

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação  
Departamento de Engenharia de Computação e Automação Industrial

**Redes Imunológicas Artificiais para Otimização em  
Espaços Contínuos: Uma Proposta Baseada em  
Concentração de Anticorpos**

Guilherme Palermo Coelho

**Orientador:** Prof. Dr. Fernando José Von Zuben

**Tese de Doutorado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Campinas – São Paulo – Brasil  
Abril, 2011

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

C65r Coelho, Guilherme Palermo  
Redes imunológicas artificiais para otimização em espaços contínuos: uma proposta baseada em concentração de anticorpos / Guilherme Palermo Coelho. --Campinas, SP: [s.n.], 2011.

Orientador: Fernando José Von Zuben.  
Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Metaheurística. 2. Otimização. 3. Diversidade. 4. Algoritmos evolutivos. 5. Imunologia - Simulação por computador. I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Artificial immune networks for real-parameter optimization: a concentration-based approach

Palavras-chave em Inglês: Metaheuristic, Optimization, Diversity, Evolutionary algorithms, Immunology - Computer simulation

Área de concentração: Engenharia de Computação

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Alexandre Cláudio Botazzo Delbem, Myriam Regattieri De Biase da Silva Delgado, Christiano Lyra Filho, Romis Ribeiro de Faissol Attux

Data da defesa: 01/04/2011

Programa de Pós Graduação: Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Guilherme Palermo Coelho

**Data da Defesa:** 1 de abril de 2011

**Título da Tese:** "Redes Imunológicas Artificiais para Otimização em Espaços Contínuos: Uma Proposta Baseada em Concentração de Anticorpos"

Prof. Dr. Fernando José Von Zuben (Presidente): Fernando José Von Zuben

Prof. Dr. Alexandre Cláudio Botazzo Delbem: Alexandre Cláudio Botazzo Delbem

Profa. Dra. Myriam Regattieri de Biase da Silva Delgado: Myriam Regattieri de Biase da Silva Delgado

Prof. Dr. Christiano Lyra Filho: \_\_\_\_\_

Prof. Dr. Romis Ribeiro de Faissol Attux: Romis



# Resumo

Meta-heurísticas baseadas no paradigma de Sistemas Imunológicos Artificiais (SIAs), especialmente aquelas inspiradas na teoria da Rede Imunológica, são reconhecidamente capazes de estimular a geração de conjuntos diversos de soluções para um dado problema, mesmo utilizando-se de mecanismos muito simples de controle da dinâmica da rede. Por outro lado, na literatura de SIAs há uma série de estudos que propõem modelos computacionais mais elaborados, geralmente baseados no conceito de concentração de anticorpos, que conseguem explicar melhor o comportamento dessas redes. Diante disso, neste trabalho é proposto um novo algoritmo imunoinspirado para otimização em espaços contínuos, denominado cob-aiNet (*Concentration-based Artificial Immune Network*), que emprega o conceito de concentração de anticorpos para promover um melhor controle da dinâmica da rede, permitindo assim obter uma melhor cobertura das regiões promissoras do espaço de busca. Esta propriedade da cob-aiNet foi verificada em uma série de análises experimentais, nas quais o algoritmo foi comparado a outras técnicas baseadas em paradigmas distintos, além de dois outros SIAs já propostos na literatura. Os experimentos mostraram que o algoritmo cob-aiNet, além de sua capacidade de manutenção de diversidade ao longo de toda a execução, é competitivo na aproximação do ótimo global dos problemas. Diante disso, também foi proposta neste trabalho uma extensão da cob-aiNet para tratar problemas de otimização multiobjetivo, denominada cob-aiNet[MO] (*Concentration-based Artificial Immune Network for Multiobjective Optimization*). Assim como um conjunto bem reduzido de propostas da literatura, a cob-aiNet[MO] é capaz de tratar problemas de otimização multiobjetivo que requerem uma manutenção adequada de diversidade também no espaço das variáveis de decisão, não apenas para superar as dificuldades introduzidas pela multimodalidade mas também para facilitar o processo de escolha *a posteriori* da solução que será efetivamente adotada na prática. Uma série de análises experimentais foram feitas com o algoritmo cob-aiNet[MO], sendo observado que esta ferramenta apresentou resultados superiores na maioria dos problemas, tanto em aproximação da fronteira de Pareto quanto em manutenção de diversidade.

**Palavras-chave:** Sistemas imunológicos artificiais, manutenção de diversidade no espaço de busca, otimização em espaços contínuos, otimização multiobjetivo, redes imunológicas.



# Abstract

Metaheuristics based on the Artificial Immune System (AIS) framework, especially those inspired by the Immune Network theory, are known to be capable of stimulating the generation of diverse sets of solutions for a given problem, even though they generally implement very simple mechanisms to control the dynamics of the network. However, there are several studies in the AIS literature that propose more elaborate computational models, generally based on the concept of concentration of antibodies, which better explain the behavior of such networks. Therefore, in this work we propose a novel immune-inspired algorithm for real-parameter optimization, named cob-aiNet (*Concentration-based Artificial Immune Network*), that adopts the concept of concentration of antibodies to better control the dynamics of the network, so that a broader coverage of promising regions of the search space can be achieved. This property of cob-aiNet was verified in a series of experimental analyses, in which the algorithm was compared to several techniques based on distinct paradigms, including two popular AISs from the literature. The experiments have shown that cob-aiNet, besides being able to maintain diversity during all the iterations, is also competitive with respect to the approximation of the global optima of the problems. Therefore, it was also proposed in this work an extension of cob-aiNet to deal with multiobjective optimization problems, which was named cob-aiNet[MO] (*Concentration-based Artificial Immune Network for Multiobjective Optimization*). Like a small set of techniques from the literature, cob-aiNet[MO] is capable of dealing with multiobjective optimization problems that also require a proper maintenance of diversity in the decision space, not only to overcome difficulties introduced by multimodality but also to facilitate the post-optimization decision making process. A series of experimental analyses were also made with cob-aiNet[MO], and it was observed that this algorithm presented better results in most of the considered problems, with respect to both the approximation of the Pareto front and diversity maintenance.

**Keywords:** Artificial immune systems, diversity in decision space, real-parameter optimization, multiobjective optimization, immune networks.



*Aos meus pais Joacir e Stella e à minha irmã Luciana*



# Agradecimentos

Este espaço é dedicado àqueles que deram a sua contribuição para que esta tese fosse realizada. A todos eles deixo aqui o meu sincero agradecimento.

Ao Prof. Fernando J. Von Zuben, pela orientação neste trabalho, amizade e confiança em mim depositada. O constante estímulo e os valiosos ensinamentos passados em nossas reuniões foram essenciais para o meu crescimento como pesquisador e para a consequente elaboração deste trabalho.

Aos meus pais Joacir e Stella e à minha irmã Luciana, pelo constante incentivo e apoio incondicional ao longo de toda a minha vida.

A todos os amigos do Laboratório de Bioinformática e Computação Bio-Inspirada (LBiC), pelo companheirismo e pelas discussões que contribuíram enormemente para o desenvolvimento deste trabalho. Cabe um reconhecimento especial ao colega de LBiC Fabrício Olivetti de França, com quem mais dividi a autoria de trabalhos técnico-científicos ao longo desses últimos anos. Fabrício sempre soube encurtar a distância entre uma grande ideia e sua implementação prática.

À Faculdade de Engenharia Elétrica e de Computação (FEEC) e à Universidade Estadual de Campinas (Unicamp), pela oportunidade de realização deste trabalho e por fornecerem toda a infra-estrutura necessária para o seu desenvolvimento.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte financeiro.



# Sumário

<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xxi</b>
<b>Lista de Pseudo-códigos</b>	<b>xxiii</b>
<b>Lista de Acrônimos</b>	<b>xxv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Fundamentação e Motivação . . . . .	3
1.2 Objetivos do Trabalho . . . . .	7
1.3 Contribuições Alcançadas . . . . .	8
1.4 Organização do Documento . . . . .	8
<b>2 Sistemas Imunológicos Artificiais</b>	<b>11</b>
2.1 O Sistema Imunológico Natural . . . . .	13
2.1.1 A Resposta Imunológica Adaptativa e o Princípio da Seleção Clonal . . . . .	14
2.1.2 Teoria da Rede Imunológica . . . . .	16
2.2 Os Sistemas Imunológicos Artificiais (SIAs) . . . . .	18
2.2.1 Espaço de Formas . . . . .	20
2.2.2 Um Exemplo de SIA para Otimização: o Algoritmo opt-aiNet . . . . .	23
2.2.3 Modelos Alternativos de Rede Imunológica . . . . .	28
2.3 Considerações Finais . . . . .	33
<b>3 O Algoritmo <i>cob-aiNet</i></b>	<b>35</b>
3.1 Posicionamento da Proposta . . . . .	36
3.1.1 Posicionamento Frente a Outras Meta-heurísticas . . . . .	36
3.1.2 Posicionamento Frente a Outras Ferramentas de Otimização . . . . .	40
3.2 O Modelo de Rede Imunológica . . . . .	42
3.2.1 O Modelo de Bersini . . . . .	42
3.2.2 Adaptações ao Modelo de Bersini para Otimização . . . . .	45
3.3 O Algoritmo <i>cob-aiNet</i> . . . . .	47
3.3.1 Clonagem e Hipermutação . . . . .	49
3.3.2 Seleção e Inserção de Novos Indivíduos . . . . .	52
3.3.3 Atualização de Concentração e Supressão . . . . .	54

3.3.4	Busca Local . . . . .	57
3.4	Noções Gerais de Complexidade do Algoritmo . . . . .	59
3.4.1	Geração da População Inicial . . . . .	59
3.4.2	Avaliação de Afinidade entre Células . . . . .	59
3.4.3	Clonagem . . . . .	60
3.4.4	Hipermutação . . . . .	60
3.4.5	Seleção com Inserção . . . . .	61
3.4.6	Atualização de Concentrações . . . . .	62
3.4.7	Busca Local . . . . .	63
3.4.8	Supressão . . . . .	63
3.4.9	Número Total de Comparações e Avaliações de Função-objetivo . . . . .	63
3.5	Considerações Finais . . . . .	64
<b>4</b>	<b>Resultados Experimentais – <i>cob-aiNet</i></b>	<b>67</b>
4.1	Análise da Distribuição Final da População no Espaço de Busca . . . . .	69
4.1.1	O Algoritmo dopt-aiNet . . . . .	70
4.1.2	Metodologia Experimental . . . . .	73
4.1.3	Apresentação dos Resultados e Discussão . . . . .	74
4.2	Análise de Sensibilidade . . . . .	84
4.2.1	Tamanho Inicial da População – $nAB$ . . . . .	85
4.2.2	Números Mínimo e Máximo de Clones – $nC^{min}$ e $nC^{max}$ . . . . .	87
4.2.3	Concentração Inicial – $C_0$ . . . . .	89
4.2.4	Limiar de Supressão – $\sigma_s$ . . . . .	89
4.2.5	Parâmetros Inicial e Final do Operador de Hipermutação – $\beta^i$ e $\beta^f$ . . . . .	92
4.2.6	Número de Iterações da Busca Local – $LS_{it}$ . . . . .	94
4.2.7	Frequência da Busca Local – $LS_{freq}$ . . . . .	95
4.2.8	Intervalo de Busca Inicial da Busca Local – $SR$ . . . . .	96
4.3	Avaliação da <i>cob-aiNet</i> em Espaços de Maior Dimensão . . . . .	97
4.3.1	Algoritmos Participantes da Competição . . . . .	99
4.3.2	Apresentação dos Resultados e Discussão . . . . .	101
4.4	Considerações Finais . . . . .	113
<b>5</b>	<b>O Algoritmo <i>cob-aiNet</i>[MO]</b>	<b>115</b>
5.1	Conceitos Gerais de Otimização Multiobjetivo . . . . .	116
5.1.1	Dominância e Dominância Restrita . . . . .	117
5.1.2	Conjunto Ótimo e Fronteira de Pareto . . . . .	118
5.2	Posicionamento da Proposta . . . . .	121
5.3	O Algoritmo <i>cob-aiNet</i> [MO] . . . . .	123
5.3.1	Métricas de Afinidade e Atualização de Concentrações . . . . .	124
5.3.2	Seleção e Inserção de Novos Indivíduos . . . . .	127
5.3.3	Busca Local . . . . .	128
5.4	Considerações Finais . . . . .	129

<b>6</b>	<b>Resultados Experimentais – <i>cob-aiNet[MO]</i></b>	<b>131</b>
6.1	Metodologia Experimental . . . . .	132
6.1.1	Problemas de Teste . . . . .	132
6.1.2	Algoritmos Utilizados nas Comparações . . . . .	134
6.1.3	Métricas de Avaliação e Metodologia . . . . .	136
6.2	Análise da Capacidade de Manutenção de Diversidade no Espaço de Variáveis . . . . .	139
6.3	Resultados para os Problemas ZDT . . . . .	152
6.4	Resultados para o Problema DTLZ2 . . . . .	163
6.5	Considerações Finais . . . . .	165
<b>7</b>	<b>Comentários Finais</b>	<b>169</b>
7.1	Discussão . . . . .	173
7.2	Perspectivas Futuras . . . . .	175
<b>A</b>	<b>Problemas de Teste</b>	<b>177</b>
A.1	Problemas de Teste para Otimização Mono-objetivo . . . . .	177
A.1.1	Problemas Monomodais . . . . .	178
A.1.2	Problemas Multimodais Básicos . . . . .	180
A.1.3	Problemas Multimodais Expandidos . . . . .	183
A.1.4	Problemas Multimodais Compostos Híbridos . . . . .	184
A.2	Problemas de Teste para Otimização Multiobjetivo . . . . .	194
A.2.1	Problemas Multimodais . . . . .	195
A.2.2	A Classe de Problemas ZDT . . . . .	196
A.2.3	Problema DTLZ2 . . . . .	200
<b>B</b>	<b>Trabalhos Publicados pelo Autor</b>	<b>203</b>
B.1	Periódicos . . . . .	203
B.2	Capítulos de Livros e Anais de Congressos, Simpósios ou Encontros . . . . .	203
B.3	Resumos Expandidos . . . . .	205
	<b>Referências Bibliográficas</b>	<b>207</b>
	<b>Índice Remissivo de Autores</b>	<b>227</b>



# Lista de Figuras

1.1	Exemplos de problemas de maximização multimodais em espaços bidimensionais contínuos. . . . .	5
2.1	Resposta imunológica adaptativa. . . . .	15
2.2	Mecanismos de resposta positiva (estímulo) e negativa (supressão) da rede imunológica. . . . .	17
2.3	Complementariedade geométrica parcial das formas de anticorpos e antígenos. . . . .	21
2.4	Resultados da aplicação do algoritmo <i>opt-aiNet</i> a dois problemas bidimensionais com múltiplos ótimos. . . . .	27
2.5	Evolução do número de indivíduos na população do algoritmo <i>opt-aiNet</i> ao longo das iterações para os problemas <i>Multi</i> e <i>Roots</i> . . . . .	28
3.1	Exemplos de afinidade em um plano bidimensional $10 \times 10$ , de acordo com o modelo de Bersini. . . . .	43
3.2	Ilustração da variação do parâmetro $\beta$ em função da iteração $t$ . . . . .	51
3.3	Ilustração de propostas alternativas de variação do parâmetro $\beta$ em função da iteração $t$ . . . . .	52
3.4	Ilustração da variação do tamanho da população durante uma única execução do algoritmo <i>cob-aiNet</i> para os problemas $F_{12}$ e $F_9$ . . . . .	55
3.5	Possíveis posicionamentos finais de soluções de acordo com o raio de supressão $\sigma_s$ . . . . .	56
4.1	Dois exemplos de distância de linha entre os pontos $P_1$ e $P_2$ no espaço de busca. . . . .	72
4.2	Representação gráfica das curvas de nível do problema $F_2$ em conjunto com as populações finais retornadas pelos algoritmos <i>cob-aiNet</i> , <i>cob-aiNet</i> sem busca local, <i>opt-aiNet</i> e <i>dopt-aiNet</i> . . . . .	76
4.3	Representação gráfica das curvas de nível do problema $F_4$ em conjunto com as populações finais retornadas pelos algoritmos <i>cob-aiNet</i> , <i>cob-aiNet</i> sem busca local, <i>opt-aiNet</i> e <i>dopt-aiNet</i> . . . . .	77
4.4	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelos algoritmos <i>cob-aiNet</i> , <i>cob-aiNet</i> sem busca local, <i>opt-aiNet</i> e <i>dopt-aiNet</i> . . . . .	78
4.5	Representação gráfica das curvas de nível do problema $F_{12}$ em conjunto com as populações finais retornadas pelos algoritmos <i>cob-aiNet</i> , <i>cob-aiNet</i> sem busca local, <i>opt-aiNet</i> e <i>dopt-aiNet</i> . . . . .	79
4.6	Representação gráfica das curvas de nível do problema $F_{15}$ em conjunto com as populações finais retornadas pelos algoritmos <i>cob-aiNet</i> , <i>cob-aiNet</i> sem busca local, <i>opt-aiNet</i> e <i>dopt-aiNet</i> . . . . .	80

4.7	Representação gráfica das curvas de nível do problema $F_{20}$ em conjunto com as populações finais retornadas pelos algoritmos cob-aiNet, cob-aiNet sem busca local, opt-aiNet e dopt-aiNet. . . . .	81
4.8	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do tamanho da população inicial $nAB$ . . . . .	86
4.9	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $nAB$ . . . . .	86
4.10	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do número mínimo de clones $nC^{min}$ . . . . .	87
4.11	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $nC^{min}$ . . . . .	88
4.12	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do número máximo de clones $nC^{max}$ . . . . .	88
4.13	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $nC^{max}$ . . . . .	89
4.14	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação da concentração inicial $C_0$ de cada célula. . . . .	90
4.15	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do limiar de supressão $\sigma_s$ . . . . .	91
4.16	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $\sigma_s$ . . . . .	91
4.17	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do parâmetro inicial do operador de hipermutação $\beta^i$ . . . . .	93
4.18	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $\beta^i$ . . . . .	93
4.19	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do parâmetro final do operador de hipermutação $\beta^f$ . . . . .	94
4.20	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do número de iterações da busca local $LS_{it}$ . . . . .	95
4.21	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $LS_{it}$ . . . . .	95
4.22	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação da frequência da busca local $LS_{freq}$ . . . . .	96
4.23	Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do intervalo de busca inicial da busca local $SR$ . . . . .	98
4.24	Representação gráfica das curvas de nível do problema $F_9$ em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet quando ajustado com diferentes valores para o parâmetro $SR$ . . . . .	98

4.25	Representação gráfica da progressão do erro absoluto nas iterações finais de uma execução do algoritmo cob-aiNet para os problemas $F_3$ e $F_5$ . . . . .	112
5.1	Exemplo de mapeamento de uma solução $(x_1, y_1)$ do espaço de variáveis bidimensional para o espaço de objetivos, também bidimensional. . . . .	119
5.2	Exemplos de fronteiras de Pareto para um problema biobjetivo hipotético. . . . .	120
6.1	Representação gráfica da métrica de hipervolume. . . . .	136
6.2	Representação gráfica da métrica de diversidade baseada em hipercubos. . . . .	138
6.3	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>Deb &amp; Tiwari</i> . . . . .	145
6.4	Representação gráfica, no espaço de variáveis, dos conjuntos de soluções não dominadas obtidos pela cob-aiNet[MO] e pelo omni-optimizer para o problema <i>Deb &amp; Tiwari</i> . . . . .	146
6.5	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>EBN</i> . . . . .	147
6.6	Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>EBN</i> . . . . .	148
6.7	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>Two-on-One 4</i> . . . . .	149
6.8	Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>Two-on-One 4</i> . . . . .	150
6.9	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>Lamé Supersphere</i> . . . . .	151
6.10	Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>Lamé Supersphere</i> . . . . .	152
6.11	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>ZDT1</i> . . . . .	154
6.12	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>ZDT2</i> . . . . .	156
6.13	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>ZDT3</i> . . . . .	157
6.14	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>ZDT4</i> . . . . .	157
6.15	Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema <i>ZDT6</i> . . . . .	158
6.16	Representação gráfica, no espaço de objetivos, dos conjuntos de soluções não dominadas retornados pelos algoritmos cob-aiNet[MO] e NSGA-II, para o problema <i>DTLZ2</i> . . . . .	165
A.1	Representação gráfica, em $\mathfrak{R}^2$ , das funções de teste $F_1$ a $F_{12}$ da competição de Otimização de Parâmetros Reais do IEEE <i>Congress on Evolutionary Computation (CEC)</i> de 2005. . . . .	179
A.2	Representação gráfica, em $\mathfrak{R}^2$ , das funções de teste $F_{13}$ a $F_{24}$ da competição de Otimização de Parâmetros Reais do IEEE <i>Congress on Evolutionary Computation (CEC)</i> de 2005. . . . .	185

A.3	Representação gráfica da fronteira de Pareto de cada problema da classe ZDT utilizado neste trabalho. . . . .	197
A.4	Representação gráfica da fronteira de Pareto do problema <i>ZDT6</i> e de 100 soluções uniformemente distribuídas no conjunto de Pareto do problema. . . . .	200
A.5	Representação gráfica da fronteira de Pareto do problema <i>DTLZ2</i> . . . . .	201

# Lista de Tabelas

4.1	Resumo das principais características de cada problema da competição CEC'2005. . . . .	68
4.2	Limiar de supressão $\sigma_s$ usado nas duas versões da cob-aiNet e na opt-aiNet. . . . .	73
4.3	Média $\pm$ Desvio Padrão dos erros absolutos entre as melhores soluções obtidas por cada algoritmo e o ótimo global de cada problema, para os experimentos de análise da distribuição final de soluções. . . . .	74
4.4	Média $\pm$ Desvio Padrão do número final de soluções retornadas por cada algoritmo nos experimentos de análise da distribuição final de soluções. . . . .	75
4.5	Média $\pm$ Desvio Padrão do número de iterações executadas por cada algoritmo a cada repetição, nos experimentos de análise da distribuição final de soluções. . . . .	83
4.6	Média $\pm$ Desvio Padrão do tempo computacional (em segundos) gasto por cada algoritmo a cada repetição do problema $F_9$ , em $\mathbb{R}^2$ . . . . .	84
4.7	Lista de valores padrão e valores testados dos parâmetros do algoritmo cob-aiNet, para os experimentos de análise de sensibilidade. . . . .	85
4.8	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 10$ e as melhores soluções obtidas pelos algoritmos BLX-GL50, BLX-MA, CoEVO, DE e DMS-L-PSO. . . . .	103
4.9	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 10$ e as melhores soluções obtidas pelos algoritmos EDA, G-CMA-ES, K-PCX, L-CMA-ES e L-SaDE. . . . .	104
4.10	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 10$ e as melhores soluções obtidas pelos algoritmos SPC-PNX, opt-aiNet, dopt-aiNet e cob-aiNet. . . . .	105
4.11	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 30$ e as melhores soluções obtidas pelos algoritmos BLX-GL50, BLX-MA, CoEVO, DE e DMS-L-PSO. . . . .	106
4.12	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 30$ e as melhores soluções obtidas pelos algoritmos EDA, G-CMA-ES, K-PCX, L-CMA-ES e L-SaDE. . . . .	107
4.13	Média $\pm$ Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com $D = 30$ e as melhores soluções obtidas pelos algoritmos SPC-PNX, opt-aiNet, dopt-aiNet e cob-aiNet. . . . .	108
4.14	Rank de cada algoritmo para os problemas de teste da competição CEC'2005 em $D = 10$ . . . . .	109

4.15	Rank de cada algoritmo para os problemas de teste da competição CEC'2005 em $D = 30$ . . . . .	110
6.1	Média $\pm$ Desvio Padrão das métricas de <i>hipervolume</i> , <i>espaçamento</i> , <i>espalhamento máximo</i> e <i>diversidade baseada em hipercubos</i> para os problemas <i>Deb &amp; Tiwari</i> , <i>EBN</i> , <i>Two-on-One 4</i> e <i>Lamé Supersphere</i> . . . . .	142
6.2	Ranks de cada algoritmo, frente aos demais, obtidos a partir das métricas de <i>hipervolume</i> , <i>espaçamento</i> , <i>espalhamento máximo</i> e <i>diversidade baseada em hipercubos</i> , para os problemas <i>Deb &amp; Tiwari</i> , <i>EBN</i> , <i>Two-on-One 4</i> e <i>Lamé Supersphere</i> . . . . .	143
6.3	Média $\pm$ Desvio Padrão dos valores, em porcentagem, da métrica de <i>Cobertura entre Dois Conjuntos</i> para os problemas <i>Deb &amp; Tiwari</i> , <i>Two-on-One 4</i> e <i>Lamé Supersphere</i> . . . . .	144
6.4	Média $\pm$ Desvio Padrão do número de soluções não dominadas retornado por cada algoritmo para os problemas <i>Deb &amp; Tiwari</i> , <i>EBN</i> , <i>Two-on-One 4</i> e <i>Lamé Supersphere</i> . . . . .	151
6.5	Média $\pm$ Desvio Padrão dos valores, em porcentagem, da métrica de <i>Cobertura entre Dois Conjuntos</i> $\mathcal{C}(A, B)$ para os problemas <i>ZDT1-4</i> . . . . .	155
6.6	Média $\pm$ Desvio Padrão dos valores, em porcentagem, da métrica de <i>Cobertura entre Dois Conjuntos</i> $\mathcal{C}(A, B)$ para o problema <i>ZDT6</i> . . . . .	156
6.7	Média $\pm$ Desvio Padrão das métricas de <i>hipervolume</i> , <i>espaçamento</i> e <i>espalhamento máximo</i> para os problemas <i>ZDT</i> . . . . .	159
6.8	Ranks de cada algoritmo, frente aos demais, obtidos a partir das métricas de <i>hipervolume</i> , <i>espaçamento</i> e <i>espalhamento máximo</i> para os problemas <i>ZDT</i> . . . . .	160
6.9	Média $\pm$ Desvio Padrão do número de soluções não dominadas retornado por cada algoritmo para os problemas <i>ZDT</i> . . . . .	161
6.10	Média $\pm$ Desvio Padrão do tempo computacional (em segundos) exigido pelos algoritmos <i>cob-aiNet[MO]</i> e <i>NSGA-II</i> para os problemas <i>ZDT</i> . . . . .	162
6.11	Média $\pm$ Desvio Padrão dos valores, em porcentagem, da métrica de <i>Cobertura entre Dois Conjuntos</i> $\mathcal{C}(A, B)$ para o problema <i>DTLZ2</i> . . . . .	164

# Lista de Pseudo-códigos

2.1	Estrutura geral do algoritmo opt-aiNet. . . . .	24
3.1	Dinâmica da rede idiotípica de acordo com o modelo de Bersini (2002). . . . .	44
3.2	Passos principais do algoritmo cob-aiNet. . . . .	48
3.3	Mecanismo de seleção e inserção de novos indivíduos. . . . .	53
3.4	Operador de busca local <i>LocalSearch1</i> . . . . .	58
5.1	Passos principais do algoritmo cob-aiNet[MO]. . . . .	124



# Lista de Acrônimos

AE	– Algoritmo Evolutivo
AG	– Algoritmo Genético
AINE	– <i>Artificial Immune Network</i> (algoritmo)
aiNet	– <i>Artificial Immune Network</i> (algoritmo)
ARB	– <i>Artificial Recognition Ball</i>
cob-aiNet	– <i>Concentration-based Artificial Immune Network</i> (algoritmo)
cob-aiNet[MO]	– <i>Concentration-based Artificial Immune Network for Multiobjective Optimization</i> (algoritmo)
copt-aiNet	– <i>Artificial Immune Network for Combinatorial Optimization</i> (algoritmo)
CLONALG	– <i>Clonal Selection Algorithm</i> (algoritmo)
DCA	– <i>Dendritic Cell Algorithm</i> (algoritmo)
dopt-aiNet	– <i>Artificial Immune Network for Dynamic Optimization</i> (algoritmo)
EAM	– Erro Absoluto Médio
EDA	– <i>Estimation of Distribution Algorithm</i>
GDEA	– <i>Genetic Diversity Evolutionary Algorithm</i> (algoritmo)
MMEA	– <i>Probabilistic Model-based Multiobjective Evolutionary Algorithm</i> (algoritmo)
MOEA/D	– <i>Multiobjective Evolutionary Algorithm based on Decomposition</i> (algoritmo)
NSGA	– <i>Non-dominated Sorting Genetic Algorithm</i> (algoritmo)
NSGA-II	– <i>Non-dominated Sorting Genetic Algorithm II</i> (algoritmo)
omni-aiNet	– <i>Artificial Immune Network for Omni Optimization</i> (algoritmo)
opt-aiNet	– <i>Artificial Immune Network for Continuous Optimization</i> (algoritmo)
OMO	– Otimização Multiobjetivo
RLAIS	– <i>Resource-Limited Artificial Immune System</i> (algoritmo)
SIN	– Sistema Imunológico Natural
SIA	– Sistemas Imunológicos Artificiais
SPEA2	– <i>Strength Pareto Evolutionary Algorithm 2</i> (algoritmo)
VIS	– <i>Vector Immune System</i> (algoritmo)



# Capítulo 1

## Introdução

Cada vez mais, técnicas de *otimização* (Nocedal & Wright, 2006) se fazem presentes em uma enorme variedade de aplicações, que vão desde a definição de portfólios de investimento em bancos (Bauerle & Rieder, 2004) até a determinação da melhor composição de ligas metálicas em metalurgia (Törrönen et al., 1985). Nestas aplicações práticas, geralmente o problema em questão é modelado matematicamente e submetido a alguma técnica de otimização, para que se possa obter uma possível solução. Tal solução é então analisada e avaliada por um usuário que domine o problema, sendo este usuário responsável por definir a validade e adequação da solução obtida, e também por determinar sua implantação ou não na prática.

Em muitas situações, é altamente desejável que a ferramenta de otimização seja capaz de retornar, além da solução ótima, múltiplas soluções distintas para o problema, mesmo que muitas destas soluções não correspondam aos melhores resultados possíveis, ou seja, mesmo que muitas delas correspondam a soluções *sub-ótimas*. A modelagem de um problema real dificilmente consegue englobar todas as possíveis restrições existentes e muito menos é capaz de prever eventuais *restrições temporárias* que possam vir a surgir. Diante disso, a obtenção de um *conjunto diverso de soluções* pode evitar a necessidade de constantes remodelagens do problema e re-execuções da ferramenta de otimização, já que o próprio usuário teria a possibilidade de selecionar, dentre as soluções retornadas, aquela que atendesse a todos os critérios exigidos pelo problema no momento de sua aplicação, caso tal solução exista.

Para ilustrar, tomemos o exemplo de uma transportadora que, diariamente, realiza entregas em várias cidades de uma região. Para otimizar a entrega das encomendas, durante as madrugadas uma ferramenta de otimização é executada para que se obtenham as melhores rotas possíveis que cada caminhão deverá seguir para realizar as entregas agendadas para o dia seguinte. Dado este cenário, suponhamos então que, em uma certa manhã, descobre-se que a estrada correspondente ao primeiro trecho a ser percorrido por um dos caminhões está interdita por causa de um acidente. Diante disto,

o gerente de logística da transportadora teria as seguintes opções:

1. Seguir o roteiro proposto pela solução obtida durante a noite e deixar o caminhão parado no engarrafamento até a estrada ser liberada, o que certamente atrasaria a entrega das encomendas e até mesmo poderia levar à impossibilidade de entrega de alguns pacotes;
2. Reformular o modelo do problema, de forma a eliminar a estrada interdita dos possíveis caminhos a serem percorridos, e re-executar a ferramenta de otimização, o que poderia levar várias horas e atrasar todas as entregas do dia;
3. Utilizar a experiência própria e determinar, de acordo com algum critério subjetivo, uma nova rota para o caminhão em questão; ou
4. Caso a ferramenta de otimização seja capaz de retornar múltiplas soluções candidatas distintas, escolher dentro deste conjunto uma solução que evite a rodovia em questão. Se tal solução não estiver disponível, escolher a alternativa que pelo menos percorra o trecho problemático no final da rota, quando a possibilidade do trânsito estar liberado é maior.

De todas as opções acima, as mais viáveis seriam as terceira e quarta opções. Caso a empresa decida por adotar a terceira abordagem em todas as situações que exijam modificações de rota de última hora, ela obrigatoriamente deverá contar com um gerente de logística que, além de ter grande experiência na área, deverá conhecer muito bem as rodovias e a região de atuação da transportadora como um todo, para que consiga definir uma alternativa de rota minimamente razoável em um tempo aceitável. Além disso, ao adotar esta prática a empresa passaria a depender excessivamente do conhecimento de poucos funcionários, o que pode ser considerado uma situação de risco, já que o impacto em seu processo logístico seria grande caso tais funcionários viessem a ficar subitamente impossibilitados de trabalhar.

A quarta alternativa seria a opção mais apropriada, já que reduziria consideravelmente a dependência da empresa do conhecimento e experiência específicos de algum funcionário e traria mais agilidade, uma vez que, na maioria dos casos, bastaria escolher a solução que melhor se adequasse às restrições de última hora que eventualmente viessem a surgir. No entanto, é importante ressaltar que nenhuma ferramenta com capacidade de obter múltiplas soluções consegue encontrar um conjunto de soluções que atenda a todas as possíveis *restrições eventuais* mas, ao se adotar uma modelagem cuidadosa do problema e um bom ajuste da ferramenta, o número de casos não cobertos tende a ser reduzido.

Apesar de o exemplo apresentado aqui tratar especificamente do problema de roteamento de veículos, as vantagens de se aplicar ferramentas de otimização capazes de retornar conjuntos de soluções

não estão restritas apenas a este problema. Praticamente todas as áreas que empregam técnicas de otimização em seus processos poderiam se beneficiar de tais ferramentas, o que motivou o desenvolvimento do trabalho que será apresentado nas próximas páginas: a proposta de duas novas ferramentas imunoinspiradas para otimização de problemas em espaços contínuos, uma mono-objetivo e a outra multiobjetivo, capazes de explorar a multimodalidade de tais problemas e retornar para o usuário um conjunto de soluções distintas.

Na Seção 1.1 a seguir, o aspecto da multimodalidade de problemas de otimização será tratado em mais detalhes, bem como os pontos principais do paradigma de *sistemas imunológicos artificiais* (SIAs – de Castro & Timmis, 2002b) e as razões que levaram à escolha deste paradigma como a base principal das ferramentas propostas neste trabalho. Já a Seção 1.2 apresentará tanto os objetivos quanto os resultados esperados para este trabalho, enquanto que a Seção 1.3 trará um resumo das contribuições que foram efetivamente alcançadas. Por fim, a Seção 1.4 descreverá a organização dos demais capítulos desta tese.

## 1.1 Fundamentação e Motivação

De maneira geral, o processo de *otimização* de um dado problema pode ser visto como uma *tentativa de maximizar as propriedades desejáveis do sistema e, simultaneamente, minimizar aquelas que são indesejáveis* (Price et al., 2005).

Para que seja possível otimizar um determinado problema real, o primeiro passo é modelá-lo matematicamente em um *problema de otimização*, através da definição da *função-objetivo* que deve ser maximizada ou minimizada (ou seja, qual é a propriedade do problema real que deve ser otimizada), das *variáveis* que serão ajustadas (ou seja, quais aspectos do problema real serão trabalhados para que se otimize a característica em questão) e de quais restrições estão associadas ao problema, que devem ser respeitadas pela solução obtida.

Um mesmo problema real pode ser modelado de diversas maneiras em um problema de otimização, seja através da utilização de conjuntos de variáveis distintos ou até mesmo de funções-objetivo diferentes, o que pode acabar resultando em problemas de otimização com *propriedades* bem distintas entre si. Estas propriedades caracterizam cada problema de otimização e podem, inclusive, afetar positiva ou negativamente o desempenho da técnica de otimização escolhida para desempenhar a tarefa. Dentre as principais propriedades de um problema de otimização (Price et al., 2005), podem-se destacar:

- **Quantização das variáveis:** corresponde ao tipo das variáveis do problema, ou seja, se elas são contínuas, discretas, pertencem a um conjunto finito ou até mesmo se elas são de tipos diferentes ou não.

- **Dependência entre variáveis:** caso não haja dependência entre as variáveis, cada uma delas pode ser otimizada independentemente e diz-se que a função objetivo é *separável*. Caso exista esta dependência, o valor ótimo de uma determinada variável dependerá do valor de outra(s) variável(eis) do problema.
- **Dimensionalidade:** corresponde ao número de variáveis do problema, ou seja, ao número de variáveis que definem a função-objetivo.
- **Modalidade:** indica a presença de um único (*problemas monomodais*) ou múltiplos ótimos locais e/ou globais (*problemas multimodais*).
- **Dependência do tempo:** a posição do ótimo do problema no espaço de busca é dita *estacionária* quando ela não varia com o tempo ou *não-estacionária* (ou *dinâmica*) quando há esta variação.
- **Presença de ruído:** quando não há a presença de ruído, o valor da função-objetivo será sempre o mesmo para um mesmo conjunto de variáveis. No caso oposto, a presença de ruído provocará uma variação deste valor, mesmo quando o conjunto de variáveis permanece inalterado.
- **Existência de restrições:** corresponde à presença ou não de restrições (tanto de igualdade quanto de desigualdade) associadas à função-objetivo e às variáveis do problema.
- **Diferenciabilidade:** indica se a função-objetivo é diferenciável ou não em todos os pontos do *espaço de busca*, ou seja, em todos os pontos do domínio do problema.

Dentre todas as propriedades apresentadas acima, a que está mais diretamente relacionada à questão da possibilidade de obtenção de um conjunto de soluções distintas para um problema de otimização é a *modalidade*. Problemas multimodais, ou seja, problemas que possuem múltiplos ótimos locais, apresentam superfícies de busca com vários picos e vales (vide Figura 1.1), sendo que cada um destes picos (no caso de maximização) ou vales (para minimização) corresponde a um ótimo local. Quando o problema que se está tratando é multimodal, a abordagem mais natural para obtenção de um conjunto de soluções distintas é a tentativa de se localizar o maior número possível de ótimos locais, garantindo-se assim que cada uma destas soluções corresponderá à melhor solução em sua vizinhança (ou pelo menos estará bem próxima disso).

As ferramentas exatas de otimização (Luenberger, 1984), utilizadas principalmente em matemática e pesquisa operacional, geralmente garantem a convergência de sua execução para uma solução do problema, o que é altamente desejável em qualquer situação. No entanto, estas técnicas também apresentam uma série de limitações que dificultam sua utilização em todo tipo de problema. O principal aspecto negativo das ferramentas exatas está no fato de cada uma delas ser voltada para tratar

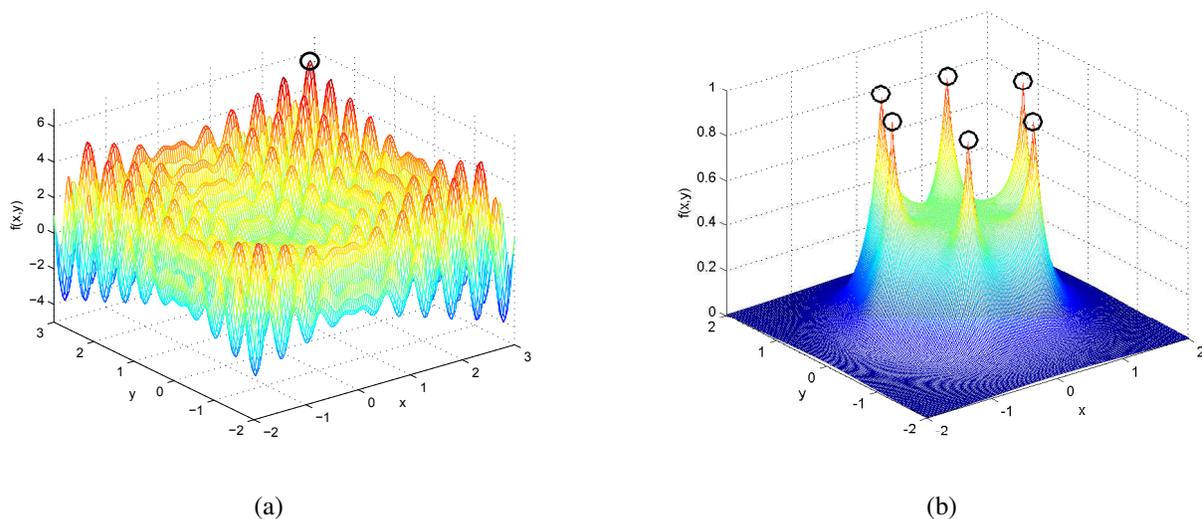


Fig. 1.1: Exemplos de problemas de maximização multimodais em espaços bidimensionais contínuos. No caso do problema (a), existem múltiplos ótimos locais de qualidades distintas, sendo que apenas o ótimo local marcado pelo círculo corresponde ao ótimo global do problema. Já para o caso (b), existem seis ótimos locais (marcados pelos círculos), e todos eles apresentam a mesma qualidade, ou seja, são ótimos globais. Estes dois problemas serão discutidos detalhadamente na Seção 2.2.2.

uma *classe específica* de problemas, exigindo que sejam atendidas uma série de restrições ou até mesmo que sejam feitas transformações no problema a ser resolvido para que ele possa ser tratado adequadamente (Neumaier, 2004). Além disso, uma outra característica presente na maioria destas ferramentas é o fato delas geralmente serem voltadas para obtenção de uma única solução, o que as torna inadequadas para os propósitos deste trabalho.

Dadas estas limitações, um paradigma de otimização que vem sendo cada vez mais utilizado é o das *meta-heurísticas populacionais*, que tem como alguns de seus principais representantes os *algoritmos evolutivos* (Bäck et al., 2000), os algoritmos de *otimização por enxame de partículas*<sup>1</sup> (Kennedy et al., 2001) e os *sistemas imunológicos artificiais* (de Castro & Timmis, 2002b). A principal vantagem destas meta-heurísticas populacionais está no fato de, como o próprio nome diz, trabalharem com uma *população de soluções-candidatas* ao problema, o que permite que tais ferramentas tenham uma menor susceptibilidade à presença de ótimos locais e também sejam capazes de obter múltiplos ótimos em uma única execução. No entanto, para que apresentem estas duas características, é preciso que tais algoritmos englobem mecanismos que permitam a *manutenção da diversidade* de suas soluções.

<sup>1</sup>Apesar dos algoritmos de otimização por enxame de partículas trabalharem com *enxames de partículas* e não com uma população de indivíduos propriamente dita, dado que tais enxames nada mais são que conjuntos de soluções-candidatas que evoluem através de interações entre si, neste trabalho estas ferramentas também serão consideradas como propostas de meta-heurísticas populacionais.

No caso específico de algoritmos evolutivos (Bäck et al., 2000) e de otimização por enxame de partículas (Kennedy et al., 2001), as propostas originais destas abordagens não preveem a presença de mecanismos de manutenção de diversidade, o que faz com que a maior parte dos indivíduos em suas populações convirja para uma única solução em problemas multimodais de otimização (Rogers & Prügel-Bennett, 1999). Para evitar que isto ocorra e também para dar a estas ferramentas uma maior capacidade de exploração do espaço de busca, aumentando assim a probabilidade de se encontrar o ótimo global do problema, foram propostos na literatura diversos mecanismos de estímulo à manutenção de diversidade (Esquivel & Coello Coello, 2003; Krohling & Coelho, 2006; Liang et al., 2006; Mori et al., 1996, 1998; Petrowski, 1996; Sareni & Krähenbühl, 1998), incluídos nas propostas originais como “adendos” aos operadores já existentes. Como resultado, estes mecanismos acabaram deixando tais ferramentas mais aptas a tratarem a questão da multimodalidade.

Já a maioria dos chamados sistemas imunológicos artificiais (SIAs – de Castro & Timmis, 2002b) não requer a adoção de mecanismos adicionais para manutenção de diversidade, uma vez que as ferramentas propostas de acordo com este paradigma se inspiram em mecanismos e teorias do sistema imunológico natural que podem levá-las a apresentar tal característica nativamente (de França et al., 2010a).

Por definição, os SIAs são ferramentas computacionais que buscam explorar e reproduzir conceitos e mecanismos do *sistema imunológico* (SI) natural dos vertebrados. Desde o início dos anos 2000, esta linha de pesquisa vem ganhando cada vez mais destaque e, com o passar do tempo, acabou se dividindo em duas frentes de pesquisa distintas:

- **Frente 1:** busca uma melhor compreensão do funcionamento do sistema imunológico natural como um todo e de seus mecanismos específicos, através do desenvolvimento e estudo de modelos computacionais;
- **Frente 2:** trata do desenvolvimento de *algoritmos imunoinspirados*, voltados para uma ampla gama de problemas reais, tais como análise de dados, otimização e outros (de França et al., 2010a).

Com relação à questão da manutenção de diversidade de soluções, diversos trabalhos da literatura mostraram que os algoritmos imunoinspirados, propostos dentro da segunda frente de pesquisa em SIAs e baseados na chamada *Teoria da Rede Imunológica* (Jerne, 1974), são de fato capazes de manter a diversidade de suas soluções ao longo da busca, retornando para o usuário um conjunto de soluções distintas em muitas das aplicações a que foram submetidos (de França et al., 2010a). No entanto, apesar de apresentarem resultados significativos com relação a este aspecto<sup>2</sup>, praticamente todas as ferramentas imunoinspiradas adotam uma implementação incompleta e muito simplificada

---

<sup>2</sup>O trabalho de de França et al. (2010b), inclusive, compara explicitamente um destes algoritmos imunoinspirados (no

dos mecanismos propostos pela Teoria da Rede Imunológica (de França et al., 2010a), o que indica que ainda existe espaço para melhorias.

Por outro lado, na primeira frente de pesquisa em SIAs, existem diversos trabalhos que propõem modelos computacionais mais elaborados dos mecanismos introduzidos pela Teoria da Rede Imunológica, modelos estes capazes inclusive de reproduzir comportamentos não presentes na maioria dos algoritmos imunoinspirados atuais. Diante disso, o principal objetivo deste trabalho é estudar e adaptar estes modelos mais elaborados, para que possam ser incorporados em algoritmos imunoinspirados voltados para otimização, de forma a permitir o desenvolvimento de novas ferramentas com maior capacidade de exploração do espaço de busca e melhor capacidade de localizar ótimos locais distintos do problema (ou seja, melhores mecanismos de manutenção de diversidade), permitindo assim que tais ferramentas sejam capazes de retornar múltiplas soluções de boa qualidade ao usuário. Essencialmente, busca-se aqui explorar contribuições recentes obtidas na literatura da Frente 1 de pesquisa em SIAs na proposição de inovações vinculadas à Frente 2 de pesquisa em SIAs.

O foco desta tese será em ferramentas desenvolvidas para tratar problemas de otimização em espaços contínuos, mas é importante ressaltar que os algoritmos propostos aqui podem ser estendidos para outros tipos de problemas de otimização, sendo necessário, para isso, a adaptação de alguns operadores sem que a estrutura geral das ferramentas tenha que ser profundamente alterada. No entanto, esta extensão foi deixada como perspectiva futura da pesquisa.

## 1.2 Objetivos do Trabalho

Como dito anteriormente, o objetivo principal deste trabalho é propor novos algoritmos imunoinspirados que, através da adoção de uma modelagem mais elaborada dos mecanismos propostos pela teoria da Rede Imunológica (Jerne, 1974), apresentem melhor capacidade de manutenção de diversidade, sem que sua capacidade de obtenção de boas soluções seja comprometida.

Para que este objetivo possa ser atingido, o primeiro passo será a revisão de propostas presentes tanto na literatura de SIAs quanto na literatura de imunologia teórica, que buscam modelar computacionalmente aspectos presentes no sistema imunológico natural (SIN). Em seguida, serão identificados quais destes modelos melhor se aplicam aos problemas de otimização em espaços contínuos, que são o foco deste trabalho. Por fim, serão identificados quais dos mecanismos propostos nestes modelos podem ser diretamente utilizados no tipo de problema em questão e quais necessitam de algum tipo de adaptação, definindo-se assim a estrutura geral das novas meta-heurísticas de otimização.

---

caso, o algoritmo *opt-aiNet* – de Castro & Timmis, 2002a), proposto para otimização em espaços contínuos, com a técnica de *Fitness Sharing* (Sareni & Krähenbühl, 1998), e conclui que, para o conjunto de problemas estudado, os resultados da *opt-aiNet* foram superiores em todos os aspectos, inclusive na manutenção de diversidade de soluções.

Formalizadas as novas ferramentas, a etapa seguinte será avaliar seus desempenhos em conjuntos abrangentes de problemas de teste, e comparar os resultados obtidos tanto com outros algoritmos imunoinspirados da literatura, que adotam mecanismos básicos similares, quanto com outras abordagens distintas, consideradas estado-da-arte dentro de suas respectivas áreas de atuação. Através desta série de experimentos, busca-se identificar quais são as características mais relevantes dos novos algoritmos, bem como suas principais vantagens e desvantagens.

### 1.3 Contribuições Alcançadas

De maneira geral, as principais contribuições obtidas com o desenvolvimento deste trabalho foram as seguintes:

- Análise e identificação dos aspectos associados à manutenção de diversidade, presentes em redes imunológicas artificiais para otimização, passíveis de serem melhorados;
- Revisão dos principais modelos de rede imunológica presentes na literatura de Sistemas Imunológicos Artificiais;
- Elaboração de um novo modelo de rede imunológica artificial para otimização em espaços contínuos, baseado no conceito de concentração de anticorpos;
- Proposição, formalização e validação de dois algoritmos de otimização capazes de manter a diversidade de soluções no espaço de busca:
  1. **cob-aiNet**: meta-heurística populacional (acrônimo do inglês *Concentration-based Artificial Immune Network*) proposta para otimização mono-objetivo em espaços contínuos. Esta ferramenta apresenta a dinâmica de sua população controlada por um novo modelo de rede imunológica, baseado na *concentração* de cada indivíduo da população, e introduz novos mecanismos de controle de diversidade.
  2. **cob-aiNet[MO]**: extensão da ferramenta cob-aiNet para problemas de otimização multi-objetivo em espaços contínuos.

### 1.4 Organização do Documento

Esta tese está dividida em sete capítulos. No Capítulo 2, será apresentada inicialmente uma visão geral do sistema imunológico natural (SIN) dos vertebrados, seguida de uma discussão sobre os

aspectos e teorias mais importantes deste sistema, que levaram ao surgimento dos chamados Sistemas Imunológicos Artificiais (SIAs). Os conceitos fundamentais de SIAs serão então detalhados, juntamente com o estudo de um exemplo de algoritmo imunoinspirado baseado na teoria da Rede Imunológica (Jerne, 1974) e voltado para otimização em espaços contínuos. Por fim, será feita uma revisão de alguns dos principais modelos computacionais de redes imunológicas, geralmente utilizados em simulações que visam tentar explicar o comportamento dinâmico de tais redes e estudar a influência que a variação de alguns aspectos específicos pode causar em seu comportamento.

Já no Capítulo 3, será apresentada a primeira proposta deste trabalho: o algoritmo cob-aiNet (*Concentration-based Artificial Immune Network*), desenvolvido para otimização de problemas mono-objetivo em espaços contínuos. Neste capítulo, a cob-aiNet será primeiramente posicionada frente a outras abordagens da literatura, e se seguirá a discussão detalhada do novo modelo de rede imunológica adotado e a formalização do algoritmo. Os resultados experimentais obtidos com o intuito de avaliar o desempenho da nova ferramenta e permitir a identificação de seus aspectos positivos e negativos serão discutidos no Capítulo 4.

O Capítulo 5, por sua vez, tratará da extensão do algoritmo cob-aiNet para otimização de problemas multiobjetivo, também em espaços contínuos. Para isso, será feita inicialmente uma breve discussão dos aspectos mais importantes desta classe de problemas e, em seguida, a ferramenta será posicionada frente a outras abordagens da literatura e formalizada. Os resultados experimentais obtidos com esta segunda ferramenta serão apresentados e discutidos no Capítulo 6.

Uma breve recapitulação do que foi proposto neste trabalho será feita no Capítulo 7, juntamente com os comentários finais sobre o trabalho e as perspectivas futuras da pesquisa. Por fim, no Apêndice A serão detalhados os problemas de teste utilizados nos Capítulos 4 e 6, enquanto que no Apêndice B estarão listadas todas as publicações do autor, realizadas até a data de conclusão desta tese.



## Capítulo 2

# Sistemas Imunológicos Artificiais

Um *Sistema Imunológico Artificial* (SIA – Dasgupta, 1998; de Castro & Timmis, 2002b) tem como meta reproduzir, em um contexto computacional, mecanismos e processos observados no sistema imunológico de organismos vertebrados. De acordo com de Castro & Timmis (2002b), o objetivo principal desta *imunoinspiração* é a resolução de problemas das mais diversas áreas, tais como engenharia, ciências médicas, economia e muitas outras (de Castro & Timmis, 2002b; de Castro, 2006). No entanto, é possível observar que, principalmente nos últimos anos, estes modelos computacionais do *Sistema Imunológico Natural* (SIN) também vêm sendo empregados como ferramentas de apoio ao estudo e compreensão do próprio SIN, através de sua utilização em validação de modelos conceituais e teorias via simulações (De Boer, 1989; Detours et al., 1994; Hart, 2006; Hart et al., 2007).

O primeiro trabalho a aplicar os conceitos de SIAs em computação foi proposto por Farmer et al. (1986). Neste trabalho, os autores propuseram um modelo dinâmico baseado na chamada *Teoria da Rede Imunológica* (Jerne, 1974) (ou *rede idiotípica*), que acabou se mostrando muito semelhante aos *sistemas classificadores* propostos por Holland (1984). Alguns anos depois, Bersini & Varela (1990) publicaram um outro trabalho, também sobre redes imunológicas, em que exploravam ainda mais a relação entre SIAs para aprendizagem de máquina e sistemas classificadores. No entanto, foi apenas em meados dos anos 90 que o paradigma de SIA passou a ser considerado como uma linha de pesquisa ativa, com a publicação dos trabalhos de Forrest et al. (1994), em que foram aplicados princípios de discriminação próprio/não-próprio (de Castro & Timmis, 2002b) a um problema de segurança computacional, e de Dasgupta & Forrest (1998), em que foi proposto o algoritmo de *Seleção Negativa*.

No final dos anos 90, de Castro & Von Zuben (1999) exploraram os conceitos do *Princípio da Seleção Clonal* (Burnet, 1978) e os aplicaram não apenas a aprendizado de máquina, mas também a problemas de otimização tanto em espaços contínuos quanto discretos. Este estudo de

de Castro & Von Zuben (1999) levou ao algoritmo CLONALG (*Clonal Selection Algorithm* – de Castro & Von Zuben, 2002b), que desde então vem sendo frequentemente aplicado em problemas reais.

Uma evolução do algoritmo CLONALG foi introduzida posteriormente por de Castro & Von Zuben (2002a) que, inspirados em mecanismos da *Teoria da Rede Imunológica* de Jerne (1974), propuseram o algoritmo *aiNet* (*Artificial Immune Network*). As principais diferenças entre os algoritmos *aiNet* e CLONALG são: (i) a existência de uma interação, em forma de *rede*, entre as soluções da população da *aiNet*; e (ii) a presença, também na *aiNet*, de mecanismos capazes de *suprimir* soluções semelhantes presentes na população, melhorando assim a utilização dos recursos computacionais. Estes mecanismos de interação em rede e supressão dão à *aiNet* uma capacidade de adaptação dinâmica do tamanho de sua população, o que permite o ajuste do número de soluções candidatas mantidas durante a busca às características do problema sendo tratado.

Originalmente, o algoritmo *aiNet* foi proposto para tratar problemas de agrupamento de dados (*clustering*), mas, diante dos resultados promissores obtidos, rapidamente surgiram diversas extensões para as mais variadas aplicações (de França et al., 2010a), tais como otimização de problemas em espaços contínuos (de Castro & Timmis, 2002a), otimização de problemas combinatórios (de Sousa et al., 2004; Gomes et al., 2004), bioinformática (Bezerra & de Castro, 2003; de Sousa et al., 2004; Gomes et al., 2004), sistemas de predição (Li et al., 2010), biclusterização (Castro et al., 2007a,c,b; Coelho et al., 2008, 2009a,b; de França et al., 2006a), otimização multiobjetivo (Coelho & Von Zuben, 2006b; Coelho et al., 2008), otimização dinâmica (de França et al., 2005, 2006b; Junqueira et al., 2005, 2006) e reconstrução de árvores filogenéticas (Coelho et al., 2007a,b, 2010).

Quase que paralelamente ao desenvolvimento da *aiNet*, Knight & Timmis (2001) propuseram o algoritmo AINE (*Artificial Immune Network*), que também se inspira nos princípios de Seleção Clonal e Rede Imunológica para tratar problemas de mineração de dados. Esta ferramenta utiliza o conceito de ARBs (do inglês *Artificial Recognition Balls*, que nada mais são que uma representação de regiões de reconhecimento por parte de anticorpos) e, diferentemente da *aiNet*, adota apenas sinais de *estímulo* em seu modelo, que podem ser provenientes tanto de antígenos quanto de outros anticorpos. A proposta da AINE foi posteriormente modificada por Timmis & Neal (2001) para trabalhar com uma quantidade limitada de recursos, ou seja, foi estipulado um limite no número de ARBs do modelo imunológico, o que levou a um novo algoritmo denominado RLAIS (*Resource-limited AIS*).

Além de algoritmos baseados nos princípios de Seleção Clonal (Burnet, 1978) e Rede Imunológica (Jerne, 1974), a literatura de SIAs também contém propostas baseadas em outros aspectos e teorias da imunologia, tais como o próprio algoritmo de *Seleção Negativa* (Dasgupta & Forrest, 1998), baseado principalmente nos mecanismos de seleção de células T no timo (de Castro & Timmis, 2002b), e os *Algoritmos de Células Dendríticas* (DCA – Greensmith, 2007), propostos para

detecção de anomalias e baseados na *Teoria do Perigo* (Matzinger, 1994; Aickelin & Cayzer, 2002). No entanto, como estas ferramentas e suas respectivas inspirações biológicas não estão diretamente associadas à proposta deste trabalho, elas não serão tratadas aqui em detalhes.

Este capítulo está estruturado em duas partes principais. Na Seção 2.1 será apresentada uma visão geral do sistema imunológico natural (SIN) dos vertebrados, seguida do detalhamento daqueles aspectos e teorias mais importantes e que serviram diretamente de inspiração para as ferramentas que serão detalhadas nos próximos capítulos. Já na Seção 2.2, serão discutidos os conceitos fundamentais de sistemas imunológicos artificiais (SIAs), para que fique claro como geralmente se dá a transposição dos conceitos presentes na imunologia teórica para os modelos computacionais. Para isso, serão tratados na Seção 2.2 os chamados *Espaços de Formas* (Perelson & Oster, 1979) e alguns modelos de rede imunológica da literatura, tanto aqueles utilizados em simulações de estudo do comportamento do sistema imunológico natural, quanto os que estão implementados em algoritmos para solução de problemas. Por fim, alguns comentários finais serão feitos na Seção 2.3.

## 2.1 O Sistema Imunológico Natural

Com o intuito de manter o organismo saudável, o *sistema imunológico natural* (SIN) possui mecanismos avançados de reconhecimento e combate a agentes infecciosos externos (chamados *patógenos*) que, por sua vez, possuem assinaturas (padrões) moleculares específicas (chamadas *antígenos*). Quando estes antígenos são reconhecidos pelas células do SIN, a *resposta imunológica*, ou seja, a reação de defesa do organismo é disparada.

Caso um agente externo consiga romper as barreiras física (pele) e bioquímicas (fluidos como saliva, lágrima e suor) de um organismo e finalmente invadí-lo, a resposta imunológica será disparada em duas frentes (Pinchuk, 2002): a primeira delas, chamada *resposta imunológica inata*, é disparada imediatamente pelo *sistema imunológico inato*; já a segunda, a *resposta imunológica adaptativa*, é uma resposta mais lenta e específica a cada tipo de patógeno, sendo disparada pelo *sistema imunológico adaptativo*. Juntos, os sistemas imunológicos inato e adaptativo formam um mecanismo de defesa extremamente eficiente e que opera em paralelo, recorrendo a uma diversidade de agentes e componentes distribuídos espacialmente e operando em rede.

As principais células do sistema inato são os chamados *fagócitos*, que possuem receptores capazes de reconhecer padrões moleculares presentes em agentes invasores e estão imediatamente disponíveis para combater eventuais infecções. Quando os fagócitos identificam algum agente invasor, eles *fagocitam* o agente identificado, ou seja, envolvem o patógeno e quebram sua estrutura molecular, simultaneamente estimulando a produção de *citocinas*. As citocinas emitem então sinais para outras células do sistema imunológico e induzem uma resposta inflamatória, elevando assim a temperatura

do corpo (febre). Com o aumento de temperatura, a atividade de alguns patógenos tende a diminuir, enquanto que a resposta adaptativa é estimulada (Leon, 2002).

Diferentemente da resposta imunológica inata, que é responsável por um combate inicial e rápido a qualquer infecção, a resposta adaptativa é bem mais específica a cada agente invasor e, conseqüentemente, requer um certo tempo para que possa começar a surtir efeitos. Enquanto as células do sistema inato são capazes de responder a uma ampla gama de agentes invasores sem a necessidade de uma exposição prévia a eles, as células do sistema adaptativo (responsáveis pela liberação dos chamados *anticorpos*) são ativadas e produzidas em resposta a infecções específicas. Em geral, as infecções se estabelecem toda vez que os agentes invasores não são devidamente neutralizados pelo sistema inato.

Dentre os vários componentes presentes no sistema imunológico adaptativo, os *linfócitos* merecem uma atenção especial. Existem dois tipos principais de linfócitos, os *linfócitos B* (ou células B) e os *linfócitos T* (ou células T), que possuem em sua superfície receptores de antígenos com alta especificidade. Estas células atuam na resposta adaptativa através do *reconhecimento* e da *eliminação* de patógenos, e também através da constituição da chamada *memória imunológica*. A memória imunológica corresponde basicamente à capacidade que as células do sistema adaptativo têm de reconhecer um mesmo antígeno (ou um antígeno semelhante) quando houver uma infecção recorrente, levando assim a uma resposta imunológica mais rápida e que pode até mesmo evitar o reestabelecimento da doença no organismo. Graças a estes mecanismos, a resposta adaptativa dá ao sistema imunológico a capacidade de *aprender e se aprimorar* a cada infecção sofrida.

Apesar da importância do sistema imunológico inato, o foco deste capítulo estará apenas nos mecanismos responsáveis pelo funcionamento da resposta imunológica adaptativa, que é a principal fonte de inspiração para os sistemas imunológicos artificiais. No entanto, é importante ressaltar que há uma forte interação dos sistemas inato e adaptativo em organismos vivos, e existem evidências claras de que os agentes responsáveis pelo disparo da resposta adaptativa são componentes do sistema imunológico inato (de Castro & Timmis, 2002b).

Cabe também destacar que as descrições de mecanismos e modelos do sistema adaptativo, constantes das próximas subseções, representam simplificações e aproximações dos mecanismos e processos reais que efetivamente têm lugar no SIN.

### 2.1.1 A Resposta Imunológica Adaptativa e o Princípio da Seleção Clonal

Quando um patógeno é “engolido” por um fagócito do sistema imunológico inato, sua estrutura molecular é fragmentada e o fagócito em questão passa a exibir, em sua superfície, peptídeos característicos deste patógeno, que são conhecidos como *antígenos* (Figura 2.1, passo I).

Uma vez completada a fagocitose de um agente invasor, os fagócitos se deslocam para os chamados *linfonodos*, que são nódulos espalhados pelo organismo que contêm uma alta concentração de

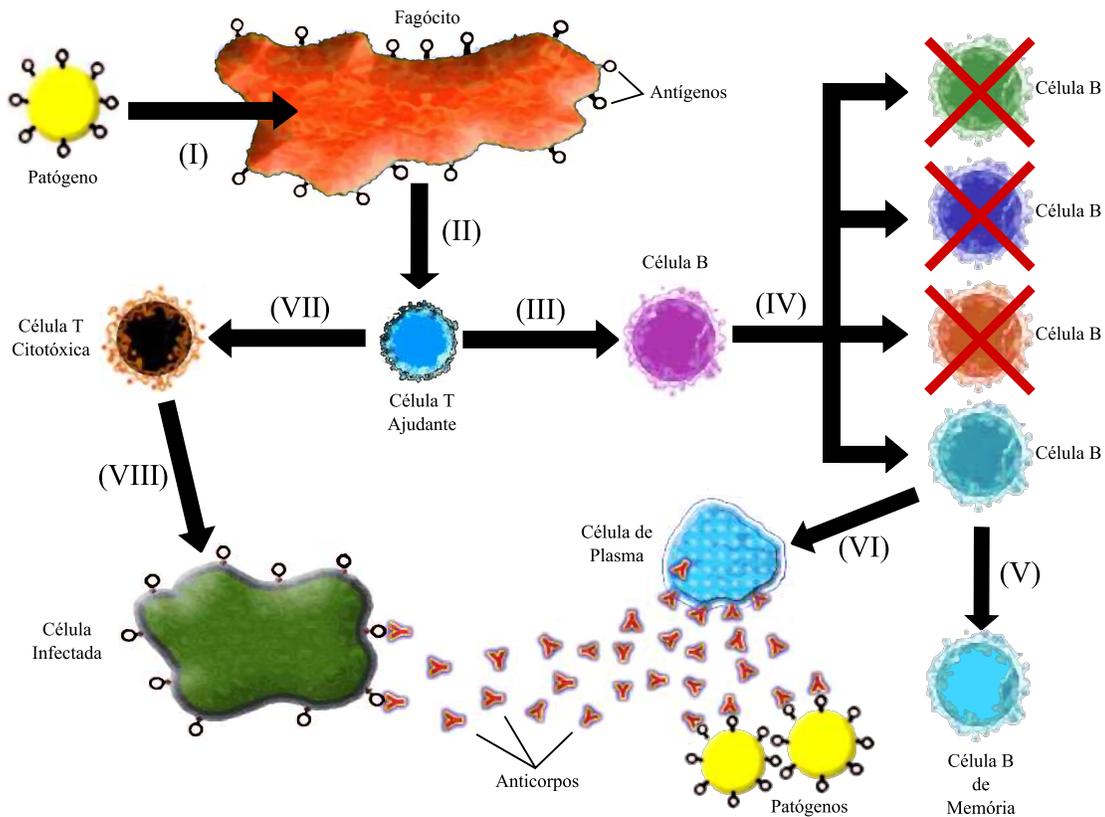


Fig. 2.1: Resposta imunológica adaptativa: (I) fagocitose e quebra do patógeno por fagócitos; (II) reconhecimento dos antígenos pelas células T ajudantes; (III) estímulo da produção de células B; (IV) maturação de afinidade; (V) diferenciação de parte das células B em células de memória; (VI) diferenciação de parte das células B em células de plasma; (VII) estímulo da produção de células T citotóxicas; e (VIII) eliminação de células do próprio organismo infectadas pelo patógeno. Esta figura ilustra uma situação hipotética em que o patógeno possui apenas um tipo de antígeno.

células do sistema imunológico. Nestes linfonodos, os antígenos que estão na superfície dos fagócitos são reconhecidos por receptores presentes nas *células T ajudantes* (também chamadas de *células T de defesa*), iniciando-se assim efetivamente a resposta adaptativa (Figura 2.1, passo II).

As células T ajudantes, por sua vez, possuem duas funções principais: (i) estimular a produção de *células B* (Figura 2.1, passo III); e (ii) estimular a produção de *células T citotóxicas* (Figura 2.1, passo VII).

Quando ocorre o estímulo para produção de células B, após a detecção de um antígeno, estas células passam a se reproduzir com maior velocidade, através de um processo de *clonagem*. Durante este processo, para que haja uma adaptação dos clones gerados aos antígenos invasores, ou seja, para que sejam produzidos anticorpos que tenham maior *afinidade* com os antígenos, os novos clones sofrem um processo de *maturação de afinidade* (Figura 2.1, passo IV), constituído basicamente de

duas etapas: *hipermutação e seleção*.

Durante a proliferação das células B, ocorre também um processo de *diferenciação* destas células, que pode levar tanto à criação de *células B de memória* (Figura 2.1, passo V), que são responsáveis por garantir uma resposta mais rápida a futuras infecções por patógenos semelhantes, quanto à criação de *células de plasma* (ou plasmócitos – Figura 2.1, passo VI), que são as principais células secretoras de *anticorpos* do organismo. Os anticorpos nada mais são que estruturas moleculares que se *ligam* aos antígenos presentes tanto nos patógenos quanto nas células infectadas, funcionando basicamente como *sinalizadores* que indicam quais indivíduos devem ser eliminados do organismo. Tanto os anticorpos quanto os antígenos possuem composições físico-químicas bem definidas, de forma que quanto maior for a *afinidade* entre eles ou seja, quanto melhor for a *complementariedade* de suas propriedades físico-químicas, melhor será a qualidade da ligação.

Já na etapa de hipermutação, os clones de cada célula B têm sua estrutura modificada, com taxas de variabilidade genética inversamente proporcionais à sua afinidade com os antígenos, ou seja, aquelas células que possuem maior capacidade de reconhecimento do antígeno sofrem uma variação genética menor, enquanto que as células não tão afins sofrem maior variação. Por fim, dentre todas as novas células geradas, aquelas que ainda possuírem baixa afinidade com os antígenos e as que, através da hipermutação, porventura se tornaram prejudiciais ao organismo, são então eliminadas do repertório.

Este processo de *expansão clonal, hipermutação e seleção* das células mais adaptadas é conhecido como *Princípio da Seleção Clonal* e foi proposto por Burnet (1978). O Princípio da Seleção Clonal é uma das principais inspirações presentes nos Sistemas Imunológicos Artificiais, e também foi adotado nas duas propostas que serão apresentadas nesta tese.

Por fim, na última etapa da resposta imunológica adaptativa, as células T citotóxicas, que também tiveram sua produção estimulada pelas células T ajudantes, identificam e eliminam as células do organismo já infectadas pelo patógeno (Figura 2.1, passo VIII).

Apesar da ilustração simplificada de resposta adaptativa dada na Figura 2.1 mostrar apenas uma situação hipotética, em que o patógeno apresenta um único tipo de antígeno, na prática um mesmo patógeno geralmente possui *múltiplos antígenos distintos*. Sendo assim, após a fagocitose, várias respostas imunológicas como as descritas aqui podem ser disparadas em paralelo (para antígenos distintos), já que as células B possuem uma propriedade conhecida como *monoespecificidade*, ou seja, cada uma delas é capaz de produzir um único tipo de anticorpo.

### 2.1.2 Teoria da Rede Imunológica

Pelo que foi apresentado na seção anterior, é possível concluir que o princípio da Seleção Clonal propõe que o sistema imunológico é composto por diversos componentes que permanecem em

repouso, até que uma resposta seja disparada por algum agente infeccioso. No entanto, existe também na literatura de SINs visões diferentes, dentre as quais se encontra a proposta de Jerne (1974), denominada *Teoria da Rede Imunológica* (também conhecida como *Teoria da Rede Idiotópica*).

Segundo Jerne, as células do sistema imunológico possuem estruturas (chamadas *idiotopos*) que permitem que elas também sejam reconhecidas pelos receptores de outras células e moléculas, o que tornaria o SIN uma enorme rede auto-organizável de componentes que continuamente reconhecem e são reconhecidos por outros componentes. Assim, de acordo com esta teoria, tal *rede imunológica* permaneceria em equilíbrio até que fosse perturbada pelo estímulo causado por algum antígeno externo, o que levaria a uma imediata alteração das *concentrações* dos componentes da rede de forma a reagir a este patógeno e estabelecer um novo estado de equilíbrio.

Existem basicamente dois mecanismos de resposta de acordo com a teoria da Rede Imunológica: *positiva* e *negativa* (vide Figura 2.2). Quando uma célula *reconhece* um antígeno de um patógeno ou um idiotopo de outro elemento da rede imunológica, ela é *estimulada por tais estruturas* e se torna *ativa*, dando início ao processo de clonagem, mutação e secreção de anticorpos. A esta ativação de uma célula imunológica provocada por um estímulo proveniente tanto de agentes externos quanto de agentes internos à rede dá-se o nome de *resposta positiva*. Por outro lado, quando uma determinada célula é *reconhecida* por outro elemento da rede, sua resposta imunológica é *suprimida*, o que pode levá-la a tolerar estímulos recebidos e até mesmo a ser eliminada da rede. Este segundo tipo de resposta, disparado pelo reconhecimento de uma célula por outros componentes do SIN, dá-se o nome de *resposta negativa*.

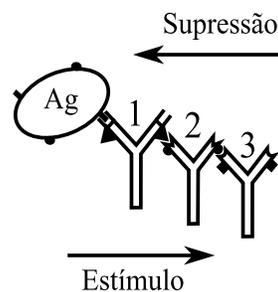


Fig. 2.2: Mecanismos de resposta positiva (estímulo) e negativa (supressão) da rede imunológica. Nesta figura, o reconhecimento do antígeno *Ag* provoca o estímulo da célula 1, enquanto que o reconhecimento da célula 1 provoca o estímulo da célula 2 e assim por diante. No sentido inverso, a célula 1 pode ter sua resposta suprimida ao ser reconhecida pela célula 2, que por sua vez também pode ser suprimida ao ser reconhecida pela célula 3. A supressão da resposta destas células pode inclusive levar à tolerância ao estímulo disparado pelo antígeno.

Sendo assim, a teoria da Rede Imunológica propõe que as principais características do sistema imunológico, tais como aprendizado, memória, auto-tolerância, tamanho e diversidade do repertório de células, são *propriedades emergentes* resultantes de mecanismos regulatórios que buscam manter

a rede em um certo nível de atividade (de Castro & Timmis, 2002b).

Ao ser proposta, a teoria de Jerne rapidamente alcançou grande sucesso e praticamente dominou o campo da Imunologia até os anos 90, quando então passou a ser mais questionada principalmente graças à ausência de evidências empíricas capazes de comprovar a existência dos mecanismos propostos. No entanto, nos últimos anos esta teoria vem despertando novamente o interesse da comunidade científica, principalmente em contextos clínicos (Behn, 2007).

Apesar da grande variação de popularidade da teoria da Rede Imunológica entre imunologistas, ao longo da última década, entre engenheiros e cientistas da computação a situação é diferente, já que, como será visto nas próximas seções, a proposta de Jerne é capaz de prover mecanismos que estimulam a diversidade entre indivíduos em algoritmos populacionais, o que a levou a servir de fonte de inspiração para diversas propostas de ferramentas imunoinspiradas bem sucedidas.

## 2.2 Os Sistemas Imunológicos Artificiais (SIAs)

De acordo com de Castro & Timmis (2002b), sistemas imunológicos artificiais (SIAs) podem ser definido como:

*“Sistemas adaptativos, inspirados pela imunologia teórica e observações de funções, princípios e modelos imunológicos, que são aplicados à resolução de problemas.”*

Dessa forma, para que um sistema possa ser caracterizado como um SIA, é necessário que ele englobe um modelo de um componente do SIN (por exemplo, uma célula, molécula ou órgão), que tenha sido desenvolvido através da incorporação de ideias da imunologia teórica e/ou experimental, e que tenha o objetivo de resolver problemas. Em outras palavras, não basta apenas atribuir uma *terminologia imunológica* para que um sistema seja considerado um SIA.

Se o sistema imunológico natural fosse considerado apenas como um sistema de reconhecimento de padrões, responsável pela identificação e eliminação de elementos perigosos ao organismo, o escopo de SIAs ficaria limitado apenas a tarefas de reconhecimento de padrões, particularmente aplicadas à segurança computacional. No entanto, o SIN possui uma grande quantidade de características interessantes computacionalmente, dentre as quais podemos ressaltar (Castro, 2009):

- **Reconhecimento de Padrões:** o SIN é capaz de reconhecer células e moléculas não pertencentes ao organismo, e eliminar aquelas potencialmente danosas.
- **Detecção de Anomalias:** o SIN é capaz de reconhecer e reagir a agentes infecciosos mesmo sem que tenha ocorrido nenhum contato prévio.

- **Detecção Imperfeita (ou tolerância a ruído):** o reconhecimento de um agente infeccioso não precisa ser perfeito para que a resposta imunológica seja disparada, ou seja, não é necessário que os receptores presentes nas células do SIN sejam exatamente complementares aos padrões moleculares dos antígenos do patógeno para que haja a ativação das células imunológicas.
- **Diversidade de Componentes:** o SIN não é constituído de apenas um componente que atua sozinho. Existe uma grande variedade de elementos (células, moléculas, etc.) que possuem funções específicas e, quando atuam em conjunto, levam à resposta imunológica.
- **Aprendizado e Memória:** graças ao mecanismo de adaptação de células B ao antígeno invasor e à sua diferenciação em células B de memória (vide Seção 2.1.1), o SIN é capaz de aprender a estrutura molecular de cada patógeno e se *lembrar* de tal assinatura em futuras infecções, o que permite uma resposta mais rápida.
- **Descentralização:** as células do SIN estão espalhadas por todo o organismo e não existe um elemento central que as controle.
- **Auto-organização:** quando ocorre a infecção do organismo por um agente infeccioso, a resposta do SIN é *emergente*, ou seja, não existe uma definição prévia e rígida de como os agentes imunológicos devem atuar em cada caso.
- **Variação Dinâmica de Componentes:** apesar de o sistema imunológico possuir uma grande quantidade de elementos, tal repertório é finito e incapaz de reconhecer todos os possíveis patógenos existentes. Diante disso, o sistema imunológico promove constantes mudanças em seu repertório de células, para que seja capaz de se adaptar a novas situações adversas o mais rápido possível.

Características como as apresentadas acima são indicadores de que os SIAs representam um paradigma que pode ser aplicado em diversas áreas, o que realmente vem ocorrendo nos últimos anos e está evidente na literatura, de onde pode-se destacar aplicações em:

- **Reconhecimento de Padrões:** Carter (2000), Carvalho & Freitas (2001), Dasgupta et al. (1999), Tarakanov et al. (2000);
- **Segurança Computacional:** Dasgupta (1999), Greensmith (2007), Hofmeyr & Forrest (2000), Kephart (1994), Kim & Bentley (1999), Somayagi et al. (1998);
- **Detecção de Anomalias:** Aisu & Mizutani (1996), Dasgupta (1996), Dasgupta & Forrest (1998), Greensmith & Aickelin (2008), Li et al. (2006), McCoy & Devarajan (1997);

- **Otimização:** Bersini & Varela (1991), Castro (2009), Coelho & Von Zuben (2006b), Coelho & Von Zuben (2010), de Castro & Timmis (2002a), de França et al. (2005), Freschi & Repetto (2006), Gaspar & Collard (2000), Gomes et al. (2004), Hajela & Yoo (1999), Toma et al. (1999);
- **Aprendizagem de Máquina:** Bersini (1998), Bezerra & de Castro (2003), Castro et al. (2005), Castro et al. (2010), de Castro & Von Zuben (2002a), Hunt & Cooke (1996), Li et al. (2010), Xu et al. (2006), Sahan et al. (2004), Timmis et al. (2000), Watkins (2001), Yue et al. (2007);
- **Controle e Robótica:** Bersini (1991), Cazangi & Von Zuben (2006), Ishiguro et al. (1998), Lee & Sim (1997), Michelan & Von Zuben (2002), Mitsumoto et al. (1996), Ootsuki & Sekiguchi (1999);
- **Memória Associativa:** Abbattista et al. (1996), Gibert & Routen (1994);
- **Ciências Biológicas:** Bersini & Varela (1994), Coelho et al. (2007a), Coelho et al. (2007b), Coelho et al. (2010), Walker (2001); e
- **Síntese Sonora:** Caetano (2006).

Na maior parte das aplicações mencionadas acima, particularmente nos casos em que há propostas alternativas de solução fundamentadas em outros paradigmas, podem ser encontradas comparações de desempenho e de princípios de operação. Sem entrar em aspectos específicos dessas comparações, cabe mencionar que os SIAs têm se mostrado competitivos e flexíveis a ponto de sustentarem uma ampla gama de aplicações concebidas ao longo de duas décadas de pesquisa.

### 2.2.1 Espaço de Formas

Observando o comportamento da resposta imunológica adaptativa, tanto de acordo com o princípio da Seleção Clonal (Seção 2.1.1) quanto com a teoria da Rede Imunológica (Seção 2.1.2), é possível notar que um dos aspectos mais importantes é a questão da *afinidade*, seja ela entre um anticorpo e um antígeno ou entre dois anticorpos<sup>3</sup>. Sendo assim, para que fosse possível descrever quantitativamente as interações de moléculas do SIN, Perelson & Oster (1979) introduziram o conceito de *espaço de formas*, que foi desenvolvido a partir de estudos teóricos sobre os mecanismos de seleção clonal.

---

<sup>3</sup>Geralmente, na literatura de SIAs, não há uma distinção explícita entre uma célula do sistema imunológico e os seus respectivos anticorpos, o que faz com que os termos “célula” e “anticorpo” acabem sendo empregados como sinônimos, referindo-se genericamente ao componente do SIN adotado no modelo em questão.

A afinidade entre um anticorpo e um antígeno ou entre dois anticorpos envolve vários processos físico-químicos, tais como interações covalentes entre as moléculas, pontes de hidrogênio e ligações de van der Waals. Sendo assim, para que tais ligações possam ser estabelecidas e o reconhecimento ocorra, é necessário que ambas as moléculas possuam estruturas *complementares* por longos trechos de sua superfície, como representado pictoricamente na Figura 2.3, a qual considera apenas aspectos geométricos. Dessa forma, é possível definir a *afinidade* entre um anticorpo e um antígeno ou entre dois anticorpos como sendo o *grau de complementariedade* que as duas moléculas apresentam entre si. Caso um anticorpo e um antígeno (ou dois anticorpos) não sejam perfeitamente complementares entre si, como é o caso da Figura 2.3, a ligação entre eles mesmo assim poderá ocorrer, embora com menor afinidade.

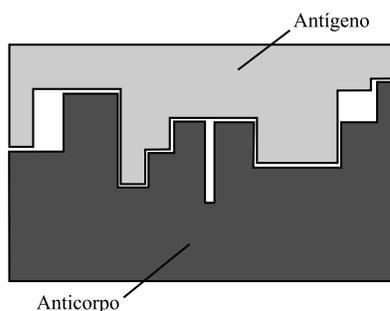


Fig. 2.3: Complementariedade geométrica parcial das formas de anticorpos e antígenos. Quanto melhor for esta complementariedade, maior será a afinidade. Esta figura apresenta uma visão pictórica de interações físico-químicas de compostos moleculares.

Tanto a forma e a distribuição de cargas elétricas, quanto a existência de grupos químicos em posições complementares na superfície dos antígenos e anticorpos são essenciais para a determinação de suas interações, e podem ser consideradas como propriedades destas moléculas. A este conjunto de atributos, dá-se então o nome de *forma generalizada*. Já do ponto de vista computacional, supondo-se que a forma generalizada de um anticorpo ou antígeno possa ser descrita por um conjunto de  $L$  atributos, o espaço  $L$ -dimensional dentro do qual esta forma generalizada corresponde a um ponto específico é denominado *espaço de formas*.

A definição de tais atributos para os modelos de anticorpos e antígenos se dá basicamente no domínio da aplicação a que se destina o sistema imunológico artificial e independentemente da inspiração biológica, de forma que a equivalência biológica entre estes parâmetros e os atributos da forma generalizada não é relevante.

Matematicamente, a forma generalizada de qualquer molécula  $\mathbf{m}$  em um espaço de formas  $S$  pode ser representada como um vetor de atributos de tamanho  $L$ , dado por  $\mathbf{m} = [m_1, m_2, \dots, m_L]$ . Tal vetor pode ser composto por atributos de qualquer tipo, como valores reais, inteiros, bits e símbolos,

o que definirá o espaço de formas a ser adotado da seguinte maneira:

- **Espaço de Formas Real:** onde os vetores de atributos são vetores de números reais;
- **Espaço de Formas Inteiro:** onde os vetores de atributos são vetores de números inteiros;
- **Espaço de Formas de Hamming:** onde os vetores de atributos são compostos por elementos pertencentes a um alfabeto finito de tamanho  $k$ ;
- **Espaço de Formas Simbólico:** onde os vetores de atributos são geralmente compostos por elementos simbólicos, tais como um *nome*, uma *cor*, etc; e
- **Espaço de Formas Híbrido:** onde os vetores de atributos agregam elementos pertencentes a mais de um dos tipos acima.

Além disso, outro aspecto crucial que está associado à definição do tipo de vetor de atributos de um sistema imunológico artificial é a definição de quais métricas serão usadas para medir as interações de indivíduos (anticorpos e/ou antígenos), ou seja, quais métricas serão empregadas para avaliar o *grau de afinidade* entre os componentes do SIA. Sendo assim, se considerarmos que as formas generalizadas de um anticorpo e de um antígeno (ou de outro anticorpo) são de mesmo tamanho (o que no caso biológico não ocorre necessariamente), sob a perspectiva de reconhecimento de padrões a *medida de afinidade* entre estes dois indivíduos é avaliada a partir de alguma métrica de *distância*, definida no espaço de formas a que pertencem os correspondentes vetores de atributos.

Tal medida de afinidade realiza um mapeamento da interação de anticorpos com antígenos em um número real não negativo que corresponde à *afinidade* ou ao *grau de similaridade* entre tais componentes. Sendo assim, como a afinidade é então proporcional ao grau de *similaridade* entre os vetores de atributos, geralmente o conceito de *complementariedade* entre moléculas não é adotado em SIAs, embora seja possível definir uma relação direta entre ambos.

O termo *afinidade* é geralmente adotado em SIAs para quantificar o grau de reconhecimento entre um anticorpo e um antígeno, mas também é possível interpretar tal afinidade como um termo geral que indica a *qualidade* de um elemento do sistema imunológico em relação ao ambiente em que está inserido (de Castro & Timmis, 2002b). Esta abordagem alternativa do conceito de afinidade é empregada principalmente em SIAs voltados para otimização, onde não existe uma representação explícita de antígenos, e os anticorpos do modelo de sistema imunológico adotado correspondem a soluções candidatas para o problema em questão. Nestas aplicações, a afinidade dos anticorpos corresponde à sua *qualidade* frente ao problema, e pode ser avaliada diretamente pela função-objetivo a ser otimizada. Estabelecendo um paralelo com os algoritmos evolutivos (AEs), esta forma de avaliação de

afinidade entre anticorpos e antígenos é equivalente à utilização de *funções de fitness* em AEs, tanto que, em muitos trabalhos de SIAs para otimização, os dois termos são utilizados como sinônimos.

Neste trabalho, como será visto nos próximos capítulos, as duas interpretações de afinidade estarão presentes. Como serão tratados problemas de otimização, os anticorpos dos modelos imunológicos utilizados corresponderão a soluções candidatas para os problemas, e terão suas afinidades com os antígenos avaliadas através de métricas do tipo *fitness*. Por outro lado, serão adotados aqui modelos imunológicos também baseados na teoria da Rede Imunológica (Jerne, 1974), o que requer a adoção de uma métrica de afinidade *entre anticorpos*. Sendo assim, para atender adequadamente esta segunda exigência, será adotada aqui também uma métrica de afinidade que indica o grau de reconhecimento entre dois indivíduos da população.

### 2.2.2 Um Exemplo de SIA para Otimização: o Algoritmo opt-aiNet

Visando aplicar os conceitos apresentados nas seções anteriores e ilustrar como geralmente é feita a modelagem computacional dos conceitos extraídos da imunologia teórica, nesta seção será discutido o algoritmo *opt-aiNet* (*Artificial Immune Network for Continuous Optimization*), proposto por de Castro & Timmis (2002a) como a primeira extensão da aiNet (de Castro & Von Zuben, 2002a) voltada para otimização mono-objetivo em espaços contínuos. Justamente por ter sido o primeiro algoritmo voltado para otimização que se baseou com sucesso na teoria de Jerne (1974) para manter a diversidade entre os anticorpos, a grande maioria dos algoritmos imunoinspirados para otimização propostos posteriormente (Coelho & Von Zuben, 2006b; de França et al., 2005; de Sousa et al., 2004; Freschi & Repetto, 2006) acabou adotando basicamente o mesmo modelo que a opt-aiNet. Nesta seção, além da apresentação detalhada da opt-aiNet, será feita também uma discussão crítica de seus aspectos positivos e, principalmente, negativos, que acabaram estimulando o desenvolvimento das duas ferramentas que representam as principais contribuições desta tese.

Para que o algoritmo opt-aiNet (de Castro & Timmis, 2002a) possa ser apresentado, é preciso definir inicialmente qual a terminologia que será adotada aqui e nos demais capítulos desta tese:

- **Anticorpos ou células:** indivíduos da população, codificados em vetores de atributos reais (vide Seção 2.2.1);
- **Fitness ou Afinidade Anticorpo/Antígeno:** medida que indica o quão adaptado está um dado anticorpo da população para detectar um eventual antígeno. Como neste trabalho serão tratados apenas problemas de otimização, o conceito explícito de *antígeno* não será empregado, de forma que será então adotada a segunda interpretação de afinidade descrita na Seção 2.2.1;
- **Afinidade Anticorpo/Anticorpo:** medida de similaridade entre dois anticorpos da população. Como neste trabalho será adotada apenas a codificação de indivíduos em vetores de atributos

reais, qualquer métrica de distância entre dois vetores com estas características pode ser utilizada como métrica de afinidade entre duas células. Quanto maior a distância entre elas, menor será a afinidade.

Definida a terminologia a ser adotada, podemos então passar à discussão do algoritmo opt-aiNet propriamente dito, cuja estrutura geral é dada no Pseudo-código 2.1.

---

**Pseudo-código 2.1** Estrutura geral do algoritmo opt-aiNet.

---

**Parâmetros:**

- $nAB$ : número inicial de células na população;
- $nC$ : número de clones a ser gerado para cada célula em cada iteração;
- $\sigma_s$ : limiar de supressão;
- $\beta$ : parâmetro do operador de hipermutação;
- $d$ : porcentagem de novos indivíduos a ser inserida após a supressão;

1- Gerar aleatoriamente a população inicial de tamanho  $nAB$ ;

**enquanto** (critério de parada não estiver satisfeito) **faça**

2- Determinar a afinidade entre anticorpos/antígenos (*fitness*);

3- Para cada célula, gerar  $nC$  clones;

4- Aplicar, a cada clone, hipermutação com taxa de variabilidade inversamente proporcional ao *fitness*;

5- Para cada conjunto (célula original  $\cup$  clones), selecionar o melhor indivíduo para permanecer na população;

**se** (não houve mudanças significativas no *fitness* médio da população) **então**

6- Determinar a afinidade anticorpo/anticorpo;

7- Suprimir todos os anticorpos, exceto o de melhor *fitness*, dentre os que possuem afinidade maior que  $\sigma_s$ ;

8- Introduzir uma porcentagem de novas células aleatoriamente geradas;

**fim se**

**fim enquanto**

9- Determinar a afinidade anticorpo/anticorpo;

10- Suprimir todos os anticorpos, exceto o de melhor *fitness*, dentre aqueles com afinidade maior que  $\sigma_s$ ;

---

O primeiro passo do algoritmo é a criação dos indivíduos que constituirão a população inicial, que consiste basicamente na geração aleatória dos vetores de atributos reais, com distribuição uniforme, respeitando-se o domínio de cada atributo. Em seguida verifica-se se as condições de parada foram atendidas e, caso tenham sido, o algoritmo executa os passos finais (9 e 10) e retorna todos os indivíduos que ainda estão presentes na população. Tais condições de parada são definidas pelo usuário de acordo com o problema sendo tratado, mas geralmente consistem em um número pré-definido de iterações ou em um número máximo de avaliações da função-objetivo do problema.

Caso as condições de parada ainda não tenham sido atendidas, o algoritmo entra então em seu *loop* principal, que inicia-se com a avaliação do *fitness* de cada indivíduo da população. Em seguida, para cada uma das células é gerado um número  $nC$  de clones ( $nC$  é um parâmetro fixo definido

pelo usuário), e estes clones sofrem um processo de *hipermutação*. Na hipermutação, a variabilidade genética que deve ser sofrida por cada clone é inversamente proporcional ao *fitness* da célula que o originou, ou seja, quanto melhor for a célula geradora dos clones, menor será a variabilidade genética aplicada a estes clones. Esta etapa de hipermutação segue a Expressão 2.1.

$$\vec{A}b_{t+1}^i = \vec{A}b_t^i + \beta \cdot e^{-f_i^{Ag}(t)} \cdot \mathcal{N}^i(\vec{0}, \mathbf{I}), \quad (2.1)$$

onde  $\vec{A}b_{t+1}^i$  é o novo anticorpo  $i$  gerado a partir de  $\vec{A}b_t^i$ ,  $\mathcal{N}^i(\vec{0}, \mathbf{I})$  é um vetor aleatório de distribuição gaussiana com média  $\vec{0}$  e covariância  $\mathbf{I}$ , amostrado para cada indivíduo  $i$ ,  $\mathbf{I}$  é a matriz identidade,  $f_i^{Ag}(t)$  é o *fitness* do indivíduo  $i$  normalizado em  $[0; 1]$  e  $\beta$  é um parâmetro de controle da amplitude da função exponencial inversa. Este parâmetro  $\beta$  é um parâmetro estático, que deve ser definido pelo usuário, e altamente dependente do problema sendo tratado, o que dificulta seu correto ajuste. Caso  $\beta$  seja definido com um valor muito alto, o algoritmo permanece, em toda a sua execução, com um comportamento altamente exploratório que pode acabar impedindo a sua convergência, enquanto que a atribuição de um valor muito baixo de  $\beta$  pode acabar levando o algoritmo a convergir prematuramente para um ótimo local. Existem algumas propostas na literatura (Coelho & Von Zuben, 2006a) que tentaram minimizar este problema mas, neste trabalho, uma nova abordagem para o controle de  $\beta$  foi proposta, como será discutido na Seção 3.3.

Uma vez aplicados os operadores de clonagem e hipermutação a todos os  $N$  indivíduos da população, a melhor célula de cada sub-conjunto de indivíduos formado pela célula original e seus clones (sub-população) é selecionada, de forma que, ao final da etapa de seleção, a população continue com o mesmo número  $N$  de indivíduos que tinha inicialmente.

Estes três passos básicos de clonagem, hipermutação e seleção (passos 3, 4 e 5 do Pseudo-código 2.1) correspondem basicamente à reprodução computacional do processo de micro-evolução proposto no princípio de Seleção Clonal (Burnet, 1978), descrito na Seção 2.1.1. Apesar da imunoinspiração, neste ponto o algoritmo opt-aiNet é muito similar aos algoritmos evolutivos, uma vez que tais mecanismos de micro-evolução acabam sendo modelados computacionalmente de maneira semelhante a operadores já existentes, baseados por sua vez nos mecanismos de seleção natural e evolução das espécies (Bäck et al., 2000; Darwin, 1859).

Por outro lado, a opt-aiNet também tem como fonte de inspiração a teoria da Rede Imunológica de Jerne (1974), cujos mecanismos estão representados nos passos 6–10 do Pseudo-código 2.1. Quando a evolução das células da população se estabiliza, ou seja, quando o *fitness* médio dos indivíduos não sofre grandes variações de uma iteração para a outra, todas as células da população são comparadas umas com as outras, através do cálculo da afinidade par a par entre anticorpos. Caso a afinidade entre dois anticorpos esteja acima de um dado limiar de supressão  $\sigma_s$  definido pelo usuário (ou seja, caso eles sejam muito semelhantes), o anticorpo de pior *fitness* é eliminado da população.

Após esta etapa de supressão de anticorpos semelhantes, é introduzida na população uma porcentagem  $d$  de novos indivíduos aleatoriamente gerados, o que contribui para o aumento da diversidade. Esta inserção de novas células no repertório do algoritmo opt-aiNet também possui inspiração no Sistema Imunológico Natural, uma vez que novas células imunológicas são constantemente geradas no organismo. Por fim, a execução retorna ao início do *loop* principal, onde são novamente verificadas as condições de parada.

Ao suprimir indivíduos semelhantes da população, o algoritmo opt-aiNet evita redundância desnecessária e libera recursos computacionais para que os novos indivíduos possam possivelmente explorar regiões do espaço de busca ainda inexploradas, aumentando assim a possibilidade de se encontrar novas regiões promissoras. No entanto, como a estratégia de inserção de novos indivíduos consiste apenas na inserção de um número pré-definido de indivíduos aleatoriamente gerados, o algoritmo fica completamente dependente da “sorte” para que tais indivíduos caiam em regiões interessantes, o que não é desejável.

Por outro lado, por possuir mecanismos que lhe permitem (i) eliminar indivíduos redundantes, (ii) periodicamente inserir novas células na população e (iii) refinar a busca através da evolução independente de diversas pequenas sub-populações, que só interagem entre si na etapa de supressão, a opt-aiNet é um algoritmo que reconhecidamente apresenta uma boa capacidade de manutenção da diversidade de suas soluções no espaço de busca (Coelho & Von Zuben, 2006a; de França et al., 2010a,b), e isto está diretamente relacionado à sua inspiração na teoria da Rede Imunológica.

Para ilustrar a boa capacidade da opt-aiNet em manter a diversidade dos indivíduos em sua população no espaço de busca, tomemos dois problemas bidimensionais de maximização, conhecidos na literatura como função *Multi* (Equação 2.2) e *Roots* (Equação 2.3). Os resultados obtidos em uma única execução do algoritmo opt-aiNet estão apresentados na Figura 2.4. Em ambos os experimentos, foram utilizados os seguintes parâmetros: 20 células na população inicial; 10 clones gerados para cada célula na população; limiar de supressão  $\sigma_s = 0,1$ ;  $\beta = 0,01$ ; e critério de parada igual a 2000 iterações.

$$f(x, y) = x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) \quad x, y \in [-2, 3] \quad (2.2)$$

$$f(z) = \frac{1}{1 + |z^6 - 1|}, \quad z = x + iy, \quad (2.3)$$

$$x, y \in [-2, 2]$$

Como pode ser observado na Figura 2.4, o algoritmo opt-aiNet foi capaz de identificar múltiplos

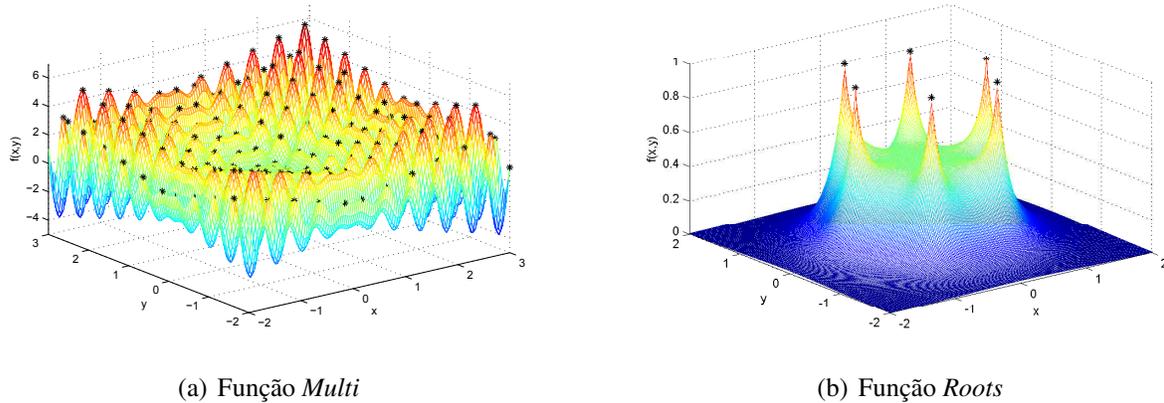


Fig. 2.4: Resultados da aplicação do algoritmo *opt-aiNet* a dois problemas bidimensionais e com múltiplos ótimos: (a) função *Multi* e (b) função *Roots* (a população final corresponde aos pontos em \*).

ótimos locais em cada problema e manter soluções em cada um destes picos até o final de sua execução. Particularmente nestes dois problemas de dimensão reduzida, todos os ótimos locais e globais foram encontrados.

Através destes dois exemplos didáticos é possível observar também uma característica da *opt-aiNet* que é comum a todos os demais algoritmos baseados na teoria da Rede Imunológica e derivados da *aiNet* (de Castro & Von Zuben, 2002a): a *variação dinâmica do tamanho da população* de acordo com as características do problema sendo tratado e com as preferências do usuário.

Na Figura 2.5 é apresentada a variação do tamanho da população da *opt-aiNet* ao longo de suas iterações, para os dois problemas tratados aqui. Para ambos os casos, foi definida uma população inicial de 20 indivíduos, e o algoritmo foi capaz de aumentar este número para cobrir todos os ótimos locais da função *Multi* (Figura 2.5(a)), e reduzi-lo para que pudesse manter em sua população apenas a quantidade necessária de indivíduos para cobrir os seis ótimos locais/globais do problema *Roots* (Figura 2.5(b)). Além disso, é interessante notar que, no final de sua execução, o algoritmo *opt-aiNet* executa uma última etapa de higienização do conjunto de soluções obtido, eliminando eventuais redundâncias que ainda permaneçam e provocando a redução brusca do número de indivíduos na última interação do algoritmo, verificada na Figura 2.5.

No entanto, é importante ressaltar aqui que essa variação do tamanho da população da *opt-aiNet* está fortemente associada ao parâmetro  $\sigma_s$  (limiar de supressão), que deve ser definido pelo usuário no início da execução do algoritmo. Caso seja definido um valor de  $\sigma_s$  muito grande (por exemplo, maior que o raio de um pico no problema *Multi*), o algoritmo poderá acabar suprimindo soluções que estejam em picos vizinhos (caso eles estejam mais próximos entre si que  $\sigma_s$ ), deixando assim de povoar alguns ótimos locais que poderiam ser relevantes para o problema em questão.

Apesar dos bons resultados ilustrados aqui nestes dois problemas didáticos, é possível notar que a

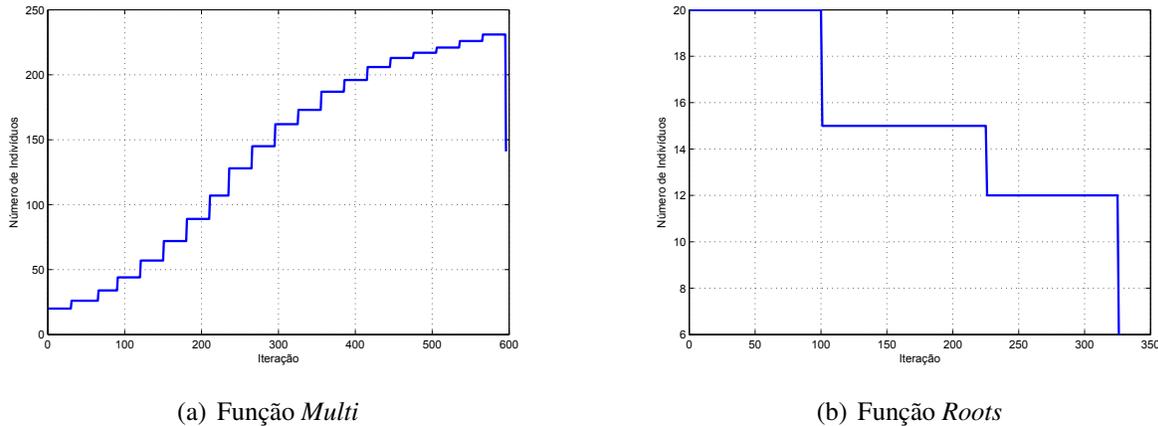


Fig. 2.5: Evolução do número de indivíduos (anticorpos) na população do algoritmo *opt-aiNet* ao longo das iterações, para os problemas (a) *Multi* e (b) *Roots*. O algoritmo foi inicializado com 20 indivíduos na população para ambos os problemas.

implementação, na *opt-aiNet*, dos mecanismos propostos pela teoria da Rede Imunológica é extremamente simples, restrita apenas a eventuais comparações entre todas as células da população (durante a execução) e a eliminação daquelas que estejam mais próximas de outras que um determinado limiar. Ou seja, em outras palavras, a *opt-aiNet* reproduz apenas os mecanismos de *supressão* propostos por Jerne (1974), deixando de lado o processo de *ativação* que pode ocorrer quando há o reconhecimento entre componentes do próprio sistema imunológico.

Este grau de simplificação do modelo imunológico não é uma característica apenas do algoritmo *opt-aiNet*, já que praticamente todos os demais algoritmos da literatura adotam uma mesma abordagem. No entanto, a literatura de sistemas imunológicos artificiais é extremamente rica em modelos mais elaborados de redes imunológicas, que são geralmente utilizados como ferramentas de auxílio à compreensão do funcionamento de mecanismos destas redes. Sendo assim, na próxima seção será feita uma breve revisão dos principais modelos da literatura, partindo-se do trabalho seminal de Farmer et al. (1986), um dos trabalhos pioneiros da área de SIAs.

### 2.2.3 Modelos Alternativos de Rede Imunológica

Existem também, na literatura de sistemas imunológicos artificiais, diversos trabalhos que se baseiam em modelos computacionais de redes imunológicas mais elaborados para, através de simulações destes modelos, tentar explicar o comportamento dinâmico de tais redes e estudar a influência que a variação de alguns aspectos específicos pode causar em seu comportamento. O intuito desta seção não é apresentar todos os modelos presentes na literatura e sim apenas os principais, para que se possa ter uma ideia das características gerais que tais modelos apresentam.

De maneira geral, a maioria dos modelos imunológicos da literatura apresenta duas partes principais, que controlam respectivamente a *dinâmica* e a *meta-dinâmica* do sistema. Compreende-se basicamente por *dinâmica* a forma com que se dá a variação com o tempo da *concentração* de componentes do sistema imunológico, sejam eles linfócitos ou anticorpos. Apesar deste conceito de *concentração* de componentes estar amplamente presente na modelagem da dinâmica do sistema imunológico, até então são poucas as propostas de ferramentas para resolução de problemas que o adotam (de Castro et al., 2003; Knidel, 2010), sendo que a maioria dos algoritmos imunoinspirados da literatura opta por modelar de maneira bem simples cada componente do sistema individualmente. Já a *meta-dinâmica* dos modelos do sistema imunológico corresponde à maneira com que é modelado tanto o processo contínuo de inserção de novos componentes na rede imunológica quanto a eliminação de células ou moléculas de seu repertório.

O primeiro trabalho a propor um modelo computacional simplificado da rede imunológica de Jerne (1974) foi o trabalho de Farmer et al. (1986). Neste modelo, células B e anticorpos foram considerados uma única estrutura, representada por vetores binários divididos em duas partes: *paratopos*, que são as estruturas que permitem reconhecer outras células e antígenos; e *epitopos*, que são as estruturas que permitem que uma dada célula ou antígeno seja reconhecido por outras células. Além disso, o reconhecimento entre duas células e entre células e antígenos é baseado no grau de *complementariedade* das cadeias binárias, podendo inclusive ocorrer um reconhecimento parcial entre dois componentes. Por fim, a dinâmica da rede imunológica modelada por Farmer et al. (1986) se dá através da variação da *concentração* de cada componente, que pode ser aumentada caso este componente reconheça outras células ou antígenos presentes na simulação, ou diminuída caso ele seja reconhecido por outras células do repertório. Em ambos os casos, esta variação na concentração é proporcional ao grau de complementariedade entre as cadeias binárias envolvidas nos processos de reconhecimento. Quando a concentração de um dado componente se torna inferior a um dado limiar, o componente é eliminado do repertório da simulação. Um modelo semelhante ao proposto por Farmer et al. (1986) foi utilizado por Martins & Slani (2005), na proposta de um sistema de vigilância de obras de arte para museus, e por Shomooka et al. (2002) para reconhecimento de caracteres chineses.

Já De Boer (1989) propôs um modelo com o objetivo de estudar como se dá a seleção de repertório de células B na presença de *antígenos próprios*, ou seja, antígenos que representam estruturas do próprio organismo. Como na proposta de Farmer et al. (1986), os componentes deste modelo também são representados por cadeias binárias e a afinidade entre eles é baseada no grau de complementariedade entre estas cadeias, que inclusive pode ser parcial. Neste modelo de De Boer (1989), existem três processos principais que determinam o controle do repertório: (i) a entrada de um fluxo de células recém-criadas na medula óssea; (ii) o decaimento (supressão) de células; e (iii) a proliferação de células. Além disso, a resposta de cada célula a um estímulo segue uma curva em forma de sino,

ou seja, interações dos componentes da rede podem levar a comportamentos distintos dependendo da concentração destes componentes na rede: caso um componente possua uma concentração intermediária, um estímulo pode disparar a proliferação desta célula, enquanto que caso a sua concentração seja muito alta ou muito baixa, este mesmo estímulo pode acabar levando a sua supressão.

Também com o objetivo de estudar o desenvolvimento de redes imunológicas no contexto de antígenos próprios, Detours et al. (1994) utilizaram um modelo computacional baseado em um espaço de formas bidimensional formado por dois planos ( $A$  e  $B$ ) com vizinhança toroidal. Neste modelo, cada um destes planos é discretizado em uma matriz  $L \times L$ , e só são permitidas conexões entre moléculas contidas no plano  $A$  e moléculas contidas no plano  $B$ . A afinidade entre duas moléculas passíveis de conexão decai exponencialmente com a distância entre elas. Com relação aos demais modelos descritos até o momento, a principal diferença do modelo de Detours et al. (1994) está no fato de esta proposta modelar tanto anticorpos livres no organismo quanto anticorpos presos à membrana de células B. Além disso, o modelo de Detours et al. (1994) define também expressões para a variação das concentrações de ambos os tipos de anticorpos e mecanismos que promovem a simulação de inserção de novos anticorpos produzidos pela medula óssea.

De Boer & Perelson (1991), por sua vez, propuseram um novo modelo de rede imunológica para estudarem a emergência de propriedades tais como tamanho e conectividade da rede. Neste trabalho, os autores também adotam cadeias binárias para representar antígenos, células B e anticorpos livres (como no modelo de Detours et al., 1994), e dividem seu modelo em duas partes principais: (i) produção de novas células B na medula óssea; e (ii) proliferação/morte de células B e secreção/remoção de anticorpos livres. A proliferação de cada célula B do sistema (através de clonagem) e a secreção de anticorpos é controlada por três variáveis: concentração de células B na rede imunológica, concentração de anticorpos livres e intervalo de tempo entre recebimento de estímulos pelas células B e a liberação de anticorpos. Por fim, a última característica relevante do modelo de De Boer & Perelson (1991) é a utilização de uma função de afinidade baseada em apenas três valores discretos: 0.0 (afinidade nula), 0.1 (afinidade baixa) e 1.0 (afinidade alta). Detours et al. (1996) adotaram basicamente o mesmo modelo de De Boer & Perelson (1991) e focaram seu trabalho justamente neste aspecto da discretização da função de afinidade, com o intuito de verificarem se a utilização de uma maior quantidade de valores próximos e equiespaçados de afinidade influenciaria negativamente a emergência das mesmas características observadas por De Boer & Perelson (1991), sendo que isto não ocorreu. Além disso, Detours et al. (1996) estudaram também a adoção de uma codificação em cadeias hexadecimais para os componentes do sistema, e observaram que o uso desta nova codificação levou a uma distribuição de valores de afinidade mais suave entre as células da rede.

Já o modelo de Bersini (2003) foi proposto com o intuito de estudar a emergência, em redes imunológicas, de comportamentos tais como capacidade de não reconhecimento de antígenos próprios e

memória. Nesta proposta, os componentes da rede (anticorpos e antígenos) também são modelados como cadeias binárias e a afinidade entre anticorpos e entre anticorpos e antígenos é proporcional tanto à sua distância de Hamming (desde que estejam mais próximos que um dado limiar) quanto à concentração de cada componente. Com relação ao modelo de concentrações, os anticorpos podem ter sua concentração aumentada ou diminuída conforme sua afinidade total (soma da afinidade com os antígenos e com outros anticorpos). Esta variação de concentração de anticorpos é equivalente a uma versão simplificada da abordagem de De Boer (1989), baseada em uma curva em forma de sino: no trabalho de Bersini (2003), a concentração de um anticorpo aumenta apenas caso sua afinidade total esteja dentro de uma faixa determinada por dois limiares, um superior e um inferior; caso contrário, sua concentração diminui. Já no caso de antígenos, sua concentração sempre diminui, proporcionalmente à sua afinidade com os anticorpos da rede.

Um modelo semelhante ao de Bersini (2003) também foi adotado em Hart et al. (2007), com o intuito de se verificar como a definição da métrica de afinidade entre componentes afeta o surgimento de *tolerância* a estímulos em redes imunológicas. No trabalho de Hart et al. (2007), além deste modelo baseado em cadeias binárias, também foi adotada uma outra proposta, apresentada inicialmente em Bersini (2002), que utiliza um espaço de formas bidimensional de valores inteiros (este modelo será discutido em maiores detalhes no Capítulo 3). De maneira geral, a dinâmica de concentrações deste segundo modelo é bastante similar à proposta de Bersini (2003), estando a principal modificação na forma de afinidade entre os componentes, que pode ser tanto complementar quanto baseada em similaridade. Este modelo de Bersini (2002) também foi adotado por: Hart (2005), em seu estudo sobre formas alternativas de *regiões de reconhecimento* de anticorpos; Hart & Ross (2005), que estudaram a influência de tais formas alternativas de *regiões de reconhecimento* de anticorpos na emergência de redes imunológicas, ou seja, no comportamento dinâmico e conectividade de tais redes; Hart (2006), no estudo de um modelo de crescimento de redes imunológicas; e por Hart et al. (2006), em seu estudo sobre como a métrica de afinidade empregada influencia a topologia de redes imunológicas.

Brede & Behn (2003) propuseram um novo modelo (também adotado por Schmidtchen & Behn, 2006) que mais uma vez emprega codificação binária dos componentes da rede imunológica (neste caso, células B e anticorpos são tratados como uma única entidade) e se baseia em um grafo-base não direcional com número de vértices pré-definido pelo usuário. Neste modelo, cada vértice do grafo é uma potencial posição que pode ser ocupada por uma célula B e, a cada iteração, vértices vazios são aleatoriamente selecionados para serem ocupados (por células também geradas aleatoriamente) e células presentes no grafo que recebam um nível de estímulo de seus vizinhos inferior a um determinado limiar  $\sigma_l$  ou superior a outro limiar  $\sigma_u$  são eliminadas. Tanto o estímulo recebido por cada célula quanto a existência ou não de arestas conectando vértices do grafo dependem da métrica de afinidade entre células, que adota o conceito de complementariedade de cadeias binárias (baseada

em distância de Hamming) e permite um certo grau de imperfeição desta complementariedade (definido pelo usuário). Em Brede & Behn (2003), os autores verificaram que, de acordo com o fluxo de novas células e os parâmetros  $\sigma_l$  e  $\sigma_u$ , é possível observar diferentes comportamentos da rede e, em alguns casos, até mesmo o surgimento de padrões estáticos e dinâmicos de comportamento. Já Schmidtchen & Behn (2006) realizaram extensas simulações neste modelo e conseguiram inclusive identificar o surgimento de grupos de vértices (células B) com propriedades estatísticas bem definidas.

O último modelo que será discutido aqui é a proposta de Trojanowski & Sasin (2006). Neste trabalho, os autores propuseram um modelo que também adota codificação binária de anticorpos e antígenos (células B e anticorpos são considerados um único componente), mas apresenta duas diferenças fundamentais para os demais modelos: a rede imunológica possui um número fixo de componentes durante toda a simulação, determinado pelo usuário, e o cálculo de afinidade entre anticorpos independe da concentração de cada um deles. Cada anticorpo é modelado com duas cadeias binárias de 32 bits (paratopos e epitopos) e dois atributos, sendo um correspondente à concentração e o outro ao tempo de vida da célula. Já os antígenos são modelados apenas com uma cadeia binária de 32 bits (epitopo) e os dois atributos. Inicialmente, os anticorpos na rede são gerados aleatoriamente e, durante a simulação, sua concentração pode aumentar, caso sejam estimulados, ou diminuir, caso haja supressão. Quando a concentração reduz abaixo de um determinado limiar, o anticorpo é eliminado e substituído por uma nova célula gerada a partir de mutação aplicada a uma das demais células com alta concentração. Neste modelo existem cinco regras de estímulo e/ou supressão, que dependem basicamente de quais são os componentes causadores e receptores do estímulo (antígenos ou outros anticorpos). Por fim, com relação à afinidade entre as cadeias binárias, foram adotadas oito métricas distintas por Trojanowski & Sasin (2006), que levaram a comportamentos diferentes da rede.

Como é possível observar, a grande maioria dos modelos de rede imunológica trabalha com o conceito de *concentração* de componentes e esta concentração está profundamente relacionada ao comportamento dinâmico de tais modelos, influenciando na maioria dos casos os processos de estímulo e supressão de células e/ou anticorpos. Além disso, com exceção do modelo de Bersini (2002), que utiliza uma codificação em valores inteiros, a maioria dos modelos da literatura utiliza a codificação dos componentes do sistema em cadeias binárias, o que facilita enormemente a implementação do conceito de *complementariedade*, largamente utilizado nas métricas de afinidade. No entanto, este modelo de Bersini (2002) tem sido um dos mais utilizados na literatura nos últimos anos, talvez justamente por ser uma das poucas alternativas a não adotar a representação binária de anticorpos e antígenos.

Considerando a utilização de modelos imunológicos extremamente simplificados nos algoritmos imunoinspirados para otimização e a vasta literatura existente sobre modelos alternativos e mais elaborados para as redes imunológicas, o objetivo principal desta tese é buscar aproximar estas duas áreas

e tentar incorporar, nos algoritmos imunoinspirados, alguns aspectos destes outros modelos que possam trazer benefícios, principalmente com relação à sua capacidade de manutenção de diversidade. Este trabalho de incorporação de ideias acabou resultando nos dois novos algoritmos apresentados nesta tese, que serão descritos nos próximos capítulos.

## 2.3 Considerações Finais

Neste capítulo, foram tratados aspectos gerais do sistema imunológico natural (SIN) e as duas principais teorias (*Seleção Clonal* e *Rede Imunológica*) que visam explicar o seu funcionamento e são amplamente utilizadas nas ferramentas propostas em sistemas imunológicos artificiais (SIAs). Em seguida, os SIAs propriamente ditos foram introduzidos, com a identificação das características do SIN que são relevantes computacionalmente e uma breve revisão bibliográfica das principais aplicações existentes na literatura desta área. Seguiu-se então uma discussão sobre dois conceitos fundamentais em SIAs, que são os *espaços de formas* e a questão da *afinidade*, que pode ocorrer tanto entre anticorpos e antígenos quanto entre dois anticorpos distintos.

Com o intuito de ilustrar como geralmente se dá a reprodução dos conceitos imunológicos em algoritmos imunoinspirados, foi discutido em detalhes o algoritmo opt-aiNet (*Artificial Immune Network for Continuous Optimization*), que baseia-se tanto no princípio da Seleção Clonal quanto na Teoria da Rede Imunológica e foi proposto para otimização em espaços contínuos (que é o foco deste trabalho). A opt-aiNet foi escolhida neste capítulo justamente por se tratar de um dos SIAs inspirados na teoria da Rede Imunológica mais utilizados da literatura, e também por permitir ao leitor observar o quão simplificado é o modelo de rede imunológica adotado não só neste algoritmo, mas na maioria das demais propostas da literatura voltadas para otimização.

Por fim, como esta questão da utilização de um modelo de rede imunológica simplificado nos algoritmos imunoinspirados é um dos aspectos que serão atacados neste trabalho, foram discutidos alguns exemplos de outros modelos mais elaborados de redes imunológicas também presentes na literatura de SIAs, para que fosse possível identificar quais são os aspectos gerais mais relevantes em tais modelos, que servirão de inspiração para a proposta de uma nova geração de algoritmos imunoinspirados voltados para otimização em espaços contínuos.



## Capítulo 3

### O Algoritmo *cob-aiNet*

Como visto no capítulo anterior, os sistemas imunológicos artificiais (SIAs) baseados na teoria da Rede Imunológica<sup>4</sup> e voltados para otimização geralmente se utilizam de modelos de rede muito simples, restritos apenas a eventuais comparações entre os indivíduos da população (durante a execução) e eliminação daqueles de pior qualidade, quando posicionados entre si a uma distância menor que um determinado limiar. Além disso, foi visto também que existem diversos modelos mais elaborados de redes imunológicas na literatura, modelos estes que, salvo algumas aplicações específicas em robótica (Bakhouya et al., 2003; Ishiguro et al., 1997; Whitbrook et al., 2008; Yu & Mao, 2010) e reconhecimento de padrões (Martins & Slani, 2005; Shomooka et al., 2002), têm sido geralmente empregados apenas em simulações computacionais que buscam melhor compreender o funcionamento do sistema imunológico natural (SIN) em si.

Mesmo utilizando modelos simplificados, as redes imunológicas artificiais voltadas para otimização reconhecidamente possuem bons mecanismos de manutenção de diversidade (de França et al., 2010a,b), o que as permite localizar e manter, em sua população, o maior conjunto possível de ótimos locais, ao mesmo tempo em que é feita a busca pelo ótimo global do problema. Diante disso, acredita-se que, com a adoção de um modelo mais elaborado de rede imunológica nestes algoritmos, seja possível aumentar ainda mais sua capacidade de exploração e localização de ótimos locais, sem que haja nenhum impacto negativo na qualidade final das soluções obtidas. Este estudo foi feito neste trabalho e, como resultado, foi proposto o algoritmo *cob-aiNet* (*Concentration-based Artificial Immune Network* – Coelho & Von Zuben, 2010), que é uma das principais contribuições desta tese.

O algoritmo *cob-aiNet*, proposto originalmente para otimização mono-objetivo de problemas em espaços contínuos, será descrito em detalhes neste capítulo, enquanto que sua extensão para problemas de otimização multiobjetivo (também em espaços contínuos) será formalmente apresentada e

---

<sup>4</sup>Ao longo do texto, será adotado o termo *rede imunológica artificial* para designar SIAs baseados na teoria da Rede Imunológica de Jerne (1974).

discutida no Capítulo 5. Apesar desta tese ter sido desenvolvida com o foco todo voltado para o tratamento de problemas de otimização em espaços contínuos, a extensão das ferramentas propostas aqui para outros tipos de problemas de otimização (por exemplo, otimização combinatória) é viável e pode adotar os mesmos passos concebidos para a família aiNet de algoritmos (de França et al., 2010a).

Este capítulo está dividido em quatro seções principais. Na Seção 3.1, será feito o posicionamento do algoritmo cob-aiNet, tanto frente a outras meta-heurísticas populacionais capazes de manter a diversidade dos indivíduos em suas populações quanto frente às ferramentas clássicas de otimização local e global. Na Seção 3.2, serão apresentados em detalhes o modelo de rede imunológica em que a cob-aiNet se baseou, juntamente com as modificações que se mostraram necessárias para adequá-lo ao contexto de problemas de otimização. O algoritmo cob-aiNet propriamente dito será descrito em detalhes na Seção 3.3, enquanto que algumas noções gerais da complexidade computacional apresentada por esta ferramenta serão discutidas na Seção 3.4. Por fim, os comentários finais serão apresentados na Seção 3.5.

## 3.1 Posicionamento da Proposta

Como mencionado anteriormente, a cob-aiNet é uma meta-heurística populacional, voltada para otimização de problemas mono-objetivo em espaços contínuos e que tem como principal característica a capacidade de manutenção de diversidade dos indivíduos em sua população. Esta particularidade faz com que a cob-aiNet seja indicada principalmente para o tratamento de problemas multimodais e para situações em que se deseja obter múltiplas soluções distintas em uma única execução do algoritmo.

Diante disso, nesta seção, a cob-aiNet será posicionada frente a outras meta-heurísticas populacionais da literatura, principalmente em relação aos seus mecanismos de manutenção de diversidade. Além disso, também serão tratados brevemente nesta seção os aspectos gerais das chamadas *ferramentas clássicas* de otimização, com o intuito de identificar qual é o comportamento destas técnicas em problemas multimodais, e como a cob-aiNet se posicionaria dentro desta área, estudada principalmente em Pesquisa Operacional.

### 3.1.1 Posicionamento Frente a Outras Meta-heurísticas

Dentro do paradigma de SIAs existem poucas propostas de ferramentas que, como a cob-aiNet, são voltadas especificamente para o problema de otimização de funções. O trabalho pioneiro nesta linha foi desenvolvido por de Castro & Von Zuben (2002b) e culminou no desenvolvimento do algoritmo CLONALG (*Clonal Selection Algorithm*), baseado no princípio da Seleção Clonal de Burnet

(1978). Apesar de o CLONALG ter obtido resultados experimentais que mostraram um bom desempenho em problemas de otimização multimodais, o que indica a boa capacidade da ferramenta em manter diversidade, este algoritmo ainda não possuía mecanismos que permitissem o ajuste dinâmico do número de indivíduos em sua população, característica esta que só viria a ser introduzida com os mecanismos inspirados na teoria da Rede Imunológica (Jerne, 1974) presentes no algoritmo aiNet (*Artificial Immune Network* – de Castro & Von Zuben, 2002a).

A aiNet foi proposta originalmente para tratar problemas de agrupamento de dados mas, pouco tempo depois da publicação de seu trabalho definitivo, acabou dando origem a uma extensão voltada para otimização, denominada opt-aiNet (*Artificial Immune Network for Continuous Optimization* – de Castro & Timmis, 2002a). Desde então, surgiram muitas outras variações de algoritmos dentro desta família ao longo dos anos, voltadas aos mais diversos tipos de problemas e aplicações (de França et al., 2010a). Dentre aquelas desenvolvidas especificamente para otimização, é possível destacar:

- **copt-aiNet** (*Artificial Immune Network for Combinatorial Optimization* – de Sousa et al., 2004; Gomes et al., 2004), desenvolvida para tratar problemas de otimização combinatória;
- **dopt-aiNet** (*Artificial Immune Network for Dynamic Optimization* – de França et al., 2005, 2006b; Junqueira et al., 2005, 2006), proposta para problemas de otimização dinâmica;
- **omni-aiNet** (*Artificial Immune Network for Omni Optimization* – Coelho & Von Zuben, 2006b) e **VIS** (*Vector Immune System* – Freschi & Repetto, 2006), criadas para problemas de otimização multiobjetivo.

Como a cob-aiNet, que será apresentada neste capítulo, todas as ferramentas da família aiNet apresentadas acima são baseadas tanto no princípio da Seleção Clonal quanto na teoria da Rede Imunológica, e possuem mecanismos capazes de estimular a diversidade dos indivíduos da população e também de permitir o ajuste dinâmico do número de indivíduos nela presentes.

Além das propostas para otimização baseadas no CLONALG e na aiNet, existe também na literatura de SIAs uma outra família de algoritmos imunoinspirados voltados para otimização, com ferramentas que substituíram os operadores de clonagem e mutação por modelos probabilísticos. O algoritmo que deu origem a esta família, denominado BAIS (*Bayesian Artificial Immune System* – Castro & Von Zuben, 2009), foi proposto para tratar problemas de otimização combinatória. Para capturar adequadamente as interações mais relevantes entre as variáveis do problema, o BAIS adota um modelo probabilístico baseado em redes bayesianas, o que o torna capaz de lidar com os chamados *blocos construtivos*<sup>5</sup>. Até o momento, o algoritmo BAIS possui três extensões: MOBAIS,

---

<sup>5</sup>Diversos problemas de otimização podem ser decompostos em sub-problemas passíveis de serem resolvidos individualmente. Após a resolução de cada um destes sub-problemas, as soluções parciais devem ser então combinadas, para

que é equivalente ao BAIS mas voltado para problemas de otimização multiobjetivo (Castro & Von Zuben, 2008); GAIS (*Gaussian Artificial Immune System for Continuous Optimization* – Castro & Von Zuben, 2010a), que é uma extensão do BAIS desenvolvida para tratar problemas de otimização em espaços contínuos e que utiliza redes gaussianas no lugar das redes bayesianas; e MOGAIS, que é uma extensão do GAIS para problemas de otimização multiobjetivo (Castro & Von Zuben, 2010b). Apesar dos algoritmos da família BAIS/GAIS não adotarem operadores de mutação e clonagem, eles empregam mecanismos de supressão e inserção de novos indivíduos bem semelhantes aos presentes nos algoritmos da família aiNet, o que indica certa inspiração na teoria da Rede Imunológica de Jerne (1974).

Diferentemente do que ocorre em SIAs, os mecanismos de manutenção de diversidade presentes nos demais paradigmas de meta-heurísticas populacionais (Bäck et al., 2000; Dorigo & Stützle, 2004; Kennedy et al., 2001; Price et al., 2005) geralmente correspondem a *adendos* inseridos posteriormente, uma vez que as propostas e inspirações originais não foram capazes de prover tal comportamento.

No caso específico de algoritmos evolutivos (Bäck et al., 2000), a manutenção de diversidade geralmente é feita através de *métodos de niching* tais como *fitness sharing* (Sareni & Krähenbühl, 1998). Em *fitness sharing*, que é aplicado principalmente em algoritmos genéticos, o *fitness* de cada indivíduo é determinado não só pela função-objetivo do problema, mas também pelos demais indivíduos da população que possuem *fitness* parecido (ou seja, que estão no mesmo *nicho*). Como o intuito é manter a diversidade, o *fitness* de tais indivíduos semelhantes é então *penalizado*, numa proporção inversa da sua distância em relação aos demais valores mais próximos. Apesar de a técnica de *fitness sharing* ter sido inicialmente concebida para manter a *diversidade de fitness* na população, geralmente ela é utilizada para manter a *diversidade de soluções*, bastando para isso substituir a distância entre valores de *fitness* por alguma medida de *distância entre soluções*, avaliada no espaço de variáveis do problema (espaço de busca).

O principal problema da técnica de *fitness sharing* está no fato de que, caso os melhores indivíduos estejam coincidentemente nas regiões mais povoadas, eles tenderão a sofrer uma forte penalização, podendo até mesmo ser eliminados da população na etapa de seleção. Em outras palavras, caso a penalização sofrida por estes melhores indivíduos seja muito alta, indivíduos piores (em relação ao valor da função-objetivo) mas posicionados em regiões isoladas do espaço de busca podem acabar obtendo um *fitness* melhor, o que os manterá na população em detrimento das melhores soluções. Para solucionar este problema foi proposta a técnica de *clearing* (Petrowski, 1996), que é semelhante

---

que se possa obter a solução global do problema original. Neste contexto, é possível que o *vetor de atributos* associado a cada solução candidata em um algoritmo populacional possa vir a apresentar *trechos* que correspondem a estas soluções parciais. Caso isto ocorra, cada um destes *trechos* é denominado *bloco construtivo* (do inglês, *building block* – Castro, 2009).

ao *fitness sharing* na maioria dos aspectos, exceto no fato de que o *fitness* original dos  $k$  melhores indivíduos sempre é preservado, evitando-se assim a perda do potencial evolutivo. É possível observar uma certa semelhança entre a técnica de *clearing* e os mecanismos de manutenção de diversidade presentes nos algoritmos imuno-inspirados baseados na Teoria da Rede (vide Seção 2.2.2), já que ambos avaliam a vizinhança de cada célula e buscam manter os melhores indivíduos. No entanto, nas ferramentas imuno-inspiradas, as comparações são feitas par a par entre todas as soluções, o que favorece todos os melhores indivíduos dentro de seus raios de vizinhança, enquanto que na técnica de *clearing* as comparações são globais, ou seja, apenas os  $k$  melhores indivíduos da população deixarão de ser penalizados, mesmo que eles sejam muito semelhantes entre si.

Outras duas técnicas que cabem ser mencionadas aqui, também relacionadas aos algoritmos evolutivos, são as chamadas *seleção termodinâmica* (Mori et al., 1996, 1998) e *substituição por torneio restrito* (*restricted tournament replacement* – RTR – Harik, 1995), que buscam manter a diversidade da população na etapa de *seleção* do algoritmo, ao invés de alterar o *fitness* individual de cada solução candidata. A abordagem de seleção termodinâmica busca minimizar, durante a seleção dos indivíduos que serão mantidos na população na próxima iteração, uma métrica denominada *energia livre*, que está diretamente relacionada à frequência com que um indivíduo (ou parte dele) se repete na população. Já na técnica RTR, para cada novo indivíduo  $i$  gerado (*offspring*), um subconjunto  $W$  de soluções é definido, formado por  $w$  indivíduos aleatoriamente escolhidos dentre aqueles já presentes na população. Em seguida, a distância entre cada indivíduo de  $W$  e  $i$  é calculada, e aquela solução  $j$  de  $W$  mais semelhante a  $i$  (menor distância) é então identificada. Por fim,  $i$  e  $j$  são comparados e, caso  $i$  seja melhor que  $j$ ,  $i$  substitui o indivíduo  $j$  na população. Caso contrário,  $i$  é descartado.

Já nas ferramentas de inteligência de enxame, tais como os algoritmos de PSO (do inglês *particle swarm optimization* – Kennedy et al., 2001), originalmente buscou-se reduzir a velocidade de convergência das ferramentas e ampliar sua capacidade de busca através da adoção de diferentes estratégias de *vizinhança* entre as partículas (Kennedy & Mendes, 2002). No entanto, atualmente a abordagem mais comum para manutenção de diversidade nestes algoritmos é através da hibridização destas ferramentas com operadores de mutação, como foi feito em (Esquivel & Coello Coello, 2003). Além disso, existem também na literatura algumas propostas que sugerem modificações na estrutura básica dos algoritmos. Neste último caso, podemos citar o trabalho de Liang et al. (2006), em que foi proposto um algoritmo de PSO que apresenta uma nova forma de atualização da *velocidade* de cada partícula, modificada para levar em conta a informação do melhor histórico de cada uma das  $N$  partículas da população, ao invés de apenas o melhor histórico da própria partícula e de sua vizinhança. Isto faz com que, na proposta de Liang et al. (2006), sejam utilizados  $N$  valores históricos na atualização de cada partícula do enxame.

As abordagens híbridas também são geralmente adotadas em algoritmos de evolução diferen-

cial (do inglês *differential evolution* – Price et al., 2005), como por exemplo na proposta de Coelho et al. (2009c), em que há um monitoramento constante da diversidade das soluções na população e, conforme a contribuição individual de cada indivíduo para a diversidade total, são aplicados diferentes operadores de mutação.

Apesar de todas estas técnicas de manutenção de diversidade para meta-heurísticas populacionais estimularem a diversidade dos indivíduos já presentes na população, elas não são capazes de dar aos algoritmos em que são empregadas a capacidade de ajustar dinamicamente o tamanho de suas populações conforme a demanda do problema, característica esta que está presente nativamente em muitos dos SIAs baseados na teoria da rede, inclusive no algoritmo *cob-aiNet* a ser proposto neste trabalho.

### 3.1.2 Posicionamento Frente a Outras Ferramentas de Otimização

Diferentemente das meta-heurísticas, que abrem mão da garantia de convergência para o ótimo do problema em favor de uma maior flexibilidade<sup>6</sup>, as *ferramentas clássicas* (ou *matemáticas*) de otimização geralmente tendem a ser mais “rígidas”, uma vez que a maioria delas é desenvolvida para tratar classes específicas de problemas que atendam a uma série de restrições, tais como diferenciabilidade, separabilidade e outras. No entanto, estas ferramentas possuem uma grande vantagem frente às meta-heurísticas, já que estão sempre acompanhadas de demonstrações formais que indicam claramente quais são suas garantias de convergência, caso todas as condições impostas sejam atendidas.

As ferramentas clássicas de otimização podem ser divididas em dois grupos: (i) as *ferramentas de otimização local* (Nocedal & Wright, 2006), que são aqueles algoritmos que garantem apenas a convergência da busca para um ótimo local do problema; e (ii) as *ferramentas de otimização global* (Horst & Pardalos, 1995) que, como o próprio nome diz, garantem a localização das melhores soluções possíveis para o problema, ou seja, dos ótimos globais.

De maneira geral, os algoritmos de otimização local consistem basicamente em uma sequência pré-definida de buscas unidirecionais, que são aplicadas iterativamente a partir de um ponto inicial e seguem uma direção de busca definida deterministicamente, a partir de informações locais. A cada iteração, a melhor solução encontrada na busca unidirecional se torna o ponto de partida da próxima busca unidirecional, a ser executada na iteração seguinte. Apesar de apresentarem uma estrutura aparentemente semelhante, a forma com que cada algoritmo de otimização local define a direção de busca tende a ser diferente, o que faz com que cada um destes algoritmos seja voltado a classes bem específicas de problemas.

---

<sup>6</sup>Por flexibilidade, neste contexto, entende-se como a possibilidade de aplicação das meta-heurísticas em quaisquer problemas de otimização, sem que haja a exigência de que tais problemas apresentem propriedades específicas tais como convexidade, diferenciabilidade, etc.

Já no contexto de otimização global, principalmente na classe dos chamados *algoritmos completos*, que são aqueles que garantem a obtenção do ótimo global dentro de uma margem de tolerância previamente definida, supondo operações computacionais exatas e limite de tempo indefinido, a grande maioria dos algoritmos se baseia na abordagem de *branching* (Neumaier, 2004), que consiste basicamente em dividir recursivamente o problema original em sub-problemas que, cedo ou tarde, serão mais fáceis de serem resolvidos. Em métodos de *branching* puros, as regiões mais promissoras tendem a ser divididas com maior frequência, enquanto que nas técnicas de *branch and bound* são definidos limites para o valor da função-objetivo, que permitem identificar quais sub-problemas não possuem soluções melhores que a melhor solução já encontrada, levando assim ao seu descarte (Neumaier, 2004).

Um aspecto fundamental tanto para os algoritmos de *branching* quanto para as técnicas de *branch and bound* é a estratégia adotada para redução do espaço de busca, ou seja, como devem ser feitas as divisões sucessivas de cada problema em sub-problemas menores. Para isso, geralmente são utilizadas abordagens baseadas em aritmética de intervalos (Moore & Bierbaum, 1979; Neumaier, 2004) ou em propagação de restrições (Dallwig et al., 1997; Neumaier, 2004), sendo que esta última explora a presença de restrições separáveis, ou seja, de restrições que dependem de uma única variável.

Definidas as estratégias de divisão do problema original em sub-problemas menores, é preciso então resolver cada um destes sub-problemas, mesmo que de maneira rudimentar, para que se possa descartar aqueles cujas soluções estão sabidamente distantes do ótimo global do problema e focar a atenção apenas naqueles mais promissores, repetindo-se o processo iterativo. A solução de cada sub-problema geralmente é obtida com a utilização de algum método de otimização local, sendo que, para aqueles problemas que possuem uma grande quantidade de ótimos locais, também deve ser empregada alguma abordagem heurística, de modo a evitar que o algoritmo em questão fique preso em tais ótimos locais, comprometendo assim o processo global de otimização (Neumaier, 2004).

Uma outra abordagem, utilizada principalmente nas técnicas de *branch and bound*, consiste em estimar os limites superiores (no caso de maximização) ou inferiores (no caso de minimização) do valor da função-objetivo para cada sub-problema, o que geralmente é feito com as chamadas técnicas de *relaxações lineares e convexas*. Estas técnicas modificam a formulação do problema para obtenção de versões convexas, permitindo assim a utilização de alguma ferramenta de otimização local para identificar tais limites inferiores ou superiores, sem que haja a preocupação com a presença de ótimos locais. No entanto, a principal desvantagem destas técnicas de relaxação está no fato delas geralmente exigirem manipulações matemáticas da formulação das funções-objetivo dos problemas, o que nem sempre é possível ou viável.

Estas limitações das técnicas de *branching* e *branch and bound*, principalmente quanto a questão da multimodalidade, acabou levando à sua não inclusão nos experimentos que serão discutidos no

Capítulo 4, já que a grande maioria dos problemas tratados apresenta grande quantidade de ótimos locais, além de outras características, tais como presença de ruído e não-diferenciabilidade em todos os pontos do espaço de busca, que dificultariam a escolha de métodos de otimização local para serem utilizados em conjunto com as técnicas de otimização global. Além disso, o foco deste trabalho é o desenvolvimento de ferramentas capazes de retornar o maior número possível de ótimos locais do problema em uma única execução (além dos ótimos globais, se possível), o que difere fundamentalmente do objetivo das técnicas de otimização locais e globais.

## 3.2 O Modelo de Rede Imunológica

Dentre todos os modelos de rede imunológica apresentados na Seção 2.2.3, o modelo de Bersini (2002) é um dos que mais se aproxima das condições encontradas em problemas de otimização em espaços contínuos, o que o fez ser selecionado como ponto de partida deste trabalho. No entanto, apesar do modelo de Bersini (2002) ser o mais próximo do contexto que será tratado aqui, uma série de modificações e adaptações se mostraram necessárias e tiveram que ser implementadas. Sendo assim, nesta seção, tanto o modelo original de Bersini (2002) quanto a nova versão, com todas as modificações propostas, serão discutidos em detalhes.

### 3.2.1 O Modelo de Bersini

O modelo de Bersini (2002) (também adotado por Hart, 2005; Hart & Ross, 2005; Hart, 2006; Hart et al., 2006, 2007) trabalha com um espaço de formas bidimensional de valores inteiros, de tamanho  $X \times Y$ , em que cada anticorpo e cada antígeno podem ocupar qualquer posição  $(x, y)$ . Neste modelo, a rede imunológica pode ser vista como um *grafo*, em que cada célula e antígeno correspondem a um nó e existem arestas que conectam apenas aqueles nós que se estimulam mutuamente.

A forma com que uma dada célula ou antígeno estimula outra célula no modelo de Bersini (2002) depende de como a *afinidade* entre células e entre células e antígenos é definida. Caso seja adotado um modelo de afinidade *complementar*, então duas células  $A$  e  $B$  terão afinidade diferente de zero caso  $B$  esteja dentro de uma região circular de raio  $r$  (raio de afinidade) centralizada no ponto  $(X - x, Y - y)$ , como ilustrado na Figura 3.1(a). Caso o modelo adotado seja um modelo de afinidade por *similaridade*, então  $A$  e  $B$  terão afinidade não-nula se  $B$  estiver dentro da região de raio  $r$  centralizada na própria célula  $A$ , como exemplificado na Figura 3.1(b). Além disso, diferentemente de outros modelos da literatura que adotam uma topologia de vizinhança toroidal no espaço de formas (Detours et al., 1994), o modelo de Bersini (2002) ignora as regiões de reconhecimento que estão fora do plano  $X \times Y$ , pertencentes a células posicionadas nas *extremidades* deste espaço (como ilustrado para a

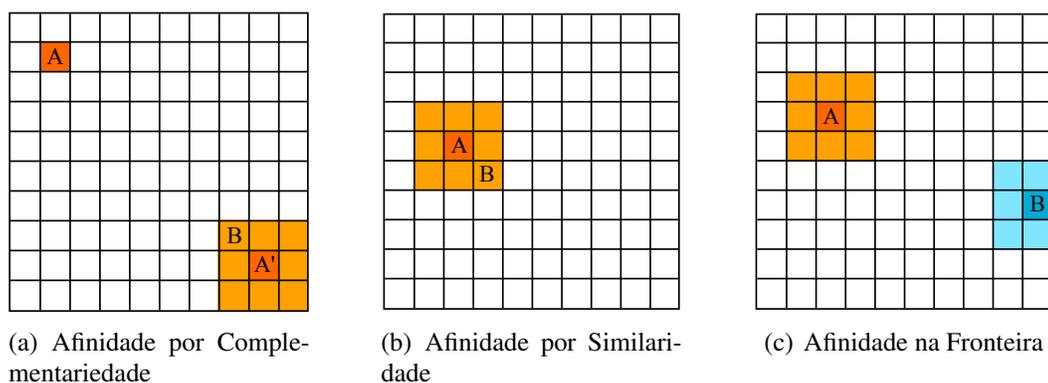


Fig. 3.1: Exemplos de afinidade em um plano bidimensional  $10 \times 10$ , de acordo com o modelo de Bersini (2002). Nestes exemplos, o raio de afinidade foi definido como sendo  $r = 1$ . Em (a) temos um exemplo de afinidade por *complementariedade*, em que  $B$  está dentro da região de afinidade de  $A$ , representado em laranja ( $A'$  é o ponto complementar de  $A$ ). Em (b) temos um exemplo de afinidade entre  $A$  e  $B$  por *similaridade* (região de afinidade de  $A$  representada em laranja). Por fim, em (c) temos um exemplo de como é a região de afinidade de uma célula posicionada na fronteira ( $B$ ), representada em azul. Em (c) foi considerado o modelo de afinidade por similaridade.

célula  $B$ , na Figura 3.1(c)).

Estabelecidas estas definições sobre o espaço de formas, a dinâmica da rede idiotípica para este modelo pode ser resumida no Pseudo-código 3.1, onde  $S_{Ab}^i$  e  $S_{Ag}^j$  são respectivamente os estímulos totais recebidos pela célula (anticorpo)  $i$  e pelo antígeno  $j$ ,  $L$  e  $U$  são limites inferiores e superiores de estímulo, que determinam quando ocorrerá aumento ou diminuição de concentração de cada célula e/ou anticorpo,  $A_c^j$  é a concentração do antígeno  $j$ ,  $f$  é um fator de escalonamento (definido pelo usuário) e  $p_{Ag}$  é a probabilidade de que um novo antígeno seja inserido na rede.

A afinidade entre duas células  $i$  e  $j$  ou entre uma célula  $i$  e um antígeno  $j$  se baseia na distância euclidiana entre os dois componentes, e é dada pela Equação 3.1:

$$afinidade(i, j) = \begin{cases} C_i \cdot [r - D(i, j')] & \text{se } D(i, j') \leq r \\ 0 & \text{caso contrário} \end{cases}, \quad (3.1)$$

onde  $C_i$  é a concentração da célula  $i$ ,  $r$  é o raio de afinidade e  $D(i, j')$  é a distância euclidiana entre os indivíduos  $i$  e  $j'$ . Caso tenha sido adotado o modelo de afinidade complementar,  $j'$  é ponto complementar a  $j$ , como ilustrado para o indivíduo  $A$  na Figura 3.1(a). Caso contrário,  $j' = j$ . Neste modelo, só há afinidade entre dois indivíduos caso eles estejam um dentro do raio de afinidade do outro, e a afinidade será máxima quando a distância  $D(i, j')$  for nula.

O estímulo total recebido por cada célula (anticorpo) do modelo, dado pela Equação 3.2, nada mais é que a soma dos estímulos recebidos tanto das demais células da rede imunológica quanto dos antígenos que estão dentro do seu raio de afinidade.

---

**Pseudo-código 3.1** Dinâmica da rede idiotípica de acordo com o modelo de Bersini (2002).

---

**enquanto** (critério de parada da simulação não estiver satisfeito) **faça**

1- Gerar aleatoriamente um novo anticorpo com coordenadas  $(x, y)$  e concentração 10;

2- Adicionar, com alguma probabilidade  $p_{Ag}$ , um novo antígeno aleatoriamente gerado com coordenadas  $(x_a, y_a)$  e concentração 1000;

**para todo** (anticorpo  $i$  da população) **faça**

3- Calcular o estímulo total  $S_{Ab}^i$  recebido pelo anticorpo  $i$ ;

**se**  $(L < S_{Ab}^i < U)$  **então**

4- Aumentar a concentração do anticorpo  $i$  em 1;

**senão**

5- Diminuir a concentração do anticorpo  $i$  em 1;

**fim se**

**fim para**

**para todo** (antígeno  $j$  da população) **faça**

6- Calcular o estímulo total  $S_{Ag}^j$  recebido pelo antígeno  $j$ ;

**se**  $(L < S_{Ag}^j)$  **então**

7- Diminuir a concentração do antígeno  $j$ :  $A_c^j = A_c^j - \frac{S_{Ag}^j}{L \cdot f}$ ;

**fim se**

**fim para**

8- Remover todos os anticorpos e antígenos com concentração 0;

**fim enquanto**

---

$$S_{Ab}^i = \sum_{\forall j \in J} \text{afinidade}(j, i) + \sum_{\forall k \in K} \text{afinidade}(k, i) \quad (3.2)$$

onde  $S_{Ab}^i$  é o estímulo total recebido pela célula  $i$ ,  $J$  é o conjunto de todos os anticorpos presentes na rede imunológica e  $K$  o conjunto de todos os antígenos. A afinidade entre células e entre células e antígenos é dada conforme a Equação 3.1.

Já o estímulo total recebido pelos antígenos presentes na rede imunológica é constituído apenas pelo estímulo recebido dos anticorpos, como dado na Equação 3.3, uma vez que não há interações entre os antígenos presentes na rede.

$$S_{Ag}^j = \sum_{\forall i \in I} \text{afinidade}(i, j) \quad (3.3)$$

onde  $S_{Ag}^j$  é o estímulo total recebido pelo antígeno  $j$  e  $I$  é o conjunto de todos os anticorpos presentes na rede imunológica. A afinidade entre o antígeno  $j$  e as demais células é dada pela Equação 3.1.

É importante ressaltar aqui que o modelo de Bersini (2002) adota uma versão simplificada do estímulo total em forma de sino, proposto inicialmente por De Boer (1989). No modelo de Bersini (2002), existem dois limiares, definidos pelo usuário (representados por  $L$  e  $U$  no Pseudo-código 3.1), que anulam o estímulo recebido por cada célula caso ele seja menor que o limiar inferior ou maior que

o limiar superior. Desta forma, tanto as células que recebem pouco estímulo quanto as células que recebem muito estímulo têm sua concentração diminuída, ou seja, sofrem um processo de supressão. Já no caso dos antígenos, como eles só recebem estímulos de células do sistema imunológico, só é definido um limiar inferior de estímulo, acima do qual sua concentração é obrigatoriamente reduzida. Neste caso, quando o estímulo total recebido pelo antígeno é superior ao limiar  $L$ , supõe-se que as células do sistema imunológico foram suficientemente ativadas, o que levou ao disparo da resposta imunológica.

Apesar de o modelo de Bersini (2002) conseguir reproduzir a contento o comportamento básico de ativação e supressão de células e antígenos em uma rede imunológica, muitos dos mecanismos propostos são específicos tanto para o contexto do espaço de formas bidimensional utilizado quanto para a presença explícita de antígenos na rede, o que não ocorre nos casos de problemas de otimização em espaços contínuos, que são o foco deste trabalho. Sendo assim, uma série de modificações se mostrou necessária para que o modelo de Bersini (2002) pudesse ser utilizado para este fim, modificações estas que serão discutidas em detalhes na próxima seção.

### 3.2.2 Adaptações ao Modelo de Bersini para Otimização

A primeira modificação sofrida pelo modelo de Bersini (2002) em sua adaptação para problemas de otimização está relacionada ao espaço de formas. Como o foco do algoritmo cob-aiNet é o tratamento de problemas de otimização em espaços contínuos, o espaço de formas mais óbvio a ser empregado é justamente o espaço contínuo multidimensional, em que cada célula do algoritmo corresponderá a um ponto em  $\mathbb{R}^n$  (onde  $n$  é a dimensão do espaço de variáveis do problema), ou seja, a um vetor de atributos reais que corresponde a uma *solução candidata* para o problema.

Além disso, como foi observado na Seção 2.2.2, no contexto de otimização de funções não existe uma representação explícita dos antígenos a serem combatidos, o que faz com que a métrica de afinidade entre anticorpos e antígenos seja definida como uma métrica direta de qualidade (ou *fitness* – vide Seção 2.2.1).

Sendo assim, supondo que o problema de otimização a ser tratado tenha função-objetivo  $f$ , o *fitness* de cada célula na população do algoritmo, ou seja, a afinidade entre cada célula com os antígenos, será dado conforme a Equação 3.4.

$$f_i^{Ag}(t) = \begin{cases} \frac{f_i(t) - \min_j[f_j(t)]}{\max_j[f_j(t)] - \min_j[f_j(t)]} & \text{se for um problema de maximização} \\ 1 - \frac{f_i(t) - \min_j[f_j(t)]}{\max_j[f_j(t)] - \min_j[f_j(t)]} & \text{se for um problema de minimização} \end{cases}, \quad (3.4)$$

onde  $f_i^{Ag}(t) \in [0, 1]$  é o *fitness* da célula  $i$  na iteração  $t$  e  $f_i(t)$  é o valor da função-objetivo para a célula  $i$  na iteração  $t$ . Em outras palavras, o *fitness*  $f_i^{Ag}(t)$  de cada célula  $i$  na iteração  $t$  do algoritmo nada mais é que o valor da função-objetivo do problema normalizado em  $[0, 1]$  e ajustado para que sempre seja maximizado, ou seja, para que as melhores células possuam os maiores valores de  $f_i^{Ag}(t)$ .

Além da afinidade entre células e antígenos, outro aspecto que precisou ser modificado nesta nova versão do modelo de rede imunológica foi a métrica de afinidade entre células da população, que agora é dada pela Equação 3.5.

$$f_i^{Ab}(t) = \begin{cases} \frac{\sum_{j \in J} C_t^j \cdot [\sigma_s - d(i, j)]}{\sum_{j \in J} C_t^j} & \text{se } J \neq \emptyset \\ 0 & \text{caso contrário} \end{cases}, \quad (3.5)$$

onde  $f_i^{Ab}(t)$  é a afinidade total entre a célula  $i$  e as demais células da população na iteração  $t$ ,  $J$  é o conjunto de anticorpos *melhores* que  $i$  e que estão *mais próximos* de  $i$  que um limiar  $\sigma_s$  (raio de afinidade ou raio de supressão),  $C_t^j$  é a concentração da célula  $j$  na iteração  $t$  e  $d(i, j)$  é o valor de uma métrica de distância entre as células  $i$  e  $j$ . Qualquer métrica de distância entre dois vetores de atributos reais pode ser utilizada em  $d(k, l)$  mas, em toda esta tese, foi adotada a distância euclidiana.

Apesar da Equação 3.5 ser aparentemente semelhante à Equação 3.1, elas têm uma diferença crucial. Enquanto, no modelo original de Bersini (2002), a afinidade dada pela Equação 3.1 é calculada para todas as células na população, na Equação 3.5 isto é feito apenas para as células que estão no conjunto  $J$ , constituído pelas células que estão mais próximas de  $i$  que o limiar de afinidade  $\sigma_s$  e, ao mesmo tempo, são *melhores* que  $i$ . Esta modificação se mostrou necessária por causa do novo mecanismo de controle das *concentrações* dos anticorpos, que será descrito a seguir.

No algoritmo cob-aiNet, a dinâmica dos indivíduos da população é controlada principalmente pela *concentração* de cada célula, que está diretamente relacionada ao *fitness* de cada indivíduo e à sua presença ou não em regiões do espaço de busca densamente povoadas por melhores soluções. Além de influenciar diretamente a métrica de afinidade entre anticorpos (vide Equação 3.5) o valor de concentração de cada célula também é responsável pelos comportamentos de *ativação* (número de clones que serão gerados) e *supressão* de cada célula, como será visto na Seção 3.3.

A concentração  $C_t^i$  de cada anticorpo  $i$  é definida inicialmente, pelo usuário, com um valor  $C_0$ , e é atualizada ao final de cada iteração do algoritmo conforme a Equação 3.6.

$$C_{t+1}^i = \begin{cases} \min[(\alpha C_t^i - f_i^{Ab}(t)), 1] & \text{se } \alpha C_t^i - f_i^{Ab}(t) \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (3.6)$$

onde  $C_{t+1}^i$  e  $C_t^i \in [0, 1]$  são respectivamente as concentrações nova e antiga da célula  $i$ ,  $f_i^{Ab}(t)$  é a afinidade do anticorpo  $i$  com as demais células da população (dada pela Equação 3.5) e  $\alpha$  é dado pela

Equação 3.7.

$$\alpha = \begin{cases} 1 + 0,1 \cdot f_i^{Ag}(t) & \text{se } f_i^{Ab}(t) = 0 \\ 0,7 & \text{caso contrário} \end{cases}, \quad (3.7)$$

onde  $f_i^{Ag}(t)$  é o *fitness* da célula  $i$ .

De acordo com as Equações 3.6 e 3.7, é possível perceber que as células que possuem afinidade total com outros anticorpos nula ( $f_i^{Ab}(t) = 0$ ), ou seja, aqueles indivíduos que são os melhores dentro de sua vizinhança, sofrem um aumento de concentração proporcional ao *fitness* de até 10%, enquanto que células que estão dentro da região de influência (vizinhança) de células de maior *fitness* ( $f_i^{Ab}(t) > 0$ ) sofrem uma redução de concentração de pelo menos 30%. Estes valores de 10% e 30% foram definidos empiricamente após uma série de experimentos preliminares.

Diante das Equações 3.6 e 3.7, fica evidente uma diferença fundamental deste novo modelo para a proposta original de Bersini (2002). No modelo original, células que recebem um estímulo muito alto são suprimidas da rede, mesmo que elas também tenham uma alta afinidade com os antígenos presentes no organismo. No contexto de otimização, este comportamento nem sempre é desejável, uma vez que, teoricamente, poderia ocorrer a supressão de todas as melhores soluções já encontradas durante o processo de otimização. Sendo assim, neste novo modelo de concentrações, a afinidade entre células só é não-nula para aqueles indivíduos que estão na vizinhança de alguma célula com *fitness* superior (conjunto  $J$  na Equação 3.5). Com esta abordagem, as melhores células de cada vizinhança nunca são suprimidas da população, o que permite que o algoritmo cob-aiNet mantenha um certo grau de elitismo em seu processo de otimização, essencial para a sua convergência.

Como é possível perceber, as modificações feitas sobre o modelo de Bersini (2002) foram profundas e acabaram levando a uma nova proposta significativamente diferente. No entanto, a ideia geral do modelo original foi mantida e serviu como base fundamental para o algoritmo cob-aiNet, que será discutido em detalhes na próxima seção.

### 3.3 O Algoritmo cob-aiNet

Os passos principais do algoritmo cob-aiNet estão apresentados no Pseudo-código 3.2. Como é possível perceber, a estrutura geral deste novo algoritmo é, de certa forma, similar à de outros SIAs da literatura como, por exemplo, a opt-aiNet (discutida na Seção 2.2.2). No entanto, como visto na Seção 3.2.2, apesar destas aparentes semelhanças a cob-aiNet é baseada em um modelo de rede imunológica completamente diferente do que vem sendo adotado até então em algoritmos imunoinspirados para otimização, além de também introduzir alguns novos operadores e utilizar versões melhoradas de outros previamente apresentados na literatura.

---

**Pseudo-código 3.2** Passos principais do algoritmo *cob-aiNet*.
 

---

**Parâmetros:**

- $nAB$ : número inicial de células na população;
- $maxAB$ : tamanho máximo da população;
- $nC^{max}$ : número máximo de clones por célula;
- $nC^{min}$ : número mínimo de clones por célula;
- $C_0$ : concentração inicial de cada célula;
- $\sigma_s$ : limiar de supressão;
- $\beta^i$ : parâmetro inicial do operador de hipermutação;
- $\beta^f$ : parâmetro final do operador de hipermutação;
- $LS_{it}$ : número de iterações de busca local;
- $LS_{freq}$ : frequência da busca local;
- $SR$ : intervalo de busca inicial, para cada variável, do operador de busca local;

- 1- Gerar aleatoriamente a população inicial de tamanho  $nAB$ ;
  - 2- Avaliar o *fitness* das células na população inicial;
  - 3- Avaliar a afinidade entre as células da população inicial;
- enquanto** (critério de parada não estiver satisfeito) **faça**
- 4- Definir o número de clones  $nC_i$  que deve ser gerado para cada célula  $i$ ;
  - 5- Gerar  $nC_i$  clones para cada célula  $i$  da população;
  - 6- Aplicar o operador de hipermutação a cada um dos clones gerados;
  - 7- Avaliar o *fitness* das novas células geradas;
  - 8- Selecionar as células que deverão ser mantidas na próxima iteração (com inserção);
  - 9- Avaliar a afinidade entre as células da população;
  - 10- Atualizar a concentração de todas as células;
- se** (iteração atual) **mod**  $LS_{freq} == 0$  **então**
- 11- Aplicar o operador de *Busca Local* a todos os indivíduos e atualizar os respectivos *fitness*;
  - 12- Avaliar a afinidade entre as células da população;
  - 13- Atualizar a concentração de todas as células;
- fim se**
- 14- Eliminar da população as células com concentração nula;
  - 15- Avaliar a afinidade entre as células da população;
- fim enquanto**
- 16- Aplicar o operador de *Busca Local* a todos os indivíduos e atualizar os respectivos *fitness*;
  - 17- Avaliar a afinidade entre as células da população;
  - 18- Atualizar a concentração de todas as células;
  - 19- Eliminar da população as células com concentração nula;
- 

Recapitulando o que foi apresentado na Seção 3.2.2, os indivíduos (células) na população da *cob-aiNet* são codificados em vetores de atributos reais e, a cada um destes indivíduos, está associado um valor de *concentração*, que é atualizado a cada iteração do algoritmo conforme a Equação 3.6. Além disso, como não existe uma representação explícita do conceito de antígenos na modelagem adotada, a *cob-aiNet* utiliza uma métrica de *afinidade entre células e antígenos* do tipo *fitness*, previamente definida na Equação 3.4. Já no caso da *afinidade entre células da população*, que é essencial para

a modelagem dos comportamentos de estímulo e supressão, o algoritmo se baseia tanto na distância euclidiana entre os vetores de atributos reais de cada indivíduo quanto em suas respectivas concentrações, como formalizado na Equação 3.5.

De maneira geral, além da inspiração proveniente da teoria da Rede Imunológica de Jerne (1974), refletida no algoritmo através do modelo descrito na Seção 3.2.2, a cob-aiNet também possui mecanismos inspirados no princípio da Seleção Clonal de Burnet (1978), que podem ser diretamente relacionados ao trio de operadores de *clonagem*, *hipermutação* e *seleção*, representados nos passos 5–9 do Pseudo-código 3.2. Tanto estes três operadores quanto os demais mecanismos presentes na cob-aiNet serão descritos em detalhes nas próximas seções.

### 3.3.1 Clonagem e Hipermutação

Na cob-aiNet, o número de clones gerado para cada célula da população a cada iteração é definido dinamicamente e de maneira diretamente proporcional à *concentração* de cada célula, como pode ser visto na Equação 3.8. Esta variação do número de clones de cada célula a cada iteração corresponde à reprodução dos mecanismos de *ativação* celular. Conforme o modelo de rede imunológica descrito na Seção 3.2.2, as células mais ativas da população possuem maior concentração, e isso é refletido, na prática, na geração de um maior número de clones.

$$nC_t^i = \lceil C_t^i \cdot (nC^{max} - nC^{min}) + nC^{min} \rceil, \quad (3.8)$$

onde  $nC_t^i$  é o número de clones que deve ser gerado para cada célula  $i$  na iteração  $t$ ,  $C_t^i$  é a concentração da célula  $i$  na iteração  $t$ ,  $nC^{max}$  e  $nC^{min}$  são, respectivamente, o número máximo e mínimo de clones que deve ser gerado para cada célula (definidos pelo usuário - vide Pseudo-código 3.2), e  $\lceil \cdot \rceil$  é o operador *teto*, que leva ao maior inteiro subsequente.

Após a etapa de clonagem, cada clone sofre um processo de hipermutação, em que os indivíduos sofrem variação genética inversamente proporcional ao seu *fitness*. No algoritmo cob-aiNet, o processo de hipermutação é relativamente similar ao do algoritmo opt-aiNet (de Castro & Timmis, 2002a), descrito na Seção 2.2.2, e é dado a seguir:

$$\vec{A}b_{t+1}^i = \vec{A}b_t^i + \beta_t \cdot e^{-f_i^{Ag}(t) \cdot C_t^i} \cdot \mathcal{N}^i(\vec{0}, \mathbf{I}), \quad (3.9)$$

onde  $\vec{A}b_{t+1}^i$  é a nova célula  $i$ ,  $\vec{A}b_t^i$  é a célula  $i$  original,  $f_i^{Ag}(t)$  é o *fitness* da célula  $i$  original,  $C_t^i$  é a concentração da célula  $i$  original,  $\mathcal{N}^i(\vec{0}, \mathbf{I})$  é um vetor aleatório gaussiano de média  $\vec{0}$  e covariância igual à matriz identidade  $\mathbf{I}$ , e  $\beta_t$  é um parâmetro que controla a amplitude da mutação na iteração  $t$ . Com relação ao operador de hipermutação original da opt-aiNet, apresentado na Seção 2.2.2, é possível perceber duas diferenças na Equação 3.9: (i) o parâmetro  $\beta$  agora é dependente da iteração

atual  $t$  em que se encontra o algoritmo; e (ii) a concentração  $C_t^i$  do indivíduo  $i$  passou a fazer parte do termo exponencial.

A introdução da concentração  $C_t^i$  no termo exponencial da Equação 3.9 se deu com o intuito de promover uma maior variabilidade genética nos clones gerados a partir de células de menor concentração. Como visto na Seção 3.2.2, a concentração de uma dada célula só é reduzida quando sua afinidade com as demais células da população é não-nula, ou seja, quando a célula em questão se encontra na vizinhança de outros indivíduos de maior *fitness*. Sendo assim, o algoritmo deve estimular o afastamento destas células de baixa concentração da região de atuação das melhores células, o que pode ser obtido com a aplicação de uma maior variação pelo mecanismo de hipermutação. Como a concentração  $C_t^i$  é multiplicada pelo *fitness*  $f_i^{Ag}(t)$  da célula original no termo exponencial da Equação 3.9, o operador de hipermutação da *cob-aiNet* tende a aplicar variações maiores àqueles clones gerados tanto a partir de células ruins (baixo *fitness*) quanto a partir de células de baixa concentração, enquanto que clones gerados a partir de bons indivíduos no contexto global (maiores valores de *fitness*) e a partir dos melhores indivíduos em suas respectivas vizinhanças (maiores concentrações) tendem a sofrer uma menor variabilidade genética.

Como visto na Seção 2.2.2, o parâmetro  $\beta$  no algoritmo *opt-aiNet* é estático e deve ser definido pelo usuário, o que pode ser uma tarefa complexa já que  $\beta$  é altamente dependente das características do problema. Altos valores de  $\beta$  tendem a manter o algoritmo com um comportamento predominantemente exploratório que pode até mesmo impedir sua convergência, enquanto que valores muito baixos para o parâmetro  $\beta$  podem acabar resultando em uma convergência prematura para ótimos locais. Sendo assim, na *cob-aiNet* foi proposto um novo mecanismo que ajusta o valor do parâmetro  $\beta$  ao longo das iterações (conforme descrito na Equação 3.10), para que o algoritmo seja capaz de explorar o espaço de busca nas iterações iniciais de sua execução e, com o passar do tempo, gradualmente alterar seu comportamento para que possa realizar, ao final, o ajuste fino de cada solução, explorando apenas sua vizinhança. Na Figura 3.2, um exemplo desta variação do parâmetro  $\beta$  da hipermutação, tomada ao longo de uma execução com 1000 iterações do algoritmo *cob-aiNet*, está ilustrada.

$$\beta_t = \frac{\beta^i - \beta^f}{1 + e^{\frac{20}{t_{max}} \cdot (t - \frac{t_{max}}{2})}} + \beta^f, \quad (3.10)$$

onde  $\beta_t$  é o valor do parâmetro  $\beta$  na iteração  $t$  (ou avaliação de função  $t$ , conforme o critério de parada),  $\beta^i$  e  $\beta^f$  são os valores inicial (máximo) e final (mínimo) de  $\beta$ , definidos pelo usuário, e  $t_{max}$  é o número máximo de iterações (ou avaliações de função, conforme o critério de parada).

Este novo mecanismo de obtenção de  $\beta$  possui duas vantagens quando comparado à abordagem original em que  $\beta$  correspondia a um parâmetro fixo definido pelo usuário: (i) apesar de o número de parâmetros a ser ajustado pelo usuário ter aumentado (o usuário deve ajustar  $\beta^i$  e  $\beta^f$ ), o algoritmo

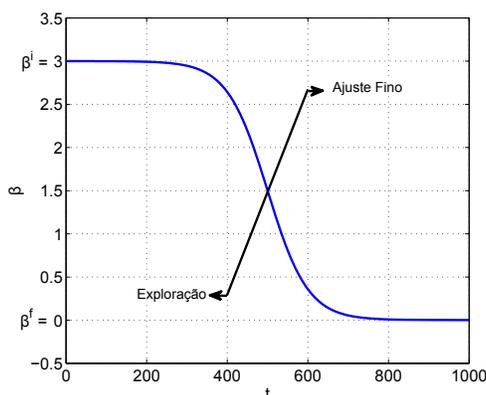


Fig. 3.2: Ilustração da variação do parâmetro  $\beta$  em função da iteração  $t$ . Neste exemplo,  $\beta$  foi calculado de acordo com a Equação 3.10, sendo  $\beta^i = 3$ ,  $\beta^f = 0,001$  e  $t_{max} = 1000$ .

agora opera em um intervalo dentro do qual  $\beta$  pode variar, o que deixa a ferramenta muito mais robusta frente à alternativa em que  $\beta$  permanecia com um valor fixo (e possivelmente inadequado) por toda a sua execução; e (ii) a variação do comportamento da cob-aiNet ao longo de sua execução, de exploratório no início para um modo de ajuste-fino ao final, permite a estabilização da concentração de cada célula nas iterações finais da execução, já que mutações capazes de deslocar significativamente uma dada célula no espaço de busca dificilmente ocorrerão e as células tenderão então apenas a convergir para os melhores pontos dos vales/picos (ótimos locais) em que foram previamente posicionadas. Neste segundo caso, durante a convergência para os ótimos locais mais próximos, caso exista mais de uma solução na região de um mesmo ótimo local, a concentração das piores células diminuirá gradualmente até que se torne nula, o que resultará na eliminação de tais células da população (supressão).

Apesar das vantagens introduzidas por este novo mecanismo de variação do parâmetro  $\beta$ , o comportamento puramente monotônico ilustrado na Figura 3.2 não permite que o algoritmo volte a ampliar o escopo de sua busca caso a melhor solução da população convirja prematuramente. O ideal, nestes casos, seria que o valor de  $\beta$  pudesse ser reiniciado como ilustrado na Figura 3.3(a), ou que pelo menos sofresse um aumento como o apresentado na Figura 3.3(b), permitindo assim a ampliação da capacidade de exploração do algoritmo e, conseqüentemente, aumentando a possibilidade de localização de melhores soluções.

A implementação deste mecanismo alternativo de variação do parâmetro  $\beta$  é direta a partir da Equação 3.10. Para que se consiga o comportamento ilustrado na Figura 3.3(a), basta substituir  $t$  por  $t - \Delta$  e  $t_{max}$  por  $t_{max} - \Delta$ , onde  $\Delta$  corresponde à iteração em que deve ocorrer o reinício. Caso o comportamento desejado seja equivalente ao ilustrado na Figura 3.3(b), além destas duas substituições deve-se também adotar  $\beta^i = \theta \cdot \beta^i$  em *todo* reinício, o que fará com que  $\beta$  seja reiniciado para o valor

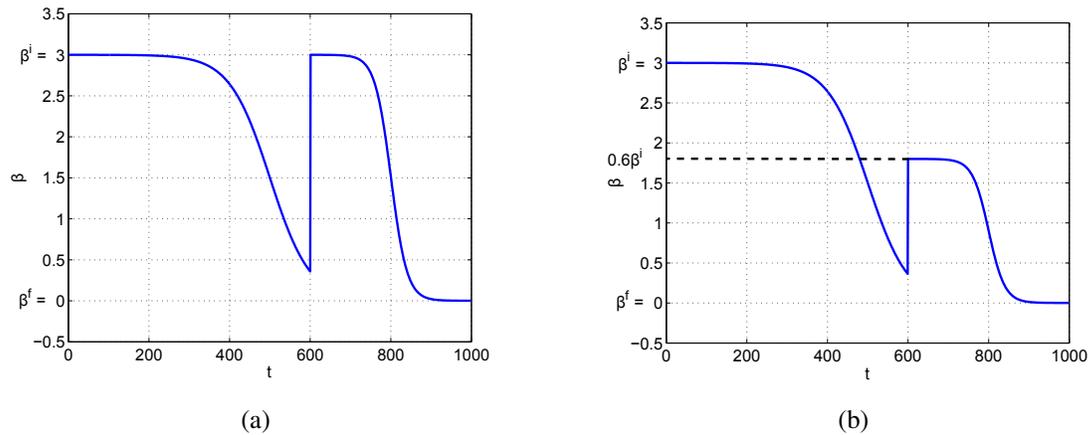


Fig. 3.3: Ilustração de propostas alternativas de variação do parâmetro  $\beta$  em função da iteração  $t$ . Neste exemplo,  $\beta$  foi calculado utilizando-se  $\beta^i = 3$ ,  $\beta^f = 0,001$  e  $t_{max} = 1000$ . Em (a),  $\beta$  foi reiniciado para  $\beta^i = 3$ , enquanto que em (b) o reinício foi feito para  $0,6 \cdot \beta^i = 1,8$ . Em ambos os casos, o reinício de  $\beta$  ocorreu em  $t = 600$ .

$\theta \cdot \beta^i$ . No exemplo dado na Figura 3.3(b), foi adotado  $\theta = 0,6$ . Para que seja detectada a estagnação do processo de busca, a forma mais simples é verificar se a melhora do *fitness* da melhor solução da população não é superior a um determinado limiar  $\xi$  por  $n$  iterações.

Apesar de tornar o algoritmo mais flexível, estes mecanismos alternativos de variação do parâmetro  $\beta$ , ao longo da busca, têm como principal desvantagem a introdução de novos parâmetros a serem ajustados pelo usuário ( $n$ ,  $\Delta$ ,  $\xi$  e  $\theta$ ), que se somam aos demais previamente listados no Pseudo-código 3.2.

Em conjunto com os experimentos que serão apresentados e discutidos no Capítulo 4, foi feita uma série de testes comparativos para se avaliar a necessidade ou não de utilização da estratégia de reinício de  $\beta$ , e observou-se que, para os problemas que serão tratados aqui, não houve nenhuma variação sensível de desempenho do algoritmo. Sendo assim, no restante desta tese foi adotado o mecanismo de variação de  $\beta$  *sem reinícios*, dado pela Equação 3.10 e ilustrado na Figura 3.2. No entanto, recomenda-se a utilização das proposições alternativas discutidas acima sempre que forem identificadas situações de convergência prematura da evolução do algoritmo.

### 3.3.2 Seleção e Inserção de Novos Indivíduos

Após os passos de clonagem e hipermutação, um mecanismo de *seleção* é aplicado a cada *subpopulação* de indivíduos, formada por cada célula original da população e seus clones após hipermutação, para que possam ser selecionados aqueles indivíduos que permanecerão na população na próxima iteração. Geralmente, nos demais algoritmos da família aiNet, este operador de seleção es-

colhe um único indivíduo de cada sub-população para permanecer na população, o que faz com que o número de indivíduos presentes na população se mantenha o mesmo de antes da etapa de clonagem (de França et al., 2010a).

Já na cob-aiNet, este operador de seleção também é responsável pela *inserção* de indivíduos extras na população, desde que tais células tenham sido posicionadas em regiões do espaço de busca ainda não exploradas pela população atual do algoritmo. Em outras palavras, a inserção no algoritmo cob-aiNet ocorre sempre que um clone é posicionado, após sofrer hipermutação, em um local em que ele esteja mais distante que o limiar de supressão  $\sigma_s$  (vide Pseudo-código 3.2), tanto da célula que lhe deu origem quanto de todas as demais células já selecionadas para permanecerem na população. Os detalhes deste novo operador de seleção e inserção estão descritos no Pseudo-código 3.3.

---

**Pseudo-código 3.3** Mecanismo de seleção e inserção de novos indivíduos.

---

**Entrada:**

- $maxAB$ : tamanho máximo da população;
- $\sigma_s$ : limiar de supressão;
- $AB$ : células originais da população;
- $C_k$ : clones originados da célula  $k$ , após hipermutação;
- $d(\cdot, \cdot)$ : distância euclidiana;

**para todo** (indivíduo  $i$  em  $AB$ ) **faça**

$c$  = melhor célula presente em  $C_i$ ;

**se** ( $[d(AB_i, c) > \sigma_s]$  e  $[|AB| < maxAB]$  e  $[d(c, k) > \sigma_s$ , para toda célula  $k$  já selecionada]) **então**

Insera tanto  $c$  quanto  $AB_i$  na nova população {Neste caso,  $c$  passa a ter a concentração inicial};

**senão**

$s$  = melhor indivíduo entre  $c$  e  $AB_i$ ;

Insera  $s$  na nova população {Mantendo a concentração de  $AB_i$ };

**fim se**

**fim para**

---

O objetivo principal da inserção de novos indivíduos nos SIAs é, basicamente, a introdução de diversidade na população de soluções candidatas, o que tende a aumentar a capacidade de exploração dos algoritmos. Diferentemente da cob-aiNet, a maioria dos outros algoritmos imunoinspirados da literatura simplesmente incorpora, de tempos em tempos, um certo número de novos indivíduos aleatoriamente gerados (Castro & Von Zuben, 2010a; de França et al., 2010a), o que certamente é menos custoso computacionalmente do que o procedimento adotado pela cob-aiNet, uma vez que não envolve comparações entre indivíduos da população. Por outro lado, a geração puramente aleatória de indivíduos promissores se torna cada vez mais difícil ao longo da evolução do algoritmo, principalmente em problemas com número de variáveis (dimensão) elevado, uma vez que praticamente todos os novos indivíduos inseridos ou são suprimidos ao se aproximarem de outras soluções já presentes na população ou permanecem em regiões pouco interessantes do espaço de busca. Já a cob-aiNet, ape-

sar de possuir um mecanismo de inserção mais custoso, garante que apenas aquelas novas soluções que sejam tanto promissoras quanto diversas em relação às demais já presentes na população serão inseridas e receberão a chance de evoluírem para ótimos locais ainda inexplorados. Além disso, este custo extra do mecanismo de inserção pode acabar não sendo tão relevante em situações práticas, uma vez que, nestes casos, o custo computacional de avaliação da função-objetivo tende a corresponder à maior carga computacional em algoritmos de otimização.

Este mecanismo de seleção, introduzido na *cob-aiNet*, está intrinsecamente relacionado à variação dinâmica do parâmetro  $\beta$  da hipermutação (vide Seção 3.3.1), uma vez que a maioria dos novos indivíduos são inseridos nas primeiras iterações, quando o algoritmo possui um valor alto de  $\beta$  e se encontra na fase exploratória. Como a *cob-aiNet* tem uma certa tendência a apresentar, nesta fase inicial, uma explosão no número de indivíduos para alguns problemas (principalmente em situações com espaço de busca de alta dimensão), foi estabelecido aqui um limite superior para o tamanho da população, que é definido pelo usuário através do parâmetro  $maxAB$  (vide Pseudo-códigos 3.2 e 3.3). Passadas as iterações iniciais em que o espaço de busca não está completamente povoado e a capacidade de exploração do algoritmo está no seu pico, o número de células na população tende a diminuir (conforme o problema sendo tratado), uma vez que começam a ocorrer supressões e o comportamento do algoritmo gradualmente muda para a fase de ajuste fino.

Na Figura 3.4(a), é possível observar este comportamento para o problema de teste  $F_{12}$  em  $\mathbb{R}^2$ , que será detalhado no Apêndice A. Nesta figura, podemos notar que, nas iterações iniciais, a população cresce vertiginosamente até que o limite máximo  $maxAB$  seja atingido. Após algumas iterações, as soluções presentes na população começam a convergir para os ótimos locais mais próximos, supressões passam a ocorrer e o número de indivíduos diminui até estabilizar. Já para o problema  $F_9$ , representado na Figura 3.4(b) também em  $\mathbb{R}^2$  (vide Apêndice A), isto não ocorre, uma vez que a população não atinge o limite máximo  $maxAB$  e acaba se estabilizando em um número de indivíduos superior ao tamanho da população inicial.

### 3.3.3 Atualização de Concentração e Supressão

O mecanismo de atualização da concentração de cada célula na população da *cob-aiNet* já foi previamente discutido na Seção 3.2.2 (mais especificamente, na Equação 3.6) e não será repetido aqui. No entanto, é importante ressaltar que, quando uma determinada célula passa a ter concentração nula, ela é eliminada da população, já que isto é um indicativo de que tal célula se manteve na vizinhança de outros indivíduos com *fitness* superior ao seu por algumas iterações, o que levou à sua supressão (vide Seção 3.2.2).

Este mecanismo de supressão da *cob-aiNet* é diferente dos presentes em outras redes imunológicas artificiais como a *opt-aiNet* (de Castro & Timmis, 2002a), em que basta que uma dada célula se

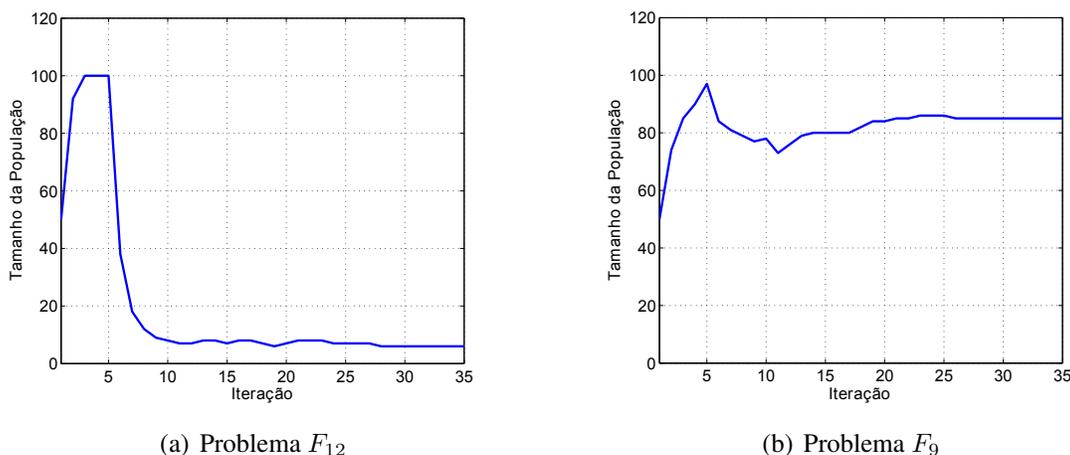


Fig. 3.4: Ilustração da variação do tamanho da população durante uma única execução do algoritmo cob-aiNet para os problemas (a)  $F_{12}$  e (b)  $F_9$ , ambos em  $\mathbb{R}^2$ . Nestes exemplos,  $n_{AB} = 50$  e  $max_{AB} = 100$ .

posicione na região de influência de outra célula de melhor *fitness* por uma única iteração para que ela seja eliminada da população. Na cob-aiNet, as soluções que se aproximam de indivíduos melhores geralmente são penalizadas, em um primeiro momento, apenas com a redução do número de clones que serão gerados na próxima iteração (que é um reflexo direto da redução de concentração), o que ainda lhes dá algumas iterações extras para tentar escapar desta região. A supressão direta só ocorre na cob-aiNet quando uma célula de baixo *fitness* entra em uma região em que estão presentes diversas células de melhor qualidade, o que faz com que tal célula sofra uma forte supressão e possa acabar tendo sua concentração zerada em uma única iteração (vide Equação 3.6).

É importante ressaltar que os mecanismos de supressão e inserção de novos indivíduos (vide Seção 3.3.2) são os dois operadores responsáveis por permitir que a cob-aiNet apresente a característica de variação dinâmica do número de indivíduos em sua população, de acordo com as características do problema. Para ilustrar como isso ocorre, suponha um problema monomodal de minimização com um único vale. Neste caso, por mais que o mecanismo de inserção gere novas soluções nas iterações iniciais do algoritmo, estas soluções vão acabar convergindo para o mesmo ótimo local do problema, o que levará à diminuição da concentração daquelas células de pior *fitness* e à sua consequente supressão, fazendo com que o algoritmo tenda a convergir para apenas uma única solução. Por outro lado, se for considerado agora um problema multimodal de minimização, em que o espaço de busca possui um grande número de vales distintos, é grande a possibilidade de que a maioria dos novos indivíduos gerados na etapa de inserção acabe convergindo para vales distintos ao longo da evolução da população, o que evita que tais indivíduos invadam a região de supressão de células melhores (desde que o raio de supressão  $\sigma_s$  – vide Pseudo-código 3.2 – esteja corretamente definido, como será discutido a seguir) e acabem sendo suprimidos.

Como é possível notar, diante de tudo que foi apresentado até o momento, o limiar (raio) de supressão  $\sigma_s$  é o principal parâmetro associado à manutenção de diversidade no algoritmo *cob-aiNet*. Caso  $\sigma_s$  seja definido com um valor muito alto frente ao domínio do problema, dependendo das características de multimodalidade presentes, é possível que cada célula tenha uma região de influência que englobe mais de um ótimo local, o que impediria que tais ótimos locais fossem encontrados por indivíduos diferentes (vide Figura 3.5(a)). Por outro lado, caso este parâmetro seja definido com valores muito baixos, provavelmente a explosão populacional observada nas primeiras iterações do algoritmo seria mais intensa, e existiria a chance de que, caso a etapa de ajuste fino não consiga refinar adequadamente as soluções, sejam retornados ao final da execução mais de um indivíduo por vale do problema (considerando apenas o caso de minimização), sendo a maioria deles sub-ótimos (vide Figura 3.5(b)). Diante disso, cabe ao usuário ponderar qual destas situações é mais desejável para o seu problema, quando for definir a estratégia de ajuste de parâmetros que será adotada.

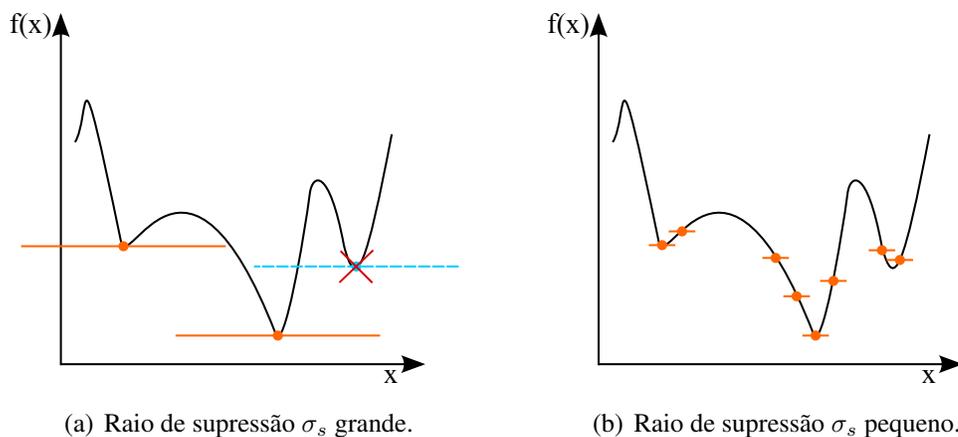


Fig. 3.5: Possíveis posicionamentos finais de soluções de acordo com o raio de supressão  $\sigma_s$  (barras horizontais). Neste exemplo, supõe-se que o problema  $f(x)$  deva ser minimizado. No caso (a)  $\sigma_s$  foi definido com um valor muito alto, o que faz com que a solução em azul seja suprimida e o ótimo local correspondente não seja retornado para o usuário. Já no caso (b),  $\sigma_s$  foi definido com um valor muito baixo, o que pode levar à permanência de múltiplas soluções em cada vale caso as iterações finais de ajuste fino não consigam refinar adequadamente as soluções. A supressão ocorre se a barra horizontal incluir o centro da barra horizontal de uma solução melhor.

Para facilitar o ajuste do parâmetro  $\sigma_s$ , pode-se torná-lo relativo à máxima distância possível entre duas soluções no espaço de busca do problema em questão, o que pode ser obtido em todos os casos em que cada variável tenha um domínio limitado. Para ilustrar, suponha que distância máxima em um espaço de busca hipotético seja  $D^{max} = 10$ . Com isso, um valor  $\sigma_s = 0,1$  determina que cada célula  $i$  da população terá uma região de influência que corresponde a uma hiper-esfera centrada em  $i$  e com raio equivalente a 10% da maior distância possível no espaço de busca, ou seja,  $r = \sigma_s \cdot D^{max} = 1$ . Ao adotar esta abordagem, deve-se garantir que  $\sigma_s \in [0, 1]$ . Em todos os experimentos que serão

apresentados e discutidos no Capítulo 4, foi empregada esta abordagem de valores relativos para o parâmetro  $\sigma_s$ .

### 3.3.4 Busca Local

Um dos principais problemas dos SIAs baseados no princípio da Seleção Clonal de Burnet (1978) está no fato de eles empregarem um único operador estocástico de mutação, que geralmente modifica todas as variáveis de uma dada solução candidata de uma única vez. Ao adotar tais operadores, o processo de ajuste fino das soluções encontradas tende a ser comprometido, uma vez que a probabilidade de obtenção da direção e amplitude ideais da modificação que deve ser aplicada a uma dada solução para que seja obtida um indivíduo melhor é baixa. Sendo assim, como a cob-aiNet é inspirada no princípio de Seleção Clonal e utiliza um operador de mutação com estas características (vide Seção 3.3.1), neste trabalho foi proposta a utilização de um mecanismo de *busca local*, com o intuito de amenizar este efeito.

Apesar de qualquer operador de busca local para problemas de otimização em espaços contínuos poder ser utilizado na cob-aiNet, neste trabalho foi escolhido o operador *LocalSearch1*, proposto como parte do algoritmo *MTS* (Tseng & Chen, 2008), que foi o vencedor da competição de *Otimização Global de Larga Escala* do *IEEE Congress on Evolutionary Computation* (CEC) em 2008 (Tang et al., 2007). Este operador de busca local opera sequencialmente e independentemente em cada dimensão (variável) das soluções candidatas do algoritmo, aumentando e diminuindo os valores presentes em cada uma destas variáveis conforme um *intervalo de busca*. Caso este processo de aumento ou diminuição do valor das variáveis não leve a uma melhora da solução candidata, o intervalo de busca é reduzido. O funcionamento detalhado deste operador de busca local está descrito no Pseudocódigo 3.4.

Na proposta original original de Tseng & Chen (2008), o intervalo de busca do operador *LocalSearch1* é inicializado com um valor correspondente a metade do domínio de cada variável, ou seja,  $SR = (\text{limite superior} - \text{limite inferior})/2$ . Como na cob-aiNet espera-se que cada solução candidata seja refinada em relação ao ótimo local mais próximo e não dentro de todo o espaço de busca, sugere-se então que o intervalo de busca de *LocalSearch1* seja inicializado com o raio de supressão  $\sigma_s$ , o que fará com que a busca local explore apenas a região vizinha a cada solução. No entanto, cabe ao usuário definir qual o seu objetivo principal ao aplicar o algoritmo cob-aiNet, e inicializar *SR* da maneira mais adequada.

Por fim, um último aspecto que cabe ressaltar aqui, a respeito deste operador de busca local, é o fato dele exigir um elevado número de avaliações da função-objetivo (até duas avaliações para cada variável de cada solução candidata em cada iteração), o que fez com que, na cob-aiNet, ele fosse aplicado apenas periodicamente (mais especificamente, a cada  $LS_{freq}$  iterações – vide Pseudo-

**Pseudo-código 3.4** Operador de busca local *LocalSearch1*.**Parâmetros de Entrada:**

- $X_k$ :  $k$ -ésima solução candidata da população;
- $D$ : número de variáveis (dimensão) do problema;
- $N$ : número de soluções candidatas na população;
- $SR$ : matriz de dimensões  $N \times D$  com intervalos de busca para cada dimensão de cada solução candidata;
- $Improve$ : vetor booleano, de tamanho  $N$ , que indica se houve melhorias ou não na última execução do operador para cada solução candidata;
- limite superior: vetor de tamanho  $D$  com os limites superiores do domínio de cada variável do problema;
- limite inferior: vetor de tamanho  $D$  com os limites inferiores do domínio de cada variável do problema;

**Parâmetros de Saída:**

- $SR$ : matriz com intervalos de busca atualizados para a solução candidata  $k$  (linha  $k$  atualizada);
- $Improve$ : vetor booleano com a posição  $k$  atualizada, que indica se houve melhorias na solução  $X_k$ ;
- $X_k$ : nova  $k$ -ésima solução candidata da população (melhor ou igual à original);

**se** ( $Improve[k] = \text{FALSE}$ ) **então**

$SR[k, :] \leftarrow SR/2$ ; {Todas as colunas da linha  $k$  são atualizadas}

**se** (algum valor da linha  $k$  de  $SR < 1 \cdot e^{-15}$ ) **então**

$SR[k, :] \leftarrow 0,4 \cdot (\text{limite superior} - \text{limite inferior})$ ; {Todas as colunas da linha  $k$  são atualizadas}

**fim se**

**fim se**

$Improve[k] \leftarrow \text{FALSE}$ ;

**para**  $i = 1$  até  $D$  **faça**

$X_k[i] \leftarrow X_k[i] - SR$ ;

Avalia o valor da função-objetivo para  $X_k$ ;

**se** (não melhorou nem piorou a solução  $X_k$ ) **então**

Restaura o valor original de  $X_k[i]$ ;

**senão**

**se** (a solução  $X_k$  piorou) **então**

Restaura o valor original de  $X_k[i]$ ;

$X_k[i] \leftarrow X_k[i] + 0,5 \cdot SR$ ;

**se** (a solução  $X_k$  não melhorou) **então**

Restaura o valor original de  $X_k[i]$ ;

**senão**

$Improve[k] \leftarrow \text{TRUE}$ ;

**fim se**

**senão**

$Improve[k] \leftarrow \text{TRUE}$ ;

**fim se**

**fim se**

**fim para**

código 3.2), em todas as soluções presentes na população e por apenas algumas iterações ( $LS_{it}$  iterações – vide Pseudo-código 3.2).

## 3.4 Noções Gerais de Complexidade do Algoritmo

Como o algoritmo cob-aiNet proposto neste trabalho é composto por uma série de operadores que levam a ferramenta a modificar seu comportamento ao longo das iterações, e também a adaptá-lo às características de cada problema, se torna difícil para o usuário identificar como o custo computacional do algoritmo varia em função de diferentes ajustes de parâmetros e, até mesmo, da dimensão (número de variáveis) do problema sendo otimizado. Sendo assim, para que se tenha uma melhor noção sobre como se dá a variação deste custo computacional da cob-aiNet, principalmente em função do número de variáveis de cada problema, nesta seção será apresentada uma estimativa de sua complexidade computacional.

Na análise que será feita aqui, o foco estará voltado principalmente para as operações de comparação entre indivíduos da população e de avaliação da função-objetivo, sempre nas situações de pior caso, uma vez que estas tendem a ser as duas operações mais custosas no contexto desta ferramenta. Detalhes específicos sobre os custos computacionais associados às demais operações executadas pela ferramenta também serão discutidos aqui, mas apenas nas situações em que merecerem destaque.

### 3.4.1 Geração da População Inicial

Na geração da população inicial do algoritmo cob-aiNet, dada no passo 1 do Pseudo-código 3.2, é criada uma população inicial com  $nAB$  células, sendo o *fitness* de tais células calculado na etapa seguinte. Como não é feita nenhuma comparação entre estes novos indivíduos, o procedimento mais custoso nesta etapa do algoritmo são as  $nAB$  avaliações da função-objetivo do problema. Além disso, cabe destacar que, nesta etapa, deve ser gerado um vetor de valores aleatórios de dimensão  $D$  para cada nova solução da população, onde  $D$  corresponde ao número de variáveis (dimensão) do problema.

### 3.4.2 Avaliação de Afinidade entre Células

Após a geração da população inicial e do cálculo do *fitness* destes novos indivíduos, no passo 3 do Pseudo-código 3.2 é feita a primeira avaliação da métrica de afinidade entre as células. No cálculo desta métrica, a distância euclidiana par a par entre todas as soluções presentes na população deve ser obtida, para que sejam identificadas quais são as soluções que estão dentro do raio de vizinhança  $\sigma_s$  de cada indivíduo (vide Seção 3.2.2). Sendo assim, o número de comparações entre indivíduos exigido a cada cálculo de afinidade entre células é dado por  $N(t)^2$ , onde  $N(t)$  corresponde ao número de indivíduos presentes na população na iteração  $t$ . No caso específico do passo 3 do Pseudo-código 3.2, são feitas  $nAB^2$  comparações, onde  $nAB$  é o número inicial de indivíduos na população do algoritmo.

Além do passo 3 do Pseudo-código3.2, a métrica de afinidade entre células é calculada também nos passos 9, 12 e 15, dentro do *loop* principal da ferramenta, e no passo 17 antes do encerramento da execução. Nestas quatro situações, o número máximo de comparações entre células (pior caso) ocorrerá quando a população estiver com o número máximo  $maxAB$  permitido de indivíduos. Sendo assim, o número total de comparações entre células exigido durante o cálculo desta métrica de afinidade, em uma execução de  $numIt$  iterações do algoritmo *cob-aiNet*, será dado por:

$$nAB^2 + (3 \cdot numIt + 1) \cdot maxAB^2. \quad (3.11)$$

Como o cálculo desta métrica de afinidade entre células não envolve a geração de nenhum novo indivíduo, nenhum cálculo adicional do valor da função-objetivo é feito. No entanto, cabe destacar que a distância euclidiana envolvida nesta operação avalia a diferença entre todas as  $D$  variáveis de cada par de soluções, o que faz com que seu cálculo varie, portanto, linearmente com a dimensão  $D$  do problema.

### 3.4.3 Clonagem

Na etapa de clonagem,  $nC_t^i$  cópias idênticas são geradas para cada célula  $i$  dentre os  $N(t)$  indivíduos presentes na população na iteração  $t$ . Por se tratar da geração de cópias do indivíduo  $i$ , nesta etapa não são feitas nem comparações entre células nem novas avaliações da função-objetivo, o que faz com que este operador não contribua para o aumento da complexidade do algoritmo pelos critérios considerados aqui.

### 3.4.4 Hipermutação

Após a geração dos  $nC_t^i$  clones para cada indivíduo  $i$  da população na iteração  $t$ , é aplicado o operador de hipermutação a tais células, o que leva a novas avaliações da função-objetivo para cada um destes novos indivíduos gerados. Sendo assim, o número de avaliações da função-objetivo executadas pelo operador de hipermutação na iteração  $t$  do algoritmo é dado por:

$$\sum_{i=1}^{N(t)} nC_t^i, \quad (3.12)$$

onde  $N(t)$  é o número de células presentes na população na iteração  $t$ .

A situação em que o operador de hipermutação apresenta o maior custo computacional possível ocorre quando são gerados  $nC^{max}$  clones para cada indivíduo  $i$  da população que, por sua vez, também deve conter o maior número possível  $maxAB$  de indivíduos. Sendo assim, no pior caso, o

número máximo de avaliações da função-objetivo exigido pelo operador de hipermutação em uma execução com  $numIt$  iterações do algoritmo cob-aiNet será dado por:

$$nC^{max} \cdot maxAB \cdot numIt, \quad (3.13)$$

onde  $nC^{max}$  e  $maxAB$  são parâmetros definidos pelo usuário (vide Pseudo-código 3.2).

Além do número de avaliações da função-objetivo dado na Equação 3.13, outro aspecto que cabe destacar aqui é o processo de modificação propriamente dito de cada uma das soluções da população. Esta modificação é feita em todas as  $D$  variáveis de cada solução da população, o que faz com que seu custo computacional varie linearmente com a dimensão  $D$  do problema sendo tratado.

### 3.4.5 Seleção com Inserção

Na etapa de seleção do algoritmo cob-aiNet, para cada conjunto de indivíduos formado por cada célula  $i$  da população, e seus respectivos clones após hipermutação, são feitas, na iteração  $t$ ,  $nC_t^i + 1$  comparações para que possa ser identificada a melhor célula. Sendo assim, no pior caso (em que se tem uma população com número máximo  $maxAB$  de indivíduos), o número total de comparações feitas entre células na etapa de seleção ao longo de uma execução completa da cob-aiNet é dado por:

$$(nC^{max} + 1) \cdot maxAB \cdot numIt \quad (3.14)$$

onde  $nC^{max}$  e  $maxAB$  são parâmetros definidos pelo usuário (vide Pseudo-código 3.2) e  $numIt$  é o número de iterações executadas pela ferramenta.

Como não são feitas novas avaliações de *fitness* nesta etapa de seleção de células, o operador de seleção não contribui para o aumento do custo computacional do algoritmo sob este aspecto.

Após a identificação da melhor solução gerada a partir de cada indivíduo  $i$  da população, através de clonagem e hipermutação, caso esta melhor célula seja um dos novos clones gerados e não a célula original  $i$ , o algoritmo avalia se tal solução deve ser inserida na população substituindo a célula original ou em conjunto com esta, caso se trate de um novo indivíduo promissor posicionado em uma região ainda inexplorada do espaço de busca (vide Seção 3.3.2). Para isso, são calculadas distâncias euclidianas entre esta nova célula selecionada, a célula  $i$  que lhe deu origem, e todos os demais indivíduos já selecionados para permanecerem na população até o momento. Sendo assim, para cada novo clone selecionado, o operador de inserção realizará o seguinte número de comparações entre células (e também de cálculos de distâncias euclidianas entre indivíduos), para decidir se deverá ser mantido na população apenas o clone selecionado ou também a célula  $i$  original:

$$\sum_{j=1}^{i-1} N^{sel}(j, t) + 1. \quad (3.15)$$

onde  $N^{sel}(j, t)$  é o número de células selecionadas, na iteração  $t$ , dentre os indivíduos presentes na sub-população formada pela célula  $j$  e seus clones após hipermutação.

Para a Expressão 3.15 acima, a situação de pior caso ocorrerá quando as etapas de inserção para os  $i - 1$  indivíduos anteriores a  $i$  levaram à manutenção tanto do melhor clone obtido quanto da célula original, ou seja,  $N^{sel}(j, t) = 2 \forall j < i$ . Sendo assim, o número de comparações dado na Expressão 3.15 se torna então:

$$\sum_{j=1}^{i-1} N^{sel}(j, t) + 1 = \sum_{j=1}^{i-1} 2 + 1 = 2 \cdot (i - 1) + 1 = 2 \cdot i - 1. \quad (3.16)$$

Supondo que o operador de inserção deve ser aplicado a todos os conjuntos de clones gerados para cada indivíduo  $i$  de uma população com tamanho máximo  $maxAB$ , no pior caso o número total de comparações (e também de cálculos de distâncias euclidianas entre indivíduos) requerido pelo operador de inserção em uma execução completa do algoritmo *cob-aiNet* com  $numIt$  iterações será dado por:

$$\begin{aligned} \left[ \sum_{i=1}^{maxAB} (2 \cdot i - 1) \right] \cdot numIt &= \left[ 2 \cdot \frac{(1 + maxAB) \cdot maxAB}{2} - maxAB \right] \cdot numIt \\ &= maxAB^2 \cdot numIt. \end{aligned} \quad (3.17)$$

Apesar do alto número de comparações entre indivíduos observado para o operador de inserção, não são feitas novas avaliações de *fitness* nesta etapa, o que faz com que o operador de inserção não contribua para o aumento da complexidade do algoritmo sob este aspecto.

### 3.4.6 Atualização de Concentrações

Nos passos de atualização de concentração do algoritmo *cob-aiNet*, a ferramenta basicamente se utiliza do *fitness* e da afinidade entre células, previamente calculados para cada indivíduo, e atualiza os respectivos valores de concentração. Diante disso, nesta etapa não há nenhuma comparação entre células da população nem novas avaliações da função-objetivo do problema, mas são feitas  $N(t)$  operações de atualização de concentração, onde  $N(t)$  é o número de indivíduos presentes na população na iteração  $t$ .

### 3.4.7 Busca Local

O operador de busca local empregado no algoritmo cob-aiNet (vide Seção 3.3.4) é o operador que tende a realizar o maior número de avaliações da função-objetivo e comparações entre indivíduos, sendo, por esta razão, empregado apenas a cada  $LS_{freq}$  iterações do *loop* principal do algoritmo (vide Pseudo-código 3.2).

No total, em uma execução completa do algoritmo cob-aiNet com  $numIt$  iterações, o operador de busca local é aplicado a todos os indivíduos da população  $(numIt/LS_{freq} + 1) \cdot LS_{it}$  vezes (vide Pseudo-código 3.2), sendo que, a cada aplicação, no máximo  $2 \cdot D$  variações são feitas em um dado indivíduo (vide Pseudo-código 3.4), onde  $D$  é o número de variáveis (dimensão) do problema. A cada modificação feita em uma célula, uma nova avaliação da função-objetivo deve ser feita, juntamente com uma comparação com a célula original. Considerando-se então que, no pior caso, a população possui o número máximo  $maxAB$  permitido de soluções, o número total de comparações entre indivíduos e de novas avaliações da função-objetivo exigidos pelo operador de busca local adotado aqui é dado por:

$$2 \cdot D \cdot maxAB \cdot LS_{it} \cdot \left( \frac{numIt}{LS_{freq}} + 1 \right). \quad (3.18)$$

### 3.4.8 Supressão

As etapas de supressão de células da população do algoritmo cob-aiNet basicamente analisa o valor da concentração de cada célula da população e, caso o valor apresentado seja nulo, a célula em questão é eliminada. Da mesma maneira que as etapas de clonagem e atualização de concentrações, na supressão não ocorrem nem comparações entre indivíduos nem novas avaliações da função-objetivo do problema, sendo feitas apenas  $N(t)$  operações de verificação de concentração, onde  $N(t)$  é o número de indivíduos presentes na população na iteração  $t$ .

### 3.4.9 Número Total de Comparações e Avaliações de Função-objetivo

De maneira geral, em uma execução completa do algoritmo cob-aiNet por  $numIt$  iterações, o número total de comparações entre soluções presentes na população é, no pior caso, dado pela Expressão 3.19 a seguir, enquanto que o número total de avaliações da função-objetivo, também no pior caso, é dado pela Expressão 3.20.

$$nAB^2 + [(3 \cdot numIt + 1) \cdot maxAB^2] + [(nC^{max} + 1) \cdot maxAB \cdot numIt] +$$

$$+ (maxAB^2 \cdot numIt) + \left[ 2 \cdot D \cdot maxAB \cdot LS_{it} \cdot \left( \frac{numIt}{LS_{freq}} + 1 \right) \right], \quad (3.19)$$

$$nAB + (nC^{max} \cdot maxAB \cdot numIt) + \left[ 2 \cdot D \cdot maxAB \cdot LS_{it} \cdot \left( \frac{numIt}{LS_{freq}} + 1 \right) \right]. \quad (3.20)$$

Diante disso, é possível notar que o número de comparações entre soluções feitas pelo algoritmo *cob-aiNet* varia quadraticamente com o tamanho da população de soluções-candidatas (representada na Expressão 3.19 pelo seu valor de pior caso  $maxAB$ ) e linearmente com a dimensão (número de variáveis)  $D$  do problema. Já o número de avaliações da função-objetivo varia linearmente tanto com a dimensão  $D$  do problema quanto com o tamanho máximo da população  $maxAB$ .

Como visto anteriormente, as etapas de geração da população inicial, aplicação do operador de hipermutação e cálculo da métrica de afinidade entre células também realizam operações que variam seu tempo computacional linearmente com a dimensão  $D$  do problema sendo tratado. Diante disso e dos resultados apresentados nas Expressões 3.19 e 3.20, pode-se concluir que, de modo geral, o custo computacional total do algoritmo *cob-aiNet* proposto neste trabalho também varia linearmente em função do número de variáveis  $D$  do problema.

### 3.5 Considerações Finais

Este capítulo teve, como objetivo principal, posicionar o algoritmo *cob-aiNet* (*Concentration-based Artificial Immune Network*) frente a outras propostas da literatura e apresentá-lo em detalhes. Para isso, na etapa de posicionamento, foram brevemente discutidas outras abordagens que buscam manter a diversidade dos indivíduos na população, dentro do contexto de meta-heurísticas populacionais. Além disso, também foi apresentada aqui uma visão geral das técnicas exatas de otimização, utilizadas principalmente em matemática e pesquisa operacional, ressaltando-se em quais aspectos elas diferem do algoritmo proposto.

O detalhamento do algoritmo *cob-aiNet*, propriamente dito, iniciou-se com a apresentação do modelo de rede imunológica artificial de Bersini (2002), que foi o modelo tomado como ponto de partida para a proposta da ferramenta discutida neste capítulo. Em seguida, foram detalhadas as modificações feitas neste modelo para adaptá-lo ao contexto de otimização em espaços contínuos e discutidos os demais mecanismos e operadores propostos na elaboração do algoritmo *cob-aiNet*. Por fim, uma estimativa geral da variação do custo computacional do algoritmo em função de seus parâmetros e da dimensão do problema de otimização sendo tratado foi apresentada.

No próximo capítulo, serão apresentados e discutidos diversos experimentos computacionais fei-

---

tos com o intuito de avaliar os principais aspectos do algoritmo cob-aiNet, além de seu desempenho geral frente a outras abordagens da literatura.



## Capítulo 4

### Resultados Experimentais – *cob-aiNet*

Para que fosse possível avaliar cada aspecto do algoritmo *cob-aiNet*, foi feita uma série de experimentos que tiveram como base o conjunto de problemas de teste proposto para a Competição de Otimização de Parâmetros Reais, da *2005 CEC Special Session on Real-Parameter Optimization*, que ocorreu no *IEEE Congress on Evolutionary Computation (CEC)* de 2005 (Suganthan et al., 2005). A escolha de tal conjunto de problemas se deu por quatro razões principais: (i) por ter sido proposto para avaliar algoritmos em uma competição organizada dentro de um dos principais congressos da área de computação evolutiva, Suganthan et al. (2005) tiveram o cuidado de elaborar um conjunto de problemas que envolvesse uma ampla gama de características que possam vir a surgir em situações reais de otimização (vide Tabela 4.1), reduzindo-se assim a possibilidade de favorecimento de alguma classe específica de ferramentas; (ii) por se tratar de uma competição, a metodologia a ser seguida por cada algoritmo foi claramente definida, o que torna possível a comparação direta entre novas propostas de algoritmos e qualquer um dos competidores, sem que seja necessário reimplementar as ferramentas já existentes; (iii) apesar de se tratar de uma competição que ocorreu em 2005, alguns dos algoritmos que apresentaram os melhores resultados ainda são considerados estado-da-arte, o que os torna bons representantes dos paradigmas em que se baseiam; e (iv) como todos os problemas deste conjunto são *escalonáveis*, eles podem ser utilizados tanto em pequenas quanto em altas dimensões, o que dá ao usuário uma maior flexibilidade ao elaborar experimentos baseados nestes problemas. É importante ressaltar aqui que, embora todos os problemas de otimização considerados nos experimentos sejam artificiais, as ferramentas propostas nesta tese também são aplicáveis a problemas reais.

Este conjunto de testes da competição CEC'2005 é composto de 25 problemas mono-objetivo contínuos que, de maneira geral, podem apresentar as seguintes características: mono ou multimodalidade; alta quantidade de ótimos locais no espaço de busca; presença ou não de ruído no valor da função-objetivo; dependência ou não entre as variáveis; presença de *plateaus* no espaço de busca; ótimos globais localizados na fronteira; não-diferenciabilidade em alguns pontos do espaço de busca;

e até mesmo combinações de vários destes aspectos em um mesmo problema. Um resumo com as características principais de cada um destes problemas pode ser visto na Tabela 4.1, enquanto que uma descrição completa sobre cada um deles pode ser encontrada no Apêndice A. Para maiores detalhes sobre a implementação computacional deste conjunto de teste, sugere-se consultar o relatório técnico de Suganthan et al. (2005).

Tab. 4.1: Resumo das principais características de cada problema da competição CEC’2005. Foram consideradas aqui apenas as características de multimodalidade, presença de ruído na função-objetivo, separabilidade das variáveis, presença de ótimo global na fronteira e não diferenciabilidade em alguns pontos do espaço de busca. Maiores detalhes sobre cada problema podem ser encontrados no Apêndice A. Nesta tabela, “X” indica a presença de uma característica, enquanto que “-” indica sua ausência.

	Multimodalidade	Ruído	Separabilidade	Ótimo na Fronteira	Não Diferenciabilidade
F <sub>1</sub>	-	-	X	-	-
F <sub>2</sub>	-	-	-	-	-
F <sub>3</sub>	-	-	-	-	-
F <sub>4</sub>	-	X	-	-	-
F <sub>5</sub>	-	-	-	X	-
F <sub>6</sub>	X	-	-	-	-
F <sub>7</sub>	X	-	-	-	-
F <sub>8</sub>	X	-	-	X	-
F <sub>9</sub>	X	-	X	-	-
F <sub>10</sub>	X	-	-	-	-
F <sub>11</sub>	X	-	-	-	X
F <sub>12</sub>	X	-	-	-	-
F <sub>13</sub>	X	-	-	-	-
F <sub>14</sub>	X	-	-	-	-
F <sub>15</sub>	X	-	X (próx. ao ótimo global)	-	-
F <sub>16</sub>	X	-	X (próx. ao ótimo global)	-	-
F <sub>17</sub>	X	X	X (próx. ao ótimo global)	-	-
F <sub>18</sub>	X	-	-	-	-
F <sub>19</sub>	X	-	-	-	-
F <sub>20</sub>	X	-	-	X	-
F <sub>21</sub>	X	-	-	-	-
F <sub>22</sub>	X	-	-	X	-
F <sub>23</sub>	X	-	-	-	X
F <sub>24</sub>	X	-	-	-	X
F <sub>25</sub>	X	-	-	-	X

Para evitar que eventuais problemas de implementação pudessem comprometer os resultados das simulações, foi utilizada aqui a versão em Matlab da implementação original do conjunto de problemas da competição CEC’2005, disponibilizada no seguinte endereço (último acesso em 29 de abril de 2011):

[http://www3.ntu.edu.sg/home/EPNSugan/index\\_files/CEC-05/CEC05.htm](http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm)

Os estudos empíricos que foram feitos sobre o algoritmo cob-aiNet se dividiram em três etapas. Na primeira delas, apresentada na Seção 4.1, o objetivo foi avaliar a principal característica do algoritmo proposto, que é a capacidade de manutenção de diversidade e a consequente possibilidade de obtenção de múltiplos ótimos locais. Para isso, foram selecionados seis problemas da competição CEC'2005, com características diversas entre si, e o algoritmo cob-aiNet foi aplicado a tais problemas em espaços de busca bidimensionais. Como a localização exata de cada ótimo local de cada problema não é conhecida *a priori*, a única maneira de se avaliar tanto a distribuição final dos indivíduos no espaço de busca quanto o seu posicionamento relativo aos ótimos locais de cada problema é visualmente, o que justifica a escolha de espaços de busca no  $\mathbb{R}^2$ . Nesta etapa, a cob-aiNet foi comparada a dois outros SIAs da literatura, também inspirados tanto no princípio de Seleção Clonal (Burnet, 1978) quanto na teoria da Rede Imunológica (Jerne, 1974): opt-aiNet (de Castro & Timmis, 2002a), previamente discutida na Seção 2.2.2, e dopt-aiNet (de França et al., 2005), que será brevemente apresentada na Seção 4.1.1. Além da questão da diversidade, nesta primeira parte dos experimentos também foram avaliados os tempos computacionais gastos por cada um dos algoritmos.

Na segunda parte dos experimentos, discutida na Seção 4.2, foi feita uma análise de sensibilidade para o algoritmo cob-aiNet, com o intuito de se avaliar como se comporta a ferramenta proposta diante de diferentes ajustes de parâmetros para um mesmo problema. Além disso, buscou-se identificar quais parâmetros causam o maior impacto no desempenho da ferramenta quando incorretamente ajustados. Para isso, todos os estudos foram feitos com o problema  $F_9$  (vide Seção A.1.2.4 no Apêndice A), que apresenta uma grande quantidade de ótimos locais, também no espaço bidimensional.

Por fim, na última etapa dos experimentos, apresentada na Seção 4.3, buscou-se avaliar o desempenho da cob-aiNet em espaços de maior dimensão e também compará-la a outras ferramentas da literatura, baseadas em diversos paradigmas. Para isso, foram utilizados a mesma metodologia e todos os 25 problemas da competição CEC'2005, tanto em  $\mathbb{R}^{10}$  quanto em  $\mathbb{R}^{30}$ , o que permitiu a comparação direta entre a cob-aiNet e todos os demais algoritmos participantes da competição.

## 4.1 Análise da Distribuição Final da População no Espaço de Busca

Como mencionado anteriormente, nesta primeira etapa de estudos experimentais sobre o algoritmo cob-aiNet buscou-se avaliar a efetividade dos novos mecanismos de manutenção de diversidade. Para isso, foram escolhidos seis problemas de teste da competição CEC'2005, que possuem características diversas entre si e que permitem que se tenha uma visão geral de como a cob-aiNet se comporta em relação à manutenção de diversidade e à distribuição final das soluções pelo espaço de busca. Os problemas escolhidos aqui foram os seguintes:

- **F<sub>2</sub> - Problema de Schwefel 1.2 deslocado:** unimodal (vide Figura A.1(b));
- **F<sub>4</sub> - Problema de Schwefel 1.2 deslocado com ruído no *fitness*:** idêntico a  $F_2$ , mas com a presença de ruído (vide Figura A.1(d));
- **F<sub>9</sub> - Função de Rastrigin deslocada:** multimodal, com um alto número de ótimos locais (vide Figura A.1(i));
- **F<sub>12</sub> - Problema de Schwefel 2.13:** multimodal (vide Figura A.1(l));
- **F<sub>15</sub> - Função composta híbrida:** multimodal, com um alto número de ótimos locais e dois *plateaus* (vide Figura A.2(c));
- **F<sub>20</sub> - Função composta híbrida rotacionada com ótimo global na fronteira:** multimodal, com um alto número de ótimos locais, dois *plateaus* e ótimo global na fronteira do espaço de busca (vide Figura A.2(h)).

Para que se pudesse ter uma base de comparação para os resultados obtidos pela *cob-aiNet*, nesta seção foram utilizados também dois SIAs da literatura que adotam mecanismos semelhantes aos da *cob-aiNet*, ou seja, que utilizam codificação real de soluções e possuem operadores inspirados tanto no princípio da Seleção Clonal (Burnet, 1978) quanto na teoria da Rede Imunológica (Jerne, 1974). Sendo assim, foram escolhidos os algoritmos *opt-aiNet* (de Castro & Timmis, 2002a), previamente descrito na Seção 2.2.2, e *dopt-aiNet* (de França et al., 2005), que será brevemente discutido a seguir. Além disso, para se avaliar o quão importante é o mecanismo de busca local empregado na *cob-aiNet* (vide Seção 3.3.4), tanto para a convergência das soluções quanto para a capacidade de manutenção de diversidade, nesta seção também foram feitos experimentos com o algoritmo *cob-aiNet* sem a utilização deste operador.

#### 4.1.1 O Algoritmo *dopt-aiNet*

Ao tentar aplicar o algoritmo *opt-aiNet* (de Castro & Timmis, 2002a) a problemas de otimização em ambientes dinâmicos, ou seja, a problemas com funções-objetivo cujos ótimos locais e globais variam com o tempo, de França et al. (2005) observaram uma série de limitações em alguns dos mecanismos da proposta original de de Castro & Timmis (2002a), sendo que algumas delas já foram inclusive discutidas na Seção 2.2.2. Diante disso, com o intuito de amenizar estes problemas e criar uma ferramenta capaz de reagir rapidamente a mudanças no ambiente, de França et al. (2005) propuseram o algoritmo *dopt-aiNet* (*Artificial Immune Network for Dynamic Optimization*).

Como na *opt-aiNet* e na *cob-aiNet*, as soluções candidatas na *dopt-aiNet* são representadas como vetores de atributos reais, gerados inicialmente de maneira aleatória e com distribuição uniforme

dentro do domínio do problema. Além disso, a estrutura geral da dopt-aiNet é semelhante à da opt-aiNet (vide Seção 2.2.2), já que ambos os algoritmos possuem etapas de clonagem, hipermutação, seleção e supressão. No entanto, algumas diferenças importantes estão presentes, dentre as quais podemos ressaltar a inclusão de novos operadores de mutação e a utilização de uma nova métrica de distância na etapa de supressão, proposta com o intuito de deixar o algoritmo menos sensível a ajustes inadequados do limiar  $\sigma_s$ .

Com relação aos novos operadores de mutação, dada a dificuldade de ajuste do parâmetro  $\beta$  do operador original da opt-aiNet (vide Equação 2.1), que é de responsabilidade do usuário, de França et al. (2005) propuseram um mecanismo de ajuste automático deste parâmetro, baseado na estratégia de busca linear conhecida como *seção áurea* (Bazaraa et al., 1993). Apesar de possuir a grande vantagem de tirar a responsabilidade de ajuste de  $\beta$  do usuário, este novo operador demanda um maior número de avaliações da função-objetivo a cada iteração do algoritmo, já que, para cada valor de  $\beta$  obtido pela busca linear, uma série de novos indivíduos devem ser criados com o operador de mutação para que seja possível avaliar sua qualidade.

Além desta mudança no cálculo de  $\beta$ , de França et al. (2005) também identificaram a possível desvantagem associada ao fato de o operador de mutação da opt-aiNet alterar todas as variáveis de cada indivíduo ao mesmo tempo (vide Seção 3.3.4). Diante disso, um segundo operador de mutação foi introduzido (chamado de *mutação unidimensional*), que gera uma nova população de indivíduos de tamanho igual ao número de variáveis do problema e aplica uma variação gaussiana a uma única variável de cada um destes novos indivíduos. Ou seja, cada nova solução é alterada pelo operador de mutação unidimensional ao longo de uma única dimensão.

Um terceiro operador de mutação também foi proposto no algoritmo dopt-aiNet, chamado de *duplicação gênica*. Basicamente, o operador de duplicação gênica escolhe aleatoriamente o valor presente em uma das variáveis de uma solução-candidata e o replica para as demais variáveis iterativamente, mantendo a alteração sempre que o novo indivíduo gerado for melhor que o anterior. Apesar de este operador acelerar a convergência da população, ele também possui duas desvantagens principais, identificadas por Coelho et al. (2010): (i) como cada variável geralmente pertence a um domínio específico, a replicação do valor de uma variável para as demais não é direta, sendo necessária uma adequação ao domínio de cada nova variável, o que aumenta o custo computacional e pode levar a interpretações distintas de como isso deve ser feito; e (ii) cada replicação do valor escolhido para as demais variáveis exige uma nova avaliação da função-objetivo, o que aumenta significativamente o custo computacional por iteração do algoritmo.

Outra modificação introduzida pela dopt-aiNet foi a substituição da distância euclidiana, na etapa de supressão, por outro operador denominado *distância de linha*. A motivação para este novo operador está justamente no fato de que o uso direto da distância euclidiana pode levar a situações em

que células posicionadas em picos distintos podem ser suprimidas caso o limiar de supressão  $\sigma_s$  não esteja corretamente ajustado, como foi previamente discutido e ilustrado na Seção 3.3.3.

O cálculo da distância de linha entre dois pontos  $P_1$  e  $P_2$  se dá basicamente através da criação de um segmento de reta entre estes dois pontos e da identificação de  $P$ , que corresponde ao ponto, na superfície do problema, mais próximo do ponto médio  $P_b$  da linha criada, como ilustrado na Figura 4.1. Identificados  $P$  e  $P_b$ , a distância de linha entre os pontos  $P_1$  e  $P_2$  será então equivalente à distância euclidiana entre  $P$  e  $P_b$ , o que nada mais é que distância entre uma aproximação simples de função utilizando dois pontos e a superfície real do problema. Apesar desta nova métrica de distância ser capaz de indicar mais efetivamente quando dois indivíduos estão em picos/vales diferentes do problema, novamente a principal desvantagem é o impacto negativo no custo computacional total do algoritmo, já que mais avaliações da função-objetivo se tornam necessárias para a determinação de  $P$ ,  $P_b$  e a distância entre eles.

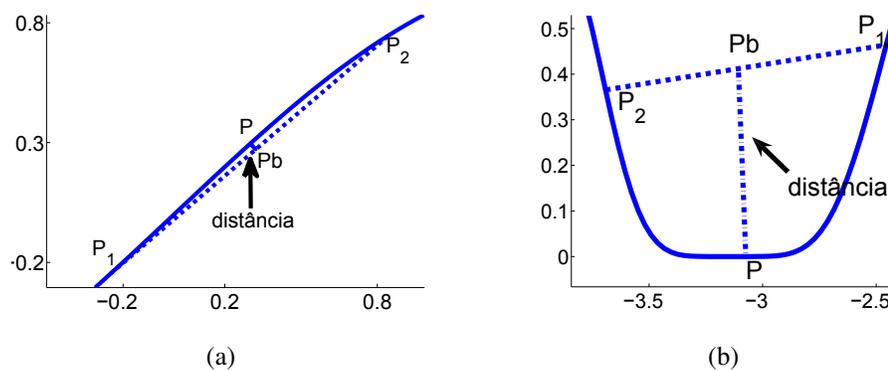


Fig. 4.1: Dois exemplos de distância de linha entre os pontos  $P_1$  e  $P_2$  no espaço de busca.  $P_b$  é o ponto médio da linha formada por  $P_1$  e  $P_2$ , e  $P$  é o ponto mais próximo de  $P_b$  na superfície real da função-objetivo. Em (a),  $P_1$  e  $P_2$  estão no mesmo pico, o que leva a uma distância de linha baixa, enquanto que em (b), eles estão em picos diferentes, o que resulta em um maior valor da distância de linha.

Por fim, a última modificação da dopt-aiNet em relação à opt-aiNet está no uso do conceito de *múltiplas populações*. Como a dopt-aiNet foi proposta para tratar cenários em constante modificação, foi introduzida uma população de *memória*, que armazena todas as células da população que já convergiram (não podem mais ser melhoradas) para uso futuro, o que é especialmente importante em situações em que a função-objetivo varia ciclicamente.

Apesar de ter sido proposta originalmente visando tratar problemas dinâmicos de otimização contínua, a dopt-aiNet também foi aplicada com sucesso em problemas estáticos (de França et al., 2005), o que motivou o seu uso nas comparações com a cob-aiNet.

### 4.1.2 Metodologia Experimental

Nesta primeira etapa dos experimentos, a execução de cada algoritmo foi repetida 25 vezes para cada problema e, em cada repetição, cada algoritmo foi executado até atingir  $4 \cdot 10^4$  avaliações da função-objetivo ou obter um erro absoluto médio entre a melhor solução-candidata e o ótimo global de cada problema inferior a  $1 \cdot 10^{-9}$ . Os parâmetros para cada algoritmo foram ajustados empiricamente em experimentos preliminares e, no caso de operadores idênticos presentes em dois ou mais algoritmos, valores iguais foram atribuídos a todos os parâmetros associados a eles. Sendo assim, os parâmetros finais da dopt-aiNet foram: tamanho da população inicial  $nAB = 50$ , tamanho máximo da população  $maxAB = 100$ , número de clones  $nC = 3$ , limiar de supressão (para distância de linha)  $\sigma_s = 5$ , número de iterações entre operações de duplicação gênica  $n_{gene} = 10$ , e  $rank = 100$ . Os parâmetros adotados para a opt-aiNet foram: tamanho inicial da população  $nAB = 50$ , número de clones  $nC = 10$ ,  $\beta = 1$ , número de iterações entre etapas de supressão  $n_{supp} = 10$ , e porcentagem de novas células que devem ser inseridas após a supressão  $d = 40\%$ . Por fim, a cob-aiNet foi executada com tamanho inicial da população  $nAB = 50$ , tamanho máximo da população  $maxAB = 100$ , números mínimo e máximo de clones  $nC^{min} = 2$  e  $nC^{max} = 10$ ,  $\beta^i = 2$ ,  $\beta^f = 0,01$ , e o operador de busca local foi executado por  $LS_{it} = 5$  iterações a cada  $LS_{freq} = 5$  iterações do *loop* principal, com intervalo de busca inicial  $SR = \sigma_s$ . A versão da cob-aiNet sem busca local foi executada com os mesmos parâmetros da cob-aiNet original, em todos os problemas tratados aqui.

Os limiares de supressão tanto para as duas versões da cob-aiNet quanto para a opt-aiNet, para cada problema, estão representados na Tabela 4.2. Nestes três algoritmos, a distância entre células da população foi avaliada pela distância euclidiana normalizada pela maior distância possível no espaço de busca, o que facilita a definição dos limiares de supressão já que tais parâmetros passam a corresponder a porcentagens desta maior distância, devendo sempre ser definidos no intervalo  $[0, 1]$ .

Tab. 4.2: Limiar de supressão  $\sigma_s$  usado nas duas versões da cob-aiNet e na opt-aiNet.

Problema	F <sub>2</sub>	F <sub>4</sub>	F <sub>9</sub>	F <sub>12</sub>	F <sub>15</sub>	F <sub>20</sub>
$\sigma_s$	0,20	0,50	0,05	0,03	0,05	0,07

Nesta primeira parte dos experimentos, foram avaliados tanto o erro absoluto médio (EAM) entre o valor da função-objetivo da melhor solução encontrada e o ótimo global de cada problema, nas 25 repetições, quanto o número médio final de soluções retornadas por cada algoritmo, além do posicionamento entre cada solução retornada e o ótimo local mais próximo do problema (feito visualmente). Para verificar a significância estatística dos resultados obtidos, o *Wilcoxon's Rank Sum test* (Moore et al., 2007) foi aplicado, com um limiar 0,05, entre os resultados obtidos pela cob-aiNet e os resultados individuais de cada um dos demais algoritmos.

O *Wilcoxon's Rank Sum test* (Moore et al., 2007) é um teste estatístico não-paramétrico que se baseia na *ordenação conjunta* dos valores presentes nos dois conjuntos de amostras que devem ser avaliados. Após a ordenação, as posições, ou seja, os *ranks* de cada amostra de cada conjunto nesta lista ordenada são *somados* e as somas comparadas. Caso a soma  $W$  dos *ranks* das amostras da primeira distribuição esteja muito distante da média dos *ranks* (considerando-se as diferenças nos números de amostras nos dois conjuntos), o *Wilcoxon's Rank Sum test* rejeita sua hipótese nula, que propõe que as amostras presentes nos dois conjuntos sendo avaliados foram extraídas a partir de distribuições idênticas. Nas tabelas que se seguirão, os resultados para os quais a hipótese nula do teste de Wilcoxon pôde ser rejeitada com significância 0,05 foram marcados com \*.

### 4.1.3 Apresentação dos Resultados e Discussão

Considerando-se primeiramente os erros absolutos médios (EAM) entre o ótimo global de cada problema e a melhor solução encontrada por cada algoritmo imunoinspirado ao longo das 25 repetições, dados na Tabela 4.3, é possível observar que o algoritmo *cob-aiNet* obteve os menores valores em todos os problemas estudados, e também que a diferença de desempenho entre a *cob-aiNet* e os demais algoritmos foi estatisticamente significativa em todos os casos.

Tab. 4.3: Média  $\pm$  Desvio Padrão dos erros absolutos entre as melhores soluções obtidas por cada algoritmo e o ótimo global de cada problema, avaliados em 25 repetições. Os melhores resultados estão em negrito e aqueles para os quais a hipótese nula do teste de Wilcoxon (quando comparados com os da *cob-aiNet*) pode ser rejeitada com significância 0,05 estão marcados com \*.

	<b>cob-aiNet</b>	cob-aiNet sem Busca Local	opt-aiNet	dopt-aiNet
<b>F<sub>2</sub></b>	<b><math>1,91 \cdot 10^{-11} \pm 2,65 \cdot 10^{-11}</math></b>	$2,78 \cdot 10^{-9} \pm 2,70 \cdot 10^{-9} *$	$6,25 \cdot 10^{-5} \pm 6,74 \cdot 10^{-5} *$	$2,32 \cdot 10^1 \pm 3,95 \cdot 10^1 *$
<b>F<sub>4</sub></b>	<b><math>3,48 \cdot 10^{-11} \pm 8,53 \cdot 10^{-11}</math></b>	$3,64 \cdot 10^{-9} \pm 3,53 \cdot 10^{-9} *$	$3,06 \cdot 10^{-5} \pm 2,40 \cdot 10^{-5} *$	$1,19 \cdot 10^2 \pm 1,69 \cdot 10^2 *$
<b>F<sub>9</sub></b>	<b><math>3,38 \cdot 10^{-10} \pm 1,08 \cdot 10^{-9}</math></b>	$1,10 \cdot 10^{-3} \pm 5,04 \cdot 10^{-3} *$	$3,75 \cdot 10^{-2} \pm 3,08 \cdot 10^{-2} *$	$5,93 \cdot 10^{-1} \pm 4,84 \cdot 10^{-1} *$
<b>F<sub>12</sub></b>	<b><math>9,67 \cdot 10^{-12} \pm 1,04 \cdot 10^{-11}</math></b>	$4,19 \cdot 10^{-2} \pm 4,80 \cdot 10^{-2} *$	$5,14 \cdot 10^{-2} \pm 5,36 \cdot 10^{-2} *$	$7,35 \cdot 10^{-1} \pm 6,92 \cdot 10^{-1} *$
<b>F<sub>15</sub></b>	<b><math>1,01 \cdot 10^{-11} \pm 1,41 \cdot 10^{-11}</math></b>	$1,54 \cdot 10^{-3} \pm 1,75 \cdot 10^{-3} *$	$6,14 \cdot 10^{-1} \pm 3,96 \cdot 10^{-1} *$	$7,81 \cdot 10^0 \pm 1,22 \cdot 10^1 *$
<b>F<sub>20</sub></b>	<b><math>7,13 \cdot 10^{-11} \pm 1,06 \cdot 10^{-10}</math></b>	$1,02 \cdot 10^{-2} \pm 8,53 \cdot 10^{-3} *$	$3,33 \cdot 10^{-1} \pm 2,86 \cdot 10^{-1} *$	$1,33 \cdot 10^2 \pm 6,94 \cdot 10^1 *$

Além disso, observando-se o número final de indivíduos retornados por cada algoritmo, apresentados na Tabela 4.4, é possível perceber sinais de que a *cob-aiNet* também possui mecanismos mais eficientes de controle de diversidade e ajuste do tamanho da população, já que ela foi o único algoritmo que convergiu para uma única solução em todas as repetições dos problemas monomodais ( $F_2$  e  $F_4$ ). Esta característica foi observada tanto na versão original da *cob-aiNet*, com busca local, quanto em sua versão com este operador desabilitado.

Tab. 4.4: Média  $\pm$  Desvio Padrão do número final de soluções retornadas por cada algoritmo em 25 repetições para cada problema. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados com os da cob-aiNet) pode ser rejeitada com significância 0,05 estão marcados com \*.

	cob-aiNet	cob-aiNet sem Busca Local	opt-aiNet	dopt-aiNet
<b>F<sub>2</sub></b>	1,00 $\pm$ 0,00	1,00 $\pm$ 0,00	4,40 $\pm$ 1,12 *	2,64 $\pm$ 1,11 *
<b>F<sub>4</sub></b>	1,00 $\pm$ 0,00	1,00 $\pm$ 0,00	1,56 $\pm$ 0,51 *	2,84 $\pm$ 1,21 *
<b>F<sub>9</sub></b>	91,32 $\pm$ 4,11	98,52 $\pm$ 2,87 *	25,80 $\pm$ 4,06 *	4,00 $\pm$ 0,00 *
<b>F<sub>12</sub></b>	9,56 $\pm$ 1,61	167,64 $\pm$ 10,29 *	6,00 $\pm$ 1,29 *	8,96 $\pm$ 1,14
<b>F<sub>15</sub></b>	53,96 $\pm$ 2,81	65,04 $\pm$ 3,83 *	16,56 $\pm$ 3,16 *	21,08 $\pm$ 1,98 *
<b>F<sub>20</sub></b>	17,68 $\pm$ 2,75	17,44 $\pm$ 2,04	8,40 $\pm$ 1,71 *	11,56 $\pm$ 1,94 *

No entanto, o número final de indivíduos apresentado na Tabela 4.4 não permite, isoladamente, uma correta avaliação da capacidade de cobertura do espaço de busca de cada algoritmo para os problemas multimodais, já que não é possível saber qual a posição relativa entre cada uma destas soluções e os ótimos locais de cada problema. Sendo assim, para que seja possível avaliar este aspecto, as populações finais retornadas por cada algoritmo nas repetições 1, 12 e 25 dos problemas  $F_2$ – $F_{20}$ , em conjunto com as respectivas *curvas de nível* de cada problema, estão representadas nas Figuras 4.2–4.7. Por questões de espaço e para evitar quaisquer favorecimentos, foram escolhidas apenas as repetições inicial, final e intermediária para serem apresentadas aqui, já que elas permitem que se tenha uma boa visualização do comportamento geral de cada algoritmo. É importante ressaltar que todos os problemas tratados são de *minimização* e, quanto mais escuro for o tom de azul de uma curva de nível das Figuras 4.2–4.7, menor será o valor da função-objetivo naquela região, enquanto que quanto mais escuro o tom de vermelho da curva, maior o valor da função-objetivo.

Considerando-se primeiramente os problemas monomodais  $F_2$  e  $F_4$  (Figuras 4.2 e 4.3), é possível observar que as duas versões da cob-aiNet foram capazes de convergir para uma única solução em todas as repetições (vide Tabela 4.4), posicionadas praticamente no ótimo global de cada problema. Já a opt-aiNet e a dopt-aiNet não conseguiram convergir para uma única solução na maioria das repetições, sendo que a dopt-aiNet deixou inclusive de localizar o ótimo global em diversas situações como as ilustradas nas Figuras 4.2(j) e 4.3(k), o que resultou no alto EAM apresentado por este algoritmo na Tabela 4.3.

No entanto, é nos problemas multimodais que as diferenças entre os mecanismos de manutenção de diversidade e de adaptação do tamanho da população de cada algoritmo se tornam mais evidentes. Para o problema  $F_9$  (Figura 4.4), que é o que apresenta o maior número de ótimos locais, a quantidade de vales identificados pelas duas variações da cob-aiNet foi significativamente maior que a das demais ferramentas, sendo que ambos os algoritmos convergiram inclusive para um número de soluções

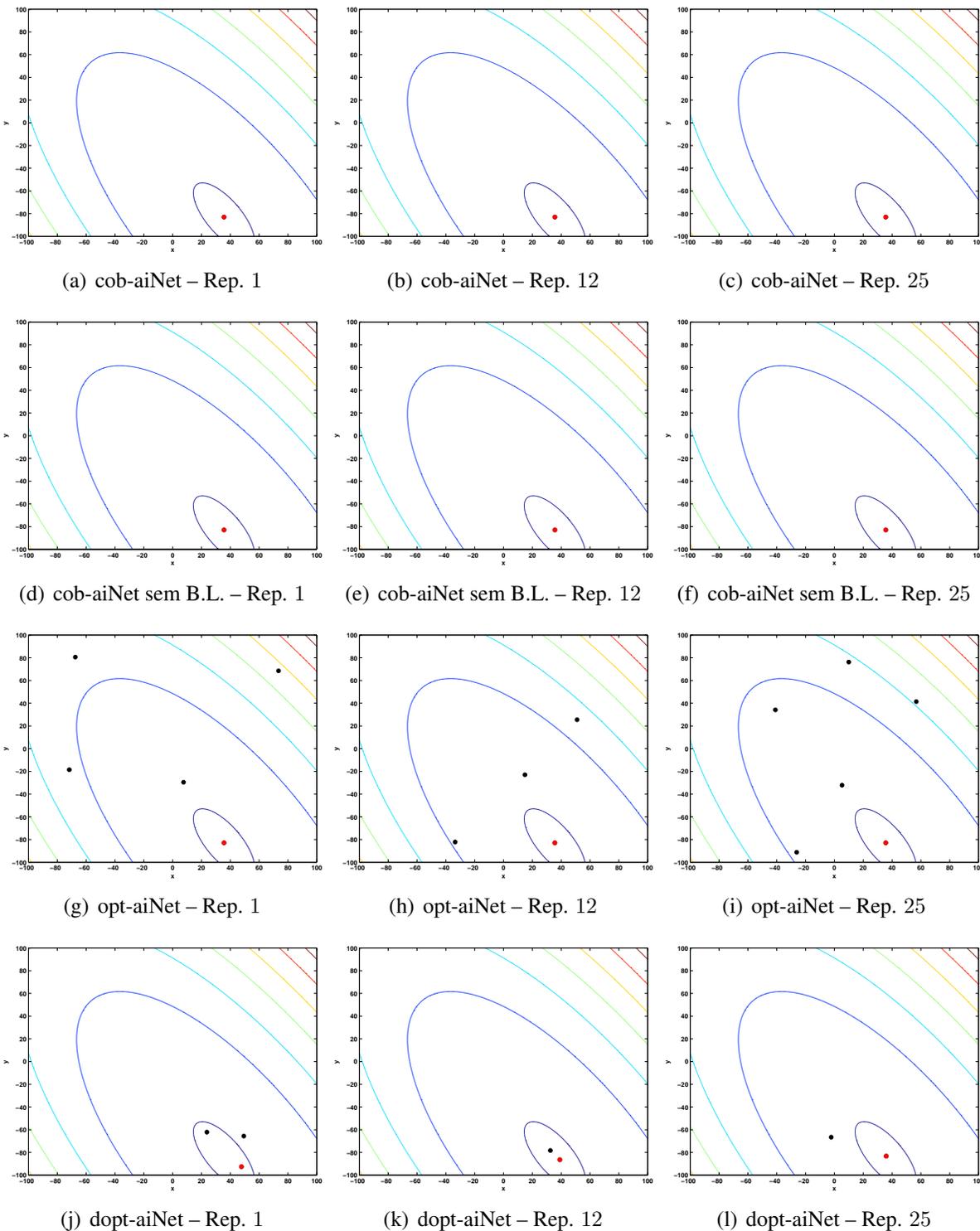


Fig. 4.2: Representação gráfica das curvas de nível do problema  $F_2$  em conjunto com as populações finais retornadas pelos algoritmos *cob-aiNet* [(a), (b) e (c)], *cob-aiNet* sem busca local [(d), (e) e (f)], *opt-aiNet* [(g), (h) e (i)] e *dopt-aiNet* [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

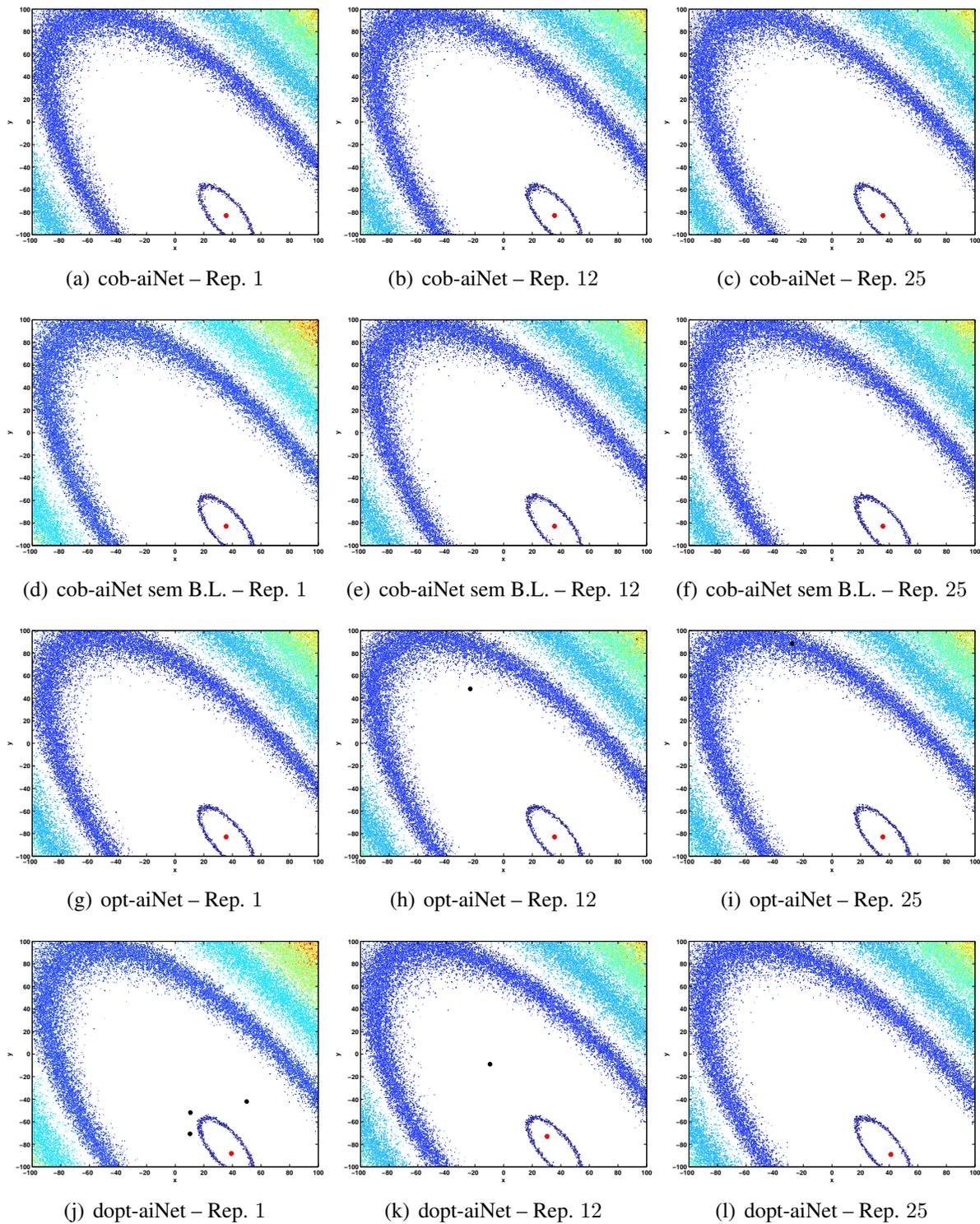


Fig. 4.3: Representação gráfica das curvas de nível do problema  $F_4$  em conjunto com as populações finais retornadas pelos algoritmos cob-aiNet [(a), (b) e (c)], cob-aiNet sem busca local [(d), (e) e (f)], opt-aiNet [(g), (h) e (i)] e dopt-aiNet [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

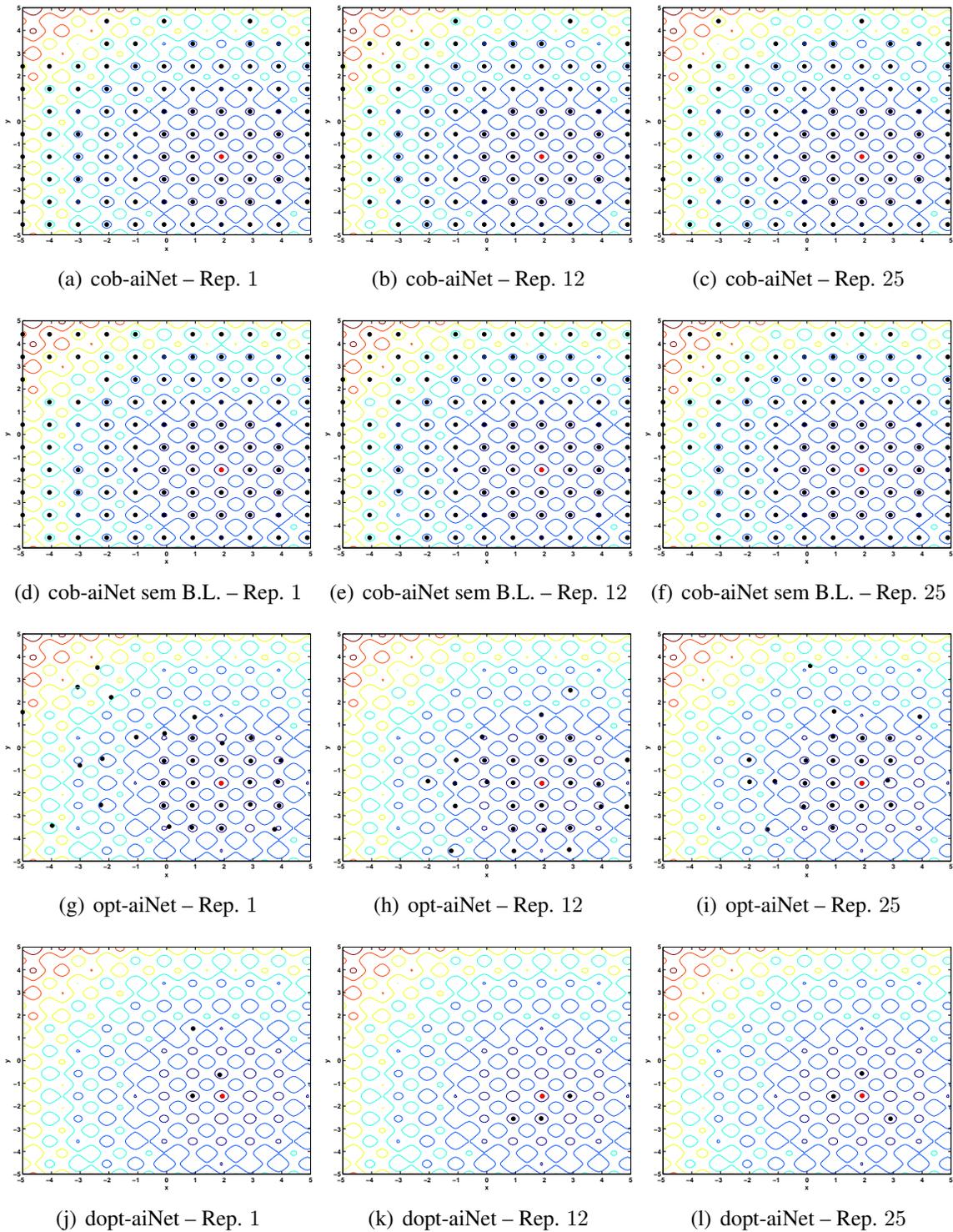


Fig. 4.4: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelos algoritmos *cob-aiNet* [(a), (b) e (c)], *cob-aiNet* sem busca local [(d), (e) e (f)], *opt-aiNet* [(g), (h) e (i)] e *dopt-aiNet* [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

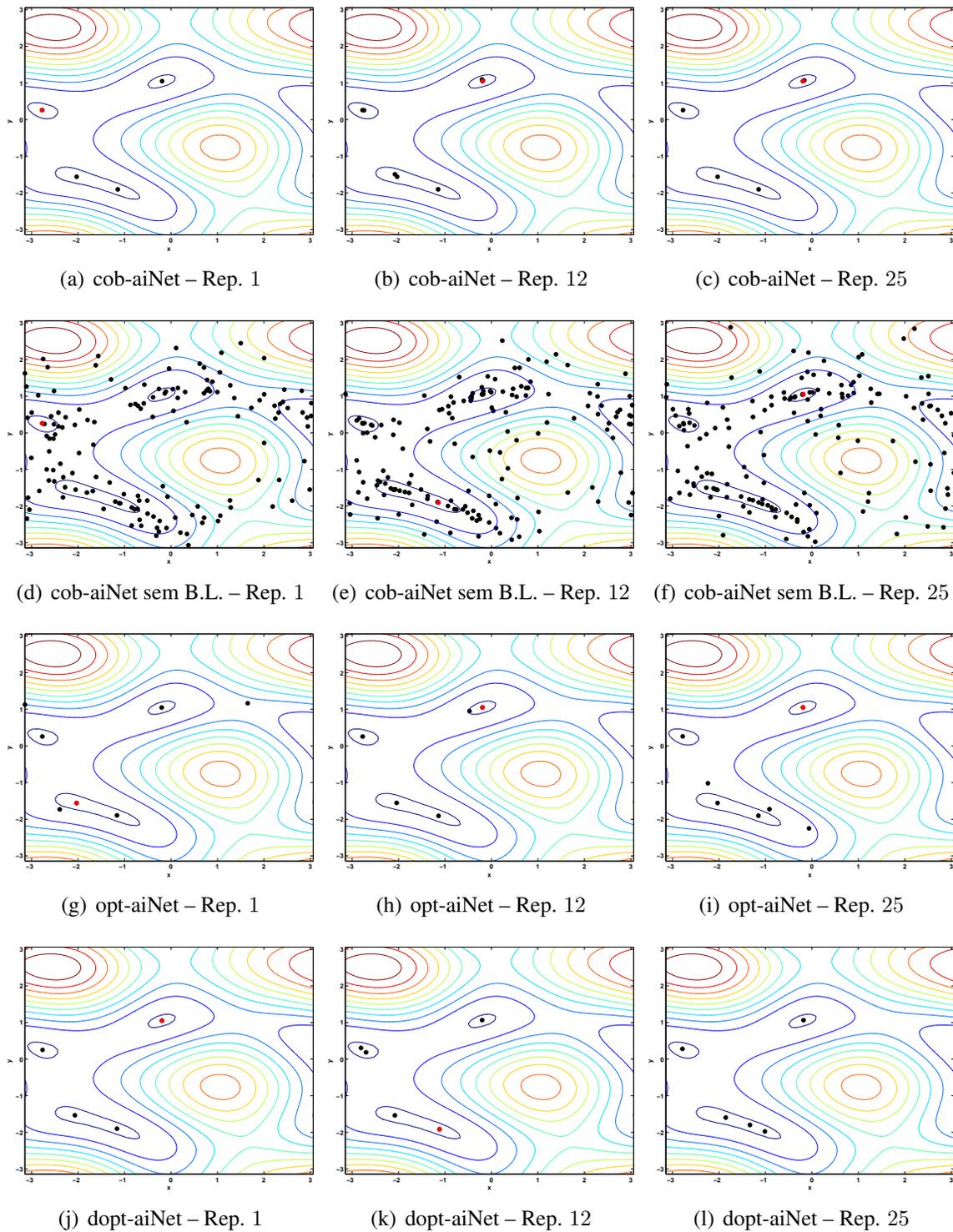


Fig. 4.5: Representação gráfica das curvas de nível do problema  $F_{12}$  em conjunto com as populações finais retornadas pelos algoritmos cob-aiNet [(a), (b) e (c)], cob-aiNet sem busca local [(d), (e) e (f)], opt-aiNet [(g), (h) e (i)] e dopt-aiNet [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

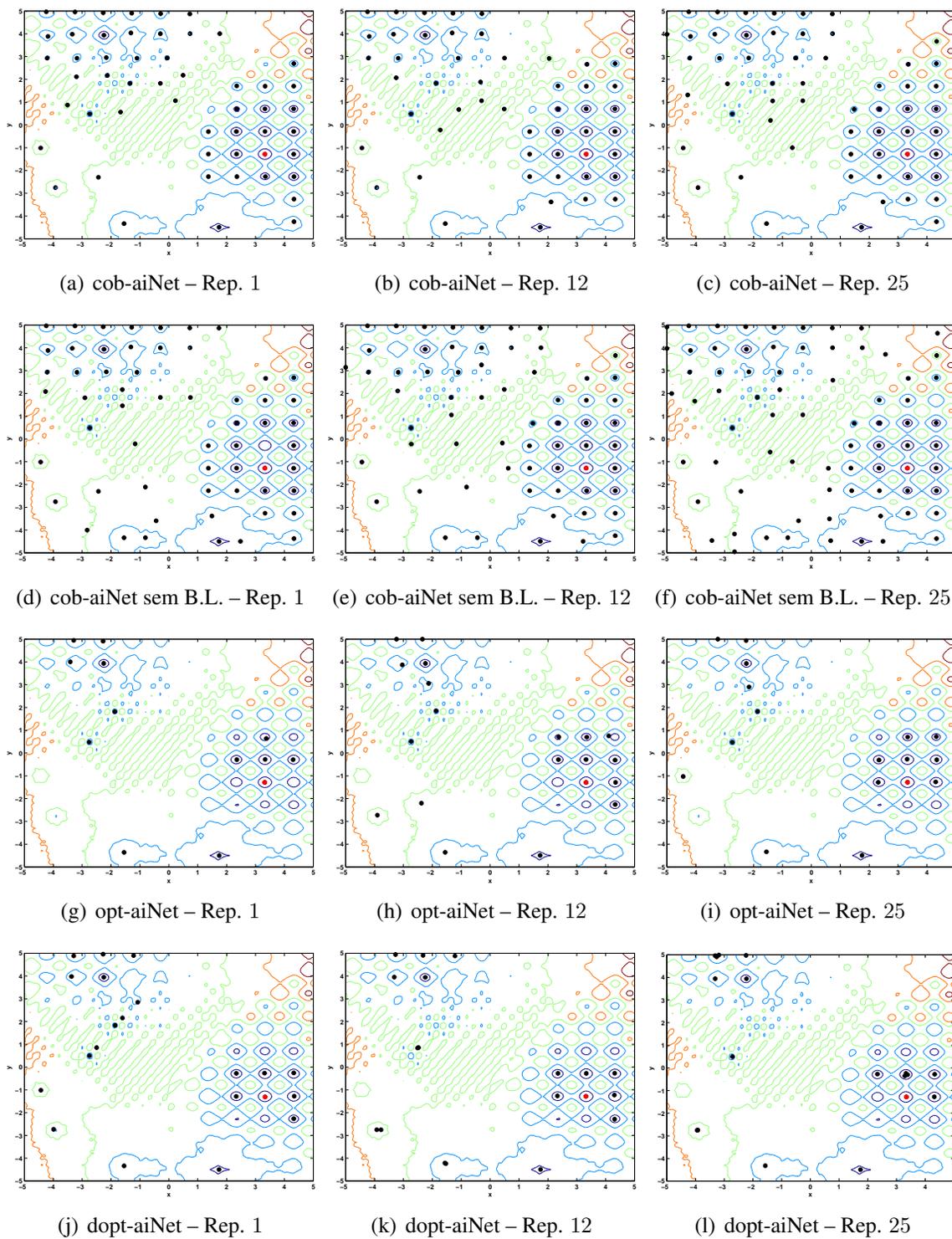


Fig. 4.6: Representação gráfica das curvas de nível do problema  $F_{15}$  em conjunto com as populações finais retornadas pelos algoritmos cob-aiNet [(a), (b) e (c)], cob-aiNet sem busca local [(d), (e) e (f)], opt-aiNet [(g), (h) e (i)] e dopt-aiNet [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

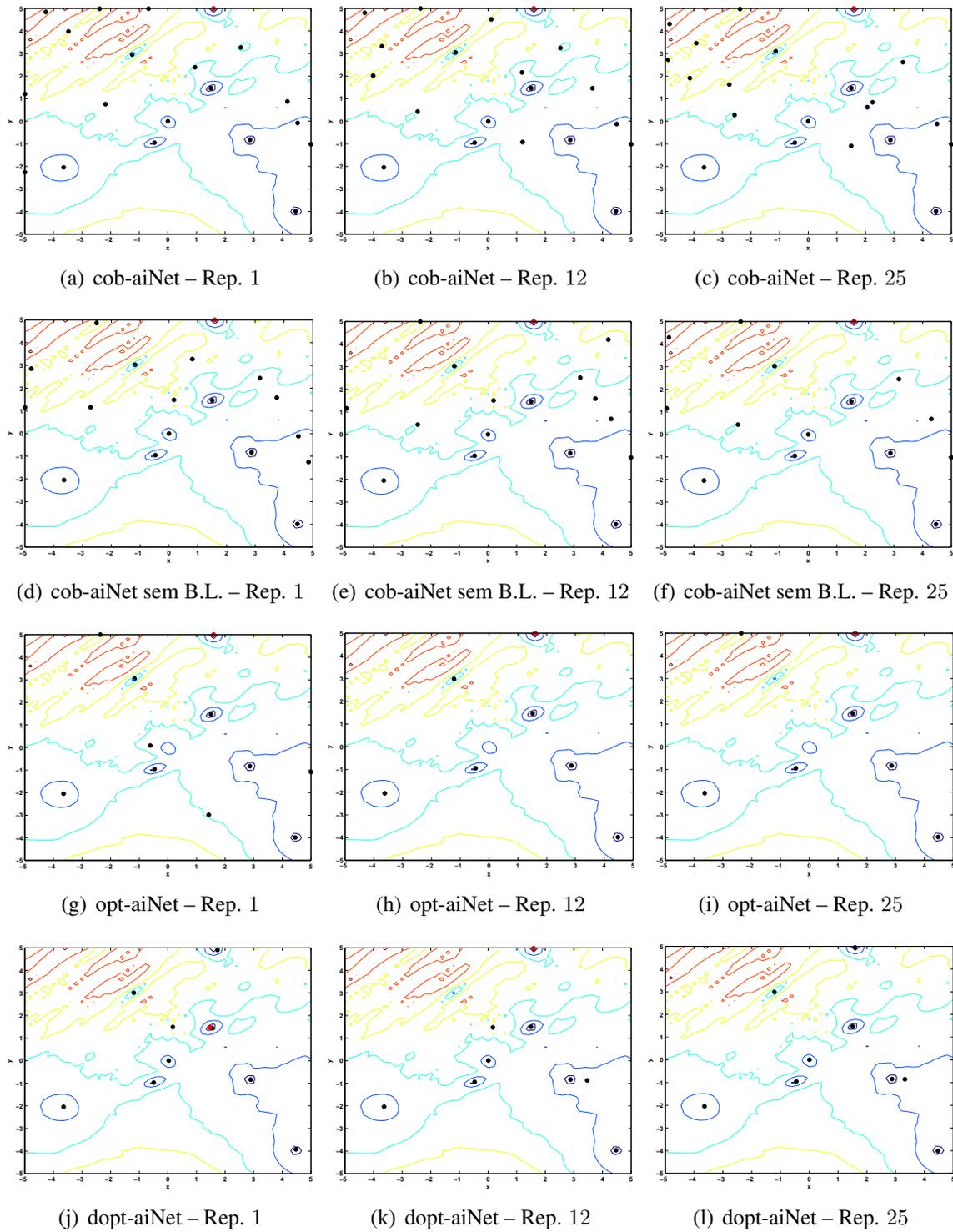


Fig. 4.7: Representação gráfica das curvas de nível do problema  $F_{20}$  em conjunto com as populações finais retornadas pelos algoritmos cob-aiNet [(a), (b) e (c)], cob-aiNet sem busca local [(d), (e) e (f)], opt-aiNet [(g), (h) e (i)] e dopt-aiNet [(j), (k) e (l)] nas repetições 1, 12 e 25 dos experimentos. As melhores soluções estão representadas em vermelho.

maior que o tamanho de suas populações definido inicialmente (vide Tabela 4.4). Além disso, é possível observar na Figura 4.4 que nem todas as soluções obtidas pela *opt-aiNet* e pela *dopt-aiNet* correspondem efetivamente a ótimos locais do problema, já que apesar de algumas delas estarem no vale correspondente a um ótimo local, elas se encontram deslocadas em relação à posição efetiva do ótimo (sempre centralizado em seu vale correspondente).

No problema  $F_9$ , é possível também começar a perceber com maior clareza o impacto que o operador de busca local tem no algoritmo *cob-aiNet*. Observando-se a Tabela 4.3, nota-se que este operador permitiu a obtenção de um EAM diversas ordens de grandeza menor que o obtido pela versão do algoritmo sem busca local. Isto evidencia a limitação de capacidade de ajuste fino presente em SIAs baseados em apenas um operador de mutação, que foi discutida na Seção 3.3.4. Já quanto à capacidade de manutenção de diversidade, é possível observar na Figura 4.4 que a versão da *cob-aiNet* sem busca local foi capaz de encontrar um número ligeiramente maior de ótimos locais do que o encontrado por sua versão completa, o que indica que o uso ou não do operador de busca local pode determinar a escolha entre a obtenção de um melhor ajuste fino das soluções ou a localização de um número maior de ótimos locais.

Com relação ao problema  $F_{12}$ , a versão da *cob-aiNet* sem busca local apresentou um comportamento completamente inesperado, ilustrado na Figura 4.5. Como é possível observar, esta variante da *cob-aiNet* retornou um número elevado de indivíduos ao final de todas as repetições deste problema, num comportamento bem diferente do observado para a *cob-aiNet* completa. Analisando-se estes resultados mais detalhadamente, é possível notar que, em média, nas populações finais da versão completa da *cob-aiNet* cerca de  $20 \pm 12\%$  das soluções retornadas possuem concentração inferior a 0,5, enquanto que  $11 \pm 9\%$  destas soluções possui concentração inferior a 0,1. Já na versão do algoritmo sem a busca local, estes valores são respectivamente  $62 \pm 5\%$  e  $39 \pm 6\%$ , o que evidencia que a dinâmica de concentrações dos indivíduos da população ainda não havia se estabilizado suficientemente no final das execuções. Diante disso, fica claro que, para este problema, o operador de busca local é essencial para acelerar a convergência das soluções, permitindo assim a estabilização do modelo e a supressão das soluções não-promissoras da população.

Já nos problemas  $F_{15}$  e  $F_{20}$ , que são os dois mais desafiadores dentre os estudados aqui, a mesma tendência observada nos demais problemas se manteve. A *cob-aiNet* com busca local conseguiu EAMs significativamente menores que os demais algoritmos, enquanto que seus mecanismos de manutenção de diversidade levaram à localização de uma maior quantidade de ótimos locais quando comparada à obtida pela *opt-aiNet* e pela *dopt-aiNet*. Considerando-se as duas versões da *cob-aiNet*, resultados equivalentes aos observados para  $F_9$  foram obtidos, com a versão sem busca local sendo capaz de encontrar um número um pouco maior de ótimos locais que a versão completa (exceto para  $F_{20}$ ), mas apresentando um EAM significativamente maior.

Tab. 4.5: Média  $\pm$  Desvio Padrão do número de iterações executadas por cada algoritmo a cada repetição. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados com os da cob-aiNet) pode ser rejeitada com significância 0,05 estão marcados com \*.

	cob-aiNet	cob-aiNet sem Busca Local	opt-aiNet	dopt-aiNet
<b>F<sub>2</sub></b>	37,60 $\pm$ 3,74	3646,36 $\pm$ 175,71 *	538,64 $\pm$ 36,51 *	26,00 $\pm$ 0,00 *
<b>F<sub>4</sub></b>	58,80 $\pm$ 45,26	3855,72 $\pm$ 145,20 *	1268,48 $\pm$ 32,20 *	26,00 $\pm$ 0,00 *
<b>F<sub>9</sub></b>	34,28 $\pm$ 2,19	44,52 $\pm$ 1,29 *	104,88 $\pm$ 8,60 *	25,04 $\pm$ 0,79 *
<b>F<sub>12</sub></b>	38,20 $\pm$ 2,53	30,00 $\pm$ 0,00 *	367,60 $\pm$ 6,75 *	24,00 $\pm$ 0,41 *
<b>F<sub>15</sub></b>	39,60 $\pm$ 4,21	56,96 $\pm$ 2,46 *	149,48 $\pm$ 9,76 *	20,32 $\pm$ 1,38 *
<b>F<sub>20</sub></b>	61,20 $\pm$ 4,20	163,48 $\pm$ 13,10 *	265,00 $\pm$ 14,91 *	24,08 $\pm$ 0,76 *

De maneira geral, os menores erros absolutos médios obtidos pela cob-aiNet se devem tanto ao novo mecanismo de variação dinâmica do parâmetro  $\beta$  de hipermutação quanto ao operador de busca local. Apesar de a maior contribuição neste aspecto ter sido introduzida pela busca local, o novo operador de hipermutação também levou à obtenção de menores erros para todos os problemas tratados, como pode ser visto na Tabela 4.3 ao se comparar os resultados da cob-aiNet sem busca local e da opt-aiNet, que adota o mesmo operador de mutação mas com o parâmetro  $\beta$  fixo. No entanto, com relação à capacidade de manutenção de diversidade e de adaptação do tamanho da população às características do problema, o novo modelo de rede imunológica proposto no algoritmo cob-aiNet é o principal responsável pelas grandes melhorias observadas, ficando o operador de busca local responsável apenas por contribuir para uma convergência mais rápida de cada solução para o ótimo local mais próximo, reduzindo-se assim a possibilidade de ocorrência de situações espúrias, como a observada no problema  $F_{12}$ .

Um outro aspecto que chama a atenção nestes experimentos é o fato da dopt-aiNet apresentar resultados significativamente piores que os das demais ferramentas, o que pode ser de certa forma explicado pelos critérios de parada adotados aqui (número máximo de avaliações da função-objetivo ou erro absoluto médio inferior a um dado limiar). Como discutido na Seção 4.1.1, a dopt-aiNet possui uma série de operadores de mutação e busca linear que exigem um grande número de avaliações da função-objetivo a cada iteração, o que acabou levando este algoritmo a ser executado aqui por um número de iterações significativamente menor que o das demais ferramentas (vide Tabela 4.5), possivelmente limitando sua capacidade de exploração do espaço de busca. Em todas as repetições feitas para todos os problemas, a dopt-aiNet encerrou sua execução ao atingir o número máximo permitido de avaliações da função-objetivo, diferentemente da cob-aiNet, que, na maioria dos casos, encerrou a simulação ao convergir para soluções dentro do limiar de erro desejado.

Por fim, o último aspecto que será discutido aqui diz respeito ao custo computacional de cada

algoritmo. Como os experimentos apresentados e discutidos até o momento foram feitos em máquinas e condições diferentes para cada algoritmo e cada problema, não é possível utilizá-los para estabelecer uma comparação direta entre os tempos computacionais envolvidos em cada ferramenta. No entanto, para que este aspecto não fosse deixado em aberto, foi escolhido o problema  $F_9$ , pela sua maior quantidade de ótimos locais, e reexecutadas as quatro ferramentas que estão sendo analisadas aqui sob as mesmas condições experimentais. O número de repetições, critérios de parada e demais parâmetros de cada algoritmo foram os mesmos apresentados na Seção 4.1.2, e os experimentos foram feitos em um computador com processador Intel Core2Duo T7500 de 2,2 GHz, 3 GB de RAM, Windows 7 Professional e Matlab R2010a. Todas as implementações foram feitas em linguagem Matlab.

Os tempos computacionais médios, em 25 repetições de cada algoritmo para o problema  $F_9$  estão apresentados na Tabela 4.6. Como é possível observar, os mecanismos responsáveis pelo novo modelo de rede imunológica introduzido em ambas as versões da *cob-aiNet* (com e sem busca local) trazem consigo um custo computacional importante, já que exigem comparações par a par entre todos os indivíduos da população a cada iteração. No entanto, como foi observado anteriormente, a *cob-aiNet* conseguiu, em muitos casos, atingir a precisão de solução desejada com um número de avaliações de função menor que o limite máximo estabelecido, o que pode fazer com que seu custo computacional maior seja diluído em situações em que as avaliações da função-objetivo são as etapas mais custosas do processo de otimização, tornando a ferramenta proposta vantajosa também sob este aspecto. No entanto, desconsiderando-se esta situação hipotética, pode-se entender este maior custo computacional da *cob-aiNet* como o preço a ser pago tanto pela sua maior capacidade de manutenção de diversidade quanto por seu ajuste dinâmico do tamanho da população mais eficiente, extensivamente verificados nos resultados discutidos ao longo desta seção.

Tab. 4.6: Média  $\pm$  Desvio Padrão do tempo computacional (em segundos) gasto por cada algoritmo a cada repetição do problema  $F_9$ , em  $\mathcal{R}^2$ . Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados com os da *cob-aiNet*) pode ser rejeitada com significância 0,05 estão marcados com \*.

	<b>cob-aiNet</b>	<b>cob-aiNet sem Busca Local</b>	<b>opt-aiNet</b>	<b>dopt-aiNet</b>
<b><math>F_9</math></b>	62,94 $\pm$ 17,51	69,98 $\pm$ 17,82 *	1,23 $\pm$ 0,07 *	5,86 $\pm$ 0,08 *

## 4.2 Análise de Sensibilidade

Nesta segunda parte dos experimentos de estudo da *cob-aiNet*, o objetivo foi realizar uma análise de sensibilidade do algoritmo aos seus principais parâmetros, para que fosse possível identificar qual o impacto que diferentes ajustes têm no desempenho geral da ferramenta. Para isso, nesta seção foi es-

colhido como base para os estudos do problema  $F_9$  com espaço de busca bidimensional, pelos mesmos motivos apresentados na Seção 4.1.3. A metodologia adotada aqui consistiu basicamente na variação de cada parâmetro do algoritmo individualmente, enquanto todos os demais permaneceram fixos em valores *padrão*, como apresentado na Tabela 4.7. Para cada valor de cada parâmetro, os experimentos foram repetidos 25 vezes e analisou-se tanto o erro absoluto médio obtido (entre a melhor solução encontrada e o ótimo global) quanto a distribuição final dos indivíduos no espaço de busca. O único parâmetro que não foi avaliado aqui foi o tamanho máximo da população  $maxAB$ , mantido sempre em  $maxAB = 100$ , uma vez que este parâmetro consiste basicamente em um limitante superior que evita a explosão populacional do algoritmo, sendo intuitiva a relação entre sua variação, o número de ótimos locais encontrados (conforme o problema) e o custo computacional total do algoritmo.

Tab. 4.7: Lista de valores padrão e valores testados dos parâmetros do algoritmo cob-aiNet, para os experimentos de análise de sensibilidade.

Parâmetro	Valor Padrão	Valores Testados
$nAB$ (tamanho inicial da população)	50	1; 25; 50; 75; 100
$nC^{min}$ (número mínimo de clones)	2	0; 2; 5; 7; 10
$nC^{max}$ (número máximo de clones)	10	2; 5; 10; 15; 20
$C_0$ (concentração inicial)	0,5	0,1; 0,5; 1
$\sigma_s$ (limiar de supressão)	0,05	0,001; 0,01; 0,05; 0,1; 0,25
$\beta^i$ (hipermutação)	3	0,005; 0,5; 1; 2; 3
$\beta^f$ (hipermutação)	0,005	0,005; 0,5; 1; 2; 3
$LS_{it}$ (número de iterações da busca local)	5	0; 1; 5; 10; 20
$LS_{freq}$ (frequência da busca local)	5	1; 5; 10; 20
$SR$ (intervalo inicial de busca da busca local)	$\sigma_s$	$\sigma_s$ ; metade do domínio de cada variável

### 4.2.1 Tamanho Inicial da População – $nAB$

O tamanho inicial da população é responsável por determinar o grau de exploração do espaço de busca com que o algoritmo iniciará sua execução. Como a cob-aiNet evolui a sua busca a partir dos indivíduos atuais em sua população, a definição de um número maior de soluções iniciais aumentará as chances de o algoritmo encontrar regiões promissoras logo nas primeiras iterações, enquanto que a adoção de poucas soluções iniciais tenderá a retardar este processo e atribuirá toda a responsabilidade de exploração do espaço de busca ao mecanismo de hipermutação do algoritmo, que deverá gerar novas soluções em regiões ainda inexploradas para que sejam inseridas na população.

Na Figura 4.8 estão representados a evolução do erro absoluto médio e do número final médio de indivíduos em função da variação do parâmetro  $nAB$ . Nestes experimentos,  $nAB$  assumiu valores 1, 25, 50, 75 e 100. Como é possível observar, a variação do tamanho da população teve um impacto baixo no erro absoluto médio, já que, apesar de ter sido observada alguma variação no valor médio, os

desvios padrão associados foram altos. No entanto, quando foi utilizado um valor muito baixo para o tamanho inicial da população o número final de indivíduos sofreu uma forte redução, que se refletiu diretamente na distribuição destas soluções no espaço de busca, como pode ser visto na Figura 4.9.

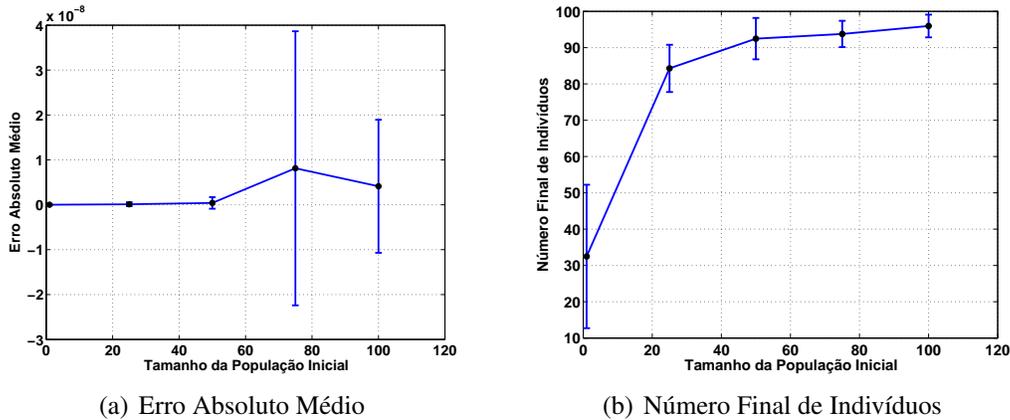


Fig. 4.8: Valores médios do erro absoluto e do número final de indivíduos retornado pela *cob-aiNet* em função da variação do tamanho da população inicial  $nAB$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

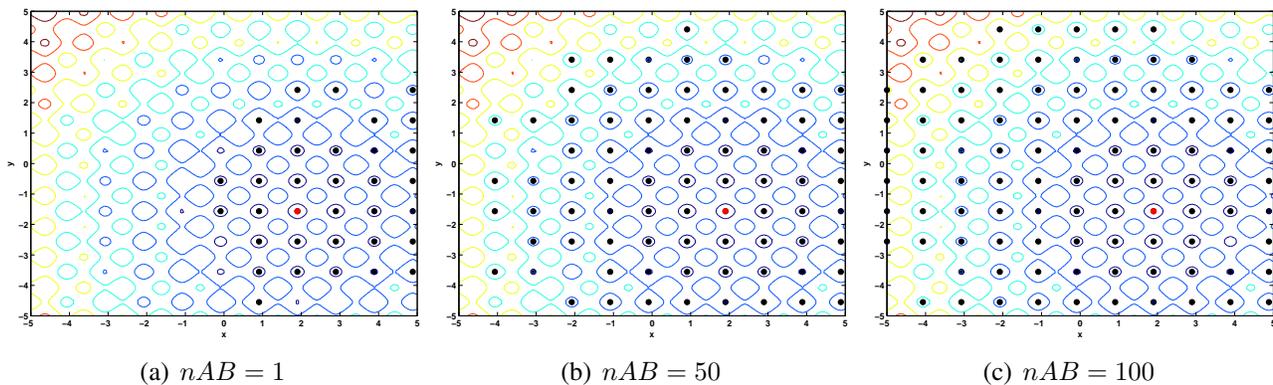


Fig. 4.9: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo *cob-aiNet* com (a)  $nAB = 1$ , (b)  $nAB = 50$  e (c)  $nAB = 100$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

Diante disso, é possível concluir que a utilização de um número maior de indivíduos na população inicial da *cob-aiNet* tende a ser benéfica, mesmo que isso leve a uma maior exigência computacional nas primeiras iterações do algoritmo.

### 4.2.2 Números Mínimo e Máximo de Clones – $nC^{min}$ e $nC^{max}$

Como visto na Seção 3.3, para reproduzir os mecanismos de ativação de células em redes imunológicas foi proposto um mecanismo de clonagem em que o número de clones  $nC$  gerado para cada indivíduo é diretamente proporcional à sua concentração. Para o seu correto funcionamento, este operador de clonagem exige a definição de dois parâmetros para que  $nC$  opere dentro de uma faixa específica, que são os números mínimo  $nC^{min}$  e máximo  $nC^{max}$  de clones, os quais serão analisados aqui.

O parâmetro  $nC^{min}$  recebeu os valores 0, 2, 5, 7 e 10, sendo que, neste último caso, temos  $nC^{min} = nC^{max}$  e a variação do número de clones proporcionalmente à concentração de cada célula deixa de ocorrer. Os resultados obtidos para o erro absoluto médio e número final médio de indivíduos em função da variação de  $nC^{min}$  estão representados na Figura 4.10.

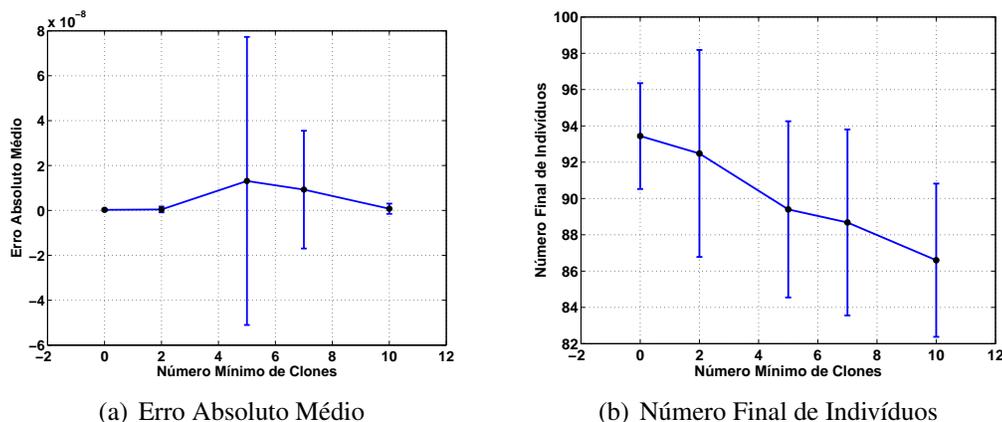


Fig. 4.10: Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do número mínimo de clones  $nC^{min}$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

Como é possível perceber, a variação do erro absoluto médio foi pequena diante da variação de  $nC^{min}$ , o que indica que, neste aspecto, o desempenho da cob-aiNet não é muito sensível a diferentes ajustes deste parâmetro, pelo menos nas condições estudadas aqui. Já com relação ao número final de soluções, representado na Figura 4.10(b), apesar dos elevados desvios padrão presentes, é possível perceber claramente uma suave tendência de redução no número médio de indivíduos retornados, mas isto não se refletiu significativamente na distribuição dos indivíduos no espaço de busca, como ilustrado na Figura 4.11.

Já o parâmetro  $nC^{max}$  foi analisado com os valores 2, 5, 10, 15 e 20, sendo o primeiro caso novamente equivalente à situação em que  $nC^{min} = nC^{max}$ . Pelos resultados obtidos com a variação deste parâmetro, ilustrados na Figura 4.12, podemos perceber que novamente o erro absoluto médio praticamente não se altera em função da variação de  $nC^{max}$ , exceto para o caso em que  $nC^{max} = 20$ ,

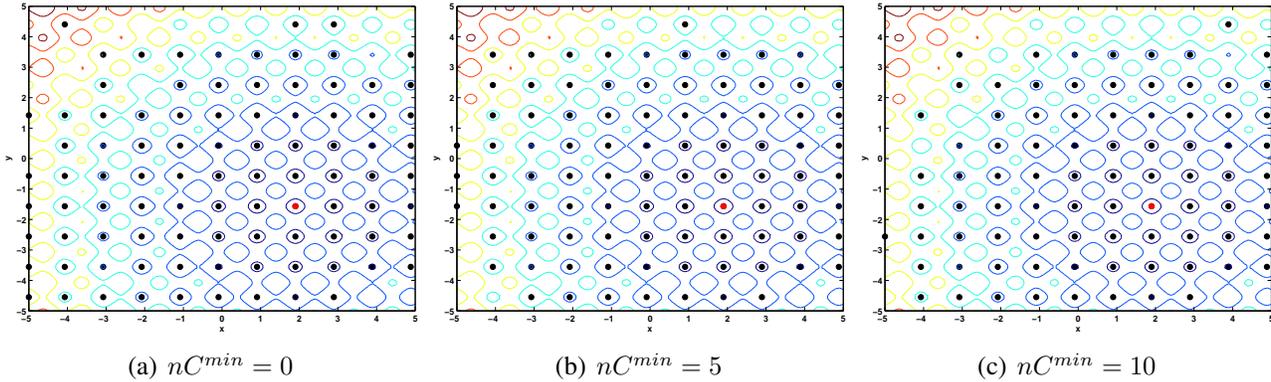


Fig. 4.11: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo *cob-aiNet* com (a)  $nC^{min} = 0$ , (b)  $nC^{min} = 5$  e (c)  $nC^{min} = 10$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

que apresentou um erro ligeiramente maior que o observado para os demais valores deste parâmetro. Com relação ao número final médio de indivíduos retornados, é possível observar um comportamento semelhante ao obtido para  $nC^{min}$ , ou seja, uma tendência de redução no número de soluções retornadas com o aumento de  $nC^{max}$ , exceto para o caso em que  $nC^{max} = nC^{min} = 2$ , que apresentou um valor médio de soluções retornadas menor que para a situação em que  $nC^{max} = 5$ . No entanto, apesar desta tendência de diminuição do tamanho da população final, a distribuição de soluções no espaço de busca não foi afetada significativamente, como pode ser visto na Figura 4.13.

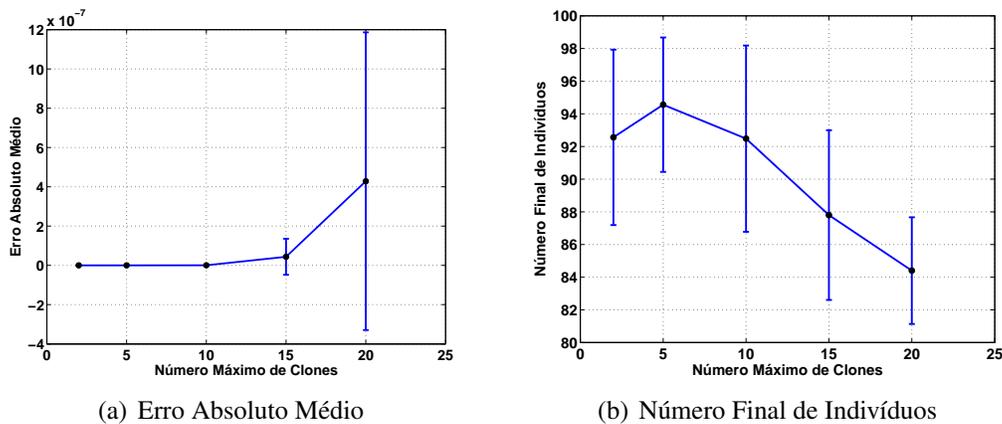


Fig. 4.12: Valores médios do erro absoluto e do número final de indivíduos retornados pela *cob-aiNet* em função da variação do número máximo de clones  $nC^{max}$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

Diante destes resultados, é possível concluir que valores menores, tanto para  $nC^{min}$  quanto para  $nC^{max}$ , tendem a levar aos menores erros absolutos médios e também ao maior número de ótimos locais encontrados, desde que tais parâmetros sejam diferentes entre si, ou seja, desde que haja uma

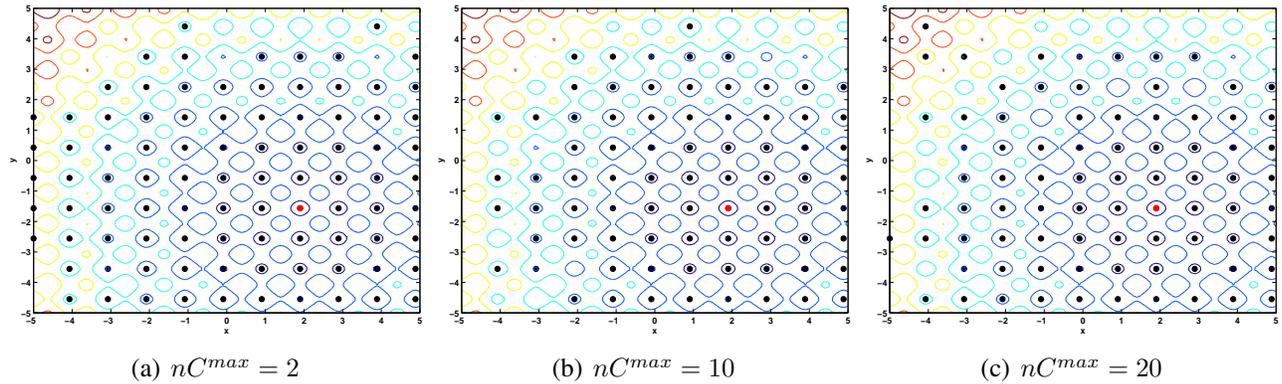


Fig. 4.13: Representação gráfica das curvas de nível do problema  $F_0$  em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet com (a)  $nC^{max} = 2$ , (b)  $nC^{max} = 10$  e (c)  $nC^{max} = 20$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

variação do número de clones gerado para cada solução em função de sua concentração. Se desconsiderarmos os desvios padrão observados, os melhores resultados destes experimentos foram obtidos com  $nC^{min} = 0$  e  $nC^{max} = 5$ .

### 4.2.3 Concentração Inicial – $C_0$

O parâmetro de concentração inicial  $C_0$  de cada célula basicamente determina, mesmo que indiretamente, a probabilidade de células novas na população serem imediatamente eliminadas ao invadirem regiões de influência de outras células de melhor *fitness*. Caso  $C_0$  seja definido com um valor muito baixo, são grandes as chances de que a penalização sofrida por uma célula nova a deixe com concentração nula, causando sua supressão (vide Equação 3.6). Por outro lado, com um valor alto de  $C_0$ , dificilmente a penalização sofrida em uma única iteração seria suficiente para zerar a concentração de uma dada célula, o que permite sua manutenção na população por um tempo maior.

Nos estudos feitos aqui, o parâmetro  $C_0$  assumiu os valores 0, 1; 0, 5 e 1, 0 e, como pode ser observado nos resultados apresentados na Figura 4.14, a variação deste parâmetro e as possíveis diferenças de comportamento descritas acima praticamente não tiveram nenhum impacto nos resultados finais da cob-aiNet, tanto em relação ao erro absoluto médio quanto ao número final médio de indivíduos retornados.

### 4.2.4 Limiar de Supressão – $\sigma_s$

Como discutido na Seção 3.3.3, o limiar de supressão  $\sigma_s$  é notoriamente um dos parâmetros de mais difícil ajuste no algoritmo cob-aiNet. Como a afinidade entre células da população se baseia na distância entre indivíduos no espaço de busca, a determinação de uma vizinhança (definida por  $\sigma_s$ )

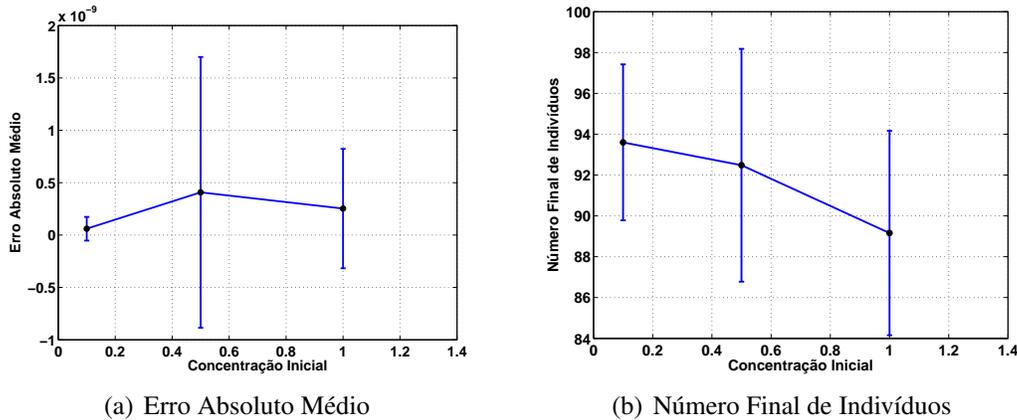


Fig. 4.14: Valores médios do erro absoluto e do número final de indivíduos retornado pela *cob-aiNet* em função da variação da concentração inicial  $C_0$  de cada célula. As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

que cubra apenas a região de ótimo local em que cada célula se encontra depende de um conhecimento prévio sobre características específicas da superfície de busca do problema, que geralmente não estão disponíveis para o usuário. Isto torna a correta configuração de  $\sigma_s$  uma tarefa nada trivial e que deve ser feita cuidadosamente pelo usuário, já que este parâmetro está diretamente relacionado tanto à capacidade da *cob-aiNet* em manter diversidade na população quanto à sua capacidade de ajuste dinâmico do tamanho da população, conforme as características do problema.

Foram feitas aqui simulações com valores 0, 0,01, 0, 01, 0, 05, 0, 1 e 0, 25 para  $\sigma_s$ , e os resultados obtidos estão representados na Figura 4.15. É importante ressaltar aqui que, em todos os experimentos desta tese, os valores de  $\sigma_s$  são relativos à máxima distância possível entre duas soluções no espaço de busca, dado o domínio do problema. Ou seja, supondo que a distância máxima em um espaço de busca hipotético seja  $D^{max} = 10$ , um valor  $\sigma_s = 0, 1$  determina que cada célula  $i$  da população terá uma região de influência que corresponde a uma hiper-esfera centrada em  $i$  e com raio  $r = \sigma_s \cdot D^{max} = 1$ .

Como é possível perceber pelos resultados na Figura 4.15, o erro absoluto médio sofreu um pico de aumento para o caso em que  $\sigma_s = 0, 1$ , mas isso ocorreu devido à presença de um *outlier* na sexta repetição da execução do algoritmo, que levou a um  $EAM = 0, 995$ . Ou seja, desconsiderando-se esta única execução, o erro absoluto médio obtido pela *cob-aiNet* não variou significativamente em função de  $\sigma_s$ . Este *outlier* corresponde, basicamente, a uma repetição em que a melhor solução da população permaneceu presa em um ótimo local do problema, sendo que o algoritmo não foi capaz de posicionar nenhum indivíduo no vale correspondente ao ótimo global. Por outro lado, como previsto, o número final de indivíduos retornados a cada execução foi severamente afetado por variações neste parâmetro, e isto se refletiu diretamente na distribuição das soluções no espaço de busca, como pode ser observado na Figura 4.16.

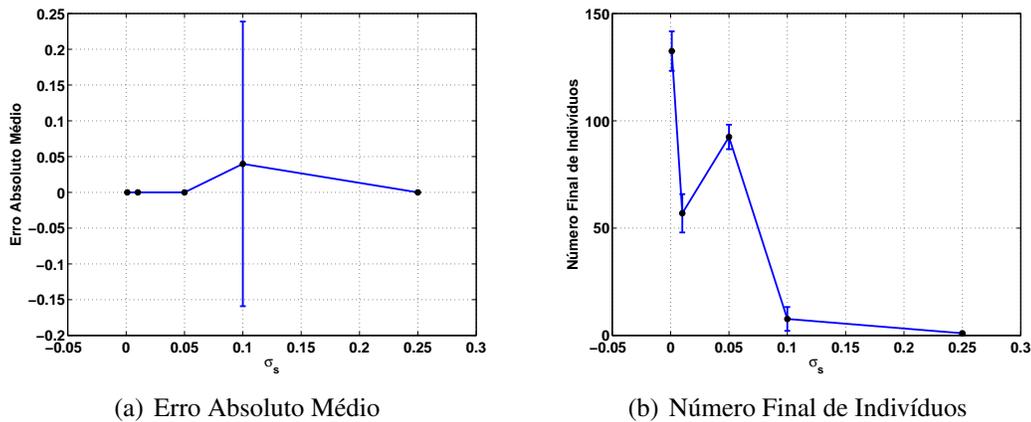


Fig. 4.15: Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do limiar de supressão  $\sigma_s$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

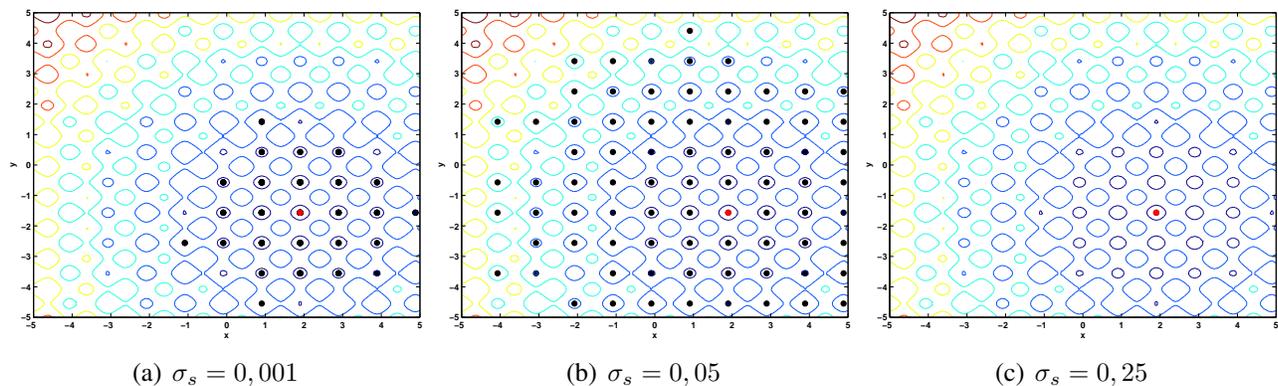


Fig. 4.16: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet com (a)  $\sigma_s = 0,001$ , (b)  $\sigma_s = 0,05$  e (c)  $\sigma_s = 0,25$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

Este impacto negativo observado na distribuição final das soluções se dá justamente pela ocorrência de situações como as descritas na Seção 3.3.3, em que limiares de supressão muito grandes englobam diversos vales ou picos enquanto que limiares muito pequenos permitem que múltiplas soluções permaneçam em um mesmo vale ou pico. Esta última situação corresponde ao ilustrado na Figura 4.16(a), em que 120 soluções estão espalhadas por apenas 26 vales do problema.

Como o correto ajuste de  $\sigma_s$  depende diretamente de características específicas do espaço de busca do problema, que dificilmente são conhecidas *a priori* pelo usuário, a estratégia de ajuste deste parâmetro consiste basicamente na execução de alguns experimentos preliminares, que permitam identificar quais valores de  $\sigma_s$  resultam no comportamento desejado: caso o foco da otimização seja apenas a obtenção da melhor solução global,  $\sigma_s$  poderá receber valores mais altos, enquanto que caso

o objetivo seja encontrar um maior número de ótimos locais, valores menores devem ser atribuídos a este parâmetro (esta abordagem foi adotada nos experimentos que serão discutidos na Seção 4.3). Para os casos em que forem adotados valores baixos para  $\sigma_s$ , o usuário deve avaliar aquelas soluções retornadas pelo algoritmo que apresentam o maior grau de similaridade entre si, e analisar se este grau de similaridade indica que as duas soluções são distintas ou não no contexto de seu problema. Caso tais soluções possam ser consideradas equivalentes, o usuário deve então aumentar o limiar de supressão  $\sigma_s$ , até que obtenha apenas soluções distintas entre si. Por fim, é importante ressaltar aqui que, caso o problema seja monomodal, a *cob-aiNet* sempre tenderá a convergir para uma única solução, a não ser que sejam utilizados valores muito baixos para o parâmetro  $\sigma_s$ .

#### 4.2.5 Parâmetros Inicial e Final do Operador de Hipermutação – $\beta^i$ e $\beta^f$

Os parâmetros  $\beta^i$  e  $\beta^f$  do operador de mutação são os responsáveis pelo chaveamento do comportamento do algoritmo *cob-aiNet* entre um modo que prioriza a exploração do espaço de busca, nas iterações iniciais, para outro que permite que seja feito um ajuste fino das soluções já encontradas nas iterações finais da execução.

Para avaliar o impacto destes parâmetros no desempenho da ferramenta, primeiramente  $\beta^f$  foi mantido fixo em 0,005 e  $\beta^i$  assumiu os valores 0,005; 0,5; 1,0; 2,0 e 3,0. O caso  $\beta^i = 0,005$  corresponde à situação em que  $\beta^i = \beta^f$ , ou seja, não há a variação do parâmetro  $\beta$  ao longo da execução do algoritmo e, conseqüentemente, não ocorre o chaveamento entre os comportamentos descritos acima.

Os resultados obtidos para a variação de  $\beta^i$  estão apresentados na Figura 4.17 e, a partir deles, pode-se observar que este parâmetro tem um impacto significativo, tanto no erro absoluto médio quanto no número final médio de indivíduos retornados. Além disso, o ajuste incorreto de  $\beta^i$  também impacta negativamente a distribuição das soluções no espaço de busca, como pode ser visto na Figura 4.18.

Quando  $\beta^i$  possui um valor muito baixo, o algoritmo tende a fazer apenas ajustes finos nas soluções geradas inicialmente, o que aumenta o erro absoluto médio final, uma vez que não há exploração do espaço de busca ao longo da execução do algoritmo. Além disso, também há uma tendência de redução no número final de soluções encontradas, já que, como visto na Seção 3.3, a inserção de novos indivíduos na população está diretamente relacionada à capacidade do mecanismo de hipermutação gerar indivíduos em regiões do espaço de busca promissoras e ainda inexploradas, o que, por sua vez, depende diretamente do parâmetro  $\beta^i$ .

Já para o parâmetro  $\beta^f$ , que recebeu valores 0,005; 0,5; 1; 2 e 3 enquanto  $\beta^i$  foi mantido fixo em 3, os resultados apresentados na Figura 4.19 mostram que sua variação não causa praticamente nenhum impacto negativo no desempenho do algoritmo, tanto em relação ao erro absoluto quanto ao

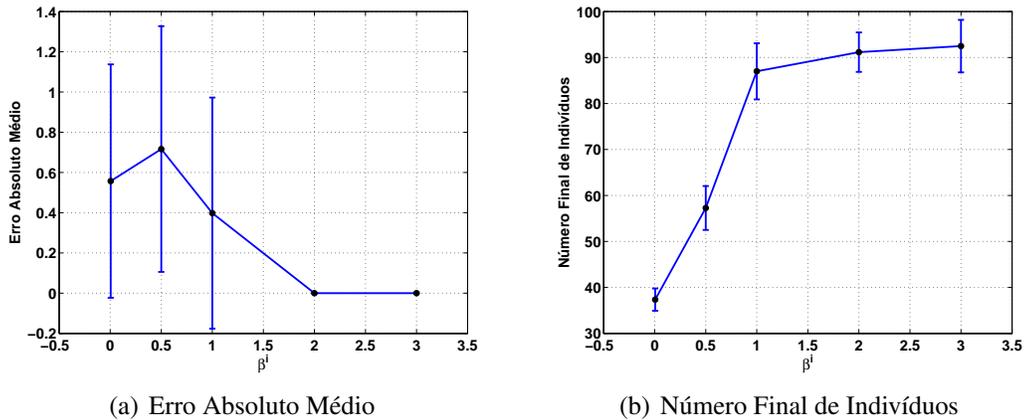


Fig. 4.17: Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do parâmetro inicial do operador de hipermutação  $\beta^i$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

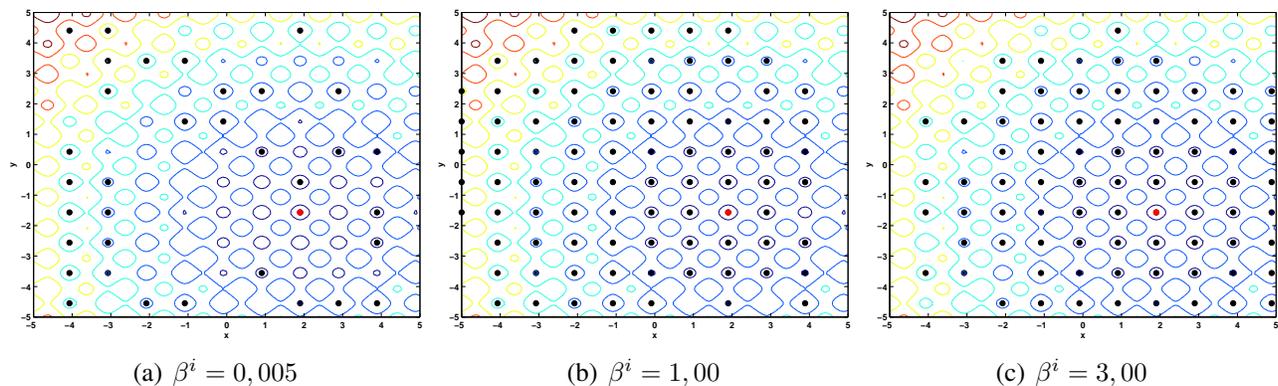


Fig. 4.18: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet com (a)  $\beta^i = 0,005$ , (b)  $\beta^i = 1,00$  e (c)  $\beta^i = 3,00$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

número final médio de soluções geradas pela cob-aiNet.

Esta aparente “robustez” da cob-aiNet em relação a variações do parâmetro  $\beta^f$  se dá pois o mecanismo de busca local consegue compensar a falta de etapas de ajuste fino nas iterações finais do algoritmo, que ocorre nas situações em que  $\beta^f$  possui valores próximos aos de  $\beta^i$ .

Sendo assim, é possível concluir que, caso o operador de busca local esteja ativo na execução do algoritmo, o usuário deve ajustar  $\beta^i$  com um valor alto para que o algoritmo seja capaz de explorar adequadamente o espaço de busca e encontrar regiões promissoras, enquanto que o ajuste de  $\beta^f$  não se torna crucial para um bom desempenho da ferramenta. No entanto, caso a busca local não esteja ativa, além de  $\beta^i$  ser ajustado com um valor alto, também é recomendado que  $\beta^f$  tenha um valor baixo, de forma a permitir que o algoritmo realize adequadamente a etapa de ajuste fino das soluções.

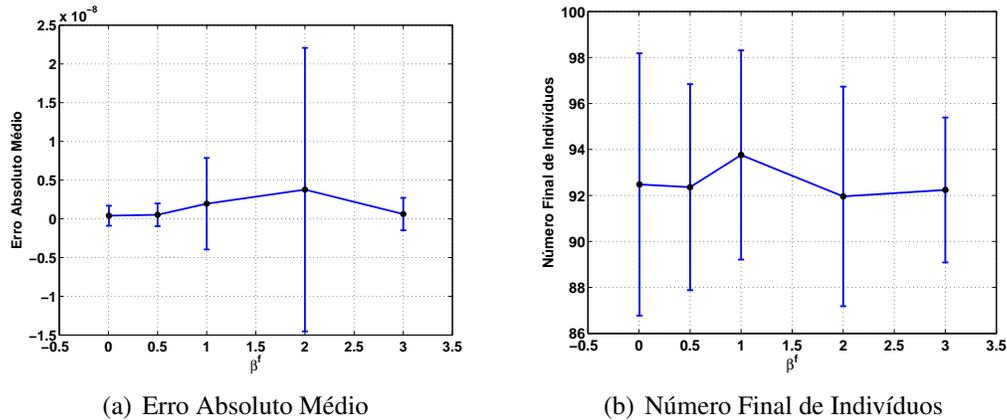


Fig. 4.19: Valores médios do erro absoluto e do número final de indivíduos retornado pela *cob-aiNet* em função da variação do parâmetro final do operador de hipermutação  $\beta^f$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

#### 4.2.6 Número de Iterações da Busca Local – $LS_{it}$

O parâmetro  $LS_{it}$  determina o número de iterações do operador de busca local que deverão ser executadas para cada indivíduo da população, sempre que a aplicação deste operador for disparada (vide Pseudo-código 3.2). Como discutido anteriormente, a utilização do operador de busca local dá ao algoritmo uma maior capacidade de refinamento de suas soluções mas, em contrapartida, eleva significativamente seu custo computacional. Diante disso, é desejável que este operador seja aplicado pelo menor número de iterações necessário para prover um ajuste adequado das soluções, evitando-se assim que ocorram avaliações desnecessárias da função-objetivo.

Nestes experimentos, o algoritmo *cob-aiNet* foi avaliado com os valores 0 (sem busca local), 1, 5, 10 e 20 para o parâmetro  $LS_{it}$ , e os resultados obtidos estão representados na Figura 4.20. Considerando-se inicialmente o erro absoluto médio, é possível observar um aumento significativo deste critério para o caso em que  $LS_{it} = 20$ , o que foi causado por duas repetições em que o ótimo global do problema não foi encontrado e o EAM se manteve em 0,995. Para os demais casos, pela Figura 4.20(a) o EAM aparentemente não sofreu variações perceptíveis mas, observando os valores propriamente ditos, o algoritmo só conseguiu atingir  $EAM < 1 \cdot 10^{-9}$  para  $LS_{it} = 5$  e  $LS_{it} = 10$  permanecendo fora da precisão desejada para os demais valores. Diante disso, para este problema, 5 é o valor mínimo que deve ser definido para o número de iterações por solução da busca local, para que se obtenha um bom erro absoluto médio final.

Com relação ao número final de indivíduos retornados, é possível observar na Figura 4.20(b) que, com o aumento do número de iterações de busca local, o número de soluções retornadas e de ótimos locais cobertos tende a diminuir, o que se dá graças à convergência das soluções para regiões de ótimos locais mais promissoras, representados na Figura 4.21 por curvas de nível em tons mais

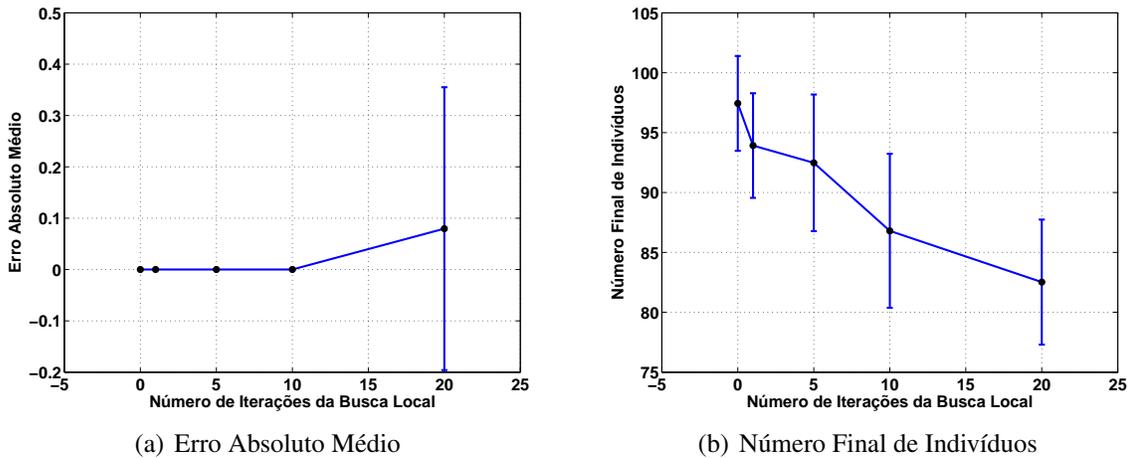


Fig. 4.20: Valores médios do erro absoluto e do número final de indivíduos retornado pela cob-aiNet em função da variação do número de iterações da busca local  $LS_{it}$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

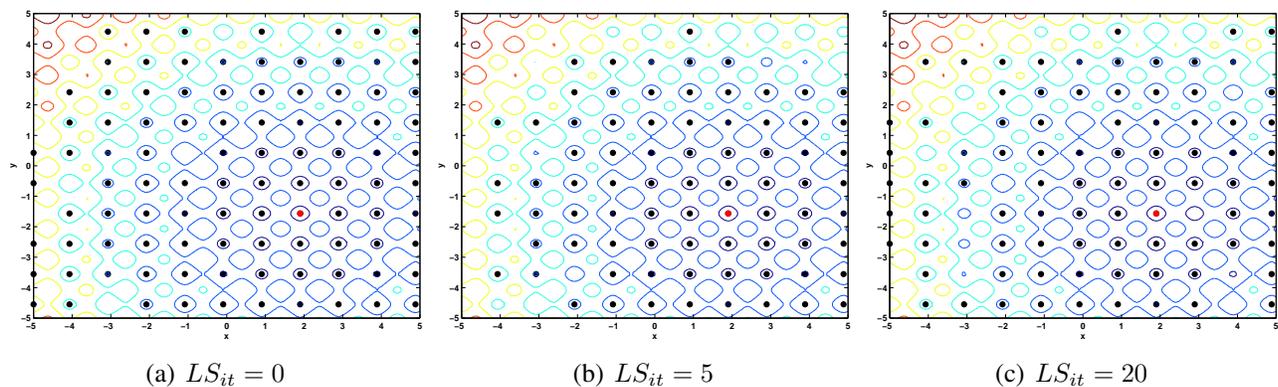


Fig. 4.21: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo cob-aiNet com (a)  $LS_{it} = 0$ , (b)  $LS_{it} = 5$  e (c)  $LS_{it} = 20$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

escuras de azul.

Sendo assim, fica claro que, para este problema,  $LS_{it} = 5$  é o melhor ajuste possível para o número de iterações da busca local, uma vez que corresponde ao menor valor para este parâmetro que leva o algoritmo cob-aiNet a satisfazer a precisão desejada para o EAM, permitindo simultaneamente a obtenção de um bom número de soluções finais, associadas aos melhores ótimos locais do problema.

#### 4.2.7 Frequência da Busca Local – $LS_{freq}$

O parâmetro  $LS_{freq}$ , também associado ao operador de busca local, determina o intervalo, em número de iterações do loop principal do algoritmo, em que tal operador será aplicado a todas as

soluções da população. Da mesma maneira que o número de iterações de busca local discutido anteriormente, a frequência com que a busca local é aplicada também tem impactos significativos no custo computacional do algoritmo, o que torna desejável que se defina  $LS_{freq}$  com o maior valor possível, desde que isto não afete as características desejadas para os resultados finais do algoritmo.

Foram feitos aqui experimentos em que  $LS_{freq}$  recebeu os valores 1 (aplicação da busca local em toda iteração do algoritmo), 5, 10 e 20, e os resultados obtidos estão representados na Figura 4.22. Como esperado, é possível observar que, quanto maior for o intervalo entre operações de busca local, ou seja, quanto menos aplicações de busca local forem feitas ao longo da execução do algoritmo, maior tendem a ser tanto o erro absoluto médio quanto o número final médio de soluções retornadas pelo algoritmo. Isto se dá pelo fato do comportamento de convergência das soluções para as regiões do espaço de busca em que estão situados os ótimos locais mais promissores poder ser diretamente associado ao operador de busca local em questão.

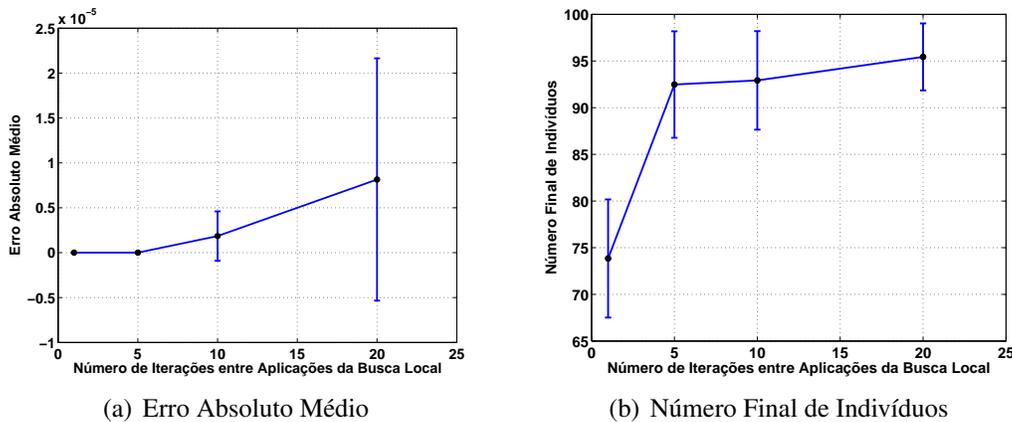


Fig. 4.22: Valores médios do erro absoluto e do número final de indivíduos retornado pela *cob-aiNet* em função da variação da frequência da busca local  $LS_{freq}$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

Especificamente para este problema, o intervalo máximo entre aplicações do operador de busca local que ainda permite que o algoritmo atinja um  $EAM < 1 \cdot 10^{-9}$  é de  $LS_{freq} = 5$  iterações do *loop* principal do algoritmo, sendo que valores maiores para este parâmetro resultam em erros superiores ao limiar desejado, mesmo levando também à identificação de um maior número de ótimos locais para o problema.

#### 4.2.8 Intervalo de Busca Inicial da Busca Local – $SR$

Por fim, o último parâmetro do algoritmo *cob-aiNet* que será estudado aqui é o intervalo inicial de busca  $SR$  do operador de busca local. Na proposta original de Tseng & Chen (2008), o intervalo

inicial do operador de busca local foi definido como sendo equivalente à metade do domínio de cada variável do problema, enquanto que, nesta tese, sugere-se que seja utilizado um intervalo inicial de busca igual ao limiar de supressão  $\sigma_s$ . Esta diferença entre as propostas se dá pelo fato do trabalho de Tseng & Chen (2008) propor o operador de busca local visando a obtenção do ótimo global do problema, enquanto que, aqui, a proposta inicial é utilizá-lo para posicionar cada solução da população da cob-aiNet no ótimo local mais próximo.

Sendo assim, para avaliar como estas duas abordagens influem no desempenho geral do algoritmo, os dois valores propostos foram utilizados nos experimentos. No primeiro caso, como todas as variáveis do problema  $F_9$  têm domínio  $[-5, 5]$ , segundo a proposta original de Tseng & Chen (2008) o intervalo inicial da busca local para todas elas foi então definido como  $SR = \frac{1}{2} \cdot [5 - (-5)] = 5$ . Já de acordo com a proposta desta tese e considerando-se que em todo este trabalho foi adotado um  $\sigma_s$  relativo à distância máxima possível no espaço de busca  $D^{max}$ , o intervalo inicial da busca local foi então definido como  $SR = \sigma_s \cdot D^{max} = 0,05 \cdot 14,1421 = 0,7071$  para todas as variáveis do problema.

Dito isto, é possível observar pelos resultados mostrados nas Figuras 4.23 e 4.24 que, como esperado, o intervalo inicial do operador de busca local tem um impacto significativo principalmente no número final de soluções obtidas pela cob-aiNet e na distribuição destas soluções no espaço de busca. Com relação ao erro absoluto médio, ambas as configurações de  $SR$  levam à obtenção de soluções com a precisão desejada de  $1 \cdot 10^{-9}$ , mas a utilização da proposta original de Tseng & Chen (2008) resulta em erros que apresentam tanto a média quanto o desvio padrão menores. Com relação ao número de soluções retornadas, a abordagem proposta nesta tese permitiu a localização de uma quantidade significativamente maior de ótimos locais, enquanto que a proposta original de Tseng & Chen (2008) levou o algoritmo a convergir apenas para aqueles ótimos locais posicionados em regiões mais promissoras do espaço de busca.

Diante disso, cabe ao usuário definir o valor inicial do intervalo de busca do operador de busca local de acordo com suas necessidades, escolhendo a alternativa proposta nesta tese caso o objetivo principal seja encontrar o maior número possível de ótimos locais, ou a abordagem original de Tseng & Chen (2008), caso o foco seja apenas na obtenção da melhor solução possível para o problema em questão.

### 4.3 Avaliação da cob-aiNet em Espaços de Maior Dimensão

Na terceira e última etapa dos experimentos de avaliação do algoritmo cob-aiNet, a ferramenta proposta foi aplicada a todos os 25 problemas da competição CEC'2005, com espaços de busca de dimensão  $D = 10$  e  $D = 30$ . Para que os resultados pudessem ser diretamente comparados com os

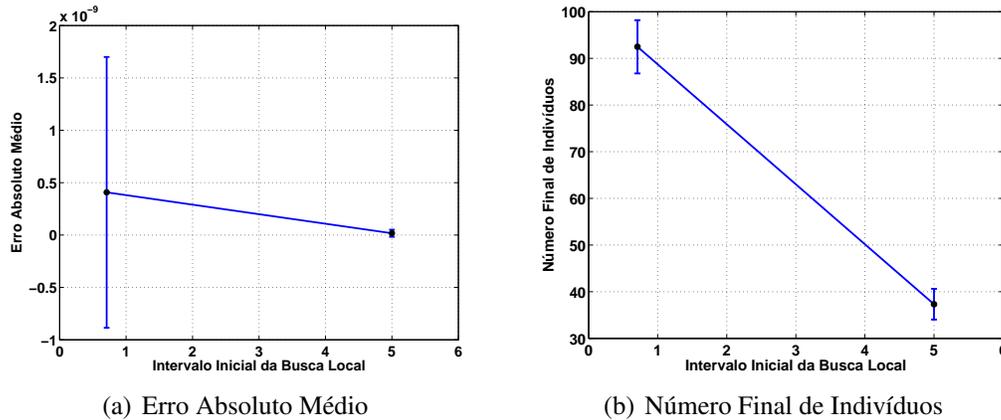


Fig. 4.23: Valores médios do erro absoluto e do número final de indivíduos retornado pela *cob-aiNet* em função da variação do intervalo de busca inicial da busca local  $SR$ . As barras verticais correspondem aos desvios padrão avaliados em 25 repetições.

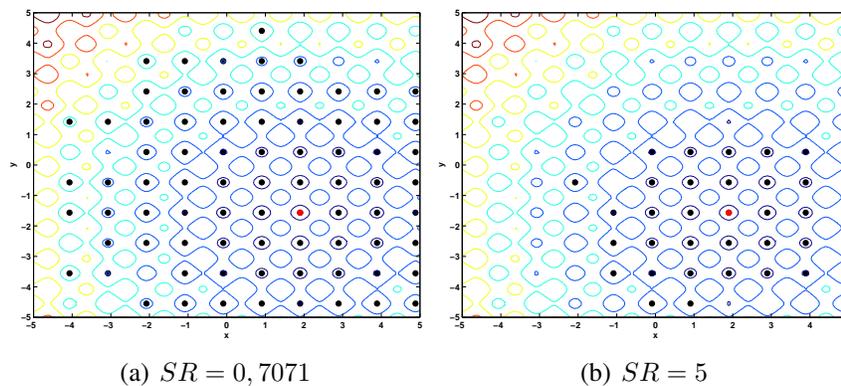


Fig. 4.24: Representação gráfica das curvas de nível do problema  $F_9$  em conjunto com as populações finais retornadas pelo algoritmo *cob-aiNet* com (a)  $SR = 0,7071$  e (b)  $SR = 5$ , na primeira repetição dos experimentos. As melhores soluções estão representadas em vermelho.

algoritmos participantes da competição, foram adotados como critérios de parada para a *cob-aiNet* tanto um número máximo de avaliações da função-objetivo igual a  $D \cdot 10^4$  (onde  $D$  é a dimensão do problema) quanto um erro absoluto médio entre a melhor solução encontrada e o ótimo global do problema menor ou igual a  $10^{-8}$ . Além disso, a população do algoritmo foi inicializada aleatoriamente nos intervalos previamente definidos para cada problema (vide Apêndice A) e foram feitas 25 repetições das execuções para cada problema. Nos resultados que serão discutidos na Seção 4.3.2, serão comparados os valores médios e desvios padrão dos erros absolutos obtidos por cada algoritmo da competição, nas 25 repetições, além dos algoritmos *cob-aiNet*, *opt-aiNet* e *dopt-aiNet*.

Como o foco desta etapa de experimentos é encontrar a melhor solução possível para os 25 problemas, os parâmetros da *cob-aiNet* foram ajustados visando a esta tarefa específica e não a localização

do maior número possível de ótimos locais, como feito anteriormente nos experimentos descritos na Seção 4.1. Além disso, foi adotado um conjunto fixo de parâmetros para todos os problemas tratados aqui, mesmo sabendo-se que a utilização de parâmetros específicos para cada problema poderia levar a resultados melhores. Sendo assim, para o caso de  $D = 10$ , a cob-aiNet foi executada com tamanho inicial da população  $nAB = 50$ , tamanho máximo da população  $maxAB = 200$ , números mínimo e máximo de clones  $nC^{min} = 2$  e  $nC^{max} = 10$ ,  $\beta^i = 3$ ,  $\beta^f = 0,001$ , limiar de supressão  $\sigma_s = 0,2$ , concentração inicial  $C_0 = 0,5$ , e operador de busca local sendo executado por  $LS_{it} = 5$  iterações a cada  $LS_{freq} = 5$  iterações do *loop* principal, com intervalo de busca inicial  $SR = \frac{1}{2} \cdot (\text{domínio})$  para cada variável de cada problema. Nos experimentos com  $D = 30$ , os parâmetros adotados foram praticamente os mesmos, exceto o tamanho máximo da população que foi definido como  $maxAB = 350$ .

Já para os outros dois SIAs, os parâmetros utilizados foram os seguintes: a dopt-aiNet foi executada com tamanho da população inicial  $nAB = 50$ , tamanho máximo da população  $maxAB = 200$  para  $D = 10$  e  $maxAB = 350$  para  $D = 30$ , número de clones  $nC = 10$ , limiar de supressão (para distância de linha)  $\sigma_s = 5$ , número de iterações entre operações de duplicação gênica  $n_{gene} = 10$ , e  $rank = 100$ ; e a opt-aiNet foi executada com tamanho inicial da população  $nAB = 50$ , número de clones  $nC = 10$ ,  $\beta = 1$ , número de iterações entre etapas de supressão  $n_{supp} = 10$ , porcentagem de novas células que devem ser inseridas após a supressão  $d = 40\%$  e limiar de supressão  $\sigma_s = 0,2$ .

Na próxima seção, os onze algoritmos que participaram da competição CEC'2005 serão brevemente discutidos e seus aspectos mais importantes serão destacados, para que se possa ter uma visão geral de quais paradigmas estão presentes nos resultados que serão discutidos na Seção 4.3.2. Caso o leitor tenha interesse em maiores detalhes sobre estas ferramentas, recomenda-se a consulta às referências originais de cada uma delas.

### 4.3.1 Algoritmos Participantes da Competição

O algoritmo que apresentou o melhor desempenho médio da competição CEC'2005, tanto para  $D = 10$  quanto para  $D = 30$ , foi o G-CMA-ES<sup>7</sup>, proposto por Auger & Hansen (2005b). O algoritmo G-CMA-ES é uma estratégia evolutiva, baseada na técnica de *adaptação da matriz de covariância* (do inglês *Covariance Matrix Adaptation Evolution Strategy* – CMA-ES – Hansen & Ostermeier, 1996), que introduz um mecanismo de reinicializações sucessivas do algoritmo, sendo que a cada reinicialização o número de indivíduos na população é dobrado. Por se basear no CMA-ES, o G-CMA-ES também gera novas soluções, a cada iteração, através de um processo de amostragem que se utiliza de uma distribuição normal multivariada. O grande diferencial do CMA-ES está justamente na maneira

<sup>7</sup>Será adotada aqui a mesma notação de siglas utilizada na competição CEC'2005 para designar cada algoritmo competidor, mesmo muitas delas não tendo sido utilizadas pelos autores proponentes de cada ferramenta nos respectivos artigos originais.

cumulativa com que a matriz de covariância é atualizada a cada iteração, que agrega as informações obtidas ao longo de toda a evolução do algoritmo. Atualmente, a abordagem CMA-ES é considerada estado-da-arte dentre as meta-heurísticas para otimização, o que tem levado à proposição de diversos novos algoritmos baseados nesta técnica (Chun-Kit & Ho-Fung, 2010; Kruisselbrink et al., 2010; Meiyi et al., 2009; Shir et al., 2007; Voss et al., 2008) e também a muitas aplicações práticas (Colutto et al., 2010; Gregory & Werner, 2010; Jebalia et al., 2007).

Uma outra ferramenta semelhante ao G-CMA-ES, denominada L-CMA-ES, também foi proposta por Auger & Hansen (2005a) para a competição CEC'2005, sendo que a única diferença entre estes dois algoritmos está no fato do L-CMA-ES não duplicar o tamanho de sua população a cada reinicialização. Em ambos os algoritmos, cada execução feita após um reinício é independente das execuções anteriores da ferramenta.

A variação da técnica de estimação de densidade EDA (Larrañaga & Lozano, 2002), utilizada por Yuan & Gallagher (2005) na competição, também adota uma estratégia de amostragem de novos indivíduos a partir de uma distribuição normal multivariada mas, diferentemente dos algoritmos baseados no CMA-ES, a matriz de covariância é determinada a cada iteração a partir dos indivíduos presentes na população, e não incrementalmente. Com relação a outras abordagens de EDA, o algoritmo de Yuan & Gallagher (2005) tem como principais diferenciais a introdução de novos mecanismos que controlam dinamicamente a convergência da variância da distribuição normal multivariada, favorecendo ou não a exploração do espaço de busca, conforme a evolução do algoritmo e as características do problema.

Dentre os representantes dos algoritmos genéticos (AGs – Bäck et al., 2000), o BLX-GL50, proposto por Garcia-Martinez & Lozano (2005b), adota codificação real dos indivíduos e emprega um operador de cruzamento (*crossover*) baseado no conceito de *pais*. Neste operador, os novos filhos são gerados na vizinhança do indivíduo *mãe* conforme uma certa distribuição de probabilidade que, por sua vez, é definida pelo indivíduo *pai*. Para determinar quais indivíduos da população serão considerados pais e quais serão mães, o algoritmo BLX-GL50 utiliza o processo de *diferenciação em machos e fêmeas*, proposto por Garcia-Martinez & Lozano (2005a).

Outro AG presente na competição CEC'2005 é o SPC-PNX, proposto por Ballester et al. (2005). Este algoritmo é basicamente um algoritmo genético incremental (ou *steady-state*) puro, que também adota codificação real para os indivíduos e utiliza os operadores de cruzamento PNX e de substituição denominado *scaled probabilistic crowding scheme*, ambos introduzidos em Ballester & Carter (2004). Um segundo algoritmo genético incremental utilizado na competição é o K-PCX (Sinha et al., 2005) que também utiliza o operador de cruzamento PNX. No entanto, diferentemente do SPC-PNX, o K-PCX seleciona aleatoriamente indivíduos da população para serem substituídos pelos novos indivíduos gerados.

Já o algoritmo BLX-MA, desenvolvido por Molina et al. (2005), é um algoritmo memético (Moscato, 1989) que utiliza codificação real de soluções e combina mecanismos capazes de manter alta diversidade e promover exploração global do espaço de busca com um operador de busca local adaptativo, que é aplicado aos indivíduos mais promissores. Este mecanismo de busca local adapta tanto a sua probabilidade de execução quanto a quantidade de esforços computacionais que serão empregados a cada execução com base no *fitness* de cada indivíduo, favorecendo as melhores soluções.

Seguindo o paradigma de evolução diferencial (Price et al., 2005), dois algoritmos fizeram parte da competição CEC'2005: o chamado algoritmo *clássico* de evolução diferencial, DE (Price et al., 2005; Rönkkönen et al., 2005), e o algoritmo L-SaDE, proposto por Qin & Suganthan (2005). A principal diferença introduzida pelo L-SaDE, quando comparado ao tradicional DE, é a capacidade de selecionar automaticamente, ao longo da execução, qual a melhor estratégia de aprendizado (mutação) e quais os parâmetros que devem ser utilizados tanto na mutação quanto no operador de cruzamento.

O algoritmo DMS-L-PSO (Liang & Suganthan, 2005), por sua vez, se baseia no paradigma de otimização por enxame de partículas (Kennedy et al., 2001), e tem como principais diferenciais a utilização de múltiplos enxames de partículas, formados a partir da separação dos indivíduos presentes na população. Estes múltiplos enxames constantemente trocam informações entre si e são agrupados e redivididos ao longo do processo de busca. Além disso, o algoritmo DMS-L-PSO adota um método quase-Newton como operador de busca local.

Por fim, o último participante da competição CEC'2005 é o algoritmo evolutivo CoEVO (Pošík, 2005), que tem como principal característica a utilização de um processo de coevolução cooperativa (Paredis, 1995) que permite evoluir tanto uma população de soluções candidatas para o problema sendo otimizado quanto uma população de *passos* para o operador de mutação. Ao adotar esta estratégia, o CoEVO busca ajustar dinamicamente seu operador de mutação ao longo da busca, conforme a evolução das soluções e características do problema sendo tratado.

Além destes onze algoritmos descritos acima e da cob-aiNet, proposta nesta tese, os experimentos com os 25 problemas da competição CEC'2005 também foram feitos com os algoritmos opt-aiNet e dopt-aiNet, previamente apresentados nas Seções 2.2.2 e 4.1.1, respectivamente.

### 4.3.2 Apresentação dos Resultados e Discussão

Os resultados correspondentes à média do erro absoluto obtido tanto pela cob-aiNet quanto pela opt-aiNet, dopt-aiNet e demais algoritmos participantes da competição CEC'2005 estão dados nas Tabelas 4.8, 4.9 e 4.10 para os experimentos com  $D = 10$ , e nas Tabelas 4.11, 4.12 e 4.13 para  $D = 30$ . Nestas tabelas, os resultados dos onze algoritmos que participaram da competição original foram extraídos da literatura (referências dadas na Seção 4.3.1), considerando-se sempre que possível

valores em notação científica com quatro casas após a vírgula<sup>8</sup>. Já os três algoritmos imunoinspirados foram implementados e executados seguindo a metodologia descrita no início desta seção. É importante destacar que, para o algoritmo DMS-L-PSO, os autores não reportaram em Liang & Suganthan (2005) resultados para os problemas  $F_{21}$  a  $F_{25}$  com  $D = 30$ , enquanto que, para o algoritmo L-SaDE (Qin & Suganthan, 2005), não há resultados na literatura para os problemas  $F_{16}$  a  $F_{25}$ , também para  $D = 30$ .

Para facilitar a comparação dos resultados obtidos pela *cob-aiNet* com os das demais ferramentas, todos os algoritmos foram ordenados, para cada um dos problemas tratados aqui, em ordem crescente de erros absolutos médios, e os *ranks* obtidos por cada algoritmo para cada problema podem ser vistos nas Tabelas 4.14 e 4.15. Para manter as comparações justas, como a precisão desejada na competição é de  $1 \cdot 10^{-8}$ , todos os valores de EAM inferiores a este limiar foram considerados nulos antes de se fazer a ordenação, evitando-se assim a penalização daquelas ferramentas que encerraram sua execução ao se atingir a precisão desejada.

Observando os *ranks* médios obtidos para cada algoritmo, é possível notar que, tanto para  $D = 10$  quanto para  $D = 30$ , a *cob-aiNet* apresentou os melhores resultados para seis dos 25 problemas de teste, ficando respectivamente nas sétima e sexta posições. Em ambos os casos, a *cob-aiNet* foi consistentemente melhor que os outros dois SIAs (*opt-aiNet* e *dopt-aiNet*) em praticamente todos os problemas estudados, o que corrobora os resultados obtidos na Seção 4.1. Além disso, o *rank* médio obtido pela *cob-aiNet* também foi superior ao dos algoritmos CoEVO e K-PCX, tanto para  $D = 10$  quanto para  $D = 30$ .

Especificamente para o caso  $D = 10$ , a *cob-aiNet* apresentou um *rank* médio melhor quando comparada aos algoritmos BLX-MA, L-CMA-ES e EDA, além dos quatro algoritmos mencionados anteriormente, enquanto que com  $D = 30$  o algoritmo proposto nesta tese superou também as ferramentas DE, L-SaDE e DMS-L-PSO, empatando com o SPC-PNX.

Analisando-se apenas o comportamento da *cob-aiNet*, é possível perceber que, quando a dimensão foi aumentada de  $D = 10$  para  $D = 30$ , o *rank* obtido pelo algoritmo melhorou para a maioria dos problemas estudados, o que indica que, em média, a *cob-aiNet* tende a ser pouco susceptível a esta variação na dimensão do espaço de busca. Por outro lado, existem também alguns problemas para os quais foram observadas piores de desempenho ao se aumentar a dimensão do espaço de busca, dentre os quais se destacam principalmente  $F_{12}$  e  $F_{18}$ , para os quais a *cob-aiNet* teve quedas de sete e seis posições no *rank*, respectivamente. No entanto, é importante ressaltar que, para o problema  $F_{18}$ , diversos algoritmos apresentaram resultados próximos, o que faz com que pequenas variações no EAM obtido por uma determinada ferramenta leve a grandes variações em sua posição no *rank*

---

<sup>8</sup>Apesar desta tentativa de sempre se considerar quatro casas após a vírgula em notação científica, é importante ressaltar que os resultados reportados na literatura para os algoritmos DE, G-CMA-ES, K-PCS e L-CMA-ES apresentam apenas *duas casas*, o que pode ter alguma influência no momento de ordenação dos resultados de cada algoritmo.

Tab. 4.8: Média  $\pm$  Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com  $D = 10$  e as melhores soluções obtidas pelos algoritmos BLX-GL50, BLX-MA, CoEVO, DE e DMS-L-PSO. Os melhores resultados estão em negrito.

	BLX-GL50	BLX-MA	CoEVO	DE	DMS-L-PSO
<b>F<sub>1</sub></b>	$8,3446 \cdot 10^{-9} \pm 1,4066 \cdot 10^{-9}$	$9,8695 \cdot 10^{-9} \pm 6,6174 \cdot 10^{-25}$	$8,8336 \cdot 10^{-9} \pm 1,3332 \cdot 10^{-9}$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$
<b>F<sub>2</sub></b>	$8,2075 \cdot 10^{-9} \pm 1,1765 \cdot 10^{-9}$	$9,9360 \cdot 10^{-9} \pm 0,0000 \cdot 10^0$	$8,5987 \cdot 10^{-9} \pm 1,3665 \cdot 10^{-9}$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$	$1,2960 \cdot 10^{-13} \pm 1,5612 \cdot 10^{13}$
<b>F<sub>3</sub></b>	$5,7053 \cdot 10^2 \pm 9,3168 \cdot 10^2$	$4,7709 \cdot 10^4 \pm 6,4965 \cdot 10^3$	$8,4856 \cdot 10^{-9} \pm 1,3452 \cdot 10^{-9}$	$1,0900 \cdot 10^1 \pm 4,6300 \cdot 10^{-6}$	$7,0064 \cdot 10^{-9} \pm 2,6589 \cdot 10^9$
<b>F<sub>4</sub></b>	$8,3177 \cdot 10^{-9} \pm 1,4172 \cdot 10^{-9}$	$1,9967 \cdot 10^{-8} \pm 2,5261 \cdot 10^{-9}$	$8,5471 \cdot 10^{-9} \pm 9,8513 \cdot 10^{-10}$	$9,0900 \cdot 10^{-15} \pm 3,1500 \cdot 10^{-14}$	$1,8851 \cdot 10^{-3} \pm 1,8932 \cdot 10^3$
<b>F<sub>5</sub></b>	$8,9375 \cdot 10^{-9} \pm 1,0046 \cdot 10^{-9}$	$2,1242 \cdot 10^{-2} \pm 1,8794 \cdot 10^{-2}$	$2,1332 \cdot 10^0 \pm 3,5736 \cdot 10^0$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$	$1,1383 \cdot 10^{-6} \pm 2,1828 \cdot 10^{-6}$
<b>F<sub>6</sub></b>	$8,8737 \cdot 10^{-9} \pm 1,4790 \cdot 10^{-9}$	$1,4596 \cdot 10^0 \pm 5,4022 \cdot 10^{-2}$	$1,2463 \cdot 10^1 \pm 8,9750 \cdot 10^0$	$1,5900 \cdot 10^{-1} \pm 7,9700 \cdot 10^{-1}$	$6,8925 \cdot 10^{-8} \pm 3,1904 \cdot 10^7$
<b>F<sub>7</sub></b>	$1,1723 \cdot 10^{-2} \pm 7,6974 \cdot 10^{-3}$	$1,9714 \cdot 10^{-1} \pm 2,0663 \cdot 10^{-2}$	$3,7051 \cdot 10^{-2} \pm 1,5106 \cdot 10^{-2}$	$1,4600 \cdot 10^{-1} \pm 1,3800 \cdot 10^{-1}$	$4,5189 \cdot 10^{-2} \pm 3,2611 \cdot 10^{-2}$
<b>F<sub>8</sub></b>	$2,0354 \cdot 10^1 \pm 8,0547 \cdot 10^{-2}$	$2,0192 \cdot 10^1 \pm 1,3761 \cdot 10^{-2}$	$2,0275 \cdot 10^1 \pm 7,6588 \cdot 10^{-2}$	$2,0400 \cdot 10^1 \pm 7,5800 \cdot 10^{-2}$	$2,0000 \cdot 10^1 \pm 5,5382 \cdot 10^{-9}$
<b>F<sub>9</sub></b>	$1,1542 \cdot 10^0 \pm 7,9597 \cdot 10^{-1}$	$4,3786 \cdot 10^{-1} \pm 9,8763 \cdot 10^{-2}$	$1,9195 \cdot 10^1 \pm 6,6191 \cdot 10^0$	$9,5500 \cdot 10^{-1} \pm 9,7300 \cdot 10^{-1}$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$
<b>F<sub>10</sub></b>	$4,9748 \cdot 10^0 \pm 2,7698 \cdot 10^0$	$5,6429 \cdot 10^0 \pm 3,6183 \cdot 10^{-1}$	$2,6765 \cdot 10^1 \pm 8,6457 \cdot 10^0$	$1,2500 \cdot 10^1 \pm 7,9600 \cdot 10^0$	$3,6217 \cdot 10^0 \pm 8,5590 \cdot 10^{-1}$
<b>F<sub>11</sub></b>	$2,3340 \cdot 10^0 \pm 1,6064 \cdot 10^0$	$4,5574 \cdot 10^0 \pm 5,2435 \cdot 10^{-1}$	$9,0289 \cdot 10^0 \pm 1,0202 \cdot 10^0$	$8,4700 \cdot 10^{-1} \pm 1,4000 \cdot 10^0$	$4,6229 \cdot 10^0 \pm 5,8400 \cdot 10^{-1}$
<b>F<sub>12</sub></b>	$4,0691 \cdot 10^2 \pm 8,8891 \cdot 10^2$	$7,4303 \cdot 10^1 \pm 4,4314 \cdot 10^1$	$6,0453 \cdot 10^2 \pm 1,0736 \cdot 10^3$	$3,1700 \cdot 10^1 \pm 1,4200 \cdot 10^2$	$2,4007 \cdot 10^0 \pm 4,3602 \cdot 10^0$
<b>F<sub>13</sub></b>	$7,4985 \cdot 10^{-1} \pm 2,1007 \cdot 10^{-1}$	$7,7356 \cdot 10^{-1} \pm 4,2796 \cdot 10^{-2}$	$1,1365 \cdot 10^0 \pm 4,3310 \cdot 10^{-1}$	$9,7700 \cdot 10^{-1} \pm 4,6700 \cdot 10^{-1}$	$3,6865 \cdot 10^{-1} \pm 5,6411 \cdot 10^{-2}$
<b>F<sub>14</sub></b>	$2,1725 \cdot 10^0 \pm 4,8722 \cdot 10^{-1}$	$2,0305 \cdot 10^0 \pm 1,4427 \cdot 10^{-1}$	$3,7065 \cdot 10^0 \pm 2,3036 \cdot 10^{-1}$	$3,4500 \cdot 10^0 \pm 4,4000 \cdot 10^{-1}$	$2,3601 \cdot 10^0 \pm 3,3750 \cdot 10^{-1}$
<b>F<sub>15</sub></b>	$4,0000 \cdot 10^2 \pm 0,0000 \cdot 10^0$	$2,6957 \cdot 10^2 \pm 3,2543 \cdot 10^1$	$2,9377 \cdot 10^2 \pm 1,0207 \cdot 10^2$	$2,5900 \cdot 10^2 \pm 1,8300 \cdot 10^2$	$4,8539 \cdot 10^0 \pm 1,3415 \cdot 10^1$
<b>F<sub>16</sub></b>	$9,3493 \cdot 10^1 \pm 5,8680 \cdot 10^0$	$1,0156 \cdot 10^2 \pm 9,9066 \cdot 10^{-1}$	$1,7718 \cdot 10^2 \pm 3,1573 \cdot 10^1$	$1,1300 \cdot 10^2 \pm 1,8000 \cdot 10^1$	$9,4756 \cdot 10^1 \pm 1,0086 \cdot 10^1$
<b>F<sub>17</sub></b>	$1,0902 \cdot 10^2 \pm 7,8033 \cdot 10^0$	$1,2696 \cdot 10^2 \pm 2,2044 \cdot 10^0$	$2,1182 \cdot 10^2 \pm 3,3746 \cdot 10^1$	$1,1500 \cdot 10^2 \pm 2,0100 \cdot 10^1$	$1,1009 \cdot 10^2 \pm 4,3453 \cdot 10^0$
<b>F<sub>18</sub></b>	$4,2002 \cdot 10^2 \pm 2,1798 \cdot 10^2$	$8,0385 \cdot 10^2 \pm 3,2808 \cdot 10^0$	$9,0154 \cdot 10^2 \pm 8,7050 \cdot 10^1$	$4,0000 \cdot 10^2 \pm 2,0400 \cdot 10^2$	$7,6067 \cdot 10^2 \pm 1,8458 \cdot 10^2$
<b>F<sub>19</sub></b>	$4,4905 \cdot 10^2 \pm 2,4767 \cdot 10^2$	$7,6279 \cdot 10^2 \pm 2,7430 \cdot 10^1$	$8,4450 \cdot 10^2 \pm 1,3374 \cdot 10^2$	$4,2000 \cdot 10^2 \pm 2,1800 \cdot 10^2$	$7,1430 \cdot 10^2 \pm 2,0105 \cdot 10^2$
<b>F<sub>20</sub></b>	$4,4599 \cdot 10^2 \pm 2,4061 \cdot 10^2$	$8,0000 \cdot 10^2 \pm 0,0000 \cdot 10^0$	$8,6290 \cdot 10^2 \pm 1,4734 \cdot 10^2$	$4,6000 \cdot 10^2 \pm 2,3800 \cdot 10^2$	$8,2196 \cdot 10^2 \pm 4,5874 \cdot 10^1$
<b>F<sub>21</sub></b>	$6,8933 \cdot 10^2 \pm 2,9841 \cdot 10^2$	$7,2180 \cdot 10^2 \pm 2,5645 \cdot 10^1$	$6,3493 \cdot 10^2 \pm 3,0040 \cdot 10^2$	$4,9200 \cdot 10^2 \pm 4,0000 \cdot 10^1$	$5,3600 \cdot 10^2 \pm 2,1772 \cdot 10^2$
<b>F<sub>22</sub></b>	$7,5865 \cdot 10^2 \pm 2,8614 \cdot 10^1$	$6,7089 \cdot 10^2 \pm 3,7244 \cdot 10^1$	$7,7886 \cdot 10^2 \pm 1,0200 \cdot 10^2$	$7,1800 \cdot 10^2 \pm 1,5800 \cdot 10^2$	$6,9242 \cdot 10^2 \pm 1,5647 \cdot 10^2$
<b>F<sub>23</sub></b>	$6,3887 \cdot 10^2 \pm 2,0620 \cdot 10^2$	$9,2673 \cdot 10^2 \pm 2,4322 \cdot 10^1$	$8,3456 \cdot 10^2 \pm 1,6418 \cdot 10^2$	$5,7200 \cdot 10^2 \pm 4,4800 \cdot 10^1$	$7,3034 \cdot 10^2 \pm 1,6620 \cdot 10^2$
<b>F<sub>24</sub></b>	$2,0000 \cdot 10^2 \pm 0,0000 \cdot 10^0$	$2,2400 \cdot 10^2 \pm 1,6277 \cdot 10^1$	$3,1384 \cdot 10^2 \pm 2,1925 \cdot 10^2$	$2,0000 \cdot 10^2 \pm 0,0000 \cdot 10^0$	$2,2400 \cdot 10^2 \pm 8,3066 \cdot 10^1$
<b>F<sub>25</sub></b>	$4,0357 \cdot 10^2 \pm 4,6781 \cdot 10^0$	$3,9573 \cdot 10^2 \pm 7,9224 \cdot 10^0$	$2,5732 \cdot 10^2 \pm 1,5531 \cdot 10^2$	$9,2300 \cdot 10^2 \pm 3,4000 \cdot 10^{-1}$	$3,6571 \cdot 10^2 \pm 1,5096 \cdot 10^2$

Tab. 4.9: Média  $\pm$  Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com  $D = 10$  e as melhores soluções obtidas pelos algoritmos EDA, G-CMA-ES, K-PCX, L-CMA-ES e L-SaDE. Os melhores resultados estão em negrito. Esta tabela é continuação da Tabela 4.8.

	EDA	G-CMA-ES	K-PCX	L-CMA-ES	L-SaDE
F <sub>1</sub>	0,0000 · 10 <sup>0</sup> ± 0,0000 · 10 <sup>0</sup>	5,2000 · 10 <sup>-9</sup> ± 1,9400 · 10 <sup>-9</sup>	8,7100 · 10 <sup>-9</sup> ± 1,2200 · 10 <sup>-9</sup>	5,1400 · 10 <sup>-9</sup> ± 1,8200 · 10 <sup>-9</sup>	0,0000 · 10 <sup>0</sup> ± 0,0000 · 10 <sup>0</sup>
F <sub>2</sub>	0,0000 · 10 <sup>0</sup> ± 0,0000 · 10 <sup>0</sup>	4,7000 · 10 <sup>-9</sup> ± 1,5600 · 10 <sup>-9</sup>	9,4000 · 10 <sup>-9</sup> ± 5,3700 · 10 <sup>-10</sup>	5,3100 · 10 <sup>-9</sup> ± 1,7700 · 10 <sup>-9</sup>	1,0459 · 10 <sup>-13</sup> ± 5,1124 · 10 <sup>-13</sup>
F <sub>3</sub>	2,1207 · 10 <sup>1</sup> ± 8,0018 · 10 <sup>1</sup>	5,6000 · 10 <sup>-9</sup> ± 1,9300 · 10 <sup>-9</sup>	3,0200 · 10 <sup>4</sup> ± 2,3800 · 10 <sup>4</sup>	4,9400 · 10 <sup>-9</sup> ± 1,4500 · 10 <sup>-9</sup>	1,6720 · 10 <sup>-5</sup> ± 3,1196 · 10 <sup>-5</sup>
F <sub>4</sub>	0,0000 · 10 <sup>0</sup> ± 0,0000 · 10 <sup>0</sup>	5,0200 · 10 <sup>-9</sup> ± 1,7100 · 10 <sup>-9</sup>	7,9400 · 10 <sup>-7</sup> ± 1,9700 · 10 <sup>-6</sup>	1,7900 · 10 <sup>6</sup> ± 4,6600 · 10 <sup>6</sup>	1,4182 · 10 <sup>-5</sup> ± 7,0912 · 10 <sup>-5</sup>
F <sub>5</sub>	2,8376 · 10 <sup>-12</sup> ± 1,0606 · 10 <sup>-12</sup>	6,5800 · 10 <sup>-9</sup> ± 2,1700 · 10 <sup>-9</sup>	4,8500 · 10 <sup>1</sup> ± 5,1700 · 10 <sup>1</sup>	6,5700 · 10 <sup>-9</sup> ± 1,8800 · 10 <sup>-9</sup>	1,2300 · 10 <sup>-2</sup> ± 1,4600 · 10 <sup>-2</sup>
F <sub>6</sub>	4,1817 · 10 <sup>-2</sup> ± 1,5075 · 10 <sup>-1</sup>	4,8700 · 10 <sup>-9</sup> ± 1,6600 · 10 <sup>-9</sup>	2,0700 · 10 <sup>1</sup> ± 3,5600 · 10 <sup>1</sup>	5,4100 · 10 <sup>-9</sup> ± 1,8100 · 10 <sup>-9</sup>	1,1987 · 10 <sup>-8</sup> ± 1,9282 · 10 <sup>-8</sup>
F <sub>7</sub>	4,2049 · 10 <sup>-1</sup> ± 1,3303 · 10 <sup>-1</sup>	3,3100 · 10 <sup>-9</sup> ± 2,0200 · 10 <sup>-9</sup>	6,4000 · 10 <sup>-2</sup> ± 5,1500 · 10 <sup>-2</sup>	4,9100 · 10 <sup>-9</sup> ± 1,6800 · 10 <sup>-9</sup>	1,9900 · 10 <sup>-2</sup> ± 1,0700 · 10 <sup>-2</sup>
F <sub>8</sub>	2,0344 · 10 <sup>1</sup> ± 6,4467 · 10 <sup>-2</sup>	2,0000 · 10 <sup>1</sup> ± 3,8900 · 10 <sup>-3</sup>	2,0000 · 10 <sup>1</sup> ± 3,2500 · 10 <sup>-7</sup>	2,0000 · 10 <sup>1</sup> ± 0,0000 · 10 <sup>0</sup>	2,0000 · 10 <sup>1</sup> ± 5,3901 · 10 <sup>-8</sup>
F <sub>9</sub>	5,4179 · 10 <sup>0</sup> ± 1,9107 · 10 <sup>0</sup>	2,3900 · 10 <sup>-1</sup> ± 4,3400 · 10 <sup>-1</sup>	1,1900 · 10 <sup>-1</sup> ± 5,9700 · 10 <sup>-1</sup>	4,4900 · 10 <sup>1</sup> ± 1,3600 · 10 <sup>1</sup>	0,0000 · 10 <sup>0</sup> ± 0,0000 · 10 <sup>0</sup>
F <sub>10</sub>	5,2891 · 10 <sup>0</sup> ± 2,7737 · 10 <sup>0</sup>	7,9600 · 10 <sup>-2</sup> ± 2,7500 · 10 <sup>-1</sup>	2,3900 · 10 <sup>-1</sup> ± 7,2000 · 10 <sup>-1</sup>	4,0800 · 10 <sup>1</sup> ± 3,3500 · 10 <sup>1</sup>	4,9685 · 10 <sup>0</sup> ± 1,6918 · 10 <sup>0</sup>
F <sub>11</sub>	3,9446 · 10 <sup>0</sup> ± 3,1202 · 10 <sup>0</sup>	9,3400 · 10 <sup>-1</sup> ± 9,0000 · 10 <sup>-1</sup>	9,1100 · 10 <sup>0</sup> ± 5,2700 · 10 <sup>-1</sup>	3,6500 · 10 <sup>0</sup> ± 1,6600 · 10 <sup>0</sup>	4,8909 · 10 <sup>0</sup> ± 6,6190 · 10 <sup>-1</sup>
F <sub>12</sub>	4,4231 · 10 <sup>2</sup> ± 1,1692 · 10 <sup>3</sup>	2,9300 · 10 <sup>1</sup> ± 1,4200 · 10 <sup>2</sup>	2,4400 · 10 <sup>4</sup> ± 8,9200 · 10 <sup>3</sup>	2,0900 · 10 <sup>2</sup> ± 4,6900 · 10 <sup>2</sup>	4,5011 · 10 <sup>-7</sup> ± 8,5062 · 10 <sup>-7</sup>
F <sub>13</sub>	1,8412 · 10 <sup>0</sup> ± 3,4013 · 10 <sup>-1</sup>	6,9600 · 10 <sup>-1</sup> ± 1,5000 · 10 <sup>-1</sup>	6,5300 · 10 <sup>-1</sup> ± 2,0600 · 10 <sup>-1</sup>	4,9400 · 10 <sup>-1</sup> ± 1,3800 · 10 <sup>-1</sup>	2,2020 · 10 <sup>-1</sup> ± 4,1100 · 10 <sup>-2</sup>
F <sub>14</sub>	2,6298 · 10 <sup>0</sup> ± 3,9422 · 10 <sup>-1</sup>	3,0100 · 10 <sup>0</sup> ± 3,4900 · 10 <sup>-1</sup>	2,3500 · 10 <sup>0</sup> ± 4,8200 · 10 <sup>-1</sup>	4,0100 · 10 <sup>0</sup> ± 3,1400 · 10 <sup>-1</sup>	2,9153 · 10 <sup>0</sup> ± 2,0630 · 10 <sup>-1</sup>
F <sub>15</sub>	3,6500 · 10 <sup>2</sup> ± 9,2291 · 10 <sup>1</sup>	2,2800 · 10 <sup>2</sup> ± 6,8000 · 10 <sup>1</sup>	5,1000 · 10 <sup>2</sup> ± 1,2700 · 10 <sup>2</sup>	2,1100 · 10 <sup>2</sup> ± 1,0200 · 10 <sup>2</sup>	3,2000 · 10 <sup>1</sup> ± 1,1076 · 10 <sup>2</sup>
F <sub>16</sub>	1,4392 · 10 <sup>2</sup> ± 1,3678 · 10 <sup>1</sup>	9,1300 · 10 <sup>1</sup> ± 3,4900 · 10 <sup>0</sup>	9,5900 · 10 <sup>1</sup> ± 6,8500 · 10 <sup>0</sup>	1,0500 · 10 <sup>2</sup> ± 1,2600 · 10 <sup>1</sup>	1,0121 · 10 <sup>2</sup> ± 6,1686 · 10 <sup>0</sup>
F <sub>17</sub>	1,5679 · 10 <sup>2</sup> ± 1,0050 · 10 <sup>1</sup>	1,2300 · 10 <sup>2</sup> ± 2,0900 · 10 <sup>1</sup>	9,7300 · 10 <sup>1</sup> ± 5,5500 · 10 <sup>0</sup>	5,4900 · 10 <sup>2</sup> ± 3,4900 · 10 <sup>2</sup>	1,1406 · 10 <sup>2</sup> ± 9,9679 · 10 <sup>0</sup>
F <sub>18</sub>	4,8319 · 10 <sup>2</sup> ± 2,4302 · 10 <sup>2</sup>	3,3200 · 10 <sup>2</sup> ± 1,1200 · 10 <sup>2</sup>	7,5200 · 10 <sup>2</sup> ± 2,0700 · 10 <sup>2</sup>	4,9700 · 10 <sup>2</sup> ± 2,1800 · 10 <sup>2</sup>	7,1939 · 10 <sup>2</sup> ± 2,0852 · 10 <sup>2</sup>
F <sub>19</sub>	5,6438 · 10 <sup>2</sup> ± 2,5071 · 10 <sup>2</sup>	3,2600 · 10 <sup>2</sup> ± 9,9300 · 10 <sup>1</sup>	7,5100 · 10 <sup>2</sup> ± 2,0400 · 10 <sup>2</sup>	5,1600 · 10 <sup>2</sup> ± 2,3400 · 10 <sup>2</sup>	7,0494 · 10 <sup>2</sup> ± 1,9040 · 10 <sup>2</sup>
F <sub>20</sub>	6,5190 · 10 <sup>2</sup> ± 2,3809 · 10 <sup>2</sup>	3,0000 · 10 <sup>2</sup> ± 0,0000 · 10 <sup>0</sup>	8,1300 · 10 <sup>2</sup> ± 1,6300 · 10 <sup>2</sup>	4,4200 · 10 <sup>2</sup> ± 2,0300 · 10 <sup>2</sup>	7,1302 · 10 <sup>2</sup> ± 2,0134 · 10 <sup>2</sup>
F <sub>21</sub>	4,8400 · 10 <sup>2</sup> ± 1,6753 · 10 <sup>2</sup>	5,0000 · 10 <sup>2</sup> ± 3,4800 · 10 <sup>-13</sup>	1,0500 · 10 <sup>3</sup> ± 1,1600 · 10 <sup>2</sup>	4,0400 · 10 <sup>2</sup> ± 1,2300 · 10 <sup>2</sup>	4,6400 · 10 <sup>2</sup> ± 1,5780 · 10 <sup>2</sup>
F <sub>22</sub>	7,7088 · 10 <sup>2</sup> ± 2,0094 · 10 <sup>1</sup>	7,2900 · 10 <sup>2</sup> ± 6,8600 · 10 <sup>0</sup>	6,5900 · 10 <sup>2</sup> ± 1,3300 · 10 <sup>2</sup>	7,4000 · 10 <sup>2</sup> ± 2,4900 · 10 <sup>1</sup>	7,3490 · 10 <sup>2</sup> ± 9,1523 · 10 <sup>1</sup>
F <sub>23</sub>	6,4052 · 10 <sup>2</sup> ± 1,3858 · 10 <sup>2</sup>	5,5900 · 10 <sup>2</sup> ± 3,2400 · 10 <sup>-11</sup>	1,0600 · 10 <sup>3</sup> ± 1,5000 · 10 <sup>2</sup>	7,9100 · 10 <sup>2</sup> ± 2,7900 · 10 <sup>2</sup>	6,6406 · 10 <sup>2</sup> ± 1,5266 · 10 <sup>2</sup>
F <sub>24</sub>	2,0000 · 10 <sup>2</sup> ± 0,0000 · 10 <sup>0</sup>	2,0000 · 10 <sup>2</sup> ± 2,2900 · 10 <sup>-6</sup>	4,0600 · 10 <sup>2</sup> ± 5,1300 · 10 <sup>-1</sup>	8,6500 · 10 <sup>2</sup> ± 6,3900 · 10 <sup>2</sup>	2,0000 · 10 <sup>2</sup> ± 0,0000 · 10 <sup>0</sup>
F <sub>25</sub>	3,7304 · 10 <sup>2</sup> ± 3,7179 · 10 <sup>0</sup>	3,7400 · 10 <sup>2</sup> ± 3,2200 · 10 <sup>0</sup>	4,0600 · 10 <sup>2</sup> ± 4,8800 · 10 <sup>-1</sup>	4,4200 · 10 <sup>2</sup> ± 3,5800 · 10 <sup>2</sup>	3,7586 · 10 <sup>2</sup> ± 3,1453 · 10 <sup>0</sup>

Tab. 4.10: Média  $\pm$  Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com  $D = 10$  e as melhores soluções obtidas pelos algoritmos SPC-PNX, opt-aiNet, dopt-aiNet e cob-aiNet. Os melhores resultados estão em negrito. Esta tabela é continuação das Tabelas 4.8 e 4.9.

	SPC-PNX		opt-aiNet		dopt-aiNet		cob-aiNet	
<b>F<sub>1</sub></b>	$8,8967 \cdot 10^{-9} \pm 9,3915 \cdot 10^{-10}$	$6,2698 \cdot 10^3 \pm 2,4151 \cdot 10^3$	$1,8793 \cdot 10^4 \pm 5,8715 \cdot 10^3$	$5,1807 \cdot 10^{-11} \pm 1,0086 \cdot 10^{-10}$				
<b>F<sub>2</sub></b>	$9,6317 \cdot 10^{-9} \pm 3,2989 \cdot 10^{-10}$	$8,1583 \cdot 10^3 \pm 2,9619 \cdot 10^3$	$1,7236 \cdot 10^4 \pm 5,9042 \cdot 10^3$	$5,5907 \cdot 10^{-10} \pm 1,3264 \cdot 10^{-10}$				
<b>F<sub>3</sub></b>	$1,0806 \cdot 10^5 \pm 8,7160 \cdot 10^4$	$2,4192 \cdot 10^7 \pm 2,2746 \cdot 10^7$	$2,0685 \cdot 10^8 \pm 1,1432 \cdot 10^8$	$4,1952 \cdot 10^4 \pm 2,9513 \cdot 10^4$				
<b>F<sub>4</sub></b>	$9,3788 \cdot 10^{-9} \pm 6,3274 \cdot 10^{-10}$	$1,4661 \cdot 10^4 \pm 4,1677 \cdot 10^3$	$2,3318 \cdot 10^4 \pm 8,1798 \cdot 10^3$	$1,1132 \cdot 10^3 \pm 1,2978 \cdot 10^3$				
<b>F<sub>5</sub></b>	$9,1535 \cdot 10^{-9} \pm 6,3186 \cdot 10^{-10}$	$1,2058 \cdot 10^4 \pm 1,9983 \cdot 10^3$	$1,8177 \cdot 10^4 \pm 3,0592 \cdot 10^3$	$2,2888 \cdot 10^{-2} \pm 9,4642 \cdot 10^{-3}$				
<b>F<sub>6</sub></b>	$1,8909 \cdot 10^1 \pm 3,9977 \cdot 10^1$	$8,6838 \cdot 10^8 \pm 6,7510 \cdot 10^8$	$4,3460 \cdot 10^9 \pm 2,6633 \cdot 10^9$	$2,0640 \cdot 10^2 \pm 3,2653 \cdot 10^2$				
<b>F<sub>7</sub></b>	$8,2610 \cdot 10^{-2} \pm 6,2418 \cdot 10^{-2}$	$2,7013 \cdot 10^1 \pm 9,5849 \cdot 10^0$	$3,3752 \cdot 10^3 \pm 2,7311 \cdot 10^2$	$3,4639 \cdot 10^{-1} \pm 3,8222 \cdot 10^{-1}$				
<b>F<sub>8</sub></b>	$2,0991 \cdot 10^1 \pm 5,7946 \cdot 10^{-2}$	$2,0363 \cdot 10^1 \pm 7,1961 \cdot 10^{-2}$	$2,0317 \cdot 10^1 \pm 6,5269 \cdot 10^{-2}$	$2,0029 \cdot 10^1 \pm 1,7140 \cdot 10^{-2}$				
<b>F<sub>9</sub></b>	$4,0196 \cdot 10^0 \pm 2,2703 \cdot 10^0$	$4,9142 \cdot 10^1 \pm 7,2519 \cdot 10^0$	$3,6143 \cdot 10^1 \pm 9,8170 \cdot 10^0$	$9,6315 \cdot 10^{-12} \pm 1,7295 \cdot 10^{-11}$				
<b>F<sub>10</sub></b>	$7,3044 \cdot 10^0 \pm 5,2116 \cdot 10^0$	$4,2088 \cdot 10^1 \pm 7,9482 \cdot 10^0$	$8,2344 \cdot 10^1 \pm 1,5718 \cdot 10^1$	$2,5492 \cdot 10^1 \pm 8,1799 \cdot 10^0$				
<b>F<sub>11</sub></b>	$1,9098 \cdot 10^0 \pm 1,1598 \cdot 10^0$	$8,8721 \cdot 10^0 \pm 6,9854 \cdot 10^{-1}$	$7,6796 \cdot 10^0 \pm 6,0765 \cdot 10^{-1}$	$5,5108 \cdot 10^0 \pm 7,7472 \cdot 10^{-1}$				
<b>F<sub>12</sub></b>	$2,5951 \cdot 10^2 \pm 4,8933 \cdot 10^2$	$3,4966 \cdot 10^3 \pm 1,1348 \cdot 10^3$	$3,5526 \cdot 10^3 \pm 1,8883 \cdot 10^3$	$2,7809 \cdot 10^1 \pm 3,4163 \cdot 10^1$				
<b>F<sub>13</sub></b>	$8,3793 \cdot 10^{-1} \pm 2,6913 \cdot 10^{-1}$	$3,2777 \cdot 10^0 \pm 5,5695 \cdot 10^{-1}$	$1,8836 \cdot 10^0 \pm 3,6872 \cdot 10^{-1}$	$8,2954 \cdot 10^{-1} \pm 3,0685 \cdot 10^{-1}$				
<b>F<sub>14</sub></b>	$3,0456 \cdot 10^0 \pm 4,3662 \cdot 10^{-1}$	$3,9324 \cdot 10^0 \pm 1,9022 \cdot 10^{-1}$	$4,2210 \cdot 10^0 \pm 2,0385 \cdot 10^{-1}$	$3,6064 \cdot 10^0 \pm 2,1196 \cdot 10^{-1}$				
<b>F<sub>15</sub></b>	$2,5376 \cdot 10^2 \pm 1,5052 \cdot 10^2$	$3,7623 \cdot 10^2 \pm 3,8332 \cdot 10^1$	$3,6264 \cdot 10^2 \pm 7,4990 \cdot 10^1$	<b><math>8,1303 \cdot 10^{-11} \pm 2,2823 \cdot 10^{-10}</math></b>				
<b>F<sub>16</sub></b>	$1,0962 \cdot 10^2 \pm 9,8654 \cdot 10^0$	$1,9856 \cdot 10^2 \pm 1,8538 \cdot 10^1$	$2,6952 \cdot 10^2 \pm 3,8471 \cdot 10^1$	$1,5783 \cdot 10^2 \pm 2,3846 \cdot 10^1$				
<b>F<sub>17</sub></b>	$1,1898 \cdot 10^2 \pm 1,0707 \cdot 10^1$	$2,3306 \cdot 10^2 \pm 2,1158 \cdot 10^1$	$3,6372 \cdot 10^2 \pm 5,6464 \cdot 10^1$	$1,6799 \cdot 10^2 \pm 2,1935 \cdot 10^1$				
<b>F<sub>18</sub></b>	$4,3956 \cdot 10^2 \pm 2,2494 \cdot 10^2$	$4,8185 \cdot 10^2 \pm 3,2129 \cdot 10^1$	$1,0405 \cdot 10^3 \pm 5,4846 \cdot 10^1$	$4,3450 \cdot 10^2 \pm 7,5443 \cdot 10^1$				
<b>F<sub>19</sub></b>	$3,8000 \cdot 10^2 \pm 1,8708 \cdot 10^2$	$4,9745 \cdot 10^2 \pm 3,8629 \cdot 10^1$	$1,0469 \cdot 10^3 \pm 5,0851 \cdot 10^1$	$4,7728 \cdot 10^2 \pm 5,0015 \cdot 10^1$				
<b>F<sub>20</sub></b>	$4,4000 \cdot 10^2 \pm 2,2913 \cdot 10^2$	$4,7682 \cdot 10^2 \pm 5,8753 \cdot 10^1$	$1,0168 \cdot 10^3 \pm 1,2905 \cdot 10^2$	$4,6706 \cdot 10^2 \pm 5,7662 \cdot 10^1$				
<b>F<sub>21</sub></b>	$6,8006 \cdot 10^2 \pm 2,6873 \cdot 10^2$	$4,0331 \cdot 10^2 \pm 5,6570 \cdot 10^1$	$1,2049 \cdot 10^3 \pm 1,1396 \cdot 10^2$	<b><math>2,5894 \cdot 10^2 \pm 9,6461 \cdot 10^1</math></b>				
<b>F<sub>22</sub></b>	$7,4927 \cdot 10^2 \pm 9,3700 \cdot 10^1$	$7,6760 \cdot 10^2 \pm 8,9542 \cdot 10^1$	$9,0155 \cdot 10^2 \pm 6,5569 \cdot 10^1$	$7,3428 \cdot 10^2 \pm 1,6413 \cdot 10^2$				
<b>F<sub>23</sub></b>	$5,7591 \cdot 10^2 \pm 8,2207 \cdot 10^1$	<b><math>4,5067 \cdot 10^2 \pm 4,9100 \cdot 10^1</math></b>	$1,1981 \cdot 10^3 \pm 1,4161 \cdot 10^2$	$4,8132 \cdot 10^2 \pm 6,4664 \cdot 10^1$				
<b>F<sub>24</sub></b>	<b><math>2,0000 \cdot 10^2 \pm 0,0000 \cdot 10^0</math></b>	$2,5254 \cdot 10^2 \pm 1,3927 \cdot 10^1$	$1,0258 \cdot 10^3 \pm 2,5088 \cdot 10^2$	<b><math>2,0000 \cdot 10^2 \pm 1,8741 \cdot 10^{-12}</math></b>				
<b>F<sub>25</sub></b>	$4,0601 \cdot 10^2 \pm 2,3848 \cdot 10^{-1}$	$3,6776 \cdot 10^2 \pm 6,3767 \cdot 10^1$	$1,2775 \cdot 10^3 \pm 1,0083 \cdot 10^2$	$3,9672 \cdot 10^2 \pm 1,3966 \cdot 10^1$				

Tab. 4.11: Média ± Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com  $D = 30$  e as melhores soluções obtidas pelos algoritmos BLX-GL50, BLX-MA, CoEVO, DE e DMS-I-PSO. Os melhores resultados estão em negrito.

	BLX-GL50	BLX-MA	CoEVO	DE	DMS-I-PSO
F <sub>1</sub>	8,8775 · 10 <sup>-9</sup> ± 9,8507 · 10 <sup>-10</sup>	9,3639 · 10 <sup>-9</sup> ± 1,8991 · 10 <sup>-10</sup>	7,9650 · 10 <sup>-1</sup> ± 2,4906 · 10 <sup>0</sup>	<b>0,0000 · 10<sup>0</sup></b> ± 0,0000 · 10 <sup>0</sup>	<b>0,0000 · 10<sup>0</sup></b> ± 0,0000 · 10 <sup>0</sup>
F <sub>2</sub>	9,8351 · 10 <sup>-9</sup> ± 1,7465 · 10 <sup>-10</sup>	8,7172 · 10 <sup>-6</sup> ± 7,4738 · 10 <sup>-7</sup>	4,4018 · 10 <sup>-1</sup> ± 1,5208 · 10 <sup>0</sup>	3,3300 · 10 <sup>-2</sup> ± 4,9000 · 10 <sup>-2</sup>	1,1757 · 10 <sup>-7</sup> ± 6,5592 · 10 <sup>-8</sup>
F <sub>3</sub>	3,1117 · 10 <sup>3</sup> ± 2,3263 · 10 <sup>3</sup>	8,7746 · 10 <sup>6</sup> ± 5,8100 · 10 <sup>4</sup>	3,6714 · 10 <sup>2</sup> ± 1,3453 · 10 <sup>3</sup>	6,9200 · 10 <sup>5</sup> ± 2,0400 · 10 <sup>5</sup>	1,6343 · 10 <sup>-6</sup> ± 3,9247 · 10 <sup>-6</sup>
F <sub>4</sub>	1,6844 · 10 <sup>1</sup> ± 9,9601 · 10 <sup>0</sup>	3,9663 · 10 <sup>1</sup> ± 8,5540 · 10 <sup>0</sup>	4,7978 · 10 <sup>3</sup> ± 3,4371 · 10 <sup>3</sup>	1,5200 · 10 <sup>1</sup> ± 1,8100 · 10 <sup>1</sup>	2,5487 · 10 <sup>3</sup> ± 3,0638 · 10 <sup>2</sup>
F <sub>5</sub>	3,3256 · 10 <sup>2</sup> ± 2,4743 · 10 <sup>2</sup>	2,1827 · 10 <sup>3</sup> ± 7,8313 · 10 <sup>1</sup>	8,3436 · 10 <sup>3</sup> ± 1,0413 · 10 <sup>3</sup>	1,7000 · 10 <sup>2</sup> ± 1,8400 · 10 <sup>2</sup>	2,1858 · 10 <sup>3</sup> ± 8,2641 · 10 <sup>2</sup>
F <sub>6</sub>	2,5982 · 10 <sup>-7</sup> ± 4,3227 · 10 <sup>-7</sup>	4,9546 · 10 <sup>1</sup> ± 1,9329 · 10 <sup>1</sup>	1,2111 · 10 <sup>3</sup> ± 1,8340 · 10 <sup>3</sup>	2,5100 · 10 <sup>1</sup> ± 2,9000 · 10 <sup>1</sup>	4,7840 · 10 <sup>-1</sup> ± 1,3222 · 10 <sup>0</sup>
F <sub>7</sub>	9,0730 · 10 <sup>-9</sup> ± 7,8752 · 10 <sup>-10</sup>	1,3292 · 10 <sup>-2</sup> ± 1,2720 · 10 <sup>-3</sup>	1,4120 · 10 <sup>-1</sup> ± 2,8329 · 10 <sup>-1</sup>	2,9600 · 10 <sup>-3</sup> ± 5,5500 · 10 <sup>-3</sup>	6,9990 · 10 <sup>-3</sup> ± 4,5371 · 10 <sup>-3</sup>
F <sub>8</sub>	2,0946 · 10 <sup>1</sup> ± 5,7488 · 10 <sup>-2</sup>	2,0710 · 10 <sup>1</sup> ± 3,7657 · 10 <sup>-2</sup>	2,0900 · 10 <sup>1</sup> ± 6,8202 · 10 <sup>-2</sup>	2,1000 · 10 <sup>1</sup> ± 5,1100 · 10 <sup>-2</sup>	<b>2,0000 · 10<sup>1</sup></b> ± <b>2,3029 · 10<sup>-4</sup></b>
F <sub>9</sub>	1,5105 · 10 <sup>1</sup> ± 5,0385 · 10 <sup>0</sup>	6,8064 · 10 <sup>-1</sup> ± 1,2144 · 10 <sup>-1</sup>	1,3135 · 10 <sup>2</sup> ± 2,3878 · 10 <sup>1</sup>	1,8500 · 10 <sup>1</sup> ± 5,2000 · 10 <sup>0</sup>	1,7591 · 10 <sup>1</sup> ± 3,0222 · 10 <sup>0</sup>
F <sub>10</sub>	3,5199 · 10 <sup>1</sup> ± 1,0321 · 10 <sup>1</sup>	9,0583 · 10 <sup>1</sup> ± 4,8923 · 10 <sup>0</sup>	2,3235 · 10 <sup>2</sup> ± 3,5060 · 10 <sup>1</sup>	9,6900 · 10 <sup>1</sup> ± 8,2300 · 10 <sup>1</sup>	3,7410 · 10 <sup>1</sup> ± 5,2883 · 10 <sup>0</sup>
F <sub>11</sub>	2,4741 · 10 <sup>1</sup> ± 3,5303 · 10 <sup>0</sup>	3,1137 · 10 <sup>1</sup> ± 1,5249 · 10 <sup>0</sup>	3,7701 · 10 <sup>1</sup> ± 1,5271 · 10 <sup>0</sup>	3,4200 · 10 <sup>1</sup> ± 1,0300 · 10 <sup>1</sup>	2,7278 · 10 <sup>1</sup> ± 1,5739 · 10 <sup>0</sup>
F <sub>12</sub>	9,5208 · 10 <sup>3</sup> ± 8,0700 · 10 <sup>3</sup>	4,3806 · 10 <sup>3</sup> ± 1,3863 · 10 <sup>3</sup>	1,0143 · 10 <sup>5</sup> ± 7,2229 · 10 <sup>4</sup>	2,7500 · 10 <sup>3</sup> ± 3,2200 · 10 <sup>3</sup>	<b>2,5359 · 10<sup>2</sup></b> ± <b>2,8883 · 10<sup>2</sup></b>
F <sub>13</sub>	5,1515 · 10 <sup>0</sup> ± 4,0209 · 10 <sup>0</sup>	3,9683 · 10 <sup>0</sup> ± 5,3815 · 10 <sup>-1</sup>	9,0196 · 10 <sup>0</sup> ± 2,2784 · 10 <sup>0</sup>	3,2300 · 10 <sup>0</sup> ± 8,2300 · 10 <sup>-1</sup>	2,3595 · 10 <sup>0</sup> ± 5,2023 · 10 <sup>-1</sup>
F <sub>14</sub>	1,2120 · 10 <sup>1</sup> ± 6,6871 · 10 <sup>-1</sup>	1,2563 · 10 <sup>1</sup> ± 7,8107 · 10 <sup>-2</sup>	1,3247 · 10 <sup>1</sup> ± 2,9231 · 10 <sup>-1</sup>	1,3400 · 10 <sup>1</sup> ± 1,4100 · 10 <sup>-1</sup>	<b>1,1961 · 10<sup>1</sup></b> ± <b>4,1146 · 10<sup>-1</sup></b>
F <sub>15</sub>	3,0400 · 10 <sup>2</sup> ± 7,3485 · 10 <sup>1</sup>	3,5610 · 10 <sup>2</sup> ± 1,5064 · 10 <sup>1</sup>	4,1063 · 10 <sup>2</sup> ± 4,3671 · 10 <sup>1</sup>	3,6000 · 10 <sup>2</sup> ± 1,0800 · 10 <sup>2</sup>	3,4400 · 10 <sup>2</sup> ± 5,0662 · 10 <sup>1</sup>
F <sub>16</sub>	8,8672 · 10 <sup>1</sup> ± 9,6921 · 10 <sup>1</sup>	3,2625 · 10 <sup>2</sup> ± 2,9967 · 10 <sup>1</sup>	3,8123 · 10 <sup>2</sup> ± 4,5382 · 10 <sup>1</sup>	<b>2,1200 · 10<sup>-2</sup></b> ± <b>1,1000 · 10<sup>2</sup></b>	1,1950 · 10 <sup>2</sup> ± 1,2068 · 10 <sup>2</sup>
F <sub>17</sub>	1,3496 · 10 <sup>2</sup> ± 1,1247 · 10 <sup>2</sup>	2,7927 · 10 <sup>2</sup> ± 2,7550 · 10 <sup>1</sup>	4,5439 · 10 <sup>2</sup> ± 5,6501 · 10 <sup>1</sup>	2,3700 · 10 <sup>2</sup> ± 1,2200 · 10 <sup>2</sup>	1,4519 · 10 <sup>2</sup> ± 7,3247 · 10 <sup>1</sup>
F <sub>18</sub>	9,0363 · 10 <sup>2</sup> ± 5,9543 · 10 <sup>-1</sup>	8,7757 · 10 <sup>2</sup> ± 3,2310 · 10 <sup>0</sup>	1,0609 · 10 <sup>3</sup> ± 3,0246 · 10 <sup>1</sup>	9,0400 · 10 <sup>2</sup> ± 3,1300 · 10 <sup>-1</sup>	9,1053 · 10 <sup>2</sup> ± 1,5761 · 10 <sup>0</sup>
F <sub>19</sub>	9,0363 · 10 <sup>2</sup> ± 5,1263 · 10 <sup>-1</sup>	8,8021 · 10 <sup>2</sup> ± 7,4025 · 10 <sup>-1</sup>	1,0489 · 10 <sup>3</sup> ± 2,9770 · 10 <sup>1</sup>	9,0400 · 10 <sup>2</sup> ± 6,2500 · 10 <sup>-1</sup>	9,1060 · 10 <sup>2</sup> ± 1,3383 · 10 <sup>0</sup>
F <sub>20</sub>	9,0346 · 10 <sup>2</sup> ± 2,4387 · 10 <sup>-1</sup>	8,7926 · 10 <sup>2</sup> ± 8,4788 · 10 <sup>-1</sup>	1,0587 · 10 <sup>3</sup> ± 4,3264 · 10 <sup>1</sup>	9,0400 · 10 <sup>2</sup> ± 6,1500 · 10 <sup>-1</sup>	9,0189 · 10 <sup>2</sup> ± 3,0719 · 10 <sup>1</sup>
F <sub>21</sub>	5,0000 · 10 <sup>2</sup> ± 8,7602 · 10 <sup>-14</sup>	5,0000 · 10 <sup>2</sup> ± 0,0000 · 10 <sup>0</sup>	6,0352 · 10 <sup>2</sup> ± 2,1808 · 10 <sup>2</sup>	5,0000 · 10 <sup>2</sup> ± 1,6600 · 10 <sup>-13</sup>	-
F <sub>22</sub>	8,7350 · 10 <sup>2</sup> ± 1,8248 · 10 <sup>1</sup>	9,5830 · 10 <sup>2</sup> ± 1,6933 · 10 <sup>0</sup>	1,1554 · 10 <sup>3</sup> ± 2,9410 · 10 <sup>1</sup>	8,9700 · 10 <sup>2</sup> ± 1,3300 · 10 <sup>1</sup>	-
F <sub>23</sub>	5,8714 · 10 <sup>2</sup> ± 7,7015 · 10 <sup>1</sup>	5,5010 · 10 <sup>2</sup> ± 2,4432 · 10 <sup>1</sup>	9,2167 · 10 <sup>2</sup> ± 1,8119 · 10 <sup>2</sup>	5,3400 · 10 <sup>2</sup> ± 4,2600 · 10 <sup>-4</sup>	-
F <sub>24</sub>	8,7694 · 10 <sup>2</sup> ± 2,5528 · 10 <sup>2</sup>	<b>2,0000 · 10<sup>2</sup></b> ± <b>0,0000 · 10<sup>0</sup></b>	1,0966 · 10 <sup>3</sup> ± 1,2076 · 10 <sup>2</sup>	<b>2,0000 · 10<sup>2</sup></b> ± <b>1,3500 · 10<sup>-12</sup></b>	-
F <sub>25</sub>	2,1060 · 10 <sup>2</sup> ± 2,1493 · 10 <sup>-1</sup>	2,1146 · 10 <sup>2</sup> ± 4,4821 · 10 <sup>-2</sup>	1,0267 · 10 <sup>3</sup> ± 1,9155 · 10 <sup>2</sup>	7,3000 · 10 <sup>2</sup> ± 3,7400 · 10 <sup>-1</sup>	-

Tab. 4.12: Média  $\pm$  Desvio Padrão dos erros absolutos entre o ótimo global de cada problema com  $D = 30$  e as melhores soluções obtidas pelos algoritmos EDA, G-CMA-ES, K-PCX, L-CMA-ES e L-SaDE. Os melhores resultados estão em negrito. Esta tabela é continuação da Tabela 4.11.

	EDA	G-CMA-ES	K-PCX	L-CMA-ES	L-SaDE
<b>F1</b>	$9,0949 \cdot 10^{-15} \pm 2,1269 \cdot 10^{-14}$	$5,4200 \cdot 10^{-9} \pm 9,8000 \cdot 10^{-10}$	$8,9500 \cdot 10^{-9} \pm 9,9000 \cdot 10^{-10}$	$5,2800 \cdot 10^{-9} \pm 9,8200 \cdot 10^{-10}$	$0,0000 \cdot 10^0 \pm 0,0000 \cdot 10^0$
<b>F2</b>	$9,0949 \cdot 10^{-15} \pm 2,1269 \cdot 10^{-14}$	$6,2200 \cdot 10^{-9} \pm 8,9500 \cdot 10^{-10}$	$1,4400 \cdot 10^{-2} \pm 9,8300 \cdot 10^{-3}$	$6,9300 \cdot 10^{-9} \pm 8,2700 \cdot 10^{-10}$	$9,7191 \cdot 10^{-8} \pm 4,8596 \cdot 10^{-7}$
<b>F3</b>	$1,7758 \cdot 10^{-12} \pm 1,7241 \cdot 10^{-12}$	$5,5500 \cdot 10^{-9} \pm 1,0900 \cdot 10^{-9}$	$5,0700 \cdot 10^{-5} \pm 1,8600 \cdot 10^{-5}$	$5,1800 \cdot 10^{-9} \pm 1,0300 \cdot 10^{-9}$	$5,0521 \cdot 10^{-4} \pm 1,5754 \cdot 10^{-5}$
<b>F4</b>	$2,2601 \cdot 10^{-12} \pm 1,9799 \cdot 10^{-12}$	$1,1100 \cdot 10^{-4} \pm 3,0200 \cdot 10^{-4}$	$1,1100 \cdot 10^{-5} \pm 4,3500 \cdot 10^{-2}$	$9,2600 \cdot 10^{-7} \pm 1,6800 \cdot 10^{-8}$	$5,8160 \cdot 10^{-6} \pm 1,4479 \cdot 10^{-5}$
<b>F5</b>	$5,5133 \cdot 10^{-2} \pm 2,5375 \cdot 10^{-2}$	$8,6200 \cdot 10^{-9} \pm 8,5300 \cdot 10^{-10}$	$2,0400 \cdot 10^{-5} \pm 3,8100 \cdot 10^{-2}$	$8,3000 \cdot 10^{-9} \pm 1,3800 \cdot 10^{-9}$	$7,8803 \cdot 10^2 \pm 1,2439 \cdot 10^3$
<b>F6</b>	$2,1103 \cdot 10^1 \pm 8,2756 \cdot 10^{-1}$	$5,9000 \cdot 10^{-9} \pm 1,6100 \cdot 10^{-9}$	$9,8900 \cdot 10^{-2} \pm 1,6900 \cdot 10^3$	$6,3100 \cdot 10^{-9} \pm 1,1400 \cdot 10^{-9}$	$2,1248 \cdot 10^1 \pm 1,3413 \cdot 10^1$
<b>F7</b>	$3,8654 \cdot 10^{-14} \pm 3,6582 \cdot 10^{-14}$	$5,3100 \cdot 10^{-9} \pm 1,4100 \cdot 10^{-9}$	$3,6300 \cdot 10^{-2} \pm 4,7700 \cdot 10^{-2}$	$6,4800 \cdot 10^{-9} \pm 1,1400 \cdot 10^{-9}$	$8,2727 \cdot 10^{-3} \pm 1,1445 \cdot 10^{-2}$
<b>F8</b>	$2,0945 \cdot 10^1 \pm 4,9832 \cdot 10^{-2}$	$2,0100 \cdot 10^{-1} \pm 2,7900 \cdot 10^{-1}$	$2,0000 \cdot 10^1 \pm 4,9700 \cdot 10^{-7}$	$2,0000 \cdot 10^1 \pm 9,6200 \cdot 10^{-15}$	$2,0140 \cdot 10^1 \pm 5,7258 \cdot 10^{-2}$
<b>F9</b>	$1,7868 \cdot 10^2 \pm 1,0242 \cdot 10^1$	$9,3800 \cdot 10^{-1} \pm 1,1800 \cdot 10^0$	$2,7900 \cdot 10^{-1} \pm 4,5600 \cdot 10^{-1}$	$2,9100 \cdot 10^2 \pm 3,5400 \cdot 10^1$	$2,2737 \cdot 10^{-15} \pm 1,1369 \cdot 10^{-14}$
<b>F10</b>	$1,8852 \cdot 10^2 \pm 1,9372 \cdot 10^1$	$1,6500 \cdot 10^0 \pm 1,3500 \cdot 10^0$	$5,1700 \cdot 10^{-1} \pm 7,1100 \cdot 10^{-1}$	$5,6300 \cdot 10^2 \pm 2,4800 \cdot 10^2$	$3,5758 \cdot 10^1 \pm 6,0809 \cdot 10^0$
<b>F11</b>	$3,9453 \cdot 10^1 \pm 1,2844 \cdot 10^0$	$5,4800 \cdot 10^0 \pm 3,1300 \cdot 10^0$	$2,9500 \cdot 10^1 \pm 3,6500 \cdot 10^0$	$1,5200 \cdot 10^1 \pm 3,5100 \cdot 10^0$	$2,6562 \cdot 10^1 \pm 1,1275 \cdot 10^0$
<b>F12</b>	$6,0542 \cdot 10^3 \pm 7,7717 \cdot 10^3$	$4,4300 \cdot 10^4 \pm 2,1900 \cdot 10^5$	$1,0400 \cdot 10^6 \pm 1,3200 \cdot 10^5$	$1,3200 \cdot 10^4 \pm 1,1500 \cdot 10^4$	$8,7345 \cdot 10^2 \pm 9,3383 \cdot 10^2$
<b>F13</b>	$1,5310 \cdot 10^1 \pm 9,5194 \cdot 10^{-1}$	$2,4900 \cdot 10^0 \pm 5,1300 \cdot 10^{-1}$	$1,1900 \cdot 10^1 \pm 3,8000 \cdot 10^0$	$2,3200 \cdot 10^0 \pm 3,4600 \cdot 10^{-1}$	$1,2070 \cdot 10^0 \pm 1,3420 \cdot 10^{-1}$
<b>F14</b>	$1,3321 \cdot 10^1 \pm 1,7390 \cdot 10^{-1}$	$1,2900 \cdot 10^1 \pm 4,1900 \cdot 10^{-1}$	$1,3800 \cdot 10^1 \pm 1,9000 \cdot 10^{-1}$	$1,4000 \cdot 10^1 \pm 4,0400 \cdot 10^{-1}$	$1,2360 \cdot 10^1 \pm 2,5936 \cdot 10^{-1}$
<b>F15</b>	$3,9600 \cdot 10^2 \pm 4,5455 \cdot 10^1$	$2,0800 \cdot 10^2 \pm 2,7500 \cdot 10^1$	$8,7600 \cdot 10^2 \pm 1,9500 \cdot 10^1$	$2,1600 \cdot 10^2 \pm 8,2900 \cdot 10^1$	$3,2775 \cdot 10^2 \pm 9,6450 \cdot 10^1$
<b>F16</b>	$2,0789 \cdot 10^2 \pm 2,6580 \cdot 10^1$	$3,5000 \cdot 10^1 \pm 2,0400 \cdot 10^1$	$7,1500 \cdot 10^1 \pm 8,1000 \cdot 10^1$	$5,8400 \cdot 10^1 \pm 7,2800 \cdot 10^0$	-
<b>F17</b>	$2,3759 \cdot 10^2 \pm 5,0793 \cdot 10^1$	$2,9100 \cdot 10^2 \pm 1,9300 \cdot 10^2$	$1,5600 \cdot 10^2 \pm 1,5800 \cdot 10^2$	$1,0700 \cdot 10^3 \pm 5,1300 \cdot 10^2$	-
<b>F18</b>	$8,4665 \cdot 10^2 \pm 5,3712 \cdot 10^1$	$9,0400 \cdot 10^2 \pm 2,8800 \cdot 10^{-1}$	$8,3000 \cdot 10^2 \pm 1,6100 \cdot 10^0$	$8,9000 \cdot 10^2 \pm 4,6000 \cdot 10^1$	-
<b>F19</b>	$8,4223 \cdot 10^2 \pm 5,2794 \cdot 10^1$	$9,0400 \cdot 10^2 \pm 2,7100 \cdot 10^{-1}$	$8,3100 \cdot 10^2 \pm 1,4700 \cdot 10^0$	$9,0300 \cdot 10^2 \pm 3,1100 \cdot 10^1$	-
<b>F20</b>	$8,5085 \cdot 10^2 \pm 5,4021 \cdot 10^1$	$9,0400 \cdot 10^2 \pm 2,4800 \cdot 10^{-1}$	$8,3100 \cdot 10^2 \pm 1,3200 \cdot 10^0$	$8,8900 \cdot 10^2 \pm 4,5500 \cdot 10^1$	-
<b>F21</b>	$5,0000 \cdot 10^2 \pm 3,3968 \cdot 10^{-13}$	$5,0000 \cdot 10^2 \pm 1,3100 \cdot 10^{-13}$	$8,5900 \cdot 10^2 \pm 5,4400 \cdot 10^{-1}$	$4,8500 \cdot 10^2 \pm 3,3900 \cdot 10^1$	-
<b>F22</b>	$8,7193 \cdot 10^2 \pm 1,1648 \cdot 10^1$	$8,0300 \cdot 10^2 \pm 1,8600 \cdot 10^1$	$1,5600 \cdot 10^3 \pm 4,8300 \cdot 10^2$	$8,7100 \cdot 10^2 \pm 2,1500 \cdot 10^1$	-
<b>F23</b>	$5,3416 \cdot 10^2 \pm 2,7077 \cdot 10^{-4}$	$5,3400 \cdot 10^2 \pm 2,2200 \cdot 10^{-4}$	$8,6600 \cdot 10^2 \pm 8,0700 \cdot 10^{-1}$	$5,3500 \cdot 10^2 \pm 1,5300 \cdot 10^0$	-
<b>F24</b>	$2,0000 \cdot 10^2 \pm 2,9008 \cdot 10^{-14}$	$9,1000 \cdot 10^2 \pm 1,4800 \cdot 10^2$	$2,1300 \cdot 10^2 \pm 3,8900 \cdot 10^{-1}$	$1,4100 \cdot 10^3 \pm 5,4400 \cdot 10^2$	-
<b>F25</b>	$2,0855 \cdot 10^2 \pm 5,2943 \cdot 10^{-2}$	$2,1100 \cdot 10^2 \pm 9,2100 \cdot 10^{-1}$	$2,1300 \cdot 10^2 \pm 5,5000 \cdot 10^{-1}$	$6,9100 \cdot 10^2 \pm 7,8700 \cdot 10^2$	-

Tab. 4.13: Média  $\pm$  Desvio Padrão dos erros absolutos entre o último global de cada problema com  $D = 30$  e as melhores soluções obtidas pelos algoritmos SPC-PNX, opt-aiNet, dopl-aiNet e cob-aiNet. Os melhores resultados estão em negrito. Esta tabela é continuação das Tabelas 4.11 e 4.12.

	SPC-PNX	opt-aiNet	dopl-aiNet	cob-aiNet
F <sub>1</sub>	9,3524 · 10 <sup>-9</sup> ± 4,6327 · 10 <sup>-10</sup>	6,5635 · 10 <sup>4</sup> ± 7,5591 · 10 <sup>3</sup>	9,9637 · 10 <sup>4</sup> ± 1,5074 · 10 <sup>4</sup>	4,8192 · 10 <sup>-11</sup> ± 2,5294 · 10 <sup>-11</sup>
F <sub>2</sub>	6,9482 · 10 <sup>-7</sup> ± 1,4911 · 10 <sup>-6</sup>	6,9055 · 10 <sup>4</sup> ± 1,1351 · 10 <sup>4</sup>	1,0773 · 10 <sup>5</sup> ± 3,0514 · 10 <sup>4</sup>	8,3417 · 10 <sup>-10</sup> ± 6,2444 · 10 <sup>-11</sup>
F <sub>3</sub>	1,1020 · 10 <sup>6</sup> ± 4,2081 · 10 <sup>5</sup>	7,1501 · 10 <sup>8</sup> ± 2,4312 · 10 <sup>8</sup>	2,0670 · 10 <sup>9</sup> ± 7,9586 · 10 <sup>8</sup>	9,5889 · 10 <sup>5</sup> ± 4,0960 · 10 <sup>5</sup>
F <sub>4</sub>	8,1320 · 10 <sup>-7</sup> ± 1,7457 · 10 <sup>-6</sup>	1,0124 · 10 <sup>5</sup> ± 1,2271 · 10 <sup>4</sup>	1,5929 · 10 <sup>5</sup> ± 4,1708 · 10 <sup>4</sup>	3,9601 · 10 <sup>4</sup> ± 5,9727 · 10 <sup>3</sup>
F <sub>5</sub>	4,2374 · 10 <sup>3</sup> ± 1,3752 · 10 <sup>3</sup>	3,1931 · 10 <sup>4</sup> ± 3,0706 · 10 <sup>3</sup>	4,3249 · 10 <sup>4</sup> ± 4,8666 · 10 <sup>3</sup>	6,7246 · 10 <sup>3</sup> ± 1,1534 · 10 <sup>3</sup>
F <sub>6</sub>	1,5197 · 10 <sup>1</sup> ± 1,4903 · 10 <sup>1</sup>	3,9875 · 10 <sup>10</sup> ± 8,2989 · 10 <sup>9</sup>	7,9607 · 10 <sup>10</sup> ± 1,8415 · 10 <sup>10</sup>	1,8111 · 10 <sup>2</sup> ± 4,1378 · 10 <sup>2</sup>
F <sub>7</sub>	1,4598 · 10 <sup>-2</sup> ± 1,2391 · 10 <sup>-2</sup>	4,0370 · 10 <sup>-1</sup> ± 6,0863 · 10 <sup>-2</sup>	1,1967 · 10 <sup>4</sup> ± 7,4855 · 10 <sup>2</sup>	9,0553 · 10 <sup>-3</sup> ± 9,6478 · 10 <sup>-3</sup>
F <sub>8</sub>	2,0932 · 10 <sup>1</sup> ± 4,5876 · 10 <sup>-2</sup>	2,0926 · 10 <sup>1</sup> ± 6,1775 · 10 <sup>-2</sup>	2,0952 · 10 <sup>1</sup> ± 4,4543 · 10 <sup>-2</sup>	2,0128 · 10 <sup>1</sup> ± 2,4342 · 10 <sup>-2</sup>
F <sub>9</sub>	2,3934 · 10 <sup>1</sup> ± 6,2477 · 10 <sup>0</sup>	3,3259 · 10 <sup>2</sup> ± 2,0030 · 10 <sup>1</sup>	3,1800 · 10 <sup>2</sup> ± 3,1418 · 10 <sup>1</sup>	6,8244 · 10 <sup>-11</sup> ± 6,0851 · 10 <sup>-11</sup>
F <sub>10</sub>	6,0297 · 10 <sup>1</sup> ± 4,0576 · 10 <sup>1</sup>	3,3290 · 10 <sup>2</sup> ± 2,0739 · 10 <sup>1</sup>	5,9156 · 10 <sup>2</sup> ± 6,1634 · 10 <sup>1</sup>	1,2966 · 10 <sup>2</sup> ± 2,5193 · 10 <sup>1</sup>
F <sub>11</sub>	1,1255 · 10 <sup>1</sup> ± 3,2979 · 10 <sup>0</sup>	3,9446 · 10 <sup>1</sup> ± 1,4409 · 10 <sup>0</sup>	3,3516 · 10 <sup>1</sup> ± 1,3824 · 10 <sup>0</sup>	2,7778 · 10 <sup>1</sup> ± 1,3818 · 10 <sup>0</sup>
F <sub>12</sub>	1,3134 · 10 <sup>4</sup> ± 1,3346 · 10 <sup>4</sup>	2,2025 · 10 <sup>5</sup> ± 2,5887 · 10 <sup>4</sup>	2,5512 · 10 <sup>5</sup> ± 5,4450 · 10 <sup>4</sup>	7,2376 · 10 <sup>4</sup> ± 1,4671 · 10 <sup>4</sup>
F <sub>13</sub>	3,5881 · 10 <sup>0</sup> ± 1,0857 · 10 <sup>0</sup>	3,0804 · 10 <sup>1</sup> ± 2,3495 · 10 <sup>0</sup>	3,5631 · 10 <sup>1</sup> ± 7,7582 · 10 <sup>0</sup>	3,4531 · 10 <sup>0</sup> ± 6,8894 · 10 <sup>-1</sup>
F <sub>14</sub>	1,3131 · 10 <sup>1</sup> ± 2,6887 · 10 <sup>-1</sup>	1,3571 · 10 <sup>1</sup> ± 1,7278 · 10 <sup>-1</sup>	1,3949 · 10 <sup>1</sup> ± 1,6098 · 10 <sup>-1</sup>	1,3147 · 10 <sup>1</sup> ± 2,6581 · 10 <sup>-1</sup>
F <sub>15</sub>	3,6822 · 10 <sup>2</sup> ± 9,4598 · 10 <sup>1</sup>	4,9194 · 10 <sup>2</sup> ± 3,1956 · 10 <sup>0</sup>	7,3948 · 10 <sup>2</sup> ± 5,8888 · 10 <sup>1</sup>	<b>7,6400 · 10<sup>-11</sup></b> ± <b>7,9718 · 10<sup>-11</sup></b>
F <sub>16</sub>	7,4683 · 10 <sup>1</sup> ± 3,6143 · 10 <sup>1</sup>	3,3260 · 10 <sup>2</sup> ± 3,3437 · 10 <sup>1</sup>	5,9955 · 10 <sup>2</sup> ± 5,8178 · 10 <sup>1</sup>	1,5524 · 10 <sup>2</sup> ± 1,8724 · 10 <sup>1</sup>
F <sub>17</sub>	<b>8,5361 · 10<sup>1</sup></b> ± <b>3,9099 · 10<sup>1</sup></b>	4,2209 · 10 <sup>2</sup> ± 2,5448 · 10 <sup>1</sup>	8,2121 · 10 <sup>2</sup> ± 1,3321 · 10 <sup>2</sup>	2,3951 · 10 <sup>2</sup> ± 3,3555 · 10 <sup>1</sup>
F <sub>18</sub>	9,0504 · 10 <sup>2</sup> ± 1,4284 · 10 <sup>0</sup>	9,3509 · 10 <sup>2</sup> ± 1,3353 · 10 <sup>1</sup>	1,1209 · 10 <sup>3</sup> ± 4,2776 · 10 <sup>1</sup>	9,2347 · 10 <sup>2</sup> ± 5,1172 · 10 <sup>0</sup>
F <sub>19</sub>	9,0482 · 10 <sup>2</sup> ± 1,0871 · 10 <sup>0</sup>	9,1884 · 10 <sup>2</sup> ± 3,3481 · 10 <sup>1</sup>	1,1134 · 10 <sup>3</sup> ± 3,4552 · 10 <sup>1</sup>	9,0427 · 10 <sup>2</sup> ± 8,3882 · 10 <sup>1</sup>
F <sub>20</sub>	9,0477 · 10 <sup>2</sup> ± 1,0987 · 10 <sup>0</sup>	9,2080 · 10 <sup>2</sup> ± 3,6923 · 10 <sup>1</sup>	1,1066 · 10 <sup>3</sup> ± 4,2209 · 10 <sup>1</sup>	8,9842 · 10 <sup>2</sup> ± 1,0703 · 10 <sup>2</sup>
F <sub>21</sub>	5,0000 · 10 <sup>2</sup> ± 0,0000 · 10 <sup>0</sup>	6,4631 · 10 <sup>2</sup> ± 1,4540 · 10 <sup>1</sup>	1,3125 · 10 <sup>3</sup> ± 2,5259 · 10 <sup>1</sup>	4,9275 · 10 <sup>2</sup> ± 2,5101 · 10 <sup>1</sup>
F <sub>22</sub>	8,8125 · 10 <sup>2</sup> ± 1,6213 · 10 <sup>1</sup>	1,0496 · 10 <sup>3</sup> ± 2,5881 · 10 <sup>1</sup>	1,2285 · 10 <sup>3</sup> ± 4,5987 · 10 <sup>1</sup>	9,8933 · 10 <sup>2</sup> ± 2,6193 · 10 <sup>1</sup>
F <sub>23</sub>	5,3416 · 10 <sup>2</sup> ± 3,4895 · 10 <sup>-4</sup>	7,0590 · 10 <sup>2</sup> ± 3,5040 · 10 <sup>1</sup>	1,3258 · 10 <sup>3</sup> ± 3,5507 · 10 <sup>1</sup>	<b>5,3195 · 10<sup>2</sup></b> ± <b>6,1406 · 10<sup>0</sup></b>
F <sub>24</sub>	<b>2,0000 · 10<sup>2</sup></b> ± <b>0,0000 · 10<sup>0</sup></b>	6,1947 · 10 <sup>2</sup> ± 9,6776 · 10 <sup>1</sup>	1,4389 · 10 <sup>3</sup> ± 2,9140 · 10 <sup>1</sup>	<b>2,0000 · 10<sup>2</sup></b> ± <b>1,1306 · 10<sup>-3</sup></b>
F <sub>25</sub>	2,1321 · 10 <sup>2</sup> ± 6,2752 · 10 <sup>-1</sup>	3,3871 · 10 <sup>2</sup> ± 2,7412 · 10 <sup>2</sup>	1,3731 · 10 <sup>3</sup> ± 2,1013 · 10 <sup>1</sup>	2,6456 · 10 <sup>2</sup> ± 1,4484 · 10 <sup>1</sup>

Tab. 4.14: Rank de cada algoritmo para os problemas de teste da competição CEC'2005 em  $D = 10$ . A antepenúltima linha apresenta a *média dos ranks* de cada algoritmo, a penúltima linha o *desvio padrão dos ranks* e a última linha o *rank final*, obtido a partir do rank médio. Os melhores resultados estão em negrito.

	BLX-GLS0	BLX-MA	CoEVO	DE	DMIS-L-PSO	EDA	G-CMA-ES	K-PCX	L-CMA-ES	L-SaDE	SPC-PNX	opt-aiNet	dept-aiNet	cob-aiNet
F <sub>1</sub>	1	1	1	1	1	1	1	1	1	1	1	13	14	1
F <sub>2</sub>	1	1	1	1	1	1	1	1	1	1	1	13	14	1
F <sub>3</sub>	8	11	1	6	1	7	1	9	1	5	12	13	14	10
F <sub>4</sub>	1	7	1	1	10	1	1	8	14	9	1	12	13	11
F <sub>5</sub>	1	9	11	1	7	1	1	12	1	8	1	13	14	10
F <sub>6</sub>	1	8	9	7	5	6	1	11	1	4	10	13	14	12
F <sub>7</sub>	3	10	5	9	6	12	1	7	1	4	8	13	14	11
F <sub>8</sub>	11	7	8	13	1	10	1	1	1	1	14	12	9	6
F <sub>9</sub>	8	6	11	7	1	10	5	4	13	1	9	14	12	1
F <sub>10</sub>	5	7	11	9	3	6	1	2	12	4	8	13	14	10
F <sub>11</sub>	4	7	13	1	8	6	2	14	5	9	3	12	11	10
F <sub>12</sub>	9	6	11	5	2	10	4	14	7	1	8	12	13	3
F <sub>13</sub>	6	7	11	10	2	12	5	4	3	1	9	14	13	8
F <sub>14</sub>	2	1	11	9	4	5	7	3	13	6	8	12	14	10
F <sub>15</sub>	13	8	9	7	2	11	5	14	4	3	6	12	10	1
F <sub>16</sub>	2	6	12	9	3	10	1	4	7	5	8	13	14	11
F <sub>17</sub>	2	8	11	5	3	9	7	1	14	4	6	12	13	10
F <sub>18</sub>	3	12	13	2	11	7	1	10	8	9	5	6	14	4
F <sub>19</sub>	4	12	13	3	10	8	1	11	7	9	2	6	14	5
F <sub>20</sub>	4	10	13	5	12	8	1	11	3	9	2	7	14	6
F <sub>21</sub>	11	12	9	6	8	5	7	13	3	4	10	2	14	1
F <sub>22</sub>	10	2	13	4	3	12	5	1	8	7	9	11	14	6
F <sub>23</sub>	6	12	11	4	9	7	3	13	10	8	5	1	14	2
F <sub>24</sub>	1	8	11	1	8	1	1	12	13	1	1	10	14	1
F <sub>25</sub>	9	7	1	13	2	4	5	10	12	6	11	3	14	8
Rank Médio	5,04	7,40	8,84	5,56	4,92	6,80	2,76	7,64	6,52	4,80	6,32	10,48	13,28	6,36
Desvio Padrão do Rank	3,80	3,38	4,38	3,73	3,63	3,72	2,28	4,91	4,89	3,06	3,92	3,89	1,37	4,04
Rank Final	4	10	12	5	3	9	1	11	8	2	6	13	14	7

Tab. 4.15: Rank de cada algoritmo os problemas de teste da competição CEC'2005 em  $D = 30$ . A antepenúltima linha apresenta a *média dos ranks* de cada algoritmo, a penúltima linha o *desvio padrão dos ranks* e a última linha o *rank final*, obtido a partir do rank médio. Os melhores resultados estão em negrito.

	BLX-GI.50	BLX-MA	CoEVO	DE	DMS-L-PSO	EDA	G-CMA-ES	K-PCX	L-CMA-ES	L-SaDE	SPE-IPNX	opt-aiNet	dopt-aiNet	cob-aiNet
F <sub>1</sub>	1	1	12	1	1	1	1	1	1	1	1	13	14	1
F <sub>2</sub>	1	9	12	11	7	1	1	10	1	6	8	13	14	1
F <sub>3</sub>	6	10	5	9	4	1	1	8	1	7	12	13	14	11
F <sub>4</sub>	5	6	9	4	8	1	10	7	14	3	2	12	13	11
F <sub>5</sub>	5	8	12	4	9	3	1	7	1	6	10	13	14	11
F <sub>6</sub>	3	9	12	8	4	6	1	11	1	7	5	13	14	10
F <sub>7</sub>	1	9	12	5	6	1	1	11	1	7	10	13	14	8
F <sub>8</sub>	12	7	8	14	1	11	4	1	1	6	10	9	13	5
F <sub>9</sub>	6	4	10	8	7	11	5	3	12	1	9	14	13	1
F <sub>10</sub>	3	7	11	8	5	10	2	1	13	4	6	12	14	9
F <sub>11</sub>	4	9	12	11	6	14	1	8	3	5	2	13	10	7
F <sub>12</sub>	6	4	11	3	1	5	9	14	8	2	7	12	13	10
F <sub>13</sub>	9	8	10	5	3	12	4	11	2	1	7	13	14	6
F <sub>14</sub>	2	4	8	10	1	9	5	12	14	3	6	11	13	7
F <sub>15</sub>	4	7	11	8	6	10	2	14	3	5	9	12	13	1
F <sub>16</sub>	6	10	12	1	7	9	2	4	3	14	5	11	13	8
F <sub>17</sub>	2	8	11	5	3	6	9	4	13	14	1	10	12	7
F <sub>18</sub>	5	3	12	6	9	2	6	1	4	14	8	11	13	10
F <sub>19</sub>	5	3	12	6	10	2	6	1	4	14	9	11	13	8
F <sub>20</sub>	7	3	12	8	6	2	8	1	4	14	10	11	13	5
F <sub>21</sub>	3	3	9	3	14	3	3	11	1	14	3	10	12	2
F <sub>22</sub>	4	7	10	6	14	3	1	12	2	14	5	9	11	8
F <sub>23</sub>	8	7	11	2	14	4	2	10	6	14	4	9	12	1
F <sub>24</sub>	8	1	10	1	14	1	9	6	11	14	1	7	12	1
F <sub>25</sub>	2	4	11	10	14	1	3	5	9	14	6	8	12	7
Rank Médio	4,72	6,04	10,60	6,28	6,96	5,16	3,88	6,96	5,32	8,16	6,24	11,32	12,92	6,24
Desvio Padrão do Rank	2,73	2,79	1,73	3,49	4,40	4,27	3,07	4,50	4,83	5,17	3,28	1,84	1,04	3,64
Rank Final	2	5	12	8	9	3	1	9	4	11	6	13	14	6

para este problema.

Considerando os paradigmas associados a cada algoritmo e tomando a cob-aiNet como base de comparação, é interessante notar que os algoritmos baseados em evolução diferencial (DE e L-SaDE) e otimização por enxame de partículas (DMS-L-PSO) sofreram mais que a cob-aiNet com o aumento de dimensão de  $D = 10$  para  $D = 30$ , já que todos os três algoritmos foram, em média, superiores à cob-aiNet no primeiro caso, e inferiores no segundo. É importante destacar aqui também que tanto o algoritmo DMS-L-PSO quanto o L-SaDE não conseguiram convergir para alguns problemas em  $D = 30$  (Liang & Suganthan, 2005; Qin & Suganthan, 2005), o que levou os respectivos autores a optarem por não reportarem estes resultados na literatura. Nestes casos, ambos os algoritmos foram classificados com *rank* 14, corresponde à última posição possível de ser obtida.

Por outro lado, os algoritmos de estimação de distribuição (EDA, L-CMA-ES e G-CMA-ES) e o algoritmo memético BLX-MA foram os que menos sofreram com o aumento da dimensionalidade, quando comparados à cob-aiNet, já que o algoritmo proposto apresentou desempenho médio superior a três deles (EDA, L-CMA-ES e BLX-MA) em  $D = 10$  e inferior a todos para o caso  $D = 30$ . O G-CMA-ES foi o algoritmo de melhor desempenho médio em ambos os casos, sendo, inclusive, considerado o vencedor da competição CEC'2005.

É importante destacar aqui que os algoritmos de estimação de distribuição (EDAs) são ferramentas que se utilizam de mecanismos mais elaborados que a simples combinação de operadores de reprodução, mutação e seleção, buscando construir, a cada iteração, modelos do espaço de busca do problema que os permitam amostrar soluções cada vez mais promissoras com o passar do tempo. Ao fazerem isso, os EDAs tendem a apresentar um progresso muito maior por iteração quando comparados a outros paradigmas, apesar de, conseqüentemente, também terem custos computacionais maiores por iteração.

Com relação aos algoritmos genéticos incrementais (K-PCX e SPC-PNX), a cob-aiNet obteve desempenho médio superior ao K-PCX nas duas situações estudadas aqui e superior ao SPC-PNX para o caso  $D = 10$ , empatando com este último para  $D = 30$ . Já o algoritmo genético geracional BLX-GL50 apresentou *rank* médio superior ao da cob-aiNet em ambas as situações.

Por fim, um último aspecto que chama a atenção nos resultados é o fato da cob-aiNet apresentar um desempenho ruim para os problemas  $F_3$  a  $F_5$ , tanto com  $D = 10$  quanto com  $D = 30$ , ainda mais considerando-se que estes problemas são monomodais. Ao analisar as características destes problemas, é possível notar que tanto  $F_3$  quanto  $F_5$  possuem regiões em torno do ótimo com pouca inclinação, o que faz com que a cob-aiNet convirja lentamente em direção ao ótimo global e não consiga atingir uma solução de qualidade equivalente à da maioria dos demais algoritmos da competição (vide Figura 4.25). O mau desempenho da cob-aiNet nestes dois problemas está diretamente associado aos seus operadores de hipermutação e busca local. O operador de hipermutação, como

dito anteriormente, altera simultaneamente todas as variáveis de uma dada solução candidata, o que dificulta a localização da direção ideal que leve à geração de soluções melhores que o indivíduo atual, principalmente no caso de problemas de maior dimensão. Já o operador de busca local, principal responsável pela evolução constante observada na Figura 4.25<sup>9</sup>, deixa de reinicializar os intervalos de busca para cada variável a partir do momento em que consegue identificar a direção e o tamanho do passo que devem ser seguidos. Sendo assim, apesar do operador de busca local ser iniciado com intervalos de busca altos, estes valores são constantemente reduzidos até que o algoritmo encontre a região promissora do espaço de busca, o que faz com que a progressão da busca local se torne contínua, impedindo assim a reinicialização dos intervalos de busca e a consequente aceleração da evolução.

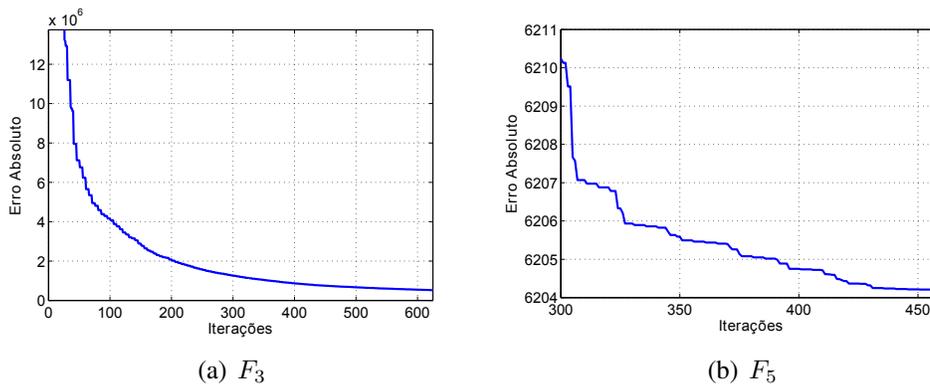


Fig. 4.25: Representação gráfica da progressão do erro absoluto nas iterações finais de uma execução do algoritmo *cob-aiNet* para os problemas  $F_3$  e  $F_5$ .

Já o problema  $F_4$ , por sua vez, também apresenta as características dos problemas  $F_3$  e  $F_5$  descritas acima, mas adicionadas da presença de ruído nos valores da função-objetivo, o que leva o algoritmo a não conseguir evoluir adequadamente em algumas das repetições, causando os altos valores de erro absoluto médio observados nas Tabelas 4.10 e 4.13.

Dito isso, de maneira geral é possível concluir que, apesar de não ter sido o algoritmo de melhor desempenho para o conjunto de problemas da competição CEC'2005, a *cob-aiNet* é uma ferramenta competitiva também para situações em que se deseja apenas a obtenção da melhor solução possível para o problema, já que, se forem considerados os resultados obtidos tanto para  $D = 10$  quanto para  $D = 30$ , apenas dois algoritmos dentre os treze competidores apresentaram um *rank* final superior ao da *cob-aiNet* em ambos os casos: o G-CMA-ES, vencedor da competição, e o BLX-GL50.

<sup>9</sup>Foram feitos experimentos adicionais com os problemas  $F_3$  a  $F_5$  sem a utilização do operador de busca local, que permitiram identificar a importância deste operador nestes três problemas. No entanto, para que esta seção não ficasse excessivamente sobrecarregada, tais resultados foram omitidos aqui.

## 4.4 Considerações Finais

Neste capítulo, foi feita uma série de experimentos com o intuito de avaliar o desempenho geral do algoritmo cob-aiNet, tanto em relação à sua capacidade de localização do ótimo global de cada problema quanto à sua capacidade de manter a diversidade das soluções na população. Além disso, buscou-se aqui estudar também o impacto que cada um de seus parâmetros tem no desempenho da ferramenta, além de comparar esta nova proposta a diversas outras abordagens da literatura. Para isso, foi adotado o conjunto de funções de teste proposto para a Competição de Otimização de Parâmetros Reais da 2005 CEC *Special Session on Real-Parameter Optimization*, e as comparações foram feitas tanto com os onze algoritmos que participaram desta competição, que são baseados em diferentes paradigmas de meta-heurísticas populacionais, quanto com duas outras ferramentas que adotam o mesmo paradigma da cob-aiNet, ou seja, dois sistemas imunológicos artificiais (opt-aiNet e dopt-aiNet).

Com relação à capacidade da cob-aiNet de localizar e manter diversos ótimos locais ao longo da busca, em problemas multimodais, as comparações feitas com os dois SIAs (Seção 4.1) mostraram que a ferramenta proposta nesta tese de fato possui mecanismos superiores de manutenção de diversidade, uma vez que em todos os problemas a cob-aiNet foi capaz de localizar uma maior quantidade de ótimos locais, além de também ter sido capaz de se adaptar mais adequadamente às características de cada problema estudado. Nos problemas monomodais, a cob-aiNet sempre convergiu para uma única solução, enquanto que nos multimodais o número de indivíduos em sua população aumentou ou diminuiu conforme a quantidade de ótimos locais existentes. Além disso, um outro aspecto em que a cob-aiNet se mostrou superior foi quanto ao erro absoluto médio obtido entre o melhor indivíduo encontrado durante a busca e o ótimo global do problema, já que em todos os casos a cob-aiNet foi capaz de atingir a precisão de  $1 \cdot 10^{-9}$  definida como desejável.

Na análise de sensibilidade do desempenho do algoritmo a diferentes ajustes de seus parâmetros (Seção 4.2), foi possível identificar como cada um destes parâmetros influencia o comportamento da cob-aiNet, e também propor sugestões sobre como devem ser ajustados. Dentre os parâmetros estudados, ficou evidente que o limiar de supressão  $\sigma_s$  é o que tem maior impacto no desempenho da ferramenta, além de ser o parâmetro de mais difícil ajuste.

Quando comparada aos onze algoritmos participantes da competição CEC'2005 e aos dois SIAs em todos os problemas de teste, tanto em dimensão 10 quanto dimensão 30 (Seção 4.3), foi observado que a cob-aiNet se mostrou competitiva frente aos demais algoritmos, mesmo não apresentando o melhor *rank* final em nenhuma das duas situações. No entanto, é importante ressaltar que a cob-aiNet foi estudada com apenas dois conjuntos de parâmetros (um para  $D = 10$  e outro para  $D = 30$ ), mantidos fixos para todos os problemas estudados. Caso fossem utilizados conjuntos de parâmetros ajustados especificamente para cada problema, os resultados tenderiam a ser melhores. Os resultados obtidos

nestes experimentos serviram também para confirmar a superioridade da *cob-aiNet* frente à *opt-aiNet* e *dopt-aiNet*, já que o algoritmo proposto nesta tese apresentou um desempenho consistentemente superior ao dos outros dois SIAs em praticamente todos os problemas tratados.

O fato da *cob-aiNet* ter se mostrado competitiva frente a algoritmos diversos, quando se busca apenas a localização da melhor solução global para um problema de otimização, e também de trazer consigo a vantagem de poder ser ajustada para localizar múltiplos ótimos locais quando tal característica se faz necessária, acabou motivando a extensão deste algoritmo para tratar problemas de otimização *multiobjetivo* em espaços contínuos, que será introduzida e discutida no capítulo a seguir.

## Capítulo 5

### O Algoritmo *cob-aiNet[MO]*

No contexto de otimização, uma classe de problemas que também requer a utilização de ferramentas que possuam boa capacidade de manutenção de diversidade é a dos chamados *problemas de otimização multiobjetivo* (OMO – Coello Coello et al., 2007; Deb, 2001), que têm como principal característica a exigência de otimização simultânea de *múltiplos critérios*, geralmente conflitantes entre si. Como visto no Capítulo 4, o algoritmo *cob-aiNet* demonstrou possuir mecanismos eficientes de manutenção de diversidade de soluções em sua população, que permitem até mesmo a localização de múltiplos ótimos locais em uma única execução, além de também ter se mostrado competitivo frente a outras ferramentas da literatura quando se busca a localização da melhor solução global para um problema de otimização. Sendo assim, neste capítulo será proposta e discutida uma extensão do algoritmo *cob-aiNet* voltada para o tratamento de problemas de otimização multiobjetivo em espaços contínuos, denominada *cob-aiNet[MO]* (*Concentration-based Artificial Immune Network for Multiobjective Optimization* – Coelho & Von Zuben, 2011).

Apesar do algoritmo *cob-aiNet[MO]* ter sido desenvolvido para o tratamento de problemas de otimização multiobjetivo em espaços contínuos, novamente é importante ressaltar que sua extensão para outros tipos de problemas de otimização multiobjetivo (como, por exemplo, otimização multiobjetivo combinatória) é viável e pode adotar os mesmos passos concebidos para a família de algoritmos *aiNet* (de França et al., 2010a).

Este capítulo está dividido em quatro seções principais. Na Seção 5.1, serão discutidos os principais conceitos associados à otimização multiobjetivo e ressaltadas as características mais relevantes que devem estar presentes em um algoritmo proposto para tratar esta classe de problemas. Na Seção 5.2, será feito o posicionamento do algoritmo *cob-aiNet[MO]* frente a outras propostas da literatura voltadas para otimização multiobjetivo. O algoritmo *cob-aiNet[MO]* propriamente dito será apresentado na Seção 5.3. Como muitos dos mecanismos desta nova ferramenta são idênticos aos da *cob-aiNet* original, previamente detalhados no Capítulo 3, na Seção 5.3 serão discutidos apenas aque-

les aspectos que diferenciam uma ferramenta da outra. Por fim, os comentários finais deste capítulo serão apresentados na Seção 5.4.

## 5.1 Conceitos Gerais de Otimização Multiobjetivo

A primeira grande diferença entre problemas de otimização mono-objetivo e problemas de otimização multiobjetivo está no conceito de *solução* propriamente dito. No caso de problemas de otimização mono-objetivo, a solução corresponde a um ou mais pontos factíveis (ou seja, que atendam a todas as restrições) que levem a valores extremos da função-objetivo. Em outras palavras, caso se tenha um problema de *minimização*, a solução corresponderá ao conjunto de pontos factíveis que possuem o *menor* valor da função-objetivo, enquanto que em problemas de *maximização* a solução será o conjunto de pontos factíveis com o *maior* valor possível da função-objetivo. Já para o caso de problemas de otimização multiobjetivo, este conceito de soluções que correspondem a valores extremos da função-objetivo não pode ser aplicado diretamente, uma vez que os *objetivos* (ou *critérios*) a serem otimizados podem ser (e geralmente são) conflitantes.

Para ilustrar esta situação em que os múltiplos objetivos são conflitantes entre si, tomemos um exemplo didático: suponha que um determinado veículo será construído e o fabricante deve escolher tanto o material que será utilizado na fabricação de suas chapas quanto a espessura que tais chapas devem ter, de forma que o veículo tenha simultaneamente uma máxima resistência a colisões, o menor custo de fabricação e o mais baixo consumo de combustível possível. Se o fabricante optar pela utilização de aço, deverão ser empregadas chapas de maior espessura para deixar o veículo mais resistente a colisões, o que acarreta um maior peso e, conseqüentemente, um maior consumo de combustível. Para reduzir o peso e diminuir o consumo de combustível, o fabricante poderia utilizar outro tipo de material, como fibra de carbono, que é altamente resistente e leve, mas isto aumentaria consideravelmente o custo de produção do veículo.

Neste exemplo, os três objetivos a serem otimizados (custo, resistência e consumo de combustível) são claramente conflitantes dadas as variáveis passíveis de serem modificadas, uma vez que não é possível escolher a solução extrema para um deles (resistência, por exemplo, com o uso de fibra de carbono) e também obter resultados ótimos para os demais (com o uso de fibra de carbono, o custo de produção do veículo é alto).

Geralmente, em problemas de otimização multiobjetivo com objetivos conflitantes, existe um *conjunto* de soluções onde cada solução corresponde a um possível *compromisso* entre os objetivos. Cada uma das soluções neste *conjunto ótimo* indica, para um dado valor de um dos objetivos, quais são os melhores valores que os demais critérios podem ter. Sendo assim, é possível afirmar que podem existir conjuntos com infinitas soluções para um dado problema de otimização multiobjetivo,

que correspondem aos infinitos compromissos possíveis entre os objetivos.

Sem perda de generalidade, os conceitos e definições que serão discutidos nesta seção considerarão a definição de problema de otimização multiobjetivo dada na Formulação 5.1, em que todos os objetivos devem ser minimizados. Além disso, supõe-se também que o problema de otimização em questão envolva apenas variáveis em espaços contínuos.

$$\begin{aligned}
 &\text{Minimizar} && (f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})), && (5.1) \\
 &\text{Sujeito a} && g_j(\vec{x}) \geq 0, && j = 1, 2, \dots, J, \\
 &&& h_k(\vec{x}) = 0, && k = 1, 2, \dots, K, \\
 &&& x_i^L \leq x_i \leq x_i^U, && i = 1, 2, \dots, n, \\
 &&& x_i \in \mathfrak{R} && i = 1, 2, \dots, n.
 \end{aligned}$$

onde  $n$  é o número de variáveis (dimensão do problema),  $M$  é o número de objetivos,  $J$  é o número de restrições de desigualdade,  $K$  é o número de restrições de igualdade,  $x_i^L$  é o limite inferior do domínio da variável  $i$  e  $x_i^U$  é o limite superior do domínio da variável  $i$ .

O conjunto de soluções ótimas para um problema de otimização multiobjetivo como o ilustrado na Formulação 5.1 é chamado de *conjunto ótimo de Pareto*, e contém apenas soluções ditas *não dominadas* (Coello Coello et al., 2007; Deb, 2001). Estes conceitos de *dominância* e *conjunto ótimo de Pareto* serão discutidos nas próximas sub-seções.

### 5.1.1 Dominância e Dominância Restrita

Diz-se que um dado vetor de solução  $\vec{x} = (x_1, \dots, x_n)$ , no espaço  $\mathfrak{R}^n$ , *domina* um vetor de solução  $\vec{y} = (y_1, \dots, y_n)$  (denotado por  $\vec{x} \preceq \vec{y}$ ) se e somente se  $\vec{x}$  é *melhor ou igual* a  $\vec{y}$  para *todas* as  $M$  funções-objetivo do problema, e *estritamente melhor* que  $\vec{y}$  em *pelo menos uma* delas. Em outras palavras, dadas duas soluções factíveis  $\vec{x}$  e  $\vec{y}$  para um problema equivalente ao da Formulação 5.1, diz-se que  $\vec{x} \preceq \vec{y}$  se e somente se  $\forall m \in \{1, \dots, M\}, f_m(\vec{x}) \leq f_m(\vec{y})$  e  $\exists i \in \{1, \dots, M\} : f_i(\vec{x}) < f_i(\vec{y})$  (Deb, 2001).

Este conceito de dominância entre soluções permite a comparação direta entre duas soluções para um problema de otimização multiobjetivo, através da avaliação de cada uma delas por todas as funções-objetivo simultaneamente. Sendo assim, caso uma dada solução  $\vec{x}$  domine uma solução  $\vec{y}$ , é possível dizer que  $\vec{x}$  corresponde a um melhor *compromisso* entre os objetivos do problema que aquele representado por  $\vec{y}$ .

Uma extensão deste conceito de dominância, que leva em consideração não apenas os objetivos

do problema mas também as restrições associadas a ele, é a chamada *dominância restrita* (Deb, 2001; Deb et al., 2002a). Diz-se que uma dada solução  $\vec{x}$  *domina com restrições* uma solução  $\vec{y}$  se qualquer uma das seguintes condições for verdadeira:

1. A solução  $\vec{x}$  é factível e a solução  $\vec{y}$  é infactível;
2. Ambas as soluções  $\vec{x}$  e  $\vec{y}$  são infactíveis mas a solução  $\vec{x}$  apresenta uma menor *violação de restrições* quando comparada à solução  $\vec{y}$ ;
3. Ambas as soluções  $\vec{x}$  e  $\vec{y}$  são factíveis e  $\vec{x} \preceq \vec{y}$ .

O grau de *violação de restrições* para uma solução  $\vec{x}$  é dado por uma métrica que indica o quão severamente  $\vec{x}$  viola as restrições do problema. Várias métricas podem ser utilizadas para fazer tal avaliação, sendo que uma delas é dada na Equação 5.2:

$$VR(\vec{x}) = \sum_{j=1}^J \sigma(g_j(\vec{x})) + \sum_{k=1}^K |h_k(\vec{x})| \quad (5.2)$$

onde  $h_k(\vec{x})$  é o valor da  $k$ -ésima restrição de igualdade para a solução  $\vec{x}$ ,  $g_j(\vec{x})$  é o valor da  $j$ -ésima restrição de desigualdade para a solução  $\vec{x}$ ,  $\sigma(g_j(\vec{x})) = 0$  se  $g_j(\vec{x}) > 0$  e  $\sigma(g_j(\vec{x})) = |g_j(\vec{x})|$  se  $g_j(\vec{x}) \leq 0$ .

Por permitirem uma comparação direta entre duas soluções, os conceitos de dominância e dominância restrita são empregados na grande maioria das meta-heurísticas populacionais voltadas para a resolução de problemas de otimização multiobjetivo.

### 5.1.2 Conjunto Ótimo e Fronteira de Pareto

Em problemas de otimização multiobjetivo, o conceito de “soluções ótimas” está associado aos conceitos de *conjunto ótimo de Pareto* e *fronteira de Pareto*. Tais definições de otimalidade, introduzidas por Francis Ysidro Edgeworth em 1881 (Edgeworth, 1881) e posteriormente generalizadas por Vilfredo Pareto em 1896 (Pareto, 1896), colocam que uma dada solução  $\vec{x}^*$  pertence ao *conjunto ótimo de Pareto* caso não exista nenhuma outra solução  $\vec{x}$  factível capaz de reduzir o valor de um dos objetivos do problema (em relação a  $\vec{x}^*$ ) sem que simultaneamente haja um aumento em pelo menos um dos demais critérios. Sendo assim, o conjunto ótimo de Pareto corresponde ao conjunto de soluções que *não são dominadas* por nenhuma outra solução factível no domínio do problema.

Um outro aspecto que diferencia os problemas de otimização multiobjetivo dos problemas mono-objetivo é a necessidade de se trabalhar com *dois* espaços distintos: (i) o espaço de busca propriamente dito, também chamado de *espaço de variáveis*; e (ii) o *espaço de objetivos* do problema, formado por suas funções-objetivo. A aplicação das funções-objetivo de um problema a uma dada solução pode

ser vista como um processo de *mapeamento* desta solução, originalmente posicionada no espaço de variáveis, no espaço de objetivos, como ilustrado na Figura 5.1. Diante disso, o mapeamento do conjunto ótimo de Pareto no espaço de objetivos do problema recebe o nome de *fronteira de Pareto*.

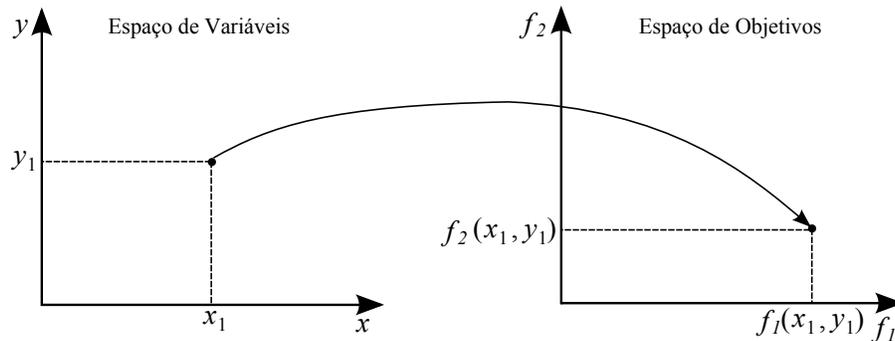


Fig. 5.1: Exemplo de mapeamento de uma solução  $(x_1, y_1)$  do espaço de variáveis bidimensional para o espaço de objetivos, também bidimensional. Este mapeamento é obtido a partir da aplicação das funções-objetivo do problema a qualquer ponto no espaço de variáveis.

Na Figura 5.1, para propósitos de visualização, ambos os espaços têm dimensão 2. No entanto, qualquer um dos dois espaços pode ter dimensão menor que a do outro, dependendo da natureza do problema de otimização.

Na Figura 5.2 estão representados exemplos de fronteiras de Pareto de um problema hipotético com dois objetivos ( $f_1$  e  $f_2$ ), sendo que as áreas hachuradas correspondem ao mapeamento da região de factibilidade do problema no espaço de objetivos. Na Figura 5.2(a) tem-se a representação gráfica da fronteira de Pareto (traço escuro na borda inferior esquerda da região de factibilidade) para o caso em que ambos os critérios devem ser minimizados, enquanto que nas Figuras 5.2(b), 5.2(c) e 5.2(d) observa-se, respectivamente, as fronteiras de Pareto para os casos em que (i)  $f_1$  deve ser minimizado e  $f_2$  maximizado, (ii)  $f_1$  maximizado e  $f_2$  minimizado, e (iii) tanto  $f_1$  quanto  $f_2$  maximizados. É importante ressaltar que, nas Figuras 5.2(b) e 5.2(d), as linhas tracejadas não fazem parte da fronteira de Pareto, e estão representadas nas figuras apenas para destacar que as regiões não convexas não possuem indivíduos não dominados, e portanto não fazem parte da fronteira de Pareto do problema.

Dados estes dois conceitos, é importante notar que, diferentemente de problemas mono-objetivo em que o objetivo principal do processo de otimização é encontrar o(s) ótimo(s) global(is), a resolução de um problema multiobjetivo tradicionalmente requer: (i) a localização um conjunto amplo e finito de soluções na fronteira de Pareto; e (ii) a garantia de que estas soluções sejam diversas *no espaço de objetivos*, para que se obtenha uma cobertura adequada da fronteira de Pareto do problema. Tal “cobertura apropriada” da fronteira de Pareto geralmente está associada a uma amostragem uniforme desta fronteira, que permite que o usuário obtenha o mais completo conjunto de *compromissos ótimos* entre os critérios possível, permitindo assim a escolha *a posteriori* daquela solução que melhor se

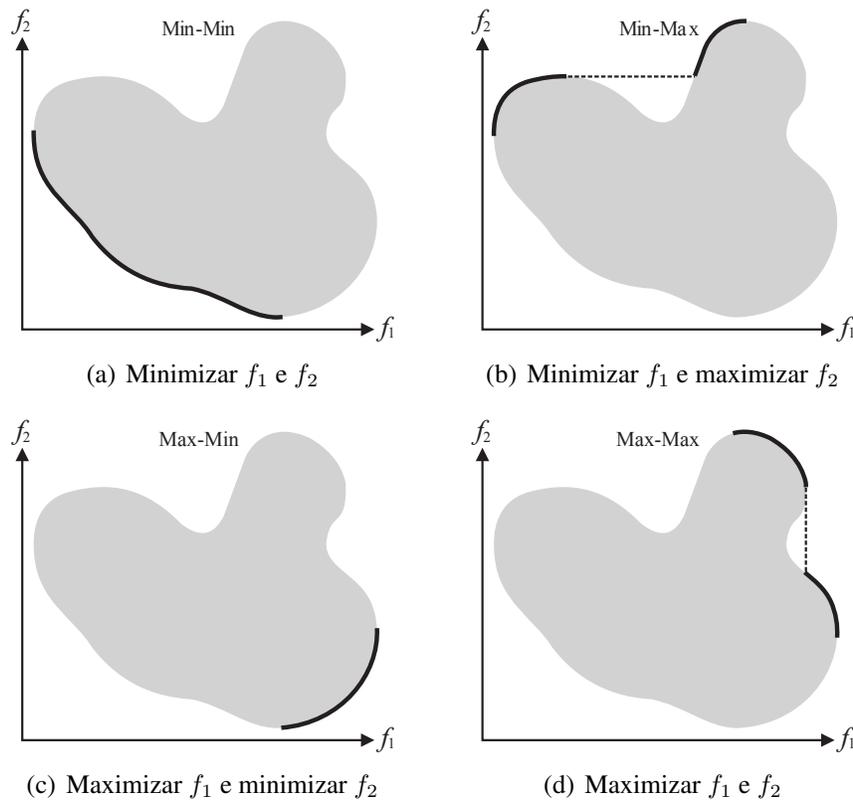


Fig. 5.2: Exemplos de fronteiras de Pareto para um problema biobjetivo hipotético. As fronteiras correspondem aos traços escuros nas bordas das áreas hachuradas, que, por sua vez, correspondem ao mapeamento da região de factibilidade do problema no espaço de objetivos. Em (a) tem-se um exemplo de fronteira de Pareto para o caso em que ambos os objetivos devem ser minimizados, em (b) para o caso em que  $f_1$  deve ser minimizado e  $f_2$  maximizado, em (c)  $f_1$  deve ser maximizado e  $f_2$  minimizado e, em (d), ambos os critérios devem ser maximizados.

adeque aos propósitos do problema.

É possível dizer que estas duas exigências para a solução de um problema multiobjetivo são, de certa forma, ortogonais, uma vez que a satisfação de uma delas não necessariamente leva à satisfação da outra. Diante disso, os algoritmos de otimização multiobjetivo precisam incorporar mecanismos capazes de enfatizar tanto a convergência do conjunto de soluções não dominadas para a fronteira de Pareto do problema quanto a sua diversidade no espaço de objetivos (Deb, 2001).

Além disso, como será visto na próxima seção, uma outra exigência, que também é importante principalmente em problemas multiobjetivo multimodais, mas que raramente é tratada pelas ferramentas propostas na literatura, é a manutenção de diversidade também no *espaço de variáveis* do problema, que corresponde a um dos objetivos almejados com a proposição do algoritmo *cob-aiNet*[MO] neste capítulo.

## 5.2 Posicionamento da Proposta

Desde os anos 1950, uma grande variedade de algoritmos vem sendo desenvolvida dentro da comunidade de Pesquisa Operacional para tratar problemas de otimização multiobjetivo (Ehrgott, 2005; Miettinen, 1999). No entanto, a grande maioria destes algoritmos tradicionais se baseia em alguma forma de combinação dos múltiplos objetivos em uma única função-objetivo, ou até mesmo na otimização de um único critério escolhido *a priori*, enquanto que os demais passam a ser considerados restrições. Estas abordagens de combinação de objetivos ou conversão de objetivos em restrições acabam tendo uma série de limitações como, por exemplo, a exigência em alguns casos de um conhecimento prévio de características do problema por parte do usuário e até mesmo a susceptibilidade de algumas técnicas à forma da fronteira de Pareto (Deb, 2001).

Nas últimas duas décadas, as meta-heurísticas populacionais passaram a ser muito utilizadas no contexto de otimização multiobjetivo, uma vez que elas são menos susceptíveis à forma da fronteira de Pareto, praticamente não demandam nenhum conhecimento prévio sobre características específicas do problema e, por trabalharem com uma população de soluções candidatas, podem ser capazes de retornar, em uma única execução, um conjunto de soluções não dominadas (Coello Coello, 1999, 2006; Deb, 2001). No entanto, apesar da vantagem de serem ferramentas de propósito geral, as meta-heurísticas populacionais para otimização multiobjetivo não garantem a obtenção do ótimo global do problema (no caso, a fronteira de Pareto), não garantem convergência e, quando convergem, não garantem custo até a convergência, como discutido no Capítulo 3 para as meta-heurísticas voltadas para otimização mono-objetivo.

Mesmo assim, dada a impossibilidade de contar com algoritmos exatos para tratar esses problemas, um grande número de meta-heurísticas populacionais, baseadas principalmente no paradigma de algoritmos evolutivos, foi proposto na literatura para tratar problemas de otimização multiobjetivo (Coello Coello & Toscano Pulido, 2001; Corne et al., 2000, 2001; Deb et al., 2002a; Knowles & Corne, 2000; Zitzler & Thiele, 1999; Zitzler et al., 2002; Zhang & Li, 2007), sendo que alguns destes algoritmos, tais como NSGA-II (Deb et al., 2002a), SPEA2 (Zitzler et al., 2002) e MOEA/D (Zhang & Li, 2007), são considerados estado-da-arte dentro desta área de pesquisa.

Especificamente no contexto de sistemas imunológicos artificiais (SIAs), o número de propostas de algoritmos imunoinspirados para otimização multiobjetivo (Campelo et al., 2007; Freschi et al., 2009) também é grande (Castro & Von Zuben, 2008, 2010b; Coello Coello & Cruz Cortés, 2005; Coelho & Von Zuben, 2006b; Freschi & Repetto, 2006; Gao & Wang, 2010; Jiao et al., 2005; Lanaridis & Stafylopatis, 2010; Lu et al., 2005; Shang & Ma, 2006; Tan et al., 2008), sendo que dentre estes pode-se destacar o algoritmo VIS (Freschi & Repetto, 2006) e a omni-aiNet (Coelho & Von Zuben, 2006b) que, como a cob-aiNet[MO] que será apresentada neste capítulo, são baseados nos princípios de seleção clonal (Burnet, 1978) e rede imunológica (Jerne, 1974).

Apesar da grande quantidade de meta-heurísticas propostas para otimização multiobjetivo e da diversidade de paradigmas adotados em tais algoritmos, o principal objetivo almejado por tais técnicas quase sempre é a obtenção de um conjunto de soluções que esteja o mais próximo possível da fronteira de Pareto do problema e que, ao mesmo tempo, tenha uma boa diversidade no *espaço de objetivos*. Estas duas exigências básicas acabaram levando os pesquisadores da área de OMO a focarem seu trabalho em processos e mecanismos que atuam principalmente no *espaço de objetivos*, deixando a questão de diversidade no *espaço de variáveis* em segundo plano. No entanto, também é importante considerar o comportamento das soluções não dominadas no espaço de variáveis, sendo que a proposta de mecanismos que estimulam a diversidade tanto no espaço de objetivos quanto no espaço de variáveis pode até mesmo ser mandatória para resolução de vários tipos de problemas, principalmente aqueles que apresentam características de multimodalidade. Além disso, a correta obtenção tanto do *conjunto ótimo de Pareto* quanto de uma boa *fronteira de Pareto* são muito importantes em uma série de problemas reais, já que facilitariam o processo de escolha *a posteriori* da melhor solução a ser implantada na prática, particularmente nas situações em que preferências e restrições específicas do usuário não estão bem definidas *a priori* (Zhou et al., 2009).

Este foco na manutenção de diversidade apenas no espaço de objetivos se fez presente até mesmo nas ferramentas baseadas no paradigma de SIAs, que notoriamente podem possuir mecanismos eficazes para manutenção de diversidade no espaço de variáveis. Na grande maioria dos SIAs propostos para otimização multiobjetivo (Campelo et al., 2007; Freschi et al., 2009), os mecanismos intrínsecos de manutenção de diversidade acabaram se voltando para o espaço de objetivos, sendo o espaço de variáveis novamente deixado em segundo plano.

Um das ideias pioneiras de exploração de diversidade no espaço de variáveis de problemas de otimização multiobjetivo foi introduzida no início da década de 90, com a primeira proposta do algoritmo NSGA (Srinivas & Deb, 1994), que incluiu uma abordagem de *fitness sharing* (Sareni & Krähenbühl, 1998) entre os seus mecanismos. No entanto, nos anos subsequentes o foco da pesquisa em OMO se voltou, por quase uma década, para a manutenção de diversidade no espaço de objetivos do problema, até que em 2003 o algoritmo GDEA (Toffolo & Benini, 2003) incorporou a diversidade no espaço de variáveis como um objetivo extra a ser maximizado durante a busca. Deste então, apesar de alguns estudos sobre a questão da diversidade propriamente dita terem sido propostos na literatura (Preuss et al., 2006; Rudolph et al., 2007; Unrich et al., 2010), apenas algumas novas ferramentas foram introduzidas, sendo duas em 2005 e duas em 2009. A primeira delas foi o algoritmo *omni-optimizer* (Deb & Tiwari, 2005), que amplia a ideia do NSGA-II (Deb et al., 2002a) de forma a considerar também a diversidade no espaço de variáveis, e é capaz de resolver tanto problemas mono-objetivo quanto multiobjetivo. Já a segunda proposta foi o algoritmo de Chan & Ray (2005), que é um algoritmo evolutivo que avalia a qualidade e a diversidade de cada solução na população,

tanto no espaço de objetivos quanto no de variáveis, através de duas métricas: a *Contribuição de Lebesgue* e a *Contagem de Vizinhança*. No ano de 2009, o primeiro algoritmo foi introduzido por Shir et al. (2009), que incluíram a diversidade no espaço de variáveis na ferramenta CMA-ES (Hansen & Ostermeier, 1996) com *niching*. Por fim, Zhou et al. (2009) propuseram o algoritmo MMEA (*Probabilistic Model-based Multiobjective Evolutionary Algorithm*), que corresponde a um EDA (Larrañaga & Lozano, 2002) baseado em agrupamento, que estimula a diversidade no espaço de variáveis através de seu operador de reprodução.

Diante desta escassez de ferramentas que buscam manter a diversidade de soluções no espaço de variáveis em problemas de otimização multiobjetivo, e dados os bons resultados obtidos pelo algoritmo cob-aiNet em problemas mono-objetivo, na próxima seção será apresentado e discutido o algoritmo cob-aiNet[MO], que é uma extensão da cob-aiNet voltada para tratar problemas de otimização multiobjetivo.

### 5.3 O Algoritmo cob-aiNet[MO]

Como mencionado anteriormente, o algoritmo cob-aiNet[MO] é uma extensão da cob-aiNet, discutida no Capítulo 3, voltada para a resolução de problemas de otimização multiobjetivo. No entanto, apesar de ambos os algoritmos possuírem operadores e mecanismos muito similares, o propósito de cada um deles é um pouco diferente. Enquanto a cob-aiNet busca encontrar, além do(s) ótimo(s) global(is) do problema, a maior quantidade possível de ótimos locais, a cob-aiNet[MO] explora seus mecanismos de manutenção de diversidade não para localizar ótimos locais do problema de otimização multiobjetivo, mas para encontrar o melhor *conjunto ótimo de Pareto* possível. Sendo assim, a cob-aiNet[MO] sempre buscará um conjunto de soluções não dominadas que, além de aproximarem adequadamente a fronteira de Pareto, também cubram ao máximo o conjunto ótimo de Pareto do problema.

A estrutura geral do algoritmo cob-aiNet[MO] está representada no Pseudo-código 5.1. Como é possível notar, o algoritmo cob-aiNet[MO] possui uma estrutura bem semelhante à da cob-aiNet (vide Pseudo-código 3.2), exceto por algumas inversões da posição das avaliações de *fitness* e pela inclusão de uma etapa final de eliminação dos indivíduos dominados ainda presentes na população, de forma que sejam retornadas ao usuário apenas soluções não dominadas. No entanto, alguns mecanismos internos do algoritmo também sofreram modificações para se adaptarem a este novo contexto, como será discutido nas próximas sub-seções. A não discussão neste capítulo de algum operador do algoritmo cob-aiNet[MO] implica que tal operador é idêntico ao originalmente proposto para o algoritmo cob-aiNet, cuja descrição detalhada pode ser encontrada no Capítulo 3.

---

**Pseudo-código 5.1** Passos principais do algoritmo *cob-aiNet*[MO].
 

---

**Parâmetros:**

- $nAB$ : número inicial de células na população;
- $maxAB$ : tamanho máximo da população;
- $nC^{max}$ : número máximo de clones por célula;
- $nC^{min}$ : número mínimo de clones por célula;
- $C_0$ : concentração inicial de cada célula;
- $\sigma_s$ : limiar de supressão;
- $\beta^i$ : parâmetro inicial do operador de hipermutação;
- $\beta^f$ : parâmetro final do operador de hipermutação;
- $LS_{it}$ : número de iterações de busca local;
- $LS_{freq}$ : frequência da busca local;
- $SR$ : intervalo de busca inicial, para cada variável, do operador de busca local;

- 1- Gerar aleatoriamente a população inicial de tamanho  $nAB$ ;
  - 2- Avaliar o *fitness* das células na população inicial;
  - 3- Avaliar a afinidade entre as células da população inicial;
- enquanto** (critério de parada não estiver satisfeito) **faça**
- 4- Definir o número de clones  $nC_i$  que deve ser gerado para cada célula  $i$ ;
  - 5- Gerar  $nC_i$  clones para cada célula  $i$  da população;
  - 6- Aplicar o operador de hipermutação a cada um dos clones gerados;
  - 7- Avaliar o *fitness* das novas células geradas;
  - 8- Selecionar as células que deverão ser mantidas na próxima iteração (com inserção);
- se** (iteração atual) **mod**  $LS_{freq} == 0$  **então**
- 9- Aplicar o operador de *Busca Local* a todos os indivíduos;
- fim se**
- 10- Avaliar o *fitness* das células da população;
  - 11- Avaliar a afinidade entre as células da população;
  - 12- Atualizar a concentração de todas as células;
  - 13- Eliminar da população as células com concentração nula;
- fim enquanto**
- 14- Aplicar o operador de *Busca Local* a todos os indivíduos;
  - 15- Avaliar o *fitness* das células da população;
  - 16- Avaliar a afinidade entre as células da população;
  - 17- Atualizar a concentração de todas as células;
  - 18- Eliminar da população as células com concentração nula;
  - 19- Eliminar da população as células dominadas;
- 

### 5.3.1 Métricas de Afinidade e Atualização de Concentrações

Como visto anteriormente, dado que não existe uma representação explícita de antígenos em SIAs voltados para otimização de funções, foi adotada no algoritmo *cob-aiNet* uma métrica do tipo *fitness* para avaliar a afinidade entre cada célula da população e os antígenos. Neste algoritmo, o *fitness* de cada célula foi definido como sendo basicamente o valor da função-objetivo do problema,

normalizado para todas as soluções presentes na população a cada iteração e ajustado para que sempre seja maximizado.

Já em problemas de otimização multiobjetivo, esta mesma abordagem não pode ser adotada, uma vez que não existe uma única função-objetivo a ser otimizada, salvo nas situações descritas na Seção 5.2 em que os critérios a serem otimizados são combinados em uma única função de custo. Sendo assim, para que as soluções presentes na população de uma meta-heurística populacional possam ser comparadas entre si, geralmente são adotadas duas abordagens diferentes ao se tratar problemas de otimização multiobjetivo: (i) as soluções podem ser *ordenadas em classes* de acordo com o critério de não dominância, de maneira semelhante à proposta no algoritmo NSGA-II (Deb et al., 2002a), sendo que as soluções dentro de uma mesma classe são diferenciadas conforme alguma métrica de diversidade (favorecendo aqueles indivíduos que ocupam regiões pouco povoadas da fronteira de Pareto); ou (ii) pode ser utilizada alguma técnica de *atribuição de fitness*, como proposto no algoritmo SPEA2 (Zitzler et al., 2002), sendo que a métrica de *fitness* deve levar em consideração tanto o critério de não dominância quanto a diversidade das soluções no espaço de objetivos.

Como todos os mecanismos propostos para o algoritmo cob-aiNet se baseiam na existência de um valor explícito de *fitness* para cada célula na população, a estratégia de *atribuição de fitness* foi escolhida para ser empregada na cob-aiNet[MO], o que permitiu que a maioria dos operadores originais da cob-aiNet se mantivesse inalterada. Além disso, outra vantagem da utilização da estratégia de *atribuição de fitness* está no fato da possibilidade de se manter todos os mecanismos de manutenção de diversidade da cob-aiNet[MO] voltados para o tratamento de diversidade no *espaço de variáveis* do problema, uma vez que o processo de otimização da própria métrica de *fitness* se encarregará da manutenção de diversidade no espaço de objetivos, além de buscar aproximar ao máximo a população de soluções da fronteira de Pareto do problema.

Diante disso, foi adotada no algoritmo cob-aiNet[MO] a estratégia de atribuição de *fitness* originalmente proposta no algoritmo SPEA2 (Zitzler et al., 2002), dada na Equação 5.3.

$$f_i^{SPEA2}(t) = R_i(t) + D_i(t), \quad (5.3)$$

onde  $f_i^{SPEA2}(t)$  é o *fitness* do algoritmo SPEA2 para o indivíduo  $i$  na iteração  $t$ ,  $R_i(t)$  é o chamado *fitness bruto* do indivíduo  $i$  na iteração  $t$ , dado pela Equação 5.4 e responsável por avaliar a qualidade da solução de acordo com o critério de não dominância, e  $D_i(t)$  é a *densidade* do indivíduo  $i$  na iteração  $t$ , dado pela Equação 5.6 e responsável por avaliar a diversidade de cada solução no espaço de objetivos. De maneira geral, o *fitness* bruto  $R_i(t)$  para o indivíduo  $i$  na iteração  $t$  é dado pela soma das forças  $S_j(t)$  de todos os indivíduos  $j$  que dominam o indivíduo  $i$ :

$$R_i(t) = \sum_{j \in \mathbf{P}(t) \text{ e } j \preceq i} S_j(t), \quad (5.4)$$

onde  $\mathbf{P}(t)$  é a população de células do algoritmo na iteração  $t$  e  $S_j(t)$  é a chamada *força* do indivíduo  $j$ , dada pela Equação 5.5.

$$S_j(t) = |\{k | k \in \mathbf{P}(t) \text{ e } j \preceq k\}|, \quad (5.5)$$

onde  $\mathbf{P}(t)$  é a população de células do algoritmo na iteração  $t$  e  $|\cdot|$  é a cardinalidade de um conjunto. Em outras palavras, a força do indivíduo  $j$  é dada pelo número de indivíduos da população  $\mathbf{P}(t)$  que  $j$  domina na iteração  $t$ .

A densidade do indivíduo  $i$  assume a seguinte forma:

$$D_i(t) = \frac{1}{\lambda_i^k(t) + 2} \quad (5.6)$$

onde  $\lambda_i^k(t)$  é a distância euclidiana entre o indivíduo  $i$  e seu  $k$ -ésimo vizinho mais próximo, avaliada no espaço de objetivos do problema. De acordo com a proposta de Zitzler et al. (2002),  $k = \sqrt{|\mathbf{P}(t)|}$ , ou seja,  $k$  é dado pela raiz quadrada do número de indivíduos presentes na população na iteração  $t$ .

Como o *fitness* do algoritmo SPEA2 deve ser minimizado ao longo da evolução do algoritmo, para manter a consistência com o algoritmo *cob-aiNet* original a métrica de *fitness* da *cob-aiNet[MO]* foi definida conforme a Equação 5.7, que nada mais é que o valor de  $f_i^{SPEA2}(t)$  normalizado em  $[0, 1]$  e invertido para que seja maximizado:

$$f_i^{Ag}(t) = 1 - \frac{f_i^{SPEA2}(t) - \min_j(f_j^{SPEA2}(t))}{\max_j(f_j^{SPEA2}(t)) - \min_j(f_j^{SPEA2}(t))}, \quad (5.7)$$

onde  $f_i^{Ag}(t) \in [0, 1]$  é o *fitness* da célula  $i$  na iteração  $t$  e  $f_i^{SPEA2}(t)$  é o valor de *fitness* do algoritmo SPEA2 para a célula  $i$ , calculado conforme a Equação 5.3.

Um aspecto importante a destacar sobre esta métrica de *fitness* adotada no algoritmo *cob-aiNet[MO]* é o fato de seu valor para cada indivíduo depender de todas as demais células na população. Sendo assim, toda vez que alguma célula da população sofrer alguma modificação ou forem inseridos ou removidos indivíduos, é necessário recalculá-la o *fitness* de *todas* as células, o que eleva consideravelmente o custo computacional do algoritmo. No entanto, esta não é uma característica exclusiva da *cob-aiNet[MO]*, já que todos os demais algoritmos que se baseiam no critério de não dominância e utilizam métricas de *fitness* relativas aos demais indivíduos na população sofrem deste mesmo problema.

Quanto à métrica de afinidade entre as células da população, que é essencial para o mecanismo

de variação de concentrações dos indivíduos, não foi necessária nenhuma alteração em relação à proposta original da cob-aiNet, ou seja, a afinidade total de cada célula da população em relação às demais é dada por:

$$f_i^{Ab}(t) = \begin{cases} \frac{\sum_{j \in J} C_t^j \cdot [\sigma_s - d(i, j)]}{\sum_{j \in J} C_t^j} & \text{se } J \neq \emptyset \\ 0 & \text{caso contrário} \end{cases}, \quad (5.8)$$

onde  $f_i^{Ab}(t)$  é a afinidade total entre a célula  $i$  e as demais células da população na iteração  $t$ ,  $J$  é o conjunto de anticorpos *melhores* que  $i$  e que estão *mais próximos* de  $i$  que um limiar  $\sigma_s$  (raio de afinidade ou raio de supressão),  $C_t^j$  é a concentração da célula  $j$  na iteração  $t$  e  $d(i, j)$  é o valor de uma métrica de distância entre as células  $i$  e  $j$ , avaliada no *espaço de variáveis* do problema.

O mecanismo de concentrações das células na população da cob-aiNet[MO] é basicamente o mesmo introduzido na cob-aiNet, ou seja, cada novo indivíduo é inicializado com uma concentração  $C_0$  (vide Pseudo-código 5.1) e, a cada iteração, este valor é atualizado conforme a Equação 5.9

$$C_{t+1}^i = \begin{cases} \min[(\alpha C_t^i - f_i^{Ab}(t)), 1] & \text{se } \alpha C_t^i - f_i^{Ab}(t) \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (5.9)$$

onde  $C_{t+1}^i$  e  $C_t^i \in [0, 1]$  são, respectivamente, as concentrações nova e antiga da célula  $i$ ,  $f_i^{Ab}(t)$  é a afinidade do anticorpo  $i$  com as demais células da população (dada pela Equação 5.8) e  $\alpha$  é dado pela Equação 5.10.

$$\alpha = \begin{cases} 1 + 0,1 \cdot f_i^{Ag}(t) & \text{se } f_i^{Ab}(t) = 0 \\ 0,85 & \text{caso contrário} \end{cases}, \quad (5.10)$$

onde  $f_i^{Ag}(t)$  é o *fitness* da célula  $i$ .

A única diferença entre o mecanismo de concentrações da cob-aiNet[MO] e da cob-aiNet é a porcentagem mínima de redução da concentração que um indivíduo sofre quando ele se encontra na vizinhança de outras soluções de maior *fitness*. Enquanto que na cob-aiNet esta redução é de 30%, foi verificado empiricamente que, para a versão multiobjetivo do algoritmo, uma redução de 15% se mostra mais apropriada (vide Equação 5.10).

### 5.3.2 Seleção e Inserção de Novos Indivíduos

Da mesma maneira que o algoritmo cob-aiNet, o operador de seleção de novos indivíduos da cob-aiNet[MO], aplicado após as etapas de clonagem e hipermutação (vide Pseudo-código 5.1), também é responsável pela inserção de novos indivíduos que sejam promissores e tenham sido gerados em

posições do espaço de busca supostamente ainda inexploradas pelas demais soluções já presentes na população. No entanto, a maneira com que são escolhidos indivíduos para serem inseridos na população da *cob-aiNet[MO]* é diferente do que ocorre na *cob-aiNet*.

De maneira geral, o processo de seleção e inserção na *cob-aiNet[MO]* se dá basicamente em três etapas:

1. O melhor indivíduo (aquele com maior  $f_i^{Ag}(t)$ ) de cada conjunto de células, formado por cada célula original e seus respectivos clones (após hipermutação), é selecionado como o representante deste conjunto de células na próxima iteração do algoritmo e herda o valor de concentração da célula original.
2. Todos os clones gerados após a etapa de hipermutação (a partir de todas as células originais) são comparados entre si, e aqueles não dominados se tornam candidatos a serem inseridos na população na próxima iteração do algoritmo.
3. Por fim, todas estas células candidatas são comparadas aos indivíduos já presentes na população do algoritmo e, caso alguma delas não esteja na vizinhança de nenhuma outra célula, ou seja, caso a distância euclidiana no espaço de variáveis não seja menor que  $\sigma_s$  para nenhuma outra célula na população, esta célula candidata recebe concentração inicial  $C_0$  e é inserida na população.

É importante ressaltar que, da mesma maneira que para o algoritmo *cob-aiNet*, todas as comparações que se utilizam do limiar de supressão  $\sigma_s$  o fazem sempre relativamente à maior distância possível no espaço de variáveis do problema, conforme previamente discutido na Seção 3.3.3.

### 5.3.3 Busca Local

Por fim, a última diferença entre os algoritmos *cob-aiNet* e *cob-aiNet[MO]* se dá no operador de busca local. Da mesma maneira que a *cob-aiNet*, a *cob-aiNet[MO]* utiliza o operador *LocalSearch1*, proposto como parte do algoritmo *MTS* (Tseng & Chen, 2008) e apresentado na Seção 3.3.4. A estrutura geral do operador *LocalSearch1* utilizado na *cob-aiNet[MO]* é exatamente a mesma previamente apresentada no Pseudo-código 3.4, sendo que a única diferença se dá na forma com que as soluções são comparadas.

Como dito anteriormente, a métrica de *fitness* empregada na *cob-aiNet[MO]* avalia cada célula da população relativamente às demais presentes na população. Sendo assim, cada modificação que um dado indivíduo sofre durante a evolução da busca pode ter um impacto direto no *fitness* de todos os outros indivíduos, o que exige que tal métrica seja constantemente recalculada para que represente

corretamente a qualidade das soluções atuais na população. Como o operador de busca local aplica variações em *todas* as dimensões de *todos* os indivíduos na população, a utilização direta do *fitness* de cada célula nas comparações entre as novas soluções geradas via busca local e a solução original acabaria resultando em um custo computacional proibitivo, já que, a cada modificação, todos os indivíduos na população precisariam ser reavaliados. Diante disso, foi utilizado aqui apenas o critério de *dominância restrita* para comparação de soluções no operador de busca local, não sendo considerado nenhum aspecto relativo à diversidade, seja no espaço de variáveis ou no espaço de objetivos. Ao adotar esta abordagem, a busca local se torna um operador que contribui apenas para a convergência das soluções para a fronteira de Pareto do problema, cabendo aos demais operadores do algoritmo cob-aiNet[MO] tratar adequadamente a questão da diversidade.

## 5.4 Considerações Finais

Neste capítulo, foi apresentada uma visão geral dos principais conceitos associados à otimização multiobjetivo, além de também ter sido introduzido o algoritmo cob-aiNet[MO] (*Concentration-based Artificial Immune Network for Multiobjective Optimization*), proposto como uma extensão do algoritmo cob-aiNet voltada para o tratamento de problemas multiobjetivo. Além disso, a cob-aiNet[MO] foi posicionada frente a outras propostas da literatura e suas principais diferenças em relação ao algoritmo original cob-aiNet foram discutidas em detalhes.

No próximo capítulo, serão apresentados e discutidos diversos experimentos computacionais, feitos com o intuito de avaliar o algoritmo cob-aiNet[MO] tanto em relação à sua capacidade de localização da fronteira de Pareto quanto em relação à sua capacidade de manutenção de diversidade, seja no espaço de variáveis ou no espaço de objetivos. Além disso, serão feitas também comparações entre a cob-aiNet[MO] e diversas outras abordagens da literatura.



## Capítulo 6

### Resultados Experimentais – *cob-aiNet[MO]*

Para avaliar a capacidade de manutenção de diversidade do algoritmo *cob-aiNet[MO]*, tanto no espaço de variáveis quanto no espaço de objetivos, e também a sua eficácia em encontrar bons conjuntos de soluções não dominadas para problemas de otimização multiobjetivo, foram feitos experimentos com dez problemas de teste extraídos da literatura, cujos resultados serão discutidos neste capítulo. Tais experimentos foram divididos em três partes: na primeira delas, o algoritmo proposto foi aplicado a quatro problemas que possuem conjuntos de Pareto bem definidos, para que se avaliasse principalmente a capacidade de manutenção de diversidade no espaço de variáveis, enquanto que na segunda parte buscou-se avaliar o desempenho geral da ferramenta em problemas de maior dimensão. Para isto, nesta segunda etapa foi adotado o conjunto de testes conhecido como ZDT (Zitzler et al., 2000). Por fim, a *cob-aiNet[MO]* foi aplicada a um dos problemas do conjunto de teste conhecido como DTLZ (Deb et al., 2002b), para que fosse possível verificar se a ferramenta proposta também é capaz de tratar adequadamente problemas multiobjetivo com mais de dois critérios a serem otimizados.

Em todos os experimentos (exceto aqueles que envolvem o problema DTLZ) a *cob-aiNet[MO]* foi comparada com as principais ferramentas da literatura capazes de manter a diversidade no espaço de variáveis, além dos algoritmos VIS (*Vector Immune System* – Freschi & Repetto, 2006), baseado no paradigma de sistemas imunológicos artificiais, e NSGA-II (*Non-dominated Sorting Genetic Algorithm II* – Deb et al., 2002a), até hoje considerado uma das ferramentas estado-da-arte para otimização multiobjetivo. Já nos experimentos com o problema com mais de dois objetivos, a *cob-aiNet[MO]* foi comparada apenas com o algoritmo NSGA-II.

Este capítulo está estruturado em cinco seções principais. Na Seção 6.1, serão brevemente apresentados os problemas de teste que foram utilizados nos experimentos que serão tratados aqui<sup>10</sup>, bem como os algoritmos utilizados nas comparações e as métricas adotadas para avaliar os resultados. Na Seção 6.2, serão apresentados e discutidos os resultados obtidos na primeira parte dos experimentos,

---

<sup>10</sup>Uma explicação detalhada sobre todos os problemas tratados aqui pode ser encontrada na Seção A.2 do Apêndice A.

enquanto que os resultados obtidos para os problemas ZDT serão detalhados na Seção 6.3. Por fim, os experimentos feitos com o problema DTLZ serão discutidos na Seção 6.4 e as considerações finais sobre este capítulo serão apresentadas na Seção 6.5.

## 6.1 Metodologia Experimental

Nesta seção, serão brevemente apresentados os problemas de teste que foram adotados neste capítulo, bem como os algoritmos utilizados nas comparações com a *cob-aiNet[MO]* e as métricas empregadas para avaliar os resultados obtidos por cada algoritmo.

### 6.1.1 Problemas de Teste

A primeira parte dos experimentos tratados neste capítulo, que será apresentada na Seção 6.2, teve como principal intuito avaliar e comparar a capacidade do algoritmo *cob-aiNet[MO]* de encontrar não só uma boa *fronteira de Pareto*, mas também um bom *conjunto de Pareto*. Para isso, foram utilizados quatro problemas multimodais de otimização multiobjetivo, que apresentam características distintas entre si:

- **Deb & Tiwari:** proposto por Deb & Tiwari (2005) para avaliar o algoritmo conhecido como *omni-optimizer*, é um problema biobjetivo, multimodal, com fronteira de Pareto convexa e que apresenta o conjunto de Pareto distribuído em regiões separadas e bem definidas do espaço de variáveis (vide Seção A.2.1.1). O problema de Deb & Tiwari possui  $n = 5$  variáveis, sendo  $x_i \in [0, 6] \forall i \in \{1, \dots, n\}$ .
- **EBN:** consiste em uma das funções da classe EBN de problemas proposta por Emmerich et al. (2005). Os problemas da classe EBN são interessantes pois todas as soluções factíveis do problema, ou seja, dentro do domínio  $[0, 1]^n$  onde  $n = 10$ , são Pareto-ótimas, o que os torna muito úteis para avaliar a capacidade de cada algoritmo distribuir as soluções no espaço de variáveis. Nesta tese, o problema EBN selecionado, também biobjetivo, apresenta fronteira de Pareto linear (vide Seção A.2.1.2).
- **Two-on-One 4:** pertencente à classe de problemas denominada *Two-on-One* (Preuss et al., 2006), é um problema biobjetivo bidimensional, cujo primeiro objetivo corresponde a um polinômio de 4ª ordem e o segundo à função esfera (vide Seção A.2.1.3). Dentre as cinco variantes existentes para este problema, nesta tese foi escolhida a quarta (*Two-on-One 4*), por apresentar fronteira de Pareto convexa e dividida em duas regiões.

- **Lamé Supersphere:** pertence à família de problemas proposta por Emmerich & Deutz (2007), cujas fronteiras de Pareto possuem geometria esférica ou superesférica. Neste problema, também biobjetivo, a fronteira de Pareto é não convexa e o conjunto de Pareto, de dimensão  $n = 4$ , consiste basicamente em conjuntos de soluções posicionadas em segmentos de linha paralelos e equiespaçados. A formulação detalhada deste problema pode ser encontrada na Seção A.2.1.4.

Já a segunda parte dos experimentos, que será apresentada na Seção 6.3, visou avaliar o desempenho geral do algoritmo cob-aiNet[MO] em problemas mais desafiadores que os discutidos anteriormente, sendo que, para isso, foram utilizados os problemas conhecidos como ZDT (Zitzler et al., 2000). Esta classe de problemas, muito utilizada na literatura (Castro & Von Zuben, 2010b; Deb et al., 2002a; Deb & Tiwari, 2005; Freschi & Repetto, 2006; Gao & Wang, 2010; Zitzler et al., 2002; Zhou et al., 2009), é composta de seis problemas biobjetivo com características diversas, sendo que cinco deles foram empregados aqui<sup>11</sup>:

- **ZDT1:** é um problema com  $n = 30$  variáveis e fronteira de Pareto contínua, convexa e com soluções distribuídas uniformemente (vide Seção A.2.2.1).
- **ZDT2:** análogo ao problema ZDT1, mas com fronteira de Pareto não convexa (vide Seção A.2.2.2).
- **ZDT3:** também possui  $n = 30$  variáveis e tem como principal característica uma fronteira de Pareto descontínua (vide Seção A.2.2.3).
- **ZDT4:** problema com  $n = 10$  variáveis e fronteira de Pareto convexa. A principal dificuldade deste problema está no fato de apresentar cerca de  $8 \cdot 10^{11}$  soluções Pareto-ótimas locais, o que leva à formação de 100 fronteiras, das quais apenas uma é a fronteira de Pareto do problema (vide Seção A.2.2.4).
- **ZDT6:** como o problema ZDT4, também possui  $n = 10$  variáveis, mas sua fronteira de Pareto não é convexa e a densidade de soluções na região ótima de Pareto não é uniforme (vide Seção A.2.2.5).

Por fim, na Seção 6.4, a cob-aiNet[MO] foi aplicada a um problema com três objetivos, pertencente à classe de problemas de teste conhecida na literatura como DTLZ (Deb et al., 2002b). O intuito principal deste último experimento com a ferramenta proposta foi verificar se a cob-aiNet[MO] é capaz de tratar adequadamente problemas com número de critérios superior a dois. De maneira geral,

---

<sup>11</sup>O problema ZDT5 não foi utilizado por se tratar de uma função booleana definida sobre cadeias binárias, o que foge do escopo deste trabalho.

as meta-heurísticas voltadas para otimização multiobjetivo tendem a encontrar sérias dificuldades ao lidarem com problemas com três ou mais critérios, uma vez que o aumento da dimensão do espaço de objetivos leva a uma redução da pressão evolutiva em direção à fronteira de Pareto, já que os indivíduos da população tendem a se espalhar pelo espaço de objetivos e se tornarem mutuamente não dominados (Sato et al., 2007). Existem algumas técnicas na literatura que visam minimizar os efeitos negativos destes problemas com muitos objetivos (de Carvalho & Pozo, 2010; López Jaimes & Coello Coello, 2009), mas elas não serão tratadas aqui por estarem fora do escopo deste trabalho.

Os problemas DTLZ (Deb et al., 2002b) correspondem a uma classe de nove problemas que têm como principal característica a escalabilidade quanto ao número de objetivos a serem otimizados, ou seja, o usuário pode definir quantos critérios ele quer utilizar em cada um destes problemas. Na Seção 6.4, foi utilizado o segundo problema da classe DTLZ, conhecido como *DTLZ2*, que possui fronteira de Pareto esférica e não convexa. Uma explicação detalhada sobre a formulação do problema *DTLZ2* pode ser encontrada na Seção A.2.3.

### 6.1.2 Algoritmos Utilizados nas Comparações

Com o intuito de avaliar o desempenho da *cob-aiNet[MO]* e posicioná-la frente a outros algoritmos da literatura, nos experimentos reportados neste capítulo a ferramenta proposta foi comparada a cinco outras abordagens da literatura, sendo três delas (*omni-optimizer*, *KPI* e *MMEA*) as principais representantes das meta-heurísticas populacionais para otimização multiobjetivo que possuem mecanismos de manutenção de diversidade no espaço de variáveis.

O algoritmo *omni-optimizer* foi proposto por Deb & Tiwari (2005) para ser uma ferramenta de propósito geral, capaz de resolver tanto problemas de otimização mono-objetivo quanto de otimização multiobjetivo, sem exigir do usuário modificações significativas na estrutura da ferramenta. De maneira geral, o *omni-optimizer* pode ser visto como uma extensão do algoritmo NSGA-II (Deb et al., 2002a), que passou a considerar também a questão da diversidade no espaço de variáveis. Isso tornou o algoritmo capaz de tratar problemas multimodais de maneira mais eficiente, ou seja, o *omni-optimizer* passou a apresentar uma menor susceptibilidade a ótimos locais quando comparado ao NSGA-II, e também passou a permitir a obtenção de conjuntos de Pareto mais completos.

Da mesma maneira que o *omni-optimizer*, o algoritmo *KPI*, proposto por Chan & Ray (2005), é um algoritmo evolutivo que avalia a qualidade e a diversidade de cada solução presente em sua população, tanto no espaço de objetivos quanto no de variáveis. No entanto, no algoritmo *KPI* isto se dá através da utilização de duas métricas: (i) a *Contribuição de Lebesgue*, que é responsável por indicar quais são as soluções que, além de estarem posicionadas em regiões menos povoadas do espaço de objetivos, também correspondem aos melhores indivíduos sob o critério de dominância; e (ii) a *Contagem de Vizinhança*, que é uma estimativa da densidade de soluções presentes na região

de cada indivíduo no espaço de variáveis do problema. Apesar do termo *KPI* não ter sido utilizado originalmente por Chan & Ray (2005), ele vem sendo empregado em outros trabalhos da literatura para designar tal proposta (Zhou et al., 2009), sendo por isso também adotado aqui.

Já a ferramenta MMEA (*Probabilistic Model-based Multiobjective Evolutionary Algorithm* – Zhou et al., 2009) é um algoritmo de estimação de distribuição (EDA – Larrañaga & Lozano, 2002) que, a cada iteração, divide as soluções presentes em sua população em sub-populações (através de uma técnica de *agrupamento*), conforme sua distribuição no espaço de objetivos. Após esta divisão dos indivíduos em grupos, é feita então uma *análise de componentes principais* (Jolliffe, 2002) para identificar a dimensão do conjunto de Pareto correspondente a cada sub-população, sendo em seguida criados modelos probabilísticos (um para cada grupo) que buscam reproduzir a distribuição das soluções ótimas de Pareto no espaço de variáveis. A partir destes modelos probabilísticos, os indivíduos que constituirão a população na próxima geração são então amostrados, e o processo se repete. O algoritmo MMEA apresentou bons resultados no trabalho de Zhou et al. (2009), superando até mesmo os algoritmos *omni-optimizer* e *KPI* nos problemas lá tratados, o que faz com que alguns autores já o considerem como a ferramenta estado-da-arte dentre os algoritmos de otimização multiobjetivo capazes de manter a diversidade no espaço de variáveis.

Além destes três algoritmos que, como a *cob-aiNet[MO]*, visam especificamente manter a diversidade no espaço de variáveis, neste capítulo foram feitas comparações também com outras duas ferramentas: o algoritmo *NSGA-II* (*Non-dominated Sorting Genetic Algorithm II* – Deb et al., 2002a), que é um algoritmo genético considerado ainda hoje estado-da-arte dentre as meta-heurísticas para otimização multiobjetivo, e o algoritmo *VIS* (*Vector Immune System* – Freschi & Repetto, 2006) que, como a *cob-aiNet[MO]*, é baseado no paradigma de sistemas imunológicos artificiais e inspirado tanto no princípio de Seleção Clonal (Burnet, 1978) quanto na teoria da Rede Imunológica (Jerne, 1974). Apesar da inspiração na teoria da Rede Imunológica, o algoritmo *VIS* realiza a supressão de soluções a partir de seu grau de similaridade no espaço de objetivos do problema, o que faz com que o único mecanismo responsável por estimular a diversidade no espaço de variáveis do problema esteja associado à inserção periódica na população de novos indivíduos aleatoriamente gerados.

Para a realização dos experimentos que serão descritos nas próximas seções, todos os algoritmos discutidos aqui foram implementados a partir das descrições presentes em suas respectivas publicações, exceto o algoritmo MMEA, cujo código-fonte original está disponível no seguinte endereço (último acesso em 29 de abril de 2011):

### 6.1.3 Métricas de Avaliação e Metodologia

Cada um dos algoritmos discutidos na Seção 6.1.2, além da *cob-aiNet[MO]*, foi aplicado aos nove primeiros problemas de teste apresentados na Seção 6.1.1 e os resultados obtidos foram avaliados através de cinco métricas, que serão discutidas aqui. No caso específico do problema DTLZ2, que será tratado na Seção 6.4, apenas a *cob-aiNet[MO]* e o algoritmo NSGA-II foram comparados, sendo para isto utilizada a métrica de cobertura entre dois conjuntos, que será apresentada na Seção 6.1.3.4.

#### 6.1.3.1 Hipervolume

A métrica de *hipervolume* (também conhecida como *S-metric* ou *medida de Lebesgue*) corresponde ao *volume*, no espaço de objetivos do problema, formado pelas soluções não dominadas encontradas por cada algoritmo e por um ponto de referência pré-definido (Coello Coello et al., 2007). No caso específico de problemas biobjetivo, a métrica de hipervolume pode ser interpretada como a *área* compreendida entre as soluções não dominadas e o ponto de referência pré-definido, como ilustrado na Figura 6.1.

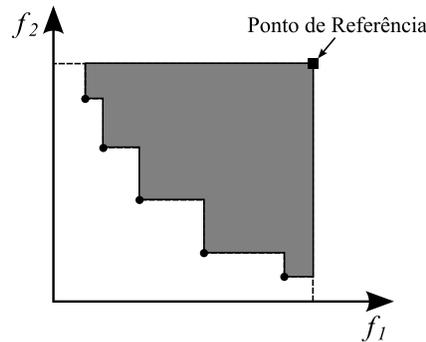


Fig. 6.1: Representação gráfica da métrica de hipervolume, que corresponde à área (representada em cinza na figura) entre as soluções não dominadas (círculos) e um ponto de referência pré-definido. Neste exemplo, foi considerado um problema biobjetivo de minimização.

Quanto maior for o valor da métrica de hipervolume para um dado conjunto de soluções não dominadas, maior será a proximidade destas soluções da fronteira de Pareto do problema e também maior será a cobertura desta fronteira.

Para os problemas que serão discutidos neste capítulo, a métrica de hipervolume foi sempre calculada utilizando-se os seguintes pontos de referência, escolhidos de forma a cobrirem todas as possíveis soluções da fronteira de Pareto de cada problema (vide Figura 6.1):  $[0, 0]$  para o problema *Deb & Tiwari*;  $[1, 1]$  para *EBN*;  $[21, 10]$  para *Two-on-One 4*;  $[1, 1]$  para *Lamé Supersphere*; e  $[1, 1]$  para os problemas *ZDT*. Quando alguma solução retornada por algum algoritmo estiver acima do ponto de referência para um dado problema, sua contribuição para o hipervolume será nula.

### 6.1.3.2 Espaçamento

A métrica de *espaçamento* (do inglês *spacing*) indica o quão bem distribuídas estão as soluções não dominadas retornadas por um dado algoritmo (Coello Coello et al., 2007). Basicamente, esta métrica avalia, no espaço de objetivos, a variância das distâncias entre cada solução em um conjunto  $A$  e seus vizinhos, de forma que, quanto menor for o valor da métrica de espaçamento, mais uniforme será a distribuição das soluções. O cálculo do espaçamento para um conjunto de soluções  $A$  é dado por:

$$S(A) = \sqrt{\frac{1}{|A| - 1} \sum_{i=1}^{|A|} (d_i - \bar{d})^2}, \quad (6.1)$$

onde  $d_i = \min_j (|f_1(\vec{x}_i) - f_1(\vec{x}_j)| + |f_2(\vec{x}_i) - f_2(\vec{x}_j)|)$ ,  $i, j = 1, \dots, |A|$ ,  $|A|$  é o número de soluções não dominadas no conjunto  $A$  e  $\bar{d}$  é o valor médio para todos os  $d_i$ . A aplicação desta métrica geralmente supõe que o algoritmo já tenha convergido para uma solução próxima à fronteira de Pareto do problema.

### 6.1.3.3 Espalhamento Máximo

O *espalhamento máximo* (do inglês *maximum spread*) avalia o comprimento da diagonal formada pelos valores extremos das funções-objetivo observados no conjunto de soluções não dominadas obtido por um dado algoritmo (Zitzler, 1999). Formalmente, o espalhamento máximo para um conjunto  $A$  é dado por:

$$MS(A) = \sqrt{\sum_{m=1}^M \left\{ \max_{i=1}^{|A|} [f_m(\vec{x}_i)] - \min_{i=1}^{|A|} [f_m(\vec{x}_i)] \right\}^2}, \quad (6.2)$$

onde  $\max_{i=1}^{|A|} [f_m(\vec{x}_i)]$  é o valor máximo para a função-objetivo  $m$  dentre as  $|A|$  soluções no conjunto não dominado  $A$ ,  $\min_{i=1}^{|A|} [f_m(\vec{x}_i)]$  é o valor mínimo da função-objetivo  $m$  dentre as  $|A|$  soluções presentes no conjunto não dominado  $A$ , e  $M$  é o número de objetivos do problema. Da mesma maneira que para a métrica de espaçamento, o espalhamento máximo também supõe que a comparação entre os resultados obtidos por duas ferramentas distintas se faz assim que ambas tenham convergido ao final de suas execuções.

### 6.1.3.4 Cobertura entre Dois Conjuntos ( $\mathcal{C}(A, B)$ )

A métrica de *cobertura entre dois conjuntos* (do inglês *two-set coverage* –  $\mathcal{C}(A, B)$ ) calcula a porcentagem de soluções presentes em um dado conjunto  $B$  que são dominadas por, no mínimo, uma

solução presente no conjunto  $A$  (Zitzler, 1999), ou seja:

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|}, \quad (6.3)$$

onde  $|B|$  é o número de soluções em  $B$ .

Quando  $C(A, B) = 1$ , tem-se um indicativo de que todas as soluções no conjunto  $B$  são dominadas pelas soluções em  $A$ , e  $C(A, B) = 0$  significa que nenhuma solução em  $B$  é dominada pelas soluções em  $A$ . É importante ressaltar que esta métrica de cobertura não é simétrica, ou seja,  $C(A, B)$  não é necessariamente igual a  $1 - C(B, A)$ .

### 6.1.3.5 Diversidade baseada em Hipercubos

A *diversidade baseada em hipercubos* (do inglês *hypercube-based diversity*) avalia o espalhamento das soluções não dominadas no espaço de variáveis do problema (Deb et al., 2002a). Para o cálculo desta métrica de diversidade, o domínio do problema é dividido em um número pré-definido de hipercubos, e o valor da métrica corresponderá ao número de hipercubos efetivamente ocupados por soluções (vide Figura 6.2).

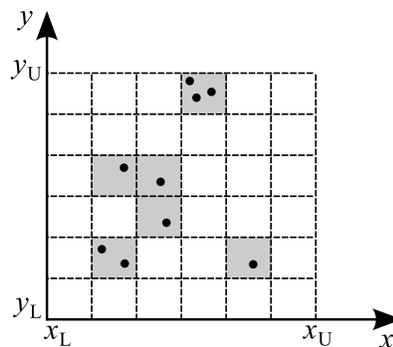


Fig. 6.2: Representação gráfica da métrica de diversidade baseada em hipercubos. Neste exemplo, tal métrica foi calculada dividindo-se o domínio de cada variável ( $x_i \in [x_L, x_U]$  e  $y_i \in [y_L, y_U]$ ) em 6 regiões, o que levou à formação de 36 hiperáreas. Como apenas 6 hiperáreas estão efetivamente ocupadas por soluções, o resultado desta métrica de diversidade para este problema será 6.

Da mesma maneira que as métricas de espaçamento e espalhamento máximo, a diversidade baseada em hipercubos supõe que os algoritmos já tenham efetivamente convergido para um conjunto de soluções não dominadas. Caso tal exigência não seja cumprida, qualquer ferramenta que distribua uniformemente um certo número de indivíduos pelo espaço de busca acabará obtendo valores altos de diversidade, apesar de tais indivíduos não necessariamente corresponderem a soluções para o problema.

Como se trata de uma métrica custosa computacionalmente, principalmente para problemas com

espaços de variáveis de alta dimensão, a diversidade baseada em hipercubos só foi empregada na primeira parte dos experimentos deste capítulo, descritos na Seção 6.2. Para os quatro problemas tratados na Seção 6.2, a métrica de diversidade baseada em hipercubos foi calculada a partir da divisão do domínio de cada variável de cada problema em 20 intervalos, para que os hipercubos pudessem ser obtidos.

Nos experimentos que serão apresentados nas próximas seções, os valores médios e os respectivos desvios padrão de todas estas métricas foram calculados a partir dos resultados obtidos após 25 repetições da execução de cada algoritmo para cada problema, e, como feito anteriormente no Capítulo 4, foi adotado o *Wilcoxon's Rank Sum test* (com limiar 0,05 – Moore et al., 2007) para avaliar a significância estatística dos resultados. Exceto para a métrica de cobertura entre dois conjuntos, para a qual todos os algoritmos foram comparados par a par, o teste de Wilcoxon foi aplicado sempre comparando os resultados obtidos pela cob-aiNet[MO] e por cada um dos demais algoritmos. No caso específico da métrica de cobertura, o teste de Wilcoxon foi aplicado sempre comparando os resultados  $\mathcal{C}(A, B)$  e  $\mathcal{C}(B, A)$  individualmente, para cada par de algoritmos.

## 6.2 Análise da Capacidade de Manutenção de Diversidade no Espaço de Variáveis

Como dito anteriormente, o principal objetivo desta primeira parte dos experimentos com o algoritmo cob-aiNet[MO] é avaliar sua capacidade de manutenção de diversidade no espaço de variáveis de cada problema. No entanto, a qualidade de cada conjunto de soluções não dominadas no espaço de objetivos também será constantemente avaliada, já que a meta principal de qualquer algoritmo de otimização multiobjetivo continua sendo a obtenção do conjunto de soluções que melhor se aproxime da fronteira de Pareto do problema e que apresente a maior cobertura possível desta fronteira.

Todos os algoritmos empregados aqui foram executados 25 vezes para cada problema, sendo que, a cada repetição, o número máximo de avaliações das funções-objetivo foi definido como  $4 \cdot 10^4$ . Os parâmetros de cada algoritmo para cada problema foram ajustados através de uma série de experimentos preliminares, além de também terem sido seguidas indicações fornecidas nos respectivos artigos originais. Diante disso, cada algoritmo foi executado com os seguintes parâmetros:

- **cob-aiNet[MO]:** número de indivíduos na população inicial  $n_{AB} = 100$ ; número mínimo e máximo de clones  $n_{C^{min}} = 3$  e  $n_{C^{max}} = 7$ , respectivamente; concentração inicial  $C_0 = 0,5$ ; tamanho máximo da população  $max_{AB} = 150$ ;  $\beta^i = 1,0$  e  $\beta^f = 0,001$ ; e, finalmente, limiar de supressão  $\sigma_s = 0,060; 0,150; 0,003$  e  $0,070$  para os problemas *Deb & Tiwari, EBN, Two-*

*on-One 4* e *Lamé Supersphere*, respectivamente. O operador de busca local não foi utilizado nesta primeira parte dos experimentos, ou seja,  $LS_{it} = 0$ . O parâmetro  $\sigma_s$  foi ajustado com os valores apresentados anteriormente para que o número final de soluções obtidas pela *cob-aiNet[MO]* fosse o mais próximo possível do dos demais algoritmos.

- **NSGA-II:** tamanho da população igual a 100 indivíduos; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 1,0; e índices de distribuição de *crossover* e de mutação (Deb et al., 2002a) iguais a 20. Tais índices de distribuição dos operadores de *crossover* e mutação são parâmetros que, simplificadaamente, definem o quão espalhadas no espaço de variáveis tenderão estar as novas soluções geradas por cada um deles.
- **omni-optimizer:** tamanho da população igual a 100 indivíduos; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 1,0; índices de distribuição de *crossover* e de mutação (Deb & Tiwari, 2005) iguais a 20; e  $\delta$  para a  $\epsilon$ -dominância igual a  $\delta = 0,001$ .
- **VIS:** tamanho da população inicial igual a 200; número de clones por célula igual a 5; porcentagem de novas células aleatoriamente geradas e introduzidas na população igual a 20%; 5 iterações entre etapas de supressão; e  $\alpha$  igual a 1,0.
- **KP1:** tamanho da população igual a 100; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 0,8; e índices de distribuição de *crossover* e de mutação (Chan & Ray, 2005) iguais a 20.
- **MMEA:** tamanho da população igual a 100;  $\alpha = 1,0$ ;  $\beta = 1,0$ ;  $\theta = 0,8$ ; e número máximo de grupos  $K_{max} = 30$ .

Os valores médios e os desvios padrão das métricas de hipervolume, espaçamento, espalhamento máximo e diversidade baseada em hipercubos, obtidos por cada algoritmo para cada um dos problemas tratados aqui, são fornecidos na Tabela 6.1. Os melhores resultados para cada métrica estão destacados em negrito na Tabela 6.1, enquanto que aqueles para os quais a hipótese nula do teste de Wilcoxon pode ser rejeitada com significância 0,05, quando comparados aos resultados da *cob-aiNet[MO]*, estão marcados com “\*”. Portanto, a presença de “\*” na Tabela 6.1 indica quais são os resultados que podem ser considerados estatisticamente distintos daqueles obtidos pelo algoritmo *cob-aiNet[MO]*. Um resumo com os *ranks* de cada algoritmo frente aos demais, para as métricas cujos resultados são dados na Tabela 6.1, é apresentado na Tabela 6.2. Já os valores médios e os respectivos desvios padrão para a métrica de cobertura, avaliada par a par entre todos os algoritmos para

os problemas *Deb & Tiwari*, *Two-on-One 4* e *Lamé Supersphere*, estão apresentados na Tabela 6.3. Os valores da métrica de cobertura para o problema *EBN* foram omitidos uma vez que, como todas as soluções factíveis deste problema também são Pareto-ótimas, o valor de  $C(A, B)$  será sempre nulo para quaisquer algoritmos  $A$  e  $B$ . Na Tabela 6.3, os resultados devem ser analisados sempre entre as posições equidistantes da diagonal principal, ou seja, deve-se comparar  $C(A, B)$  com  $C(B, A)$ . Dito isso, os valores em negrito correspondem então àquelas situações em que um dado algoritmo  $A$  domina um número maior de soluções de outro algoritmo  $B$  do que  $B$  domina de  $A$ , enquanto que valores marcados por \* correspondem àqueles para os quais a hipótese nula do teste de Wilcoxon (também aplicado para a par) pode ser rejeitada com significância 0,05, ou seja, àqueles em que a distribuição da porcentagem de soluções de  $B$  dominadas pelo algoritmo  $A$  é estatisticamente diferente da distribuição da porcentagem de soluções de  $A$  dominadas por  $B$ , considerando-se as 25 repetições para cada problema.

Tomando primeiramente o problema *Deb & Tiwari*, é possível observar na Tabela 6.3 que o NSGA-II é o algoritmo que gera o conjunto final de soluções não dominadas mais próximo da fronteira de Pareto do problema, uma vez que suas soluções sempre dominam um número maior de soluções dos demais algoritmos do que as soluções dos demais algoritmos dominam suas soluções. Num extremo oposto, o algoritmo MMEA claramente não foi capaz de convergir adequadamente para este problema, já que suas soluções foram capazes de dominar, em média, apenas 1% das soluções geradas pelo algoritmo VIS (e 0% das soluções geradas pelos demais algoritmos), enquanto que as outras ferramentas geraram conjuntos de soluções capazes de dominar, no mínimo, 93% das soluções geradas pelo MMEA. Esta incapacidade do algoritmo MMEA de convergir para este problema fica evidente nos resultados ilustrados na Figura 6.3, em que são apresentados graficamente os conjunto de soluções não dominadas obtidas por cada algoritmo para o problema *Deb & Tiwari*, na repetição em que cada um deles obteve o maior valor para a métrica de hipervolume.

O algoritmo cob-aiNet[MO], por sua vez, foi a ferramenta que apresentou o segundo melhor resultado em relação à métrica de cobertura para o problema *Deb & Tiwari*. Como pode ser visto na Tabela 6.3, o algoritmo proposto neste trabalho foi, em média, capaz de dominar uma quantidade significativamente maior de soluções quando comparado a todas as demais ferramentas utilizadas aqui, exceto o NSGA-II. Tais resultados são mais evidentes quando as comparações são feitas com os algoritmos omni-optimizer, VIS e MMEA.

Com relação às demais métricas, cujos resultados estão apresentados nas Tabelas 6.1 e 6.2, o algoritmo NSGA-II também foi a ferramenta que obteve os melhores valores médios de hipervolume para o problema *Deb & Tiwari*, enquanto que a cob-aiNet[MO] foi o melhor algoritmo segundo todas as demais métricas, principalmente na diversidade baseada em hipercubos. Este resultado significativamente superior na métrica de diversidade indica que, para este problema, a cob-aiNet[MO]

Tab. 6.1: Média ± Desvio Padrão das métricas de *hipervolume*, *espaçamento*, *espalhamento máximo* e *diversidade baseada em hipercubos*. Os melhores resultados estão em negrito. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados aos da *cob-aiNet*) pode ser rejeitada com significância 0,05 estão marcados com \*.

Hiper-volume (maior é melhor)						
	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	VIS	KP1	MMEA
<b>Deb &amp; Tiwari</b>	19, 298 ± 0,020	<b>19, 458 ± 0,007*</b>	19, 308 ± 0,022	19,049 ± 0,080*	19, 352 ± 0,020*	17, 446 ± 0,199*
<b>EBN</b>	0, 485 ± 0,002	0, 486 ± 0,004*	0, 399 ± 0,021*	0, 412 ± 0,012*	0, 490 ± 0,001*	<b>0, 494 ± 0,000*</b>
<b>Two-on-One 4</b>	126, 376 ± 0,033	126, 435 ± 0,365*	126, 298 ± 0,430*	125, 263 ± 1,030*	126, 293 ± 0,354	<b>126, 666 ± 0,006*</b>
<b>Lamé Supersphere</b>	0, 205 ± 0,000	<b>0, 210 ± 0,000*</b>	0, 207 ± 0,000*	0, 205 ± 0,002	0, 207 ± 0,000*	0, 210 ± 0,000*
Espaçamento (menor é melhor)						
	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	VIS	KP1	MMEA
<b>Deb &amp; Tiwari</b>	<b>0, 032 ± 0,012</b>	0, 033 ± 0,004	0,045 ± 0,004*	0, 061 ± 0,015*	0,057 ± 0,007*	0, 212 ± 0,069*
<b>EBN</b>	0, 015 ± 0,005	0, 017 ± 0,003	0, 010 ± 0,005*	0,016 ± 0,009	0, 012 ± 0,002*	<b>0, 003 ± 0,000*</b>
<b>Two-on-One 4</b>	0, 117 ± 0,017	0, 107 ± 0,023*	0,075 ± 0,010*	0, 201 ± 0,047*	0,094 ± 0,010*	<b>0, 031 ± 0,002*</b>
<b>Lamé Supersphere</b>	0, 004 ± 0,002	0,007 ± 0,002*	0,020 ± 0,022*	0,019 ± 0,027*	0,011 ± 0,001*	<b>0, 003 ± 0,000*</b>
Espalhamento Máximo (maior é melhor)						
	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	VIS	KP1	MMEA
<b>Deb &amp; Tiwari</b>	<b>7, 070 ± 0,006</b>	7, 004 ± 0,058*	6, 814 ± 0,147*	6, 688 ± 0,309*	7, 066 ± 0,027	6, 986 ± 0,441
<b>EBN</b>	<b>1, 412 ± 0,003</b>	1, 375 ± 0,017*	0, 550 ± 0,099*	0, 899 ± 0,060*	1, 360 ± 0,010*	1, 333 ± 0,016*
<b>Two-on-One 4</b>	<b>16, 386 ± 0,001</b>	16, 105 ± 0,974	15, 883 ± 1,195*	16, 131 ± 0,419	16, 104 ± 0,974	16, 379 ± 0,005*
<b>Lamé Supersphere</b>	1, 414 ± 0,000	1, 425 ± 0,036*	<b>1, 507 ± 0,205</b>	1, 493 ± 0,163*	1, 420 ± 0,011*	1, 417 ± 0,003*
Diversidade Baseada em Hiper-cubos (maior é melhor)						
	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	VIS	KP1	MMEA
<b>Deb &amp; Tiwari</b>	<b>77, 800 ± 2,739</b>	40, 320 ± 6,706*	64, 720 ± 7,334*	32, 520 ± 4,053*	43, 800 ± 4,272*	24, 520 ± 3,331*
<b>EBN</b>	<b>132, 640 ± 4,172</b>	67, 400 ± 6,110*	99, 480 ± 0,823*	30, 440 ± 5,140*	91, 200 ± 2,198*	99, 880 ± 0,332*
<b>Two-on-One 4</b>	11, 760 ± 0,523	11, 200 ± 1,155*	10, 960 ± 1,744	11, 400 ± 1,658	<b>12, 000 ± 1,683</b>	11, 080 ± 0,400*
<b>Lamé Supersphere</b>	<b>106, 680 ± 4,039</b>	82, 040 ± 3,878*	85, 600 ± 2,814*	69, 160 ± 10,003*	66, 840 ± 3,602*	99, 400 ± 0,816*

Tab. 6.2: Ranks de cada algoritmo, frente aos demais, obtidos a partir das métricas de *hipervolume*, *espaçamento*, *espalhamento máximo* e *diversidade baseada em hipercubos*. Na última linha é apresentado o *rank* médio obtido por cada ferramenta.

Hipervolume						
	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Deb & Tiwari	4	1	3	5	2	6
EBN	4	3	6	5	2	1
Two-on-One 4	3	2	4	6	5	1
Lamé Supersphere	5	1	4	6	3	2
Espaçamento						
	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Deb & Tiwari	1	2	3	5	4	6
EBN	4	6	2	5	3	1
Two-on-One 4	5	4	2	6	3	1
Lamé Supersphere	2	3	6	5	4	1
Espalhamento Máximo						
	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Deb & Tiwari	1	3	5	6	2	4
EBN	1	2	6	5	3	4
Two-on-One 4	1	4	6	3	5	2
Lamé Supersphere	6	3	1	2	4	5
Diversidade Baseada em Hipercubos						
	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Deb & Tiwari	1	4	2	5	3	6
EBN	1	5	3	6	4	2
Two-on-One 4	2	4	6	3	1	5
Lamé Supersphere	1	4	3	5	6	2
Rank Médio $\pm$ Desvio Padrão						
	cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
	2,63 $\pm$ 1,78	3,19 $\pm$ 1,38	3,88 $\pm$ 1,75	4,88 $\pm$ 1,20	3,38 $\pm$ 1,31	3,06 $\pm$ 2,02

possui mecanismos mais eficazes de manutenção de diversidade no espaço de variáveis, o que fica evidente nos resultados dados na Figura 6.4, que ilustram, no espaço de variáveis, os conjuntos finais de soluções não dominadas obtidas pela cob-aiNet[MO] e pelo omni-optimizer, na repetição em que cada algoritmo obteve o melhor valor para a métrica de diversidade baseada em hipercubos. O omni-optimizer foi o algoritmo que apresentou o segundo melhor desempenho segundo o critério de diversidade, e por isso foi utilizado como base de comparação. Como é possível observar na Figura 6.4, a cob-aiNet[MO] foi capaz de encontrar, ou seja, de posicionar indivíduos em todas as regiões do espaço de variáveis que possuem soluções do conjunto de Pareto, enquanto que o omni-

Tab. 6.3: Média  $\pm$  Desvio Padrão dos valores, em porcentagem, da métrica de *Cobertura entre Dois Conjuntos*  $\mathcal{C}(A, B)$ . Os resultados em negrito correspondem às situações em que um algoritmo  $A$  cobre um maior número de soluções de um algoritmo  $B$  do que  $B$  cobre de  $A$ . Os resultados para os quais a hipótese nula do teste de Wilcoxon pode ser rejeitada com significância 0,05 estão marcados com \*.

Deb & Tiwari							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	<b>16 <math>\pm</math> 5*</b>	6 $\pm$ 3*	1 $\pm$ 1*	11 $\pm$ 5*	0 $\pm$ 0*
	NSGA-II	9 $\pm$ 4*	–	4 $\pm$ 2*	0 $\pm$ 1*	7 $\pm$ 3*	0 $\pm$ 0*
	omni-optimizer	<b>25 <math>\pm</math> 5*</b>	<b>44 <math>\pm</math> 5*</b>	–	6 $\pm$ 3*	<b>29 <math>\pm</math> 5*</b>	0 $\pm$ 0*
	VIS	<b>56 <math>\pm</math> 9*</b>	<b>82 <math>\pm</math> 6*</b>	<b>62 <math>\pm</math> 9*</b>	–	<b>67 <math>\pm</math> 9*</b>	1 $\pm$ 2*
	KP1	<b>16 <math>\pm</math> 5*</b>	<b>27 <math>\pm</math> 7*</b>	12 $\pm$ 4*	3 $\pm$ 2*	–	0 $\pm$ 0*
	MMEA	<b>98 <math>\pm</math> 3*</b>	<b>99 <math>\pm</math> 2*</b>	<b>98 <math>\pm</math> 2*</b>	<b>93 <math>\pm</math> 6*</b>	<b>99 <math>\pm</math> 2*</b>	–
Two-onOne 4							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	4 $\pm$ 2*	3 $\pm$ 2*	1 $\pm$ 1*	2 $\pm$ 2*	<b>5 <math>\pm</math> 2*</b>
	NSGA-II	<b>10 <math>\pm</math> 6*</b>	–	2 $\pm$ 1*	1 $\pm$ 1*	3 $\pm$ 4*	<b>8 <math>\pm</math> 5*</b>
	omni-optimizer	<b>12 <math>\pm</math> 3*</b>	<b>15 <math>\pm</math> 5*</b>	–	4 $\pm$ 2*	10 $\pm$ 3*	<b>21 <math>\pm</math> 4*</b>
	VIS	<b>24 <math>\pm</math> 7*</b>	<b>30 <math>\pm</math> 8*</b>	<b>27 <math>\pm</math> 6*</b>	–	<b>24 <math>\pm</math> 5*</b>	<b>38 <math>\pm</math> 7*</b>
	KP1	<b>19 <math>\pm</math> 5*</b>	<b>22 <math>\pm</math> 5*</b>	<b>19 <math>\pm</math> 3*</b>	7 $\pm$ 3*	–	<b>29 <math>\pm</math> 6*</b>
	MMEA	3 $\pm$ 2*	3 $\pm$ 2*	2 $\pm$ 2*	1 $\pm$ 1*	2 $\pm$ 1*	–
Lamé Superspheres							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	3 $\pm$ 2*	1 $\pm$ 1*	1 $\pm$ 1*	2 $\pm$ 1*	1 $\pm$ 1*
	NSGA-II	4 $\pm$ 2*	–	0 $\pm$ 1*	0 $\pm$ 1*	1 $\pm$ 1*	1 $\pm$ 1*
	omni-optimizer	<b>13 <math>\pm</math> 3*</b>	<b>19 <math>\pm</math> 3*</b>	–	<b>9 <math>\pm</math> 4*</b>	<b>13 <math>\pm</math> 3*</b>	<b>16 <math>\pm</math> 4*</b>
	VIS	<b>10 <math>\pm</math> 4*</b>	<b>15 <math>\pm</math> 4*</b>	5 $\pm$ 2*	–	<b>10 <math>\pm</math> 3*</b>	<b>12 <math>\pm</math> 3*</b>
	KP1	<b>10 <math>\pm</math> 3*</b>	<b>11 <math>\pm</math> 4*</b>	4 $\pm$ 2*	5 $\pm$ 2*	–	<b>8 <math>\pm</math> 4*</b>
	MMEA	<b>7 <math>\pm</math> 2*</b>	<b>11 <math>\pm</math> 3*</b>	4 $\pm$ 2*	3 $\pm$ 1*	5 $\pm$ 3*	–

optimizer, mesmo em sua melhor execução, não conseguiu posicionar nenhuma solução nas regiões destacadas por retângulos.

Com relação ao problema *EBN*, o algoritmo MMEA foi o que apresentou o melhor resultado em relação ao hipervolume, enquanto que a *cob-aiNet[MO]* foi superior tanto em espalhamento máximo quanto em diversidade no espaço de variáveis. Quanto ao espaçamento, o algoritmo MMEA também foi a ferramenta que obteve os melhores valores, o que fica evidente nos resultados apresentados

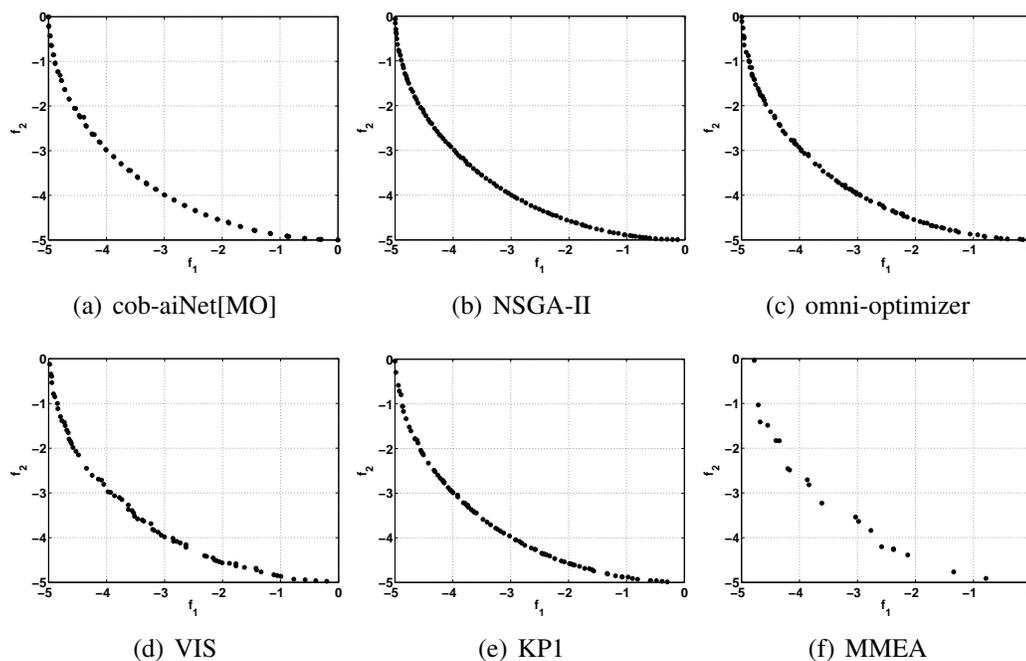


Fig. 6.3: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *Deb & Tiwari*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

na Figura 6.5, que correspondem à representação gráfica, no espaço de objetivos, das soluções obtidas por cada algoritmo para o problema *EBN*. Como é possível observar na Figura 6.5, apesar da *cob-aiNet[MO]* ter conseguido posicionar soluções nos dois extremos da fronteira de Pareto, o que conferiu a este algoritmo um maior valor de espalhamento máximo, de modo geral suas soluções não estão tão uniformemente distribuídas quanto as obtidas pelo *MMEA*, o que garantiu a este último valores melhores para a métrica de espaçamento.

Quanto à diversidade no espaço de variáveis, a Figura 6.6 ilustra os conjuntos de soluções não dominadas obtidos por cada algoritmo para o problema *EBN*, na repetição que levou aos maiores valores de diversidade baseada em hipercubos. Dado que o problema *EBN* possui  $n = 10$  dimensões, por questões de espaço estão representadas na Figura 6.6 apenas as duas primeiras variáveis,  $x_1$  e  $x_2$ . Como visto anteriormente, o problema *EBN* tem como principal característica o fato de todas as suas soluções factíveis pertencerem ao conjunto de Pareto, o que faz com que a distribuição das soluções no espaço de variáveis seja um critério fundamental a ser avaliado para este problema. Sendo assim, como pode ser observado na Figura 6.6, a *cob-aiNet[MO]* é o algoritmo que mais se aproxima da situação desejada, sendo seguida pelas ferramentas *omni-optimizer* e *MMEA*, o que é corroborado pelos resultados de diversidade baseada em hipercubos apresentados na Tabela 6.1.

Para o problema *Two-on-One 4*, os resultados apresentados nas Tabelas 6.1, 6.2 e 6.3 mostram que

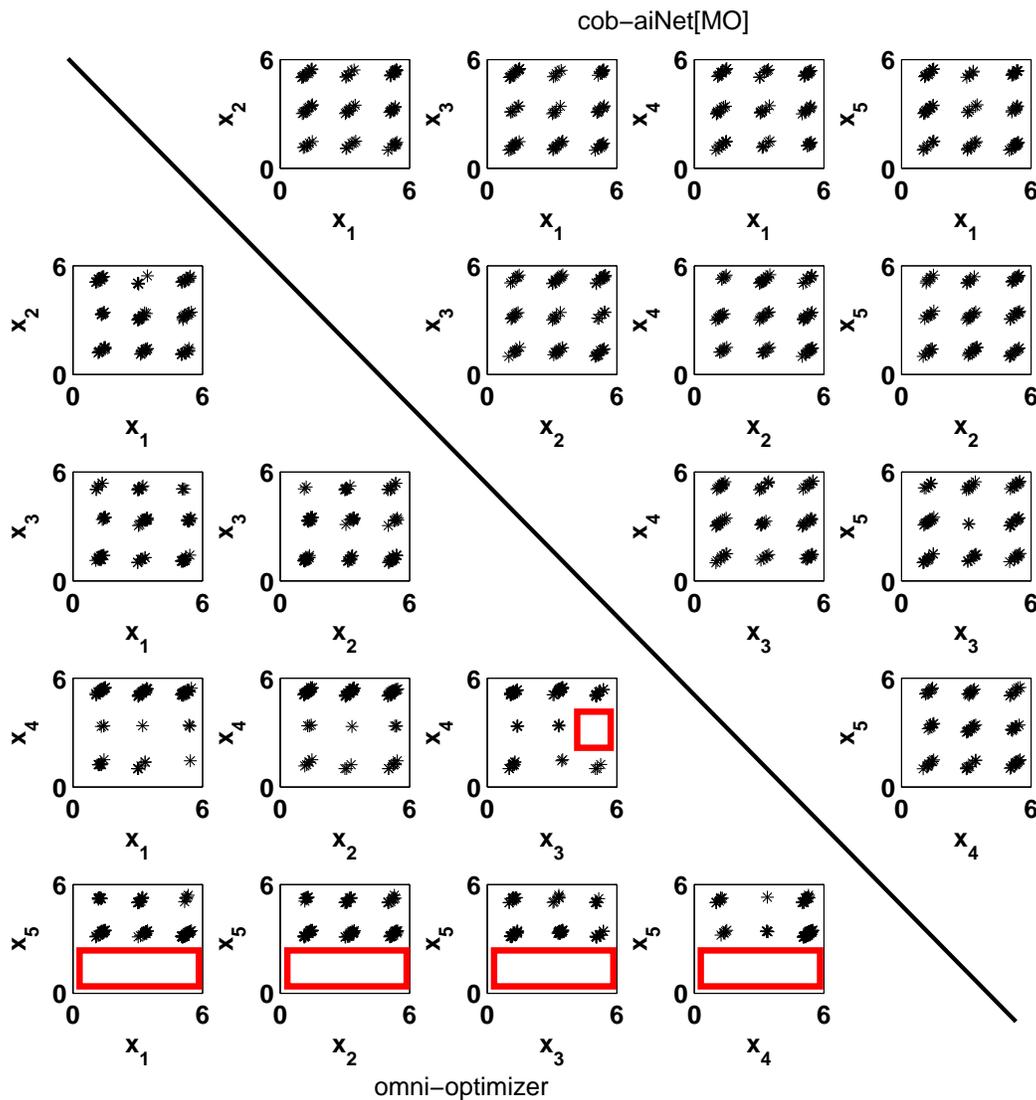


Fig. 6.4: Representação gráfica, no espaço de variáveis, dos conjuntos de soluções não dominadas obtidos pela *cob-aiNet[MO]* (região superior à direita) e pelo *omni-optimizer* (região inferior à esquerda) para o problema *Deb & Tiwari*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de diversidade baseada em hipercubos. As áreas destacadas por retângulos indicam regiões do conjunto de Pareto que não foram identificadas pelo *omni-optimizer*.

a ferramenta MMEA foi o algoritmo que obteve o conjunto de soluções mais próximo da fronteira real do problema (vide Tabela 6.3), o que resultou em um hipervolume significativamente maior que o obtido pelos demais algoritmos. Além disso, o MMEA foi a ferramenta que conseguiu o melhor espaçamento entre as soluções, como pode ser visto graficamente na Figura 6.7, que ilustra, no espaço de objetivos, o conjunto de soluções não dominadas obtido por cada algoritmo na repetição em que

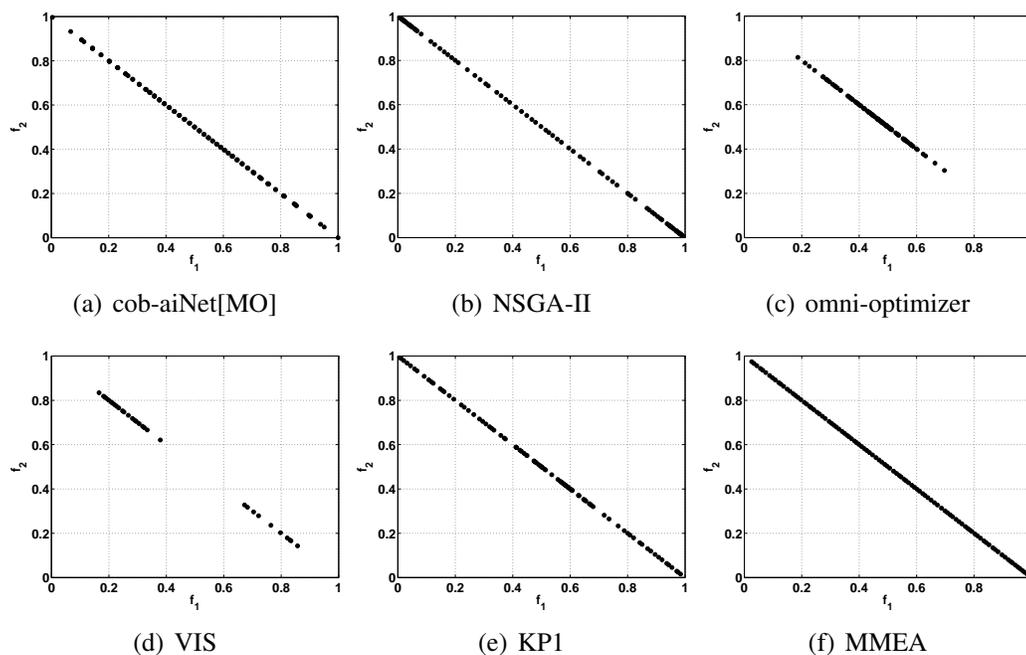


Fig. 6.5: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *EBN*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

cada um deles obteve o melhor hipervolume.

Já a *cob-aiNet[MO]*, por sua vez, obteve o melhor espalhamento e ficou em segundo lugar no critério de diversidade baseada em hipercubos, atrás do algoritmo *KP1*. Esta maior diversidade no espaço de variáveis obtida pelo algoritmo *KP1* fica evidente na Figura 6.8, em que são apresentadas as distribuições das soluções não dominadas retornadas por cada algoritmo no espaço de variáveis. No entanto, é importante destacar que este maior espalhamento conseguido pelo algoritmo *KP1* foi obtido às custas de uma pior aproximação da fronteira de Pareto do problema, como pode ser visto nos resultados da Tabela 6.3: enquanto a *cob-aiNet[MO]* obteve um conjunto de soluções que dominam, em média, 19% das soluções obtidas pelo *KP1*, as soluções obtidas por este último algoritmo são capazes de dominar apenas 2% daquelas retornadas pela *cob-aiNet[MO]*.

Um outro aspecto que cabe ressaltar diante dos resultados para o problema *Two-on-One 4*, apresentados nas Tabelas 6.1 e 6.2, é o fato do algoritmo *omni-optimizer* ter obtido os piores resultados de espalhamento máximo, o que indica que esta ferramenta nem sempre conseguiu localizar os extremos da fronteira de Pareto do problema. O algoritmo *VIS*, por sua vez, foi a ferramenta que obteve o pior espaçamento entre as soluções, o que fica evidente principalmente no segmento inferior da fronteira de Pareto ilustrada na Figura 6.7. Estes dois algoritmos apresentaram resultados semelhantes também para o problema *EBN* (vide Figura 6.5), para o qual não conseguiram cobrir apropriadamente a

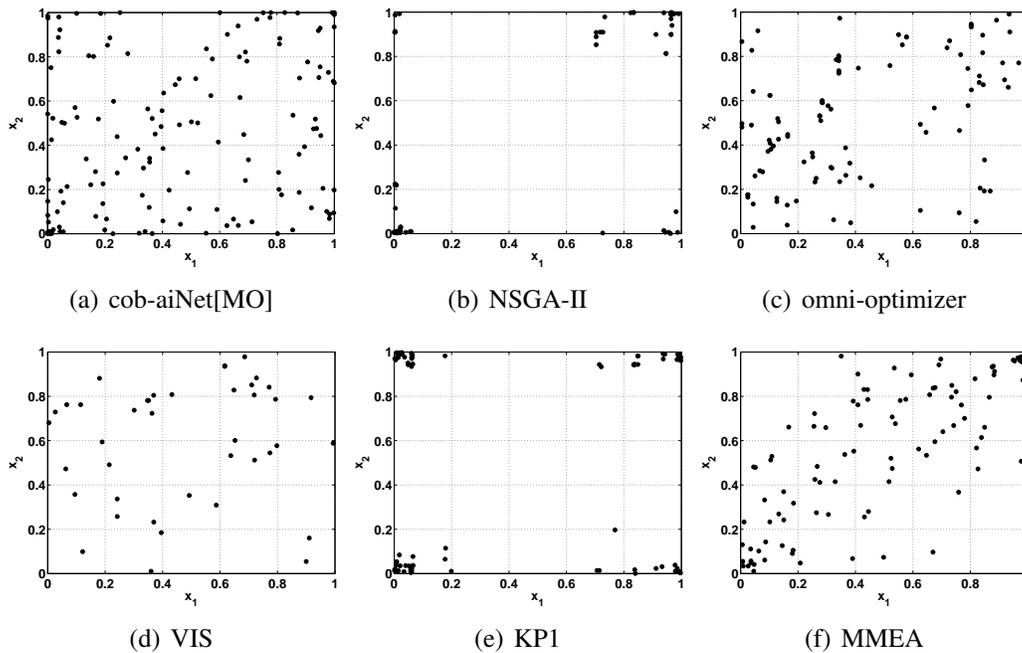


Fig. 6.6: Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *EBN*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de diversidade baseada em hipercubos.

fronteira de Pareto.

Por fim, para o problema *Lamé Supersphere*, os resultados mostram que, pelo critério de cobertura (vide Tabela 6.3 e Figura 6.9), o algoritmo *cob-aiNet[MO]* foi a ferramenta que mais se aproximou da fronteira de Pareto do problema, uma vez que, em média, suas soluções foram capazes de dominar uma porcentagem maior das soluções retornadas pelos outros algoritmos. Além disso, novamente a *cob-aiNet[MO]* foi o algoritmo que apresentou a melhor diversidade no espaço de variáveis (vide Tabelas 6.1 e 6.2), o que pode ser visto nos resultados apresentados na Figura 6.10. Na Figura 6.10, a primeira variável do problema é mostrada em função da média das demais no espaço de variáveis, para que fique evidente as linhas paralelas equiespaçadas que constituem o conjunto de Pareto do problema. Como é possível ver na Figura 6.10, a *cob-aiNet[MO]* foi a única ferramenta capaz de posicionar soluções em cinco segmentos do conjunto de Pareto do problema, enquanto os demais algoritmos concentraram todos os seus indivíduos em apenas três segmentos. Com relação ao hipervolume, o algoritmo NSGA-II apresentou o melhor desempenho, enquanto que, em espaçamento e espalhamento, as melhores ferramentas foram os algoritmos MMEA e omni-optimizer, respectivamente.

De maneira geral, é possível concluir, a partir dos resultados desta primeira parte dos experimentos, que a ferramenta *cob-aiNet[MO]* não só possui mecanismos eficazes de manutenção de diversi-

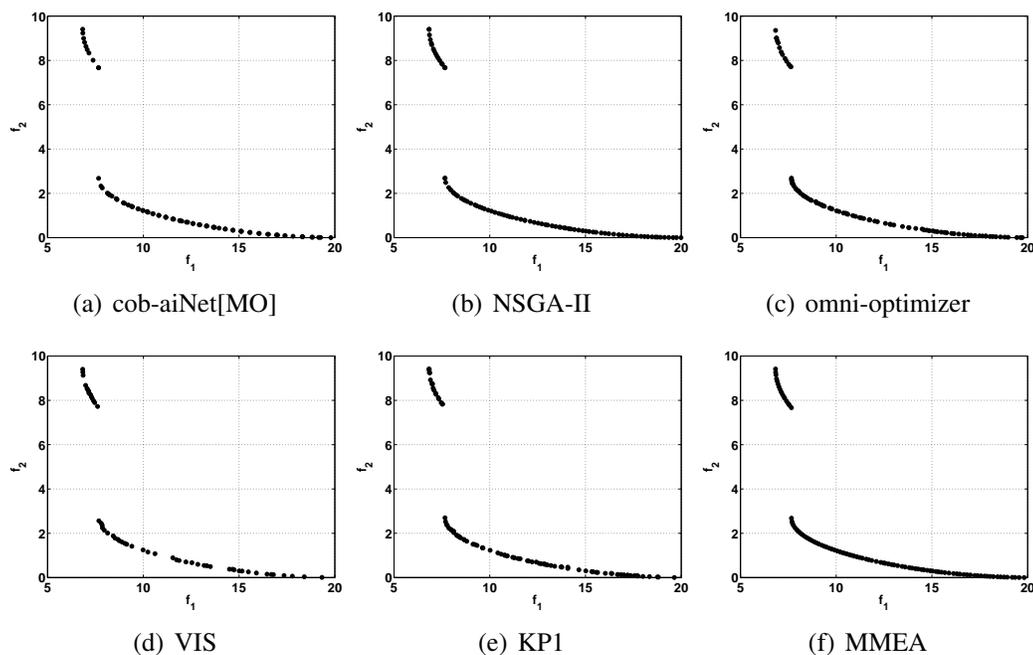


Fig. 6.7: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *Two-on-One 4*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

dade no espaço de variáveis, já que apresentou os melhores resultados para a métrica de diversidade baseada em hipercubos em três dos quatro problemas estudados (e ficou em segundo lugar para o quarto problema), mas também é capaz de aproximar adequadamente a fronteira de Pareto, uma vez que obteve as maiores coberturas entre dois conjuntos para o problema *Lamé Supersphere* e ficou em segundo lugar para os problemas *Deb & Tiwari* e *Two-on-One 4*. Além disso, a *cob-aiNet[MO]* possui bons mecanismos de manutenção de diversidade também no espaço de objetivos dos problemas, uma vez que apresentou os melhores resultados médios de espalhamento para três dos quatro problemas tratados aqui.

Com relação ao critério de hipervolume, os algoritmos *NSGA-II* e *MMEA* apresentaram os melhores resultados em dois dos quatro problemas tratados cada um, enquanto que, em relação à métrica de espaçamento, o algoritmo *MMEA* foi o melhor em três dos quatro problemas. O algoritmo *MMEA* só não apresentou os melhores resultados de espaçamento para o problema *Deb & Tiwari*, para o qual ele também não conseguiu convergir adequadamente. Esta dificuldade do algoritmo *MMEA* no problema *Deb & Tiwari* já havia sido observada anteriormente no próprio artigo em que o algoritmo foi proposto (Zhou et al., 2009), sendo que a possível causa para isto foi atribuída pelos autores ao fato do algoritmo *MMEA* se basear na distância no espaço de objetivos do problema para fazer o agrupamento das soluções em sub-populações, o que provavelmente impede que o algoritmo identifique as

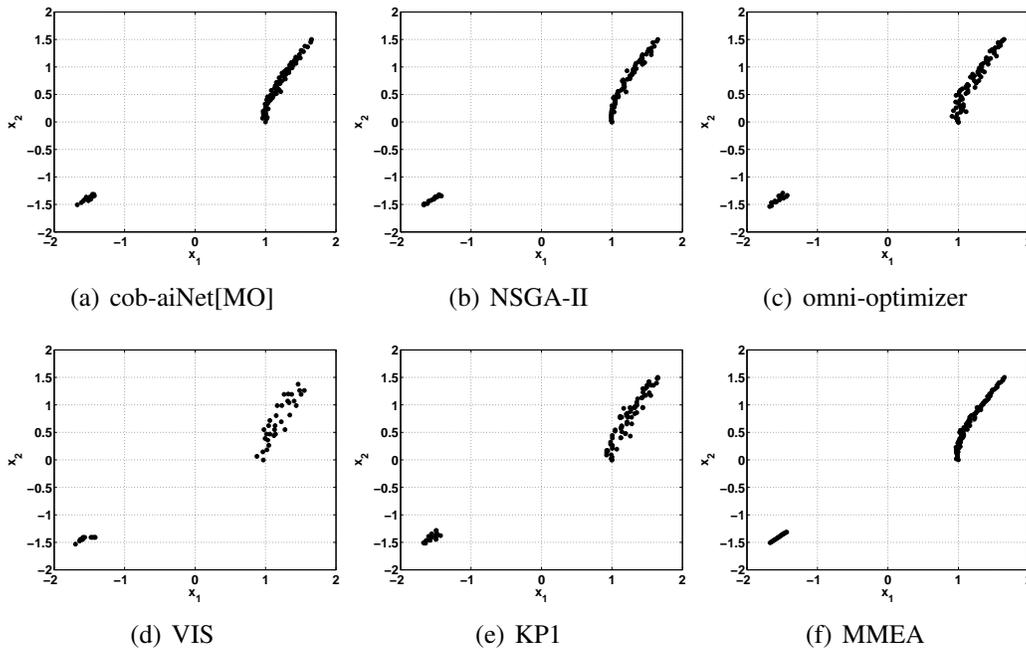


Fig. 6.8: Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *Two-on-One 4*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de diversidade baseada em hipercubos.

regiões distintas do conjunto de Pareto no espaço de variáveis.

É importante destacar aqui que as métricas de hipervolume e espaçamento são diretamente afetadas pelo número de soluções não dominadas obtido por cada algoritmo. No caso do hipervolume, considerando-se dois conjuntos com números diferentes de soluções não dominadas e supondo-se que ambos apresentam espaçamento ideal, aquele conjunto que tiver o maior número de soluções tenderá a apresentar maior hipervolume, já que o número de “retângulos” utilizados no cálculo desta métrica será maior (vide Figura 6.1), ou seja, a resolução da fronteira será melhor. Já no caso de espaçamento, um maior número de soluções para uma mesma fronteira tenderá a levar a indivíduos mais próximos e aumentará a possibilidade de se obter uma menor variância da distância entre eles, o que resultará em melhores valores de espaçamento. Dito isso, é importante destacar que tanto a *cob-aiNet[MO]* quanto o algoritmo *VIS* são ferramentas que *variam o número de soluções* em suas populações ao longo da execução, o que pode levar à obtenção de um número de indivíduos não dominados significativamente diferente do das demais ferramentas. Como pode ser visto na Tabela 6.4, que mostra o número médio de soluções retornadas por cada algoritmo para cada problema, o algoritmo *VIS* geralmente retornou um número significativamente menor de soluções quando comparado aos algoritmos que trabalham com população fixa (*NSGA-II*, *omni-optimizer*, *KP1* e *MMEA*), enquanto que a *cob-aiNet[MO]* apresentou situações em que obteve tanto um número maior quanto um número menor de soluções não

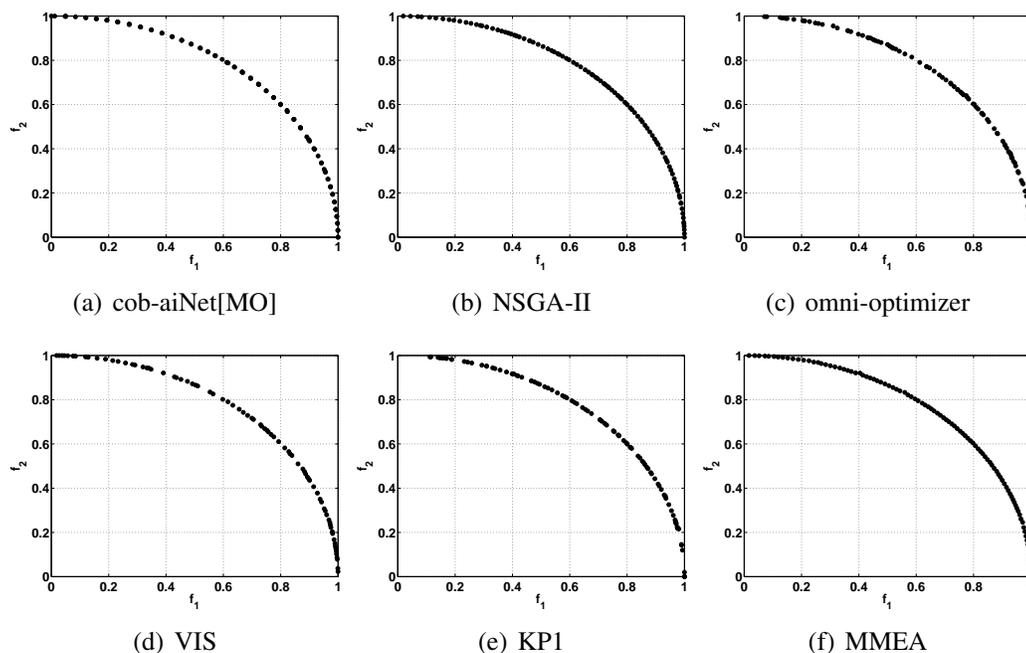


Fig. 6.9: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *Lamé Supersphere*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

dominadas. Esta diferença de comportamento entre os dois algoritmos imuno-inspirados se dá pelo fato da *cob-aiNet[MO]* permitir que o usuário ajuste manualmente o limiar de supressão  $\sigma_s$ , enquanto que o *VIS* procura fazer isto automaticamente, a partir do número de indivíduos na população inicial. Sendo assim, apesar de nos experimentos apresentados aqui o parâmetro  $\sigma_s$  ter sido ajustado visando-se a obtenção de um número final de soluções o mais próximo possível do dos demais algoritmos, é possível perceber, pelos resultados apresentados na Tabela 6.4, que esta tarefa não se mostrou trivial.

Tab. 6.4: Média  $\pm$  Desvio Padrão do número de soluções não dominadas retornado por cada algoritmo para cada problema. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados aos da *cob-aiNet*) pode ser rejeitada com significância 0,05, ou seja, os resultados que possuem distribuição estatisticamente distinta da apresentada pelo algoritmo *cob-aiNet[MO]*, ao longo das 25 repetições, estão marcados com \*.

	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	VIS	KP1	MMEA
<b>Deb &amp; Tiwari</b>	78 $\pm$ 3	100 $\pm$ 0*	100 $\pm$ 0*	67 $\pm$ 7*	100 $\pm$ 0*	25 $\pm$ 4*
<b>EBN</b>	134 $\pm$ 4	100 $\pm$ 0*	100 $\pm$ 0*	30 $\pm$ 5*	100 $\pm$ 0*	100 $\pm$ 0*
<b>Two-on-One 4</b>	84 $\pm$ 5	100 $\pm$ 0*	98 $\pm$ 2*	45 $\pm$ 5*	100 $\pm$ 0*	100 $\pm$ 0*
<b>Lamé Supersphere</b>	107 $\pm$ 4	100 $\pm$ 0*	97 $\pm$ 2*	70 $\pm$ 10*	100 $\pm$ 0*	100 $\pm$ 0*

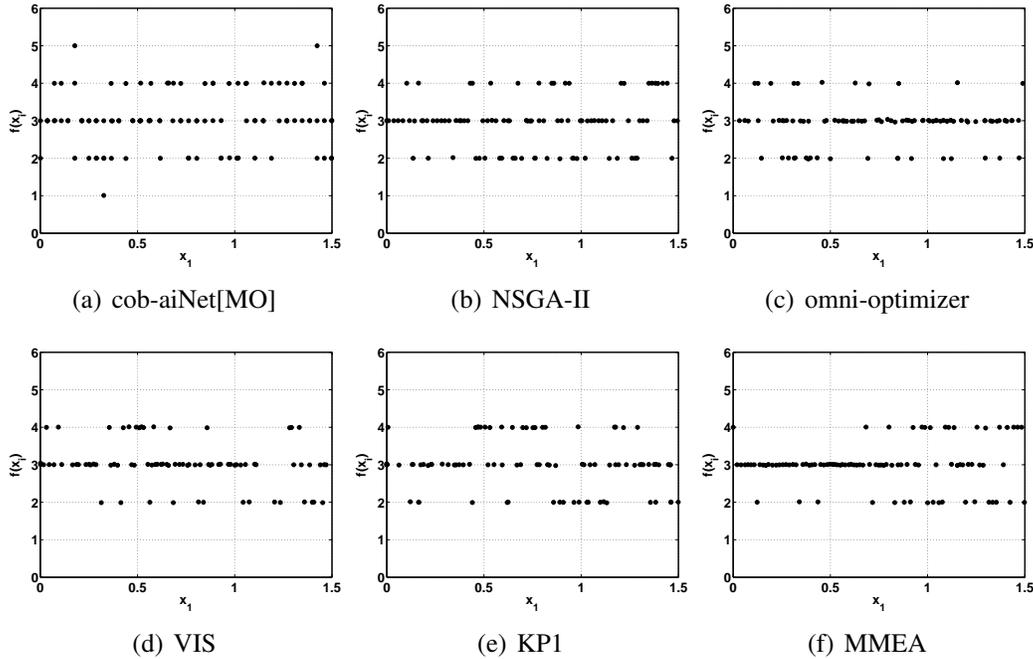


Fig. 6.10: Representação gráfica, no espaço de variáveis, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *Lamé Supersphere*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de diversidade baseada em hipercubos. Nestas figuras, a primeira variável do problema ( $x_1$ ) está ilustrada em função da média das demais, ou seja,  $f(x_i) = \frac{1}{n-1} \sum_{i=2}^n x_i$ , onde  $n = 4$ .

Ainda sobre os resultados apresentados na Tabela 6.4, cabe comentar que o pequeno número médio de soluções não dominadas retornado pelo algoritmo MMEA para o problema *Deb & Tiwari* se deu pelo fato desta ferramenta não ter conseguido convergir apropriadamente para este problema, como discutido anteriormente.

### 6.3 Resultados para os Problemas ZDT

Como dito anteriormente, nesta segunda parte dos experimentos, o algoritmo *cob-aiNet[MO]* foi avaliado com o conjunto de problemas conhecido na literatura como ZDT (Zitzler et al., 2000). A metodologia adotada para estes problemas é análoga à da primeira etapa dos experimentos, exceto pelo fato da métrica de diversidade baseada em hipercubos não ter sido utilizada aqui. Os algoritmos adotados nas comparações foram os mesmos previamente discutidos na Seção 6.1.2. Cada ferramenta foi executada 25 vezes para cada problema, sendo que, a cada repetição, o número máximo de avaliações das funções-objetivo foi definido como  $2,5 \cdot 10^5$ . Os parâmetros de cada algoritmo para cada problema foram novamente ajustados através de uma série de experimentos preliminares, além de também terem sido seguidas indicações fornecidas nos respectivos artigos originais. Sendo assim,

cada algoritmo foi executado com os seguintes parâmetros:

- **cob-aiNet[MO]:** número de indivíduos na população inicial  $nAB = 55$ ; número mínimo e máximo de clones  $nC^{min} = 2$  e  $nC^{max} = 10$ , respectivamente; concentração inicial  $C_0 = 0,5$ ; tamanho máximo da população  $maxAB = 200$ ;  $\beta^i = 0,1$  e  $\beta^f = 0,0001$ ; limiar de supressão  $\sigma_s = 0,0100; 0,0030; 0,0030; 0,0008$  e  $0,0010$  para os problemas  $ZDT1 - 6$ , respectivamente; número de iterações de busca local  $LS_{it} = 5$ ; número de iterações entre etapas de busca local  $LS_{freq} = 5$ ; e intervalo de busca inicial da busca local  $SR$  igual à metade do domínio de cada variável. O parâmetro  $\sigma_s$  foi novamente ajustado aqui, com os valores apresentados anteriormente, de forma que o número final de soluções obtidas pela cob-aiNet[MO] fosse o mais próximo possível do dos demais algoritmos.
- **NSGA-II:** tamanho da população igual a 100 indivíduos; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 1,0; e índices de distribuição de *crossover* e de mutação (Deb et al., 2002a) iguais a 20.
- **omni-optimizer:** tamanho da população igual a 100 indivíduos; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 1,0; índices de distribuição de *crossover* e de mutação (Deb & Tiwari, 2005) iguais a 20; e  $\delta$  para a  $\epsilon$ -dominância igual a  $\delta = 0,001$ .
- **VIS:** tamanho da população inicial igual a 200; número de clones por célula igual a 5; porcentagem de novas células aleatoriamente geradas introduzidas na população igual a 20%; 5 iterações entre etapas de supressão; e  $\alpha$  igual a 0,1.
- **KP1:** tamanho da população igual a 100; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 0,8; e índices de distribuição de *crossover* e de mutação (Chan & Ray, 2005) iguais a 20.
- **MMEA:** tamanho da população igual a 100;  $\alpha = 1,0$ ;  $\beta = 1,0$ ;  $\theta = 0,8$ ; e número máximo de grupos  $K_{max} = 30$ .

Nas Tabelas 6.5 e 6.6, são apresentados os resultados médios e os respectivos desvios padrão obtidos para a métrica de cobertura obtida por cada algoritmo em cada problema tratado aqui. Nestes resultados, o aspecto mais notório é o fato do algoritmo VIS não ter conseguido convergir adequadamente para nenhum dos problemas tratados, o que também pode ser observado nas Figuras 6.11–6.15, que ilustram o conjunto de soluções não dominadas obtido por cada algoritmo para cada problema na repetição em que foi obtido o melhor valor de hipervolume. A partir destes resultados, percebe-se que as soluções retornadas pelo VIS são integralmente dominadas pelas soluções obtidas pelos

demais algoritmos em todas as repetições de todos os problemas, exceto para o ZDT4, em que o algoritmo MMEA também apresentou problemas de convergência e teve, em média, 16% de suas soluções dominadas pelo algoritmo VIS.

O fato do VIS não ter convergido para nenhum destes problemas provavelmente está associado à utilização de apenas um operador de mutação para promover a variação e possível melhoria dos indivíduos da população. Tomando-se a *cob-aiNet[MO]* como base para comparações, ambos os algoritmos se utilizam de um operador de mutação semelhante, mas o algoritmo proposto aqui introduziu uma série de melhorias neste operador (vide Seção 3.3.1) que acabam influenciando diretamente o comportamento do algoritmo. Além disso, também foi empregado na *cob-aiNet[MO]* o mecanismo de busca local, que tende a acelerar a convergência das soluções.

Diante destes resultados, o algoritmo VIS não será mais considerado nas discussões que se seguem, e seus resultados para as métricas de hipervolume, espaçamento e espalhamento não serão apresentados aqui. Sendo assim, na Tabela 6.7 são apresentados os valores médios e os respectivos desvios padrão obtidos pelos demais algoritmos para estas três métricas, para os cinco problemas tratados, enquanto que na Tabela 6.8 são apresentados os *ranks* obtidos por cada um destes algoritmos frente aos demais.

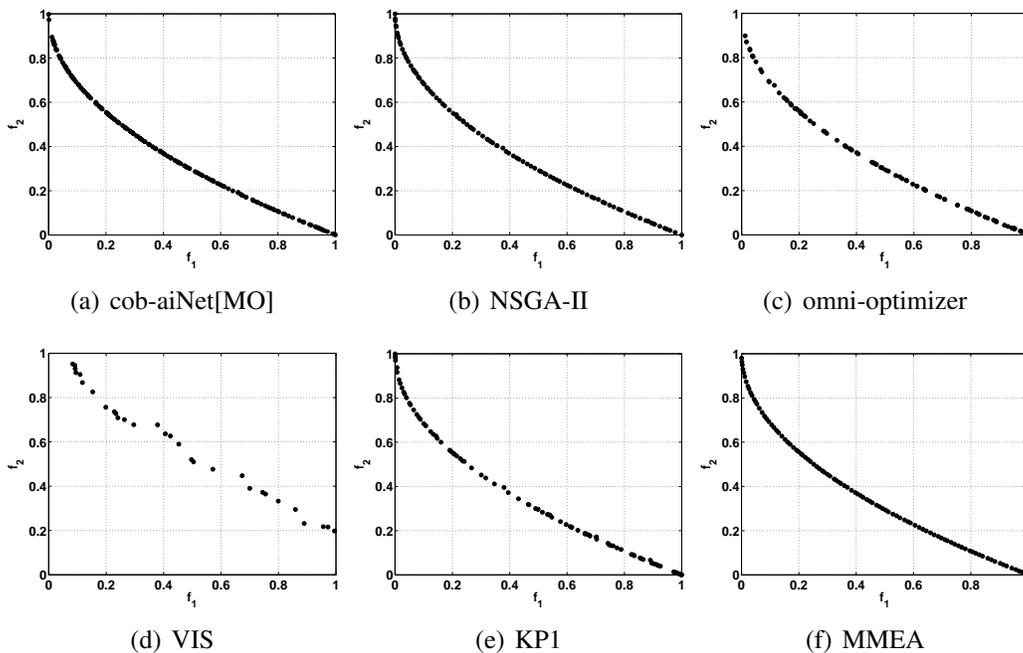


Fig. 6.11: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema ZDT1. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

Tab. 6.5: Média  $\pm$  Desvio Padrão dos valores, em porcentagem, da métrica de *Cobertura entre Dois Conjuntos*  $\mathcal{C}(A, B)$  para os problemas ZDT1–4. Os resultados em negrito correspondem às situações em que um algoritmo  $A$  cobre um maior número de soluções de um algoritmo  $B$  do que  $B$  cobre de  $A$ . Os resultados para os quais a hipótese nula do teste de Wilcoxon pode ser rejeitada com significância 0,05 estão marcados com \*.

ZDT1							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	<b>17 <math>\pm</math> 2*</b>	2 $\pm$ 1*	0 $\pm$ 0*	<b>10 <math>\pm</math> 2</b>	1 $\pm$ 1*
	NSGA-II	2 $\pm$ 2*	–	1 $\pm$ 1*	0 $\pm$ 0*	2 $\pm$ 2*	1 $\pm$ 1*
	omni-optimizer	<b>23 <math>\pm</math> 6*</b>	<b>28 <math>\pm</math> 5*</b>	–	0 $\pm$ 0*	<b>18 <math>\pm</math> 4*</b>	7 $\pm$ 4
	VIS	100 $\pm$ 0*	100 $\pm$ 0*	<b>90 <math>\pm</math> 7*</b>	–	<b>100 <math>\pm</math> 1*</b>	<b>100 <math>\pm</math> 1*</b>
	KP1	9 $\pm$ 3	<b>12 <math>\pm</math> 4*</b>	4 $\pm$ 2*	0 $\pm$ 0*	–	6 $\pm$ 3*
	MMEA	<b>27 <math>\pm</math> 10*</b>	<b>34 <math>\pm</math> 7*</b>	9 $\pm$ 5	0 $\pm$ 0*	<b>22 <math>\pm</math> 6*</b>	–
ZDT2							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	<b>7 <math>\pm</math> 3*</b>	0 $\pm$ 0*	0 $\pm$ 0*	2 $\pm$ 1*	0 $\pm$ 0*
	NSGA-II	1 $\pm$ 1*	–	1 $\pm$ 1*	0 $\pm$ 0*	1 $\pm$ 1*	0 $\pm$ 1*
	omni-optimizer	<b>20 <math>\pm</math> 6*</b>	<b>23 <math>\pm</math> 5*</b>	–	0 $\pm$ 0*	<b>13 <math>\pm</math> 5*</b>	2 $\pm$ 2*
	VIS	100 $\pm$ 0*	100 $\pm$ 0*	<b>75 <math>\pm</math> 11*</b>	–	<b>98 <math>\pm</math> 5*</b>	<b>84 <math>\pm</math> 32*</b>
	KP1	<b>13 <math>\pm</math> 4*</b>	<b>17 <math>\pm</math> 5*</b>	4 $\pm$ 3*	0 $\pm$ 0*	–	2 $\pm$ 2*
	MMEA	<b>66 <math>\pm</math> 20*</b>	<b>73 <math>\pm</math> 18*</b>	<b>26 <math>\pm</math> 17*</b>	0 $\pm$ 0*	<b>38 <math>\pm</math> 21*</b>	–
ZDT3							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	<b>11 <math>\pm</math> 2*</b>	0 $\pm$ 0*	0 $\pm$ 0*	7 $\pm$ 2*	0 $\pm$ 0*
	NSGA-II	2 $\pm$ 2*	–	1 $\pm$ 1*	0 $\pm$ 0*	2 $\pm$ 1*	1 $\pm$ 1*
	omni-optimizer	<b>44 <math>\pm</math> 7*</b>	<b>36 <math>\pm</math> 6*</b>	–	0 $\pm$ 0*	<b>28 <math>\pm</math> 6*</b>	4 $\pm$ 3*
	VIS	100 $\pm$ 2*	100 $\pm$ 0*	<b>87 <math>\pm</math> 6*</b>	–	<b>100 <math>\pm</math> 0*</b>	<b>100 <math>\pm</math> 0*</b>
	KP1	<b>14 <math>\pm</math> 5*</b>	<b>16 <math>\pm</math> 5*</b>	4 $\pm$ 2*	0 $\pm$ 0*	–	2 $\pm$ 2*
	MMEA	<b>75 <math>\pm</math> 12*</b>	<b>67 <math>\pm</math> 13*</b>	<b>32 <math>\pm</math> 15*</b>	0 $\pm$ 0*	<b>49 <math>\pm</math> 14*</b>	–
ZDT4							
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	<b>18 <math>\pm</math> 10*</b>	<b>10 <math>\pm</math> 7</b>	0 $\pm$ 0*	<b>14 <math>\pm</math> 9*</b>	0 $\pm$ 0*
	NSGA-II	2 $\pm$ 1*	–	0 $\pm$ 0*	0 $\pm$ 0*	<b>3 <math>\pm</math> 3</b>	0 $\pm$ 0*
	omni-optimizer	7 $\pm$ 5	<b>41 <math>\pm</math> 12*</b>	–	0 $\pm$ 0*	<b>31 <math>\pm</math> 8*</b>	0 $\pm$ 0*
	VIS	100 $\pm$ 0*	100 $\pm$ 0*	<b>97 <math>\pm</math> 2*</b>	–	<b>100 <math>\pm</math> 0*</b>	<b>79 <math>\pm</math> 35*</b>
	KP1	2 $\pm$ 1*	2 $\pm$ 2	0 $\pm$ 1*	0 $\pm$ 0*	–	0 $\pm$ 0*
	MMEA	100 $\pm$ 0*	100 $\pm$ 0*	<b>61 <math>\pm</math> 21*</b>	16 $\pm$ 26*	<b>100 <math>\pm</math> 0*</b>	–

Tab. 6.6: Média  $\pm$  Desvio Padrão dos valores, em porcentagem, da métrica de *Cobertura entre Dois Conjuntos*  $\mathcal{C}(A, B)$  para o problema *ZDT6*. Os resultados em negrito correspondem às situações em que um algoritmo  $A$  cobre um maior número de soluções de um algoritmo  $B$  do que  $B$  cobre de  $A$ . Os resultados para os quais a hipótese nula do teste de Wilcoxon pode ser rejeitada com significância 0,05 estão marcados com \*.

		ZDT6					
		Algoritmo A					
		cob-aiNet[MO]	NSGA-II	omni-optimizer	VIS	KP1	MMEA
Algoritmo B	cob-aiNet[MO]	–	$4 \pm 2^*$	$0 \pm 1^*$	$0 \pm 0^*$	$0 \pm 0^*$	$0 \pm 0^*$
	NSGA-II	<b><math>71 \pm 11^*</math></b>	–	$0 \pm 0^*$	$0 \pm 0^*$	$0 \pm 0^*$	<b><math>5 \pm 19^*</math></b>
	omni-optimizer	<b><math>42 \pm 9^*</math></b>	<b><math>4 \pm 2^*</math></b>	–	$0 \pm 0^*$	$1 \pm 1^*$	$1 \pm 1^*$
	VIS	<b><math>100 \pm 2^*</math></b>	<b><math>100 \pm 0^*</math></b>	<b><math>88 \pm 7^*</math></b>	–	<b><math>96 \pm 6^*</math></b>	<b><math>95 \pm 6^*</math></b>
	KP1	<b><math>87 \pm 4^*</math></b>	$5 \pm 1^*$	<b><math>95 \pm 3^*</math></b>	$0 \pm 0^*$	–	<b><math>98 \pm 1^*</math></b>
	MMEA	<b><math>46 \pm 8^*</math></b>	$4 \pm 1^*$	<b><math>49 \pm 13^*</math></b>	$0 \pm 0^*$	$0 \pm 0^*$	–

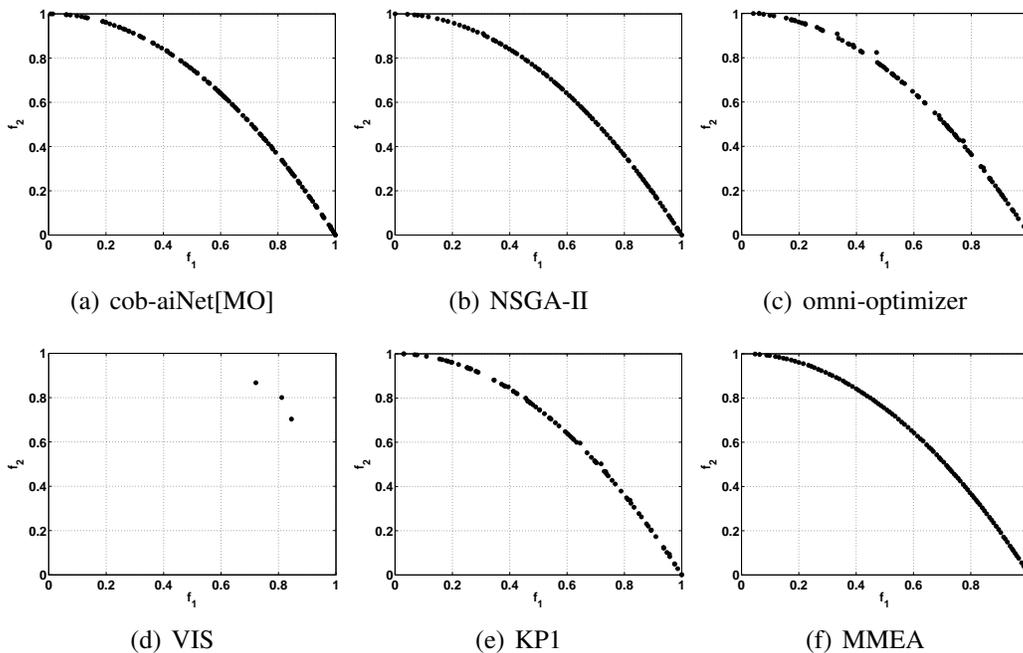


Fig. 6.12: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *ZDT2*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

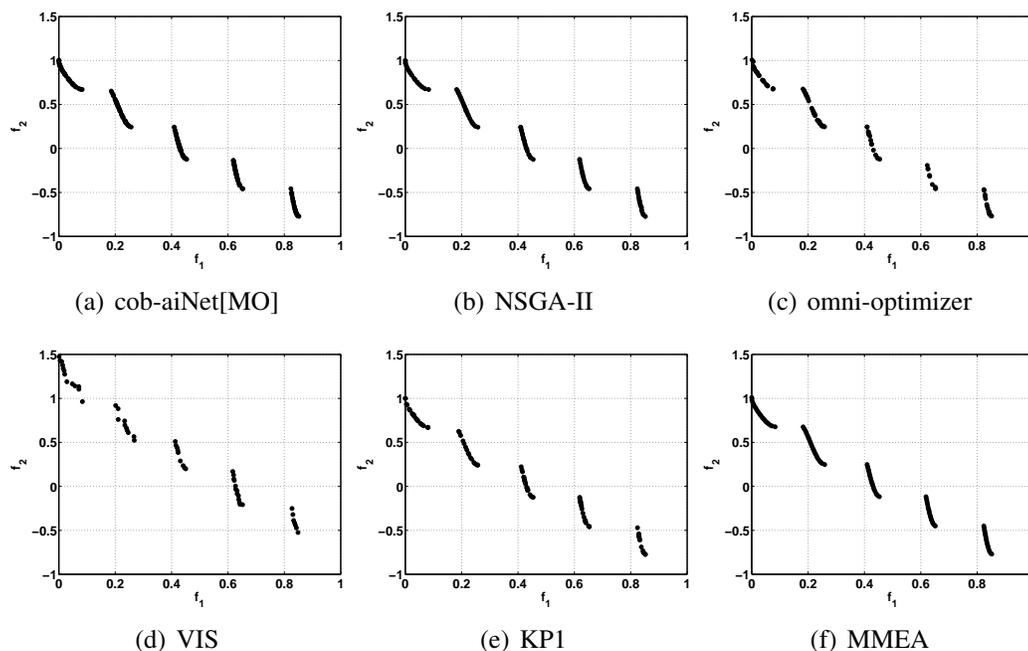


Fig. 6.13: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *ZDT3*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

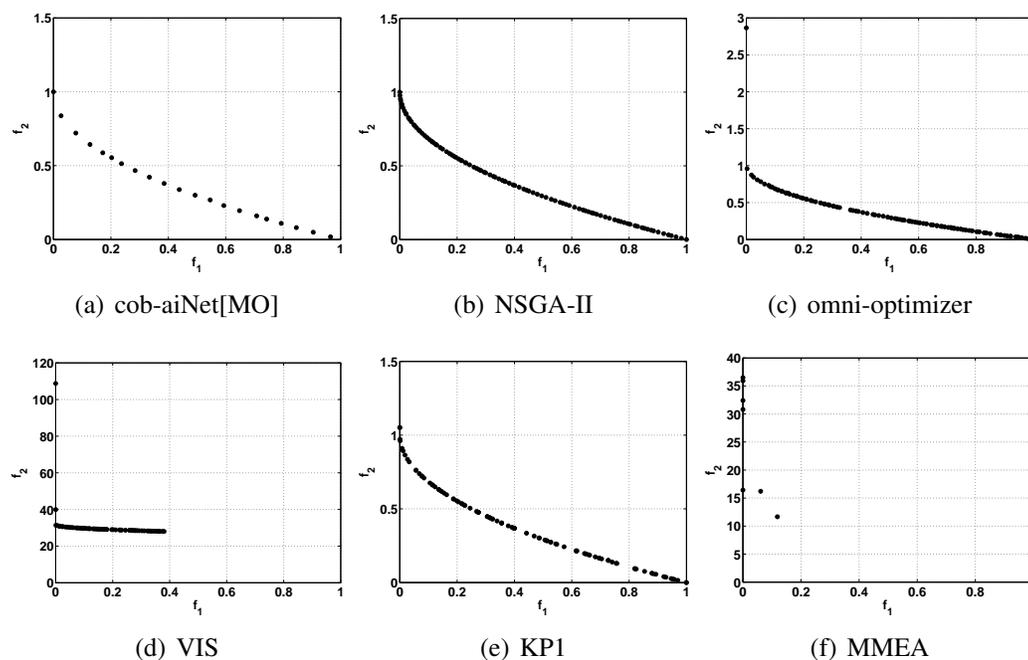


Fig. 6.14: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *ZDT4*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume. Note que a escala dos eixos verticais nas figuras (c), (d) e (f) é diferente das demais figuras.

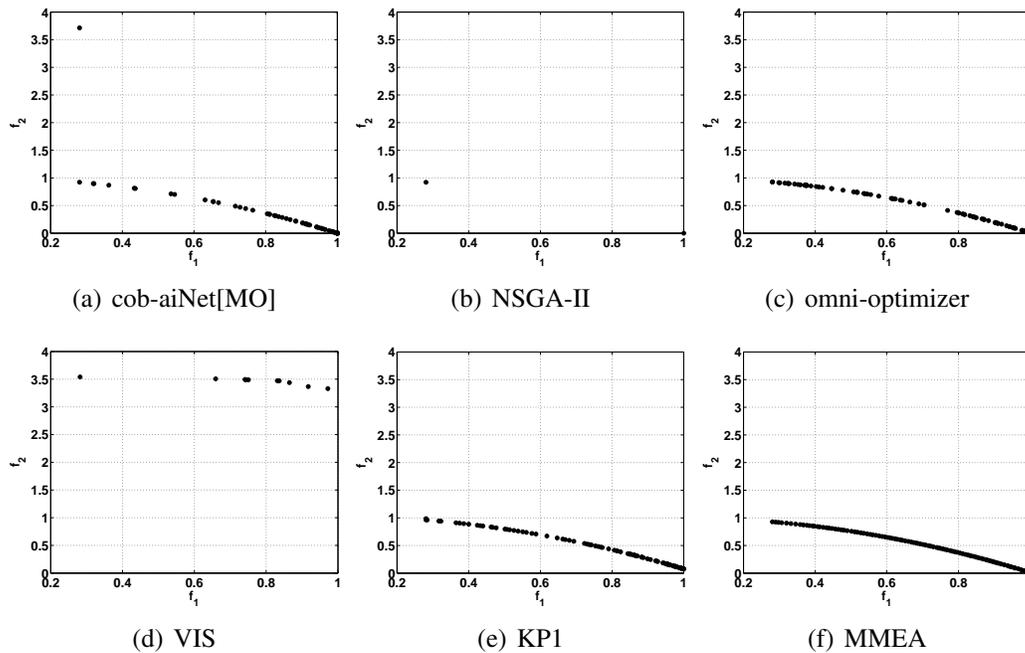


Fig. 6.15: Representação gráfica, no espaço de objetivos, do conjunto de soluções não dominadas retornado por cada algoritmo para o problema *ZDT6*. Tais soluções foram obtidas na repetição em que cada algoritmo atingiu o maior valor para a métrica de hipervolume.

Observando-se as Tabelas 6.5, 6.6, 6.7 e 6.8, pode-se perceber que o algoritmo NSGA-II foi a ferramenta que obteve o conjunto de soluções não dominadas que mais se aproximou da fronteira de Pareto dos problemas *ZDT1*, *ZDT2*, *ZDT3* e *ZDT4*, uma vez que tanto o hipervolume médio quanto as coberturas médias obtidos por este algoritmo foram maiores que os obtidos pelos demais. No entanto, para o problema *ZDT6*, que é um dos mais desafiadores deste conjunto de testes, o NSGA-II não foi capaz de apresentar uma boa cobertura da fronteira de Pareto, uma vez que, como pode ser visto na Figura 6.15(b), os indivíduos presentes em sua população acabaram convergindo para poucos pontos na fronteira de Pareto. Isto acabou resultando em um menor valor médio de hipervolume para este problema e também para um valor nulo de espaçamento, que não deve ser interpretado como um resultado positivo.

O algoritmo *cob-aiNet[MO]*, por sua vez, foi o que apresentou as melhores coberturas médias para o problema *ZDT6*, além de um espalhamento médio significativamente maior que o obtido pelas demais ferramentas. No entanto, como é possível notar nas Tabelas 6.7 e 6.8, e no exemplo dado na Figura 6.15(a), o espaçamento de suas soluções tende a ser pior que o das soluções obtidas pelos demais algoritmos. A principal característica do problema *ZDT6* é fazer com que um conjunto de soluções uniformemente espalhadas no conjunto de Pareto tenha uma distribuição não uniforme na fronteira de Pareto. Sendo assim, como a *cob-aiNet[MO]* possui uma série de mecanismos que esti-

Tab. 6.7: Média  $\pm$  Desvio Padrão das métricas de *hipervolume*, *espaçamento* e *espalhamento máximo*. Os melhores resultados estão em negrito. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados aos da cob-aiNet) pode ser rejeitada com significância 0,05 estão marcados com \*.

Hypervolume (maior é melhor)					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KPI	MMEA
ZDT1	0,660 $\pm$ 0,000	<b>0,661 <math>\pm</math> 0,000*</b>	0,651 $\pm$ 0,003*	0,655 $\pm$ 0,001*	0,658 $\pm$ 0,001*
ZDT2	0,326 $\pm$ 0,000	<b>0,328 <math>\pm</math> 0,000*</b>	0,317 $\pm$ 0,006*	0,322 $\pm$ 0,001*	0,257 $\pm$ 0,131*
ZDT3	1,041 $\pm$ 0,000	<b>1,041 <math>\pm</math> 0,000*</b>	1,024 $\pm$ 0,014*	1,036 $\pm$ 0,010*	1,029 $\pm$ 0,008*
ZDT4	0,634 $\pm$ 0,004	<b>0,661 <math>\pm</math> 0,000*</b>	0,652 $\pm$ 0,002*	0,655 $\pm$ 0,001*	0,000 $\pm$ 0,000*
ZDT6	0,299 $\pm$ 0,008	0,056 $\pm$ 0,000*	0,308 $\pm$ 0,003*	0,273 $\pm$ 0,003*	<b>0,310 <math>\pm</math> 0,003*</b>
Espaçamento (menor é melhor)					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KPI	MMEA
ZDT1	0,006 $\pm$ 0,001	0,008 $\pm$ 0,001*	0,011 $\pm$ 0,002*	0,011 $\pm$ 0,001*	<b>0,003 <math>\pm</math> 0,000*</b>
ZDT2	0,008 $\pm$ 0,001	0,007 $\pm$ 0,000	0,012 $\pm$ 0,003*	0,011 $\pm$ 0,002*	<b>0,002 <math>\pm</math> 0,001*</b>
ZDT3	0,005 $\pm$ 0,001	0,008 $\pm$ 0,001*	0,013 $\pm$ 0,003*	0,013 $\pm$ 0,003*	<b>0,004 <math>\pm</math> 0,000*</b>
ZDT4	0,045 $\pm$ 0,011	<b>0,007 <math>\pm</math> 0,001*</b>	0,480 $\pm$ 0,905	0,453 $\pm$ 0,796	2,314 $\pm$ 1,454*
ZDT6	0,149 $\pm$ 0,141	<b>0,000 <math>\pm</math> 0,000*</b>	0,076 $\pm$ 0,105*	0,012 $\pm$ 0,001*	0,003 $\pm$ 0,000*
Espalhamento Máximo (maior é melhor)					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KPI	MMEA
ZDT1	<b>1,418 <math>\pm</math> 0,003</b>	1,414 $\pm$ 0,000*	1,321 $\pm$ 0,068*	1,414 $\pm$ 0,000*	1,416 $\pm$ 0,001*
ZDT2	1,414 $\pm$ 0,000	1,414 $\pm$ 0,000*	1,313 $\pm$ 0,088*	<b>1,415 <math>\pm</math> 0,002*</b>	1,126 $\pm$ 0,575*
ZDT3	1,971 $\pm$ 0,002	1,967 $\pm$ 0,000*	1,849 $\pm$ 0,115*	1,955 $\pm$ 0,075*	<b>1,976 <math>\pm</math> 0,010*</b>
ZDT4	1,410 $\pm$ 0,012	1,414 $\pm$ 0,000*	6,212 $\pm$ 9,524	6,570 $\pm$ 9,006*	<b>23,626 <math>\pm</math> 6,622*</b>
ZDT6	<b>3,651 <math>\pm</math> 1,063</b>	1,168 $\pm$ 0,000*	1,911 $\pm$ 1,176*	1,156 $\pm$ 0,006*	1,175 $\pm$ 0,006*

mulam a obtenção de soluções com a distribuição mais uniforme possível no espaço de variáveis, no caso específico deste problema tais mecanismos acabam levando a uma distribuição não uniforme no espaço de objetivos (e a um alto valor de espaçamento), por mais que a função de *fitness* adotada pelo algoritmo também estimule a busca por uma boa distribuição das soluções na fronteira de indivíduos não dominados.

Para os problemas *ZDT2* e *ZDT3*, a cob-aiNet[MO] apresentou coberturas médias inferiores apenas às obtidas pelo NSGA-II, ficando em segundo lugar também no critério de hipervolume, para os dois problemas, e espaçamento para o problema *ZDT3*. Para o problema *ZDT2*, a cob-aiNet[MO] obteve o terceiro melhor espaçamento mas, como é possível ver nas figuras 6.12 e 6.13, de maneira geral a distribuição de suas soluções é visualmente muito semelhante à obtida pelos algoritmos NSGA-II e MMEA, que são os melhores neste critério para estes dois problemas.

Com relação ao problema *ZDT1*, a cob-aiNet[MO] apresentou coberturas médias inferiores tanto ao NSGA-II quanto ao algoritmo KPI mas, no critério de hipervolume, seus resultados são inferiores

Tab. 6.8: Ranks de cada algoritmo (exceto VIS), frente aos demais, obtidos a partir das métricas de *hipervolume*, *espaçamento* e *espalhamento máximo* para os problemas ZDT. Na última linha é apresentado o *rank* médio obtido por cada ferramenta.

Hiper-volume					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KP1	MMEA
ZDT1	2	1	5	4	3
ZDT2	2	1	4	3	5
ZDT3	2	1	5	3	4
ZDT4	4	1	3	2	5
ZDT6	3	5	2	4	1
Espaçamento					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KP1	MMEA
ZDT1	2	3	5	4	1
ZDT2	3	2	5	4	1
ZDT3	2	3	5	4	1
ZDT4	2	1	4	3	5
ZDT6	5	1	4	3	2
Espalhamento Máximo					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KP1	MMEA
ZDT1	1	3	5	4	2
ZDT2	3	2	4	1	5
ZDT3	2	3	5	4	1
ZDT4	5	4	3	2	1
ZDT6	1	4	2	5	3
Rank Médio ± Desvio Padrão					
	cob-aiNet[MO]	NSGA-II	omni-optimizer	KP1	MMEA
	2,60 ± 1,24	2,33 ± 1,35	4,07 ± 1,10	3,33 ± 1,05	2,67 ± 1,72

apenas aos obtidos pelo NSGA-II. Com relação ao espaçamento, a cob-aiNet[MO] ficou atrás apenas do MMEA, enquanto que seu espalhamento foi o melhor dentre os algoritmos considerados.

Por fim, para o problema *ZDT4* a cob-aiNet[MO] encontrou mais dificuldades de convergência, ficando atrás de NSGA-II, KP1 e omni-optimizer nos critérios de cobertura e hipervolume, mas atrás apenas do NSGA-II em espaçamento. Com relação ao espalhamento, tanto a cob-aiNet[MO] quanto o NSGA-II apresentaram valores muito inferiores aos obtidos pelos demais algoritmos, mas é importantes destacar que: (i) o algoritmo MMEA não conseguiu convergir para este problema, como pode ser visto na Tabela 6.5 e na Figura 6.14(f); e (ii) apesar da maioria das soluções obtidas pelos algoritmos omni-optimizer e KP1 terem convergido para a fronteira de Pareto, algumas delas, próximas a  $f_1(\vec{x}) = 0$ , permaneceram distantes da fronteira (vide figuras 6.14(c) e 6.14(e)), o que aumenta sig-

nificativamente o espalhamento máximo. Observando-se a Figura 6.14(a) e a Tabela 6.9, que contém o número médio final de soluções não dominadas retornado por cada algoritmo, pode-se perceber também que a *cob-aiNet[MO]* convergiu para um número menor de soluções para o problema *ZDT4*, quando comparada aos demais algoritmos.

Tab. 6.9: Média  $\pm$  Desvio Padrão do número de soluções não dominadas retornado por cada algoritmo para os problemas ZDT. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados aos da *cob-aiNet*) pode ser rejeitada com significância 0,05 estão marcados com \*.

	<i>cob-aiNet[MO]</i>	NSGA-II	omni-optimizer	KP1	MMEA
<b>ZDT1</b>	162 $\pm$ 6	100 $\pm$ 0*	100 $\pm$ 0*	100 $\pm$ 0*	100 $\pm$ 0*
<b>ZDT2</b>	114 $\pm$ 8	100 $\pm$ 0*	100 $\pm$ 1*	100 $\pm$ 0*	81 $\pm$ 38*
<b>ZDT3</b>	192 $\pm$ 3	100 $\pm$ 0*	100 $\pm$ 0*	100 $\pm$ 0*	100 $\pm$ 0*
<b>ZDT4</b>	20 $\pm$ 1	100 $\pm$ 0*	94 $\pm$ 5*	100 $\pm$ 0*	8 $\pm$ 2*
<b>ZDT6</b>	68 $\pm$ 7	100 $\pm$ 0*	93 $\pm$ 5*	100 $\pm$ 0*	100 $\pm$ 0*

Sobre o algoritmo MMEA, apesar da sua aparente boa convergência observada nas figuras 6.11–6.15 (exceto para o problema *ZDT4*), faltou a esta ferramenta uma capacidade maior de ajuste fino de suas soluções, uma vez que, pelos resultados apresentados nas tabelas 6.5 e 6.6, pode-se perceber que suas soluções foram, em sua grande maioria, dominadas pelas soluções obtidas pelos demais algoritmos. Com relação à cobertura, o MMEA só foi superior ao algoritmo KP1 (desconsiderando-se o VIS) no problema *ZDT6*, problema este em que a maioria das ferramentas discutidas aqui encontrou dificuldades.

O algoritmo KP1, por sua vez, foi a melhor ferramenta para o problema *ZDT4*, que apresenta uma grande quantidade de ótimos locais, e ficou em segundo lugar, atrás apenas do NSGA-II, para o problema *ZDT1*, quando avaliado pela métrica de cobertura. Por outro lado, também sob este critério, foi a pior ferramenta para o problema *ZDT6*. Em relação às demais métricas utilizadas aqui, o algoritmo KP1 não teve grandes destaques, apresentando apenas o melhor espalhamento máximo para o problema *ZDT2*.

Por fim, quanto ao algoritmo *omni-optimizer*, pode-se dizer que ele apresentou um comportamento mediano, conseguindo convergir razoavelmente para a fronteira de Pareto na maioria dos casos sob o critério de cobertura, como pode ser visto na Tabela 6.5, mas não apresentando nenhum destaque positivo em relação às demais métricas. Na verdade, o *omni-optimizer* apresentou até mesmo alguns destaques negativos, já que obteve os piores valores médios para hipervolume, espaçamento e espalhamento para os problemas *ZDT1* e *ZDT3*, e também o pior espaçamento para o problema *ZDT2*.

Sendo assim, de maneira geral, para os problemas tratados aqui pode-se concluir que a *cob-aiNet[MO]* se mostrou uma das ferramentas mais competitivas dentre os algoritmos comparados. Apesar de não ter sido o melhor algoritmo em todos os problemas segundo nenhum dos critérios (nenhum algoritmo foi), a ferramenta proposta neste trabalho só apresentou o pior desempenho na métrica de espaçamento para o problema *ZDT6*, problema este em que, por outro lado, a *cob-aiNet[MO]* obteve a melhor cobertura frente aos demais algoritmos. Além disso, outro aspecto que deve ser destacado é o fato da *cob-aiNet[MO]* ter efetivamente convergido em *todos* os problemas estudados, diferentemente de outros algoritmos, como o NSGA-II, que, apesar de ter apresentado os melhores desempenhos para a maioria dos problemas, também teve sérias dificuldades de convergência para o *ZDT6*. Ou seja, em outras palavras, pode-se dizer que a *cob-aiNet[MO]* foi a ferramenta mais *robusta* diante da variação de características dos problemas estudados nesta seção, o que é um aspecto altamente positivo.

Um último ponto que ainda precisa ser analisado nesta seção diz respeito ao custo computacional exigido pela ferramenta proposta aqui. Na Tabela 6.10, é possível observar o tempo médio exigido pelos algoritmos *cob-aiNet[MO]* e NSGA-II (o algoritmo mais rápido empregado aqui) para os problemas *ZDT1* a *ZDT6*, avaliados nas 25 repetições feitas para cada um deles. Os tempos computacionais gastos pelos demais algoritmos foram omitidos da Tabela 6.10 uma vez que os experimentos com tais ferramentas não foram feitos em uma mesma plataforma, além de algumas destas ferramentas terem sido implementadas em linguagens diferentes das demais. Nos resultados apresentados na Tabela 6.10, tanto o NSGA-II quanto a *cob-aiNet[MO]* foram implementados em Visual C# .NET, e os experimentos foram feitos em um computador com processador Intel Core2Duo T7500 (2, 2 Ghz), 3 GB de RAM, Windows 7 Professional e Microsoft .NET Framework 4.

Tab. 6.10: Média  $\pm$  Desvio Padrão do tempo computacional (em segundos) exigido pelos algoritmos *cob-aiNet[MO]* e NSGA-II para os problemas *ZDT1*–*ZDT6*. Os resultados para os quais a hipótese nula do teste de Wilcoxon (quando comparados aos da *cob-aiNet[MO]*) pode ser rejeitada com significância 0,05 estão marcados com \*.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
<b>cob-aiNet[MO]</b>	33,37 $\pm$ 1,68	25,72 $\pm$ 3,53	59,30 $\pm$ 17,74	62,53 $\pm$ 19,97	72,61 $\pm$ 19,55
<b>NSGA-II</b>	11,50 $\pm$ 3,93*	12,06 $\pm$ 4,73*	13,83 $\pm$ 8,80*	13,86 $\pm$ 6,46*	11,78 $\pm$ 4,60*

Como pode-se observar nos resultados apresentados na Tabela 6.10, os tempos computacionais exigidos pela *cob-aiNet[MO]* são, no mínimo, cerca de duas vezes maiores que os do NSGA-II, o que era esperado. Os algoritmos baseados na Teoria da Rede Imunológica (Jerne, 1974) geralmente tendem a apresentar custos computacionais mais altos quando comparados a outros algoritmos evolutivos, uma vez que adotam mecanismos que frequentemente requerem comparações par a par entre

todas as soluções em suas populações. Nos algoritmos cob-aiNet e cob-aiNet[MO], este aspecto é ainda mais evidente, já que o modelo de concentração e os operadores de seleção/inserção (vide seções 3.3.2 e 5.3.2) empregados exigem que tais comparações sejam feitas com uma frequência ainda maior. No entanto, da mesma maneira que para o algoritmo cob-aiNet, este custo computacional extra apresentado pela cob-aiNet[MO] pode ser visto como o preço a ser pago pela maior capacidade de manutenção de diversidade observada nos resultados discutidos na Seção 6.2. Além disso, este maior custo computacional é independente do custo de avaliação das funções objetivo, o que pode levar a uma aproximação do custo exigido pela cob-aiNet[MO] ao de outras ferramentas evolutivas em problemas reais, nos quais a avaliação das funções objetivo tendem a ser as operações mais custosas em um processo de otimização.

## 6.4 Resultados para o Problema DTLZ2

Nesta terceira e última parte dos experimentos de avaliação do algoritmo cob-aiNet[MO], a ferramenta proposta foi aplicada no problema de teste conhecido na literatura como *DTLZ2* (Deb et al., 2002b), ajustado para otimização (minimização) de três objetivos. Como o intuito deste experimento é apenas avaliar se o algoritmo cob-aiNet[MO] também é capaz de tratar adequadamente problemas multiobjetivo com mais de dois critérios, as comparações que serão feitas aqui tomaram como base apenas o algoritmo NSGA-II, que apresentou a maior capacidade de convergência para a fronteira de Pareto na maioria dos problemas ZDT discutidos na Seção 6.3. Além disso, como muitas das métricas de avaliação discutidas na Seção 6.1.3 não se aplicam ou requerem alguma forma de refinamento para que possam ser utilizadas em problemas com mais de dois objetivos, as comparações desta seção foram feitas apenas com a métrica de cobertura entre dois conjuntos, que permite avaliar a capacidade relativa, entre os dois algoritmos, de aproximação da fronteira de Pareto.

Como nos experimentos feitos nas Seções 6.2 e 6.3, os algoritmos cob-aiNet[MO] e NSGA-II foram executados 25 vezes para o problema DTLZ2, e os resultados médios obtidos para a métrica de cobertura foram avaliados. As ferramentas cob-aiNet[MO] e NSGA-II foram executados aqui com os seguintes parâmetros:

- **cob-aiNet[MO]:** número de indivíduos na população inicial  $n_{AB} = 100$ ; número mínimo e máximo de clones  $n_{C^{min}} = 2$  e  $n_{C^{max}} = 10$ , respectivamente; concentração inicial  $C_0 = 0,5$ ; tamanho máximo da população  $max_{AB} = 250$ ;  $\beta^i = 0,1$  e  $\beta^f = 0,0001$ ; limiar de supressão  $\sigma_s = 0,03$ ; número de iterações de busca local  $LS_{it} = 5$ ; número de iterações entre etapas de busca local  $LS_{freq} = 5$ ; e intervalo de busca inicial da busca local  $SR$  igual à metade do domínio de cada variável. O parâmetro  $\sigma_s$  foi novamente ajustado aqui buscando-se um número final de soluções o mais próximo possível do retornado pelo NSGA-II.

- **NSGA-II:** tamanho da população igual a 100 indivíduos; probabilidade de mutação igual a  $1/n$ , onde  $n$  é a dimensão do problema; probabilidade de *crossover* igual a 1,0; e índices de distribuição de *crossover* e de mutação (Deb et al., 2002a) iguais a 20.

Os resultados médios e os respectivos desvios padrão obtidos pela *cob-aiNet[MO]* e pelo NSGA-II para a métrica de cobertura estão apresentados na Tabela 6.11. Como é possível perceber, o NSGA-II foi, em média, capaz de cobrir um número maior de soluções obtidas pela *cob-aiNet[MO]* do que o oposto, o que indica uma melhor aproximação da fronteira de Pareto. No entanto, a porcentagem de soluções cobertas pelo NSGA-II não chegou a 10%, o que indica que ambos os algoritmos apresentam uma boa capacidade de aproximação da fronteira.

Tab. 6.11: Média  $\pm$  Desvio Padrão dos valores, em porcentagem, da métrica de *Cobertura entre Dois Conjuntos*  $\mathcal{C}(A, B)$  para o problema *DTLZ2*. O resultado em negrito mostra que o NSGA-II cobriu um número maior de soluções da *cob-aiNet[MO]*, enquanto que o \* indica que os resultados permitem a rejeição da hipótese nula do teste de Wilcoxon com significância 0,05, ou seja, suas distribuições são estatisticamente diferentes.

DTLZ2			
		Algoritmo A	
		cob-aiNet[MO]	NSGA-II
Algoritmo B	cob-aiNet[MO]	–	<b>9 <math>\pm</math> 4*</b>
	NSGA-II	4 $\pm$ 2*	–

Com relação à distribuição destas soluções sobre a fronteira, na Figura 6.16 são apresentados os conjuntos finais de indivíduos não dominados obtidos tanto pela *cob-aiNet[MO]* quanto pelo NSGA-II, nas repetições inicial, intermediária e final dos experimentos. Como é possível perceber na Figura 6.16, apesar do algoritmo NSGA-II ter sido capaz de retornar, em média, um número maior de soluções não dominadas ( $100 \pm 0$ ) que a *cob-aiNet[MO]* ( $62 \pm 5$ ), a ferramenta imunoinspirada claramente foi capaz de distribuir mais uniformemente suas soluções sobre a fronteira. Pode-se observar também que a única região do espaço de objetivos em que a *cob-aiNet[MO]* manteve um certo acúmulo de soluções foi em torno de  $f_3(\vec{x}) = 1$ , que coincidentemente também corresponde à região onde se concentra a maior parte das soluções dominadas pelos indivíduos retornados pelo NSGA-II.

De maneira geral, pode-se então concluir que a *cob-aiNet[MO]* apresentou um comportamento similar ao observado para o NSGA-II para este problema, sendo até mesmo capaz de obter uma melhor distribuição das soluções no espaço de objetivos. Isto indica que a ferramenta proposta aqui tem um comportamento compatível com o das demais meta-heurísticas populacionais da literatura, também em problemas com três objetivos. No entanto, para casos em que o número de objetivos a serem otimizados é muito alto, espera-se que também ocorra a degeneração de desempenho observada

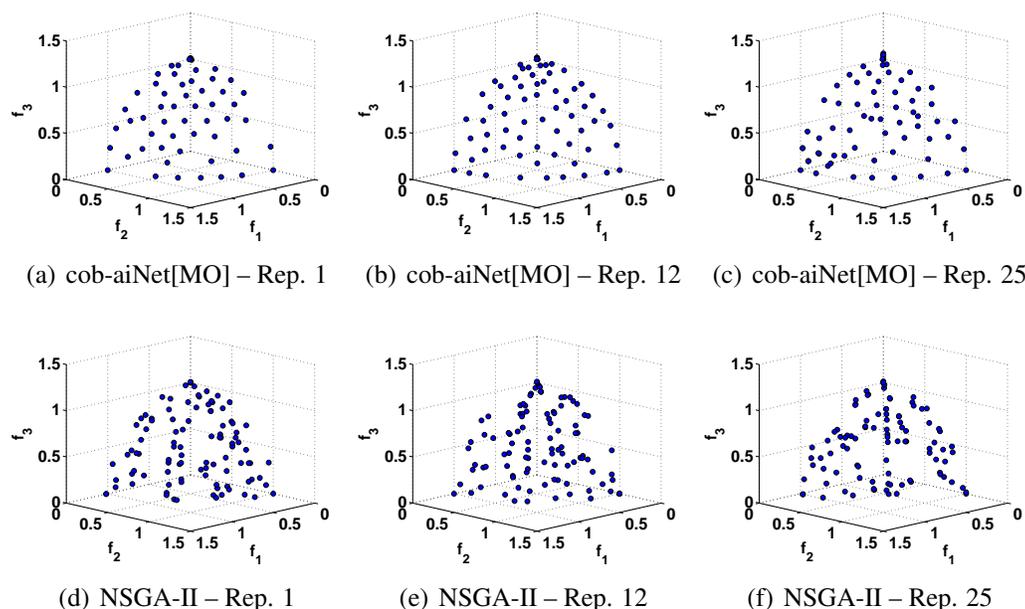


Fig. 6.16: Representação gráfica, no espaço de objetivos, dos conjuntos de soluções não dominadas retornados pelos algoritmos cob-aiNet[MO] e NSGA-II, para o problema *DTLZ2*, nas repetições 1, 12 e 25 dos experimentos.

nas demais ferramentas da literatura, causada pela perda de pressão evolutiva sobre os indivíduos da população (Sato et al., 2007).

## 6.5 Considerações Finais

Neste capítulo, foi feita uma série de experimentos com o objetivo de avaliar o desempenho geral do algoritmo cob-aiNet[MO], tanto em relação à sua capacidade de manutenção de diversidade no espaço de variáveis quanto em relação à sua capacidade de obter conjuntos de soluções que se aproximem adequadamente da fronteira de Pareto de um dado problema e que apresentem uma boa cobertura desta fronteira, ou seja, que sejam diversas também no espaço de objetivos do problema. Para isso foram utilizados dois conjuntos de problemas de teste, sendo o primeiro voltado a avaliar principalmente a capacidade de manutenção de diversidade do algoritmo no espaço de variáveis, considerando-se também sempre a qualidade da fronteira obtida, e o segundo voltado para avaliar o desempenho geral da ferramenta em problemas de maior dimensão, frequentemente utilizados na literatura de otimização objetivo (no caso, os problemas ZDT – Zitzler et al., 2000). Os resultados obtidos foram avaliados em relação às métricas de hipervolume (*hypervolume* ou *S-metric*), espaçamento (*spacing*), espalhamento máximo (*maximum spread*), cobertura entre dois conjuntos (*two-set coverage*) e diversidade baseada em hipercubos (*hypercube-based diversity*), e as comparações fo-

ram feitas com os algoritmos NSGA-II (Deb et al., 2002a), omni-optimizer (Deb & Tiwari, 2005), VIS (Freschi & Repetto, 2006), KP1 (Chan & Ray, 2005) e MMEA (Zhou et al., 2009). Além disso, para avaliar se a ferramenta proposta aqui é capaz de tratar adequadamente problemas com mais de dois objetivos, foram feitos também experimentos com o problema DTLZ2 (Deb et al., 2002b), sendo os resultados obtidos pela *cob-aiNet[MO]* comparados aos obtidos pelo NSGA-II.

Diante dos resultados obtidos para o primeiro conjunto de problemas de teste, pôde-se observar que a *cob-aiNet[MO]* não só possui mecanismos eficazes de manutenção de diversidade no espaço de variáveis, já que apresentou os melhores resultados para a métrica de diversidade baseada em hiper-cubos em três dos quatro problemas estudados (e ficou em segundo lugar para o quarto problema), mas também é capaz de aproximar adequadamente a fronteira de Pareto, uma vez que sempre esteve entre os algoritmos com as melhores coberturas entre dois conjuntos para todos os problemas. Além disso, a *cob-aiNet[MO]* também possui bons mecanismos de manutenção de diversidade no espaço de objetivos dos problemas, uma vez que apresentou os melhores resultados médios de espalhamento para três dos quatro problemas tratados aqui.

Já pelos resultados obtidos para o conjunto de problemas *ZDT*, observou-se que a *cob-aiNet[MO]* foi uma das ferramentas mais competitivas dentre as estudadas, já que, apesar de não ter sido o melhor algoritmo em *todos* os problemas segundo nenhum dos critérios (nenhum algoritmo foi), a *cob-aiNet[MO]* só apresentou o pior desempenho na métrica de espaçamento para o problema *ZDT6*, problema este em que, por outro lado, a ferramenta proposta aqui obteve a melhor cobertura frente aos demais algoritmos. Além disso, outro aspecto que deve ser destacado é o fato da *cob-aiNet[MO]* ter efetivamente convergido para *todos* os problemas estudados, diferentemente de outros algoritmos que apresentaram os melhores desempenhos em alguns problemas e tiveram sérios problemas de convergência em outros.

Por fim, os resultados obtidos com o problema DTLZ2 mostraram que a *cob-aiNet[MO]* possui um comportamento compatível com o de outras meta-heurísticas populacionais da literatura também em problemas com três objetivos, uma vez que a ferramenta proposta aqui apresentou uma cobertura entre dois conjuntos próxima à obtida pelo NSGA-II (apesar de inferior) e uma capacidade superior de espalhamento das soluções sobre a fronteira de Pareto.

Sendo assim, é possível concluir que, como a *cob-aiNet*, a *cob-aiNet[MO]* é uma ferramenta competitiva frente às demais alternativas da literatura, tendo como principal vantagem possuir uma excelente capacidade de manutenção de diversidade no espaço de variáveis, superior até mesmo à de outros algoritmos propostos para otimização multiobjetivo com este mesmo propósito, além de também apresentar uma boa capacidade de convergência e cobertura da fronteira de Pareto. Por outro lado, a *cob-aiNet[MO]* também exige uma quantidade maior de recursos computacionais quando comparada a outras ferramentas, o que é inerente à inspiração na teoria da Rede Imunológica em que

se baseia e deve ser considerado pelo usuário na ponderação entre necessidade de manutenção de diversidade no espaço de variáveis e recursos computacionais disponíveis para resolução de um dado problema.



# Capítulo 7

## Comentários Finais

De maneira geral, o objetivo principal desta tese foi explorar modelos alternativos de redes imunológicas (Bersini, 2003), presentes na literatura de sistemas imunológicos artificiais (SIAs – de Castro & Timmis, 2002b), para desenvolver novos algoritmos imunoinspirados voltados para otimização em espaços contínuos. A principal característica almejada nestas novas ferramentas é a presença de mecanismos que as levem a apresentar uma melhor capacidade de manutenção de diversidade de soluções no espaço de busca.

É conhecido que muitos dos algoritmos imunoinspirados já existentes e voltados para otimização, principalmente aqueles baseados na teoria da Rede Imunológica (Jerne, 1974), são capazes de manter uma boa diversidade de soluções durante o processo de busca (de França et al., 2010a,b). No entanto, geralmente tais ferramentas empregam mecanismos e operadores baseados em modelos extremamente simplificados do sistema imunológico natural (SIN). Por outro lado, dentro da própria literatura de SIAs existe uma série de trabalhos que buscam compreender melhor o funcionamento do sistema imunológico natural dos vertebrados, trabalhos estes que empregam modelos computacionais significativamente mais elaborados e refinados que os atualmente adotados nas ferramentas imunoinspiradas para otimização. Neste trabalho buscou-se aproximar estas duas vertentes, através da incorporação de mecanismos presentes em alguns destes modelos em novas ferramentas de otimização.

No Capítulo 2, foram apresentados os aspectos gerais do sistema imunológico natural (SIN) e os princípios de Seleção Clonal (Burnet, 1978) e Rede Imunológica (Jerne, 1974), que são duas das principais teorias que visam explicar o funcionamento do SIN, além de também serem amplamente utilizadas como fonte de inspiração para os sistemas imunológicos artificiais (SIAs). Em seguida, os SIAs propriamente ditos foram introduzidos, através da identificação de algumas características presentes no SIN que são computacionalmente relevantes, e de uma breve revisão bibliográfica de aplicações existentes na literatura desta área. Seguiu-se uma discussão sobre os conceitos fundamen-

tais em SIAs e o detalhamento do algoritmo opt-aiNet (*Artificial Immune Network for Continuous Optimization* – de Castro & Timmis, 2002a), utilizado para exemplificar como geralmente se dá a transposição dos conceitos presentes na imunologia para um contexto de engenharia. O algoritmo opt-aiNet foi escolhido por se tratar de um dos SIAs inspirados na teoria da Rede Imunológica mais empregados da literatura e por também permitir ao leitor observar o quão simplificado é o modelo de rede imunológica geralmente adotado na maioria das ferramentas imunoinspiradas voltadas para otimização. Por fim, foram discutidos alguns outros exemplos de modelos mais elaborados de redes imunológicas presentes na literatura de SIAs, para que fosse possível identificar quais os aspectos mais relevantes presentes em cada um deles e quais possuem mecanismos que podem ser aproveitados no desenvolvimento de novos algoritmos.

Feito isso, no Capítulo 3 foi proposto um novo modelo de rede imunológica, baseado no trabalho de Bersini (2002), que possui uma série de características que o tornam diretamente aplicável ao problema de otimização em espaços contínuos. A partir deste modelo, foi desenvolvido o primeiro algoritmo proposto nesta tese, a cob-aiNet (*Concentration-based Artificial Immune Network*). A estrutura geral do algoritmo cob-aiNet e o funcionamento de cada um de seus operadores foram extensivamente discutidos no Capítulo 3, onde também foi feito o posicionamento desta nova ferramenta frente tanto a outras meta-heurísticas populacionais da literatura quanto a algoritmos exatos de otimização.

O algoritmo cob-aiNet foi extensivamente avaliado no Capítulo 4, onde foi feita uma série de experimentos visando avaliar o desempenho geral da ferramenta, tanto em relação à sua capacidade de localização do ótimo global de cada problema quanto à sua capacidade de manter a diversidade das soluções na população. No Capítulo 4, a cob-aiNet foi comparada a uma série de outras abordagens de otimização presentes na literatura, tanto baseadas em SIAs quanto em outros paradigmas, e avaliou-se também o impacto que variações em cada um de seus parâmetros provocam no desempenho geral da ferramenta. Todos os experimentos realizados no Capítulo 4 foram feitos com o conjunto de funções de teste proposto para a Competição de Otimização de Parâmetros Reais da *2005 CEC Special Session on Real-Parameter Optimization* (Suganthan et al., 2005) e seguindo-se a mesma metodologia, o que permitiu a comparação direta da cob-aiNet com os algoritmos participantes da competição.

Quando comparada com outros SIAs da literatura (de Castro & Timmis, 2002a; de França et al., 2005), também baseados na teoria da Rede Imunológica, os resultados obtidos nos experimentos do Capítulo 4 mostraram que a cob-aiNet possui mecanismos superiores de manutenção de diversidade, já que em todos os problemas ela foi capaz de localizar uma maior quantidade de ótimos locais, além de também ter sido capaz de se adaptar mais adequadamente às características de cada problema estudado. Nos problemas monomodais, a cob-aiNet sempre convergiu para uma única solução, enquanto que nos multimodais o número de indivíduos em sua população aumentou ou diminuiu conforme a

quantidade de ótimos locais existentes. Além disso, um outro aspecto em que a cob-aiNet se mostrou superior foi quanto ao erro absoluto médio obtido entre o melhor indivíduo encontrado durante a busca e o ótimo global do problema, já que em todos os casos o algoritmo proposto aqui foi capaz de atingir a precisão desejada.

Quando comparada aos algoritmos participantes da competição CEC'2005 e aos dois SIAs em problemas de maior dimensão, foi observado que a cob-aiNet se mostrou competitiva frente aos demais algoritmos, e novamente superou as duas ferramentas baseadas em SIAs. Mesmo não apresentando o melhor *rank* final médio quando avaliada em todos os problemas de teste e comparada a todos os algoritmos, os resultados mostraram que a cob-aiNet apresentou o melhor desempenho em vários dos problemas tratados e que, em nenhum deles, ela obteve o pior desempenho.

Por fim, na análise de sensibilidade do desempenho do algoritmo a diferentes ajustes de seus parâmetros, também feita no Capítulo 4, foi possível identificar quais são os parâmetros que têm o maior impacto no desempenho geral da ferramenta quando incorretamente ajustados, e propor sugestões sobre como cada um deles deve ser configurado. Dentre os parâmetros estudados, ficou evidente que o limiar de supressão  $\sigma_s$  é o que tem maior impacto no desempenho da ferramenta, além de ser o parâmetro de mais difícil ajuste.

O fato da cob-aiNet ter se mostrado competitiva frente a algoritmos diversos, tanto na localização da melhor solução global para um problema de otimização, quanto na possibilidade de localização de múltiplos ótimos locais quando tal característica se faz necessária, acabou motivando a extensão deste algoritmo para tratar problemas de otimização *multiobjetivo* em espaços contínuos, o que deu origem ao segundo algoritmo proposto nesta tese, a cob-aiNet[MO] (*Concentration-based Artificial Immune Network for Multiobjective Optimization*).

O algoritmo cob-aiNet[MO] foi introduzido e detalhado no Capítulo 5, onde também foi apresentada uma visão geral dos principais conceitos associados à otimização multiobjetivo e feito o posicionamento desta nova ferramenta frente a outras propostas da literatura. Além de possuir uma série de adaptações nos mecanismos introduzidos pela cob-aiNet para tratar problemas de otimização multiobjetivo, a cob-aiNet[MO] possui uma diferença conceitual sutil em relação à ferramenta que lhe deu origem: enquanto a cob-aiNet busca localizar não só o(s) ótimo(s) global(is) do problema mas também o maior número possível de ótimos locais, a cob-aiNet[MO] busca explorar seus mecanismos de manutenção de diversidade no espaço de busca para encontrar não só a fronteira de Pareto do problema, mas o maior número possível de soluções pertencentes ao seu conjunto de Pareto. Ou seja, enquanto a versão mono-objetivo do algoritmo pode intencionalmente retornar para o usuário um conjunto de soluções sub-ótimas, sua variante multiobjetivo busca retornar o mais completo conjunto de soluções que aproximam a fronteira de Pareto do problema. Este aspecto de manutenção de diversidade no espaço de variáveis em problemas de otimização multiobjetivo ainda foi pouco explo-

rado na literatura da área, sendo que apenas algumas poucas ferramentas foram propostas com este fim.

No Capítulo 6, foram discutidos os resultados obtidos em uma série de experimentos que visaram avaliar tanto a capacidade do algoritmo cob-aiNet[MO] de obter conjuntos de soluções que aproximem adequadamente a fronteira de Pareto de um dado problema e levem a uma boa cobertura desta fronteira, quanto sua capacidade de obtenção de conjuntos de Pareto com soluções diversas, ou seja, sua capacidade de manutenção de diversidade no espaço de variáveis do problema. Para isso, foram utilizados dois conjuntos de problemas de teste, sendo o primeiro constituído de problemas com conjuntos de Pareto conhecidos e bem definidos, o que permitiu principalmente a avaliação dos mecanismos de manutenção de diversidade do algoritmo, e o segundo o conjunto de problemas conhecido na literatura como ZDT (Zitzler et al., 2000). Os resultados obtidos foram avaliados sob diversos aspectos, através de uma série de métricas consolidadas na literatura, e comparações foram feitas com os principais algoritmos de otimização multiobjetivo voltados para manutenção de diversidade no espaço de variáveis (Chan & Ray, 2005; Deb & Tiwari, 2005; Zhou et al., 2009), além dos algoritmos NSGA-II (Deb et al., 2002a) e VIS (Freschi & Repetto, 2006). Um terceiro aspecto que também se buscou avaliar no Capítulo 6 foi se o algoritmo cob-aiNet[MO] é capaz de tratar adequadamente problemas com mais de dois objetivos, sendo que, para isso, foram feitos experimentos com o problema de teste conhecido na literatura como DTLZ2 (Deb et al., 2002b), e os resultados obtidos pela cob-aiNet[MO] comparados aos obtidos pelo NSGA-II.

Os resultados obtidos com o primeiro conjunto de problemas permitiram concluir que o algoritmo cob-aiNet[MO] não só possui mecanismos eficazes de manutenção de diversidade no espaço de variáveis como também é capaz de aproximar adequadamente a fronteira de Pareto de cada problema, além de apresentar um bom espalhamento das soluções sobre a fronteira encontrada, ou seja, além de manter a diversidade também no espaço de objetivos. Já diante dos resultados obtidos com os problemas ZDT, é possível afirmar que, apesar da cob-aiNet[MO] não ter sido o melhor algoritmo em *todos* os problemas segundo nenhum dos critérios adotados (nenhum algoritmo estudado no Capítulo 6 foi) ela efetivamente convergiu para *todos* os problemas estudados, diferentemente de outros algoritmos que apresentaram os melhores desempenhos para alguns dos problemas e tiveram sérios problemas de convergência para outros. De maneira geral, foi possível concluir que a cob-aiNet[MO] apresenta uma boa capacidade de convergência e cobertura da fronteira de Pareto, além de possuir também uma excelente capacidade de manutenção de diversidade no espaço de variáveis, superior até mesmo à de outros algoritmos propostos para otimização multiobjetivo com este mesmo propósito. Por fim, os resultados obtidos para o problema DTLZ2 mostraram que a cob-aiNet[MO] possui um comportamento compatível com o de outras meta-heurísticas populacionais da literatura também em problemas com mais de dois objetivos, uma vez que a ferramenta proposta aqui apresentou uma co-

bertura entre dois conjuntos próxima à obtida pelo NSGA-II (apesar de inferior) e uma capacidade superior de espalhamento das soluções sobre a fronteira de Pareto.

Diante dos resultados obtidos ao longo desta tese, é possível concluir que a utilização de operadores mais avançados e também de um modelo mais elaborado de rede imunológica, em algoritmos imunoinspirados voltados para otimização, levou ao desenvolvimento de ferramentas com uma capacidade significativamente melhor de localizar e manter soluções diversas em sua população ao longo do processo evolutivo, quando comparadas a outras abordagens da literatura, e que também foram capazes de localizar o(s) ótimo(s) global(is) da maioria dos problemas estudados. No entanto, apesar dos bons resultados obtidos aqui, também foram identificados alguns aspectos passíveis de serem melhorados, que serão discutidos na próxima seção.

## 7.1 Discussão

O primeiro aspecto negativo, identificado ao longo dos experimentos, foi que tanto a cob-aiNet quanto a cob-aiNet[MO] tendem a exigir uma quantidade maior de recursos computacionais, quando comparadas às demais ferramentas. Esta característica não era totalmente inesperada, uma vez que os dois algoritmos são inspirados na teoria da Rede Imunológica e, por isso, requerem comparações par a par entre todos os indivíduos na população. Vários outros algoritmos inspirados neste mesmo paradigma também realizam tais comparações mas, no caso específico dos algoritmos propostos aqui, este aspecto é acentuado, já que o novo modelo de rede imunológica exige que tais comparações sejam feitas sempre que alguma célula da população sofrer modificações e as concentrações de cada indivíduo tiverem que ser atualizadas. No entanto, ficou claro nos resultados experimentais que o novo modelo de rede imunológica e os novos operadores introduzidos na cob-aiNet e cob-aiNet[MO] se refletiram em uma melhor capacidade de manutenção de diversidade no espaço de variáveis, cabendo então ao usuário ponderar entre a sua real necessidade de manutenção de diversidade e os recursos computacionais disponíveis.

Uma possível maneira de aliviar a carga computacional dos algoritmos propostos aqui seria através da divisão de suas populações em sub-populações menores, que evoluiriam isoladamente ou com base em etapas restritas de comunicação. Com isso, periodicamente ocorreriam comparações entre as várias sub-populações, que poderiam levar a supressões ou até mesmo a redistribuições dos indivíduos. Uma abordagem semelhante a esta já foi adotada nos algoritmos MOM-aiNet (Coelho et al., 2008) e MOM-aiNet+ (Coelho et al., 2009a), propostos para biclusterização. A vantagem de se adotar este conceito de sub-populações que evoluem independentemente está na possibilidade de utilização de paralelismo, através da alocação de cada sub-população a um dado núcleo de processamento. No entanto, as comparações entre sub-populações ainda teriam que ser feitas em um núcleo

central, de forma que só haveria ganhos caso o custo de transferência de indivíduos entre os núcleos de processamento fosse significativamente inferior ao custo de evolução dos indivíduos.

Outra desvantagem apresentada pelos dois algoritmos propostos nesta tese está relacionada ao número de parâmetros que devem ser ajustados pelo usuário. Apesar de muitos destes parâmetros serem intuitivos em relação ao impacto que têm sobre o desempenho das ferramentas, alguns deles exigem que o usuário disponha de algum grau de conhecimento dos respectivos operadores a que estão associados, para que sejam ajustados corretamente. Dentre estes parâmetros, o limiar de supressão  $\sigma_s$ , que determina o raio ao redor de cada indivíduo dentro do qual ocorrerá o reconhecimento mútuo entre células, é sem dúvida o parâmetro de mais difícil ajuste, uma vez que depende também das características do problema sendo tratado. Uma possível alternativa ao ajuste manual destes parâmetros seria a adoção de alguma técnica automatizada para realizar esta tarefa (Smit & Eiben, 2009), mas tal opção geralmente envolve um alto custo computacional. Uma outra abordagem seria reformular os operadores do algoritmo de forma a introduzir heurísticas que ajustem dinamicamente alguns destes parâmetros, mas, além desta opção também levar a um maior custo computacional, nem sempre a elaboração de tais heurísticas é uma tarefa trivial.

Um outro aspecto que ainda poderia ser explorado seria alguma forma de hibridização das duas ferramentas propostas nesta tese com mecanismos capazes de combinar características de algum grupo de soluções da população na elaboração de novos indivíduos. Em paradigmas como o de algoritmos genéticos e o dos algoritmos de estimação de densidade, os novos indivíduos da população são criados a partir de vários indivíduos previamente existentes, ou seja, os novos indivíduos são criados a partir da *combinação* de características presentes em *múltiplos indivíduos*. Já no paradigma de SIAs, não há esta troca de informação entre as células, e cada indivíduo evolui independentemente, através de um processo acelerado de hipermutações sucessivas.

Esta característica dos SIAs geralmente faz com que os algoritmos baseados nesta abordagem apresentem uma convergência mais lenta e encontrem dificuldades em fazer ajustes finos em seus indivíduos. Diante disso, os SIAs puros, voltados para otimização, geralmente acabam apresentando desempenho insatisfatório em relação à aproximação do ótimo global, principalmente em problemas de maior dimensão. Nas duas ferramentas propostas nesta tese, esta desvantagem acabou compensada pela utilização do operador de busca local, que, além de aumentar a velocidade de convergência dos algoritmos, também lhes conferiu uma melhor capacidade de realizar ajustes finos em seus indivíduos. Outras abordagens da literatura geralmente fazem hibridizações com operadores propostos dentro de outros paradigmas, como no caso do algoritmo EMOIA (Tan et al., 2008), proposto para otimização multiobjetivo, que emprega um operador de *crossover* além da tradicional mutação.

Apesar da combinação dos operadores de mutação e busca local terem compensado as deficiências nos problemas tratados aqui, eles ainda não realizam a combinação de informações provenientes de

múltiplos indivíduos e também acabam não preservando eventuais *blocos construtivos*<sup>12</sup> que possam vir a surgir em um dado indivíduo. Com a introdução de operadores capazes não só de identificar e manter blocos construtivos mas também de combinar blocos construtivos presentes em indivíduos diferentes na geração de novas soluções, o desempenho das ferramentas propostas aqui tenderia a melhorar consideravelmente (caso os problemas tratados efetivamente possuam blocos construtivos). Na literatura, pode-se citar como exemplos de hibridizações entre SIAs e mecanismos de detecção e preservação de blocos construtivos o algoritmo GAIS (Castro & Von Zuben, 2010a), proposto para otimização mono-objetivo, e o algoritmo MOGAIS (Castro & Von Zuben, 2010b), que corresponde à extensão do GAIS para otimização multiobjetivo.

Como é possível perceber, mesmo diante dos bons resultados reportados ao longo desta tese ainda existe uma série de aspectos a serem explorados que podem levar a melhorias significativas no desempenho das ferramentas propostas aqui, tanto em relação ao tempo de execução quanto à qualidade das soluções obtidas, mesmo que para isso sejam introduzidos novos mecanismos que acabem distanciando estas ferramentas de sua inspiração biológica.

## 7.2 Perspectivas Futuras

Além de todos os pontos levantados na seção anterior, existem também diversos outros aspectos e possibilidades de trabalho em aberto. Ao longo desta tese, apesar de terem sido tratados tanto problemas de otimização mono-objetivo quanto problemas multiobjetivo, o foco foi concentrado apenas em problemas irrestritos com variáveis contínuas, que representam apenas uma fração dos problemas de otimização existentes. Sendo assim, uma das perspectivas de trabalho a ser seguida daqui pra frente consiste na adaptação das duas ferramentas propostas aqui para tratar problemas de otimização *discreta e combinatória*. Para isso, precisarão ser desenvolvidos não só novos operadores de variação genética (mutação e busca local) mas também deverá ser definida uma nova maneira de identificação de *vizinhança* entre soluções, já que o conceito de distância euclidiana não será mais aplicável.

Outra linha que poderá ser seguida diz respeito ao tratamento de *problemas com restrições*. Neste trabalho, foram considerados apenas problemas com restrições de canalização para cada variável, mas existe todo um universo, dentro da área de otimização, voltado para resolver problemas que possuem outros tipos de restrições, como as de igualdade e desigualdade. Nestes problemas com restrições, uma das dificuldades está justamente na localização de conjuntos de soluções factíveis,

---

<sup>12</sup>Diversos problemas de otimização podem ser decompostos em sub-problemas passíveis de serem resolvidos individualmente. Após a resolução de cada um destes sub-problemas, as soluções parciais devem ser então combinadas, para que se possa obter a solução global do problema original. Neste contexto, é possível que o *vetor de atributos* associado a cada solução candidata em um algoritmo populacional possa vir a apresentar *trechos* que correspondem a estas soluções parciais. Caso isto ocorra, cada um destes *trechos* é denominado *bloco construtivo* (do inglês, *building block* – Castro, 2009).

o que exige uma boa capacidade de exploração do espaço de variáveis por parte dos algoritmos. Como as ferramentas propostas aqui possuem mecanismos eficazes de manutenção de diversidade no espaço de variáveis, acredita-se que elas possam vir a ter bons desempenhos também em problemas com restrições, mas estudos aprofundados devem ser feitos para que isto seja devidamente verificado.

Os mecanismos de manutenção de diversidade das ferramentas cob-aiNet e cob-aiNet[MO] podem ainda ser explorados em problemas de otimização *dinâmica*, ou seja, problemas cujas características variam ao longo do processo de otimização. Por manterem um conjunto de soluções bem distribuídas pelo espaço de variáveis, teoricamente os algoritmos cob-aiNet e cob-aiNet[MO] têm condições de se adaptarem mais rapidamente a cada modificação do problema, já que as chances de alguma das soluções presentes na população estar em uma região do espaço de busca que se tornou promissora são maiores. No entanto, sabe-se que os algoritmos propostos aqui têm como desvantagem o custo computacional mais alto que o de outras abordagens da literatura, o que pode impedir que tais ferramentas tenham tempo suficiente para se adaptarem às novas condições do problema antes que ocorra uma nova modificação.

Por fim, uma outra linha de trabalho a ser seguida diz respeito à aplicação das ferramentas propostas aqui em problemas práticos. Apesar dos problemas de teste serem bastante genéricos e desafiadores, e mesmo considerando que os resultados obtidos correspondem a bons indicativos sobre o comportamento dos algoritmos, sua utilização em problemas reais é fundamental para que se possa efetivamente avaliar o seu desempenho, aplicabilidade e escalabilidade. Sendo assim, a aplicação dos algoritmos cob-aiNet e cob-aiNet[MO] a problemas reais consiste em uma das perspectivas futuras mais importantes, dentre aquelas a serem seguidas de agora em diante.

Diante de tudo o que foi discutido aqui, é possível concluir que o trabalho realizado nesta tese pode ser considerado apenas o ponto de partida no desenvolvimento dos algoritmos cob-aiNet e cob-aiNet[MO], uma vez que ainda existem alguns aspectos passíveis de serem melhorados e uma grande quantidade de caminhos a serem explorados, que certamente contribuirão não só para a evolução e o amadurecimento destas ferramentas, mas também para o surgimento de novas variações voltadas para outros tipos de problemas.

# Apêndice A

## Problemas de Teste

Neste apêndice, serão apresentados os problemas de teste utilizados para a avaliação dos dois algoritmos propostos nesta tese. Na Seção A.1, serão discutidos os problemas voltados para otimização *mono-objetivo* em espaços contínuos, empregados no Capítulo 4 para avaliação do algoritmo cob-aiNet. Já na Seção A.2, serão detalhados os problemas voltados para otimização *multiobjetivo*, utilizados nos estudos feitos no Capítulo 6 junto à segunda ferramenta proposta nesta tese: o algoritmo cob-aiNet[MO].

### A.1 Problemas de Teste para Otimização Mono-objetivo

O conjunto de problemas de teste utilizado para avaliação do algoritmo cob-aiNet foi proposto originalmente para a Competição de Otimização de Parâmetros Reais da *2005 CEC Special Session on Real-Parameter Optimization*, ocorrida no *IEEE Congress on Evolutionary Computation (CEC)* de 2005 (Suganthan et al., 2005). De maneira geral, este conjunto de problemas de teste é composto de 25 problemas mono-objetivo contínuos, que englobam uma ampla gama de características distintas, permitindo assim uma avaliação ampla daqueles algoritmos que os utilizam como casos de estudo. Além disso, outra característica relevante destes problemas é o fato deles serem *escalonáveis*, ou seja, é possível utilizá-los tanto em pequenas quanto em altas dimensões, o que dá ao usuário uma maior flexibilidade ao elaborar experimentos baseados nestes problemas.

Nesta seção, serão discutidos os aspectos gerais de cada um dos 25 problemas deste conjunto de testes, mas sugere-se que o leitor consulte Suganthan et al. (2005) para obter os detalhes específicos de implementação computacional de cada um deles. Todos os arquivos de dados com os vetores de deslocamento, ótimos globais e matrizes de transformação linear podem ser obtidos no seguinte endereço (último acesso em 29 de abril de 2011):

[http://www3.ntu.edu.sg/home/EPNSugan/index\\_files/CEC-05/CEC05.htm](http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm)

## A.1.1 Problemas Monomodais

### A.1.1.1 Problema $F_1$ : Função Esfera Deslocada

O problema  $F_1$ , dado na Equação A.1, é um problema monomodal, com variáveis separáveis, baseado no deslocamento da função esfera tradicional.

$$F_1 = \sum_{i=1}^D z_i^2 + f_1^{bias}, \quad (\text{A.1})$$

onde  $\vec{z} = \vec{x} - \vec{o}$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $D$  é o número de dimensões do problema,  $\vec{x} \in [-100; 100]^D$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_1(x^*) = f_1^{bias} = -450$ . A representação gráfica deste problema no espaço bidimensional é dada na Figura A.1(a).

### A.1.1.2 Problema $F_2$ : Problema 1.2 de Schwefel Deslocado

O problema  $F_2$ , dado na Equação A.2, é um problema monomodal, com variáveis não separáveis, baseado no deslocamento do problema 1.2 de Schwefel.

$$F_2(\vec{x}) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_2^{bias}, \quad (\text{A.2})$$

onde  $\vec{z} = \vec{x} - \vec{o}$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $D$  é o número de dimensões do problema,  $\vec{x} \in [-100; 100]^D$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_2(x^*) = f_2^{bias} = -450$ . A representação gráfica deste problema no espaço bidimensional é dada na Figura A.1(b).

### A.1.1.3 Problema $F_3$ : Função Elíptica Altamente Condicionada, Deslocada e Rotacionada

O problema  $F_3$ , dado na Equação A.3, é um problema monomodal, com variáveis não separáveis, baseado em uma função elíptica deslocada e rotacionada.

$$F_3(\vec{x}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} z_i^2 + f_3^{bias}, \quad (\text{A.3})$$

onde  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz ortogonal,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $D$  é o número de dimensões do problema,  $\vec{x} \in [-100; 100]^D$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_3(x^*) = f_3^{bias} = -450$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(c).

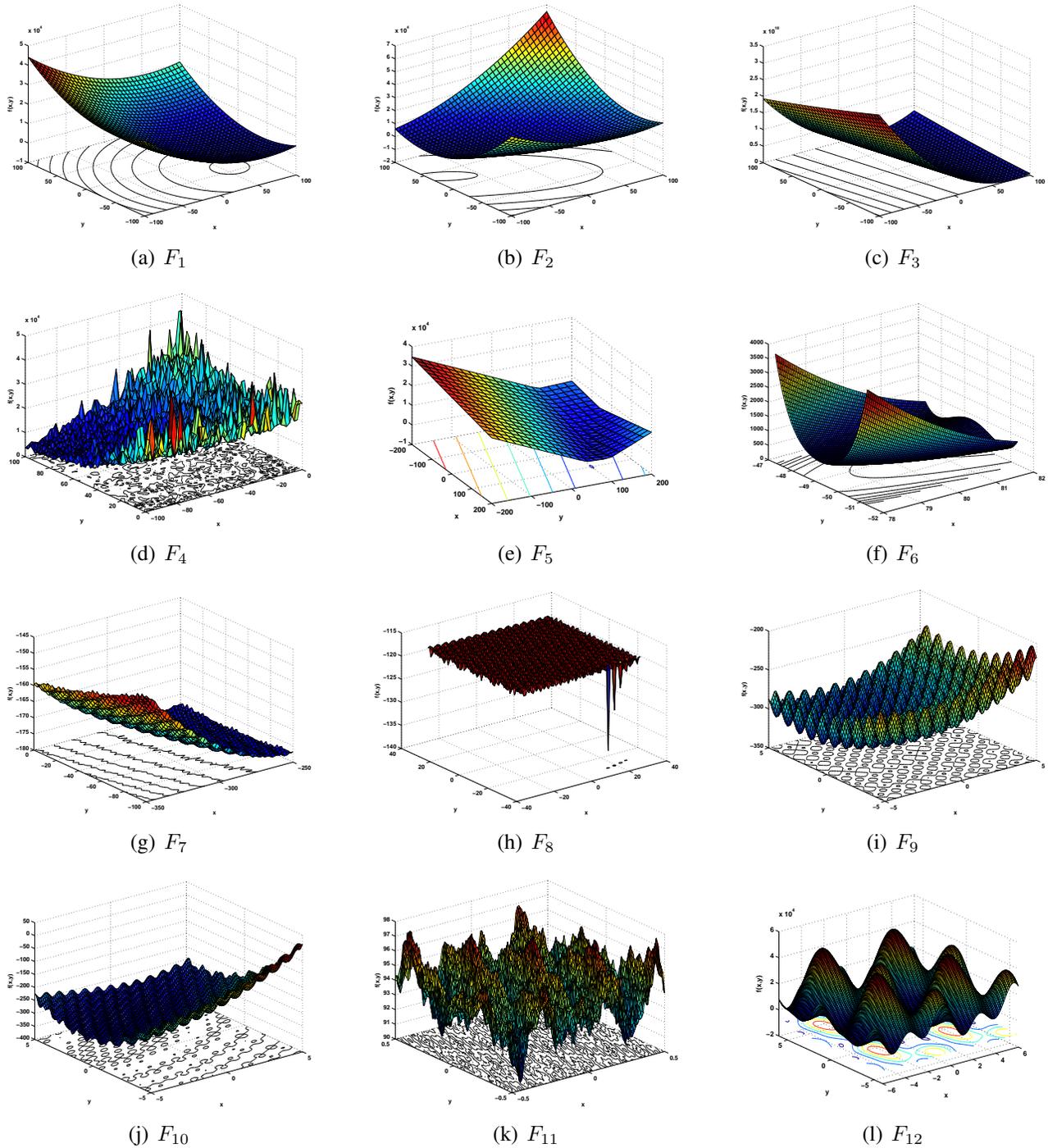


Fig. A.1: Representação gráfica, em  $\mathbb{R}^2$ , das funções de teste  $F_1$  a  $F_{12}$  da competição de Otimização de Parâmetros Reais do IEEE *Congress on Evolutionary Computation* (CEC) de 2005.

#### A.1.1.4 Problema $F_4$ : Problema 1.2 de Schwefel Deslocado com Ruído no *Fitness*

O problema  $F_4$ , dado na Equação A.4, é um problema monomodal, com variáveis não separáveis, baseado no problema 1.2 de Schwefel deslocado e com a presença de ruído no *fitness*.

$$F_4(\vec{x}) = \left[ \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 \right] \cdot (1 + 0,4 \cdot |\mathcal{N}(0,1)|) + f_4^{bias}, \quad (\text{A.4})$$

onde  $\vec{z} = \vec{x} - \vec{o}$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema,  $F_4(x^*) = f_4^{bias} = -450$ ,  $\vec{x} \in [-100; 100]^D$ ,  $D$  é o número de dimensões do problema e  $\mathcal{N}(0,1)$  é um ruído gaussiano de média zero e variância unitária. A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(d).

#### A.1.1.5 Problema $F_5$ : Problema 2.6 de Schwefel com Ótimo Global na Fronteira

O problema  $F_5$ , dado na Equação A.6, é um problema monomodal, com variáveis não separáveis e ótimo global na fronteira, que basicamente consiste na extensão da função dada na Equação A.5 para  $D$  dimensões.

$$f(\vec{x}) = \max \{|x_1 + 2 \cdot x_2 - 7|, |2 \cdot x_1 + x_2 - 5|\}, \quad (\text{A.5})$$

onde  $f(x^*) = 0$  e  $x^* = [1, 3]$ .

$$F_5(\vec{x}) = \max \{|\mathbf{A}_i \cdot \vec{x} - \mathbf{B}_i|\} + f_5^{bias}, \quad (\text{A.6})$$

onde  $i = 1, \dots, D$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-100; 100]^D$ ,  $D$  é o número de dimensões do problema,  $\mathbf{A}$  é uma matriz  $D \times D$  cujos elementos são valores aleatórios em  $[-500; 500]$  e  $\det(\mathbf{A}) \neq 0$ ,  $\mathbf{A}_i$  é a  $i$ -ésima linha de  $\mathbf{A}$ ,  $\mathbf{B}_i = \mathbf{A}_i \cdot \vec{o}$  e  $\vec{o}$  é um vetor de tamanho  $D \times 1$  com elementos  $o_i$  aleatórios no intervalo  $[-100; 100]$ . Após a geração aleatória do vetor  $\vec{o}$ , deve-se ajustá-lo da seguinte forma:  $o_i = -100$  para  $i = 1, \dots, \lceil \frac{D}{4} \rceil$  e  $o_i = 100$  para  $i = \lfloor \frac{3D}{4} \rfloor, \dots, D$ .

O ótimo global deste problema é dado por  $x^* = \vec{o}$  e  $F_5(x^*) = f_5^{bias} = -310$ . Sua representação gráfica pode ser vista na Figura A.1(e).

## A.1.2 Problemas Multimodais Básicos

### A.1.2.1 Problema $F_6$ : Função de Rosenbrock Deslocada

O problema  $F_6$ , dado na Equação A.7, é um problema multimodal, com variáveis não separáveis, formado basicamente pela função de Rosenbrock deslocada. A principal característica deste problema

é a existência de um vale estreito ligando um ótimo local ao ótimo global.

$$F_6(\vec{x}) = \sum_{i=1}^{D-1} [100 \cdot (z_i^2 - z_{i+1})^2 + (z_i - 1)^2] + f_6^{bias}, \quad (\text{A.7})$$

onde  $\vec{z} = \vec{x} - \vec{o} + 1$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $D$  é o número de dimensões do problema,  $\vec{x} \in [-100; 100]^D$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_6(x^*) = f_6^{bias} = 390$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(f).

### A.1.2.2 Problema $F_7$ : Função de Griewank Ilimitada, Rotacionada e Deslocada

O problema  $F_7$ , dado na Equação A.8, é um problema multimodal, com variáveis não separáveis e ilimitadas, formado pela função de Griewank deslocada e rotacionada. Apesar de ser uma função ilimitada, a população inicial deve ser gerada em  $[0; 600]^D$ , onde  $D$  é o número de dimensões do problema, sendo que o ótimo global  $x^* = \vec{o}$  do problema está fora deste intervalo de inicialização.

$$F_7(\vec{x}) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_7^{bias}, \quad (\text{A.8})$$

onde  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz de transformação linear com número de condição 3,  $\vec{x} = [x_1, x_2, \dots, x_D]$  e  $F_7(x^*) = f_7^{bias} = -180$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(g).

### A.1.2.3 Problema $F_8$ : Função de Ackley Rotacionada, Deslocada e com Ótimo na Fronteira

O problema  $F_8$ , dado na Equação A.9, é um problema multimodal, com variáveis não separáveis e ótimo global localizado na fronteira do espaço de busca, formado pela função de Ackley deslocada e rotacionada.

$$F_8(\vec{x}) = -20 \cdot \exp\left(-0,2 \cdot \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2 \cdot \pi \cdot z_i)\right) + 20 + e + f_8^{bias}, \quad (\text{A.9})$$

onde  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz de transformação linear com número de condição 100,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-32; 32]^D$ ,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_8(x^*) = f_8^{bias} = -140$ . Após o carregamento do arquivo com o vetor  $\vec{o}$ , seus elementos devem ser ajustados da seguinte forma:  $o_{2j-1} = -32 \cdot o_{2j}$  para  $j = 1, \dots, \lfloor \frac{D}{2} \rfloor$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(h).

#### A.1.2.4 Problema $F_9$ : Função de Rastrigin Deslocada

O problema  $F_9$ , dado na Equação A.10, é um problema multimodal, com variáveis separáveis, formado basicamente pela função de Rastrigin deslocada. A principal característica deste problema é a existência de uma grande quantidade de ótimos locais.

$$F_9(\vec{x}) = \sum_{i=1}^D [z_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot z_i) + 10] + f_9^{bias}, \quad (\text{A.10})$$

onde  $\vec{z} = \vec{x} - \vec{o}$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-5; 5]^D$ ,  $D$  é o número de dimensões do problema,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_9(x^*) = f_9^{bias} = -330$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(i).

#### A.1.2.5 Problema $F_{10}$ : Função de Rastrigin Deslocada e Rotacionada

O problema  $F_{10}$ , dado na Equação A.11, é basicamente o problema  $F_9$  rotacionado, o que faz com que suas variáveis percam a separabilidade. Como o problema  $F_9$ ,  $F_{10}$  também tem como principal característica a presença de uma grande quantidade de ótimos locais.

$$F_{10}(\vec{x}) = \sum_{i=1}^D [z_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot z_i) + 10] + f_{10}^{bias}, \quad (\text{A.11})$$

onde  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz de transformação linear com número de condição 2,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-5; 5]^D$ ,  $D$  é o número de dimensões do problema,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_{10}(x^*) = f_{10}^{bias} = -330$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(j).

#### A.1.2.6 Problema $F_{11}$ : Função de Weierstrass Deslocada e Rotacionada

O problema  $F_{11}$ , dado na Equação A.12, é um problema multimodal, com variáveis não separáveis, formado basicamente pela função de Weierstrass deslocada e rotacionada. A principal característica deste problema é a diferenciabilidade em apenas alguns pontos do espaço de busca, apesar de se tratar de uma função contínua.

$$F_{11}(\vec{x}) = \sum_{i=1}^D \left\{ \sum_{k=0}^{k_{max}} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot (z_i + 0, 5))] \right\} - D \cdot \sum_{k=0}^{k_{max}} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot 0, 5)] + f_{11}^{bias}, \quad (\text{A.12})$$

onde  $k_{max} = 20$ ,  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz de transformação linear com número de condição 5,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-0, 5; 0, 5]^D$ ,  $D$  é o número de dimensões do problema,  $\vec{o} = x^*$  é

um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_{11}(x^*) = f_{11}^{bias} = 90$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.1(k).

### A.1.2.7 Problema $F_{12}$ : Problema 2.13 de Schwefel

O problema 2.13 de Schwefel ( $F_{12}$ ), dado na Equação A.13, é um problema multimodal cujas variáveis não são separáveis.

$$F_{12}(\vec{x}) = \sum_{i=1}^D [\mathbf{A}_i - \mathbf{B}_i(\vec{x})]^2 + f_{12}^{bias}, \quad (\text{A.13})$$

onde  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-\pi; \pi]^D$ ,  $D$  é o número de dimensões do problema,

$$\mathbf{A}_i = \sum_{j=1}^D [a_{ij} \cdot \sin(\alpha_j) + b_{ij} \cdot \cos(\alpha_j)], \quad (\text{A.14})$$

$$\mathbf{B}_i(\vec{x}) = \sum_{j=1}^D [a_{ij} \cdot \sin(x_j) + b_{ij} \cdot \cos(x_j)], \quad (\text{A.15})$$

$\mathbf{A}$  e  $\mathbf{B}$  são duas matrizes de dimensão  $D \times D$ ,  $a_{ij}$  e  $b_{ij}$  são números inteiros aleatórios no intervalo  $[-100; 100]$ ,  $\vec{\alpha} = [\alpha_1, \dots, \alpha_D]$  e  $\alpha_j$  são valores aleatórios no intervalo  $[-\pi; \pi]$ . O ótimo global deste problema é dado por  $F_{12}(x^*) = F_{12}(\vec{\alpha}) = f_{12}^{bias} = -460$ , e sua representação gráfica no espaço bidimensional pode ser vista na Figura A.1(l).

## A.1.3 Problemas Multimodais Expandidos

As funções expandidas, que serão discutidas nesta seção, basicamente se baseiam em outras funções pré-existentes, geralmente de dimensões  $D' < D$ , e as aplicam sucessivamente a pares de todas as variáveis do espaço de busca, como ilustrado na Equação A.16. Nesta equação, tem-se uma função expandida  $EF(\vec{x})$  para um espaço de dimensão  $D$  ( $\vec{x} \in \mathfrak{R}^D$ ), que toma como base a função arbitrária de duas variáveis  $F(x, y)$ .

$$EF(x_1, x_2, \dots, x_D) = F(x_1, x_2) + F(x_2, x_3) + \dots + F(x_{D-1}, x_D) + F(x_D, x_1). \quad (\text{A.16})$$

### A.1.3.1 Problema $F_{13}$ : Função Expandida de Griewank Somada a Rosenbrock e Deslocada

O problema  $F_{13}$ , dado na Equação A.19, é um problema multimodal com variáveis não separáveis, que se utiliza da combinação das funções de Griewank e de Rosenbrock, dadas nas Equações A.17 e A.18, respectivamente.

$$F_G(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (\text{A.17})$$

$$F_R(\vec{x}) = \sum_{i=1}^{D-1} [100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2], \quad (\text{A.18})$$

$$F_{13}(\vec{x}) = F_G(F_R(z_1, z_2)) + F_G(F_R(z_2, z_3)) + \cdots + F_G(F_R(z_{D-1}, z_D)) + F_G(F_R(z_D, z_1)) + f_{13}^{bias}, \quad (\text{A.19})$$

onde  $\vec{z} = \vec{x} - \vec{o} + 1$ ,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-3; 1]^D$ ,  $D$  é o número de dimensões do problema,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_{13}(x^*) = f_{13}^{bias} = -130$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.2(a).

### A.1.3.2 Problema $F_{14}$ : Função $F6$ de Schaffer Expandida, Deslocada e Rotacionada

O problema  $F_{14}$ , dado na Equação A.21, é um problema multimodal com variáveis não separáveis, que se baseia na função  $F6$  de Schaffer, dada na Equação A.20, expandida, deslocada e rotacionada.

$$F_S(x, y) = 0,5 + \frac{[\sin^2(\sqrt{x^2 + y^2}) - 0,5]}{[1 + 0,001 \cdot (x^2 + y^2)]^2}, \quad (\text{A.20})$$

$$F_{14}(\vec{x}) = F_S(z_1, z_2) + F_S(z_2, z_3) + \cdots + F_S(z_{D-1}, z_D) + F_S(z_D, z_1) + f_{14}^{bias}, \quad (\text{A.21})$$

onde  $\vec{z} = (\vec{x} - \vec{o}) \cdot \mathbf{M}$ ,  $\mathbf{M}$  é uma matriz de transformação linear com número de condição 3,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $\vec{x} \in [-100; 100]^D$ ,  $D$  é o número de dimensões do problema,  $\vec{o} = x^*$  é um vetor de dimensão  $D$  com o ótimo global deslocado do problema e  $F_{14}(x^*) = f_{14}^{bias} = -300$ . A representação gráfica deste problema no espaço bidimensional pode ser vista na Figura A.2(b).

### A.1.4 Problemas Multimodais Compostos Híbridos

Os problemas multimodais compostos híbridos, que serão apresentados nesta seção, são problemas construídos a partir da combinação de uma série de outros problemas que geralmente possuem propriedades distintas. Todos os problemas que serão discutidos nesta seção têm a forma dada na Equação A.22, que será detalhada a seguir.

$$F_{Comp.Hib.}(\vec{x}) = \sum_{i=1}^n \left\{ w_i \cdot \left[ f'_i \left( \frac{(\vec{x} - \vec{o}_i)}{\lambda_i} \cdot \mathbf{M}_i \right) + bias_i \right] \right\} + f^{bias}, \quad (\text{A.22})$$

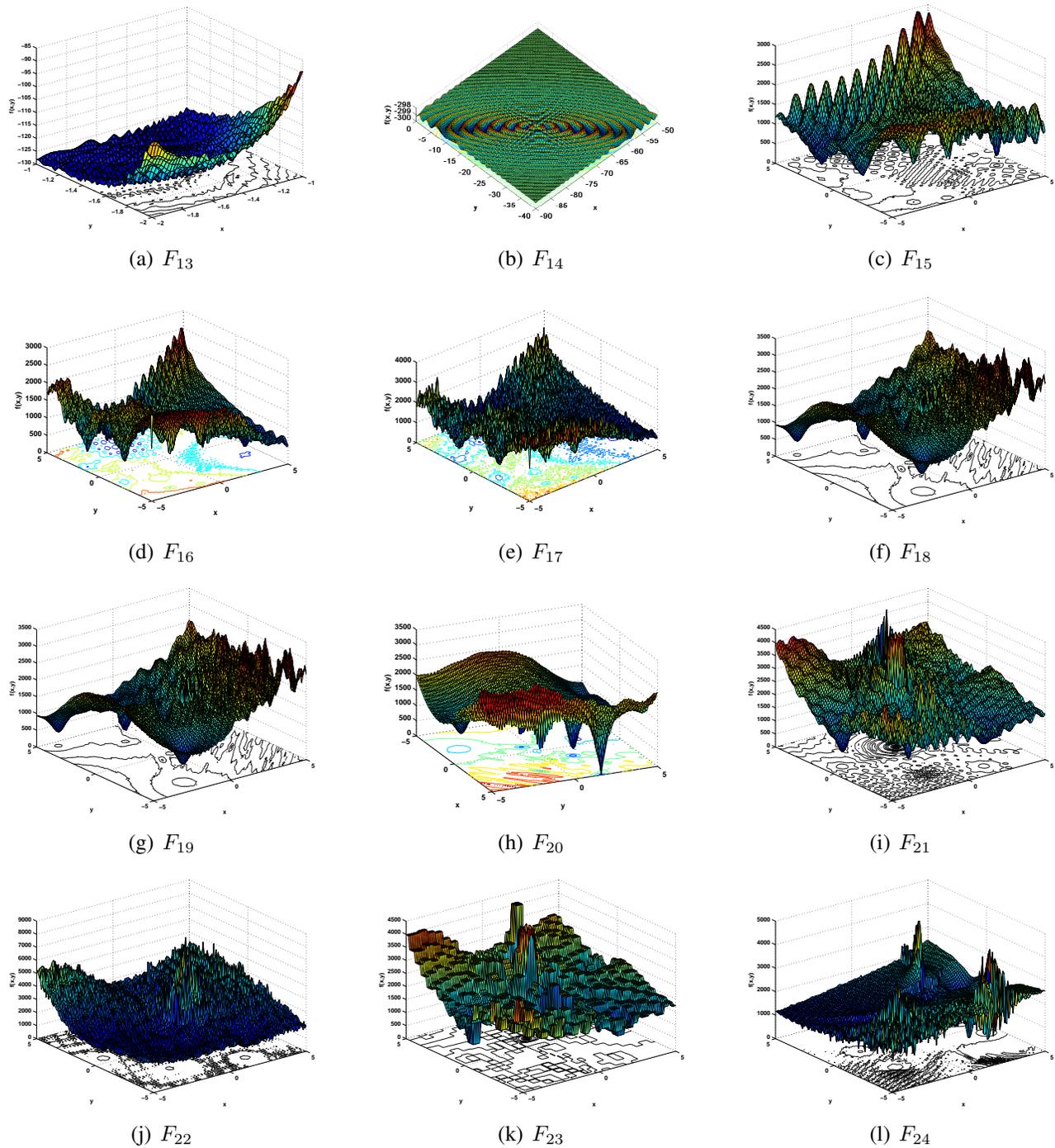


Fig. A.2: Representação gráfica, em  $\mathcal{R}^2$ , das funções de teste  $F_{13}$  a  $F_{24}$  da competição de Otimização de Parâmetros Reais do IEEE Congress on Evolutionary Computation (CEC) de 2005. A função  $F_{25}$  (não apresentada acima) é idêntica à  $F_{24}$ , exceto pelo fato do espaço de busca não ser limitado.

onde  $f_i(\vec{x})$  é a  $i$ -ésima função-base usada na construção da função composta,  $n$  é o número de funções-base,  $\vec{x} = [x_1, x_2, \dots, x_D]$ ,  $D$  é a dimensão do problema,  $\mathbf{M}_i$  é a matriz de transformação linear de cada  $f_i(\vec{x})$ ,  $\vec{o}_i$  é o ótimo deslocado de cada  $f_i(\vec{x})$ ,  $w_i$  é o peso atribuído a cada função-base,  $\lambda_i$  é um parâmetro utilizado para expandir ( $\lambda_i > 1$ ) ou comprimir ( $\lambda_i < 1$ ) a função  $f_i(\vec{x})$  e  $bias_i$  determina se o ótimo global de  $f_i(\vec{x})$  será ótimo global ( $bias_i = 0$ ) ou local ( $bias_i > 0$ ) de  $F_{Comp.Hib.}(\vec{x})$ .

O peso  $w_i$  atribuído a cada função-base  $f_i(\vec{x})$  é calculado através dos seguintes passos, sendo  $\sigma_i$  utilizado para controlar a cobertura de cada função-base  $f_i(\vec{x})$ :

1. 
$$w_i = \exp\left(-\frac{\sum_{k=1}^D (x_k - o_{ik})^2}{2 \cdot D \cdot \sigma_i^2}\right), \quad (\text{A.23})$$

2. 
$$w_i = \begin{cases} w_i & \text{se } w_i = \max(w_i) \\ w_i \cdot (1 - \max(w_i)^{10}) & \text{se } w_i \neq \max(w_i) \end{cases}, \quad (\text{A.24})$$

3. 
$$w_i = \frac{w_i}{\sum_{i=1}^n w_i} \quad (\text{A.25})$$

Caso as funções-base sejam diferentes, possivelmente elas também terão amplitudes diferentes, ou seja, elas terão valores máximos e mínimos diferentes entre si. Neste caso, para que se obtenha uma melhor composição  $F_{Comp.Hib.}(\vec{x})$ , sugere-se que o valor máximo  $f_i^{max}(\vec{x})$  para cada função-base seja estimado, e tais funções normalizadas conforme a Equação A.26.

$$f'_i(\vec{x}) = C \cdot \frac{f_i(\vec{x})}{|f_i^{max}(\vec{x})|}, \quad (\text{A.26})$$

onde  $C$  é uma constante pré-definida,  $|f_i^{max}(\vec{x})|$  é estimado por  $|f_i^{max}(\vec{x})| = f_i(\frac{\vec{x}'}{\lambda_i}) \cdot \mathbf{M}_i$  e  $\vec{x}' = [5, 5, \dots, 5]$ .

Em todas as funções que serão descritas a seguir, foi adotado o mesmo número de funções-base  $n = 10$ ,  $\vec{o}$  como sendo uma matriz de tamanho  $n \times D$  que define os ótimos globais de cada função-base  $f_i(\vec{x})$ ,  $C = 2000$  e  $bias = [0; 100; 200; 300; 400; 500; 600; 700; 800; 900]$ , o que determina que o ótimo global de  $F_{Comp.Hib.}(\vec{x})$  sempre será o ótimo global de sua primeira função-base  $f_1(\vec{x})$ .

#### A.1.4.1 Problema $F_{15}$ : Função Composta Híbrida

O problema  $F_{15}$  segue a Equação A.22, e tem as seguintes funções-base:

- $f_{1-2}(\vec{x})$  – Função de Rastrigin:

$$f_{1-2}(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) + 10], \quad (\text{A.27})$$

- $f_{3-4}(\vec{x})$  – Função de Weierstrass:

$$f_{3-4}(\vec{x}) = \sum_{i=1}^D \left\{ \sum_{k=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot (x_i + 0, 5))] \right\} - D \cdot \sum_{i=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot 0, 5)], \quad (\text{A.28})$$

- $f_{5-6}(\vec{x})$  – Função de Griewank:

$$f_{5-6}(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (\text{A.29})$$

- $f_{7-8}(\vec{x})$  – Função de Ackley:

$$f_{7-8}(\vec{x}) = -20 \cdot \exp\left(-0, 2 \cdot \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2 \cdot \pi \cdot x_i)\right) + 20 + e, \quad (\text{A.30})$$

- $f_{9-10}(\vec{x})$  – Função Esfera:

$$f_{9-10}(\vec{x}) = \sum_{i=1}^D x_i^2. \quad (\text{A.31})$$

Os parâmetros para este problema são:

- $\sigma_i = 1$  para  $i = 1, 2, \dots, D$ ;
- $\lambda = [1; 1; 10; 10; \frac{5}{60}; \frac{5}{60}; \frac{5}{32}; \frac{5}{32}; \frac{5}{100}; \frac{5}{100}]$ ; e
- $\mathbf{M}_i$  são todas matrizes-identidade.

O problema  $F_{15}$  é um problema multimodal com variáveis separáveis na região próxima ao ótimo global (onde predomina a função de Rastrigin), e tem como principais características a presença de grande quantidade de ótimos locais e duas áreas planas, introduzidas pela função Esfera. Para este problema, o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{15}(x^*) = f_{15}^{bias} = 120$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(c).

#### A.1.4.2 Problema $F_{16}$ : Versão Rotacionada da Função Composta Híbrida $F_{15}$

O problema  $F_{16}$  possui exatamente as mesmas características do problema  $F_{15}$ , exceto que, ao invés de  $M_i$  serem matrizes-identidade, neste caso elas são matrizes de transformação linear com número de condição 2. Para este problema, o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{16}(x^*) = f_{16}^{bias} = 120$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(d).

#### A.1.4.3 Problema $F_{17}$ : Problema $F_{16}$ com Ruído no *Fitness*

Se considerarmos  $G(\vec{x}) = F_{16}(\vec{x}) - f_{16}^{bias}$ , então o problema  $F_{17}$  será dado por:

$$F_{17}(\vec{x}) = G(\vec{x}) \cdot (1 + 0.2 \cdot |\mathcal{N}(0, 1)|) + f_{17}^{bias}, \quad (\text{A.32})$$

onde  $\mathcal{N}(0, 1)$  é um ruído gaussiano de média zero e variância unitária.

O problema  $F_{17}$  possui as mesmas características de  $F_{15}$  e  $F_{16}$ , exceto pela adição de ruído. O domínio deste problema é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{17}(x^*) = f_{17}^{bias} = 120$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(e).

#### A.1.4.4 Problema $F_{18}$ : Função Composta Híbrida Rotacionada

O problema  $F_{18}$  segue a Equação A.22, tendo as seguintes funções-base:

- $f_{1-2}(\vec{x})$  – Função de Ackley:

$$f_{1-2}(\vec{x}) = -20 \cdot \exp\left(-0, 2 \cdot \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2 \cdot \pi \cdot x_i)\right) + 20 + e, \quad (\text{A.33})$$

- $f_{3-4}(\vec{x})$  – Função de Rastrigin:

$$f_{3-4}(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) + 10], \quad (\text{A.34})$$

- $f_{5-6}(\vec{x})$  – Função Esfera:

$$f_{5-6}(\vec{x}) = \sum_{i=1}^D x_i^2, \quad (\text{A.35})$$

- $f_{7-8}(\vec{x})$  – Função de Weierstrass:

$$f_{7-8}(\vec{x}) = \sum_{i=1}^D \left\{ \sum_{k=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot (x_i + 0, 5))] \right\} - D \cdot \sum_{i=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot 0, 5)], \quad (\text{A.36})$$

- $f_{9-10}(\vec{x})$  – Função de Griewank:

$$f_{9-10}(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (\text{A.37})$$

Os parâmetros para este problema são:

- $\vec{\sigma} = [1; 2; 1, 5; 1, 5; 1; 1; 1, 5; 1, 5; 2; 2];$
- $\lambda = \left[2 \cdot \frac{5}{32}; \frac{5}{32}; 2 \cdot 1; 1; 2 \cdot \frac{5}{100}; \frac{5}{100}; 2 \cdot 10; 10; 2 \cdot \frac{5}{60}; \frac{5}{60}\right];$
- $\mathbf{M}_i$  são todas matrizes de rotação, com números de condição  $[2; 3; 2; 3; 2; 3; 20; 30; 200; 300];$  e
- $\vec{\sigma}_{10} = [0; 0; \dots 0].$

O problema  $F_{18}$  é um problema multimodal com variáveis não separáveis, e tem como principais características a presença de grande quantidade de ótimos locais, sendo um deles na origem, e também de duas áreas planas, introduzidas pela utilização da função Esfera. Para este problema, o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{18}(x^*) = f_{18}^{bias} = 10$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(f).

#### A.1.4.5 Problema $F_{19}$ : Função Composta Híbrida Rotacionada com Ótimo Global em um Vale Estreito

O problema  $F_{19}$  utiliza as mesmas funções-base do problema  $F_{18}$  com os mesmos parâmetros, exceto:

- $\vec{\sigma} = [0, 1; 2; 1, 5; 1, 5; 1; 1; 1, 5; 1, 5; 2; 2];$  e
- $\lambda = \left[0, 1 \cdot \frac{5}{32}; \frac{5}{32}; 2 \cdot 1; 1; 2 \cdot \frac{5}{100}; \frac{5}{100}; 2 \cdot 10; 10; 2 \cdot \frac{5}{60}; \frac{5}{60}\right]$

Com estas modificações nos parâmetros  $\vec{\sigma}$  e  $\vec{\lambda}$ , além das características já descritas para  $F_{18}$ ,  $F_{19}$  passa a apresentar também ótimo local dentro de um vale estreito. Para este problema, o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{19}(x^*) = f_{19}^{bias} = 10$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(g).

#### A.1.4.6 Problema $F_{20}$ : Função Composta Híbrida Rotacionada com ótimo global em um vale estreito

O problema  $F_{20}$  utiliza as mesmas funções-base do problema  $F_{18}$ , com os mesmos parâmetros. A diferença neste problema é que o ótimo global deslocado passa a ter  $\vec{\sigma}_{1(2j)} = 5$  para  $i = 1; 2; \dots; \lfloor \frac{D}{2} \rfloor$ , o que o posiciona na fronteira do espaço de busca. O domínio para este problema é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{20}(x^*) = f_{20}^{bias} = 10$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(h).

#### A.1.4.7 Problema $F_{21}$ : Função Composta Híbrida Rotacionada

O problema  $F_{21}$  novamente segue a Equação A.22, mas se utiliza de duas funções-base expandidas, como descrito na Seção A.1.3. Suas funções-base são:

- $f_{1-2}(\vec{x})$  – Função  $F6$  de Schaffer Expandida e Rotacionada:

$$F_S(x, y) = 0,5 + \frac{[\sin^2(\sqrt{x^2 + y^2}) - 0,5]}{[1 + 0,001 \cdot (x^2 + y^2)]^2}, \quad (\text{A.38})$$

$$f_{1-2}(\vec{x}) = F_S(x_1, x_2) + F_S(x_2, x_3) + \dots + F_S(x_{D-1}, x_D) + F_S(x_D, x_1), \quad (\text{A.39})$$

- $f_{3-4}(\vec{x})$  – Função de Rastrigin:

$$f_{3-4}(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) + 10], \quad (\text{A.40})$$

- $f_{5-6}(\vec{x})$  – Função Expandida de Griewank Somada a Rosenbrock:

$$F_G(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (\text{A.41})$$

$$F_R(\vec{x}) = \sum_{i=1}^{D-1} [100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2], \quad (\text{A.42})$$

$$f_{5-6}(\vec{x}) = F_G(F_R(x_1, x_2)) + F_G(F_R(x_2, x_3)) + \dots + F_G(F_R(x_{D-1}, x_D)) + F_G(F_R(x_D, x_1)), \quad (\text{A.43})$$

- $f_{7-8}(\vec{x})$  – Função de Weierstrass:

$$f_{7-8}(\vec{x}) = \sum_{i=1}^D \left\{ \sum_{k=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot (x_i + 0, 5))] \right\} - D \cdot \sum_{i=0}^{20} [0, 5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot 0, 5)], \quad (\text{A.44})$$

- $f_{9-10}(\vec{x})$  – Função de Griewank:

$$f_{9-10}(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (\text{A.45})$$

Os parâmetros para este problema são:

- $\vec{\sigma} = [1; 1; 1; 1; 1; 2; 2; 2; 2; 2];$
- $\lambda = \left[5 \cdot \frac{5}{100}; \frac{5}{100}; 5 \cdot 1; 1; 5 \cdot 1; 1; 5 \cdot 10; 10; 5 \cdot \frac{5}{200}; \frac{5}{200}\right]; \mathbf{e}$
- $\mathbf{M}_i$  são todas matrizes ortogonais.

O problema  $F_{21}$  é um problema multimodal com variáveis não separáveis, e tem como principais características a presença de grande quantidade de ótimos locais. Para este problema o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{20}(x^*) = f_{20}^{bias} = 360$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(i).

#### A.1.4.8 Problema $F_{22}$ : Função Composta Híbrida Rotacionada com Matrizes Mal Condicionadas

O problema  $F_{22}$  utiliza as mesmas funções-base do problema  $F_{21}$ , com os mesmos parâmetros exceto as matrizes  $\mathbf{M}_i$ , que passam a ter números de condição  $[10; 20; 50; 100; 200; 1000; 2000; \dots \dots 3000; 4000; 5000]$ . Isto faz com que este problema apresente o ótimo global na fronteira do espaço de busca.

O domínio para este problema é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{22}(x^*) = f_{22}^{bias} = 360$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(j).

#### A.1.4.9 Problema $F_{23}$ : Função Composta Híbrida Rotacionada Descontínua

O problema  $F_{23}$  utiliza as mesmas funções-base do problema  $F_{21}$  com os mesmos parâmetros. A única diferença é que este problema aplica um tratamento especial a todas as variáveis (Equação A.46), levando-o a perder suas características de continuidade.

$$x_j = \begin{cases} x_j & \text{se } |x_j - o_{1j}| < \frac{1}{2} \\ \frac{\text{round}(2 \cdot x_j)}{2} & \text{se } |x_j - o_{1j}| \geq \frac{1}{2} \end{cases}, \quad (\text{A.46})$$

onde  $j = 1; 2; \dots; D$ ,  $D$  é a dimensão do problema e o operador  $\text{round}(\cdot)$  é dado por:

$$\text{round}(x) = \begin{cases} a - 1 & \text{se } x \leq 0 \text{ e } b \geq 0,5 \\ a & \text{se } b < 0,5 \\ a + 1 & \text{se } x > 0 \text{ e } b \geq 0,5 \end{cases}, \quad (\text{A.47})$$

onde  $a$  é a parte inteira de  $x$  e  $b$  é sua parte decimal.

O domínio para este problema é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{23}(x^*) \approx f_{23}^{bias} = 360$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(k).

#### A.1.4.10 Problema $F_{24}$ : Função Composta Híbrida Rotacionada

Como o problema  $F_{21}$ , o problema  $F_{24}$  também segue a Equação A.22 e se utiliza de algumas funções-base expandidas, como descritas na Seção A.1.3, mas, diferentemente de  $F_{21}$ , algumas funções-base de  $F_{24}$  são descontínuas. As funções-base para este problema são:

- $f_1(\vec{x})$  – Função de Weierstrass:

$$f_1(\vec{x}) = \sum_{i=1}^D \left\{ \sum_{k=0}^{20} [0,5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot (x_i + 0,5))] \right\} - D \cdot \sum_{i=0}^{20} [0,5^k \cdot \cos(2 \cdot \pi \cdot 3^k \cdot 0,5)], \quad (\text{A.48})$$

- $f_2(\vec{x})$  – Função  $F6$  de Schaffer Expandida e Rotacionada:

$$F_S(x, y) = 0,5 + \frac{[\sin^2(\sqrt{x^2 + y^2}) - 0,5]}{[1 + 0,001 \cdot (x^2 + y^2)]^2}, \quad (\text{A.49})$$

$$f_2(\vec{x}) = F_S(x_1, x_2) + F_S(x_2, x_3) + \dots + F_S(x_{D-1}, x_D) + F_S(x_D, x_1). \quad (\text{A.50})$$

- $f_3(\vec{x})$  – Função Expandida de Griewank somada a Rosenbrock:

$$F_G(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (\text{A.51})$$

$$F_R(\vec{x}) = \sum_{i=1}^{D-1} [100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2], \quad (\text{A.52})$$

$$f_3(\vec{x}) = F_G(F_R(x_1, x_2)) + F_G(F_R(x_2, x_3)) + \cdots + F_G(F_R(x_{D-1}, x_D)) + F_G(F_R(x_D, x_1)), \quad (\text{A.53})$$

- $f_4(\vec{x})$  – Função de Ackley:

$$f_4(\vec{x}) = -20 \cdot \exp \left( -0,2 \cdot \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2 \cdot \pi \cdot x_i) \right) + 20 + e, \quad (\text{A.54})$$

- $f_5(\vec{x})$  – Função de Rastrigin:

$$f_5(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) + 10], \quad (\text{A.55})$$

- $f_6(\vec{x})$  – Função de Griewank:

$$f_6(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, \quad (\text{A.56})$$

- $f_7(\vec{x})$  – Função  $F_6$  de Schaffer Expandida e Descontínua:

$$F_S(x, v) = 0,5 + \frac{[\sin^2(\sqrt{x^2 + v^2}) - 0,5]}{[1 + 0,001 \cdot (x^2 + v^2)]^2}, \quad (\text{A.57})$$

$$f_7(\vec{x}) = F_S(y_1, y_2) + F_S(y_2, y_3) + \cdots + F_S(y_{D-1}, y_D) + F_S(y_D, y_1). \quad (\text{A.58})$$

$$y_j = \begin{cases} x_j & \text{se } |x_j| < \frac{1}{2} \\ \frac{\text{round}(2 \cdot x_j)}{2} & \text{se } |x_j| \geq \frac{1}{2} \end{cases} \quad \text{para } j = 1, 2, \dots, D, \quad (\text{A.59})$$

- $f_8(\vec{x})$  – Função de Rastrigin Descontínua:

$$f_8(\vec{x}) = \sum_{i=1}^D (y_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot y_i) + 10), \quad (\text{A.60})$$

$$y_j = \begin{cases} x_j & \text{se } |x_j| < \frac{1}{2} \\ \frac{\text{round}(2 \cdot x_j)}{2} & \text{se } |x_j| \geq \frac{1}{2} \end{cases} \text{ para } j = 1, 2, \dots, D, \quad (\text{A.61})$$

- $f_9(\vec{x})$  – Função Elíptica Altamente Condicionada:

$$f_9(\vec{x}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2, \quad (\text{A.62})$$

- $f_{10}(\vec{x})$  – Função Esfera com Ruído no *Fitness*:

$$f_{10}(\vec{x}) = \left( \sum_{i=1}^D x_i^2 \right) \cdot (1 + 0,1 \cdot |\mathcal{N}(0, 1)|), \quad (\text{A.63})$$

O operador  $\text{round}(\cdot)$  é o mesmo apresentado na Equação A.47 e os parâmetros para este problema são:

- $\sigma_i = 2$  para  $i = 1, 2, \dots, D$ ;
- $\lambda = \left[ 10; \frac{5}{20}; 1; \frac{5}{32}; 1; \frac{5}{100}; \frac{5}{50}; 1; \frac{5}{100}; \frac{5}{100} \right]; \mathbf{e}$
- $\mathbf{M}_i$  são matrizes de rotação com números de condição  $[100; 50; 30; 10; 5; 5; 4; 3; 2; 2]$ .

O problema  $F_{24}$  é um problema multimodal com variáveis não separáveis, e tem como principais características a combinação de uma grande quantidade de ótimos locais com a presença de regiões planas, introduzidas pelas funções-base monomodais. Para este problema o domínio é  $\vec{x} \in [-5; 5]^D$  e o valor da função-objetivo para o ótimo global é  $F_{24}(x^*) = f_{24}^{bias} = 260$ . Sua representação gráfica no espaço bidimensional pode ser vista na Figura A.2(1).

#### A.1.4.11 Problema $F_{25}$ : Função Composta Híbrida Rotacionada Ilimitada

O problema  $F_{25}$  é idêntico ao  $F_{24}$ , exceto pela ausência de um intervalo de busca pré-definido ( $F_{25}$  é ilimitado). Sendo assim, o domínio de inicialização das variáveis para este problema é  $\vec{x} \in [2; 5]^D$  e o ótimo global, que está posicionado fora do intervalo de inicialização, possui valor de função-objetivo  $F_{25}(x^*) = f_{25}^{bias} = 260$ .

## A.2 Problemas de Teste para Otimização Multiobjetivo

Os experimentos de otimização multiobjetivo, feitos com o intuito de avaliar o desempenho geral do algoritmo cob-aiNet[MO], se basearam em três conjuntos de problemas. O primeiro deles, cujas

funções serão discutidas na Seção A.2.1, é formado por quatro problemas com conjuntos de Pareto bem definidos, e foi utilizado principalmente para avaliar a capacidade do algoritmo em manter a diversidade no espaço de variáveis. Já o segundo conjunto de problemas, cujas funções serão discutidas na Seção A.2.2, se trata do grupo de funções de teste conhecido na literatura como funções *ZDT* (Zitzler et al., 2000). Por fim, o algoritmo cob-aiNet[MO] também foi aplicado ao problema conhecido como *DTLZ2* (Deb et al., 2002b), discutido na Seção A.2.3, para que fosse possível avaliar sua capacidade de tratar problemas com mais de dois objetivos a serem otimizados.

## A.2.1 Problemas Multimodais

### A.2.1.1 Deb & Tiwari

O primeiro problema que foi utilizado nos experimentos, designado aqui por *Deb & Tiwari*, foi proposto por Deb & Tiwari (2005) para avaliar o algoritmo conhecido como *omni-optimizer*. O problema *Deb & Tiwari* é um problema biobjetivo, multimodal, com fronteira de Pareto convexa e cujo conjunto de Pareto é distribuído em regiões separadas e bem definidas do espaço de variáveis. Os dois critérios deste problema são dados por:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = \sum_{i=1}^n \sin(\pi \cdot x_i) \\ f_2(\vec{x}) = \sum_{i=1}^n \cos(\pi \cdot x_i) \end{cases}, \quad (\text{A.64})$$

onde  $n = 5$  variáveis (dimensão do problema) e  $x_i \in [0, 6] \forall i \in \{1, \dots, n\}$ .

### A.2.1.2 EBN

O problema *EBN* utilizado aqui consiste em uma das funções da classe EBN de problemas proposta por Emmerich et al. (2005). Os problemas desta classe são interessantes pois todas as soluções factíveis também são Pareto-ótimas, o que os torna muito úteis para avaliar a capacidade que um dado algoritmo tem de distribuir as soluções no espaço de variáveis. Neste trabalho, foi considerado o problema EBN com fronteira de Pareto linear, cujas duas funções-objetivo são dadas por:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = \frac{1}{n} \cdot \sum_{i=1}^n |x_i| \\ f_2(\vec{x}) = \frac{1}{n} \cdot \sum_{i=1}^n |x_i - 1| \end{cases}, \quad (\text{A.65})$$

onde  $n = 10$  (dimensão do problema) e  $x_i \in [0, 1] \forall i \in \{1, \dots, n\}$ .

### A.2.1.3 Two-on-One 4

O problema *Two-on-One 4*, adotado aqui, corresponde à quarta função da classe de problemas *Two-on-One*, proposta por Preuss et al. (2006), e possui fronteira de Pareto convexa e não contínua. O problema *Two-on-One 4* é um problema biobjetivo bidimensional, cujo primeiro objetivo corresponde a um polinômio de 4ª ordem e o segundo à função esfera, como segue:

$$\text{Minimizar} \begin{cases} f_1(x_1, x_2) = x_1^4 + x_2^4 - x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20 \\ f_2(x_1, x_2) = (x_1 - 1)^2 + (x_2)^2 \end{cases}, \quad (\text{A.66})$$

onde  $x_i \in [-3, 3] \forall i \in \{1, 2\}$ .

### A.2.1.4 Lamé Supersphere

O problema denominado *Lamé Supersphere* pertence à família de problemas proposta por Emmerich & Deutz (2007), cujas fronteiras de Pareto possuem geometria esférica ou superesférica. Neste problema, também biobjetivo, a fronteira de Pareto é côncava e o conjunto de Pareto, de dimensão  $n = 4$ , é formado por conjuntos de soluções posicionadas em segmentos de linha paralelos e equiespaçados. As funções-objetivo deste problema são:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = (1 + r) \cdot \cos(x_1) \\ f_2(\vec{x}) = (1 + r) \cdot \sin(x_1) \end{cases}, \quad (\text{A.67})$$

onde  $r = \sin^2(\pi \cdot \xi)$ ,  $\xi = \frac{1}{n-1} \sum_{i=2}^n x_i$ ,  $x_1 \in [0, \frac{\pi}{2}]$ , e  $x_i \in [1, 5] \forall i \in \{2, \dots, n\}$ .

## A.2.2 A Classe de Problemas ZDT

A classe de problemas ZDT (Zitzler et al., 2000), muito utilizada na literatura de otimização multiobjetivo para avaliar o desempenho de novos algoritmos, é constituída de seis problemas biobjetivo com características diversas entre si, que possuem a seguinte estrutura:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) \\ f_2(\vec{x}) = g(\vec{x}) \cdot h(f_1(\vec{x}), g(\vec{x})) \end{cases}. \quad (\text{A.68})$$

De maneira geral, os seis problemas ZDT diferem basicamente na forma com que as três funções  $f_1(\vec{x})$ ,  $g(\vec{x})$  e  $h(\vec{x})$  são definidas, e todos eles, exceto o *ZDT5*, têm a fronteira de Pareto formada por  $g(\vec{x}) = 1$ .

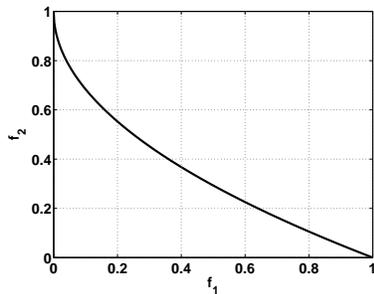
Dentre estes seis problemas, apenas o *ZDT5* não foi utilizado aqui, por se tratar de uma função booleana definida sobre cadeias binárias, o que foge do escopo deste trabalho. Os demais problemas serão detalhados a seguir.

A.2.2.1 ZDT1

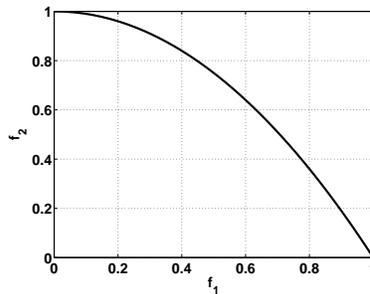
O problema *ZDT1* é um problema com  $n = 30$  variáveis e fronteira de Pareto contínua, convexa (vide Figura A.3(a)) e com soluções distribuídas uniformemente. Sua estrutura geral é a mesma apresentada na Equação A.68, com as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = x_1 \\ g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1(\vec{x}), g(\vec{x})) = 1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}} \end{cases} . \quad (\text{A.69})$$

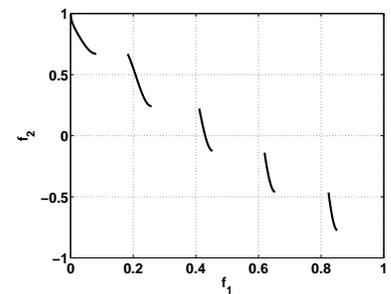
Neste problema, todas as variáveis pertencem ao domínio  $x_i \in [0; 1]$  e a região ótima de Pareto corresponde a  $0 \leq x_1^* \leq 1$  e  $x_i^* = 0$  para  $i = 2, \dots, n$ . De maneira geral, o problema *ZDT1* tende a ser o problema mais simples da classe ZDT, já que a única dificuldade que algum algoritmo pode vir a enfrentar está relacionada ao alto número de variáveis de decisão.



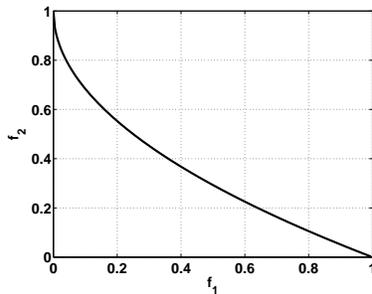
(a) *ZDT1*



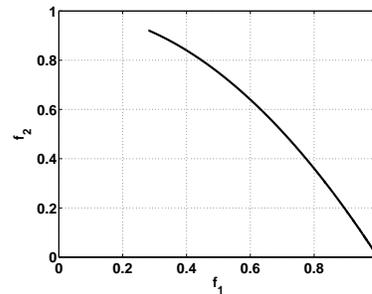
(b) *ZDT2*



(c) *ZDT3*



(d) *ZDT4*



(e) *ZDT6*

Fig. A.3: Representação gráfica da fronteira de Pareto de cada problema da classe ZDT utilizado neste trabalho.

### A.2.2.2 ZDT2

Da mesma maneira que o problema *ZDT1*, o problema *ZDT2* também possui  $n = 30$  variáveis e fronteira de Pareto contínua e com soluções distribuídas uniformemente mas, neste caso, ela passa a ser não convexa (vide Figura A.3(b)). Sua estrutura geral é a mesma apresentada na Equação A.68, com as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = x_1 \\ g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1(\vec{x}), g(\vec{x})) = 1 - \left( \frac{f_1(\vec{x})}{g(\vec{x})} \right)^2 \end{cases} . \quad (\text{A.70})$$

Todas as variáveis do problema *ZDT2* pertencem ao domínio  $x_i \in [0; 1]$  e a região ótima de Pareto corresponde a  $0 \leq x_1^* \leq 1$  e  $x_i^* = 0$  para  $i = 2, \dots, n$ . A principal dificuldade deste problema está no fato de sua fronteira não ser convexa, o que pode levar ferramentas baseadas em ponderação de objetivos a não conseguir uma boa distribuição das soluções.

### A.2.2.3 ZDT3

O problema *ZDT3* também possui  $n = 30$  variáveis mas sua principal característica está em sua fronteira de Pareto descontínua, formada por cinco segmentos (vide Figura A.3(c)). Sua estrutura geral é a mesma apresentada na Equação A.68, com as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = x_1 \\ g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1(\vec{x}), g(\vec{x})) = 1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}} - \left( \frac{f_1(\vec{x})}{g(\vec{x})} \right) \cdot \sin(10 \cdot \pi \cdot f_1(\vec{x})) \end{cases} . \quad (\text{A.71})$$

Todas as variáveis do problema *ZDT3* pertencem ao domínio  $x_i \in [0; 1]$  e a região ótima de Pareto corresponde a  $x_i^* = 0$  para  $i = 2, \dots, n$  e a alguns pontos em  $0 \leq x_1^* \leq 1$ . A principal dificuldade associada a este problema está justamente no fato de sua fronteira de Pareto ser descontínua, o que permite avaliar a capacidade de um dado algoritmo encontrar todas as regiões desta fronteira e povoá-las com soluções bem espalhadas.

**A.2.2.4 ZDT4**

O problema *ZDT4* possui  $n = 10$  variáveis, uma fronteira de Pareto convexa (vide Figura A.3(d)) e a mesma estrutura geral apresentada na Equação A.68, com as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = x_1 \\ g(\vec{x}) = 1 + 10 \cdot (n - 1) + \sum_{i=2}^n [x_i^2 - 10 \cdot \cos(4 \cdot \pi \cdot x_i)] \\ h(f_1(\vec{x}), g(\vec{x})) = 1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}} \end{cases} . \quad (\text{A.72})$$

A variável  $x_1$  pertence ao domínio  $[0; 1]$  e todas as demais pertencem a  $[-5; 5]$ . A principal dificuldade deste problema está no fato dele possuir  $21^9$  (cerca de  $8 \cdot 10^{11}$  – Deb, 2001) soluções Pareto-ótimas locais, cada uma correspondendo a  $0 \leq x_1^* \leq 1$  e  $x_i^* = 0, 5 \cdot m$ , onde  $m$  é qualquer valor inteiro em  $[-10; 10]$  e  $i = 2, \dots, n$ . O ótimo global de Pareto corresponde a  $0 \leq x_1^* \leq 1$  e  $x_i^* = 0$  para  $i = 2, \dots, n$ , o que leva a  $g(\vec{x}^*) = 1, 0$ . No entanto, existem também outras 99 fronteiras correspondentes a ótimos locais do problema, posicionadas em  $g(\vec{x}) = 1, 25$ ,  $g(\vec{x}) = 1, 5$  e assim por diante até  $g(\vec{x}) = 25$ . Esta alta quantidade de ótimos locais torna o problema *ZDT4* altamente desafiador para algoritmos de otimização multiobjetivo, mesmo se tratando de um problema com apenas 10 variáveis.

**A.2.2.5 ZDT6**

Como o problema *ZDT4*, o problema *ZDT6* também possui  $n = 10$  variáveis, mas sua fronteira de Pareto não é convexa (vide Figura A.3(e)) e a densidade de soluções na fronteira de Pareto não é uniforme. Sua estrutura geral é a mesma apresentada na Equação A.68, com as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = 1 - \exp(-4 \cdot x_1) \cdot \sin^6(6 \cdot \pi \cdot x_1) \\ g(\vec{x}) = 1 + 9 \cdot \left(\frac{\sum_{i=2}^n x_i}{9}\right)^{0,25} \\ h(f_1(\vec{x}), g(\vec{x})) = 1 - \left(\frac{f_1(\vec{x})}{g(\vec{x})}\right)^2 \end{cases} . \quad (\text{A.73})$$

Todas as variáveis do problema *ZDT6* pertencem ao domínio  $x_i \in [0; 1]$  e a região ótima de Pareto corresponde a  $0 \leq x_1^* \leq 1$  e  $x_i^* = 0$  para  $i = 2, \dots, n$ .

Para ilustrar a não uniformidade da distribuição das soluções na fronteira de Pareto, na Figura A.4 estão representadas, no espaço de objetivos, 100 soluções uniformemente distribuídas no conjunto de Pareto, ou seja, soluções ótimas do problema e que estão uniformemente distribuídas no espaço de variáveis ( $x_1^* \in [0; 1]$ ). Como é possível observar na Figura A.4, apesar da distribuição uniforme no

espaço de variáveis, no espaço de objetivos as soluções tendem a se agrupar na região da fronteira em que  $f_2(\vec{x}) \approx 0$ . Esta característica do problema *ZDT6*, aliada à não convexidade de sua fronteira de Pareto, pode dificultar a convergência de muitos algoritmos de otimização multiobjetivo.

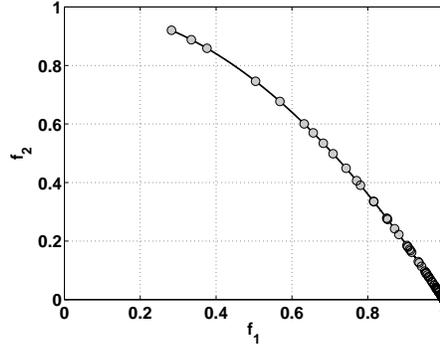


Fig. A.4: Representação gráfica da fronteira de Pareto do problema *ZDT6* (linha contínua) e de 100 soluções uniformemente distribuídas no conjunto de Pareto do problema (círculos).

### A.2.3 Problema DTLZ2

A classe de problemas conhecida como *DTLZ*, proposta por Deb et al. (2002b), consiste em um conjunto de nove problemas de teste genéricos, cuja principal característica é a escalabilidade quanto ao número de objetivos a serem otimizados. Justamente por apresentarem tal característica, os problemas *DTLZ* são geralmente utilizados para avaliar o comportamento de algoritmos em situações com elevado número de objetivos, situações estas que reconhecidamente levam a uma diminuição da pressão evolutiva sobre os indivíduos da população (Sato et al., 2007).

Dentre as nove funções da classe *DTLZ*, foi utilizado neste trabalho o problema conhecido como *DTLZ2*, que tem como principal característica uma fronteira de Pareto esférica e não convexa, como ilustrado na Figura A.5. As funções-objetivo para este problema são definidas da seguinte forma:

$$\text{Minimizar} \begin{cases} f_1(\vec{x}) = (1 + g(\vec{x}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \cos(x_2 \cdot \frac{\pi}{2}) \cdots \cos(x_{M-2} \cdot \frac{\pi}{2}) \cos(x_{M-1} \cdot \frac{\pi}{2}), \\ f_2(\vec{x}) = (1 + g(\vec{x}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \cos(x_2 \cdot \frac{\pi}{2}) \cdots \cos(x_{M-2} \cdot \frac{\pi}{2}) \sin(x_{M-1} \cdot \frac{\pi}{2}), \\ f_3(\vec{x}) = (1 + g(\vec{x}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \cos(x_2 \cdot \frac{\pi}{2}) \cdots \sin(x_{M-2} \cdot \frac{\pi}{2}) \\ \vdots \\ f_{M-1}(\vec{x}) = (1 + g(\vec{x}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \sin(x_2 \cdot \frac{\pi}{2}), \\ f_M(\vec{x}) = (1 + g(\vec{x}_M)) \sin(x_1 \cdot \frac{\pi}{2}), \end{cases} \quad (\text{A.74})$$

onde  $g(\vec{x}_M) = \sum_{x_i \in X_M} (x_i - 0,5)^2$ ,  $0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$ ,  $M$  é o número de objetivos do

problema e  $X_M$  é um subconjunto de  $k$  variáveis de cada solução  $\vec{x}$ . Para este problema, recomenda-se que  $k = 10$  e a dimensão de cada solução corresponda a  $n = M + k - 1$ . Sendo assim, como neste trabalho foi adotado  $M = 3$  objetivos para o problema *DTLZ2*, a dimensão de cada solução candidata foi definida como  $n = 3 + 10 - 1 = 12$  variáveis, sendo que apenas  $k = 10$  destas variáveis são efetivamente utilizadas no cálculo das funções-objetivo.

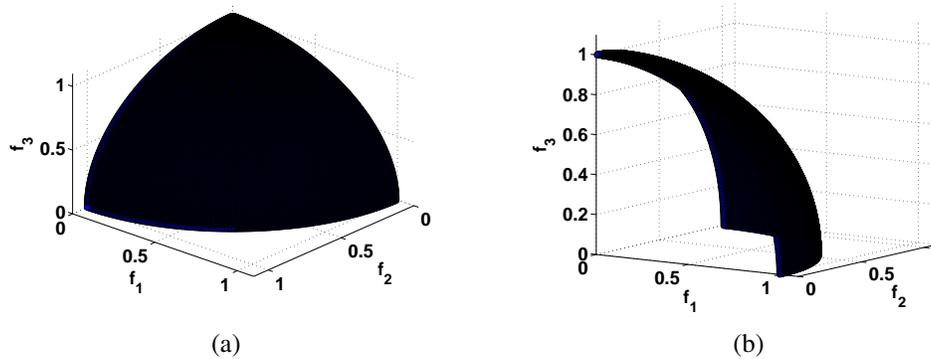


Fig. A.5: Representação gráfica da fronteira de Pareto do problema *DTLZ2*.



# Apêndice B

## Trabalhos Publicados pelo Autor

### B.1 Periódicos

1. Castro, P. A. D., de França, F. O., Ferreira, H. M., Coelho, G. P., & Von Zuben, F. J. (2010). Query expansion using an immune-inspired biclustering algorithm. *Natural Computing*, 9(3), 579–602.
2. Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2010). An immune-inspired multi-objective approach to the reconstruction of phylogenetic trees. *Journal of Neural Computing and Applications*, 19(8), 1103–1132.
3. de França, F. O., Coelho, G. P., Castro, P. A. D., & Von Zuben, F. J. (2010). Conceptual and practical aspects of the aiNet family of algorithms. *International Journal of Natural Computing Research*, 1(1), 1–35.
4. Pasti, R., de Castro, L. N., Coelho, G. P., & Von Zuben, F. J. (2010). Neural network ensembles: Immune-inspired approaches to the diversity of components. *Natural Computing*, 9, 625–653.
5. Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009). Multi-objective biclustering: When non-dominated solutions are not enough. *Journal of Mathematical Modelling and Algorithms*, 8(2), 175–202.

### B.2 Capítulos de Livros e Anais de Congressos, Simpósios ou Encontros

1. Azuma, R. M., Coelho, G. P., & Von Zuben, F. J. (2011). Evolutionary multi-objective optimization for the vendor-managed inventory routing problem. Em *Proc. of the 2011 IEEE Congress on Evolutionary Computation (CEC)*. New Orleans, USA.

2. Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2011). A concentration-based artificial immune network for combinatorial optimization. Em *Proc. of the 2011 IEEE Congress on Evolutionary Computation (CEC)*. New Orleans, USA.
3. Coelho, G. P., & Von Zuben, F. J. (2011). A concentration-based artificial immune network for multi-objective optimization. Em R. Takahashi, K. Deb, E. Wanner, & S. Greco (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 6th. International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, vol. 6576 de *Lecture Notes in Computer Science*, (pp. 343–357). Springer Berlin / Heidelberg.
4. Coelho, G. P., & Von Zuben, F. J. (2010). A concentration-based artificial immune network for continuous optimization. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 108–115). Barcelona, Spain.
5. de França, F. O., Coelho, G. P., & Von Zuben, F. J. (2010). On the diversity mechanisms of opt-aiNet: A comparative study with fitness sharing. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 3523–3530). Barcelona, Spain.
6. Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009). Improving a multi-objective multipopulation artificial immune network for biclustering. Em *Proc. of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2748–2755). Trondheim, Norway.
7. Coelho, G. P., & Von Zuben, F. J. (2009). Uma proposta imuno-inspirada para omni-otimização. Em *Anais do Segundo Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial (EADCA)*. Campinas, Brazil.
8. de França, F. O., Coelho, G. P., & Von Zuben, F. J. (2009). Coherent recommendations using biclustering. Em *Anais do 30º Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia (CILAMCE)*. Armação dos Búzios, Brazil.
9. Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2008). A multi-objective multipopulation approach for biclustering. Em *Artificial Immune Systems – Proc. of the 7th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 5132 de *Lecture Notes in Computer Science*, (pp. 71–82). Springer Berlin / Heidelberg.
10. de França, F. O., Coelho, G. P., Von Zuben, F. J., & Attux, R. R. F. (2008). Multivariate ant colony optimization in continuous search spaces. Em *Proc. of the 10th Conference on Genetic and Evolutionary Computation (GECCO)*, (pp. 9–16). Atlanta, USA.
11. Reis, P. A., Lyra Filho, C., Cavellucci, C., Von Zuben, F. J., Usberti, F. L., Gonzáles, J. F. V., Coelho, G. P., & Ferreira, H. M. (2008). Problema de alocação ótima de recursos de manutenção: Formulação

- e estudos de caso. Em *Anais do XL Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, vol. 1, (pp. 447–459).
12. Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007). A multiobjective approach to phylogenetic trees: Selecting the most promising solutions from the Pareto front. Em *Proc. of the 7th. International Conference on Intelligent Systems Design and Applications (ISDA)*, (pp. 837–842). Rio de Janeiro, Brazil
  13. Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007). Evolving phylogenetic trees: A multiobjective approach. Em M.-F. Sagot, & M. Walter (Eds.) *Advances in Bioinformatics and Computational Biology – Proc. of the 2007 Brazilian Symposium on Bioinformatics (BSB)*, vol. 4643 de *Lecture Notes in Computer Science*, (pp. 113–125). Springer Berlin / Heidelberg.
  14. Coelho, G. P., & Von Zuben, F. J. (2006). The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. Em *Proc. of the 2006 IEEE International Joint Conference on Neural Networks (IJCNN)*, (pp. 5132–5139). Vancouver, Canada.
  15. Coelho, G. P., & Von Zuben, F. J. (2006). omni-aiNet: An immune-inspired approach for omni optimization. Em *Artificial Immune Systems – Proc. of the 5th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 294–308). Springer Berlin / Heidelberg.
  16. Castro, P. A. D., Coelho, G. P., Caetano, M. F., & Von Zuben, F. J. (2005). Designing ensembles of fuzzy classification systems: An immune-inspired approach. Em C. Jacob, M. Pilat, P. Bentley, & J. Timmis (Eds.) *Artificial Immune Systems – Proc. of the 4th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 3627 de *Lecture Notes in Computer Science*, (pp. 469–482). Springer Berlin / Heidelberg.
  17. Coelho, G. P., Castro, P. A. D., & Von Zuben, F. J. (2005). The effective use of diverse rule bases in fuzzy classification. Em *Anais do VII Congresso Brasileiro de Redes Neurais (CBRN)*. Natal, Brazil.

### B.3 Resumos Expandidos

1. de França, F. O., Coelho, G. P., & Von Zuben, F. J. (2008). bicACO: An ant colony inspired biclustering algorithm. Em M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, & A. Winfield (Eds.) *Ant Colony Optimization and Swarm Intelligence*, vol. 5217 de *Lecture Notes in Computer Science*, (pp. 401–402). Springer Berlin / Heidelberg.



# Referências Bibliográficas

- Abbattista, F., Di Gioia, G., Di Santo, G., & Farinelli, A. M. (1996). An associative memory based on the immune networks. Em *Proc. of the 1996 IEEE International Conference on Neural Networks*, (pp. 519–523).
- Aickelin, U., & Cayzer, S. (2002). The danger theory and its application to artificial immune systems. Em J. Timmis, & P. J. Bentley (Eds.) *Proc. of the 1st. International Conference on Artificial Immune Systems (ICARIS)*, (pp. 141–148).
- Aisu, H., & Mitzutani, H. (1996). Immunity-based learning – Integration of distributed search and constraint relaxation. Em *Proc. of the 1996 International Conference on Multi-Agent Systems*, (pp. 124–135). Kyoto, Japan.
- Auger, A., & Hansen, N. (2005a). Performance evaluation of an advanced local search evolutionary algorithm. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1777–1784).
- Auger, A., & Hansen, N. (2005b). A restart CMA evolution strategy with increasing population size. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1769–1776).
- Bakhouya, M., Rodriguez, S., Hilaire, V., Koukam, A., & Gaber, J. (2003). Intelligent immune-based system for autonomous soccer robots. Em *Proc. of the 2003 FIRA Robot World Congress*.
- Ballester, P. J., & Carter, J. N. (2004). An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. Em *Proc. of the 2004 Genetic and Evolutionary Computation Conference (GECCO)*, vol. 3102 de *Lecture Notes in Computer Science*, (pp. 901–913). Springer Berlin / Heidelberg.
- Ballester, P. J., Stephenson, J., Carter, J. N., & Gallagher, K. (2005). Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 498–505).

- Bauerle, N., & Rieder, U. (2004). Portfolio optimization with Markov-modulated stock prices and interest rates. *IEEE Transactions on Automatic Control*, 49(3), 442–447.
- Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (1993). *Nonlinear Programming: Theory and Algorithms*. Wiley, 2 ed.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.) (2000). *Evolutionary Computation I: Basic Algorithms and Operators*. Taylor & Francis.
- Behn, U. (2007). Idiotypic networks: Toward a renaissance? *Immunological reviews*, 216, 142–152.
- Bersini, H. (1991). Immune network and adaptive control. Em *Proc. of the 1st. European Conference on Artificial Life*, (pp. 217–226).
- Bersini, H. (1998). The endogenous double plasticity of the immune network and the inspiration to be drawn for engineering aircrafts. Em D. Dasgupta (Ed.) *Artificial Immune Systems and their Applications*, (pp. 22–41). Springer.
- Bersini, H. (2002). Self-assertion vs self-recognition: A tribute to Francisco Varela. Em *Proc. of the 1st International Conference on Artificial Immune Systems (ICARIS)*, (pp. 107–112).
- Bersini, H. (2003). Revisiting idiotypic immune networks. Em *Advances in Artificial Life – Proc. of the 7th European Conference on Advances in Artificial Life (ECAL)*, vol. 2801 de *Lecture Notes in Computer Science*, (pp. 164–174). Springer Berlin / Heidelberg.
- Bersini, H., & Varela, F. J. (1990). Hints for adaptive problem solving gleaned from immune networks. Em *Proc. of the 1990 Parallel Problem Solving from Nature (PPSN I)*, (pp. 343–354).
- Bersini, H., & Varela, F. J. (1991). Hints for adaptive problem solving gleaned from immune networks. Em H.-P. Schwefel, & R. Männer (Eds.) *Parallel Problem Solving from Nature*, vol. 496 de *Lecture Notes in Computer Science*, (pp. 343–354). Springer Berlin / Heidelberg.
- Bersini, H., & Varela, F. J. (1994). The immune learning mechanisms: Reinforcement, recruitment and their applications. Em R. Paton (Ed.) *Computing with Biological Metaphors*. Chapman & Hall.
- Bezerra, G. B., & de Castro, L. N. (2003). Bioinformatics data analysis using an artificial immune network. Em *Artificial Immune Systems – Proc. of the 2nd. International Conference on Artificial Immune Systems (ICARIS)*, vol. 2787 de *Lecture Notes in Computer Science*, (pp. 22–33). Springer Berlin / Heidelberg.

- Brede, M., & Behn, U. (2003). Patterns in randomly evolving networks: Idiotypic networks. *Physical Review E*, 67(3), 031920.
- Burnet, F. M. (1978). Clonal selection and after. Em G. I. Bell, A. S. Perelson, & G. H. Pimbley Jr (Eds.) *Theoretical Immunology*, (pp. 63–85). Marcel Dekker Inc.
- Caetano, M. F. (2006). *Síntese Sonora Auto-Organizável Através da Aplicação de Algoritmos Bio-Inspirados*. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas, Brasil.
- Campelo, F., Guimarães, F., & Igarashi, H. (2007). Overview of artificial immune systems for multi-objective optimization. Em S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 4th. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 4403 de *Lecture Notes in Computer Science*, (pp. 937–951). Springer Berlin / Heidelberg.
- Carter, J. H. (2000). The immune system as a model for pattern recognition and classification. *Journal of the American Medical Informatics Association*, 7(1), 28–41.
- Carvalho, D. R., & Freitas, A. A. (2001). An immunological algorithm for discovering small-disjunct rules in data mining. Em *Proc. of the 2001 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 401–404).
- Castro, P. A. D. (2009). *Sinergia entre Sistemas Imunológicos Artificiais e Modelos Gráficos Probabilísticos*. Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- Castro, P. A. D., Coelho, G. P., Caetano, M. F., & Von Zuben, F. J. (2005). Designing ensembles of fuzzy classification systems: An immune-inspired approach. Em C. Jacob, M. Pilat, P. J. Bentley, & J. Timmis (Eds.) *Artificial Immune Systems – Proc. of the 4th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 3627 de *Lecture Notes in Computer Science*, (pp. 469–482). Springer Berlin / Heidelberg.
- Castro, P. A. D., de França, F. O., Ferreira, H. M., Coelho, G. P., & Von Zuben, F. J. (2010). Query expansion using an immune-inspired biclustering algorithm. *Natural Computing*, 9(3), 579–602.
- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007a). Applying biclustering to perform collaborative filtering. Em *Proc. of the 7th. International Conference on Intelligent Systems Design and Applications (ISDA)*, (pp. 421–426).

- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007b). Applying biclustering to text mining: An immune-inspired approach. Em *Artificial Immune Systems – Proc. of the 6th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 4628 de *Lecture Notes in Computer Science*, (pp. 83–94). Springer Berlin / Heidelberg.
- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007c). Evaluating the performance of a biclustering algorithm applied to collaborative filtering – A comparative analysis. Em *Proc. of the 7th. International Conference on Hybrid Intelligent Systems (HIS)*, (pp. 65–70).
- Castro, P. A. D., & Von Zuben, F. J. (2008). MOBAIS: A Bayesian artificial immune system for multi-objective optimization. Em P. J. Bentley, D. Lee, & S. Jung (Eds.) *Artificial Immune Systems – Proc. of the 7th International Conference on Artificial Immune Systems (ICARIS)*, vol. 5132 de *Lecture Notes in Computer Science*, (pp. 48–59). Springer Berlin / Heidelberg.
- Castro, P. A. D., & Von Zuben, F. J. (2009). BAIS: A Bayesian artificial immune system for the effective handling of building blocks. *Information Sciences*, 179(10), 1426–1440.
- Castro, P. A. D., & Von Zuben, F. J. (2010a). GAIS: A Gaussian artificial immune system for continuous optimization. Em E. Hart, C. McEwan, J. Timmis, & A. Hone (Eds.) *Artificial Immune Systems – Proc. of the 9th International Conference on Artificial Immune Systems (ICARIS)*, vol. 6209 de *Lecture Notes in Computer Science*, (pp. 171–184). Springer Berlin / Heidelberg.
- Castro, P. A. D., & Von Zuben, F. J. (2010b). A Gaussian artificial immune system for multi-objective optimization in continuous domains. Em *Proc. of the 10th International Conference on Hybrid Intelligent Systems (HIS)*, (pp. 159–164). Atlanta, USA.
- Cazangi, R. R., & Von Zuben, F. J. (2006). Immune learning classifier networks: Evolving nodes and connections. Em *Proc. of the 2006 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 7994–8001). Vancouver, Canada.
- Chan, K. P., & Ray, T. (2005). An evolutionary algorithm to maintain diversity in the parametric and the objective space. Em *Proc. of the 2005 International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS)*.
- Chun-Kit, A., & Ho-Fung, L. (2010). Improving CMA-ES by random evaluation on the minor eigenspace. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*.
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2008). A multi-objective multipopulation approach for biclustering. Em *Artificial Immune Systems – Proc. of the 7th. International Conference on*

- Artificial Immune Systems (ICARIS)*, vol. 5132 de *Lecture Notes in Computer Science*, (pp. 71–82). Springer Berlin / Heidelberg.
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009a). Improving a multi-objective multi-population artificial immune network for biclustering. Em *Proc. of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2748–2755).
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009b). Multi-objective biclustering: When non-dominated solutions are not enough. *Journal of Mathematical Modelling and Algorithms*, 8(2), 175–202.
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007a). Evolving phylogenetic trees: A multiobjective approach. Em M.-F. Sagot, & M. Walter (Eds.) *Advances in Bioinformatics and Computational Biology – Proc. of the 2007 Brazilian Symposium on Bioinformatics (BSB)*, vol. 4643 de *Lecture Notes in Computer Science*, (pp. 113–125). Springer Berlin / Heidelberg.
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007b). A multiobjective approach to phylogenetic trees: Selecting the most promising solutions from the Pareto front. Em *Proc. of the 7th. International Conference on Intelligent Systems Design and Applications (ISDA)*, (pp. 837–842).
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2010). An immune-inspired multi-objective approach to the reconstruction of phylogenetic trees. *Journal of Neural Computing and Applications*, 19(8), 1103–1132.
- Coelho, G. P., & Von Zuben, F. J. (2006a). The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. Em *Proc. of the 2006 IEEE International Joint Conference on Neural Networks (IJCNN)*, (pp. 5132–5139). Vancouver, Canada.
- Coelho, G. P., & Von Zuben, F. J. (2006b). omni-aiNet: An immune-inspired approach for omni optimization. Em *Artificial Immune Systems – Proc. of the 5th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 294–308). Springer Berlin / Heidelberg.
- Coelho, G. P., & Von Zuben, F. J. (2010). A concentration-based artificial immune network for continuous optimization. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 108–115).
- Coelho, G. P., & Von Zuben, F. J. (2011). A concentration-based artificial immune network for multi-objective optimization. Em R. Takahashi, K. Deb, E. Wanner, & S. Greco (Eds.) *Evolutionary*

- Multi-Criterion Optimization – Proc. of the 6th. International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, vol. 6576 de *Lecture Notes in Computer Science*, (pp. 343–357). Springer Berlin / Heidelberg.
- Coelho, L. S., Souza, R. C. T., & Mariani, V. C. (2009c). Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems. *Mathematics and Computers in Simulation*, 79(10).
- Coello Coello, C. A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3), 129–156.
- Coello Coello, C. A. (2006). Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1), 28–36.
- Coello Coello, C. A., & Cruz Cortés, N. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6, 163–190.
- Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, 2 ed.
- Coello Coello, C. A., & Toscano Pulido, G. (2001). Multiobjective optimization using a micro-genetic algorithm. Em *Proc. of the 2001 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 274–282). San Francisco, USA.
- Colutto, S., Fruhauf, F., Fuchs, M., & Scherzer, O. (2010). The CMA-ES on Riemannian manifolds to reconstruct shapes in 3-D voxel images. *IEEE Transactions on Evolutionary Computation*, 14(2), 227–245.
- Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. Em *Proc. of the 2001 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 283–290). San Francisco, USA.
- Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). The Pareto envelope-based selection algorithm for multiobjective optimization. Em M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo-Guervós, & H.-P. Schwefel (Eds.) *Parallel Problem Solving from Nature – PPSN VI*, vol. 1917 de *Lecture Notes in Computer Science*, (pp. 839–848). Springer Berlin / Heidelberg.
- Dallwig, S., Neumaier, A., & Schichl, H. (1997). GLOPT – A program for constrained global optimization. Em I. M. Bomze, T. Csendes, R. Horst, & P. M. Pardalos (Eds.) *Developments in global optimization*, Nonconvex Optimization and its Applications. Kluwer Academic Publishers.

- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London, UK: John Murray.
- Dasgupta, D. (1996). A new algorithm for anomaly detection in time series data. Em *Proc. of the 1996 International Conference on Knowledge-Based Computer Systems (KBCS)*, vol. 1.
- Dasgupta, D. (Ed.) (1998). *Artificial Immune Systems and their Applications*. Springer Verlag.
- Dasgupta, D. (1999). Immunity-based intrusion detection system. Em *Proc. of the 22nd. National Information Systems Security Conference*, (pp. 147–160). Arlington, USA.
- Dasgupta, D., Cao, Y., & Yang, C. (1999). An immunogenetic approach to spectra recognition. Em *Proc. of the 1999 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 149–155).
- Dasgupta, D., & Forrest, S. (1998). An anomaly detection algorithm inspired by the immune system. Em D. Dasgupta (Ed.) *Artificial Immune Systems and Their Applications*, (pp. 262–277). Springer.
- De Boer, R. J. (1989). Poor repertoire selection in symmetric idiotypic network models. *Immunology Letters*, 22, 101–112.
- De Boer, R. J., & Perelson, A. S. (1991). Size and connectivity as emergent properties of a developing immune network. *Journal of Theoretical Biology*, 149(3), 381–424.
- de Carvalho, A. B., & Pozo, A. (2010). Analyzing the control of dominance area of solutions in particle swarm optimization for many-objective. Em *Proc. of the 10th International Conference on Hybrid Intelligent Systems (HIS)*, (pp. 103–108).
- de Castro, L. N. (2006). *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. Chapman & Hall/CRC Computer & Information Science Series. Chapman & Hall/CRC.
- de Castro, L. N., & Timmis, J. (2002a). An artificial immune network for multimodal function optimization. Em *Proc. of the 2002 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 699–704).
- de Castro, L. N., & Timmis, J. (2002b). *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer Verlag.
- de Castro, L. N., & Von Zuben, F. J. (1999). Artificial immune systems: Part I – basic theory and applications. Relatório Técnico DCA-RT 01/99, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

- de Castro, L. N., & Von Zuben, F. J. (2002a). aiNet: An artificial immune network for data analysis. Em H. A. Abbass, R. A. Sarker, & C. S. Newton (Eds.) *Data Mining: A Heuristic Approach*, (pp. 231–259). IGI Global.
- de Castro, L. N., & Von Zuben, F. J. (2002b). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3), 239–251.
- de Castro, L. N., Von Zuben, F. J., & de Deus Jr., G. A. (2003). The construction of a Boolean competitive neural network using ideas from immunology. *Neurocomputing*, 50, 51–85.
- de França, F. O., Bezerra, G. B., de Castro, L. N., & Von Zuben, F. J. (2006a). New perspectives for the biclustering problem. Em *Proc. of the 2006 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2768–2775).
- de França, F. O., Coelho, G. P., Castro, P. A. D., & Von Zuben, F. J. (2010a). Conceptual and practical aspects of the aiNet family of algorithms. *International Journal of Natural Computing Research*, 1(1), 1–35.
- de França, F. O., Coelho, G. P., & Von Zuben, F. J. (2010b). On the diversity mechanisms of opt-aiNet: A comparative study with fitness sharing. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 3523–3530).
- de França, F. O., de Castro, L. N., & Von Zuben, F. J. (2005). Artificial immune network for multi-modal function optimization on dynamic environments. Em *Proc. of the 2005 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 289–296).
- de França, F. O., Gomes, L. C. T., de Castro, L. N., & Von Zuben, F. J. (2006b). Handling time-varying TSP instances. Em *Proc. of the 2006 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 9735–9742).
- de Sousa, J. S., Gomes, L. C. T., de Castro, L. N., Bezerra, G. B., & Von Zuben, F. J. (2004). An immune-evolutionary algorithm for multiple rearrangements of gene expression data. *Genetic Programming and Evolvable Machines*, 5(2), 157–179.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002b). Scalable multi-objective optimization test problems. Em *Proc. of the 2002 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 825–830).
- Deb, K., & Tiwari, S. (2005). Omni-optimizer: A procedure for single and multi-objective optimization. Em C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 3rd. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 3410 de *Lecture Notes in Computer Science*, (pp. 47–61). Springer.
- Detours, V., Bersini, H., Stewart, J., & Varela, F. J. (1994). Development of an idiotypic network in shape space. *Journal of Theoretical Biology*, 170(4), 401–414.
- Detours, V., Sulzer, B., & Perelson, A. S. (1996). Size and connectivity of the idiotypic network are independent of the discreteness of the affinity distribution. *Journal of Theoretical Biology*, 183(4), 409–416.
- Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.
- Edgeworth, F. Y. (1881). *Mathematical Physics*. London, England: P. Keagan.
- Ehrgott, M. (2005). *Multicriteria Optimization*. Springer.
- Emmerich, M. T. M., Beume, N., & Naujoks, B. (2005). An EMO algorithm using the hypervolume measure as selection criterion. Em C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 3rd. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 3410 de *LNCS*, (pp. 62–76). Springer.
- Emmerich, M. T. M., & Deutz, A. H. (2007). Test problems based on Lamé superspheres. Em S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 4th. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 4403 de *LNCS*, (pp. 922–936). Springer.
- Esquivel, S. C., & Coello Coello, C. A. (2003). On the use of particle swarm optimization with multimodal functions. Em *Proc. of the 2003 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1130–1136).
- Farmer, J. D., Packard, N., & Perelson, A. S. (1986). The immune system, adaptation and machine learning. *Physica D*, 2(1–3), 187–204.

- Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994). Self-nonsel self discrimination in a computer. Em *Proc. of the 1994 IEEE Symposium on Research in Security and Privacy*, (pp. 202–212).
- Freschi, F., Coello Coello, C. A., & Repetto, M. (2009). Multiobjective optimization and artificial immune systems: A review. Em H. Mo (Ed.) *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies*. IGI.
- Freschi, F., & Repetto, M. (2006). VIS: An artificial immune network for multi-objective optimization. *Engineering Optimization*, 38(8), 975–996.
- Gao, J., & Wang, J. (2010). WBMOAIS: A novel artificial immune system for multiobjective optimization. *Computers & Operations Research*, 37, 50–61.
- Garcia-Martinez, C., & Lozano, M. (2005a). Chromosome differentiation for the application of parent-centric real-parameter crossover operators. Relatório técnico, University of Granada, Spain.
- Garcia-Martinez, C., & Lozano, M. (2005b). Hybrid real-coded genetic algorithms with female and male differentiation. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 896–903).
- Gaspar, A., & Collard, P. (2000). Two models of immunization for time dependent optimization. Em *Proc. of the 2000 IEEE System, Man and Cybernetics Conference*, (pp. 113–118).
- Gibert, C. J., & Routen, T. W. (1994). Associative memory in an immune-based system. Em *Proc. of the 12th. National Conference on Artificial Intelligence*, (pp. 852–857).
- Gomes, L. C. T., de Sousa, J. S., Bezerra, G. B., de Castro, L. N., & Von Zuben, F. J. (2004). copt-aiNet and the gene ordering problem. *Revista Tecnologia da Informação*, 3(2), 27–34.
- Greensmith, J. (2007). *The Dendritic Cell Algorithm*. Tese de Doutorado, University of Nottingham.
- Greensmith, J., & Aickelin, U. (2008). The deterministic dendritic cell algorithm. Em P. J. Bentley, D. Lee, & S. Jung (Eds.) *Artificial Immune Systems - Proc. of the 7th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 5132 de *Lecture Notes in Computer Science*, (pp. 291–302). Springer Berlin / Heidelberg.
- Gregory, M. D., & Werner, D. H. (2010). Design of high performance compact linear ultra-wideband arrays with the CMA evolutionary strategy. Em *Proc. of the 2010 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, (pp. 1–4).

- Hajela, P., & Yoo, J. S. (1999). Immune network modelling in design optimization. Em D. W. Corne, M. Dorigo, & F. Glover (Eds.) *New Ideas in Optimization*, (pp. 203–215). McGraw Hill.
- Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. Em *Proc. of 1996 IEEE International Conference on Evolutionary Computation (ICEC)*, (pp. 312–317).
- Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. Em *Proc. of the 6th International Conference on Genetic Algorithms*, (pp. 24–31). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Hart, E. (2005). Not all balls are round: An investigation of alternative recognition-region shapes. Em C. Jacob, M. Pilat, P. J. Bentley, & J. Timmis (Eds.) *Artificial Immune Systems – Proc. of the 4th International Conference on Artificial Immune Systems (ICARIS)*, vol. 3627 de *Lecture Notes in Computer Science*, (pp. 29–42). Springer Berlin / Heidelberg.
- Hart, E. (2006). Analysis of a growth model for idiotypic networks. Em H. Bersini, & J. Carneiro (Eds.) *Artificial Immune Systems – Proc. of the 5th. International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 66–80). Springer Berlin / Heidelberg.
- Hart, E., Bersini, H., & Santos, F. C. (2006). Tolerance vs intolerance: How affinity defines topology in an idiotypic network. Em H. Bersini, & J. Carneiro (Eds.) *Artificial Immune Systems – Proc. of the 5th International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 109–121). Springer Berlin / Heidelberg.
- Hart, E., Bersini, H., & Santos, F. C. (2007). How affinity influences tolerance in an idiotypic network. *Journal of Theoretical Biology*, 249(3), 422–436.
- Hart, E., & Ross, P. (2005). The impact of the shape of antibody recognition regions on the emergence of idiotypic networks. *International Journal of Unconventional Computing*, 1, 281–313.
- Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation*, 7(1), 45–68.
- Holland, J. H. (1984). Genetic algorithms and adaptation. Em O. G. Selfridge, E. L. Rissland, & M. A. Arbib (Eds.) *Adaptive Control of Ill-Defined Systems*, (pp. 317–333). Plenum Press.
- Horst, R., & Pardalos, P. M. (Eds.) (1995). *Handbook of Global Optimization*, vol. 2 de *Nonconvex Optimization and Its Applications*. Springer.

- Hunt, J. E., & Cooke, D. E. (1996). Learning using an artificial immune system. *Journal of Network and Computer Applications*, 19(2), 189–212.
- Ishiguro, A., Ichikawa, S., Shibata, T., & Uchikawa, Y. (1998). Moderationism in the immune system: Gait acquisition of a legged robot using the metadynamics function. Em *Proc. of the 1998 IEEE System, Man, and Cybernetics Conference*, (pp. 3827–3832).
- Ishiguro, A., Kondo, T., Watanabe, Y., Shirai, Y., & Uchikawa, Y. (1997). Emergent construction of artificial immune networks for autonomous mobile robots. Em *Proc. of the 1997 IEEE International Conference on Systems, Man, and Cybernetics – Computational Cybernetics and Simulation*, vol. 2, (pp. 1222–1228).
- Jebalia, M., Auger, A., Schoenauer, M., James, F., & Postel, M. (2007). Identification of the isotherm function in chromatography using CMA-ES. Em *Proc. of the 2007 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 4289–4296).
- Jerne, N. K. (1974). Towards a network theory of the immune system. *Annales d'immunologie*, 125(1–2), 373–389.
- Jiao, L., Gong, M., Shang, R., Du, H., & Lu, B. (2005). Clonal selection with immune dominance and anergy based multiobjective optimization. Em C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 3rd. International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, vol. 3410 de *Lecture Notes in Computer Science*, (pp. 474–489). Springer Berlin / Heidelberg.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. NY, USA: Springer, 2 ed.
- Junqueira, C., de França, F. O., Attux, R. R. F., Panazio, C. M., de Castro, L. N., Von Zuben, F. J., & Romano, J. M. T. (2006). Immune-inspired dynamic optimization for blind spatial equalization in undermodeled channels. Em *Proc. of the 2006 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 9801–9808).
- Junqueira, C., de França, F. O., Attux, R. R. F., Suyama, R., de Castro, L. N., Von Zuben, F. J., & Romano, J. M. T. (2005). A proposal for blind FIR equalization of time-varying channels. Em *Proc. of the 2005 IEEE International Workshop on Machine Learning for Signal Processing*, (pp. 9–14).
- Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm Intelligence*. The Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann.

- Kennedy, J., & Mendes, R. (2002). Population structure and particle swarm performance. Em *Proc. of the 2002 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1671–1676).
- Kephart, J. O. (1994). A biologically inspired immune system for computers. Em R. A. Brooks, & P. Maes (Eds.) *Artificial Life IV – Proc. of the 4th. International Workshop on the Synthesis and Simulation of Living Systems*, (pp. 130–139). MIT Press.
- Kim, J., & Bentley, P. J. (1999). The human immune system and network intrusion detection. Em *Proc. of the 7th. European Congress on Intelligent Techniques and Soft Computing*.
- Knidel, H. (2010). *Extensões e Aplicações de Redes Neuro-Imunológicas*. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas, Brasil.
- Knight, T., & Timmis, J. (2001). AINE: an immunological approach to data mining. Em *Proc. of the 2001 IEEE International Conference on Data Mining (ICDM)*, (pp. 297–304).
- Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2), 149–172.
- Krohling, R. A., & Coelho, L. S. (2006). Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(6), 1407–1416.
- Kruisselbrink, J. W., Emmerich, M. T. M., Deutz, A. H., & Bäck, T. (2010). A robust optimization approach using Kriging metamodels for robustness approximation in the CMA-ES. Em *Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC)*.
- Lanaridis, A., & Stafylopatis, A. (2010). An artificial immune network for multi-objective optimization. Em K. Diamantaras, W. Duch, & L. Iliadis (Eds.) *Artificial Neural Networks – Proc. of the 2010 International Conference on Artificial Neural Networks (ICANN)*, vol. 6353 de *Lecture Notes in Computer Science*, (pp. 531–536). Springer Berlin / Heidelberg.
- Larrañaga, P., & Lozano, J. A. (Eds.) (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers.
- Lee, D.-W., & Sim, K.-B. (1997). Artificial immune network-based cooperative control in collective autonomous mobile robots. Em *Proc. of the 1997 IEEE International Workshop on Robotics and Human Communications*, (pp. 58–63).

- Leon, L. R. (2002). Cytokine regulation of fever: Studies using gene knockout mice. *Journal of Applied Physiology*, 92(6), 2648–2655.
- Li, C., Ma, T., & Zhu, X. (2010). aiNet- and GIS-based regional prediction system for the spatial and temporal probability of rainfall-triggered landslides. *Natural Hazards*, 52(1), 57–78.
- Li, M., Wang, N., Du, H., Zhuang, J., & Wang, S. (2006). Art-artificial immune network and application in fault diagnosis of the reciprocating compressor. Em L. Jiao, L. Wang, X. Gao, J. Liu, & F. Wu (Eds.) *Advances in Natural Computation – Proc. of the 2nd International Conference on Natural Computation (ICNC)*, vol. 4222 de *Lecture Notes in Computer Science*, (pp. 502–505). Springer Berlin / Heidelberg.
- Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3), 281–295.
- Liang, J. J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer with local search. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 522–528).
- López Jaimes, A., & Coello Coello, C. A. (2009). Study of preference relations in many-objective optimization. Em *Proc. of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, (pp. 611–618). ACM.
- Lu, B., Jiao, L., Du, H., & Gong, M. (2005). IFMOA: Immune forgetting multiobjective optimization algorithm. Em *Proc. of the 1st. International Conference on Natural Computation (ICNC)*, (pp. 399–408). Changsha, China.
- Luenberger, D. G. (Ed.) (1984). *Linear and Nonlinear Programming*. Addison–Wesley, 2 ed.
- Martins, F., & Slani, N. (2005). Computing with idiotypic networks. Em M. Klopotek, S. Wierchón, & K. Trojanowski (Eds.) *Intelligent Information Processing and Web Mining – Proc. of the 2005 International Intelligent Information Systems Conference (IIS)*, vol. 31 de *Advances in Soft Computing*, (pp. 673–684). Springer Berlin / Heidelberg.
- Matzinger, P. (1994). Tolerance, danger and the extended family. *Annual Review of Immunology*, 12(1), 991–1045.
- McCoy, D. F., & Devarajan, V. (1997). Artificial immune systems and aerial image segmentation. Em *Proc. of the IEEE System, Man, and Cybernetics Conference*, (pp. 867–872). Orlando, USA.

- Meiyi, L., Qiong, W., & Wei, Y. (2009). The K-CMA algorithm for solving multi-modal function optimization problems. Em *Proc. of the 2009 WRI Global Congress on Intelligent Systems (GCIS)*, vol. 2, (pp. 89–93).
- Michelan, R., & Von Zuben, F. J. (2002). Decentralized control system for autonomous navigation based on an evolved artificial immune network. Em *Proc. of the 2002 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 1021–1026).
- Miettinen, K. M. (1999). *Nonlinear Multiobjective Optimization*. Boston, USA: Kluwer Academic Publishers.
- Mitsumoto, N., Fukuda, T., Shimojima, K., & Ogawa, A. (1996). Self-organizing multiple robotic system (a population control through biologically inspired immune network architecture). Em *Proc. of the 1996 IEEE International Conference on Robotics and Automation*, (pp. 1614–1619).
- Molina, D., Herrera, F., & Lozano, M. (2005). Adaptive local search parameters for real-coded memetic algorithms. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 888–895).
- Moore, D. S., McCabe, G. P., & Craig, B. (2007). *Introduction to the Practice of Statistics*. W. H. Freeman, 6 ed.
- Moore, R., & Bierbaum, F. (1979). *Methods and applications of interval analysis*. SIAM Studies in Applied Mathematics. SIAM – Society for Industrial and Applied Mathematics.
- Mori, N., Kita, H., & Nishikawa, Y. (1996). Adaptation to a changing environment by means of the thermodynamical genetic algorithm. Em H.-M. Voigt, W. Ebeling, I. Rechenberg, & H.-P. Schwefel (Eds.) *Parallel Problem Solving from Nature – PPSN IV*, vol. 1141 de *Lecture Notes in Computer Science*, (pp. 513–522). Springer Berlin / Heidelberg.
- Mori, N., Kita, H., & Nishikawa, Y. (1998). Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm. Em A. E. Eiben, T. Bäck, M. Schoenauer, & H.-P. Schwefel (Eds.) *Parallel Problem Solving from Nature – PPSN V*, vol. 1498 de *Lecture Notes in Computer Science*, (pp. 149–158). Springer Berlin / Heidelberg.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts – Towards memetic algorithms. Relatório Técnico C3P – 826, California Institute of Technology, USA.
- Neumaier, A. (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13, 271–369.

- Nocedal, J., & Wright, S. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2 ed.
- Ootsuki, J. T., & Sekiguchi, T. (1999). Application of the immune system network concept to sequential control. Em *Proc. of the 1999 IEEE System, Man, and Cybernetics Conference*, vol. 3, (pp. 869–874).
- Paredis, J. (1995). Coevolutionary computation. *Artificial Life*, 2, 355–375.
- Pareto, V. (1896). *Cours D'Economie Politique*. Lausanne, Switzerland: F. Rouge.
- Perelson, A. S., & Oster, G. F. (1979). Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonsel self discrimination. *J. Theor. Biol.*, 81, 645–670.
- Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. Em *Proc. of the IEEE International Conference on Evolutionary Computation*, (pp. 798–803).
- Pinchuk, G. (Ed.) (2002). *Theory and problems of immunology*. McGraw-Hill.
- Pošík, P. (2005). Real-parameter optimization using the mutation step co-evolution. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 872–879).
- Preuss, M., Naujoks, B., & Rudolph, G. (2006). Pareto set and EMOA behavior for simple multimodal multiobjective functions. Em T. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. Whitley, & X. Yao (Eds.) *Parallel Problem Solving from Nature – PPSN IX*, vol. 4193 de *Lecture Notes in Computer Science*, (pp. 513–522). Springer.
- Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag.
- Qin, A. K., & Suganthan, P. N. (2005). Self-adaptive differential evolution algorithm for numerical optimization. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1785–1791).
- Rönkkönen, J., Kukkonen, S., & Price, K. V. (2005). Real-parameter optimization with differential evolution. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 506–513).
- Rogers, A., & Prügél-Bennett, A. (1999). Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, 3(4), 298–303.

- Rudolph, G., Naujoks, B., & Preuss, M. (2007). Capabilities of EMOA to detect and preserve equivalent Pareto subsets. Em S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 4th. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 4403 de *Lecture Notes in Computer Science*, (pp. 36–50). Springer.
- Sahan, S., Kodaz, H., Güneş, S., & Polat, K. (2004). A new classifier based on attribute weighted artificial immune system (AWAIS). Em C. Aykanat, T. Dayar, & I. Körpeoglu (Eds.) *Computer and Information Sciences - Proc. of the 2004 International Symposium on Computer and Information Sciences (ISCIS)*, vol. 3280 de *Lecture Notes in Computer Science*, (pp. 11–20). Springer Berlin / Heidelberg.
- Sareni, B., & Krähenbühl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3), 97–106.
- Sato, H. I., Aguirre, H. E., & Tanaka, K. (2007). Controlling dominance area of solutions and its impact on the performance of MOEAs. Em S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 4th. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 4403 de *Lecture Notes in Computer Science*, (pp. 5–20). Springer Berlin / Heidelberg.
- Schmidtchen, H., & Behn, U. (2006). Randomly evolving idiotypic networks: Analysis of building principles. Em H. Bersini, & J. Carneiro (Eds.) *Artificial Immune Systems – Proc. of the 5th International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 81–94). Springer Berlin / Heidelberg.
- Shang, R., & Ma, W. (2006). Immune clonal MO algorithm for ZDT problems. Em *Proc. of the 2nd. International Conference on Natural Computation (ICNC)*, (pp. 100–109). Xi'an, China.
- Shir, O. M., Emmerich, M. T. M., & Bäck, T. (2007). Self-adaptive niching CMA-ES with Mahalanobis metric. Em *Proc. of the 2007 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 820–827).
- Shir, O. M., Preuss, M., Naujoks, B., & Emmerich, M. T. M. (2009). Enhancing decision space diversity in evolutionary multiobjective algorithms. Em M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, & M. Sevaux (Eds.) *Evolutionary Multi-Criterion Optimization – Proc. of the 5th. Evolutionary Multi-Criterion Optimization (EMO)*, vol. 5467 de *Lecture Notes in Computer Science*, (pp. 95–109). Springer.

- Shomooka, T., Kikuchi, Y., & Shimizu, K. (2002). Idiotypic network model for feature extraction in pattern recognition. Em E. Damiani, R. J. Howlett, L. C. Jain, & N. Ichalkaranje (Eds.) *Knowledge-based Intelligent Information Engineering Systems and Allied Technologies – Proc. of the 6th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES)*, vol. 82 de *Frontiers in Artificial Intelligence and Applications*, (pp. 734–738). IOS Press Amsterdam.
- Sinha, A., Tiwari, S., & Deb, K. (2005). A population-based, steady-state procedure for real-parameter optimization. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, (pp. 514–521).
- Smit, S. K., & Eiben, A. E. (2009). Comparing parameter tuning methods for evolutionary algorithms. Em *Proc. of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 399–406).
- Somayagi, A., Hofmeyr, S. A., & Forrest, S. (1998). Principles of a computer immune system. Em *Proc. of the 1997 Workshop on New Security Paradigms*, (pp. 75–82).
- Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Relatório técnico, Nanyang Technological University, Singapore.
- Tan, K. C., Goh, C. K., Mamun, A. A., & Ei, E. Z. (2008). An evolutionary artificial immune system for multi-objective optimization. *European Journal of Operational Research*, 187, 371–392.
- Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y.-P., Chen, C. M., & Yang, Z. (2007). Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Relatório técnico, Nature Inspired Computation and Appl. Lab., USTC, China.
- Tarakanov, A., Sokolova, S., Abramov, B., & Aikimbayev, A. (2000). Immunocomputing of the natural plague foci. Em *Proc. of the 2000 Genetic and Evolutionary Computation Conference (GECCO)*, (pp. 38–39).
- Timmis, J., & Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge-Based Systems*, 14(3–4), 121–130.
- Timmis, J., Neal, M., & Hunt, J. E. (2000). An artificial immune system for data analysis. *BioSystems*, 55(1–3), 143–150.

- Toffolo, A., & Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, 11(2), 151–167.
- Toma, N., Endo, S., & Yamada, K. (1999). Immune algorithm with immune network and MHC for adaptive problem solving. Em *Proc. of the 1999 IEEE System, Man, and Cybernetics Conference*, vol. 4, (pp. 271–276). Tokyo, Japan.
- Trojanowski, K., & Sasin, M. (2006). The idiotypic network with binary patterns matching. Em H. Bersini, & J. Carneiro (Eds.) *Artificial Immune Systems – Proc. of the 5th International Conference on Artificial Immune Systems (ICARIS)*, vol. 4163 de *Lecture Notes in Computer Science*, (pp. 95–108). Springer Berlin / Heidelberg.
- Törrönen, K., Wallin, K., Saario, T., Hänninen, H., Rintamaa, R., & Forstén, J. (1985). Optimization of metallurgical variables in fracture prevention. *Nuclear Engineering and Design*, 87, 225–237.
- Tseng, L. Y., & Chen, C. (2008). Multiple trajectory search for large scale global optimization. Em *Proc. of the 2008 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 3052–3059).
- Unrich, T., Bader, J., & Zitzler, E. (2010). Integrating decision space diversity into hypervolume-based multiobjective search. Em *Proc. of the 2010 Genetic and Evolutionary Computation Conference – (GECCO)*.
- Voss, T., Beume, N., Rudolph, G., & Igel, C. (2008). Scalarization versus indicator-based selection in multi-objective CMA evolution strategies. Em *Proc. of the 2008 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 3036–3043).
- Walker, B. (2001). Ecosystems and immune systems: Useful analogy or stretching a metaphor. *Conservation Ecology*, 5(1), 16.
- Watkins, A. (2001). *AIRS: A Resource Limited Artificial Immune Classifier*. Dissertação de Mestrado, Mississippi State University, Mississippi, EUA.
- Whitbrook, A., Aickelin, U., & Garibaldi, J. (2008). An idiotypic immune network as a short-term learning architecture for mobile robots. Em P. J. Bentley, D. Lee, & S. Jung (Eds.) *Artificial Immune Systems – Proc. of the 7th International Conference on Artificial Immune Systems (ICARIS)*, vol. 5132 de *Lecture Notes in Computer Science*, (pp. 266–278). Springer Berlin / Heidelberg.
- Xu, L., Mo, H., Wang, K., & Tang, N. (2006). Document clustering based on modified artificial immune network. Em G. Wang, J. Peters, A. Skowron, & Y. Yao (Eds.) *Rough Sets and Knowledge*

- Technology – Proc. of the 1st. International Conference on Rough Sets and Knowledge Technology (RSKT)*, vol. 4062 de *Lecture Notes in Computer Science*, (pp. 516–521). Springer Berlin / Heidelberg.
- Yu, R., & Mao, Z.-Y. (2010). A new idiotypic immune network based PID controller design. Em *Proc. of the 8th World Congress on Intelligent Control and Automation (WCICA)*, (pp. 2486–2491).
- Yuan, B., & Gallagher, M. (2005). Experimental results for the special session on real-parameter optimization at CEC 2005: A simple, continuous EDA. Em *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, (pp. 1792–1799).
- Yue, X., Abraham, A., Chi, Z.-X., Hao, Y.-Y., & Mo, H. (2007). Artificial immune system inspired behavior-based anti-spam filter. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 11(8), 729–740.
- Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6).
- Zhou, A., Zhang, Q., & Jin, Y. (2009). Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5), 1167–1189.
- Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization*. Tese de Doutorado, Swiss Federal Institute of Technology, Zürich, Switzerland.
- Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8, 173–195.
- Zitzler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm. Em *Proc. of the 2001 EUROGEN Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, (pp. 95–100).
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271.

# Índice Remissivo de Autores

- Abbass, H. A. 12, 20, 23, 27, 37, 214  
Abbattista, F. 20, 207  
Abraham, A. 20, 226  
Abramov, B. 19, 224  
Agarwal, S. 118, 121, 122, 125, 131, 133–135, 138, 140, 153, 164, 166, 172, 214  
Aguirre, H. E. 134, 165, 200, 223  
Aickelin, U. 13, 19, 35, 207, 216, 225  
Aikimbayev, A. 19, 224  
Aisu, H. 19, 207  
Allen, L. 11, 216  
Arbib, M. A. 11, 217  
Attux, R. R. F. 12, 37, 218  
Auger, A. 67, 68, 99, 100, 170, 177, 207, 218, 224  
Aykanat, C. 20, 223  
  
Bader, J. 122, 225  
Bakhouya, M. 35, 207  
Ballester, P. J. 100, 207  
Baskar, S. 6, 39, 220  
Bauerle, N. 1, 208  
Bazaraa, M. S. 71, 208  
Bäck, T. 5, 6, 25, 38, 39, 100, 208, 219, 221, 223  
Behn, U. 18, 31, 32, 208, 209, 223  
Bell, G. I. 11, 12, 16, 25, 37, 49, 57, 69, 70, 121, 135, 169, 209  
  
Benini, E. 122, 225  
Bentley, P. J. 13, 19, 20, 31, 35, 38, 42, 121, 207, 209, 210, 216, 217, 219, 225  
Bersini, H. xxiii, 11, 20, 30–32, 42–47, 64, 169, 170, 208, 215, 217, 223, 225  
Beume, N. 100, 132, 195, 215, 225  
Beyer, H.-G. 122, 132, 196, 222  
Bezerra, G. B. 12, 20, 23, 37, 208, 214, 216  
Bierbaum, F. 41, 221  
Bomze, I. M. 41, 212  
Brede, M. 31, 32, 209  
Brooks, R. A. 19, 219  
Burke, E. 122, 132, 196, 222  
Burnet, F. M. 11, 12, 16, 25, 37, 49, 57, 69, 70, 121, 135, 169, 209  
  
Caetano, M. F. 20, 209  
Campelo, F. 121, 122, 209  
Cao, Y. 19, 213  
Carneiro, J. 11, 31, 32, 42, 217, 223, 225  
Carter, J. H. 19, 209  
Carter, J. N. 100, 207  
Carvalho, D. R. 19, 209  
Castro, P. A. D. 6, 7, 12, 18, 20, 26, 35–38, 53, 115, 121, 133, 169, 175, 209, 210, 214  
Cayzer, S. 13, 207  
Cazangi, R. R. 20, 210

- Chan, K. P. 122, 134, 135, 140, 153, 166, 172, 210
- Chen, C. 57, 96, 97, 128, 225
- Chen, C. M. 57, 224
- Chen, Y.-P. 57, 67, 68, 170, 177, 224
- Cherukuri, R. 11, 216
- Chi, Z.-X. 20, 226
- Chun-Kit, A. 100, 210
- Coelho, G. P. 6, 7, 12, 20, 23, 25, 26, 35–37, 53, 71, 115, 121, 169, 173, 209–211, 214
- Coelho, L. S. 6, 40, 212, 219
- Coello Coello, C. A. 6, 39, 115, 117, 121, 122, 132–134, 136, 137, 140, 153, 166, 172, 195, 212, 215, 216, 218, 220
- Collard, P. 20, 216
- Colutto, S. 100, 212
- Cooke, D. E. 20, 218
- Corne, D. W. 20, 121, 212, 217, 219
- Craig, B. 73, 74, 139, 221
- Cruz Cortés, N. 121, 212
- Csendes, T. 41, 212
- Dallwig, S. 41, 212
- Damiani, E. 29, 35, 223
- Darwin, C. 25, 213
- Dasgupta, D. 11, 12, 19, 20, 208, 213
- Dayar, T. 20, 223
- De Boer, R. J. 11, 29–31, 44, 213
- de Carvalho, A. B. 134, 213
- de Castro, L. N. 3, 5–7, 11, 12, 14, 18, 20, 22, 23, 27, 29, 36, 37, 49, 54, 69–72, 169, 170, 208, 213, 214, 216, 218
- de Deus Jr., G. A. 29, 214
- de França, F. O. 6, 7, 12, 20, 23, 26, 35–37, 53, 69–72, 115, 169, 170, 173, 209–211, 214, 218
- de Sousa, J. S. 12, 20, 23, 37, 214, 216
- Deb, K. 67, 68, 100, 115, 117, 118, 120–122, 125, 131–135, 138, 140, 152, 153, 163–166, 170, 172, 177, 195, 196, 199, 200, 209, 211, 212, 214, 215, 223, 224, 226
- Detours, V. 11, 30, 42, 215
- Deutz, A. H. 100, 133, 196, 215, 219
- Devarajan, V. 19, 220
- Di Gioia, G. 20, 207
- Di Santo, G. 20, 207
- Diamantarás, K. 121, 219
- Dorigo, M. 20, 38, 215, 217
- Du, H. 19, 121, 218, 220
- Duch, W. 121, 219
- Ebeling, W. 6, 39, 221
- Eberhart, R. C. 5, 6, 38, 39, 101, 218
- Edgeworth, F. Y. 118, 215
- Ehrgott, M. 121, 123, 215, 223
- Ei, E. Z. 121, 174, 224
- Eiben, A. E. 6, 39, 174, 221, 224
- Emmerich, M. T. M. 100, 123, 132, 133, 195, 196, 215, 219, 223
- Endo, S. 20, 225
- Esquivel, S. C. 6, 39, 215
- Farinelli, A. M. 20, 207
- Farmer, J. D. 11, 28, 29, 215
- Ferreira, H. M. 12, 20, 209, 210
- Fogel, D. B. 5, 6, 25, 38, 100, 208
- Fonseca, C. M. 123, 223
- Forrest, S. 11, 12, 19, 213, 216, 217, 224
- Forstén, J. 1, 225
- Freitas, A. A. 19, 209
- Freschi, F. 20, 23, 37, 121, 122, 131, 133, 135, 166, 172, 216
- Fruhauf, F. 100, 212

- Fuchs, M. 100, 212  
Fukuda, T. 20, 221
- Gaber, J. 35, 207  
Gallagher, K. 100, 207  
Gallagher, M. 100, 226  
Gandibleux, X. 123, 223  
Gao, J. 121, 133, 216  
Gao, X. 19, 220  
Garcia-Martinez, C. 100, 216  
Garibaldi, J. 35, 225  
Gaspar, A. 20, 216  
Gibert, C. J. 20, 216  
Glover, F. 20, 217  
Güneş, S. 20, 223  
Goh, C. K. 121, 174, 224  
Gomes, L. C. T. 12, 20, 23, 37, 214, 216  
Gong, M. 121, 218, 220  
Greco, S. 115, 211  
Greensmith, J. 12, 19, 216  
Gregory, M. D. 100, 216  
Guimarães, F. 121, 122, 209
- Hajela, P. 20, 217  
Hansen, N. 67, 68, 99, 100, 123, 170, 177, 207, 217, 224  
Hao, J.-K. 123, 223  
Hao, Y.-Y. 20, 226  
Harik, G. R. 39, 217  
Hart, E. 11, 31, 38, 42, 53, 175, 210, 217  
Hernández Aguirre, A. 121, 122, 132–134, 140, 153, 166, 172, 195, 215, 218  
Herrera, F. 101, 221  
Hilaire, V. 35, 207  
Hiroyasu, T. 121, 122, 133, 134, 165, 196, 200, 209, 215, 223
- Hänninen, H. 1, 225  
Ho-Fung, L. 100, 210  
Hofmeyr, S. A. 19, 217, 224  
Holland, J. H. 11, 217  
Hone, A. 38, 53, 175, 210  
Horst, R. 40, 41, 212, 217  
Howlett, R. J. 29, 35, 223  
Hunt, J. E. 20, 218, 224
- Ichalkaranje, N. 29, 35, 223  
Ichikawa, S. 20, 218  
Igarashi, H. 121, 122, 209  
Igel, C. 100, 225  
Iliadis, L. 121, 219  
Ishiguro, A. 20, 35, 218
- Jacob, C. 20, 31, 42, 209, 217  
Jain, L. C. 29, 35, 223  
James, F. 100, 218  
Jebalia, M. 100, 218  
Jerne, N. K. 6, 7, 9, 11, 12, 17, 23, 25, 28, 29, 35, 37, 38, 49, 69, 70, 121, 135, 162, 169, 218  
Jerram, N. R. 121, 212  
Jiao, L. 19, 121, 218, 220  
Jin, Y. 122, 123, 133, 135, 149, 166, 172, 226  
Jolliffe, I. T. 135, 218  
Jung, S. 19, 35, 38, 121, 210, 216, 225  
Junqueira, C. 12, 37, 218
- Kennedy, J. 5, 6, 38, 39, 101, 218, 219  
Kephart, J. O. 19, 219  
Kikuchi, Y. 29, 35, 223  
Kim, J. 19, 219  
Kita, H. 6, 39, 221  
Klopotek, M. 29, 35, 220  
Knidel, H. 29, 219  
Knight, T. 12, 219

- Knowles, J. D. 121, 212, 219  
Kodaz, H. 20, 223  
Kondo, T. 35, 218  
Koukam, A. 35, 207  
Krähenbühl, L. 6, 7, 38, 122, 223  
Krohling, R. A. 6, 219  
Körpeoglu, I. 20, 223  
Kruisselbrink, J. W. 100, 219  
Kukkonen, S. 101, 222  
  
Lamont, G. B. 115, 117, 136, 137, 212  
Lampinen, J. A. 3, 38, 40, 101, 222  
Lanaridis, A. 121, 219  
Larrañaga, P. 100, 123, 135, 219  
Laumanns, M. 121, 125, 126, 131, 133, 134, 163, 166, 172, 195, 200, 215, 226  
Lee, D. 19, 35, 38, 121, 210, 216, 225  
Lee, D.-W. 20, 219  
Leon, L. R. 14, 220  
Li, C. 12, 20, 220  
Li, H. 121, 226  
Li, M. 19, 220  
Liang, J. J. 6, 39, 67, 68, 101, 102, 111, 170, 177, 220, 224  
Liu, J. 19, 220  
Lozano, J. A. 100, 123, 135, 219  
Lozano, M. 100, 101, 216, 221  
López Jaimes, A. 134, 220  
Lu, B. 121, 218, 220  
Luenberger, D. G. 4, 220  
Lutton, E. 121, 212  
  
Ma, T. 12, 20, 220  
Ma, W. 121, 223  
MacNish, C. 57, 224  
Maes, P. 19, 219  
  
Mamun, A. A. 121, 174, 224  
Mao, Z.-Y. 35, 226  
Mariani, V. C. 40, 212  
Martins, F. 29, 35, 220  
Matzinger, P. 13, 220  
McCabe, G. P. 73, 74, 139, 221  
McCoy, D. F. 19, 220  
McEwan, C. 38, 53, 175, 210  
Meiyi, L. 100, 221  
Mendes, R. 39, 219  
Merelo-Guervós, J. J. 121, 122, 132, 196, 212, 222  
Meyarivan, T. 118, 121, 122, 125, 131, 133–135, 138, 140, 153, 164, 166, 172, 214  
Michalewicz, Z. 5, 6, 25, 38, 100, 208  
Michelan, R. 20, 221  
Miettinen, K. M. 121, 221  
Mitsumoto, N. 20, 221  
Mitzutani, H. 19, 207  
Männer, R. 20, 208  
Mo, H. 20, 121, 122, 216, 225, 226  
Molina, D. 101, 221  
Moore, D. S. 73, 74, 139, 221  
Moore, R.E. 41, 221  
Mori, N. 6, 39, 221  
Moscato, P. 101, 221  
Murata, T. 121, 122, 133, 134, 165, 196, 200, 209, 215, 223  
  
Naujoks, B. 122, 123, 132, 195, 196, 215, 222, 223  
Neal, M. 12, 20, 224  
Neumaier, A. 5, 41, 212, 221  
Newton, C. S. 12, 20, 23, 27, 37, 214  
Nishikawa, Y. 6, 39, 221  
Nocedal, J. 1, 40, 222

- Oates, M. J. 121, 212
- Obayashi, S. 121, 122, 133, 134, 165, 196, 200, 209, 215, 223
- Ogawa, A. 20, 221
- Ootsuki, J. T. 20, 222
- Oster, G. F. 13, 20, 222
- Ostermeier, A. 99, 123, 217
- Packard, N. 11, 28, 29, 215
- Panazio, C. M. 12, 37, 218
- Pardalos, P. M. 40, 41, 212, 217
- Paredis, J. 101, 222
- Pareto, V. 118, 222
- Paton, R. 20, 208
- Perelson, A. S. 11–13, 16, 20, 25, 28–30, 37, 49, 57, 69, 70, 121, 135, 169, 209, 213, 215, 216, 222
- Peters, J. 20, 225
- Petrowski, A. 6, 38, 222
- Pilat, M. 20, 31, 42, 209, 217
- Pimgley Jr, G. H. 11, 12, 16, 25, 37, 49, 57, 69, 70, 121, 135, 169, 209
- Pinchuk, G. 13, 222
- Polat, K. 20, 223
- Poloni, C. 121, 122, 133, 134, 165, 196, 200, 209, 215, 223
- Pošík, P. 101, 222
- Postel, M. 100, 218
- Pozo, A. 134, 213
- Pratap, A. 118, 121, 122, 125, 131, 133–135, 138, 140, 153, 164, 166, 172, 214
- Preuss, M. 122, 123, 132, 196, 222, 223
- Price, K. V. 3, 38, 40, 101, 222
- Prügel-Bennett, A. 6, 222
- Qin, A. K. 6, 39, 101, 102, 111, 220, 222
- Qiong, W. 100, 221
- Ray, T. 122, 134, 135, 140, 153, 166, 172, 210
- Rechenberg, I. 6, 39, 221
- Repetto, M. 20, 23, 37, 121, 122, 131, 133, 135, 166, 172, 216
- Rieder, U. 1, 208
- Rintamaa, R. 1, 225
- Rissland, E. L. 11, 217
- Rönkkönen, J. 101, 222
- Rodriguez, S. 35, 207
- Rogers, A. 6, 222
- Romano, J. M. T. 12, 37, 218
- Ross, P. 31, 42, 217
- Routen, T. W. 20, 216
- Rudolph, G. 100, 121, 122, 132, 196, 212, 222, 223, 225
- Runarsson, T. 122, 132, 196, 222
- Saario, T. 1, 225
- Sagot, M.-F. 12, 20, 211
- Sahan, S. 20, 223
- Santos, F. C. 11, 31, 42, 217
- Sareni, B. 6, 7, 38, 122, 223
- Sarker, R. A. 12, 20, 23, 27, 37, 214
- Sasin, M. 32, 225
- Sato, H. I. 134, 165, 200, 223
- Scherzer, O. 100, 212
- Schichl, H. 41, 212
- Schmidtchen, H. 31, 32, 223
- Schoenauer, M. 6, 39, 100, 121, 212, 218, 221
- Schwefel, H.-P. 6, 20, 39, 121, 208, 212, 221
- Sekiguchi, T. 20, 222
- Selfridge, O. G. 11, 217
- Sevaux, M. 123, 223
- Shang, R. 121, 218, 223

- Sherali, H. D. 71, 208  
Shetty, C. M. 71, 208  
Shi, Y. 5, 6, 38, 39, 101, 218  
Shibata, T. 20, 218  
Shimizu, K. 29, 35, 223  
Shimajima, K. 20, 221  
Shir, O. M. 100, 123, 223  
Shirai, Y. 35, 218  
Shomooka, T. 29, 35, 223  
Silva, A. E. A. 12, 20, 71, 211  
Sim, K.-B. 20, 219  
Sinha, A. 100, 224  
Skowron, A. 20, 225  
Slani, N. 29, 35, 220  
Smit, S. K. 174, 224  
Sokolova, S. 19, 224  
Somayagi, A. 19, 224  
Souza, R. C. T. 40, 212  
Srinivas, N. 122, 224  
Stafylopatis, A. 121, 219  
Stephenson, J. 100, 207  
Stewart, J. 11, 30, 42, 215  
Storn, R. M. 3, 38, 40, 101, 222  
Stützle, T. 38, 215  
Suganthan, P. N. 6, 39, 57, 67, 68, 101, 102, 111, 170, 177, 220, 222, 224  
Sulzer, B. 30, 215  
Suyama, R. 12, 37, 218  
  
Takahashi, R. 115, 211  
Tan, K. C. 121, 174, 224  
Tanaka, K. 134, 165, 200, 223  
Tang, K. 57, 224  
Tang, N. 20, 225  
Tarakanov, A. 19, 224  
Thiele, L. 121, 125, 126, 131, 133, 134, 152, 163, 165, 166, 172, 195, 196, 200, 215, 226  
Timmis, J. 3, 5–7, 11–14, 18, 20, 22, 23, 31, 37, 38, 42, 49, 53, 54, 69, 70, 169, 170, 175, 207, 209, 210, 213, 217, 219, 224  
Tiwari, S. 67, 68, 100, 122, 132–134, 140, 153, 166, 170, 172, 177, 195, 215, 224  
Toffolo, A. 122, 225  
Toma, N. 20, 225  
Toscano Pulido, G. 121, 212  
Trojanowski, K. 29, 32, 35, 220, 225  
Törrönen, K. 1, 225  
Tseng, L. Y. 57, 96, 97, 128, 225  
  
Uchikawa, Y. 20, 35, 218  
Unrich, T. 122, 225  
  
Van Veldhuizen, D. A. 115, 117, 136, 137, 212  
Varela, F. J. 11, 20, 30, 42, 208, 215  
Voigt, H.-M. 6, 39, 221  
Von Zuben, F. J. 6, 11, 12, 20, 23, 25–27, 29, 35–38, 53, 69–72, 115, 121, 133, 169, 170, 173, 175, 209–211, 213, 214, 216, 218, 221  
Voss, T. 100, 225  
  
Walker, B. 20, 225  
Wallin, K. 1, 225  
Walter, M. 12, 20, 211  
Wang, G. 20, 225  
Wang, J. 121, 133, 216  
Wang, K. 20, 225  
Wang, L. 19, 220  
Wang, N. 19, 220  
Wang, S. 19, 220  
Wanner, E. 115, 211  
Watanabe, Y. 35, 218  
Watkins, A. 20, 225

- Wei, Y. 100, 221
- Werner, D. H. 100, 216
- Whitbrook, A. 35, 225
- Whitley, L. 122, 132, 196, 222
- Wierzchon, S. 29, 35, 220
- Wright, S. 1, 40, 222
- Wu, F. 19, 220
- Xu, L. 20, 225
- Yamada, K. 20, 225
- Yang, C. 19, 213
- Yang, Z. 57, 224
- Yao, X. 57, 121, 122, 132, 196, 212, 222, 224
- Yao, Y. 20, 225
- Yoo, J. S. 20, 217
- Yu, R. 35, 226
- Yuan, B. 100, 226
- Yue, X. 20, 226
- Zhang, Q. 121–123, 133, 135, 149, 166, 172, 226
- Zhou, A. 122, 123, 133, 135, 149, 166, 172, 226
- Zhu, X. 12, 20, 220
- Zhuang, J. 19, 220
- Zitzler, E. 121, 122, 125, 126, 131–134, 137,  
138, 140, 152, 153, 163, 165, 166, 172, 195,  
196, 200, 215, 218, 225, 226

