $\alpha_i$

Os clones  $C_j$  de um mesmo anticorpo  $\mathbf{Ab}_i$  compartilharão o valor de  $\alpha_i$ .

- I. Para os anticorpos factíveis, a amplitude da mutação é inversamente proporcional à afinidade, ou seja, quanto maior a afinidade do anticorpo-pai (ou quanto menor a classificação), menor será a amplitude da mutação:

$$\alpha_i = \beta_F \cdot \exp[\text{rank}(\mathbf{Ab}_i)/\sigma_F] \quad (4.7)$$

Onde:  $\beta_F$  e  $\sigma_F$  são parâmetros que controlam a amplitude da mutação;  $\beta_F$  é o valor mínimo da mutação (para as soluções não-dominadas) e  $\sigma_F$  é definido para controlar o crescimento da função exponencial.

- II. Para os anticorpos infactíveis, a amplitude da mutação é proporcional ao tamanho da violação da factibilidade, ou seja, quanto maior a violação da factibilidade (ou quanto maior a classificação) do anticorpo-pai  $\mathbf{Ab}_i$  em relação à violação dos outros anticorpos, maior será a mutação.

$$\alpha_i = \beta_I \cdot \exp[\text{rank}(\mathbf{Ab}_i)/\sigma_I] \quad (4.8)$$

$\beta_I$  e  $\sigma_I$  têm a mesma função que  $\beta_F$  e  $\sigma_F$ .  $\sigma_I$  foi definido como a soma de todas as violações de factibilidade, ou seja, considerando os anticorpos infactíveis,  $\sigma_I = \sum_{i \text{ infactível}} \text{rank}(\mathbf{Ab}_i)$ .

#### 4.1.5.2 Cálculo de $\mathbf{d}_j$

Cada clone  $C_j$  possui sua respectiva direção de perturbação  $\mathbf{d}_j$ .

Esta direção é calculada de duas maneiras:

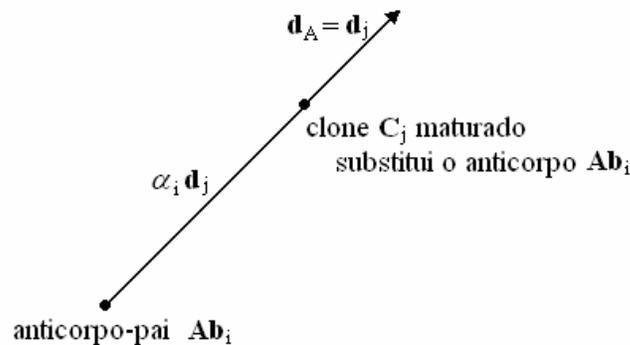
- I. Dois terços dos clones,  $(2 \cdot n_c(\mathbf{Ab}_i)/3)$ , possuem direção aleatória,  $\mathbf{d}_j$  no intervalo  $[-1,1]$ . O objetivo é a exploração do espaço de busca.

II. Para um terço dos clones,  $(n_c(\mathbf{Ab}_i)/3)$ , a direção é calculada da seguinte maneira:

$$\mathbf{d}_j = \mathbf{w} + p_1 \cdot \mathbf{d}_A + p_2 \cdot \mathbf{d}_B \quad (4.9)$$

A definição da direção  $\mathbf{d}_j$  desta maneira, foi inspirada nas propostas de algoritmos de Enxame de Partículas<sup>6</sup> para a resolução de problemas de Otimização Multiobjetivo (Seção 2.2.2.3). Esta é uma tentativa de utilizar as informações do problema e as experiências de iterações anteriores na busca por melhores soluções (soluções não-dominadas).

A direção  $\mathbf{d}_A$  é referente ao anticorpo resultante da iteração imediatamente anterior<sup>7</sup> e é definida da seguinte maneira: se na iteração anterior o anticorpo-pai  $\mathbf{Ab}_i$  foi substituído pelo clone  $\mathbf{C}_j$ , então a direção  $\mathbf{d}_A$  recebe o valor da direção que originou o clone  $\mathbf{C}_j$  (Figura 4.3); ou, se na iteração anterior o anticorpo  $\mathbf{Ab}_i$  não foi substituído (nenhum clone  $\mathbf{C}_j$  era melhor que  $\mathbf{Ab}_i$ ), então a direção  $\mathbf{d}_A$  recebe valor zero.



**Figura 4.3.** A direção  $\mathbf{d}_A$ , no caso em que o clone  $\mathbf{C}_j$  substitui o anticorpo-pai  $\mathbf{Ab}_i$ .

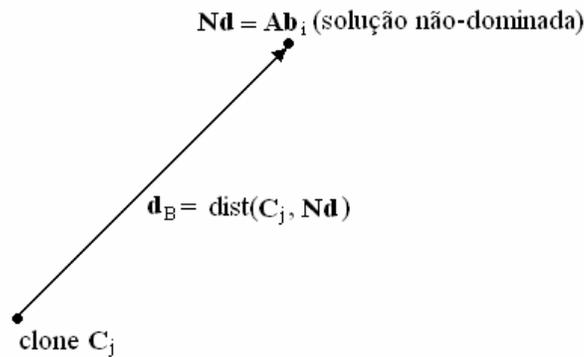
A direção  $\mathbf{d}_B$  refere-se a uma solução não-dominada (líder) e é definida (Figura 4.4) como sendo a direção do clone  $\mathbf{C}_j$  a esse líder<sup>8</sup>.

<sup>6</sup> Nos algoritmos da Enxame de Partículas  $\mathbf{d}_j$  é definida como a velocidade da  $j$ -ésima partícula. É baseada no compartilhamento de informações e experiências anteriores entre as partículas.

<sup>7</sup> Para o cálculo de  $\mathbf{d}_j$ , os algoritmos de Enxame de Partículas, utilizam a direção  $\mathbf{d}_A$  como a velocidade da  $j$ -ésima partícula no ciclo anterior.

<sup>8</sup> Nos algoritmos de Enxame de Partículas, para calcular  $\mathbf{d}_j$ , define-se  $\mathbf{d}_B$  como a distância entre a melhor partícula do enxame e a posição atual da  $j$ -ésima partícula.

Como definido nos algoritmos de Enxame de Partículas, o líder deve guiar a busca para regiões promissoras. Cada clone possuirá seu próprio líder (aleatoriamente escolhido dentre as soluções não-dominadas). Ou seja, os clones serão guiados por diferentes soluções não-dominadas com o intuito de evitar que as soluções se concentrem em uma única região. Além disso, os clones de soluções não-dominadas não possuirão líderes, ou seja,  $\mathbf{d}_B = 0$ . A proposta é que as soluções dominadas sejam guiadas por soluções não-dominadas<sup>9</sup>. Alguns problemas poderiam ocorrer se soluções não-dominadas fossem guiadas por soluções não-dominadas: perda de uma solução potencial; aglomeração de soluções em algumas regiões; porções da Fronteira de Pareto não representadas.



**Figura 4.4.** A direção  $\mathbf{d}_B$ , considerando uma solução não-dominada qualquer como líder.

Cada  $\mathbf{d}_j$  possui diferentes valores de pesos  $p_1$  e  $p_2$ . Ou seja, cada clone possuirá diferentes pesos, aleatoriamente gerados no intervalo  $[0,1]$ . Variando assim a importância de cada uma das direções: a que foi uma direção boa na iteração anterior ( $\mathbf{d}_A$ ) por resultar em uma melhoria (na maturação do clone que substituiu o anticorpo-pai); e uma direção ( $\mathbf{d}_B$ ) que guia a busca para uma região promissora.

Os pesos  $\mathbf{w}$  são empregados para controlar o impacto das direções  $\mathbf{d}_A$  e  $\mathbf{d}_B$  no cálculo da direção de maturação dos clones<sup>10</sup>. Desta maneira,  $\mathbf{w}$  é responsável pelo equilíbrio entre exploração global e local (Parsopoulos e Vrahatis, 2002). Valores grandes de  $\mathbf{w}$  facilitam a exploração global (com a busca por novas áreas),

<sup>9</sup> Caso não exista nenhuma solução não-dominada, considera-se  $\mathbf{d}_B = 0$ . Isto pode ocorrer nos problemas fortemente restritos.

<sup>10</sup> Em algoritmos de Enxame de Partículas é chamado peso de inércia.

enquanto que valores pequenos de  $\mathbf{w}$  facilitam a exploração local (ou seja, o refinamento das áreas de busca já visitadas). Assim  $\mathbf{w}$  é definido como um vetor de variáveis aleatórias geradas com Distribuição Gaussiana com média igual a zero e variância igual a um, de maneira a privilegiar a exploração local (já que a exploração global já está sendo realizada quando a direção  $\mathbf{d}_j$  recebe valores aleatoriamente gerados, caso I).

Para a padronização, todos os vetores de direção  $\mathbf{d}_A$ ,  $\mathbf{d}_B$ ,  $\mathbf{d}_j$  (dos casos, I e II), são normalizados. Após a maturação, todos os clones são classificados de acordo com as regras da Sessão 4.1.3.

### 4.1.6 Infactibilidade

O algoritmo pode gerar soluções infactíveis em dois casos: quando o anticorpo gerado é infactível (Seção 4.1.8) ou quando uma solução infactível ocorre após o processo de maturação. Se, a partir do anticorpo-pai factível obtém-se um herdeiro infactível, reduz-se progressivamente a amplitude da mutação com a regra da bissecção até que o clone se torne factível. Esta técnica pode ser aplicada sem nenhuma hipótese sobre as restrições (linearidade, convexidade).

### 4.1.7 Seleção

O processo de seleção dos clones é inspirado nos princípios da Seleção Clonal (de Castro e Von Zuben, 1999). Não apresenta comportamento retrógrado durante as iterações e privilegia o clone com maior grau de afinidade. Assim, o clone  $\mathbf{C}_j^*$  mais adaptado (de acordo com a classificação realizada a partir das considerações de Pareto-dominância e factibilidade) é selecionado entre os anticorpos do repertório de clones  $\mathbf{C} = \{\mathbf{C}_j; j = 1, \dots, n_c(\mathbf{Ab}_i)\}$  e comparado com o seu anticorpo-pai  $\mathbf{Ab}_i$ . Apenas o melhor é preservado no repertório.

O método é elitista (Michalewicz, 1996), o que assegura que as melhores soluções em cada geração não serão perdidas.

Algumas situações podem ocorrer:

- I. Anticorpos factíveis geram somente anticorpos factíveis (Seção 4.1.11). Desta maneira a seleção é realizada considerando as relações de Pareto-dominância entre o anticorpo-pai e os clones.
- II. Anticorpos infactíveis podem gerar clones factíveis e infactíveis. A seleção é realizada da seguinte maneira:
  - i. Se forem gerados somente clones infactíveis, seleciona-se, entre o anticorpo-pai e os clones, o menos infactível. Ou seja, aquele que possuir menor classificação (Seção 4.1.3) será selecionado.
  - ii. Se forem gerados somente clones factíveis, a seleção é realizada no repertório de clones segundo os conceitos de Pareto-dominância (neste caso o anticorpo-pai é descartado).
  - iii. Se forem gerados alguns clones factíveis e outros infactíveis, descartam-se os infactíveis juntamente com o anticorpo-pai, e o conceito de Pareto-dominância é utilizado para seleção entre os anticorpos factíveis.

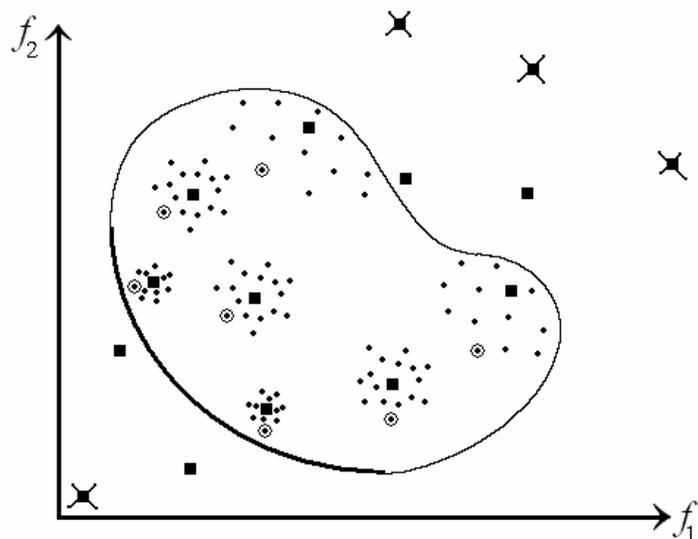
### 4.1.8 Deleção

Ao final de cada iteração, os anticorpos que foram criados infactíveis e não obtiveram melhora ou não satisfizeram o limiar de infactibilidade são excluídos da população. Também ocorre a deleção de anticorpos dominados que não apresentarem melhora durante um certo número de iterações.

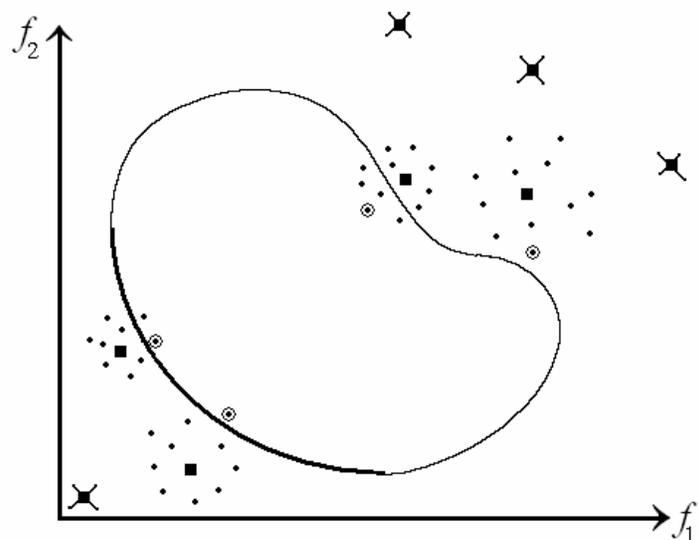
Na teoria imunológica, deleção significa a eliminação dos anticorpos imaturos. Além disso, alguns anticorpos, ao invés de sofrerem a deleção, podem apresentar uma edição de receptores: desenvolvimento de novos receptores através de recombinação genética.

As Figuras 4.5 e 4.6 ilustram os processos de clonagem, mutação e seleção do melhor clone para cada anticorpo, primeiro para os anticorpos factíveis e depois para os anticorpos infactíveis. Além disso, a deleção dos anticorpos infactíveis que não satisfazem

o limiar de infactibilidade é realizada. Todas as ilustrações são baseadas no espaço das funções e na Fronteira de Pareto apresentadas no Capítulo 2 (Figuras 2.3 e 2.4).



**Figura 4.5.** Exemplo de clonagem e maturação, para anticorpos factíveis.



**Figura 4.6.** Exemplo de clonagem e maturação, para anticorpos infactíveis.

A Figura 4.7 ilustra o repertório após os processos de clonagem, mutação, seleção e deleção apresentados acima.

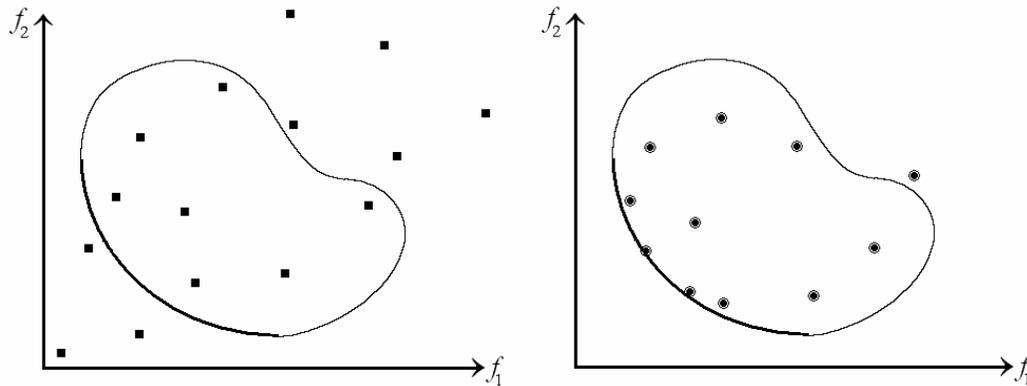


Figura 4.7. Continuação do exemplo, resultado da Seleção Clonal.

### 4.1.9 Diversidade

Os Sistemas Imunológicos são inerentemente capazes de manter a diversidade da população. Módulos com algumas semelhanças com *niching* (nichos) e *fitness sharing* (compartilhamento de fitness) são partes intrínsecas do algoritmo (Coelho e Von Zuben, 2006).

Para ajudar na manutenção da diversidade durante as iterações do algoritmo, mecanismos de deleção (Seção 4.1.8), supressão (Seção 4.1.11) e inserção de novos anticorpos (Seção 4.1.12), de acordo com os conceitos da imunologia, são propostos.

### 4.1.10 Memória

Como foi explicado anteriormente, a memória é uma característica inerente ao Sistema Imunológico.

No algoritmo proposto, utiliza-se a definição de um arquivo de memória. Essa memória (externa) é utilizada no armazenamento de soluções que possam representar possíveis ótimos locais (soluções que não apresentam melhora durante um certo número de iterações ( $it_m$ ), mas que são factíveis e não-dominadas). Não seria conveniente carregar esse tipo de solução durante todas as iterações, mas ao mesmo tempo existe o interesse de armazená-las, ou para um tratamento especial (outro tipo de mutação, na tentativa de melhora) ou para a possibilidade de pertencer ao conjunto-solução.

### 4.1.11 Supressão

Foi definida uma métrica, inspirada em Kukkonen e Deb (2006), baseada em quão cheia está uma região (*Crowding Distance*).

Para a identificação dos anticorpos a serem suprimidos, pode-se definir um limiar de aglomeração  $\text{lim}_{\text{crowd}}$ . Assim, apenas os anticorpos que identificam uma determinada região em potencial ( $\text{Crowd}(\mathbf{Ab}_i) \leq \text{lim}_{\text{crowd}}$ ) são mantidos.

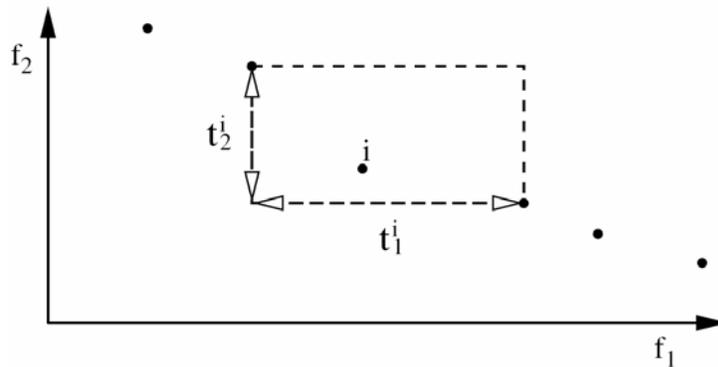
É importante ressaltar que a supressão<sup>11</sup> proposta para o algoritmo pode ocorrer tanto no espaço nas funções-objetivo, quanto no espaço das variáveis. Porém, a proposta de Supressão fica restrita a espaços bidimensionais<sup>12</sup>.

#### 4.1.11.1 Crowding Distance

Para cada anticorpo  $\mathbf{Ab}_i$ , podemos definir sua Crowding Distance como:

$$\text{Crowd}(\mathbf{Ab}_i) = t_1^i + t_2^i \quad (4.10)$$

Sendo que  $t_1^i$  e  $t_2^i$  definem uma vizinhança (limitada pelos anticorpos mais próximos do anticorpo  $i$ ), como ilustrado na Figura 4.8.



**Figura 4.8.** Exemplo bidimensional, cálculo do Crowding Distance do anticorpo  $i$ .

<sup>11</sup> A supressão não ocorre em todas as iterações. Para diminuir o custo computacional, a supressão é aplicada com frequência  $t_s$ .

<sup>12</sup> Os casos que abordam dimensões maiores serão estudados futuramente.

### 4.1.12 Repertório com tamanho dinâmico

O tamanho do repertório de anticorpos é limitado em  $n_T$  anticorpos. Devido aos mecanismos de deleção (Seção 4.1.8) e supressão (Seção 4.1.11) o tamanho do repertório pode diminuir. Isto também ocorre no Sistema Imunológico biológico, onde o número total de linfócitos não permanece absolutamente constante.

Quando o número de anticorpos no repertório for menor ou igual ao limiar  $n_{T0}$ , novos anticorpos são inseridos. Esta inserção está de acordo com o conceito de Repertório Disponível, anticorpos que não estão ativos no momento, mas podem ser utilizados (de Castro, 2001).

O processo de substituição de indivíduos não clonados pode ser entendido como uma *exploração* no espaço de busca; já o princípio de Seleção Clonal permite uma busca local em diferentes direções, a *exploração* (Coello e Cortés, 2005).

### 4.1.13 Maturação Final

O método de seleção proposto foi baseado no conceito de Pareto-dominância. Ironicamente, não necessariamente todas as soluções do conjunto não-dominado final se aproximam suficientemente da verdadeira Fronteira de Pareto, isto devido aos vários níveis de não-dominância. Isto significa que nem todas as soluções não-dominadas encontradas pelo algoritmo serão soluções Pareto-ótimas. A Pareto-otimalidade de cada uma das soluções encontradas pelo algoritmo no final do processo pode ser checada através de uma perturbação aleatória, localmente (tanto no repertório corrente quanto na memória).

$$\mathbf{Ab}' = \mathbf{Ab} + \alpha \cdot \mathbf{d} \quad (4.11)$$

Considerando  $\mathbf{d} = (d_1, d_2, \dots, d_n)$  uma direção aleatória, onde  $d_i \in [-1, 1]$ ,  $i = 1, 2, \dots, n$ ; e  $\alpha \in (0, 1)$  aleatório.

Finalmente, o repertório final é formado pelas soluções factíveis e não-dominadas, analisando simultaneamente o repertório da última iteração e o arquivo de memória.

## 4.2 Pseudocódigo

O algoritmo possui os seguintes parâmetros de entrada:

Parâmetros de Entrada	
$n_T$	Tamanho do repertório.
$c_{min}$	Número mínimo de clones.
$c_{max}$	Número máximo de clones.
$\beta_I$	Mutação mínima para anticorpos factíveis.
$\beta_F$	Mutação mínima para anticorpos infactíveis.
$n_{T0}$	Limite mínimo para o tamanho do repertório.
$lim_{crowd}$	Limiar mínimo de aglomeração, para supressão.
$t_S$	Frequência de ocorrência da supressão.
$lim_{infac}$	Limiar de infactibilidade aceitável.
$It_{total}$	Número total de iterações.
$it_m$	Iterações sem melhora, para deleção ou inserção na memória.
$n_m$	Tamanho da memória.

**Tabela 4.1.** Parâmetros de Entrada.

O pseudocódigo do algoritmo é apresentado a seguir:

[teste] = Escolher Problema Teste()

$[n_T, c_{min}, c_{max}, \beta_I, \beta_F, n_{T0}, lim_{crowd}, lim_{infac}, It_{total}, t_S, it_m, n_m]$  = Definir Parâmetros(teste)

[**Ab**] = Inicializar Repertório( $n_T$ )

Para  $t = 0$  até  $It_{total}$  faça:

[**F**, **R**] = Avaliar funções e restrições(**Ab**,  $n_T$ )

[**fac**, **D**, **ND**,  $\sigma_I, \sigma_F$ ] = Avaliar factibilidade e dominância(**Ab**, **F**, **R**,  $n_T$ )

Para  $i = 0$  até  $n_T$

[rank(**Ab<sub>i</sub>**)] = Classificar anticorpos(**Ab**, **fac**, **D**, **ND**,  $n_T$ )

Se  $(rank(\mathbf{Ab}_i) / \sum_{j \text{ infactível}} rank(\mathbf{Ab}_j) \leq lim_{infac})$  Então

[ $n_c(\mathbf{Ab}_i)$ ] = Selecionar e calcular número de clones(rank(**Ab<sub>i</sub>**),  $c_{min}$ ,  $c_{max}$ , **fac**)

[ $\alpha_i$ ] = Calcular a amplitude da mutação(**fac**,  $\beta_I$ ,  $\beta_F$ ,  $\sigma_I$ ,  $\sigma_F$ , rank(**Ab<sub>i</sub>**))

[**C**] = Clonar anticorpo(**Ab<sub>i</sub>**)

Para  $j = 0$  até  $n_c(\mathbf{Ab}_i)$

[**d<sub>j</sub>**] = Calcular a direção da mutação(**Ab<sub>i</sub>**)

[**C<sub>j</sub>**] = Maturar clone(**Ab<sub>i</sub>**,  $\alpha_i$ , **d<sub>j</sub>**)

fim-Para

[**C\***] = Selecionar o melhor clone(**C**)

```

    [Abi] = Selecionar o melhor anticorpo(C*, Abi)
  fim-Se
fim-Para
[Ab, nT] = Deleção(Ab, fac)
Se (resto(t/tξ) = 0) Então
  [Ab, nT] = Supressão(Ab, nT, Lcrowd)
fim-Se
[M] = Atualizar Memória(ND, itm, nm)
Se (nT = nT0) Então
  [Ab, nT] = Inserir novos anticorpos aleatórios(Ab, nT)
fim-Se
fim-Para
[Ab, M] = Maturação Final(Ab, M, nT, nm)
[fac, ND] = Avaliar factibilidade e dominância(Ab, M, F, R, nT, nm)
[Abfinal, nfinal] = Seleção final(fac, ND, Ab, M, nT, nm)

```

Basicamente, os passos do algoritmo são:

- 1) Escolha do problema a ser resolvido;
- 2) Definição dos parâmetros apresentados na Tabela 4.1;
- 3) Inicialização aleatória do repertório;
- 4) Para cada iteração:
  - i) Avaliar as funções e restrições de acordo com o repertório corrente **Ab**, gerando as estruturas **F** e **R**;
  - ii) Avaliar a factibilidade e a dominância do repertório **Ab** de acordo com as estruturas **F** e **R**;
  - iii) Classificar os anticorpos de acordo com critérios de factibilidade e dominância;
  - iv) Seleção para clonagem dos anticorpos que satisfazem o limiar de infactibilidade;
  - v) Para cada anticorpo **Ab<sub>i</sub>** selecionado para clonagem, calcular o número de clones  $n_c(\mathbf{Ab}_i)$  e a amplitude de mutação  $\alpha_i$ ;
  - vi) Clonagem dos anticorpos selecionados para clonagem;
  - vii) Cálculo da direção de maturação;
  - viii) Maturação dos clones;

- ix) Selecionar o melhor entre clones e anticorpos-pai;
  - x) Processo de deleção dos anticorpos excedentes;
  - xi) Supressão;
  - xii) Atualização da memória **M**;
  - xiii) Inserção de novos anticorpos aleatórios;
- 5) Maturação final;
- 6) Seleção final das soluções não-dominadas **Ab<sub>final</sub>** do repertório corrente e da memória.

# Capítulo 5

## Experimentos e Resultados

A maioria dos testes abordou problemas restritos com dois objetivos. Até mesmo neste caso, as características essenciais da Otimização Multiobjetivo podem ser notadas e estudadas. Além disso, é necessário um bom conhecimento dos mecanismos, componentes e parâmetros do Sistema Imunológico Artificial que podem ou não funcionar em problemas bi-objetivos antes que problemas maiores sejam explorados. Problemas com três e cinco objetivos também foram submetidos aos testes, com o intuito de ilustrar a generalidade do algoritmo.

As seguintes observações em relação aos problemas restritos devem ser ressaltadas:

- As restrições podem introduzir uma complexidade adicional na resolução do problema devido à indução de regiões inactiváveis no espaço de busca, além de obstruir o progresso do algoritmo na busca pela Fronteira de Pareto global (Deb, 1999).
- Praticamente todos os problemas reais de otimização apresentam pelo menos uma restrição.
- Existem poucos estudos em Algoritmos Bio-inspirados especificamente designados para lidar com restrições.

Os problemas testes foram divididos em quatro grupos:

1º - Problemas bi-objetivos, comumente estudados na literatura, com restrições consideravelmente simples e mais relaxadas;

2º - Problemas bi-objetivos, com restrições mais elaboradas;

3º - Problemas com três e cinco funções-objetivo;

4º - Problema com três funções-objetivo, fortemente restrito.

Quando se mede o desempenho de um algoritmo de resolução de problemas de Otimização Multiobjetivo, três aspectos relevantes devem ser levados em consideração (Cortés, 2004):

- Aproximação da verdadeira Fronteira de Pareto.
- Dispersão uniforme das soluções.
- Maximização da quantidade de elementos diferentes na verdadeira Fronteira de Pareto.

Para medir o desempenho do algoritmo proposto neste trabalho em relação à dispersão das soluções será utilizada a métrica proposta por Schott (1995). Comparações gráficas em relação à verdadeira Fronteira de Pareto, gerada através de métodos de enumeração e disponíveis da literatura, também serão realizadas. O Sistema Imunológico proposto adequará a quantidade de elementos do repertório final através dos mecanismos de supressão, deleção, inserção e repertório dinâmico (Seção 4.1).

**Métrica de Dispersão:** Mede a variância da distância de cada membro do Conjunto Pareto-ótimo (formado por  $n_{\text{final}}$  anticorpos) em relação ao seu vizinho mais próximo. Quanto menor o valor de  $D$ , melhor é a distribuição de soluções não-dominadas encontradas pelo algoritmo.

$$D = \sqrt{\frac{1}{n_{\text{final}}} \sum_{i=1}^{n_{\text{final}}} (\bar{d} - d_i)^2} \quad (5.1)$$

$$d_i = \min_{j \neq i} \left( \sum_{k=1}^m |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})| \right) \quad (5.2)$$

$$\bar{d} = \frac{\sum_{i=1}^{n_{\text{final}}} d_i}{n_{\text{final}}} \quad (5.3)$$

O algoritmo NSGA2 (Seção 2.2.2.1) foi escolhido para a comparação dos resultados. Além de fazer parte dos algoritmos representativos do estado da arte na área de Otimização Multiobjetivo, o NSGA2 também é o mais comumente escolhido por outras abordagens para comparação. A definição dos parâmetros e resultados do NSGA2, aqui reportados, também foram apresentados em Deb *et. al.* (2002), Coello e Cortés (2005), Freschi e Repetto (2005), Reddy e Kumar (2007), Deb *et. al.* (2000).

Para cada exemplo foram realizadas 10 rodadas de testes. Os dados das tabelas apresentadas nas seções subseqüentes resumem todas as rodadas. Os gráficos apresentados são referentes ao melhor teste realizado.

Os testes foram realizados em uma plataforma Linux, processador Intel Core 2Duo 1.73 GHz e 2 Gb de memória.

## 5.1 Variação dos parâmetros

Durante a execução do algoritmo, algumas considerações sobre as variações dos parâmetros de entrada foram realizadas:

- $n_T$ : O número total de anticorpos no repertório foi definido de acordo com a dificuldade do problema. Para os problemas da primeira e segunda etapas de testes, definiu-se  $n_T = 100$ ; estes problemas apresentam espaço de busca mais simples (menos variáveis, duas funções-objetivo, menos restrições). Para os problemas da terceira e quarta etapa de testes definiu-se  $n_T = 100$  e  $n_T = 200$ , respectivamente para os problemas de 3 e 5 objetivos e  $n_T = 500$  para o problema fortemente restrito. Este é um parâmetro importante para o problema, já que os anticorpos são responsáveis pela exploração do espaço de busca. Com poucos anticorpos a busca seria ineficiente; por outro lado, um número muito grande de anticorpos resultaria num alto custo computacional (memória e tempo de execução).
- $c_{min}$  (número mínimo de clones) e  $c_{max}$  (número máximo de clones). Para a definição destes valores foram realizados testes variando entre 10 e 100 o número de clones gerados. Percebeu-se que, quanto menor o número de clones, menos anticorpos não-dominados são encontrados no final do processo, porém o tempo total de execução é

menor; e quanto maior o número de clones, mais anticorpos não-dominados são encontrados no final do processo, mas o tempo de execução aumenta consideravelmente. Decidiu-se por equilibrar a quantidade de anticorpos no repertório final e o tempo total de execução. Considerando também que o número de clones varia de acordo com a afinidade do anticorpo-pai (Seção 4.1.4), definiu-se  $c_{min} = 40$  e  $c_{max} = 60$ .

- $\beta_I$  e  $\beta_F$ : O valor mínimo da amplitude da mutação para anticorpos inactíveis e factíveis é definido de acordo com as características do problema. Estes valores devem ser cuidadosamente definidos. Valores muito altos resultariam em passos muito grandes e a busca se assemelharia a uma busca aleatória; valores muito pequenos resultariam em passos muito pequenos, prejudicando a convergência do processo. Além disso, considerou-se que, quanto maior a afinidade do anticorpo-pai, menor a amplitude da mutação (Seção 4.1.5.1).
- $n_{T0}$ : O limiar mínimo de anticorpos da população está diretamente ligado ao parâmetro  $n_T$ , portanto varia de acordo com o problema, mas sem prejudicar a exploração do espaço de busca. Considerou-se  $n_{T0}$  igual a 70% de  $n_T$ . Desta maneira o tamanho do repertório é dinâmico, de acordo com as necessidades do problema (Seção 4.1.12).
- $lim_{crowd}$  define o limiar de aglomeração de uma região; quanto maior o limiar mais espalhadas estarão as soluções.
- A frequência de supressão,  $t_s$ , é definida considerando que o algoritmo deve ter tempo suficiente de explorar a vizinhança de cada solução antes da supressão de um anticorpo.
- $lim_{infac}$ : o limiar de inactibilidade permitido para que as soluções sejam selecionadas para clonagem varia de acordo com as características do problema. Problemas com restrições mais relaxadas possuem limiar de inactibilidade menor (próximo ou igual a 0), de maneira a privilegiar apenas as soluções factíveis e economizar recursos computacionais; problemas fortemente restritos possuem limiar de inactibilidade maior (próximo ou igual a 1), de maneira a tratar (tentar factibilizar) algumas soluções inactíveis que possam ser interessantes.

- $It_{total}$ : O número total de iterações foi definido de maneira a permitir uma busca eficiente (exploração do espaço de busca e convergência das soluções).
- $it_m$  é o número de iterações que um anticorpo pode passar sem melhora para depois ser retirado do repertório corrente. Ou seja, se um anticorpo é dominado (factível ou infactível) e não obteve melhora durante  $it_m$  iterações seguidas, ele é retirado do repertório definitivamente (Seção 4.1.11); se um anticorpo é não-dominado (e factível) e não obteve melhora durante  $it_m$  iterações seguidas, ele é retirado do repertório e alocado à memória (Seção 4.1.9). Portanto,  $it_m$  deve ser definido cuidadosamente, pois ele representará a quantidade de iterações necessárias para a convergência das soluções (já que o repertório final, ou seja, o conjunto final de soluções não-dominadas, é formado pelos anticorpos da memória). Um valor pequeno para  $it_m$  resultará em convergência prematura e um valor grande resultará em maior gasto computacional e menor eficiência (pois os anticorpos ficarão no repertório corrente por maior tempo, impedindo a inserção de novos anticorpos).
- O tamanho da memória  $n_m$  é definido com o intuito de se armazenar um número grande de soluções não-dominadas.

## 5.2 Primeira Etapa de Testes

Na primeira etapa de testes foram considerados 6 problemas.

Para estes problemas foram definidos os parâmetros da Tabela 5.1, respectivamente: tamanho total do repertório, número total de iterações, frequência de iterações em que a supressão é aplicada, limiar de aglomeração, número mínimo de clones, número máximo de clones, limiar inferior para o tamanho da população, número de iterações sem melhora para armazenamento na memória e tamanho da memória. O mecanismo de Supressão (Seção 4.1.11) foi aplicado no espaço das funções-objetivo.

$n_T$	$It_{total}$	$t_s$	$lim_{crowd}$	$c_{min}$	$c_{max}$	$n_{t0}$	$it_m$	$n_m$
100	1000	200	0,0001	40	60	$0,7 * n_T$	50	$6 * n_T$

**Tabela 5.1.** Parâmetros comuns da primeira etapa de testes.

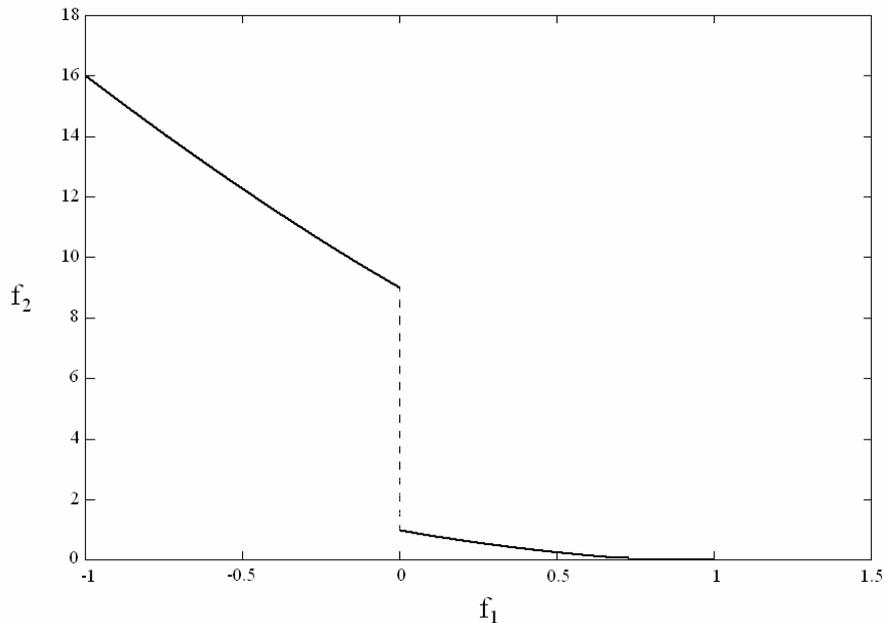
**Exemplo 1:** O primeiro problema proposto por Schaffer (1984) possui duas funções-objetivo e a Fronteira de Pareto está fragmentada em 2 segmentos.

$$\text{Minimizar } f_1(x) = \begin{cases} -x & \text{se } x \leq 1 \\ -2+x & \text{se } 1 < x \leq 3 \\ 4-x & \text{se } 3 < x \leq 4 \\ -4+x & \text{se } x > 4 \end{cases}$$

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

$$x \in [-5, 10]$$

A Figura 5.1 ilustra a verdadeira Fronteira de Pareto deste exemplo.



**Figura 5.1.** Verdadeira Fronteira de Pareto - Exemplo 1.

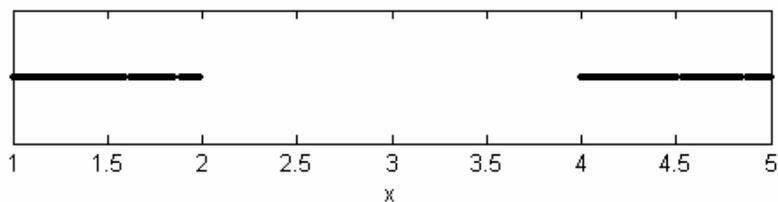
Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.2):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,3	0,1	0,1

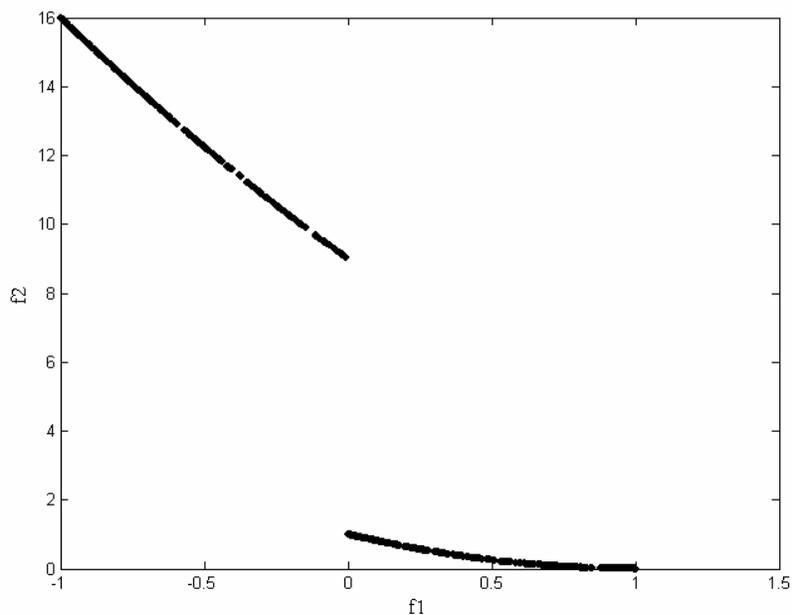
**Tabela 5.2.** Parâmetros específicos - Exemplo 1.

A Figura 5.2 apresenta o conjunto de soluções não-dominadas encontradas pelo algoritmo proposto. A Figura 5.3 apresenta a respectiva Fronteira de Pareto.

O tempo de execução da melhor rodada foi de 41 segundos e 590 soluções não-dominadas foram encontradas.



**Figura 5.2.** Conjunto de soluções não-dominadas obtido - Exemplo 1.



**Figura 5.3.** Fronteira de Pareto obtida - Exemplo 1.

A verdadeira Fronteira de Pareto foi encontrada e tanto no espaço das funções-objetivo, quanto no espaço das variáveis, o algoritmo obteve uma distribuição uniforme das soluções.

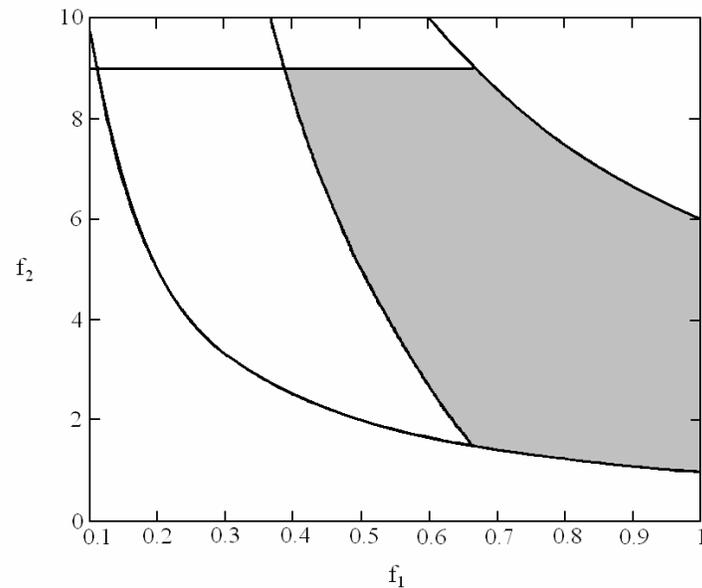
**Exemplo 2:** No segundo problema (Deb *et. al.*, 2002) bi-objetivo, parte da região Pareto-ótima irrestrita não é factível. Assim, a região Pareto-ótima restrita será formada pela concatenação da primeira restrição e parte da região Pareto-ótima irrestrita (Figura 5.4).

Minimizar	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = (1 + x_2)/x_1$
s.a.	$g_1(\mathbf{x}) = x_2 + 9x_1 \geq 6$
	$g_2(\mathbf{x}) = -x_2 + 9x_1 \geq 1$
$x_1 \in [0.1, 1.0]$	$x_2 \in [0, 5]$

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.3):

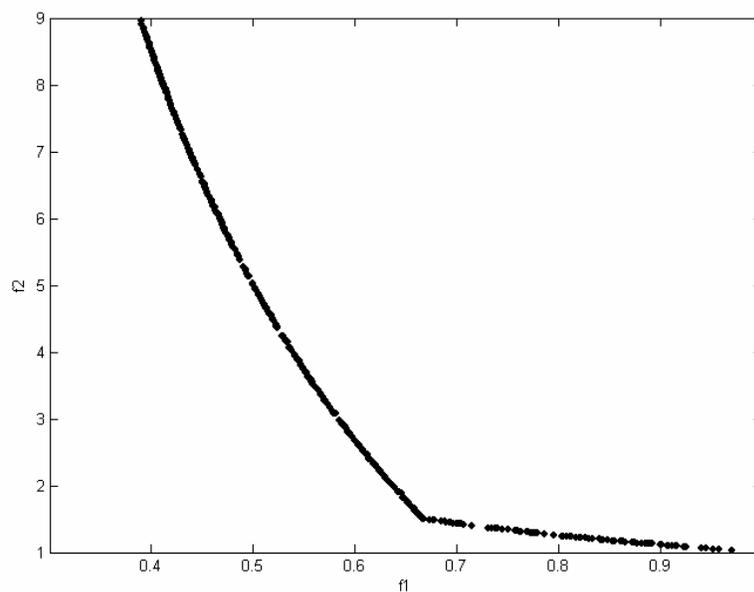
$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
[0,05; 0,3]	[0,01; 0,1]	0,5

**Tabela 5.3.** Parâmetros específicos - Exemplo 2.



**Figura 5.4.** Verdadeira Fronteira de Pareto - Exemplo 2.

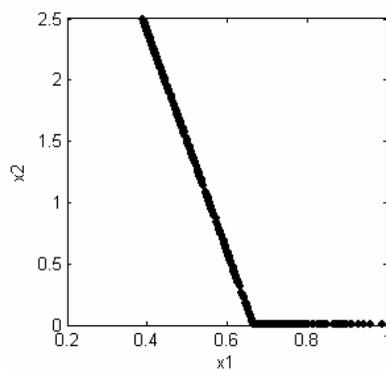
A Figura 5.5 apresenta Fronteira de Pareto obtida.



**Figura 5.5.** Fronteira de Pareto obtida - Exemplo 2.

A Figura 5.6 apresenta o respectivo conjunto de soluções não-dominadas.

O tempo de execução da melhor rodada foi de 37 segundos e 420 soluções não-dominadas foram encontradas.



**Figura 5.6.** Conjunto de soluções não-dominadas obtido - Exemplo 2.

Tanto no espaço das soluções quanto no espaço das funções-objetivo, as soluções se distribuíram uniformemente na região Pareto-ótima.

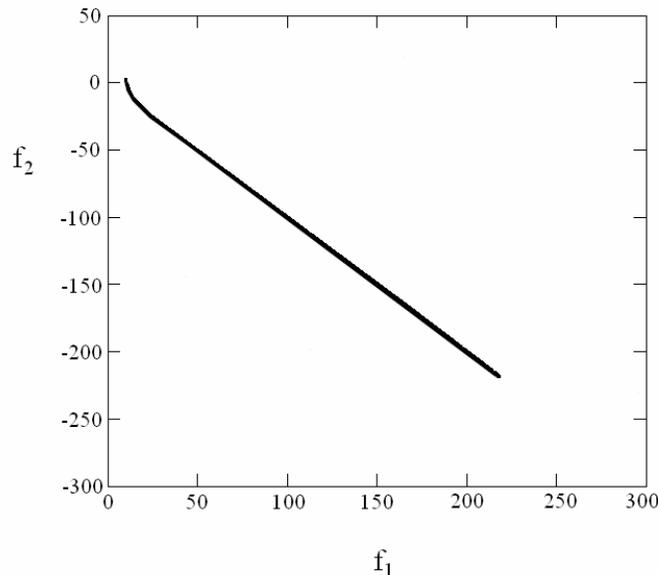
**Exemplo 3:** O terceiro problema bi-objetivo foi utilizado no estudo original do algoritmo NSGA (Seção 2.2.2.1). O Conjunto Pareto-ótimo restrito é um subconjunto do conjunto Pareto-ótimo irrestrito e está ilustrado na Figura 5.7.

Minimizar	$f_1(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$
	$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$
s.a.	$g_1(\mathbf{x}) = x_1^2 + x_2^2 \leq 225$
	$g_2(\mathbf{x}) = x_1 - 3x_2 \leq 10$
	$x_1, x_2 \in [-20, 20]$

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.4):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,5	0,3	0,5

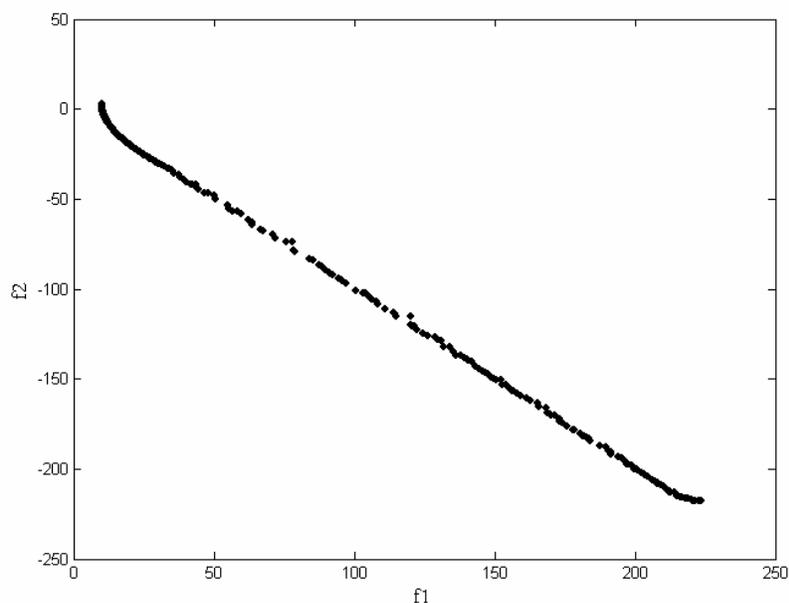
**Tabela 5.4.** Parâmetros específicos - Exemplo 3.



**Figura 5.7.** Verdadeira Fronteira de Pareto - Exemplo 3.

A seguir, a Figura 5.8 apresenta a Fronteira de Pareto obtida para o Exemplo 3.

O tempo de execução da melhor rodada foi de 44 segundos e 299 soluções não-dominadas foram encontradas.



**Figura 5.8.** Fronteira de Pareto obtida - Exemplo 3.

Observou-se que as melhores soluções (em relação à convergência e distribuição) se encontraram nas extremidades da Fronteira de Pareto. A figura da distribuição dos pontos encontrados no espaço das variáveis foi suprimida (as soluções se encontraram distribuídas de forma aleatória, numa pequena região contínua do espaço).

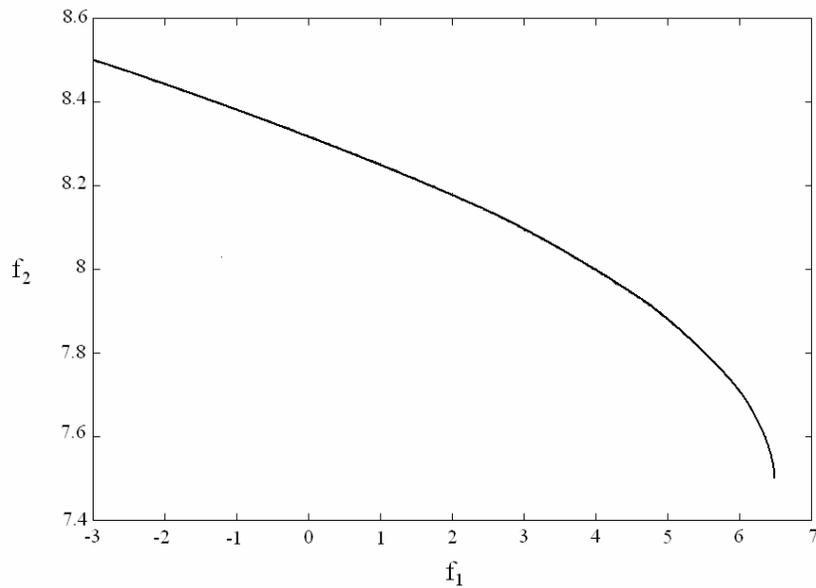
**Exemplo 4:** O quarto problema bi-objetivo, proposto por Kita *et. al.* (1996), possui Fronteira de Pareto côncava (Figura 5.9).

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.5):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,06	0,04	0,7

**Tabela 5.5.** Parâmetros específicos - Exemplo 4.

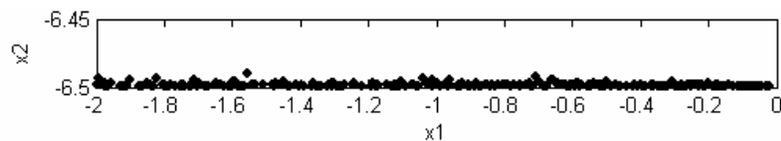
$$\begin{aligned}
 &\text{Maximizar} && f_1(\mathbf{x}) = x_1^2 + x_2 \\
 &&& f_2(\mathbf{x}) = \frac{1}{2}x_1 + x_2 + 1 \\
 &\text{s.a.} && g_1(\mathbf{x}) = -\frac{13}{2} + \frac{1}{6}x_1 + x_2 \leq 0 \\
 &&& g_2(\mathbf{x}) = -\frac{15}{2} + \frac{1}{2}x_1 + x_2 \leq 0 \\
 &&& g_3(\mathbf{x}) = -30 + 5x_1 + x_2 \leq 0 \\
 &&& x_1, x_2 \geq 0
 \end{aligned}$$



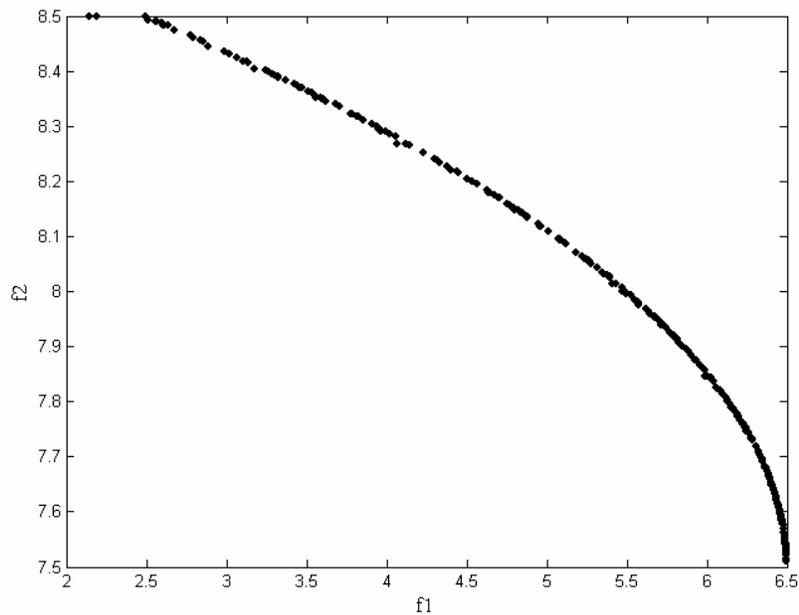
**Figura 5.9.** Verdadeira Fronteira de Pareto - Exemplo 4.

O conjunto de soluções não-dominadas encontradas e a respectiva Fronteira de Pareto estão apresentados a seguir (Figura 5.10 e Figura 5.11).

O tempo de execução da melhor rodada foi de 50 segundos e 294 soluções não-dominadas foram encontradas.



**Figura 5.10.** Conjunto de soluções não-dominadas obtido - Exemplo 4.



**Figura 5.11.** Fronteira de Pareto obtida - Exemplo 4.

O algoritmo não foi capaz de encontrar uma porção da Fronteira de Pareto, compreendida no intervalo  $[-3; 2,5]$ . Entre  $[2,5; 6]$ , o algoritmo convergiu para a verdadeira Fronteira de Pareto, com melhor distribuição na região com maiores valores de  $f_1$  (lado direito da Figura 5.11). Esta pode ser uma dificuldade encontrada pelo algoritmo: o tratamento de problemas com Fronteira de Pareto côncava. Além disso, observa-se que a Fronteira de Pareto possui um formato “platô”, enquanto  $f_1$  varia 10 unidades (de -3 a 7),  $f_2$  varia apenas 1 unidade (de 7,5 a 8,5). Portanto, a parte ainda mais estreita (com  $f_1$  variando de -3 a 2) possui menos anticorpos, que não foram capazes de explorar a região rumo à Fronteira de Pareto.

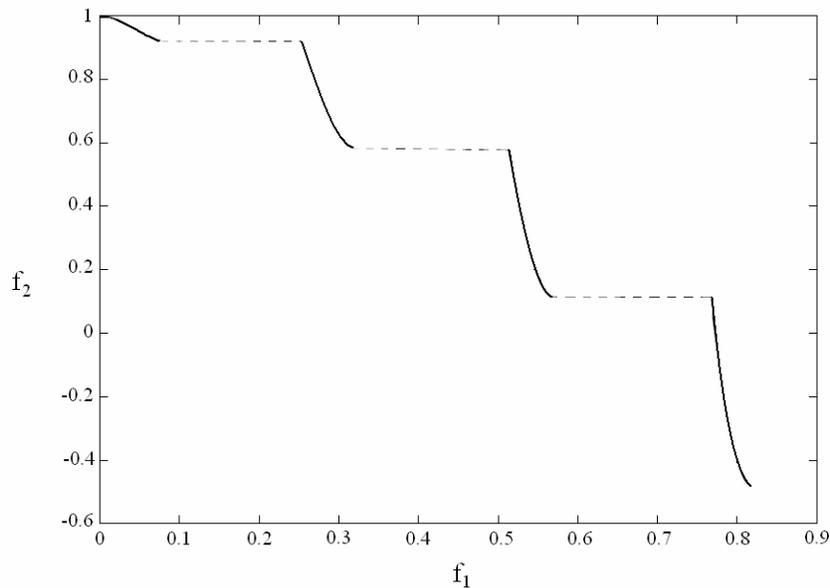
**Exemplo 5:** No quinto problema bi-objetivo, proposto por Deb (1999), o Conjunto Pareto-ótimo e a verdadeira Fronteira de Pareto são desconexos (a Fronteira de Pareto é formada por 4 curvas, Figura 5.12).

$$\begin{aligned} \text{Minimizar } & f_1(\mathbf{x}) = x_1 \\ & f_2(\mathbf{x}) = (1 + 10x_2) \cdot \left[ 1 - \left( \frac{x_1}{1 + 10x_2} \right)^2 - \frac{x_1}{1 + 10x_2} \sin(8\pi x_1) \right] \\ & x_1, x_2 \in [0, 1] \end{aligned}$$

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.6):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,7

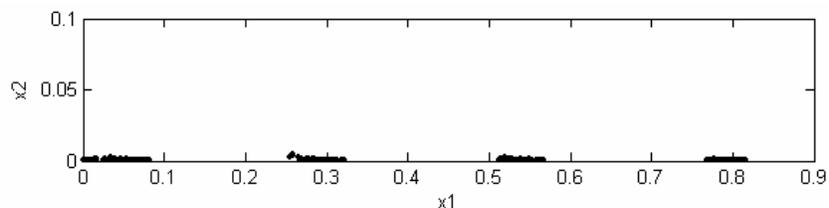
**Tabela 5.6.** Parâmetros específicos - Exemplo 5.



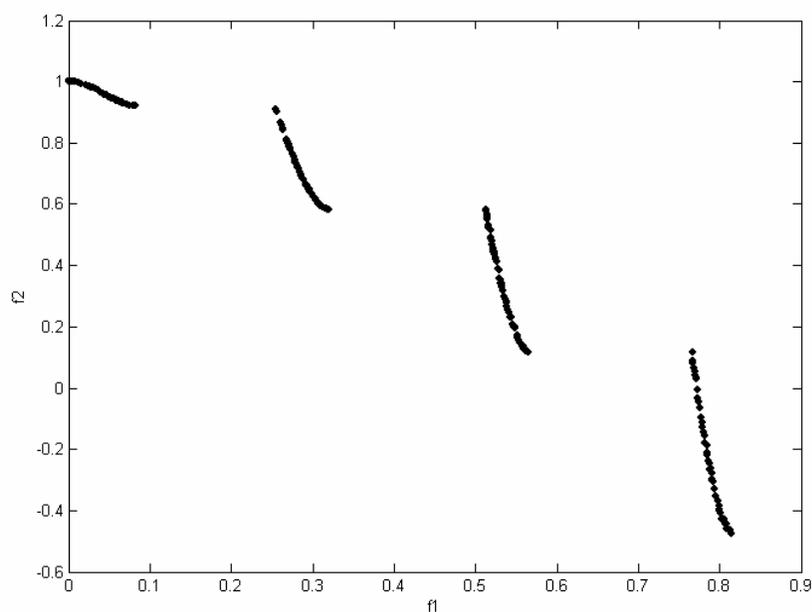
**Figura 5.12.** Verdadeira Fronteira de Pareto - Exemplo 5.

A Fronteira de Pareto e o respectivo conjunto de soluções não-dominadas encontradas estão apresentados a seguir (Figura 5.13 e Figura 5.14).

O tempo de execução da melhor rodada foi de 51 segundos e 228 soluções não-dominadas foram encontradas.



**Figura 5.13.** Conjunto de soluções não-dominadas obtido - Exemplo 5.

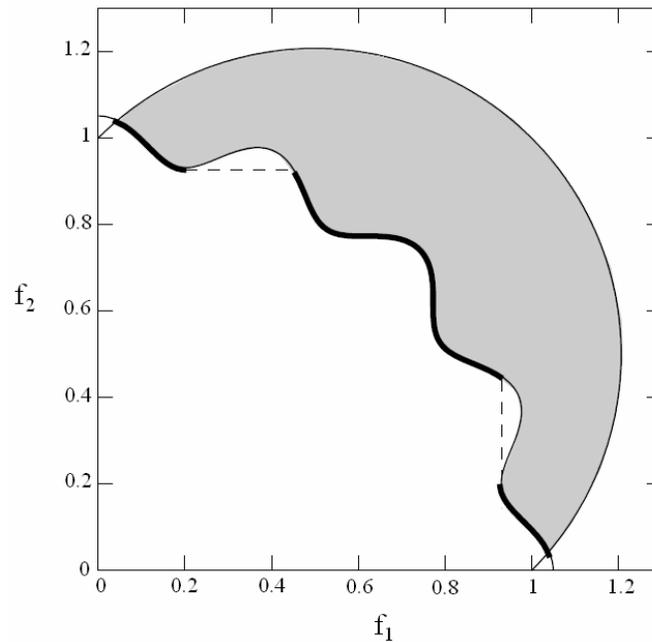


**Figura 5.14.** Fronteira de Pareto obtida - Exemplo 5.

O algoritmo conseguiu convergir para a verdadeira Fronteira de Pareto, com distribuição uniforme de soluções, tanto no espaço das funções-objetivo, quanto no espaço das soluções, ambos desconexos.

**Exemplo 6:** O sexto problema bi-objetivo, proposto por Tanaka (1995), possui região Pareto-ótima descontínua, localizada totalmente no contorno da primeira restrição (Figura 5.15).

Minimizar	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = x_2$
s.a.	$g_1(\mathbf{x}) = -x_1^2 - x_2^2 + 1 + 0.1 \cos(16 \arctan(x_1/x_2)) \leq 0$
	$g_2(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$
	$x_1, x_2 \in [0, \pi]$



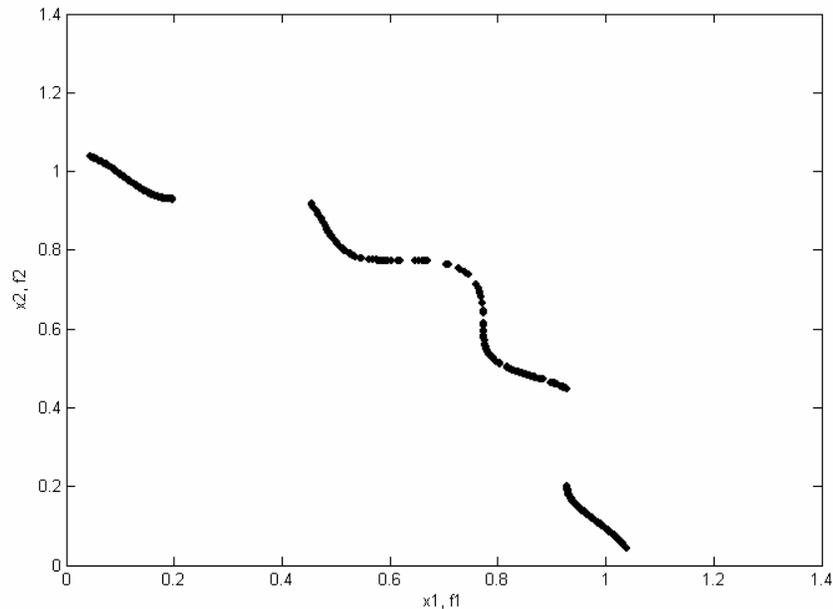
**Figura 5.15.** Verdadeira Fronteira de Pareto - Exemplo 6.

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes máximas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.7):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,02	0,008	0,7

**Tabela 5.7.** Parâmetros específicos - Exemplo 6.

A Fronteira de Pareto e o respectivo conjunto de soluções não-dominadas encontradas estão apresentados na Figura 5.16.



**Figura 5.16.** Fronteira de Pareto e soluções não-dominadas obtidos - Exemplo 6.

Para o Exemplo 6, as melhores soluções (em relação à distribuição) se localizaram nas extremidades na Fronteira (e soluções não tão bem distribuídas na região côncava da Fronteira).

O tempo de execução da melhor rodada foi de 49 segundos e 452 soluções não-dominadas foram encontradas.

Com a análise gráfica foi possível analisar o desempenho do algoritmo na tentativa de aproximação da Fronteira de Pareto.

A análise em relação à distribuição das soluções através da Fronteira, pode ser melhor realizada através da Métrica de Dispersão (D) discutida anteriormente.

Os resultados obtidos pelo algoritmo proposto (Tabela 5.8) podem ser comparados com os resultados obtidos pelo algoritmo NSGA2 (Tabela 5.9). A seguir, a média e o desvio da dispersão, a melhor e a pior dispersão; considerando as 10 rodadas de testes realizadas para cada problema.

Algoritmo Proposto	Média	Desvio	Melhor	Pior
<b>Exemplo 1</b>	0.01427	0.00133	0.0122	0.0163
<b>Exemplo 2</b>	0.01045	0.00069	0.0097	0.0118
<b>Exemplo 3</b>	1.04862	0.09437	0.9323	1.2144
<b>Exemplo 4</b>	0.0801	0.11218	0.0114	0.362
<b>Exemplo 5</b>	0.00662	0.0011	0.0051	0.0089
<b>Exemplo 6</b>	0.00237	0.00033	0.0019	0.0029

Tabela 5.8. Métrica de Dispersão - Algoritmo Proposto.

NSGA 2	Média	Desvio	Melhor	Pior
<b>Exemplo 1</b>	0.04422	0.00292	0.03906	0.05035
<b>Exemplo 2</b>	0.0465	0.0032	0.04126	0.05265
<b>Exemplo 3</b>	1.5969	0.1307	1.25455	1.68121
<b>Exemplo 4</b>	0.02201	0.00742	0.00724	0.03504
<b>Exemplo 5</b>	0.0076	0.00065	0.00657	0.0092
<b>Exemplo 6</b>	0.0064	0.00084	0.00479	0.00857

Tabela 5.9. Métrica de Dispersão - Algoritmo NSGA2.

O algoritmo proposto obteve resultados satisfatórios e competitivos em todos os testes desta primeira etapa.

### 5.3 Segunda Etapa de Testes

Na segunda etapa de testes foram considerados 8 problemas, apresentados em (Deb *et. al.*, 2000) e denominados CTPs (*Constrained Test Problems*).

Para estes problemas foram definidos os parâmetros da Tabela 5.10, respectivamente: tamanho total do repertório, número total de iterações, frequência de iterações em que a supressão é aplicada, limiar de aglomeração, número mínimo de clones, número máximo de clones, limiar inferior para o tamanho da população, número de iterações sem melhora para armazenamento na memória e tamanho da memória. O mecanismo de Supressão (Seção 4.1.11) foi aplicado no espaço das funções-objetivo.

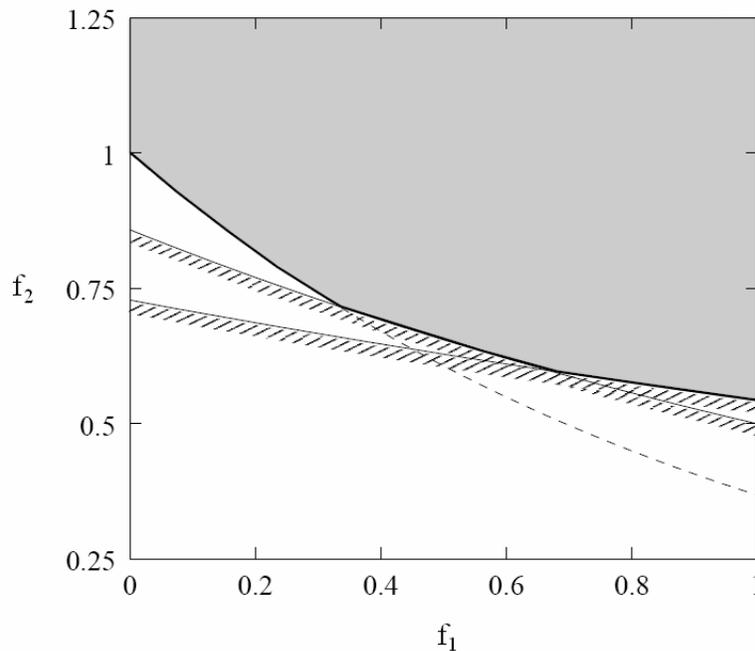
$n_T$	$It_{total}$	$t_s$	$lim_{crowd}$	$c_{min}$	$c_{max}$	$n_{t0}$	$it_m$	$n_m$
100	1000	200	0,00001	40	60	$0,7 * n_T$	50	$6 * n_T$

Tabela 5.10. Parâmetros comuns da segunda etapa de testes.

**CTP1:** Neste problema, as restrições fazem com que parte da região Pareto-ótima irrestrita se torne inactível. Desta maneira, a Fronteira de Pareto restrita será formada pela concatenação da Fronteira de Pareto irrestrita com parte dos limites das restrições. Por isso, algumas variáveis de decisão deverão satisfazer a restrição na igualdade. E isto pode se tornar uma dificuldade nos casos de restrições não-lineares.

$$\begin{array}{ll}
 \text{Minimizar} & f_1(\mathbf{x}) = x_1 \\
 & f_2(\mathbf{x}) = h(\mathbf{x}) \exp[-f_1(\mathbf{x})/h(\mathbf{x})] \\
 \text{s.a.} & g_1(\mathbf{x}) = f_2(\mathbf{x}) - 0.858 \exp[-0.541 f_1(\mathbf{x})] \geq 0 \\
 & g_2(\mathbf{x}) = f_2(\mathbf{x}) - 0.728 \exp[-0.295 f_1(\mathbf{x})] \geq 0 \\
 & h(\mathbf{x}) = 1 + x_2 \quad x_1, x_2 \in [0, 1]
 \end{array}$$

A Figura 5.17 ilustra parte da Fronteira de Pareto irrestrita (linha tracejada), as restrições do problema e a verdadeira Fronteira de Pareto restrita.



**Figura 5.17.** Verdadeira Fronteira de Pareto – CTP1.

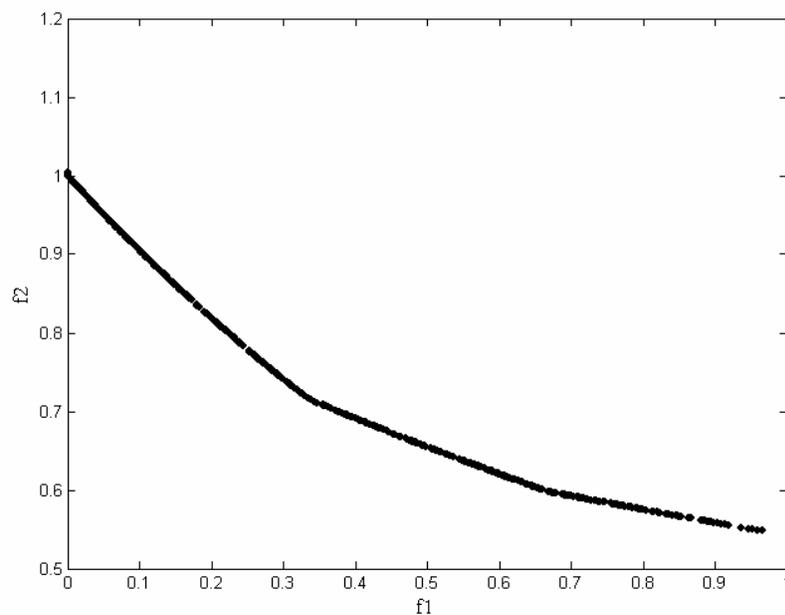
Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (inactível e factível) e limiar de inactibilidade permitido, respectivamente (Tabela 5.11):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,007	0,002	0,1

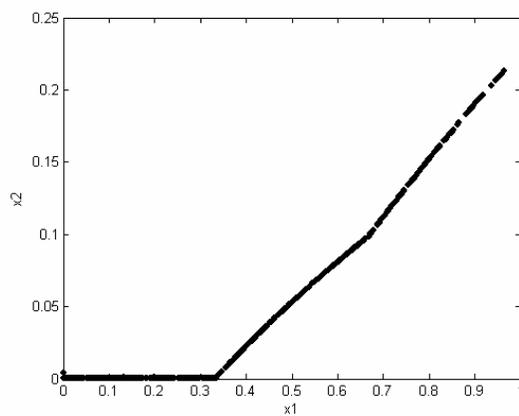
**Tabela 5.11.** Parâmetros específicos – CTP1.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontradas estão apresentados, respectivamente, na Figura 5.18 e na Figura 5.19.

O tempo de execução da melhor rodada foi de 53 segundos e 480 soluções não-dominadas foram encontradas.



**Figura 5.18.** Fronteira de Pareto obtida - CTP1.

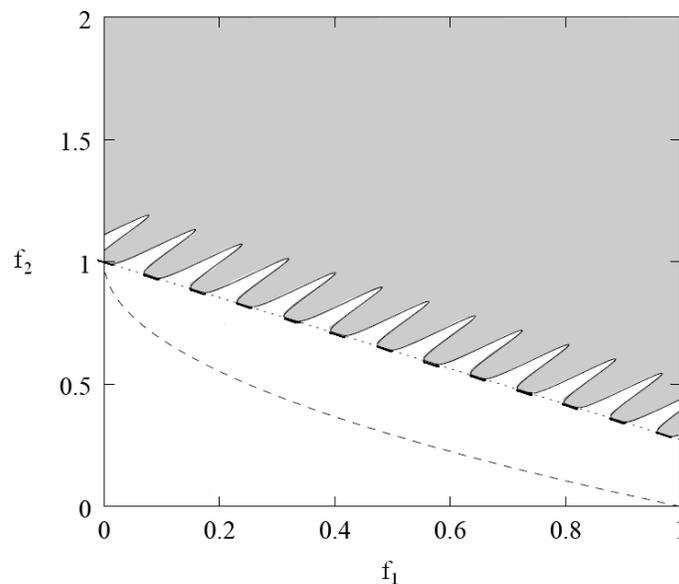


**Figura 5.19.** Conjunto de soluções não-dominadas obtido - CTP1.

Para o problema CTP1, o algoritmo obteve bons resultados, conseguindo convergir para a verdadeira Fronteira de Pareto, com soluções bem distribuídas, tanto no espaço dos objetivos, quanto no espaço das variáveis.

**CTP2:** Neste problema, a região Pareto-ótima irrestrita se torna inactível na presença da restrição (Figura 5.20). A natureza periódica dos limites da restrição forma uma Fronteira de Pareto descontínua (formada por regiões contínuas desconexas).

$\begin{aligned} \text{Minimizar} \quad & f_1(\mathbf{x}) = x_1 \\ & f_2(\mathbf{x}) = h(\mathbf{x}) \left[ 1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})} \right] \\ \text{s.a.} \quad & g_1(\mathbf{x}) = \cos(\theta) [f_2(\mathbf{x}) - 1] - \sin(\theta) f_1(\mathbf{x}) \geq 0.2 \left  \sin \left\{ 10\pi \left[ \sin(\theta) (f_2(\mathbf{x}) - 1) + \cos(\theta) f_1(\mathbf{x}) \right] \right\} \right ^6 \\ & h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.2\pi \quad x_1, x_2 \in [0, 1] \end{aligned}$
---



**Figura 5.20.** Verdadeira Fronteira de Pareto – CTP2.

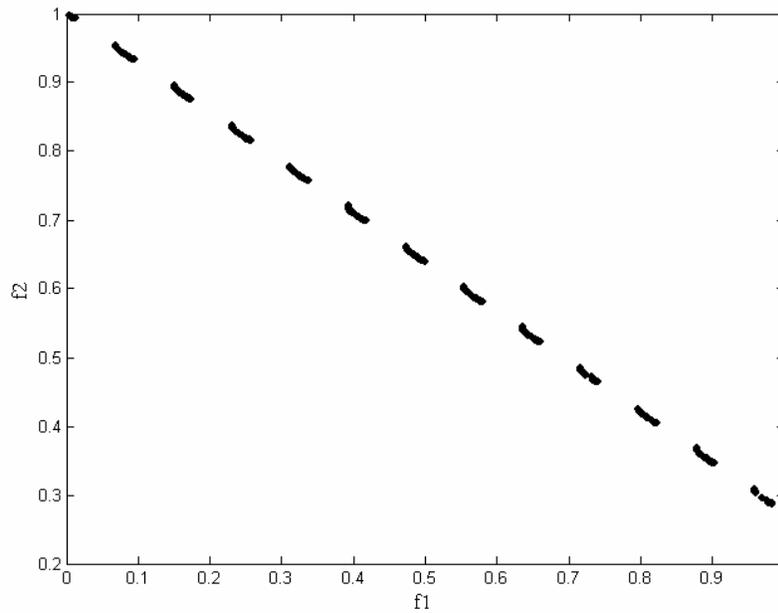
Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (inactível e factível) e limiar de inactibilidade permitido, respectivamente (Tabela 5.12):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,009	0,007	0,3

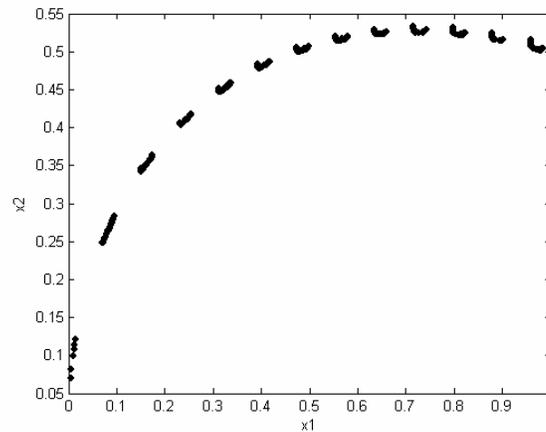
**Tabela 5.12.** Parâmetros específicos – CTP2.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontradas estão apresentados, respectivamente, na Figura 5.21 e na Figura 5.22.

O tempo de execução da melhor rodada foi de 54 segundos e 392 soluções não-dominadas foram encontradas.



**Figura 5.21.** Fronteira de Pareto obtida – CTP2.

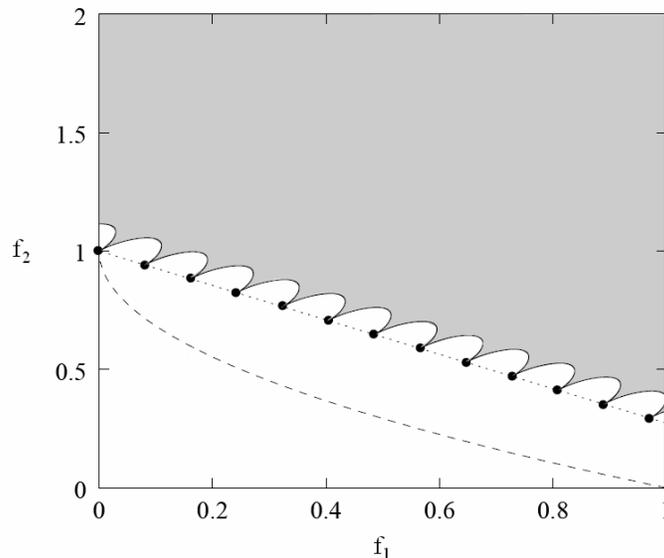


**Figura 5.22.** Conjunto de soluções não-dominadas obtido – CTP2.

Novamente, para a CTP2, o algoritmo obteve resultados satisfatórios; conseguiu encontrar a verdadeira Fronteira de Pareto em todas as regiões contínuas desconexas, com soluções bem distribuídas.

**CTP3:** Este problema possui a mesma estrutura do CTP2, porém, com maior dificuldade. Em cada região desconexa existe somente uma solução Pareto-ótima (Figura 5.23). Embora a maior parte da região factível seja contínua, próximo a Fronteira de Pareto restrita, as regiões factíveis não possuem conexão. Um algoritmo enfrentará dificuldades em encontrar todas as soluções Pareto-ótimas discretas por causa das alterações contínuas e descontínuas da região factível próxima à Fronteira de Pareto.

$$\begin{aligned}
 &\text{Minimizar} && f_1(\mathbf{x}) = x_1 \\
 &&& f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right) \\
 \text{s.a.} &&& g_1(\mathbf{x}) = \cos(\theta)[f_2(\mathbf{x}) - 1] - \sin(\theta)f_1(\mathbf{x}) \geq 0.1 \left| \sin \left\{ 10\pi [\sin(\theta)(f_2(\mathbf{x}) - 1) + \cos(\theta)f_1(\mathbf{x})]^1 \right\} \right|^{0.5} \\
 &&& h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.2\pi \quad x_1, x_2 \in [0, 1]
 \end{aligned}$$



**Figura 5.23.** Verdadeira Fronteira de Pareto – CTP3.

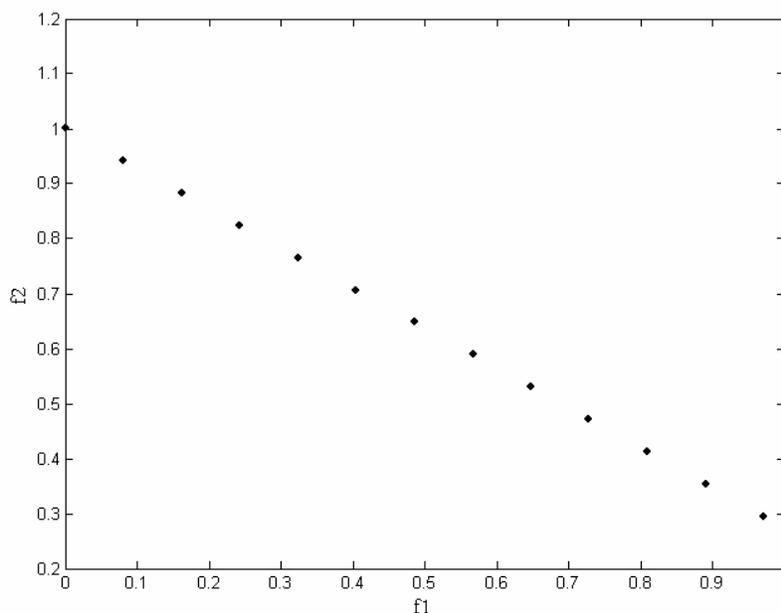
Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.13):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,7

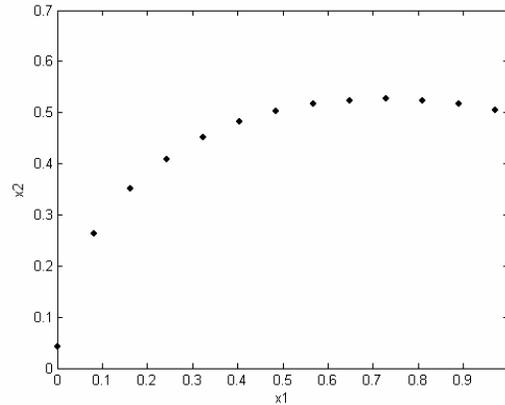
**Tabela 5.13.** Parâmetros específicos – CTP3.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontradas estão apresentados, respectivamente, na Figura 5.24 e na Figura 5.25.

O tempo de execução da melhor rodada foi de 65 segundos e 15 soluções não-dominadas foram encontradas.



**Figura 5.24.** Fronteira de Pareto obtida – CTP3.



**Figura 5.25.** Conjunto de soluções não-dominadas obtido – CTP3.

Mais uma vez, todos os pontos da Fronteira de Pareto foram encontrados, o que confirma a capacidade do algoritmo em manter a diversidade do repertório alta, durante todas as iterações.

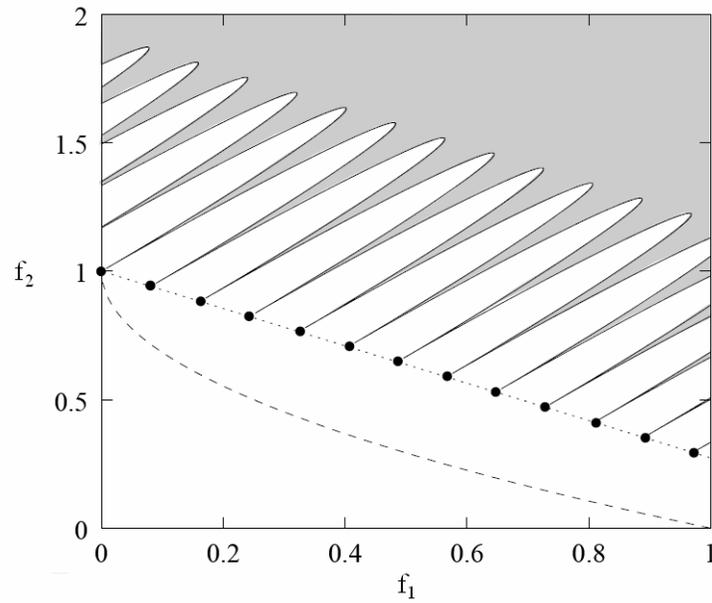
**CTP4:** Semelhante ao problema CTP3, mas com uma dificuldade maior, as restrições do problema CTP4 fazem com que a Fronteira de Pareto se distancie das regiões factíveis contínuas. Agora, um algoritmo de resolução precisa passar por um estreito caminho factível na busca por uma solução Pareto-ótima (Figura 5.26).

<p>Minimizar <math>f_1(\mathbf{x}) = x_1</math>  <math>f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right)</math></p> <p>s.a. <math>g_1(\mathbf{x}) = \cos(\theta)[f_2(\mathbf{x}) - 1] - \sin(\theta)f_1(\mathbf{x}) \geq 0.75 \left  \sin \left\{ 10\pi \left[ \sin(\theta)(f_2(\mathbf{x}) - 1) + \cos(\theta)f_1(\mathbf{x}) \right]^1 \right\} \right ^{0.5}</math></p> <p><math>h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.2\pi \quad x_1, x_2 \in [0, 1]</math></p>
---

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infactível e factível) e limiar de infactibilidade permitido, respectivamente (Tabela 5.14):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,9

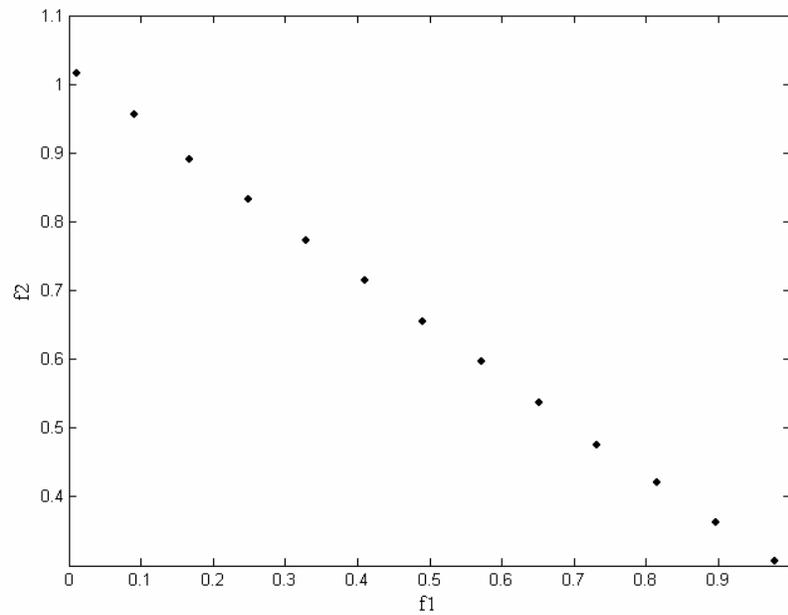
**Tabela 5.14.** Parâmetros específicos – CTP4.



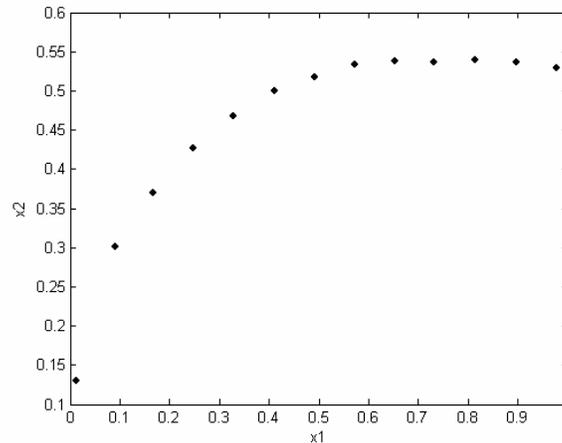
**Figura 5.26.** Verdadeira Fronteira de Pareto – CTP4.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontrado estão apresentados, respectivamente, na Figura 5.27 e na Figura 5.28.

O tempo de execução da melhor rodada foi de 67 segundos e 15 soluções não-dominadas foram encontradas.



**Figura 5.27.** Fronteira de Pareto obtida – CTP4.



**Figura 5.28.** Conjunto de soluções não-dominadas obtido – CTP4.

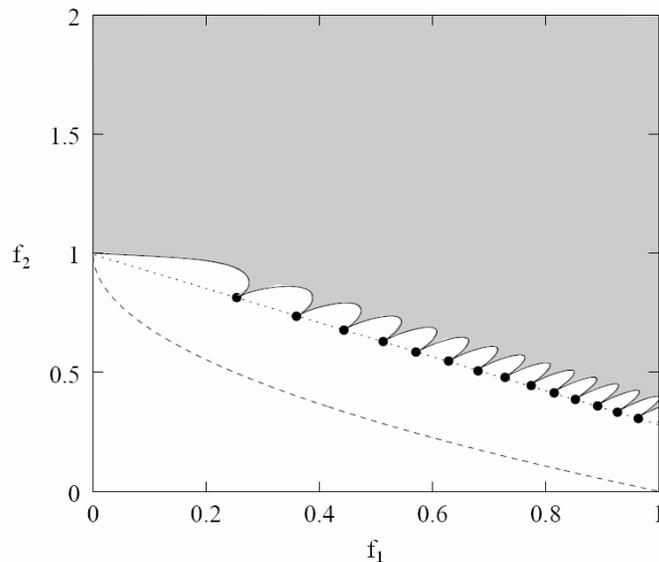
**CTP5:** Os problemas CTP2, CTP3 e CTP4 possuem regiões não conectadas igualmente distribuídas no espaço nos objetivos. No problema CTP5, existem mais soluções Pareto-ótimas à direita (para valores de  $f_1$  maiores). A dificuldade deste problema está em encontrar todas as soluções Pareto-ótimas (Figura 5.29).

$$\begin{array}{ll}
 \text{Minimizar} & f_1(\mathbf{x}) = x_1 \\
 & f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right) \\
 \text{s.a.} & g_1(\mathbf{x}) = \cos(\theta)[f_2(\mathbf{x}) - 1] - \sin(\theta)f_1(\mathbf{x}) \geq 0.1 \left| \sin\left\{10\pi[\sin(\theta)(f_2(\mathbf{x}) - 1) + \cos(\theta)f_1(\mathbf{x})]^2\right\} \right|^{0.5} \\
 & h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.2\pi \quad x_1, x_2 \in [0, 1]
 \end{array}$$

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.15):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,9

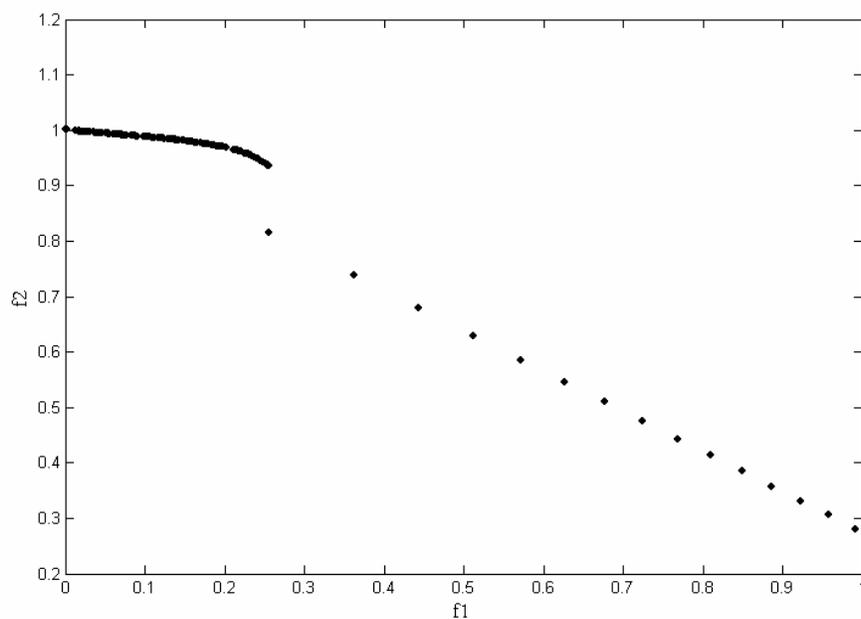
**Tabela 5.15.** Parâmetros específicos – CTP5.



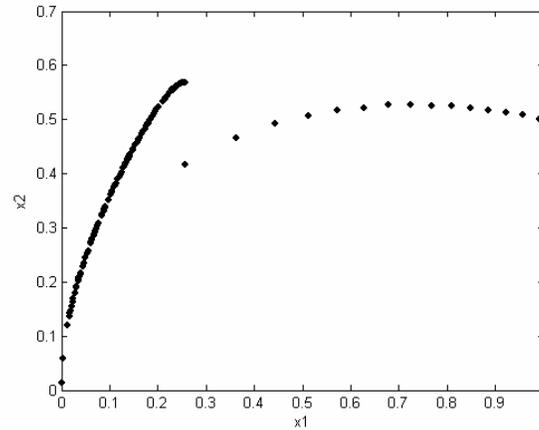
**Figura 5.29.** Verdadeira Fronteira de Pareto – CTP5.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontrado estão apresentados, respectivamente, na Figura 5.30 e na Figura 5.31.

O tempo de execução da melhor rodada foi de 61 segundos e 141 soluções não-dominadas foram encontradas.



**Figura 5.30.** Fronteira de Pareto obtida – CTP5



**Figura 5.31.** Conjunto de soluções não-dominadas obtido – CTP5.

**CTP6:** Os problemas CTP2, CTP3, CTP4 e CTP5 representaram a dificuldade na vizinhança da região Pareto-ótima. Outro tipo de dificuldade se encontra em problemas com regiões inactíveis presentes em todo o espaço de busca (Figura 5.32).

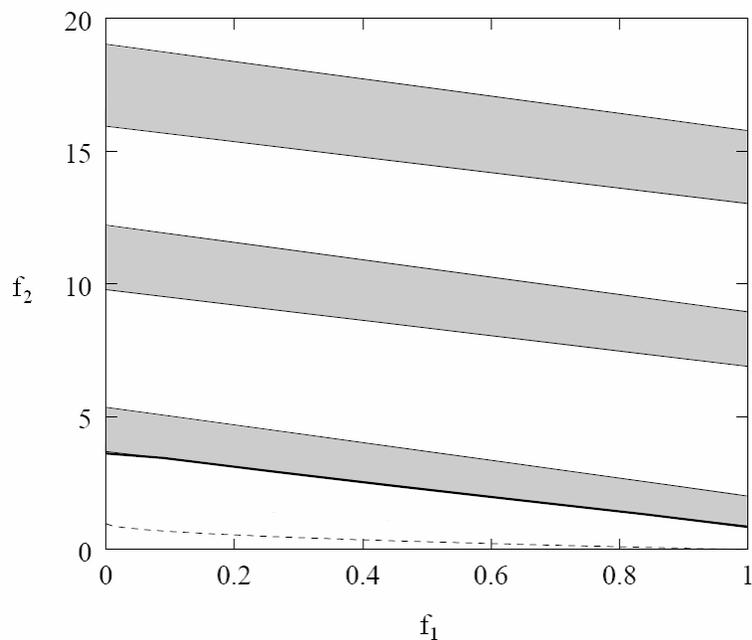
O algoritmo de resolução precisa superar as regiões inactíveis antes de convergir para a região Pareto-ótima.

Minimizar $f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right)$
s.a. $g_1(\mathbf{x}) = \cos(\theta)(f_2(\mathbf{x}) + 2) - \sin(\theta)f_1(\mathbf{x}) \geq 40 \left  \sin\left(0.5\pi(\sin(\theta)(f_2(\mathbf{x}) + 2) + \cos(\theta)f_1(\mathbf{x}))\right) \right $
$h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.1\pi \quad x_1 \in [0, 1] \quad x_2 \in [0, 5]$

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (inactível e factível) e limiar de inactibilidade permitido, respectivamente (Tabela 5.16):

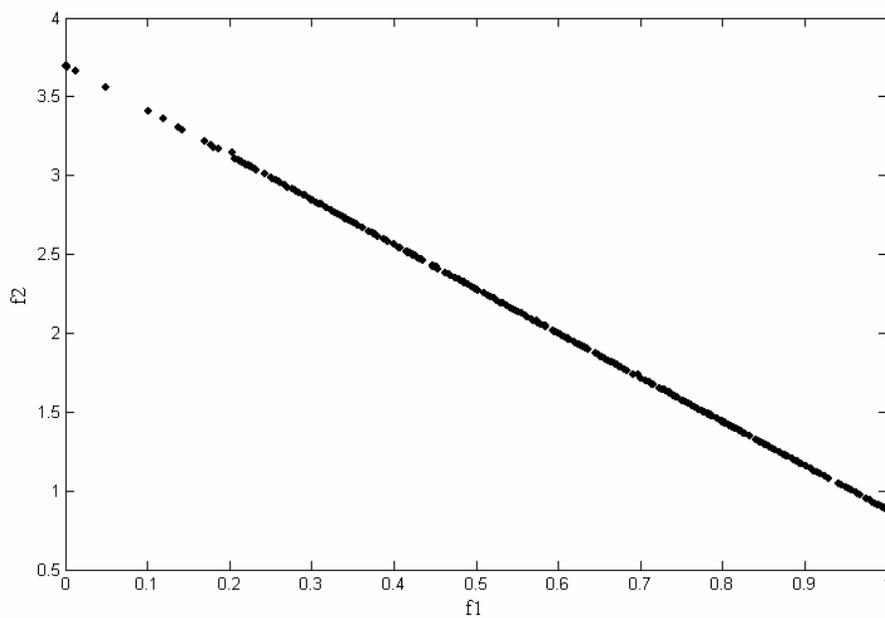
$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,9

**Tabela 5.16.** Parâmetros específicos – CTP6.

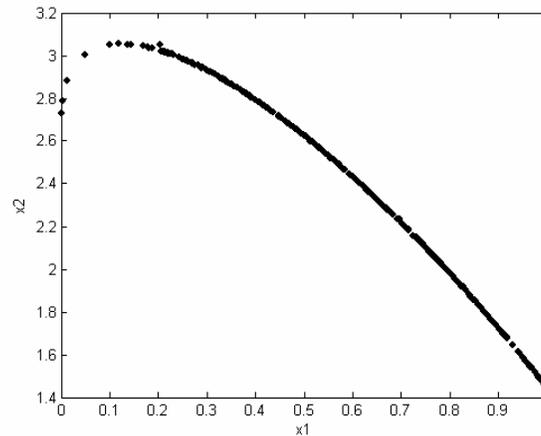


**Figura 5.32.** Verdadeira Fronteira de Pareto – CTP6.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontrado estão apresentados, respectivamente, na Figura 5.33 e na Figura 5.34.



**Figura 5.33.** Fronteira de Pareto obtida – CTP6.



**Figura 5.34.** Conjunto de soluções não-dominadas obtido – CTP6.

O tempo de execução da melhor rodada foi de 55 segundos e 343 soluções não-dominadas foram encontradas.

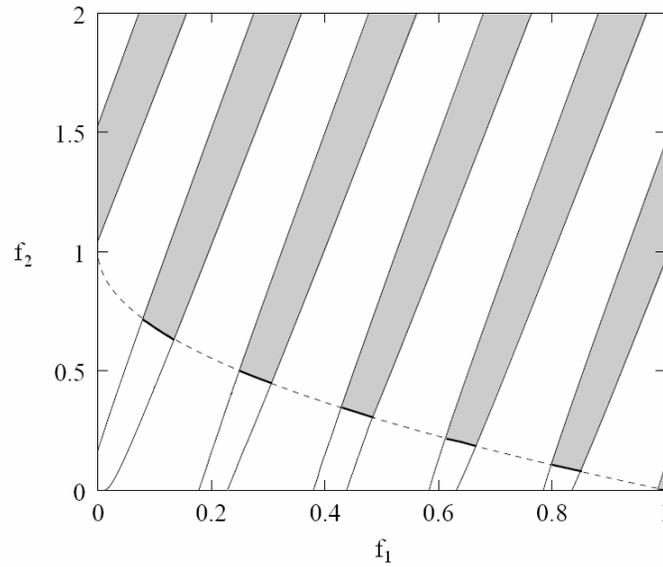
**CTP7:** Neste problema, algumas porções da região Pareto-ótima irrestrita são factíveis, formando um conjunto não conectado de regiões contínuas (Figura 5.35).

<p>Minimizar <math>f_1(\mathbf{x}) = x_1</math>  <math>f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right)</math></p> <p>s.a. <math>g_1(\mathbf{x}) = \cos(\theta) f_2(\mathbf{x}) - \sin(\theta) f_1(\mathbf{x}) \geq 40 \left  \sin\left(5\pi(\sin(\theta) f_2(\mathbf{x}) + \cos(\theta) f_1(\mathbf{x}))\right) \right ^6</math></p> <p><math>h(\mathbf{x}) = 1 + x_2 \quad \theta = -0.05\pi \quad x_1, x_2 \in [0, 1]</math></p>
--

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (infectível e factível) e limiar de infectibilidade permitido, respectivamente (Tabela 5.17):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,01	0,007	0,9

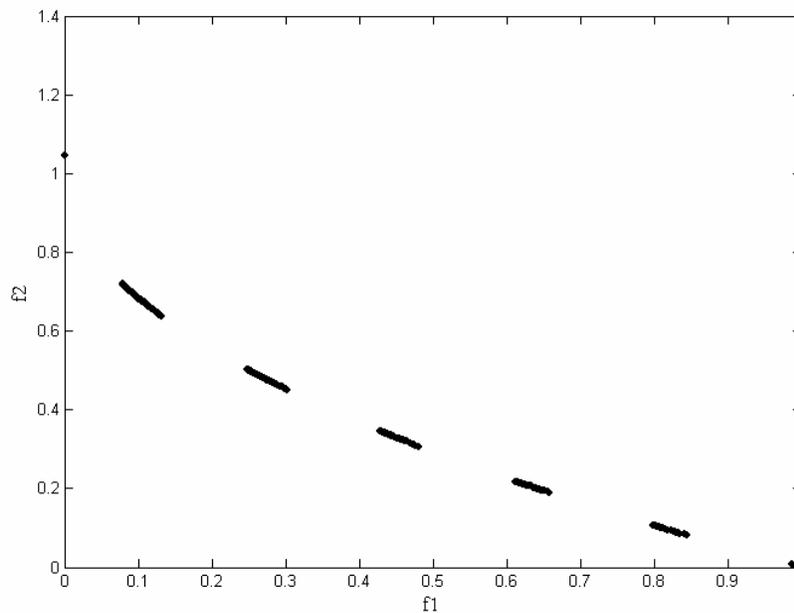
**Tabela 5.17.** Parâmetros específicos – CTP7.



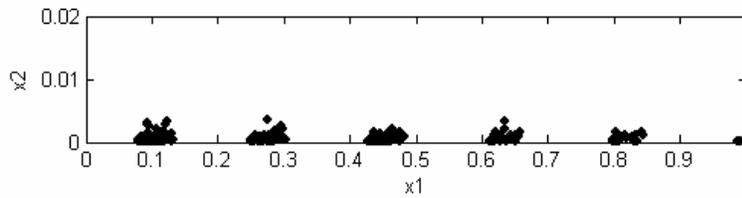
**Figura 5.35.** Verdadeira Fronteira de Pareto – CTP7.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontrado estão apresentados, respectivamente, na Figura 5.36 e na Figura 5.37.

Novamente, todas as regiões desconexas da Fronteira de Pareto foram encontradas, com soluções bem distribuídas no espaço das variáveis (também desconexo) e no espaço das funções-objetivo.



**Figura 5.36.** Fronteira de Pareto obtida – CTP7.



**Figura 5.37.** Conjunto de soluções não-dominadas obtido – CTP7.

O tempo de execução da melhor rodada foi de 62 segundos e 317 soluções não-dominadas foram encontradas.

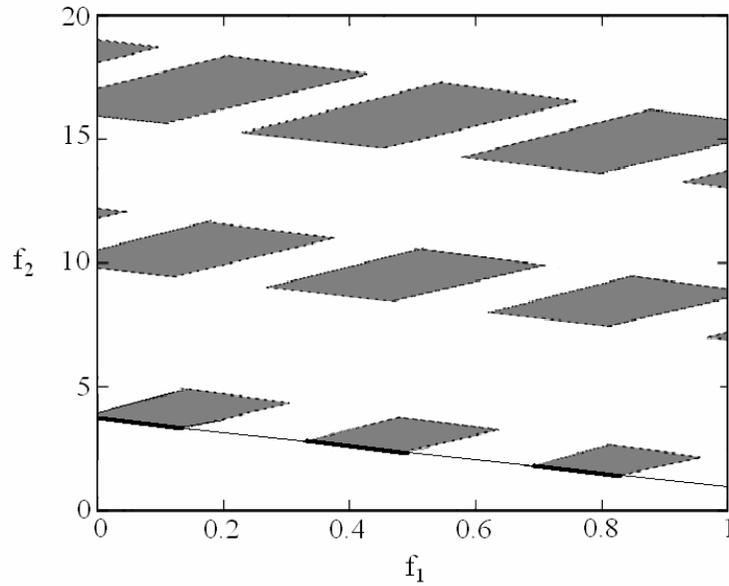
**CTP8:** Este problema também possui porções da região Pareto-ótima irrestrita inactíveis, formando um conjunto não conectado de regiões contínuas. A diferença é que não existe um caminho factível a ser percorrido, mas regiões factíveis cercadas por regiões inactíveis (Figura 5.38).

<p>Minimizar <math>f_1(\mathbf{x}) = x_1</math>  <math>f_2(\mathbf{x}) = h(\mathbf{x}) \left(1 - \sqrt{f_1(\mathbf{x})/h(\mathbf{x})}\right)</math></p> <p>s.a. <math>g_1(\mathbf{x}) = \cos(\theta_1)(f_2(\mathbf{x}) + 2) - \sin(\theta_1)f_1(\mathbf{x}) \geq 40 \left  \sin \left\{ 0.5\pi [\sin(\theta_1)(f_2(\mathbf{x}) + 2) + \cos(\theta_1)f_1(\mathbf{x})]^l \right\} \right </math>  <math>g_2(\mathbf{x}) = \cos(\theta_2)f_2(\mathbf{x}) - \sin(\theta_2)f_1(\mathbf{x}) \geq 40 \left  \sin \left\{ 2\pi [\sin(\theta_2)f_2(\mathbf{x}) + \cos(\theta_2)f_1(\mathbf{x})]^l \right\} \right ^6</math></p> <p><math>h(\mathbf{x}) = 1 + x_2 \quad \theta_1 = 0.1\pi \quad \theta_2 = -0.05\pi \quad x_1 \in [0, 1] \quad x_2 \in [0, 10]</math></p>
---

Para a resolução deste problema foram considerados os seguintes parâmetros: amplitudes mínimas de mutação (inactível e factível) e limiar de inactibilidade permitido, respectivamente (Tabela 5.18):

$\beta_I$	$\beta_F$	$\text{lim}_{\text{infac}}$
0,02	0,017	0,5

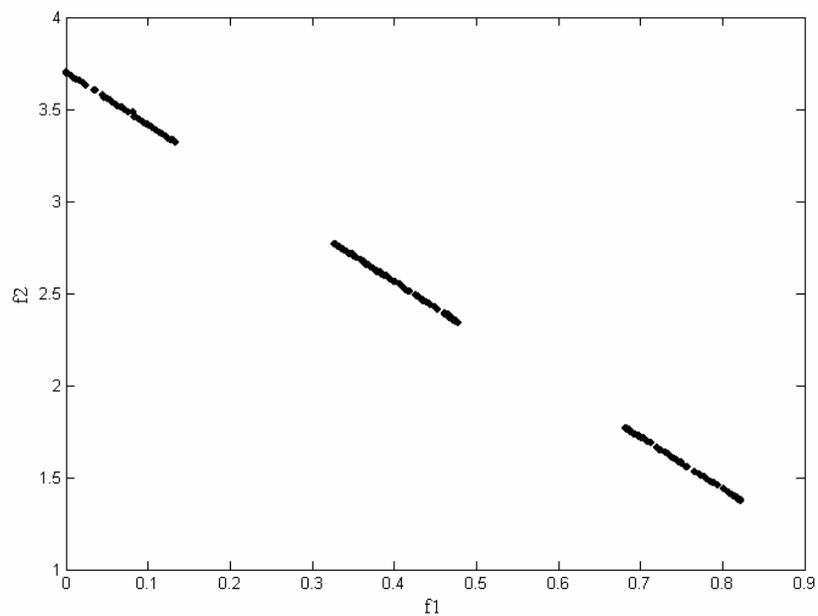
**Tabela 5.18.** Parâmetros específicos – CTP8



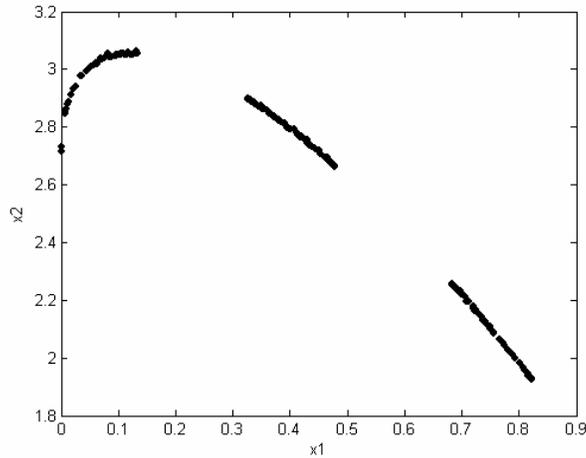
**Figura 5.38.** Verdadeira Fronteira de Pareto – CTP8.

A Fronteira de Pareto e o conjunto de soluções não-dominadas encontrado estão apresentados, respectivamente, na Figura 5.39 e na Figura 5.40.

O tempo de execução da melhor rodada foi de 61 segundos e 248 soluções não-dominadas foram encontradas.



**Figura 5.39.** Fronteira de Pareto obtida – CTP8.



**Figura 5.40.** Conjunto de soluções não-dominadas obtido – CTP8.

Mais uma vez, a verdadeira Fronteira de Pareto foi encontrada, com soluções bem distribuídas por toda a Fronteira de Pareto e no Conjunto Pareto-ótimo.

A medida de Dispersão não foi utilizada na comparação nesta segunda etapa de testes devido às características dos problemas: Fronteira de Pareto formada por várias regiões desconexas (e em alguns casos, cada região é representada por somente uma solução). Neste grupo de problemas, o interesse foi verificar o potencial do algoritmo na busca pelas regiões Pareto-ótimas.

É importante ressaltar que o algoritmo foi capaz de manter uma diversidade adequada desde as iterações iniciais até a solução final. Esta é uma característica dos Sistemas Imunológicos Artificiais e contribuiu fortemente para a obtenção dos bons resultados.

## 5.4 Terceira Etapa de Testes

O objetivo nesta terceira etapa de testes é testar o desempenho do algoritmo proposto em problemas com maior número de funções-objetivo.

**VNT:** O problema, proposto por Viennet (1995) apresenta várias características interessantes, espaço dos objetivos de maior dimensão, conjunto Pareto-ótimo descontínuo e vários mínimos locais nas funções-objetivo.

$$\begin{aligned} \text{Minimizar} \quad f_1(\mathbf{x}) &= \frac{1}{2}(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\ f_2(\mathbf{x}) &= \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\ f_3(\mathbf{x}) &= \frac{1}{(x_1^2 + x_2^2 + 1)} - 1.1 \exp(-x_1^2 - x_2^2) \\ x_1, x_2 &\in [-3, 3] \end{aligned}$$

Para o problema VNT foram considerados os parâmetros apresentados pela Tabela 5.19, respectivamente, tamanho do repertório, número total de iterações, frequência de supressão, limiar de aglomeração, número mínimo de clones, número máximo de clones, limiar inferior para o tamanho da população, número de iterações sem melhora para armazenamento na memória, tamanho da memória, amplitudes mínimas de mutação (para anticorpos inactiváveis e factíveis, respectivamente) e limiar de inactivabilidade permitido. O mecanismo de Supressão (Seção 4.1.11) foi aplicado no espaço das variáveis.

$n_T$	$It_{total}$	$t_S$	$lim_{crowd}$	$c_{min}$	$c_{max}$	$n_{t0}$	$it_m$	$n_m$	$\beta_I$	$\beta_F$	$lim_{infac}$
100	1500	200	0,0001	40	60	$0,7 * n_T$	100	$6 * n_T$	0,1	0,07	0,1

**Tabela 5.19.** Parâmetros – VNT.

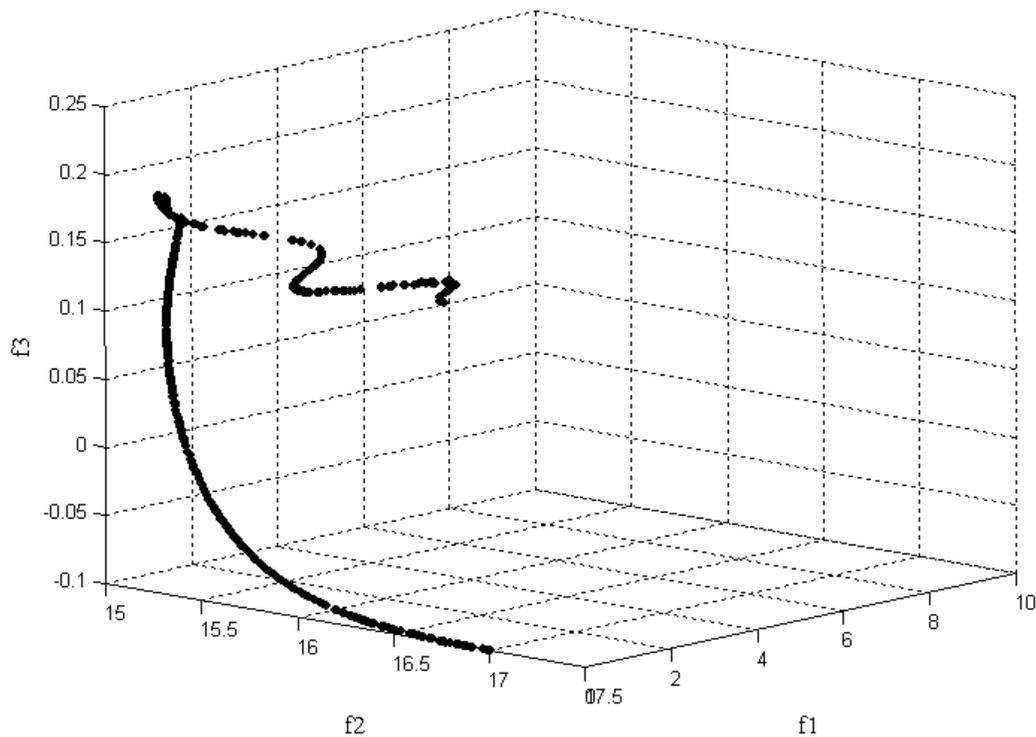
O tempo de execução da melhor rodada foi de 69 segundos e 533 soluções não-dominadas foram encontradas.

A Tabela 5.20 apresenta os resultados obtidos pelo algoritmo proposto e pelo algoritmo NSGA2, em relação à Métrica de Dispersão.

VNT	Média	Desvio	Melhor	Pior
<b>Algoritmo Proposto</b>	0.02612	0.00279	0.0214	0.0314
<b>NSGA2</b>	0.04028	0.00408	0.03009	0.0459

**Tabela 5.20.** Métrica de Dispersão – VNT.

Analisando a Figura 5.41, observa-se que o algoritmo proposto conseguiu obter uma boa representação e distribuição da Fronteira de Pareto.



**Figura 5.41.** Fronteira de Pareto obtida – VNT.

**WATER:** Com cinco objetivos e sete restrições, o problema resolvido em Deb *et. al.* (2002), dificulta a discussão do efeito das restrições na região Pareto-ótima irrestrita.

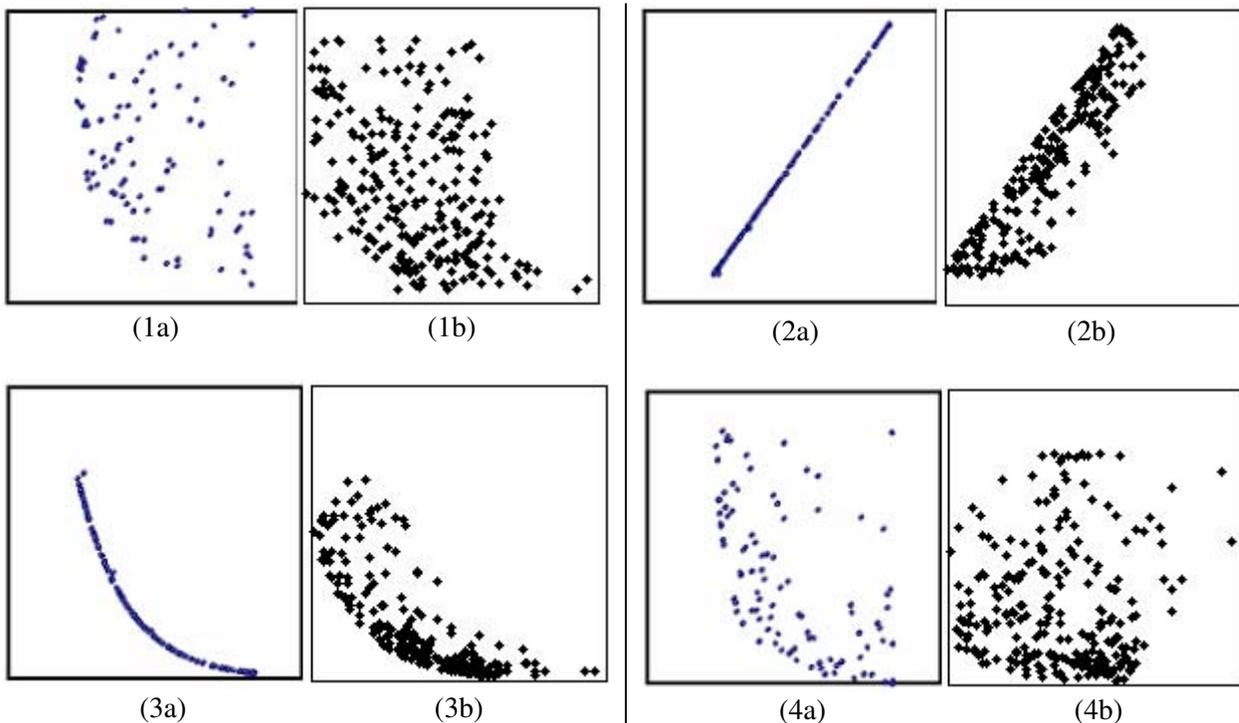
Para o problema WATER foram considerados os parâmetros apresentados pela Tabela 5.21, respectivamente, tamanho do repertório, número total de iterações, número mínimo de clones, número máximo de clones, limiar inferior para o tamanho da população, número de iterações sem melhora para armazenamento na memória, tamanho da memória, amplitudes mínimas de mutação (para anticorpos inactíveis e factíveis, respectivamente) e limiar de inactibilidade permitido. O mecanismo de Supressão (Seção 4.1.11) não foi aplicado por estar restrito a espaços bidimensionais.

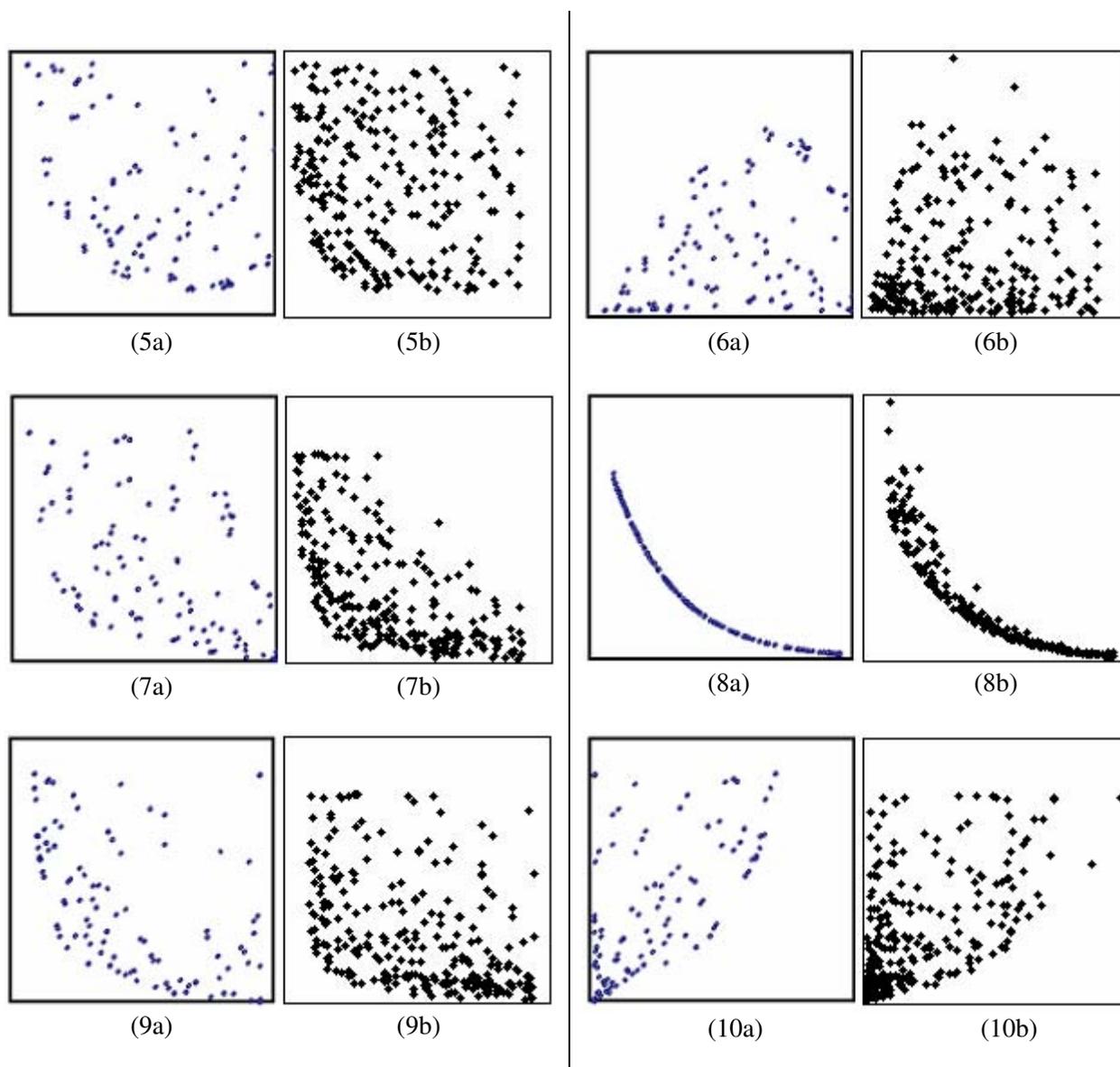
$n_T$	$It_{total}$	$c_{min}$	$c_{max}$	$n_{t0}$	$it_m$	$n_m$	$\beta_I$	$\beta_F$	$lim_{infac}$
200	1000	40	60	$0,7 * n_T$	500	$6 * n_T$	0,0005	0,0001	0,8

**Tabela 5.21.** Parâmetros – WATER.

Minimizar	$f_1(\mathbf{x}) = 106780.37(x_2 + x_3) + 61704.67$
	$f_2(\mathbf{x}) = 3000x_1$
	$f_3(\mathbf{x}) = (305700)2289x_2 / (137.34)^{0.65}$
	$f_4(\mathbf{x}) = (250)2289 \exp(-39.75x_2 + 9.9x_3 + 2.74)$
	$f_5(\mathbf{x}) = 25(1.39 / (x_1x_2) + 4940x_3 - 80)$
s.a.	$g_1(\mathbf{x}) = 0.00139 / (x_1x_2) + 4.94x_3 - 0.08 \leq 1$
	$g_2(\mathbf{x}) = 0.000306 / (x_1x_2) + 1.082x_3 - 0.0986 \leq 1$
	$g_3(\mathbf{x}) = 12.307 / (x_1x_2) + 49408.24x_3 + 4051.02 \leq 50000$
	$g_4(\mathbf{x}) = 2.098 / (x_1x_2) + 8046.33x_3 - 696.71 \leq 16000$
	$g_5(\mathbf{x}) = 2.138 / (x_1x_2) + 7883.39x_3 - 705.04 \leq 10000$
	$g_6(\mathbf{x}) = 0.417 / (x_1x_2) + 1721.26x_3 - 136.54 \leq 2000$
	$g_7(\mathbf{x}) = 0.164 / (x_1x_2) + 631.13x_3 - 54.48 \leq 550$
	$x_1 \in [0.01, 0.45] \quad x_2 \in [0.01, 0.10] \quad x_3 \in [0.01, 0.10]$

Através da análise gráfica, observa-se que o algoritmo proposto conseguiu obter bons resultados. Porém, em algumas situações (2, 3, 8) foi superado pela algoritmo NSGA2 (Figura 5.42).





**Figura 5.42.** Fronteiras de Pareto obtidas pelo algoritmo NSGA2 e pelo algoritmo proposto, a e b respectivamente – WATER. Na seguinte ordem: (1)  $[f_1, f_2]$ , (2)  $[f_1, f_3]$ , (3)  $[f_1, f_4]$ , (4)  $[f_1, f_5]$ , (5)  $[f_2, f_3]$ , (6)  $[f_2, f_4]$ , (7)  $[f_2, f_5]$ , (8)  $[f_3, f_4]$ , (9)  $[f_3, f_5]$ , (10)  $[f_4, f_5]$ .

O tempo de execução da melhor rodada foi de 168 segundos e 251 soluções não-dominadas foram encontradas.

## 5.5 Quarta Etapa de Testes: Espaços de Busca fortemente restritos

Existem poucos trabalhos na literatura que contemplam a resolução de problemas de Otimização Multiobjetivo com espaço de busca fortemente restrito. Entretanto, isto é bastante comum nas aplicações do mundo real e é necessário desenvolver novas técnicas que possam lidar eficientemente com estes tipos de problemas (Coello, 2005).

Com três funções-objeto e 8 restrições, o problema resolvido em Moura (2002)<sup>13</sup>, possui região factível não-convexa, fortemente restrita. Em problemas deste tipo pode-se encontrar grandes dificuldades na geração de soluções factíveis, comprometendo a convergência dos algoritmos de resolução.

Foi devido a estas situações, que no algoritmo proposto considerou-se a clonagem e maturação de algumas soluções infactíveis (respeitando um limiar de infactibilidade). Quanto mais restrito for o problema, mais soluções infactíveis podem ser selecionadas.

Foram considerados os parâmetros apresentados pela Tabela 5.22, respectivamente, tamanho do repertório, número total de iterações, número mínimo de clones, número máximo de clones, limiar inferior para o tamanho da população, número de iterações sem melhora para armazenamento na memória, tamanho da memória, amplitudes mínimas de mutação (para anticorpos infactíveis e factíveis, respectivamente) e limiar de infactibilidade permitido. O mecanismo de Supressão (Seção 4.1.11) não foi aplicado por estar restrito a espaços bidimensionais.

O tempo de execução da melhor rodada foi de 1290 segundos.

$n_T$	$It_{total}$	$c_{min}$	$c_{max}$	$n_{t0}$	$it_m$	$n_m$	$\beta_I$	$\beta_F$	$lim_{infac}$
500	1500	40	60	$0,7 * n_T$	30	$6 * n_T$	0,4	0,2	1

Tabela 5.22. Parâmetros – Problema fortemente restrito.

<sup>13</sup> Moura (2002) estudou problemas de Otimização Multiobjetivo *Fuzzy*. Para a resolução nesta dissertação, foram considerados apenas os valores modais dos coeficientes *Fuzzy*.

Minimizar	$f_1(\mathbf{x}) = 7x_1^2 - x_2^2 + x_1x_2 + 8(x_3 - 10)^2 - 14x_1 - 16x_2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2$ $+ 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + x_{10}^2 + 45$
	$f_2(\mathbf{x}) = (x_1 - 5)^2 + 5(x_2 - 12)^2 + 0.5x_3^4 + 3(x_4 - 11)^2 + 0.2x_5^5 + 7x_6^2 + 0.1x_7^4 - 4x_6x_7$ $- 10x_6 - 8x_7 + x_8^2 + 3(x_9 - 5)^2 + (x_{10} - 5)^2$
	$f_3(\mathbf{x}) = x_1^3 + (x_2 - 5)^2 + 3(x_3 - 9)^2 - 12x_3 + 2x_4^3 + 4x_5^2 + (x_6 - 5)^2 + 6x_7^2 + 3(x_7 - 2)x_8^2$ $- x_9x_{10} + 4x_9^3 + 5x_1 - 8x_1x_7$
s.a.	$g_1(\mathbf{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 + 2x_5x_5x_8 - 120 \leq 0$ $g_2(\mathbf{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$ $g_3(\mathbf{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 + 6x_5x_6 \leq 0$ $g_4(\mathbf{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_5x_8 - 30 \leq 0$ $g_5(\mathbf{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$ $g_6(\mathbf{x}) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leq 0$ $g_7(\mathbf{x}) = 10x_1 - 8x_2 - 18x_7 + 2x_8 \leq 0$ $g_8(\mathbf{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$
	$x_i \in [-5, 10] \quad i = 1, \dots, 10$

Empiricamente, foi constatado que em problemas fortemente restritos, como este, não é bom que as soluções infactíveis sejam descartadas durante as iterações do algoritmo (como fazem alguns algoritmos da literatura).

Considerando os mesmos dados da Tabela 5.22, porém com limiar de infactibilidade igual a 0 (ou seja, descartando todas as soluções infactíveis), no final das iterações, o número de soluções não-dominadas e factíveis encontradas, em cada uma das 10 rodadas de testes, foi: [5, 1, 1, 2, 2, 6, 3, 2, 1, 4].

Por outro lado, considerando um limiar de infactibilidade igual a um (todas as soluções infactíveis são seleccionadas para clonagem e maturação) obteve-se, nas 10 rodadas de testes, o seguinte número de soluções não-dominadas e factíveis: [52, 64, 59, 56, 72, 60, 68, 59, 56, 58].

Isto justifica o maior gasto computacional na tentativa de melhorar (ou tornar factível) algumas soluções infactíveis, em determinados problemas.

## Capítulo 6

### Conclusões e Perspectivas Futuras

Certas características dos problemas de Otimização Multiobjetivo podem resultar em maior (ou menor) grau de dificuldade de resolução. O algoritmo proposto se mostrou competitivo quando comparado com um algoritmo representativo do estado da arte (NSGA2), conseguindo obter bons resultados na maioria dos casos estudados.

O algoritmo também conseguiu manter alta a diversidade do repertório (com destaque para os resultados obtidos nos testes com os CTPs), o que contribuiu na obtenção dos resultados. Além disso, mostrou-se capaz de tratar problemas de diferentes características de forma satisfatória.

Houve interesse em testar o desempenho do algoritmo frente a problemas fortemente restritos (e pouco estudados na literatura) e concluiu-se que, nestes casos, é importante não desconsiderar as soluções ineficazes durante a resolução do problema, mesmo que isso signifique maior custo computacional.

Outra vantagem interessante do algoritmo é que não houve a necessidade de incorporação das restrições às funções-objetivo através de técnicas de penalização.

Perspectivas futuras e de continuação dos estudos na área devem levar em consideração algumas observações:

- É intuitivo que, com o aumento do número de funções-objetivo, a região Pareto-ótima passa a ser representada por superfícies multi-dimensionais. Por isso, o algoritmo deve ser capaz de manter maior diversidade de soluções não-dominadas em cada iteração. Trabalhos futuros poderão incluir novos mecanismos para controle de diversidade em problemas com mais de dois objetivos.
- Mais comparações com diferentes implementações podem ser realizadas. Mas de que maneira um conjunto de solução pode ser comparado em relação a outro com

menor número de soluções Pareto-ótimas, porém com grande diversidade? E que tipo de comparação deve ser realizada entre os diferentes algoritmos Bio-inspirados?

- O desempenho do algoritmo proposto em aplicações de problemas de Otimização Multiobjetivo do mundo real deve ser estudado.
- Outras hibridizações, como no caso do cálculo da direção de maturação (Seção 4.1.5.2), podem ser interessantes na resolução dos problemas.

# Bibliografia

**Abido, M. A. (2003).** Environmental/economic power dispatch using multiobjective evolutionary algorithms, *IEEE Transactions on Power Systems*, 18(4), pp. 1529-1537.

**Amorim, E. A., Romero, R. A. e Mantovani, J. R. S. (2006).** Algoritmo Evolutivo Multiobjetivo para a solução do Modelo de Controle de Potências Ativa e Reativa, XXXVIII Simpósio Brasileiro de Pesquisa Operacional, Pesquisa Operacional na Sociedade, Educação, Meio ambiente e Desenvolvimento, pp. 478-489, Goiás, Brasil.

**Bäck, T., Fogel, D. B. e Michalewicz, Z. (2000a).** *Evolutionary Computation 1: Basic Algorithms and Operators*, 1 edn, Institute of Physics Publishing, Bristol, UK.

**Bäck, T., Fogel, D. B. e Michalewicz, Z. (2000b).** *Evolutionary Computation 2: Advanced Algorithms and Operators*, 1 edn, Institute of Physics Publishing, Bristol, UK.

**Barán, B., Vallejos, J., Ramos, R., Fernández, U. (2001).** Reactive Power Compensation using a Multi-objective Evolutionary Algorithm, Proceedings of the 2001 IEEE Porto Power Tech Conference, Porto, Portugal.

**Berbert, P. C., Freitas Filho, L. J. R., Almeida, T. A., Carvalho, M. B. e Yamakami, A. (2007).** Artificial Immune System to Find a Set of k-Spanning Trees with Low Costs and Distinct Topologies. In: *Lecture Notes in Computer Science*. de Castro, L.N; Von Zuben, F.J.; Knidel, H. (Eds.). Proceedings of the 6th International Conference on Artificial Immune Systems, ICARIS 2007, Springer, 4628, pp. 395-406, Santos, Brasil.

**Coelho, G. P. e Von Zuben, F. J. (2006).** Omni-aiNet: An Immune Inspired Approach for Omni Optimization. In *Lecture Notes in Computer Science*. Bersini, H.; Carneiro, J. (Eds.). Proceedings of the 5th International Conference on Artificial Immune Systems, ICARIS 2006, Springer, 4163, pp. 294-308, Oeiras, Portugal.

**Coello, C. A. C. (2001).** A Short Tutorial on Evolutionary Multiobjective Optimization, *Lecture Notes In Computer Science*, Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, 1993, pp. 21-40.

**Coello, C. A. C. (2002a).** Evolutionary Multiobjective Optimization: Past, Present and Future, CINVESTAV-IPN, Depto. de Ingeniería Eléctrica, Sección de Computación, México. <http://www.cs.cinvestav.mx/~EVOCINV/download/tutorial-moea.pdf>

**Coello, C. A. C. (2002b).** Aplicaciones de los Algoritmos Evolutivos Multiobjetivo, CINVESTAV-IPN, Depto. de Ingeniería Eléctrica, Sección de Computación, México. <http://neo.lcc.uma.es/pdf-charlas/apli-MOEA.pdf>

**Coello, C. A. C. (2005).** Recent Trends in Evolutionary Multiobjective Optimization, in Ajith Abraham, Lakhmi Jain and Robert Goldberg (editors), *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*, pp. 7-32, Springer-Verlag, London.

**Coello, C. A. C. e Lechuga, M. S. (2002).** MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization, in Congress on Evolutionary Computation, 2, pp. 1051-1056, IEEE Service Center, Piscataway, New Jersey.

**Coello, C. A. C. e Cortés, N. C. (2005).** Solving Multiobjective Optimization Problems Using an Artificial Immune System, Genetic Programming and Evolvable Machines, 6(2), pp.163-190.

**Cortés, N. C. (2004).** Sistema Imune Artificial para solucionar problemas de optimización, Tese de Doutorado, Centro de Investigación y de Estudios Avanzados Del Instituto Politécnico Nacional, México.

**Dasgupta, D. (1998).** *Artificial Immune Systems and Their Applications*, 1 edn, Springer-Verlag, Berlin, Germany.

**Dasgupta, D. (2006).** Advances in Artificial Immune Systems, IEEE Computational Intelligence Magazine, november edn, pp. 40-49.

**de Castro, L. N. (2001).** Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais, Tese de Doutorado, Universidade Estadual de Campinas, Campinas, SP, Brasil.

[www.dca.fee.unicamp.br/~vonzuben/research/lnunes\\_dout/index.html](http://www.dca.fee.unicamp.br/~vonzuben/research/lnunes_dout/index.html).

**de Castro, L. N. e Timmis, J. (2002).** An Artificial Immune Network for Multimodal Function Optimization, In Proceedings of the IEEE Congress on Evolutionary Computation, 1, pp. 699-674, May, Hawaii.

**de Castro, L. N. e Von Zuben, F. J. (1999).** Artificial Immune Systems: Part I – Basic Theory and Applications, Technical Report – RT DCA 01/99, p. 95.

**de Castro, L. N. e Von Zuben, F. J. (2000).** Artificial Immune Systems: Part II – A Survey of Applications, Technical Report – RT DCA 02/00, p. 65.

**de Castro, L. N. e Von Zuben, F. J. (2002).** Learning and Optimization using the Clonal Selection Principle, Special Issue on Artificial Immune Systems, IEEE Transactions on Evolutionary Computation, 6(3), pp. 239-252.

**de França, F. O., Zuben, F. J. V., de Castro, L. N. (2005).** An artificial immune network for multimodal function optimization on dynamic environments, In Proceedings of Genetic and Evolutionary Computation Conference, pp. 289-296, ACM Press, New York.

**Deb, K. (1999).** Multi-objective genetic algorithms: Problems difficulties and construction of test problems, Evolutionary Computation, 7(3), pp. 205-230.

**Deb, K., Mathur, A. P. e Meyarivan, T. (2000).** Constrained test problems for multiobjective evolutionary optimization, Technical Report No. 200002, Kanpur: Kanpur Genetic Algorithms Laboratory, IIT Kanpur, Índia.

<http://citeseer.ist.psu.edu/deb00constrained.html>

**Deb, K., Pratap, A., Agarwal, S. e Meyarivan, T. (2002).** A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6(2), pp. 182-197.

**Deb, K., Jain, P., Gupta, N.K. e Maji, H.K. (2004).** Multiobjective placement of electronic components using evolutionary algorithms, IEEE Transactions on Components and Packaging Technologies, 27(3), pp. 480-492.

**Deb, K. e Tiwari, S. (2005).** Omni-optimizer: A Procedure for single and Multi-objective Optimization. In Coello, C. A. C.; Aguirre, A.; Zitzler, E. (Eds.). Proceedings of the 3rd EMO, Mexico, LNCS, 3410, pp. 47-61.

**Dimopoulos, C. (2004).** A Review of Evolutionary Multiobjective Optimization: Applications in the Area of Production Reserch, Proceedings of the Congress on Evolutionary Computation (CEC 04), 2, pp.1487-1494, Oregon, USA.

**Erickson, M., Mayer, A. e Horn, J. (2001).** The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems, In Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag, Lecture Notes in Computer Science, 1993, pp. 681-695.

**Ferreira, P. A. V. (1999a).** IA-527 Programação Multiobjetivo, Notas de aula, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

**Ferreira, P. A. V. (1999b).** Otimização Multiobjetivo: Teoria e Aplicações, Tese de Livre Docência, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

**Fonseca, C. M. e Fleming, P. J. (1993).** Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Stephanie Forrest (ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423, San Mateo, Califórnia.

**Freschi, F., Repetto, M. (2005).** Multiobjective optimization by a modified artificial immune system algorithm, 4th International Conference on Artificial Immune Systems, Banff, Alberta, Canadá, Lecture Notes In Computer Science, 3627, pp. 248-261.

**Hongyun, M. e Sanyang, L. (2003).** ISPEA: Improvement for the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization with Immunity, In Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications, pp. 368-372, IEEE Computer Society.

**Horn, J., Nafpliotis, N. e Goldberg, D. E. (1994).** A Niche Pareto Genetic Algorithm for multiobjective optimization, In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 1, pp. 82-87, IEEE Press, Piscataway, New Jersey.

**Kepler, T. B. e Perelson, A. S. (1993).** Somatic Hypermutation in B Cells: Na Optimal Control Treatment, Theoretical biology, 164, pp. 37-64.

**Kita, H., Yabumoto, Y., Mori, N. e Nishikawa, Y. (1996).** Multi-objective Optimization by means of the thermodynamical genetic algorithm, in Parallel Problem Solving from Nature, Lecture Notes in Computer Science, pp. 504-512, Springer-Verlag, Berlin, Germany.

**Knowles, J. D. e Corne, D. W. (2000).** Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy, Evolutionary Computation, 8(2), pp. 149-172, Cambridge, MA, USA.

**Kukkonen, S. e Deb, K. (2006).** Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems, IEEE Congress on Evolutionary Computation, pp. 1179-1186, Vancouver, BC, Canada.

**Lederberg, J. (1988).** Ontogeny of the Clonal Selection Theory of Antibody Formation, Annals of the New York Ac. of Sc., 546, pp. 175-182.

**Michalewicz, Z. (1996).** Genetic Algorithms + Data Structures = Evolution Programs, 3 edn, Springer-Verlag and Heidelberg GmbH & Co. K, Berlin, Germany.

**Moura, L. (2002).** Um Algoritmo Genético para Otimização Multiobjetivo Fuzzy, Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, SP, Brasil.

**Osyczka, A. (1985).** Multicriteria optimization for engineering desing, Gero Ed. Design Optimization, Academic Press, pp. 193-227.

**Parsopoulos, K. E. e Vrahatis, M. N. (2002).** Particle swarm optimization method in multiobjective problems, Proceedings of the ACM Symposium on Applied Computing, pp. 11-14, Madrid, Spain.

**Reddy, M. J. e Kumar, D. N. (2007).** Multiobjective Differential Evolution with Application to Reservoir System Optimization, Journal of Computing in civil Engineering, 21(2), pp. 136-146.

**Schaffer, J. D. (1984).** Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Ph. D. thesis, Vanderbilt University, Nashville, Tennessee.

**Schott, J. R. (1995).** Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, Dissertação de Mestrado, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

**Srinivas, N. e Deb, K. (1994).** Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), pp. 221-248.

**Tanaka, M. (1995).** GA-based decision support system for multicriteria optimization, in *Proceedings IEEE International Conference Systems, Man and Cybernetics-2*, pp. 1556-1561.

**Toro, F., Ros, E., Mota, S. e Ortega, J. (2006).** Evolutionary Algorithms for Multiobjective and Multimodal Optimization of Diagnostic Schemes. *IEEE Transactions on Biomedical Engineering*, 53(2), 178-189.

**Toscano-Pulido, G., Santana-Quintero, L. V. e Coello, C. A. C. (2007).** EMOPSO: A Multi-Objective Particle Swarm Optimizer with Emphasis on Efficiency, *Evolutionary Multi-Criterion Optimization*, Springer-Verlag, Lecture Notes in Computer Science, 4403, pp. 272-285.

**Viennet, R., Fontiex, C. e Marc, I. (1995).** New Multicriteria Optimization Method Based on the Use of a Diploid Genetic Algorithm: Example of an Industrial Problem. In *Proceedings of Artificial Evolution (European Conference, selected papers)*, Springer-Verlag, pp. 120-127, Brest, France.

**Zitzler, E. e Thiele, L. (1999).** Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, 3(4), pp. 257-271.

**Zitzler, E., Laumanns, M. e Thiele, L. (2001).** SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.