

UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

**Aprendizado Genético de Funções de
Pertinência na Modelagem Nebulosa**

Matheus Giovanni Pires

São Carlos
Agosto/2004

UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Aprendizado Genético de Funções de Pertinência na Modelagem Nebulosa

Matheus Giovanni Pires

Orientação:

Profa. Dra. Heloisa de Arruda Camargo

*Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos como requisito parcial exigido à obtenção do grau de **Mestre em Ciência da Computação**.*

São Carlos
Agosto/2004

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

P667ag

Pires, Matheus Giovanni.

Aprendizado genético de funções de pertinência na modelagem nebulosa / Matheus Giovanni Pires. -- São Carlos : UFSCar, 2005.

128 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2004.

1. Inteligência artificial. 2. Ajuste de funções de pertinência. 3. Projeto automático de sistemas nebulosos. 4. Aprendizado genético. 5. Sistema genético nebuloso. I. Título.

CDD: 006.3 (20^a)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

“Aprendizado Genético de Funções de Pertinência na Modelagem Nebulosa”

MATHEUS GIOVANNI PIRES

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:



Profa. Dra. Heloisa de Arruda Camargo
(Orientadora - DC/UFSCar)



Prof. Dr. Orides Morandin Junior
(DC/UFSCar)



Prof. Dr. Nelson Francisco Favilha Ebecken
(UFRJ)

São Carlos
Agosto/2004

À DEUS,
aos meus pais Germano e Aparecida,
e as minhas irmãs Marina e Marisa.

Agradeço

À Deus, por ter iluminado o meu caminho durante toda a trajetória deste trabalho, pelo qual enfrentei e venci vários obstáculos graças a sua ajuda.

À Profa. Heloisa de Arruda Camargo pelos seus ensinamentos e orientação, essenciais para o meu aprendizado e para o desenvolvimento deste projeto.

Aos meus pais e irmãs por terem sempre acreditado em mim, ao apoio financeiro de meu pai e, principalmente, pelas orações de minha mãe.

À minha namorada Fabiana, que apesar de tê-la conhecido após um ano de mestrado, sempre me ajudou com seu carinho, incentivo e com o seu Athlon XP ☺.

Aos meus amigos da graduação e do “AP” em São Carlos, Rodrigo (Plotze), Lucas, Thiago (Thiagão) e Jean, pelos momentos descontraídos e carregados com muitas gargalhadas, os quais nunca esquecerei.

Aos amigos da pós-graduação, e em especial a Taciana, Milena e Adriana, que ficaram marcados pelos nossos encontros na sala de estudos.

À Dona Ofélia e Jorgina (Dona Vera), que desde o primeiro dia de aula me recebeu com carinho e atenção, e que sempre torceram pelo sucesso dos alunos da pós-graduação.

Sumário

| | |
|--|-------------|
| LISTA DE FIGURAS | III |
| LISTA DE TABELAS..... | VIII |
| RESUMO | X |
| ABSTRACT | XI |
| CAPÍTULO 1 INTRODUÇÃO | 1 |
| 1.1 CONSIDERAÇÕES INICIAIS | 1 |
| 1.2 OBJETIVOS E MOTIVAÇÃO..... | 2 |
| 1.3 ORGANIZAÇÃO DA DISSERTAÇÃO | 4 |
| CAPÍTULO 2 COMPUTAÇÃO EVOLUTIVA | 5 |
| 2.1 CONSIDERAÇÕES INICIAIS | 5 |
| 2.2 UM BREVE HISTÓRICO | 5 |
| 2.3 PORQUE USAR COMPUTAÇÃO EVOLUTIVA | 7 |
| 2.4 ALGORITMOS GENÉTICOS | 8 |
| 2.4.1 <i>Terminologia Biológica</i> | 10 |
| 2.4.2 <i>Métodos de Seleção</i> | 11 |
| 2.4.3 <i>Operadores Genéticos</i> | 14 |
| 2.4.4 <i>Exemplo de Aplicação de um Algoritmo Genético</i> | 19 |
| 2.5 CONSIDERAÇÕES FINAIS | 24 |
| CAPÍTULO 3 SISTEMAS NEBULOSOS..... | 25 |
| 3.1 CONSIDERAÇÕES INICIAIS | 25 |
| 3.2 CONJUNTOS NEBULOSOS..... | 26 |
| 3.2.1 <i>Funções de Pertinência</i> | 27 |
| 3.2.2 <i>Definições Básicas em Conjuntos Nebulosos</i> | 31 |
| 3.2.3 <i>Operações Básicas com Conjuntos Nebulosos</i> | 33 |
| 3.3 VARIÁVEIS LINGÜÍSTICAS | 34 |
| 3.4 RACIOCÍNIO APROXIMADO..... | 36 |

| | |
|---|------------|
| 3.5 SISTEMAS NEBULOSOS | 38 |
| 3.5.1 <i>Sistema Nebuloso Mamdani</i> | 40 |
| 3.6 CONSIDERAÇÕES FINAIS | 43 |
| CAPÍTULO 4 APRENDIZADO GENÉTICO DA BC DE SISTEMAS NEBULOSOS..... | 44 |
| 4.1 CONSIDERAÇÕES INICIAIS | 44 |
| 4.2 ABORDAGENS GENÉTICAS NO PROJETO AUTOMÁTICO DE SISTEMAS NEBULOSOS | 46 |
| 4.3 APRENDIZADO GENÉTICO DE FUNÇÕES DE PERTINÊNCIA NA MODELAGEM NEBULOSA..... | 50 |
| 4.3.1 <i>Otimização Genética das Funções de Pertinência</i> | 51 |
| 4.3.2 <i>Construção Genética das Funções de Pertinência</i> | 52 |
| 4.3.3 <i>Representação</i> | 55 |
| 4.3.4 <i>População Inicial</i> | 56 |
| 4.3.5 <i>Operadores Genéticos</i> | 59 |
| 4.3.6 <i>Formato das Regras Nebulosas e Método de Raciocínio Nebuloso</i> | 59 |
| 4.3.7 <i>Função de Avaliação</i> | 60 |
| 4.3.8 <i>Outros Parâmetros</i> | 61 |
| 4.4 CONSIDERAÇÕES FINAIS | 62 |
| CAPÍTULO 5 EXPERIMENTOS E RESULTADOS | 63 |
| 5.1 CONSIDERAÇÕES INICIAIS | 63 |
| 5.2 DETALHAMENTO DOS CONJUNTOS DE DADOS..... | 64 |
| 5.3 OBJETIVOS DOS EXPERIMENTOS..... | 66 |
| 5.4 RESULTADOS PARA O CONJUNTO AUTO-MPG..... | 67 |
| 5.5 RESULTADOS PARA O CONJUNTO IRIS PLANTS..... | 71 |
| 5.6 RESULTADOS PARA O CONJUNTO WINE | 73 |
| 5.7 RESULTADOS PARA O CONJUNTO BOSTON HOUSING | 76 |
| 5.8 CONSIDERAÇÕES FINAIS | 80 |
| CAPÍTULO 6 CONCLUSÕES..... | 81 |
| APÊNDICE A MÉTODO DE APRENDIZADO DE WANG E MENDEL..... | 83 |
| APÊNDICE B RESULTADOS PARA O CONJUNTO DE DADOS AUTO-MPG..... | 85 |
| APÊNDICE C RESULTADOS PARA O CONJUNTO DE DADOS IRIS PLANTS..... | 95 |
| APÊNDICE D RESULTADOS PARA O CONJUNTO DE DADOS WINE..... | 102 |
| APÊNDICE E RESULTADOS PARA O CONJUNTO BOSTON HOUSING..... | 112 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 122 |

Lista de Figuras

| | |
|---|----|
| FIGURA 2.1: EXEMPLO DE UMA ROLETA DE SELEÇÃO..... | 12 |
| FIGURA 2.2: SELEÇÃO POR AMOSTRAGEM UNIVERSAL ESTOCÁSTICA..... | 13 |
| FIGURA 2.3: CRUZAMENTO DE UM-PONTO..... | 15 |
| FIGURA 2.4: CRUZAMENTO DE TRÊS-PONTOS..... | 15 |
| FIGURA 2.5: CRUZAMENTO UNIFORME..... | 16 |
| FIGURA 2.6: MUTAÇÃO SIMPLES..... | 16 |
| FIGURA 2.7: CRUZAMENTO SIMPLES..... | 17 |
| FIGURA 2.8: CRUZAMENTO ARITMÉTICO..... | 18 |
| FIGURA 2.9: CRUZAMENTO HEURÍSTICO..... | 18 |
| FIGURA 2.10: MUTAÇÃO UNIFORME..... | 18 |
| FIGURA 2.11: MUTAÇÃO LIMITE..... | 19 |
| FIGURA 2.12: GRÁFICO DA FUNÇÃO $F(x) = x \sin(10 \pi x) + 1$ | 20 |
| FIGURA 3.1: GRÁFICO DE UMA FUNÇÃO TRIANGULAR..... | 28 |
| FIGURA 3.2: GRÁFICO DE UMA FUNÇÃO Γ | 29 |
| FIGURA 3.3: GRÁFICO DE UMA FUNÇÃO S..... | 29 |
| FIGURA 3.4: GRÁFICO DE UMA FUNÇÃO TRAPEZOIDAL..... | 30 |
| FIGURA 3.5: GRÁFICO DE UMA FUNÇÃO GAUSSIANA..... | 31 |
| FIGURA 3.6: GRÁFICO DE UM CONJUNTO UNITÁRIO (SINGLETON)..... | 31 |
| FIGURA 3.7: EXEMPLOS DE PARTIÇÃO NEBULOSA DE DIFERENTES GRANULARIDADES..... | 35 |
| FIGURA 3.8: VARIÁVEL LINGÜÍSTICA VELOCIDADE..... | 36 |
| FIGURA 3.9: ESTRUTURA BÁSICA DE UM SISTEMA NEBULOSO..... | 39 |
| FIGURA 3.10: INFERÊNCIA NO MODELO MAMDANI..... | 42 |
| FIGURA 4.1: DERIVAÇÃO GENÉTICA DA BR..... | 47 |
| FIGURA 4.2: PROCESSO DE AJUSTE DE UMA BD..... | 48 |
| FIGURA 4.3: APRENDIZADO GENÉTICO DOS COMPONENTES DA BC SIMULTANEAMENTE..... | 49 |
| FIGURA 4.4: DEFINIÇÃO GENÉTICA DA BD..... | 49 |
| FIGURA 4.5: FLUXOGRAMA DO AG QUE REALIZA O AJUSTE DA BD..... | 52 |
| FIGURA 4.6: FLUXOGRAMA DO AG QUE REALIZA A CONSTRUÇÃO DA BD..... | 54 |
| FIGURA 4.7: REPRESENTAÇÃO CROMOSSÔMICA DA BD..... | 55 |
| FIGURA 4.8: PARÂMETROS DE UMA FUNÇÃO TRIANGULAR..... | 56 |

| | |
|---|----|
| FIGURA 4.9: INTERVALO DE VARIAÇÃO DOS PONTOS DAS FUNÇÕES DE PERTINÊNCIA. | 56 |
| FIGURA 4.10: PARTIÇÃO NEBULOSA UNIFORME DA VARIÁVEL V. | 57 |
| FIGURA 4.11: EXEMPLO DE PARTIÇÃO NEBULOSA GERADA A PARTIR DO CROMOSSOMO C(0). | 58 |
| FIGURA 5.1: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 8 VARIÁVEIS E GRANULARIDADE 3 (AUTO-MPG)..... | 68 |
| FIGURA 5.2: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD (AUTO-MPG)..... | 68 |
| FIGURA 5.3: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD (AUTO-MPG). | 69 |
| FIGURA 5.4: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD (AUTO-MPG)..... | 70 |
| FIGURA 5.5: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD (AUTO-MPG). | 70 |
| FIGURA 5.6: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM GRANULARIDADE 3 (IRIS PLANTS)..... | 72 |
| FIGURA 5.7: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD (IRIS PLANTS). | 72 |
| FIGURA 5.8: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD (IRIS PLANTS)..... | 73 |
| FIGURA 5.9: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 3 (WINE). | 74 |
| FIGURA 5.10: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 3 (WINE)..... | 74 |
| FIGURA 5.11: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 3 (WINE). | 75 |
| FIGURA 5.12: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD (WINE). | 75 |
| FIGURA 5.13: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD (WINE)..... | 76 |
| FIGURA 5.14: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 3 (BOSTON HOUSING)..... | 77 |
| FIGURA 5.15: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 3 (BOSTON HOUSING)..... | 78 |
| FIGURA 5.16: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 3 (BOSTON HOUSING)..... | 78 |
| FIGURA 5.17: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD (BOSTON HOUSING)..... | 79 |
| FIGURA 5.18: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD (BOSTON HOUSING)..... | 79 |
| FIGURA B.1: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 8 VARIÁVEIS E GRANULARIDADE 5 (AUTO-MPG)..... | 86 |
| FIGURA B.2: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 5 (AUTO-MPG). | 86 |

| | |
|---|-----|
| FIGURA B.3: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 5 (AUTO-MPG)..... | 87 |
| FIGURA B.4: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (AUTO-MPG)..... | 87 |
| FIGURA B.5: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (AUTO-MPG)..... | 88 |
| FIGURA B.6: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 8 VARIÁVEIS E GRANULARIDADE 7 (AUTO-MPG)..... | 89 |
| FIGURA B.7: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 7 (AUTO-MPG)..... | 89 |
| FIGURA B.8: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 7 (AUTO-MPG)..... | 90 |
| FIGURA B.9: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (AUTO-MPG)..... | 90 |
| FIGURA B.10: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (AUTO-MPG)..... | 91 |
| FIGURA B.11: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 8 VARIÁVEIS E GRANULARIDADE 9 (AUTO-MPG)..... | 92 |
| FIGURA B.12: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 9 (AUTO-MPG)..... | 92 |
| FIGURA B.13: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 8 VARIÁVEIS E GRANULARIDADE 9 (AUTO-MPG)..... | 93 |
| FIGURA B.14: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (AUTO-MPG)..... | 93 |
| FIGURA B.15: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (AUTO-MPG)..... | 94 |
| FIGURA C.1: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM GRANULARIDADE 5 (IRIS PLANTS)..... | 96 |
| FIGURA C.2: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (IRIS PLANTS)..... | 96 |
| FIGURA C.3: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (IRIS PLANTS)..... | 97 |
| FIGURA C.4: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM GRANULARIDADE 7 (IRIS PLANTS)..... | 98 |
| FIGURA C.5: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (IRIS PLANTS)..... | 98 |
| FIGURA C.6: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (IRIS PLANTS)..... | 99 |
| FIGURA C.7: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM GRANULARIDADE 9 (IRIS PLANTS)..... | 100 |

| | |
|---|-----|
| FIGURA C.8: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (IRIS PLANTS)..... | 100 |
| FIGURA C.9: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (IRIS PLANTS)..... | 101 |
| FIGURA D.1: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 5 (WINE)..... | 103 |
| FIGURA D.2: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 5 (WINE)..... | 103 |
| FIGURA D.3: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 5 (WINE)..... | 104 |
| FIGURA D.4: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (WINE)..... | 104 |
| FIGURA D.5: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (WINE)..... | 105 |
| FIGURA D.6: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 7 (WINE)..... | 106 |
| FIGURA D.7: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 7 (WINE)..... | 106 |
| FIGURA D.8: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 7 (WINE)..... | 107 |
| FIGURA D.9: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (WINE)..... | 107 |
| FIGURA D.10: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (WINE)..... | 108 |
| FIGURA D.11: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 9 (WINE)..... | 109 |
| FIGURA D.12: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 9 (WINE)..... | 109 |
| FIGURA D.13: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 9 (WINE)..... | 110 |
| FIGURA D.14: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (WINE)..... | 110 |
| FIGURA D.15: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (WINE)..... | 111 |
| FIGURA E.1: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 5 (BOSTON HOUSING)..... | 113 |
| FIGURA E.2: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 5 (BOSTON HOUSING)..... | 113 |
| FIGURA E.3: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 5 (BOSTON HOUSING)..... | 114 |

| | |
|--|-----|
| FIGURA E.4: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (BOSTON HOUSING). | 114 |
| FIGURA E.5: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (BOSTON HOUSING). | 115 |
| FIGURA E.6: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 7 (BOSTON HOUSING). | 116 |
| FIGURA E.7: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 7 (BOSTON HOUSING). | 116 |
| FIGURA E.8: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 7 (BOSTON HOUSING). | 117 |
| FIGURA E.9: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (BOSTON HOUSING). | 117 |
| FIGURA E.10: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (BOSTON HOUSING). | 118 |
| FIGURA E.11: COMPARAÇÃO ENTRE AS ABORDAGENS DO DESEMPENHO DO AG COM 14 VARIÁVEIS E GRANULARIDADE 9 (BOSTON HOUSING). | 119 |
| FIGURA E.12: DESEMPENHO DOS INDIVÍDUOS NA CONSTRUÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 9 (BOSTON HOUSING). | 119 |
| FIGURA E.13: DESEMPENHO DOS INDIVÍDUOS NA OTIMIZAÇÃO DA BD COM 14 VARIÁVEIS E GRANULARIDADE 9 (BOSTON HOUSING). | 120 |
| FIGURA E.14: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (BOSTON HOUSING). | 120 |
| FIGURA E.15: RELAÇÃO TAXA DE CLASSIFICAÇÃO VS. QUANTIDADE DE REGRAS NA OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (BOSTON HOUSING). | 121 |

Lista de Tabelas

| | |
|---|----|
| TABELA 2.1: ALGUNS RESULTADOS DE 150 GERAÇÕES..... | 23 |
| TABELA 4.1: VALORES DOS PARÂMETROS DAS FUNÇÕES DE PERTINÊNCIA DA VARIÁVEL V... 57 | 57 |
| TABELA 4.2: PARÂMETROS DO AG..... | 61 |
| TABELA 5.1: INFORMAÇÕES DOS ATRIBUTOS DO CONJUNTO DE DADOS AUTO-MPG..... | 64 |
| TABELA 5.2: INFORMAÇÕES DOS ATRIBUTOS DO CONJUNTO DE DADOS BOSTON HOUSING. | 64 |
| TABELA 5.3: INFORMAÇÕES DOS ATRIBUTOS DO CONJUNTO DE DADOS WINE. | 65 |
| TABELA 5.4: INFORMAÇÕES DOS ATRIBUTOS DO CONJUNTO DE DADOS ÍRIS. | 66 |
| TABELA 5.5: RESULTADOS DA CONSTRUÇÃO GENÉTICA DA BD (AUTO-MPG). | 67 |
| TABELA 5.6: RESULTADOS DA OTIMIZAÇÃO GENÉTICA DA BD (AUTO-MPG)..... | 67 |
| TABELA 5.7: RESULTADOS DA CONSTRUÇÃO GENÉTICA DA BD (IRIS PLANTS). | 71 |
| TABELA 5.8: RESULTADOS DA OTIMIZAÇÃO GENÉTICA DA BD (IRIS PLANTS). | 71 |
| TABELA 5.9: RESULTADOS DA CONSTRUÇÃO GENÉTICA DA BD (WINE)..... | 73 |
| TABELA 5.10: RESULTADOS DA OTIMIZAÇÃO GENÉTICA DA BD (WINE). | 73 |
| TABELA 5.11: RESULTADOS DA CONSTRUÇÃO GENÉTICA DA BD (BOSTON HOUSING). | 76 |
| TABELA 5.12: RESULTADOS DA OTIMIZAÇÃO GENÉTICA DA BD (BOSTON HOUSING). | 77 |
| TABELA B.1: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (AUTO-MPG)..... | 85 |
| TABELA B.2: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (AUTO-MPG). | 85 |
| TABELA B.3: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (AUTO-MPG). | 88 |
| TABELA B.4: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (AUTO-MPG). | 88 |
| TABELA B.5: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (AUTO-MPG). | 91 |
| TABELA B.6: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (AUTO-MPG). | 91 |
| TABELA C.1: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (IRIS PLANTS). | 95 |
| TABELA C.2: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (IRIS PLANTS). | 95 |

| | |
|--|-----|
| TABELA C.3: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (IRIS PLANTS)..... | 97 |
| TABELA C.4: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (IRIS PLANTS)..... | 97 |
| TABELA C.5: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (IRIS PLANTS)..... | 99 |
| TABELA C.6: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (IRIS PLANTS)..... | 99 |
| TABELA D.1: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (WINE) .. | 102 |
| TABELA D.2: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (WINE). | 102 |
| TABELA D.3: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (WINE) .. | 105 |
| TABELA D.4: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (WINE). | 105 |
| TABELA D.5: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (WINE) .. | 108 |
| TABELA D.6: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (WINE). | 108 |
| TABELA E.1: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 5 (BOSTON HOUSING)..... | 112 |
| TABELA E.2: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 5 (BOSTON HOUSING)..... | 112 |
| TABELA E.3: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 7 (BOSTON HOUSING)..... | 115 |
| TABELA E.4: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 7 (BOSTON HOUSING)..... | 115 |
| TABELA E.5: RESULTADOS PARA A CONSTRUÇÃO DA BD COM GRANULARIDADE 9 (BOSTON HOUSING)..... | 118 |
| TABELA E.6: RESULTADOS PARA A OTIMIZAÇÃO DA BD COM GRANULARIDADE 9 (BOSTON HOUSING)..... | 118 |

Resumo

O sucesso dos *Sistemas Baseados em Regras Nebulosas* deve-se à habilidade para representar o conhecimento vago e incerto e à facilidade de expressar o comportamento do sistema em uma linguagem facilmente interpretável pelos seres humanos. A aquisição de conhecimento de um Sistema Baseado em Regras Nebulosas é feita a partir de um especialista humano, a qual é conhecida como abordagem direta para aquisição do conhecimento, ou através de métodos que automaticamente extraem este conhecimento a partir de informações numéricas que representam amostras ou exemplos do problema. *Algoritmos Genéticos* vêm demonstrando ser uma poderosa ferramenta para a construção de bases de regras, otimização de bases de regras, geração de funções de pertinência e otimização de funções de pertinência. O principal foco de pesquisa deste trabalho é a investigação das abordagens de modelagem automática de sistemas nebulosos aplicados a problemas de classificação de padrões, através de algoritmos genéticos para a *definição* e *sintonia* dos conjuntos nebulosos que compõem as partições nebulosas dos domínios envolvidos, considerando a manutenção da interpretabilidade dos valores lingüísticos. O aprendizado genético é empregado somente na Base de Dados do sistema nebuloso, isto é, nas funções de pertinência, tanto pela maneira mais tradicional, de sintonizar as funções já definidas depois da definição das regras, como pela abordagem mais recente, de construir as funções de pertinência antes da definição das regras. Esse tipo de investigação supõe o uso de outros métodos para a geração das regras, sejam elas geradas antes ou depois da aplicação do algoritmo genético. A combinação de Sistemas Nebulosos + Algoritmos Genéticos, conhecida como *Sistemas Genéticos Nebulosos*, tem grande aceitação na comunidade científica, uma vez que estes sistemas são robustos e capazes de encontrar boas soluções em espaços complexos e irregulares.

Abstract

The success of Fuzzy Rule Based Systems is due to their ability to represent vague and uncertain knowledge and to the facility to express the system behavior in a language easily interpretable by human beings. The knowledge acquisition of a Fuzzy Rule Based System is done from a human expert, which is known as the direct approach to knowledge acquisition, or through methods that automatically extract this knowledge from numerical information that represent samples or examples of the problem. Genetic Algorithms have demonstrated to be a powerful tool to rule bases construction, rule bases optimization, membership functions generation and membership functions optimization. The main research focus of this work is the investigation of fuzzy systems automatic generation approaches applied to pattern classification problems, using genetic algorithms to the definition and tuning of fuzzy sets that belong to the fuzzy partitions of the domains involved, considering interpretability maintenance of the linguistic values. The genetic learning is employed only in the fuzzy system data base, that is, in the membership functions, either through the traditional way of tuning previously defined functions after the rule base has been defined or through the more recent approach of constructing membership functions before the rules definition. This kind of investigation requires the use of different rule generation methods, the rules being generated before or after the use of genetic algorithm. The combination Fuzzy Systems + Genetic Algorithms, known as Genetic Fuzzy Systems, has great acceptance in the scientific community, since this systems are robust and capable of finding good solutions in complex and irregular spaces.

Capítulo 1

Introdução

1.1 Considerações Iniciais

Na década de 90 Zadeh [Zadeh 1997] introduziu o conceito de *Computação Flexível* (CF), o qual representa uma combinação de técnicas da Inteligência Computacional, como por exemplo, Lógica Nebulosa, Redes Neurais e Algoritmos Genéticos (alguns autores consideram além destas, outras técnicas que formam a CF, no entanto, aqui são consideradas apenas estas três como principais componentes). A utilização cooperativa destas técnicas oferece formas de raciocínio e busca para a solução de problemas complexos do mundo real que apresentam situações indeterminadas, tais como, domínios de aplicação caracterizados por dados imprecisos e conhecimento incompleto. Exemplos destes tipos de problemas são diagnóstico, sistemas de identificação, estimação e controle [Bonissone et al. 1999]. Um aspecto essencial da CF é o fato de que as metodologias que a constitui são complementares e simbióticas, ao invés de competitivas e exclusivas [Zadeh 1998].

A Lógica Nebulosa introduzida por Zadeh [Zadeh 1965], nos dá uma linguagem com sintaxe e semântica capaz de traduzir o conhecimento do domínio do problema em sentenças lingüísticas de fácil compreensão para o ser humano, sendo que a principal característica da Lógica Nebulosa é a robustez do seu mecanismo de raciocínio no tratamento das informações representadas por estas sentenças. Redes Neurais [Haykin 1994] são estruturas computacionais que simulam o comportamento do cérebro humano, as quais possuem a capacidade de aprender através de exemplos e de generalizar a informação aprendida. Os Algoritmos Genéticos propostos por John Holland [Holland 1975] são algoritmos de

otimização e busca baseados nos mecanismos da seleção natural e genética, capazes de executar uma procura global em um espaço de solução complexo e irregular.

Considerando as características apresentadas de cada técnica e utilizando-as de uma forma onde as vantagens de uma sobreponham as desvantagens de outra, é possível construir sistemas híbridos cada vez mais robustos para a resolução de problemas do mundo real.

1.2 Objetivos e Motivação

Modelagem Nebulosa é o processo de modelagem de sistemas pelo uso de termos lingüísticos, capazes de processar e representar imprecisão e incerteza. Os sistemas que utilizam regras nebulosas para representar e processar informações imprecisas são conhecidos como *Sistemas Baseados em Regras Nebulosas (SBRN)*, os quais têm sido aplicados com sucesso em problemas de classificação, modelagem e controle em um número considerável de aplicações [Cordón et al. 2004]. As chaves para o sucesso e interesse nesse tipo de sistema são a habilidade para representar o conhecimento vago e incerto e a facilidade de expressar o comportamento do sistema em uma linguagem facilmente interpretável pelos seres humanos. O bom desempenho de um SBRN está fortemente relacionado à composição de sua *Base de Conhecimento (BC)*, pois é o local onde todo o conhecimento do domínio do problema que será tratado pelo sistema é expresso. A *BC* é constituída por uma *Base de Regras (BR)*, onde está definida a quantidade e o formato das regras e por uma *Base de Dados (BD)*, onde é especificado o domínio das variáveis utilizadas pelo sistema, assim como a partição nebulosa de cada variável, ou seja, o número de conjuntos nebulosos e as funções de pertinência associadas a cada conjunto.

Existem duas maneiras típicas de aquisição do conhecimento para um SBRN. Uma delas é a extração do conhecimento a partir de um especialista humano, a qual é conhecida como abordagem direta para aquisição do conhecimento. No entanto, muitas vezes o especialista não consegue detalhar todo o seu conhecimento sobre o domínio do problema ou não há um especialista disponível para o problema abordado. Em outros casos, o conhecimento está disponível como exemplos inseridos em bases de dados de dimensão elevada, sendo necessário um processo de extração, pois nem sempre esta é uma tarefa fácil para um operador humano. Portanto, a segunda maneira de realizar a aquisição do conhecimento é extraí-lo automaticamente por meio de métodos que generalizam este

conhecimento a partir de informações numéricas que representam amostras ou exemplos do problema.

Nos últimos anos, os métodos para extração automática de conhecimento têm recebido especial atenção dos pesquisadores e o número de trabalhos enfocando técnicas voltadas para esse fim tem crescido substancialmente. Em particular, os métodos baseados em *Algoritmos Genéticos (AG)* [Goldberg 1989] têm sido explorados com sucesso, sendo muitas vezes usados de forma combinada com outras metodologias. Essa categoria de sistemas enquadra-se no que passou a ser chamado de *Sistemas Genéticos Nebulosos (SGN)* [Cordón et al. 2004].

Um SGN é basicamente um sistema nebuloso combinado com um processo de aprendizado baseado em algoritmos genéticos. A combinação genético-nebulosa tem grande aceitação na comunidade científica devido principalmente a sua aderência ao princípio de balanceamento de vantagens da inteligência computacional [Pedrycz 1998] [Rezende 2003], a qual se fundamenta nos conceitos da computação flexível, onde metodologias diferentes colaboram entre si potencializando a utilidade e aplicabilidade dos sistemas resultantes. As linhas de pesquisa mais extensamente exploradas aplicam os algoritmos genéticos para realizar a sintonia dos conjuntos nebulosos, para construir as regras nebulosas a partir de conjuntos previamente definidos ou para projetar ambos os componentes da BC.

Uma das dificuldades apontadas no tratamento da questão do aprendizado da BC com algoritmos genéticos está relacionada à complexidade do espaço de busca, o que causa um impacto na eficiência do algoritmo. Assim, as abordagens que combinam algoritmos genéticos para geração de apenas um componente da base de conhecimento, com outros métodos eficientes, são vistas como uma opção viável, no sentido de simplificar o processo de construção do sistema, sem prejuízo do desempenho final.

O objetivo deste trabalho é a investigação das abordagens que aplicam algoritmos genéticos na BD, tanto pela maneira mais tradicional, de sintonizar a BD predefinida depois da definição da BR, como pela abordagem mais recente, de construir a BD antes da definição das regras. Esse tipo de investigação supõe o uso de outros métodos para a geração das regras, sejam elas geradas antes ou depois da aplicação do algoritmo genético.

1.3 Organização da Dissertação

Esta dissertação está organizada da seguinte maneira: No Capítulo 2 é apresentado uma visão geral sobre Computação Evolutiva, com ênfase em Algoritmos Genéticos. No Capítulo 3 são descritos os conceitos básicos sobre Conjuntos Nebulosos, Lógica Nebulosa e Sistemas Nebulosos. A proposta propriamente dita do trabalho é apresentada no Capítulo 4. Os resultados e experimentos realizados são detalhados no Capítulo 5. Finalmente, no Capítulo 6, as conclusões e futuros trabalhos são comentados.

Capítulo 2

Computação Evolutiva

2.1 Considerações Iniciais

Este capítulo tem por objetivo apresentar as vantagens oferecidas pelos mecanismos de evolução presentes em sistemas biológicos quando aplicados a sistemas computacionais. Estes por sua vez, possuem a tarefa de encontrar as soluções de um determinado problema dentre um conjunto de soluções em potencial.

Na seção 2.2 é apresentado um breve histórico da computação evolutiva, relatando os principais paradigmas que formam a base da computação evolutiva – estratégias evolutivas, programação evolutiva e algoritmos genéticos. Na seção 2.3 tem-se por objetivo apresentar as razões do uso da computação evolutiva. Finalmente, na seção 2.4, são discutidos os conceitos específicos dos algoritmos genéticos, descrevendo suas formas de representação, métodos de seleção e os operadores genéticos utilizados, além de trazer um exemplo de aplicação.

2.2 Um Breve Histórico

Nas décadas de 50 e 60 vários cientistas da computação estudaram sistemas evolutivos com o objetivo de utilizar a evolução como uma ferramenta de otimização para problemas de engenharia, como, por exemplo, controle de processos estocásticos, aprendizado de máquina e otimização de funções. A idéia era evoluir uma população de soluções candidatas para o problema, usando operadores inspirados pela genética e seleção natural.

Na década de 60, Rechenberg [Rechenberg 1965] [Rechenberg 1973] introduziu *Estratégias Evolutivas (EE)*, para otimização de parâmetros com valores reais. Esta idéia foi mais tarde desenvolvida por Schwefel [Schwefel 1975] [Schwefel 1977]. O campo das estratégias evolutivas tem permanecido como uma área de pesquisa ativa, sendo desenvolvido independentemente dos algoritmos genéticos, embora recentemente eles tenham começado a se interagir. Para mais detalhes sobre as estratégias evolutivas ver [Bäck et al. 1991].

Fogel, Owens e Walsh [Fogel et al. 1966] desenvolveram a *Programação Evolutiva (PE)*, uma técnica que representavam as soluções candidatas como máquinas de estados finitos, as quais eram desenvolvidas pela mutação aleatória de seus diagramas de transição e pela seleção dos mais aptos. Este tipo de máquina foi treinada para prever ciclos repetitivos de símbolos de saída usando as técnicas da programação evolutiva de Fogel, Owens e Walsh, que consiste fundamentalmente de dois operadores: *seleção* e *mutação*.

De uma forma bem simples, o operador de seleção escolhe a melhor máquina entre duas, a máquina pai ou a máquina filho resultante da mutação. O operador de mutação era o único mecanismo utilizado para modificar as estruturas das máquinas. Para Fogel, Owens e Walsh, a mutação tinha o seguinte sentido:

“Um filho de uma máquina é produzido através da mutação, isto é, através de uma simples modificação nas máquinas pais de acordo com alguma distribuição. O modo da mutação é determinado por um intervalo, de onde um número aleatório é selecionado. Os intervalos são escolhidos de acordo com a distribuição da probabilidade sobre os modos permitidos da mutação. Números adicionais são selecionados para determinar detalhes específicos da mutação. Portanto, o filho é diferente de seu pai pelo símbolo de saída, ou pelo estado de transição, ou pelo número de estados ou pelo estado inicial”.

Os *Algoritmos Genéticos (AG's)* foram introduzidos John Holland nos anos 60 e foram desenvolvidos por ele e seus alunos e colegas da Universidade de Michigan nas décadas de 60 e 70. Em contraste com as estratégias evolutivas e a programação evolutiva, o objetivo original de Holland não era projetar algoritmos para solucionar problemas específicos, mas estudar o fenômeno da adaptação na natureza e desenvolver formas onde os mecanismos da adaptação natural pudessem ser simulados em computadores. Em seu livro [Holland 1975], apresentou os algoritmos genéticos como uma abstração da evolução biológica. O algoritmo genético introduzido por Holland é um método que retira *cromossomos* (cadeias de uns e zeros, ou “bits”) de uma população de cromossomos e insere-os para uma nova população, usando um tipo de *seleção natural* juntamente com os operadores de cruzamento e mutação

inspirados na genética. Cada cromossomo consiste de *genes* (bits), e cada gene representa uma instância de um *alelo* particular (zero ou um). O operador de seleção escolhe, dentre os cromossomos da população, aqueles que irão se reproduzir. Em média, os cromossomos mais fortes (melhores adaptados) produzem mais descendentes do que aqueles mais fracos (menos adaptados). Os cromossomos selecionados podem sofrer modificações em suas características através dos operadores de *cruzamento* e *mutação*, gerando descendentes para a próxima geração [Mitchell 1996].

Juntos, as estratégias evolutivas, programação evolutiva e algoritmos genéticos, formam a base da *Computação Evolutiva (CE)*.

2.3 Porque usar Computação Evolutiva

Muitos problemas computacionais requerem uma busca através de um alto número de possíveis soluções, chamado de *espaço de busca*. O termo espaço de busca refere-se a uma coleção de soluções candidatas para um determinado problema [Mitchell 1996]. Um exemplo é o problema de encontrar computacionalmente, isto é, por meio de um algoritmo, entre um vasto número de possíveis seqüências de aminoácidos, a estrutura de uma proteína com propriedades específicas. Um outro exemplo é encontrar uma forma para construir um conjunto de regras ou equações que predizem os altos e baixos de um mercado financeiro, tal como o valor de uma moeda estrangeira.

O uso do *paralelismo* para a resolução de tais problemas de busca pode ser uma boa opção, pois várias possibilidades são exploradas simultaneamente. Por exemplo, no problema da busca da proteína com propriedades específicas, ao invés de analisar uma seqüência de aminoácidos de cada vez, é muito mais rápido avaliar várias ao mesmo tempo. No entanto, para obter a eficiência do paralelismo, deve-se adotar uma estratégia inteligente para escolher qual será o próximo conjunto de seqüências de aminoácidos que serão avaliados. Além do paralelismo, muitos problemas computacionais necessitam de algoritmos *adaptativos*, para serem aplicados em ambientes em constante alteração. Este é um problema típico de robótica, onde um robô precisa ser controlado para executar tarefas em tais ambientes. Outros problemas requerem algoritmos *inovadores*, que construam algo realmente novo e original. Por fim, há problemas que exigem soluções complexas, como por exemplo, a criação de inteligência artificial [Mitchell 1996].

A Computação Evolutiva se mostra particularmente apropriada para lidar com estes tipos de problemas, porque incorpora todas estas características (busca da solução, paralelismo, adaptação e inovação).

2.4 Algoritmos Genéticos

Algoritmos Genéticos (AG's) são algoritmos de busca baseados no mecanismo da seleção natural e da genética natural que foram introduzidos por John Holland e desenvolvidos por ele e seus alunos e colegas da Universidade de Michigan nas décadas de 60 e 70.

A idéia que há por trás dos AG's é imitar o que a natureza faz. Considere-se uma população de coelhos qualquer como um exemplo. Alguns destes coelhos são mais rápidos e espertos do que outros. Estes coelhos mais rápidos e espertos possuem maiores chances de sobreviverem aos seus predadores, portanto, irão fazer o que mais sabem fazer: mais coelhos. É claro que desta população, alguns coelhos lentos e bobos sobreviverão, mas porque eles são sortudos. A população sobrevivente de coelhos dará então início à reprodução, a qual resultará em uma boa mistura de material genético de coelhos: alguns coelhos lentos se reproduzirão com coelhos rápidos, alguns coelhos rápidos com coelhos rápidos, alguns coelhos espertos com coelhos bobos, e assim por diante. O resultado será filhotes de coelhos (em sua maioria) mais rápidos e espertos em relação aos coelhos da população original, devido que os coelhos pais sobreviventes aos predadores são em sua maioria rápidos e espertos [Michalewicz 1996]. Em outras palavras, os algoritmos genéticos simulam processos naturais de sobrevivência e reprodução dos seres vivos que são essenciais em sua evolução. Na natureza, os indivíduos de uma mesma população competem entre si, buscando principalmente a sobrevivência, e, quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.

Em se tratando de propósitos computacionais, um algoritmo genético simples possui uma estrutura conforme o pseudocódigo a seguir [Michalewicz 1996]:

```
procedure evolution program
begin
   $t \leftarrow 0$ 
  initialize  $P(t)$ 
  evaluate  $P(t)$ 
  while (not termination condition) do
    begin
```

```

     $t \leftarrow t + 1$ 
    select  $P(t)$  from  $P(t - 1)$ 
    alter  $P(t)$ 
    evaluate  $P(t)$ 
  end
end

```

Durante a iteração t , o algoritmo genético mantém uma população de soluções candidatas (cromossomos), $P(t) = \{x_1^t, \dots, x_n^t\}$. Cada solução x_i^t é avaliada para medir seu *fitness* (ou *aptidão*), ou seja, a qualidade da solução do problema representada por este cromossomo. Então, uma nova população (iteração $t + 1$) é formada pela *seleção* dos indivíduos mais aptos. Alguns membros desta nova população sofrerão alterações devido à ação dos operadores genéticos de *cruzamento* e *mutação*, enquanto outros permanecerão intactos. O cruzamento combina as características de dois cromossomos pais para formar dois cromossomos filhos. O objetivo da aplicação do cruzamento é trocar informações entre soluções em potencial. A mutação altera aleatoriamente um ou mais genes de um cromossomo selecionado, com o intuito de introduzir informação extra para a população.

Segundo [Michalewicz 1996], para a implementação de um algoritmo genético deve-se primeiramente definir alguns aspectos importantes, tais como:

- Uma *representação genética* para as soluções do problema. A representação binária é tradicionalmente usada em algoritmos genéticos uma vez que é de fácil utilização e manipulação, além de simples de analisar teoricamente. Contudo, apresenta algumas desvantagens quando aplicada a problemas multidimensionais e a problemas numéricos de alta precisão [Michalewicz 1996].
- Uma forma de criar uma *população inicial* das soluções. A escolha de uma população inicial maior que a população a ser utilizada nas gerações subseqüentes pode melhorar a representação do espaço de busca. A população inicial pode ser gerada de várias maneiras. Se uma população inicial pequena for gerada aleatoriamente, provavelmente algumas regiões do espaço de busca não serão representadas [Lacerda e Carvalho 1980].
- Uma *função de avaliação* que desempenha o papel do ambiente, ou seja, avalia as soluções em termos de *fitness*. [Lacerda e Carvalho 1980].

- *Operadores genéticos* que alteram a composição dos filhos. O operador de cruzamento tem o intuito de trocar informações entre diferentes soluções em potencial. Já a aplicabilidade do operador de mutação, tem por objetivo introduzir uma variabilidade extra na população [Michalewicz 1996].
- Valores para os parâmetros utilizados pelo AG (*tamanho da população, probabilidades de aplicação dos operadores genéticos e número de gerações*). A influência de cada parâmetro no desempenho do algoritmo depende da classe de problemas que se está tratando. Assim, a determinação de um conjunto de valores otimizado para estes parâmetros dependerá da realização de um grande número de experimentos e testes. Existem estudos que utilizam um AG como método de otimização para a escolha dos parâmetros de outro AG [Grefenstette 1986] [Bramlette 1991], devido à importância da escolha correta destes parâmetros.

2.4.1 Terminologia Biológica

Na Biologia, a teoria da evolução diz que o meio ambiente seleciona, em cada geração, os seres vivos mais aptos de uma população para sobrevivência. Como resultado, somente os mais aptos conseguem geralmente se reproduzir, uma vez que os menos adaptados geralmente são eliminados antes de gerarem descendentes. Durante a reprodução, ocorrem fenômenos como mutação e cruzamento, que atuam sobre o material genético armazenado nos cromossomos. Estes fenômenos levam à variabilidade dos seres vivos na população. Sobre esta população diversificada age a seleção natural, permitindo a sobrevivência apenas dos seres mais adaptados.

Um algoritmo genético é a metáfora desses fenômenos, o que explica porque os algoritmos genéticos possuem muitos termos originados da biologia. A lista que segue descreve os principais termos encontrados na literatura [Lacerda e Carvalho 1980]:

- **Genoma e Cromossomo:** Na biologia, genoma é o conjunto completo de genes de um organismo. Um genoma pode ter vários cromossomos. Nos AG's, representam a estrutura de dados que codifica uma solução para um problema, ou seja, um cromossomo representa um simples ponto no espaço de busca.

- **Gene:** Na biologia é a unidade de hereditariedade que é transmitida pelo cromossomo e que controla as características do organismo. Nos AG's, é um parâmetro codificado no cromossomo, ou seja, um elemento do vetor que representa o cromossomo.
- **Indivíduo:** Um simples membro da população. Nos AG's, um indivíduo é formado pelo cromossomo e sua aptidão.
- **Genótipo:** Na biologia representa a composição genética contida no Genoma. Nos AG's, representa a informação contida no cromossomo ou genoma.
- **Fenótipo:** Nos AG's representa o objeto, estrutura ou organismo construído a partir das informações do genótipo. É o cromossomo decodificado. Por exemplo, considere que o cromossomo codifica parâmetros como as dimensões das vigas em um projeto de construção de um edifício, ou as conexões e pesos de uma Rede Neural. O fenótipo seria o edifício construído ou a Rede Neural.
- **Alelo:** Na biologia representa uma das formas alternativas de um gene. Nos AG's, representa os valores que o gene pode assumir. Por exemplo, um gene que representa o parâmetro cor de um objeto poderia ter o alelo azul, preto, verde, etc.

2.4.2 Métodos de Seleção

Dada uma população em que a cada indivíduo foi atribuído um valor de aptidão, existem vários métodos para selecionar os indivíduos sobre os quais serão aplicados os operadores genéticos de cruzamento e mutação. Estes indivíduos selecionados formarão uma população, que é conhecida como *população intermediária*. A maioria dos métodos de seleção são projetados para escolher preferencialmente indivíduos com maiores notas de aptidão, embora não exclusivamente, a fim de manter a diversidade da população. A seguir veremos alguns destes métodos, com base em [Michalewicz 1996].

Seleção pela Roleta

O método de seleção da Roleta é um método de seleção muito utilizado, onde os indivíduos de uma geração são escolhidos para fazer parte da população intermediária através de um sorteio de roleta. Neste método cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão, como é mostrado na Figura 2.1. Assim, aos

indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes dependendo do tamanho da população e são escolhidos como indivíduos que participarão da população intermediária aqueles sorteados na roleta.

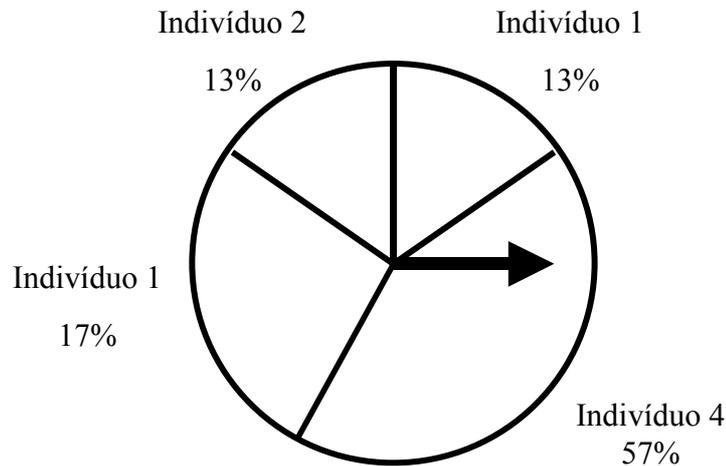


Figura 2.1: Exemplo de uma roleta de seleção.

Seleção por Amostragem Universal Estocástica

Para visualizar este método, considere um círculo dividido em n regiões (tamanho da população), onde a área de cada região é proporcional à aptidão do indivíduo (Figura 2.2). Coloca-se sobre este círculo uma "roleta" com n cursores, igualmente espaçados. Após um giro da roleta a posição dos cursores indica os indivíduos selecionados. Evidentemente, os indivíduos cujas regiões possuem maior área, terão maior probabilidade de serem selecionados várias vezes. Como consequência, a seleção de indivíduos pode conter várias cópias de um mesmo indivíduo enquanto outros podem desaparecer. Este tipo de seleção é mais rápido que a Roleta, pois em uma só rodada já são selecionados todos os indivíduos.

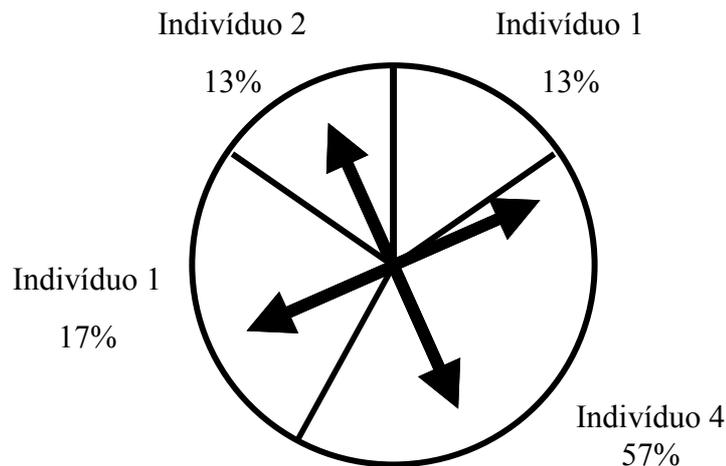


Figura 2.2: Seleção por amostragem universal estocástica.

Seleção por Torneio

No método de seleção por Torneio são escolhidos aleatoriamente (com probabilidades iguais) n indivíduos da população e o que possuir maior aptidão é selecionado para a população intermediária. O processo repete-se até completar a população intermediária.

Seleção por Truncamento

O método de seleção por Truncamento tem como base um valor T entre zero e um. A seleção é feita aleatoriamente entre os T melhores indivíduos. Por exemplo, se $T = 0.4$, então a seleção é feita entre os 40% melhores indivíduos e os outros 60% são descartados.

Seleção Elitista ou Elitismo

A seleção elitista ou elitismo, introduzido por Kenneth De Jong (1975), é um método de seleção que força o algoritmo genético a reter um determinado número de melhores indivíduos em cada geração. A aplicação do elitismo previne que tais indivíduos sejam destruídos pela aplicação dos operadores de cruzamento ou mutação [Mitchell 1996]. Devido ao seu comportamento, a seleção elitista é sempre utilizada em conjunto com algum outro método de seleção.

2.4.3 Operadores Genéticos

A mutação é necessária para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais indivíduos, fornecendo assim, meios para a introdução de novos indivíduos na população. O operador de mutação é aplicado aos indivíduos com uma probabilidade dada pela *taxa de mutação*. Uma baixa taxa de mutação pode assegurar a diversidade na população, mas o contrário, pode destruir toda a informação contida no indivíduo que foi adquirida durante as gerações passadas.

O operador de cruzamento é aplicado aos cromossomos de acordo com uma probabilidade, conhecida como *taxa de cruzamento*. Quanto maior for esta taxa, mais rapidamente novos indivíduos serão introduzidos na população. Mas se esta for muito alta, indivíduos com boas aptidões poderão ser retirados mais rapidamente, assim, a maior parte da população será substituída e poderá ocorrer a perda de indivíduos com alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento. Assim, as taxas de mutação e cruzamento devem ser definidas empiricamente para cada domínio de problema, levando em conta essas considerações.

A escolha dos operadores genéticos está intimamente ligada à codificação adotada para a representação genética [Delgado 2002], ou seja, existe uma variação nos comportamentos dos operadores, os quais estão relacionados à codificação empregada nos cromossomos. Em outras palavras, há operadores genéticos que foram concebidos para o uso com codificação binária e operadores genéticos para o uso com codificação real ou inteira.

Operadores Genéticos para Codificação Binária

A seguir serão descritos alguns operadores genéticos de cruzamento e mutação para cromossomos com codificação binária, com base em [Michalewicz 1996].

Cruzamento de Um-Ponto

O *cruzamento de um-ponto* é a forma mais simples deste operador. É aplicado a um par de cromossomos retirados da população intermediária, gerando dois cromossomos filhos. Cada um dos cromossomos pais é cortado em uma posição aleatória produzindo duas cabeças e

duas caudas. As caudas são trocadas gerando dois novos cromossomos. A Figura 2.3 ilustra o comportamento deste operador.

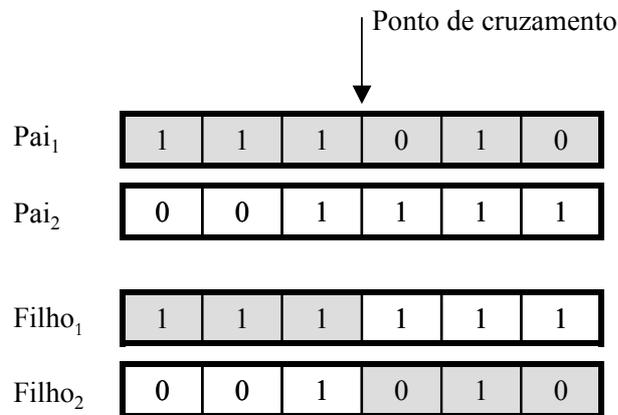


Figura 2.3: Cruzamento de um-ponto.

Cruzamento de N-Pontos

O *cruzamento de n-pontos* funciona da mesma forma que o cruzamento de um-ponto, porém, neste há vários pontos de corte nos cromossomos pais. Na Figura 2.4 é exemplificado o cruzamento de três-pontos.

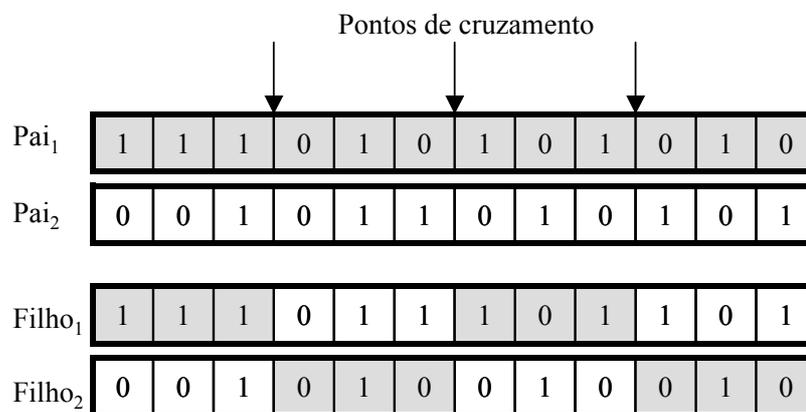


Figura 2.4: Cruzamento de três-pontos.

Cruzamento Uniforme

No *cruzamento uniforme* cada gene do cromossomo filho é criado copiando o gene correspondente de um dos pais, escolhido de acordo com uma máscara de cruzamento gerada aleatoriamente. Onde houver um na máscara de cruzamento, o gene correspondente será copiado do primeiro pai e onde houver zero será copiado do segundo. O processo é repetido

com os pais trocados para produzir o segundo descendente. Uma nova máscara de cruzamento é criada para cada par de pais. O número de pontos de cruzamento não é fixo, mas em geral é usado $L/2$ (onde L é o comprimento do cromossomo). Na Figura 2.5 o processo é ilustrado graficamente.

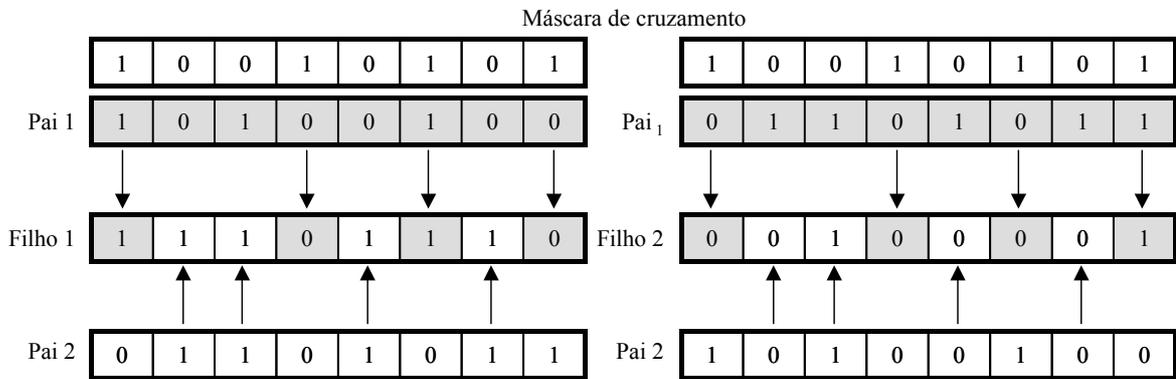


Figura 2.5: Cruzamento uniforme.

Estes três tipos de cruzamento foram concebidos para uso com codificação binária. Entretanto, não existe nenhum impedimento quanto ao uso destes operadores junto a outros tipos de codificação. Mas, para problemas com restrições, existem operadores de cruzamento específicos, que garantem a factibilidade dos descendentes originários de indivíduos factíveis [Delgado 2002].

Mutação Simples

O operador de *mutação simples* é aplicado aleatoriamente em alguns genes do cromossomo que sofrerá a mutação. Este processo inverte os valores dos genes, ou seja, muda o valor de um dado gene de '1' para '0' ou de '0' para '1'. Na Figura 2.6 é ilustrado o comportamento deste operador.

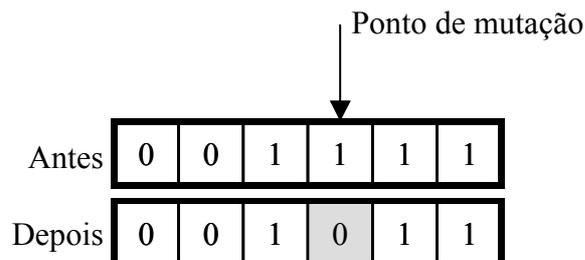


Figura 2.6: Mutação simples.

Operadores Genéticos para Codificação Real (ou Inteira)

A seguir serão descritos os operadores genéticos de cruzamento e mutação considerando que os indivíduos possuem uma codificação real ou inteira de acordo com [Michalewicz 1996], mas antes é necessário definir o conceito de *domínio*. *Domínio* define o intervalo de possíveis valores que os genes dos cromossomos podem assumir. Desta forma, o domínio para os genes dos cromossomos com codificação binária abrange os valores zero ou um, contudo, para os cromossomos com codificação real ou inteira, isto dependerá do problema que será tratado.

Cruzamento Simples

O operador genético *cruzamento simples* é definido da seguinte forma: se os cromossomos $C_1=(x_1,\dots,x_q)$ e $C_2=(y_1,\dots,y_q)$ são cruzados depois da k -ésima posição, os descendentes serão: $C'_1=(x_1,\dots,x_k,y_{k+1},\dots,y_q)$ e $C'_2=(y_1,\dots,y_k,x_{k+1},\dots,x_q)$. Tal operador pode produzir descendentes fora do domínio em questão. Para evitar isto, usa-se a propriedade do *espaço convexo*, onde $a \in [0,1]$ tal que:

$$\begin{aligned} C'_1 &= (x_1,\dots, x_k, y_{k+1} \cdot a + x_{k+1} \cdot (1-a),\dots, y_q \cdot a + x_q \cdot (1-a)), \\ C'_2 &= (y_1,\dots, y_k, x_{k+1} \cdot a + y_{k+1} \cdot (1-a),\dots, x_q \cdot a + y_q \cdot (1-a)). \end{aligned} \quad (2.1)$$

Na Figura 2.7 é ilustrado um exemplo do funcionamento de um cruzamento simples, sendo $a=0.7$.

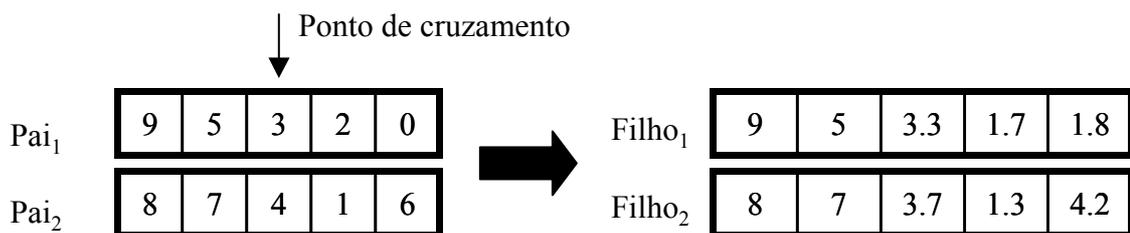


Figura 2.7: Cruzamento simples.

Cruzamento Aritmético

O *cruzamento aritmético* é definido como uma combinação linear de dois vetores. Se x_1 e x_2 são cruzados, os descendentes serão $x'_1=a.x_1+(1-a).x_2$ e $x'_2=a.x_2+(1-a).x_1$. Este operador usa

um valor aleatório $a \in [0,1]$, para garantir que x'_1 e x'_2 estejam dentro do domínio de x_1 e x_2 . Na Figura 2.8 é ilustrado o funcionamento deste operador, onde $a=0.7$.

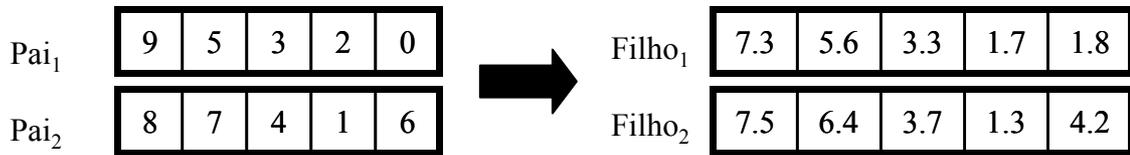


Figura 2.8: Cruzamento aritmético.

Cruzamento Heurístico

O *cruzamento heurístico* utiliza valores da função de *fitness* na determinação da direção da busca, gerando somente um descendente ou não gerar descendente algum. O cromossomo filho x_3 é gerado a partir de dois cromossomos pais x_1 e x_2 , de acordo com a seguinte regra: $x_3=r(x_2 - x_1) + x_2$, onde r é um número aleatório entre zero e um. Na Figura 2.9 é ilustrado o comportamento deste operador, onde $r=0.8$.

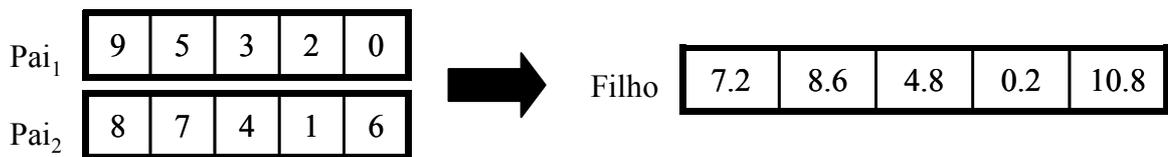


Figura 2.9: Cruzamento heurístico.

Mutação Uniforme

Este operador seleciona um componente aleatório $k \in \{1,2,\dots,q\}$ do cromossomo $C=(x_1,\dots,x_k,\dots,x_q)$ e produz $C'=(x_1,\dots,x'_k,\dots,x_q)$, onde x'_k é um valor aleatório (distribuição probabilística uniforme) do intervalo $[esquerda(k), direita(k)]$, onde $esquerda(k)$ é o limite inferior e $direita(k)$ é o limite superior do intervalo que representa o domínio do gene x'_k . Na Figura 2.10 é ilustrado um exemplo, onde $q=5$ e $k=4$.



Figura 2.10: Mutação uniforme.

Mutação Limite

O operador genético de *mutação limite* é uma variação do operador de mutação uniforme com x'_k sendo ou esquerda(k) ou direita(k), cada um com a mesma probabilidade. Na Figura 2.11 é ilustrado um exemplo onde o gene alterado recebe o valor do limite superior, sendo $q=5$, $k=4$ e direita(k) = 10.



Figura 2.11: Mutação limite.

Mutação Não-Uniforme

Seja um cromossomo pai $C=(x_1, \dots, x_k, \dots, x_q)$. Se o elemento x_k for selecionado para mutação o resultado será $C'=(x_1, \dots, x'_k, \dots, x_q)$, onde:

$$x'_k = \begin{cases} x_k + \Delta(t, \text{direita}(k) - x_k) & \text{se o dígito binário é 0} \\ x_k + \Delta(t, x_k - \text{esquerda}(k)) & \text{se o dígito binário é 1} \end{cases} \quad (2.2)$$

A função $\Delta(t,y)$ retorna um valor no intervalo $[0,y]$ tal que, quando a probabilidade de $\Delta(t,y)$ se aproxima de zero, ela é incrementada de acordo com o aumento de t (número de gerações). Esta propriedade faz o operador buscar uniformemente pelo espaço (quando t é pequeno), e localmente em estágios posteriores. A função $\Delta(t,y)$ é definida pela expressão: $\Delta(t,y)=y.r.(1-t/T)^b$, onde r é um número aleatório de $[0,1]$, T é o número máximo de gerações e b é um parâmetro do sistema que determina o grau de não-uniformidade.

Na próxima seção serão discutidas as principais características de um algoritmo genético utilizando um problema de otimização de uma função.

2.4.4 Exemplo de Aplicação de um Algoritmo Genético

Nesta seção serão discutidas as características básicas de um algoritmo genético para otimização de uma função com uma variável.

Otimização é a procura pelo melhor desempenho em direção a um ou alguns pontos ótimos [Goldberg 1989], ou seja, é a busca da melhor solução para um dado problema.

Consiste em tentar várias soluções e utilizar a informação obtida neste processo de forma a encontrar soluções cada vez melhores. Um exemplo simples de otimização é a melhoria da imagem das televisões com antena acoplada no próprio. Através do ajuste manual da antena, várias soluções são testadas, guiadas pela qualidade de imagem obtida na TV, até a obtenção de uma resposta ótima, ou seja, uma boa imagem.

Em termos matemáticos, a otimização consiste em achar a solução que corresponda ao ponto de *máximo* ou *mínimo* de uma função. Como exemplo, considere o seguinte problema de otimização: Problema 1) Seja a função definida por:

$$f(x) = x \operatorname{sen}(10 \pi x) + 1 \quad (2.3)$$

ilustrada na Figura 2.12. Encontre $x \in [-1..2]$, que maximiza a função f , isto é, encontrar x_0 tal que $f(x_0) \geq f(x)$, para todo $x \in [-1..2]$.

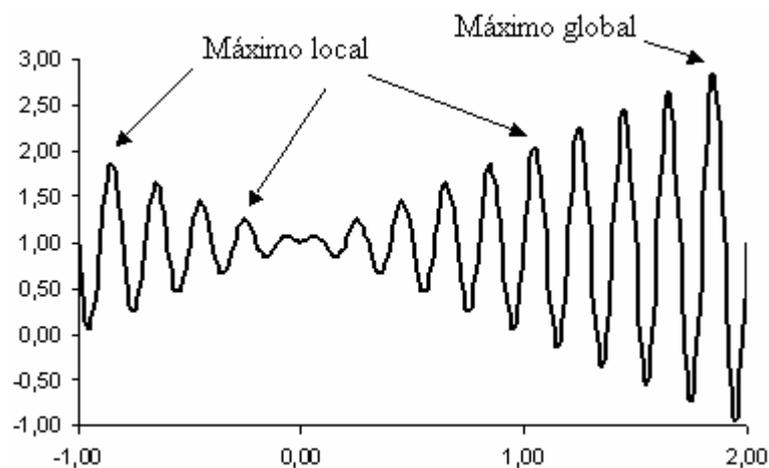


Figura 2.12: Gráfico da função $f(x) = x \operatorname{sen}(10 \pi x) + 1$

Embora aparentemente simples, o problema 1 não é de fácil solução. Existem vários pontos de máximos nesta função (pontos que maximizam o valor da função), mas muitos não representam o maior que a função pode atingir, conforme ilustrado na Figura 2.12. Tais pontos são denominados **máximos locais**, uma vez que a função nestes pontos atinge valores maiores do que na vizinhança destes pontos. A melhor solução para este problema está no ponto em que a função possui valor máximo, o **máximo global**. Neste problema, o máximo global encontra-se no ponto cujo valor de x é igual a 1,85055. Neste ponto, a função assume o valor 2,85027. A seguir será descrito os principais componentes de um AG para encontrar o máximo global para o problema 1.

Representação dos Parâmetros

Um vetor binário na forma de um cromossomo é usado para representar os valores reais da variável x . O tamanho do vetor dependerá da precisão desejada, a qual, neste caso, é de seis casas decimais.

O domínio da variável x possui tamanho três; a precisão requerida implica que o intervalo $[-1..2]$ deva ser dividido em pelo menos $3 \cdot 1000000$ intervalos iguais. Isto significa que 22 bits são necessários para o cromossomo:

$$2097152 = 2^{21} < 3000000 \leq 2^{22} = 4194304.$$

O mapeamento da cadeia de bits $\langle b_{21}b_{20}...b_0 \rangle$ para um número real x a partir do intervalo $[-1..2]$ é feito em dois passos:

1. converter a cadeia binária $\langle b_{21}b_{20}...b_0 \rangle$ da base 2 para a base 10:

$$\langle \langle b_{21}b_{20}...b_0 \rangle \rangle_2 = \left(\sum_{i=0}^{21} b_i * 2^i \right)_{10} = x', \quad (2.4)$$

2. encontrar um número real x correspondente:

$$x = -1,0 + x' * \frac{3}{2^{22} - 1}, \quad (2.5)$$

onde $-1,0$ é o limite esquerdo do domínio e 3 é o tamanho do domínio.

Por exemplo, um cromossomo (1000101110110101000111) representa o número 0,637197, desde que: $x' = (1000101110110101000111)_2 = 2288967$ e

$$x = -1,0 + 2288967 * \frac{3}{4194303} = 0,637197.$$

É claro que os cromossomos (0000000000000000000000) e (1111111111111111111111) representam os limites do domínio, -1 e 2 , respectivamente.

População Inicial

Uma população inicial de cromossomos é criada, onde cada cromossomo é representado por um vetor binário com 22 bits. Todos os 22 bits para cada cromossomo são inicializados aleatoriamente.

Função de Avaliação

A função de avaliação $eval$ para o vetor binário v é equivalente à função f :

$$eval(v) = f(x), \quad (2.6)$$

onde o cromossomo v representa o valor real x .

Por exemplo, os seguintes três cromossomos: $v_1 = (1000101110110101000111)$, $v_2 = (000000111000000010000)$ e $v_3 = (1110000000111111000101)$, correspondem aos valores $x_1 = 0,637197$, $x_2 = -0,958973$ e $x_3 = 1,627888$, respectivamente. Conseqüentemente, a função de avaliação retornará os seguintes valores: $eval(v_1) = f(x_1) = 1,586345$, $eval(v_2) = f(x_2) = 0,078878$, $eval(v_3) = f(x_3) = 2,250650$. Claramente, o cromossomo v_3 é o melhor dos três.

Operadores Genéticos

Durante a fase de alteração do algoritmo genético, dois operadores genéticos clássicos são utilizados: mutação e cruzamento. A mutação altera um ou mais genes com uma probabilidade igual à taxa de mutação. Assumimos que o quinto gene do cromossomo v_3 foi selecionado para uma mutação. Sendo que o quinto gene deste cromossomo é 0, ele será trocado por 1. Portanto, após a mutação, o cromossomo v_3 será: $v'_3 = (1110100000111111000101)$.

Este cromossomo representa o valor $x'_3 = 1,721638$ e $f(x'_3) = -0,082257$. Logo, podemos verificar que esta mutação em particular resultou em um decréscimo significativo no valor de aptidão do cromossomo v_3 . Por outro lado, se o décimo gene sofresse mutação, o cromossomo resultante seria: $v''_3 = (111000000111111000101)$.

O valor correspondente é $x''_3 = 1,630818$ e $f(x''_3) = 2,343555$. Neste caso, o valor de aptidão do cromossomo v_3 melhorou em relação ao valor original ($f(x_3) = 2,250650$).

Agora, será ilustrada a aplicação do operador de cruzamento nos cromossomos v_2 e v_3 . Assumi-se que o ponto de cruzamento foi selecionado aleatoriamente após o quinto gene:

$$v_2 = (00000|01110000000010000),$$

$$v_3 = (11100|00000111111000101).$$

Os descendentes resultantes são:

$$v'_2 = (00000|00000111111000101),$$

$$v'_3 = (11100|01110000000010000).$$

Os valores de aptidão para estes descendentes são:

$$f(v'_2) = f(-0,998113) = 0,940865,$$

$$f(v'_3) = f(1,666028) = 2,459245.$$

Note que o segundo descendente possui um valor de aptidão melhor do que os seus pais.

Parâmetros

Para este problema em particular, os seguintes parâmetros são usados: tamanho da população = 50, probabilidade de cruzamento = 0,25 e probabilidade de mutação = 0,01.

Experimentos

Na Tabela 2.1 são relatados alguns valores obtidos pela função de avaliação nas respectivas gerações. O melhor cromossomo depois de 150 gerações foi $v_{max} = (1111001101000100000101)$, que corresponde ao valor $x_{max} = 1,850773$. Como esperado, $f(x_{max})$ é pouco maior do que 2,85.

Tabela 2.1: Alguns resultados de 150 gerações.

| Geração | Função de avaliação |
|----------------|----------------------------|
| 1 | 1,441942 |
| 6 | 2,250003 |
| 8 | 2,250283 |
| 9 | 2,250284 |
| 10 | 2,250363 |
| 12 | 2,328077 |
| 39 | 2,344251 |
| 40 | 2,345087 |
| 51 | 2,738930 |
| 99 | 2,849246 |
| 137 | 2,850217 |
| 145 | 2,850227 |

2.5 Considerações Finais

Este capítulo descreveu uma visão geral sobre os paradigmas que formam a base da Computação Evolutiva, com ênfase nos Algoritmos Genéticos. Um exemplo da aplicação dos algoritmos genéticos para problemas de otimização foi apresentado, com o intuito de apontar os principais componentes de um AG: população inicial, representação genética, função de avaliação, operadores genéticos e parâmetros (tamanho da população, probabilidades de aplicação dos operadores, número de gerações). No próximo capítulo serão apresentados os conceitos fundamentais dos Sistemas Nebulosos.

Capítulo 3

Sistemas Nebulosos

3.1 Considerações Iniciais

Aristóteles, filósofo grego (384 – 322 a.C.), foi o fundador da lógica, e estabeleceu um conjunto de regras rígidas para que as conclusões pudessem ser aceitas logicamente. O emprego da lógica de Aristóteles é sintetizado em uma linha de raciocínio baseada em premissas e conclusões. Desde a época de Aristóteles, a Lógica Ocidental, assim chamada, tem sido binária, isto é, uma declaração é falsa ou verdadeira, não podendo ser ao mesmo tempo parcialmente verdadeira ou parcialmente falsa. *A Lógica Proposicional* e *a Lógica de Predicados*, conhecidas como *Lógicas Clássicas*, são compatíveis com esta suposição [Bittencourt 1998].

As *Lógicas Multivaloradas* [Rescher 1969] [Lukasiewicz 1970] surgem como uma opção para o tratamento de situações que não sejam tão determinísticas (somente verdadeiro ou falso). Na lógica multivalorada, o valor verdade é visto como graus de verdade pertencentes ao intervalo unitário $[0,1]$, ou seja, existem infinitos valores verdade [Pedrycz e Gomide 1998]. Ela possibilita o tratamento dos valores indeterminados, que não são totalmente verdadeiros ou totalmente falsos.

A *Lógica Nebulosa* foi introduzida por Lofti A. Zadeh em seu artigo [Zadeh 1965]. Neste artigo, Zadeh introduziu a teoria em que objetos – *conjuntos nebulosos (fuzzy sets)* – são conjuntos com limites não precisos. A pertinência em um conjunto nebuloso não é uma questão de afirmação ou negação, mas uma questão de *grau*. O significado do artigo de Zadeh não foi apenas um desafio à Teoria das Probabilidades como única forma para a representação de incerteza, mas também questionou as bases da Teoria das Probabilidades, ou seja, a lógica

aristotélica de dois valores. Se A é um conjunto nebuloso e x é um objeto relevante, a proposição " x é um elemento de A " não é necessariamente verdadeira ou falsa, como exigido pela lógica de dois valores, mas pode ser verdade apenas com um certo grau, isto é, o grau que representa a pertinência de x a A [Klir e Yuan 1995].

Com base na teoria dos *Conjuntos Nebulosos*, a Lógica Nebulosa tem a capacidade de tratar diversos tipos de imperfeições da informação e de representar conceitos vagos e incertos em linguagem natural.

Neste capítulo, a seção 3.2 apresenta a teoria de conjuntos nebulosos, as operações e definições básicas para conjuntos nebulosos, e também, as principais funções parametrizadas que são utilizadas para representá-los. Na seção 3.3, um importante conceito em conjuntos nebulosos é definido, o de variável lingüística. Em seguida, na seção 3.4, é descrito a forma de raciocínio empregado para se obter uma conclusão, quando se utiliza premissas e regras nebulosas. Encerrando o capítulo, a seção 3.5 discute as características de um sistema nebuloso e apresenta um modelo como exemplo.

3.2 Conjuntos Nebulosos

A partir da teoria apresentada por Zadeh [Zadeh 1965] e de vários outros trabalhos publicados [Kandel 1986] [Klir e Folger 1988] [Klir e Yuan 1995] [Pedrycz e Gomide 1998] [Zimmermann 1990], um conjunto nebuloso tem sido definido como uma coleção de objetos com valores de pertinência entre 0 (exclusão completa) e 1 (inclusão completa). Os valores de pertinência expressam os graus de compatibilidade dos objetos com as propriedades ou características distintas da coleção. Conjuntos nebulosos são formalmente definidos como:

Definição: Um conjunto nebuloso é caracterizado por uma *função de pertinência* que mapeia os elementos do domínio, espaço ou universo de discurso \mathbf{X} no intervalo $[0,1]$. Isto é,

$$A: \mathbf{X} \rightarrow [0,1] \quad (3.1)$$

Portanto, um conjunto nebuloso A em \mathbf{X} pode ser representado como um conjunto de pares ordenados de um elemento genérico $x \in \mathbf{X}$,

$$A = \{(A(x) / x) \mid x \in \mathbf{X}\} \quad (3.2)$$

onde $A(x)$ representa o grau de x em A .

De acordo com um exemplo descrito em [Zimmermann 1990], considere um vendedor de casas que deseja classificar as casas oferecidas aos seus clientes. Um indicador de conforto é dado pelo número de quartos presentes em cada casa. Seja $X = \{1, 2, 3, 4, \dots, 10\}$ o conjunto disponível de tipos de casas descritas por $x =$ número de quartos. Então, o conjunto nebuloso “casa confortável para uma família com quatro pessoas” pode ser descrito como:

$$A = \{(0.2 / 1), (0.5 / 2), (0.8 / 3), (1 / 4), (0.7 / 5), (0.3 / 6)\} \quad (3.3)$$

O primeiro elemento do par representa o grau de pertinência de x em A e o segundo elemento denota o valor de x . O conceito de função de pertinência desempenha um papel fundamental na teoria dos conjuntos nebulosos e será discutido na próxima seção.

3.2.1 Funções de Pertinência

Em princípio qualquer função da forma $A: X \rightarrow [0,1]$ descreve uma função de pertinência associada a um conjunto nebuloso A , que depende não somente do conceito que representa, mas também do contexto no qual é utilizado [Pedrycz e Gomide 1998]. Por exemplo, aplicando o conceito de *temperatura alta* para o clima e para um reator nuclear, este será necessariamente representado por conjuntos nebulosos totalmente diferentes [Klir e Yuan 1995].

Os gráficos das funções podem ter diferentes formas e podem ter algumas propriedades específicas (por exemplo, continuidade). Na prática é mais conveniente usar as funções de formato mais simples. Uma das formas mais utilizadas para representar conjuntos nebulosos é pelas famílias das funções parametrizadas. Além dos formatos tradicionais existe uma forma bastante utilizada em aplicações práticas: o conjunto unitário (*singleton*).

A seguir serão descritos alguns tipos de funções de pertinência parametrizadas e o conjunto unitário (*singleton*).

Função Triangular

A função Triangular é definida da seguinte forma:

$$A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } x \in [a, m] \\ \frac{b-x}{b-m}, & \text{se } x \in [m, b] \\ 0, & \text{se } x \geq b \end{cases} \quad (3.4)$$

onde m é um valor modal, e a e b denotam o limite inferior e superior, respectivamente, para valores não nulos de $A(x)$. Outra notação também pode ser usada: $A(x, a, m, b) = \max\{\min[(x-a)/(m-a), (b-x)/(b-m)], 0\}$. Na Figura 3.1 é ilustrado o gráfico de uma função triangular, sendo $a = -2,5$, $b = 2,5$ e $m = 0$.

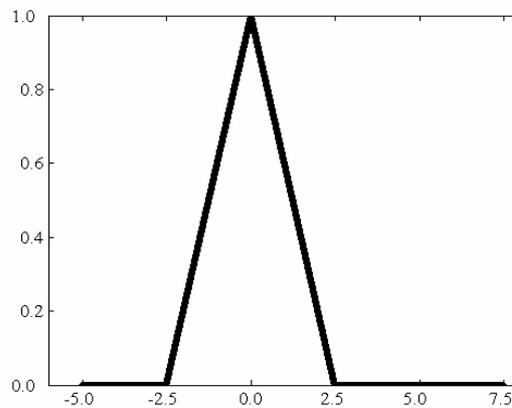


Figura 3.1: Gráfico de uma função triangular.

Função Γ

A função Γ é definida por:

$$A(x) = \begin{cases} 0, & \text{se } x \leq a \\ 1 - e^{-k(x-a)^2}, & \text{se } x > a \end{cases} \quad \text{ou} \quad A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{k(x-a)^2}{1+k(x-a)^2}, & \text{se } x > a \end{cases} \quad (3.5)$$

onde $k > 0$. Na Figura 3.2 é ilustrada a forma de uma função Γ , com os valores de $k=2$ e $a=1$.

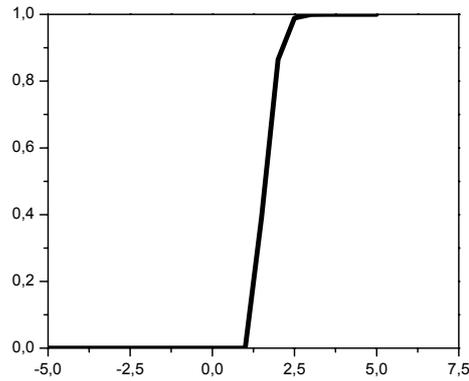


Figura 3.2: Gráfico de uma função Γ .

Função S

A função S é definida da seguinte forma:

$$A(x) = \begin{cases} 0, & \text{se } x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & \text{se } x \in [a,m] \\ 1-2\left(\frac{x-b}{b-a}\right)^2, & \text{se } x \in [m,b] \\ 1, & \text{se } x > b \end{cases} \quad (3.6)$$

O ponto $m = a+b/2$ é conhecido como um cruzamento da função S. Na Figura 3.3 é ilustrado o gráfico de uma função S, com os valores de $a = -1$, $b = 3$ e $m = 1$.

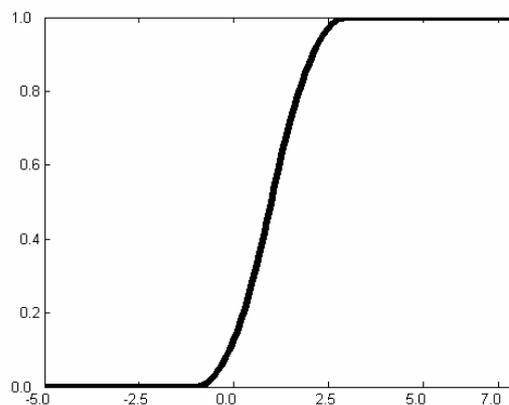


Figura 3.3: Gráfico de uma função S.

Função Trapezoidal

A função Trapezoidal é definida por:

$$A(x) = \begin{cases} 0, & \text{se } x < a \\ \frac{x-a}{m-a}, & \text{se } x \in [a,m] \\ 1, & \text{se } x \in [m,n] \\ \frac{b-x}{b-n}, & \text{se } x \in [n,b] \\ 0, & \text{se } x > b \end{cases} \quad (3.7)$$

Usando uma notação equivalente, obtemos: $A(x, a, m, b) = \max\{\min[(x-a)/(m-a), 1, (b-x)/(b-n)], 0\}$. Na Figura 3.4 é ilustrado o gráfico de uma função trapezoidal, sendo $a = -2,5$, $m = 0$, $n = 2,5$ e $b = 5,0$.

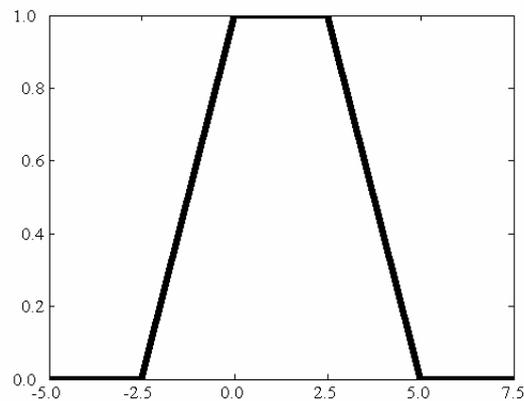


Figura 3.4: Gráfico de uma função trapezoidal.

Função Gaussiana

A função Gaussiana é definida por:

$$A(x) = e^{-k(x-m)^2}, \quad \text{onde } k > 0 \quad (3.8)$$

Na Figura 3.5 é ilustrada a forma de uma função gaussiana, com $m = 2$ e $k = 2$.

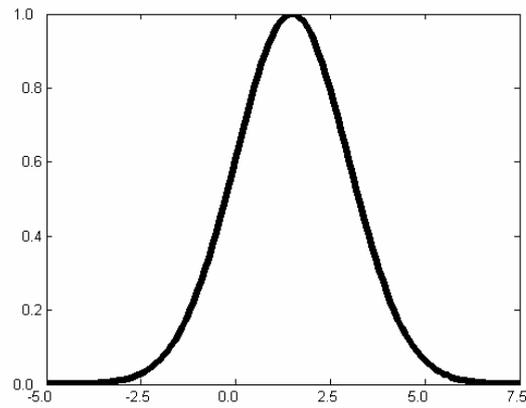


Figura 3.5: Gráfico de uma função gaussiana.

Conjunto Unitário (*Singleton*)

O conjunto unitário (singleton) é definido por:

$$A(x) = \begin{cases} h & \text{se } x = m \\ 0 & \text{caso contrário} \end{cases} \quad (3.9)$$

O formato de um conjunto unitário é dado por apenas um ponto, como mostra a Figura 3.6:

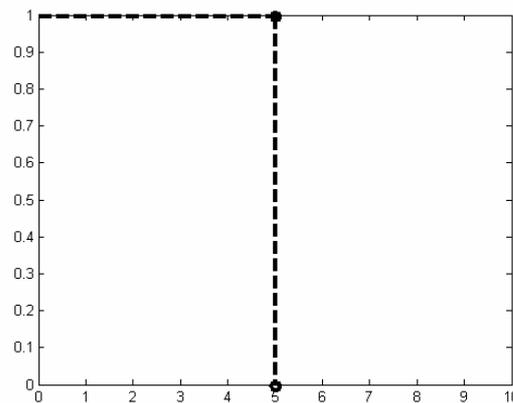


Figura 3.6: Gráfico de um conjunto unitário (singleton).

3.2.2 Definições Básicas em Conjuntos Nebulosos

Segundo [Klir e Yuan 1995], alguns conceitos básicos estão associados aos conjuntos nebulosos. Entre eles encontram-se: *Corte α* , *Conjunto de Níveis*, *Suporte*, *Altura* e

Normalização. A seguir tais conceitos serão apresentados de forma sintética, supondo que A é um conjunto nebuloso sobre o conjunto base \mathbf{X} .

Corte α

O Corte α (${}^\alpha A$) de um conjunto nebuloso A é o conjunto de todos os elementos do universo \mathbf{X} com grau de pertinência em A maior ou igual a α . O Corte α forte (${}^{\alpha+}A$) contém todos os elementos do universo \mathbf{X} com grau maior que α , onde $\alpha \in [0,1]$.

$${}^\alpha A = \{x \in \mathbf{X} \mid A(x) \geq \alpha\} \quad (3.10)$$

$${}^{\alpha+} A = \{x \in \mathbf{X} \mid A(x) > \alpha\} \quad (3.11)$$

Conjunto de Níveis

O conjunto de níveis (Λ) de um conjunto nebuloso A correspondente a um conjunto que contém todos os valores $\alpha \in [0,1]$ e que representam Cortes α de A distintos. O conjunto de níveis do conjunto nebuloso A é representado formalmente por:

$$\Lambda(A) = \{\alpha \mid A(x) = \alpha \text{ para algum } x \in \mathbf{X}\} \quad (3.12)$$

Suporte

O suporte de um conjunto nebuloso A , em um universo \mathbf{X} , é o conjunto que contém todos os elementos de \mathbf{X} que possuem grau de pertinência em A diferente de zero. Claramente, o suporte de A é exatamente o mesmo que o Corte α forte de A para $\alpha = 0$.

$${}^{0+} A = \{x \in \mathbf{X} \mid A(x) > 0\} \quad (3.13)$$

Altura

A altura $h(A)$ de um conjunto *nebuloso* A corresponde ao maior grau de pertinência entre todos os elementos do conjunto. Ou seja, o supremo da função de pertinência de A , como expresso na equação (3.14).

$$h(A) = \sup_{x \in X} A(x) \quad (3.14)$$

Normalização

Um conjunto nebuloso A é dito *Normal* quando a sua altura é igual a um, ou seja, pelo menos um grau de pertinência dos elementos do conjunto atinge valor máximo, enquanto que os conjuntos que não possuem altura igual a um são chamados de *subnormais*. Portanto, A é normal se $h(A) = 1$ e A é subnormal se $h(A) < 1$.

Caso um conjunto nebuloso possua apenas um elemento com grau de pertinência igual a um, este elemento é denominado *protótipo* do conjunto.

3.2.3 Operações Básicas com Conjuntos Nebulosos

Com base na teoria dos conjuntos clássicos, Zadeh [Zadeh 1965] definiu as operações de união, interseção e complemento para conjuntos nebulosos, a partir da função de pertinência.

Sejam A e B dois conjuntos nebulosos definidos no universo U com suas funções de pertinência μ_A e μ_B , respectivamente. Na teoria de conjuntos nebulosos, a intersecção é implementada por uma família de operadores denominados *T-normas* e a união é implementada por uma família de operadores denominados *T-conormas* [Klir e Yuan 1995].

União

A função de pertinência $\mu_A \cup \mu_B$ é definida ponto a ponto para todo $u \in U$ por:

$$\mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \} \quad \text{ou} \quad \mu_{A \cup B}(u) = \mu_A(u) \vee \mu_B(u) \quad (3.15)$$

Intersecção

A função de pertinência $\mu_A \cap \mu_B$ é definida ponto a ponto para todo $u \in U$ por:

$$\mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \} \quad \text{ou} \quad \mu_{A \cap B}(u) = \mu_A(u) \wedge \mu_B(u) \quad (3.16)$$

Complemento

A função de pertinência $\mu_{\neg A}$ do complemento de um conjunto nebuloso A é definida ponto a ponto para todo $u \in U$ por:

$$\mu_{\neg A} = 1 - \mu_A(u) \quad (3.17)$$

As operações de união e intersecção apresentadas anteriormente podem ser generalizadas substituindo-se os operadores de mínimo e máximo por outros operadores das classes T-normas e T-conormas [Klir e Yuan 1995], respectivamente.

3.3 Variáveis Lingüísticas

Uma variável lingüística pode ser definida, de uma maneira informal, como uma variável cujos valores são palavras ou sentenças, ao invés de números [Pedrycz e Gomide 1998]. Zadeh [Zadeh 1975], definiu formalmente uma variável lingüística sendo como uma quintupla denotada por:

$$\langle X, T(X), U, G, M \rangle \quad (3.18)$$

onde,

- X é o nome da variável, cuja variável base é x ;
- $T(X)$ é o conjunto de termos lingüísticos. Cada elemento de $T(X)$ representa um rótulo L dos termos que a variável X pode assumir;
- U é o universo de discurso da variável lingüística X ;
- G é a gramática para a geração dos termos ou rótulos;
- M é a regra semântica que associa a cada rótulo L , um conjunto nebuloso no universo U , representando o seu significado $M(L)$. Por exemplo, considere a variável lingüística (ver Figura 3.8) *velocidade* ($X = velocidade$) com universo $U = [0,150]$ e variável base $x \in X$. Um possível conjunto de termos associados à variável velocidade poderia ser

$$T(velocidade) = \{\text{muito baixa, baixa, média, alta, muito alta}\}.$$

Seja $F(X)$ uma família de conjuntos nebulosos definidos no universo X . Então, a regra M mapeia $T(X) \rightarrow F(X)$, ou seja, associa a cada rótulo do conjunto de termos $T(X)$, um conjunto nebuloso definido em X .

A gramática G define como os termos primários {baixa, média, alta} serão associados aos modificadores {muito, pouco, maior, menor, ou, não} para formar os nomes dos termos não-primários. Usualmente, as funções de pertinência associadas às variáveis lingüísticas através de termos primários (baixa, média, alta), têm formatos conhecidos: triangular, trapezoidal, Gaussiana. Os conjuntos nebulosos associados aos termos não-primários (muito-alta, pouco-alta, não-alta) podem ser derivados através do uso de modificadores pré-especificados [Delgado 2002] (para maiores detalhes, ver [Pedrycz e Gomide 1998]).

Os números de valores lingüísticos definem a *granularidade*, isto é, a especificação e distribuição dos termos lingüísticos e, por conseguinte, a partição nebulosa do universo correspondente. Um número pequeno de termos lingüísticos define uma partição esparsa ou grossa, ao passo que um número maior resulta numa partição fina [Pedrycz e Gomide 1998]. A Figura 3.7 mostra dois exemplos de possíveis partições para a variável lingüística *velocidade*, definida em U .

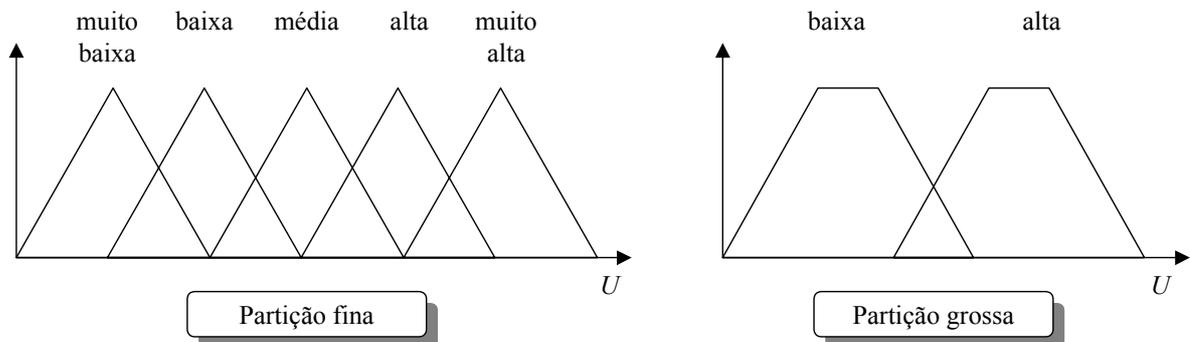


Figura 3.7: Exemplos de partição nebulosa de diferentes granularidades.

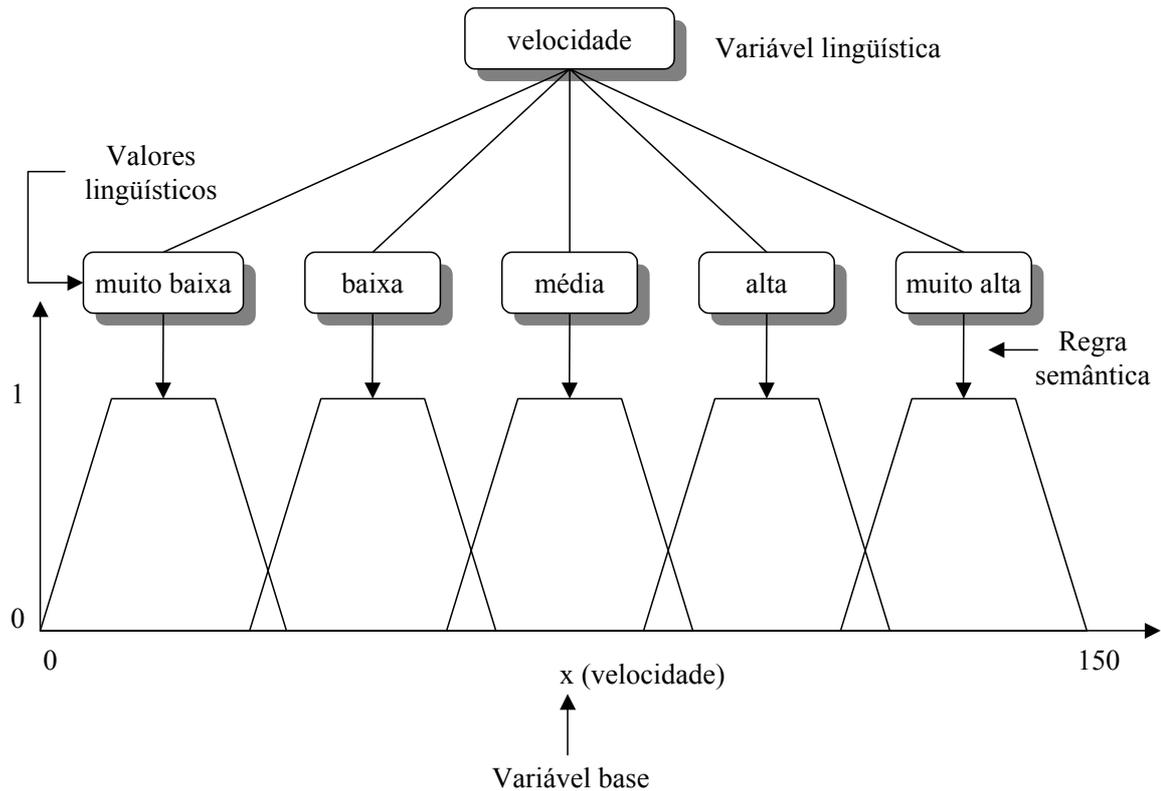


Figura 3.8: Variável linguística velocidade.

3.4 Raciocínio Aproximado

Na lógica tradicional de dois valores, as principais ferramentas de raciocínio são as tautologias, como por exemplo, o *modus ponens*, isto é, $(A \wedge (A \Rightarrow B)) \Rightarrow B$ ou

| | |
|-------------|------------------|
| Premissa: | A é verdade |
| Implicação: | Se A então B |
| Conclusão: | B é verdade |

A e B são sentenças ou proposições e A na premissa é idêntico ao A da implicação assim como B na sentença condicional é idêntico ao B da conclusão. Como exemplo, considere que a premissa é “tomate é vermelho” e a implicação “se o tomate é vermelho então ele está maduro”. Pode-se então concluir que “o tomate está maduro”.

Segundo [Zimmermann 1990], duas óbvias generalizações do *modus ponens* são: (1) permitir que sentenças fossem caracterizadas por conjuntos nebulosos e (2) Relaxar levemente a exigência de igualdade dos B 's da implicação e da conclusão.

Esta versão generalizada do modus ponens foi então chamada de “*modus ponens generalizado*” [Zadeh 1975]. Seja A, A', B, B' conjuntos nebulosos. A inferência do modus ponens generalizado ocorre da seguinte forma:

$$\begin{array}{ll} \text{Premissa:} & x \text{ é } A' \\ \text{Implicação:} & \frac{\text{Se } x \text{ é } A \text{ então } y \text{ é } B}{y \text{ é } B'} \\ \text{Conclusão:} & \end{array}$$

De acordo com o exemplo de [Mizumoto e Zimmermann 1982], temos:

$$\begin{array}{ll} \text{Premissa:} & \text{Este tomate está muito vermelho} \\ \text{Implicação:} & \frac{\text{Se um tomate é vermelho então o tomate está maduro}}{\text{Este tomate está muito maduro}} \\ \text{Conclusão:} & \end{array}$$

Em [Zadeh 1975], foi sugerido a *regra de inferência composicional* para o tipo de inferência apresentada pelo modus ponens generalizado. A seguir será descrito como obter uma conclusão a partir de premissas e implicações nebulosas, considerando a regra de inferência composicional de Zadeh.

Regra de Inferência Composicional

Para um bom entendimento da regra de inferência composicional, conceitos como relações, implicações e proposições nebulosas, normas-t, normas-s, são necessários (para mais detalhes, ver [Klir e Yuan 1995] [Pedrycz e Gomide 1998] [Zimmermann 1990]). A seguir segue a definição proposta por [Zadeh 1975]:

Assume-se que R é uma relação nebulosa em $\mathbf{X} \times \mathbf{Y}$, e A, B são conjuntos nebulosos em \mathbf{X} e \mathbf{Y} , respectivamente. Então, se R e A são dados, B pode ser obtido pela equação:

$$B(y) = \sup \min[A(x), R(x,y)] \quad (3.19)$$

para todo $y \in \mathbf{Y}$. Esta equação também pode ser escrita da seguinte forma:

$$B = A \circ R \quad (3.20)$$

A seguinte interpretação da inferência pode ser considerada de acordo com [Zimmermann 1990]:

x é pequeno
 x e y são aproximadamente iguais

 y é mais ou menos pequeno

3.5 Sistemas Nebulosos

Em geral, um *sistema nebuloso* é qualquer sistema cujas variáveis (ou pelo menos, algumas delas) assumem estados que são conjuntos nebulosos. Para cada variável, os conjuntos nebulosos são definidos em algum universo relevante, o qual é freqüentemente um intervalo de números reais. Neste caso, os conjuntos nebulosos são números nebulosos, e as variáveis associadas são variáveis lingüísticas [Klir e Yuan 1995].

O sistema nebuloso tem como objetivo modelar o raciocínio aproximado, permitindo o desenvolvimento de sistemas que imitem a habilidade humana de tomar decisões racionais em um ambiente de incerteza e imprecisão. Deste modo, a lógica nebulosa é uma ferramenta capaz de capturar informações imprecisas, em linguagem natural, e convertê-las em uma forma numérica [Mendel 1995].

A maneira mais comum de armazenar informações em uma base de conhecimento de um sistema nebuloso é a representação por meio de *regras nebulosas* que estabelecem relações entre diversas variáveis nebulosas e um ou mais conjuntos nebulosos. Estas regras possuem o seguinte formato:

$$\mathbf{Se} \langle \textit{antecedente} \rangle \mathbf{Então} \langle \textit{conseqüente} \rangle. \quad (3.21)$$

Os antecedentes descrevem uma condição (premissa), enquanto a parte conseqüente descreve uma conclusão ou uma ação que pode ser esboçada quando as premissas se verificam. Os antecedentes definem uma região nebulosa no espaço das variáveis de entrada do sistema. Já os conseqüentes descrevem uma região no espaço das variáveis de saída do sistema, qual seja a sua conclusão/ação.

A estrutura básica de um sistema nebuloso é constituída por (ver Figura 3.9): um módulo de *Fuzificação*, que tem como função converter os valores de entrada do sistema (números reais) para termos lingüísticos representados por conjuntos nebulosos; uma *Base de Conhecimento (BC)*, onde todo o conhecimento sobre o domínio do problema em questão é armazenado. A BC é formada por uma *Base de Regras (BR)*, que contém o conjunto de regras nebulosas e por uma *Base de Dados (BD)*, que define as funções de pertinência usadas nas

regras nebulosas; um *Motor de Inferência*, que é responsável pelo desenvolvimento do raciocínio nebuloso baseado no conhecimento representado na BC; e por um módulo de *Desfuzificação*, que realiza uma transformação da resposta do sistema nebuloso, a qual está representada por um conjunto nebuloso, em uma resposta não-nebulosa.

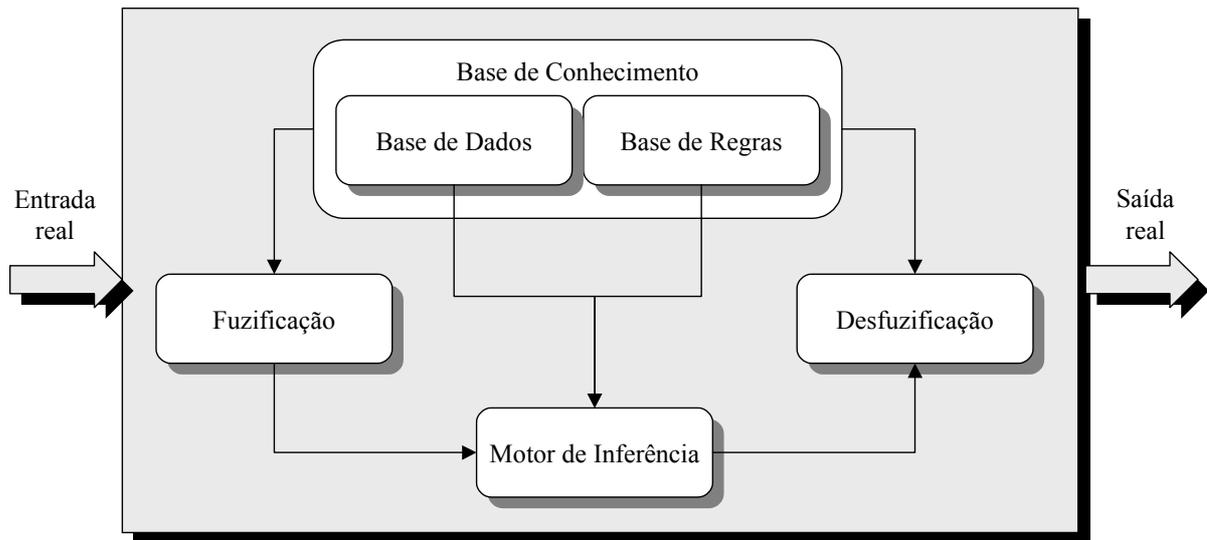


Figura 3.9: Estrutura básica de um sistema nebuloso.

Existem vários modelos de sistemas nebulosos. Na maioria dos casos, o antecedente é formado por proposições lingüísticas e a distinção entre os modelos se dá no conseqüente das regras nebulosas. Entre os modelos mais conhecidos podemos destacar [Delgado 2002]:

- modelo de **Mamdani** [Mamdani e Assilian 1975], que utiliza conjuntos nebulosos também nos conseqüentes das regras nebulosas. A saída final é representada por um conjunto nebuloso resultante da agregação da saída inferida de cada regra. Para se obter uma saída final não nebulosa adota-se um dos métodos de transformação da saída nebulosa em não-nebulosa.
- modelo de **Takagi-Sugeno** [Takagi e Sugeno 1983], no qual o conseqüente é representado por uma função das variáveis de entrada. A saída final é obtida pela média ponderada das saídas inferidas de cada regra. Os coeficientes da ponderação são dados pelos graus de ativação das respectivas regras.
- modelo de **Tsukamoto** [Tsukamoto 1979], que utiliza funções de pertinência monotônicas no conseqüente. Assim como no modelo Takagi-Sugeno, é inferido um

valor não-nebuloso induzido pelo nível de ativação da regra. A saída final é obtida por média ponderada das saídas inferidas de cada regra.

3.5.1 Sistema Nebuloso Mamdani

De acordo com [Rezende 2003], serão detalhados todos os processos envolvidos no sistema nebuloso Mamdani. A característica básica do modelo de Mamdani é o fato de utilizar conjuntos nebulosos nos antecedentes e conseqüentes das regras nebulosas. Uma regra típica deste modelo possui a forma como descrito em (3.21).

A estrutura de um sistema nebuloso Mamdani inclui todos os módulos ilustrados na Figura 3.9. Assim, o motor de inferência recebe os valores lingüísticos provenientes do módulo de fuzificação, processa as regras existentes na base de regras e gera um conjunto nebuloso de saída para o módulo de desfuzificação, a partir da composição das saídas de todas as regras disparadas. Por uma regra disparada, entende-se uma regra cujo processamento do antecedente para as entradas atuais gerou graus de pertinência não-nulos, ou seja, a relação nebulosa entre as entradas e os termos primários do antecedente é maior que zero.

A obtenção da conclusão de cada regra depende da semântica escolhida e dos resultados da combinação e agregação dos antecedentes. Da agregação das conclusões inferidas de cada regra, resulta um conjunto nebuloso. A inferência tradicionalmente utilizada pelo modelo de Mamdani é chamada de inferência Máx-Mín (ver detalhes em [Klir e Yuan 1995]). Ela utiliza as operações de união e intersecção entre conjuntos. Considerando uma regra nebulosa genérica como mostrada em (3.22), o método de inferência Máx-Mín proposto por Mamdani será detalhado a seguir.

$$\text{Se } x_1=A_i \text{ E } x_2=A_j \text{ E...E } x_p=A_j \text{ Então } y_1=B_i \text{ E } y_2=B_m \quad (3.22)$$

Em 3.21, x_i são as entradas do sistema (variáveis lingüísticas), A_1, \dots, A_j são conjuntos nebulosos nos universos definidos de cada variável de entrada, y_1 e y_2 são variáveis de saída (variáveis lingüísticas) e B_1, \dots, B_m são conjuntos nebulosos nos universos definidos de cada variável de saída.

Durante o processo de fuzificação, os antecedentes de cada regra são processados por meio da intersecção entre os graus de pertinência das entradas atuais nos termos primários definidos em cada uma. Este processo gera um grau de disparo para cada regra de produção. Em outras palavras, é calculado para a k -ésima regra da BR um coeficiente de disparo $D^{(k)}$,

definido em 3.22, onde os índices k nos conjuntos nebulosos denotam os termos primários que compõem a regra k na BR. Este processamento transforma informações quantitativas em informações qualitativas, e é chamado *fuzificação*.

$$D^{(k)} = T[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)] = \min[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)] \quad (3.23)$$

Todas as regras para as quais o coeficiente de disparo for maior que zero são ditas regras que dispararam para as entradas atuais. Isto quer dizer que elas vão contribuir para o cálculo da saída correspondente do sistema de inferência. Os coeficientes de disparo, por sua vez, vão limitar os valores máximos dos conjuntos nebulosos de saída gerados por estas regras. Finalmente, uma operação global de união vai compor um conjunto nebuloso para cada variável de saída, contendo informações sobre todas as regras disparadas para as entradas atuais. Em (3.24), é mostrada a composição deste conjunto para o caso da saída y_2 , apresentada na regra de produção nebulosa genérica anterior. O universo de discurso desta variável de saída é composto pelos elementos $y \in U_{y_2}$.

$$\mu_{B_i}(y) = S_{k=1..n}[T(D^{(k)}, \mu_{B_i}(y))] = \max_{k=1..n}[\min(D^{(k)}, \mu_{B_i}(y))] \quad \forall y \in U_{y_2} \quad (3.24)$$

O processo de inferência acima descrito transforma uma informação qualitativa em outra informação qualitativa, por meio de uma conversão. O conjunto nebuloso gerado durante o processo de inferência pode então ser utilizado diretamente em um diagnóstico qualitativo de tomada de decisão, ou será convertido em um valor escalar proporcional para atuação externa por meio de atuadores convencionais, no caso de um sistema de controle. A conversão nebuloso \rightarrow escalar transforma informações qualitativas em uma informação quantitativa, conhecido como *desfuzificação*.

Existem vários métodos utilizados para converter um valor nebuloso em não-nebuloso, como por exemplo, o método do *centro de massa*. O método do centro de massa calcula, para um dado conjunto nebuloso, a abscissa (no universo de discurso definido para a variável em questão) do ponto de centro de massa correspondente, e a utiliza como valor escalar de saída. A expressão analítica da implementação deste método é mostrada em (3.25).

$$\hat{y}_2 = \frac{\sum_{y \in U_{y_2}} y \cdot \mu_{B_i}(y)}{\sum_{y \in U_{y_2}} \mu_{B_i}(y)} \quad (3.25)$$

Suponha um sistema nebuloso do tipo Mamdani, composto por duas regras na forma:

$$R_i : \text{Se } x_1 = A_1^i \text{ E } x_2 = A_2^i \text{ Então } y = B^i, i = 1, 2$$

onde x_1 e x_2 são as variáveis de entrada e A_1^i e A_2^i são conjuntos nebulosos nos universos X_1 e X_2 , respectivamente. A Figura 3.10 ilustra o processo de inferência para duas entradas não-nebulosas (representadas por valores em posições arbitrárias u_1 e u_2), utilizando o Máx-Min.

Analisando a Figura 3.10, na primeira regra o método de inferência Mamdani selecionou o menor valor de pertinência obtido para os valores de entrada u_1 e u_2 nos conjuntos A_1^1 e A_2^1 , respectivamente, gerando o conjunto nebuloso B^1 . Do mesmo modo com os conjuntos A_1^2 e A_2^2 , o conjunto nebuloso resultante foi B^2 . O conjunto nebuloso de saída B será a união dos conjuntos B^1 e B^2 .

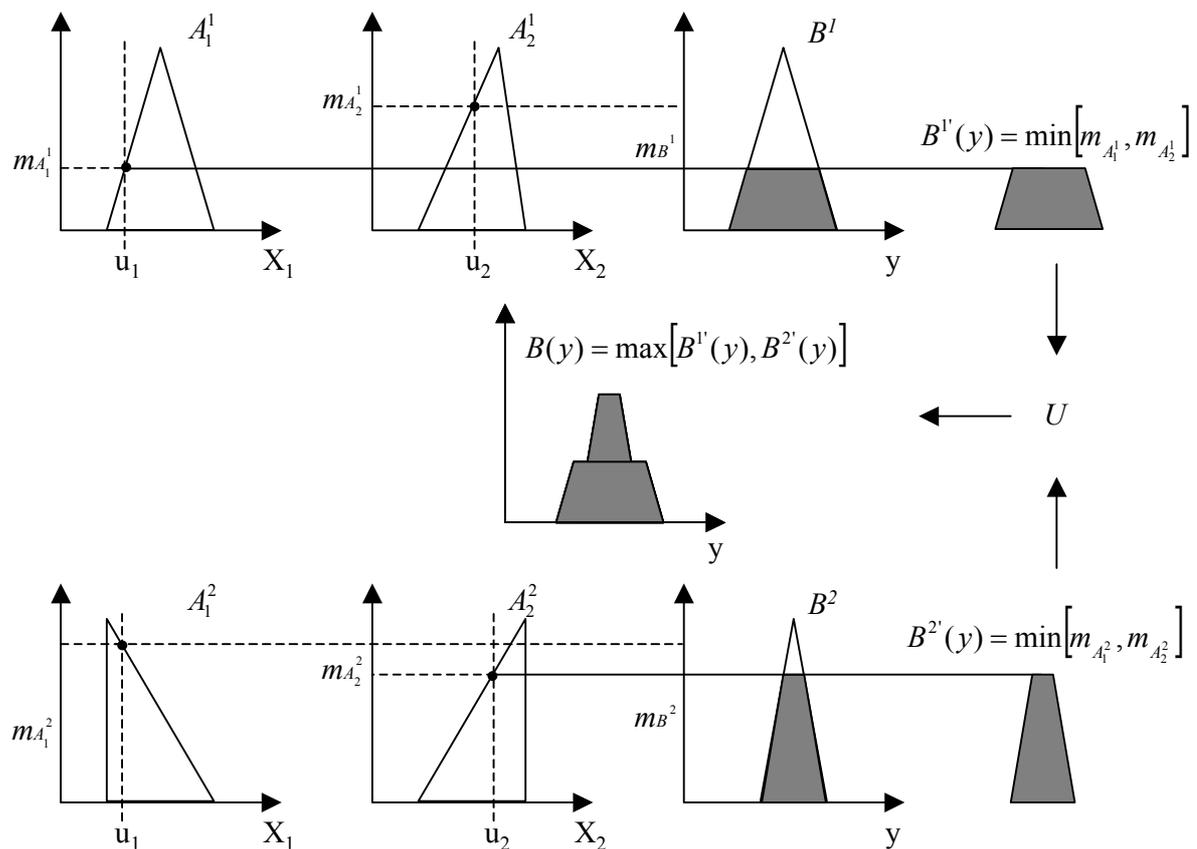


Figura 3.10: Inferência no modelo Mamdani.

3.6 Considerações Finais

O objetivo deste capítulo foi proporcionar um entendimento básico da aplicação da lógica nebulosa e dos conjuntos nebulosos como ferramentas para o tratamento do conhecimento vago e incerto, através de valores lingüísticos, a fim de uma melhor interpretação pelos seres humanos.

No próximo capítulo, será apresentado o projeto de pesquisa deste trabalho, o qual faz a união de dois paradigmas da Computação Flexível: Algoritmos Genéticos e Lógica Nebulosa.

Capítulo 4

Aprendizado Genético da BC de Sistemas Nebulosos

4.1 Considerações Iniciais

Em Ciência da Computação e Engenharia, o termo “conhecimento” significa a informação que um sistema computacional requer, para que possa se comportar inteligentemente [Pedrycz e Gomide 1998]. Devido à sua grande influência no comportamento de um sistema computacional, existem várias formas de representá-lo, como por exemplo, *frames*, *scripts*, *rede semântica* e *regras de produção se-então* [Luger e Stubblefield 1989].

Os sistemas nebulosos que utilizam regras nebulosas para representar o conhecimento do domínio do problema são denominados *Sistemas Baseados em Regras Nebulosas (SBRN)*. Uma das principais razões para o uso de regras nebulosas é a habilidade que tais sistemas possuem em representar o conhecimento vago e incerto e a facilidade de expressar o comportamento do sistema em uma linguagem de fácil compreensão para os seres humanos.

Existem duas maneiras típicas de aquisição do conhecimento para um SBRN. Uma delas é a extração do conhecimento a partir de um especialista humano, a qual é conhecida como abordagem direta para aquisição do conhecimento. No entanto, muitas vezes o especialista não consegue detalhar todo o seu conhecimento sobre o domínio do problema ou não há um especialista para o problema abordado. Em outros casos, o conhecimento está disponível como exemplos inseridos em bases de dados de dimensão elevada, sendo necessário um processo de extração, pois nem sempre esta é uma tarefa fácil para um

operador humano. Portanto, a segunda maneira de aquisição do conhecimento utiliza métodos que automaticamente extraem este conhecimento a partir de informações numéricas que representam amostras ou exemplos do problema. Entre esses pode-se destacar *redes neurais* [Nauck e Cruse 1997], *algoritmos de clusterização* [Liao et al. 1997], *métodos baseados em gradiente* [Nomura et al. 1992] e *algoritmos genéticos* [Cordón et al. 2001c].

Em particular, um alto número de publicações tem explorado o uso de algoritmos genéticos para o projeto de sistemas nebulosos durante a última década. O projeto automático de sistemas nebulosos pode ser considerado em muitos casos como um processo de otimização ou busca em um espaço de soluções em potencial (usualmente representando definições parciais ou completas das bases de conhecimento). Algoritmos genéticos são bem conhecidos e amplamente usados como técnica de busca, por sua habilidade de explorar espaços complexos e irregulares. A combinação de lógica nebulosa e algoritmos genéticos resultou em sistemas denominados *Sistemas Genéticos Nebulosos (SGN)* [Cordón et al. 2004] [Cordón et al. 2001b] [Cordón et al. 2001a].

Um aspecto fundamental que é considerado neste trabalho é o balanço entre duas importantes características da base de conhecimento: acuidade e a interpretabilidade. A acuidade se diz respeito à eficácia da BC, enquanto que a interpretabilidade, como o próprio nome já diz, representa a facilidade de interpretação por um operador humano. A dificuldade de se realizar este balanço está no fato de que estas duas características são contraditórias, ou seja, ao manter um alto grau de interpretabilidade da BC, o desempenho em termos da acuidade está sujeito a alguma degradação e vice-versa.

Neste capítulo, a seção 4.2 apresenta algumas abordagens que aplicam algoritmos genéticos na construção da BC de sistemas nebulosos. Na seção 4.3 são discutidas duas formas de aprendizado genético da BD – otimização e construção genética da BD –, as quais foram objeto de pesquisa deste trabalho. Da seção 4.3.3 até a seção 4.3.8, são apresentados os componentes do algoritmo genético implementado, a forma das regras nebulosas e o raciocínio nebuloso empregado.

4.2 Abordagens Genéticas no Projeto Automático de Sistemas Nebulosos

Um subconjunto dos *SGN* são os *Sistemas Genéticos Nebulosos Baseados em Regras (SGNBR)*, os quais empregam um algoritmo genético para aprender ou ajustar diferentes componentes de um *SBRN*. Em outras palavras, o ponto chave da utilização de um processo de aprendizado evolutivo é automatizar a geração da base de conhecimento, a qual pode ser considerada como um processo de otimização ou busca [Cordón et al. 2004] [Cordón et al. 2001a].

O primeiro passo no projeto de um *SGNBR* é decidir qual parte da base de conhecimento será otimizada pelo algoritmo genético. Esta decisão depende de dois objetivos conflitantes: a *dimensionalidade* e a *eficiência* da busca. Um espaço de busca de pequena dimensão resulta em um processo de aprendizado mais rápido e simples, mas as soluções podem não ser ótimas. Um grande e completo espaço de busca que compreenda a BC inteira e que possua uma dimensionalidade maior, é preferível no sentido de se obter soluções ótimas, mas o processo de busca pode tornar-se ineficiente e lento [Cordón et al. 2004] [Cordón et al. 2001a].

As abordagens que combinam algoritmos genéticos para geração de apenas um componente da base de conhecimento com outros métodos mais simples e rápidos, são vistas como uma opção viável, no sentido de simplificar o processo de construção do sistema pela redução da dimensionalidade do espaço de busca, sem prejuízo do desempenho final.

A seguir são descritas as diferentes possibilidades de projeto automático de *SGN*.

Geração Genética da Base de Regras

A maioria dos trabalhos encontrados na literatura focalizam o aprendizado da base de regras usando uma base de dados predefinida. Usualmente esta definição é feita escolhendo-se um número de valores lingüísticos para as variáveis lingüísticas (um número ímpar entre três e nove) e distribuindo uniformemente os valores, que são rótulos dos conjuntos nebulosos, no domínio de cada variável [Casillas et al. 2000] [Cordón e Herrera 2000] [Ishibuchi et al.

1994] [Wang e Mendel 1992] [González e Pérez 1999] [Hoffmann e Pfister 1997] [Hwang e Thompson 1994]. Na Figura 4.1 é ilustrado este tipo de aprendizado.

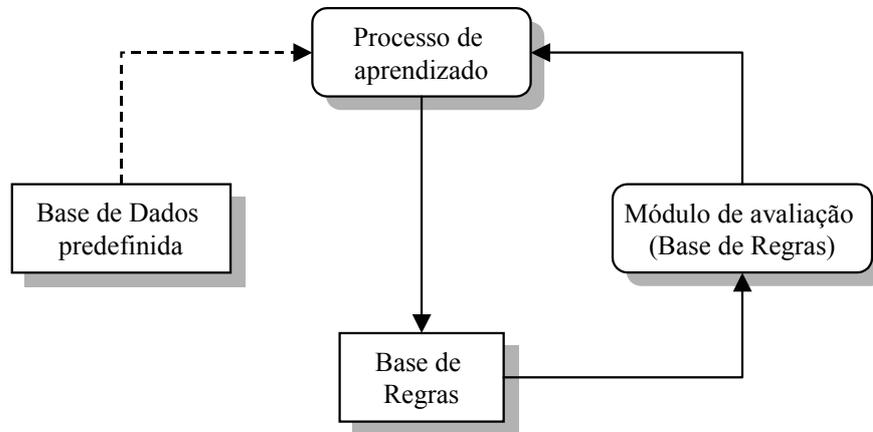


Figura 4.1: Derivação genética da BR.

Quando se trata especificamente da construção genética de regras a partir de uma BD definida previamente, existem três abordagens de aplicação dos algoritmos genéticos ao processo de aprendizado [Cordón et al. 2001a]:

- *Michigan* [Bonarini 1996] [Ishibuchi et al. 1999b]: dada uma população composta por seus cromossomos (indivíduos ou regras), uma solução candidata para o problema em questão é representada por todos os indivíduos da população, ou seja, cada cromossomo corresponde a uma regra, o qual é denominado *classificador*. Este modelo é geralmente referenciado na literatura como *Sistemas Classificadores*.
- *Pittsburgh* [Hoffmann e Pfister 1997]: na abordagem de Pittsburgh uma solução candidata para o problema em questão é representada por apenas um indivíduo da população, pois este representa toda a base de regras, ao contrário da Michigan onde cada indivíduo representa uma única regra. Portanto, a abordagem de Michigan procura a solução em uma população completa e a de Pittsburgh procura a solução em um indivíduo.
- *Iterativa* [Cordón e Herrera 1997] [González e Pérez 1999]: cada cromossomo representa uma única regra, da mesma forma que a abordagem de Michigan. Porém, a cada geração, um processo iterativo é aplicado e apenas o melhor indivíduo é considerado como uma possível solução, descartando os demais cromossomos da população.

Otimização Genética da Base de Dados

Com o objetivo de melhorar o desempenho do *SBRN*, algumas abordagens tentam ajustar a base de dados definida previamente, uma vez que a base de regras tenha sido construída. Esta base de regras é obtida por um processo de aprendizado não genético utilizando-se a definição inicial da BD. Ao chegar em uma BR definitiva, um outro processo de aprendizado, o qual é realizado por um algoritmo genético, é aplicado à BD inicial com o objetivo de encontrar uma codificação que melhor se adapta à BR definitiva. Este processo somente ajusta as formas das funções de pertinência e não o número de termos lingüísticos em cada partição nebulosa, o qual permanece fixo desde o início [Bonissone et al. 1996] [Cordón e Herrera 1997] [Herrera et al. 1995] [Karr 1991]. Uma representação gráfica deste tipo de aprendizagem é mostrada na Figura 4.2.

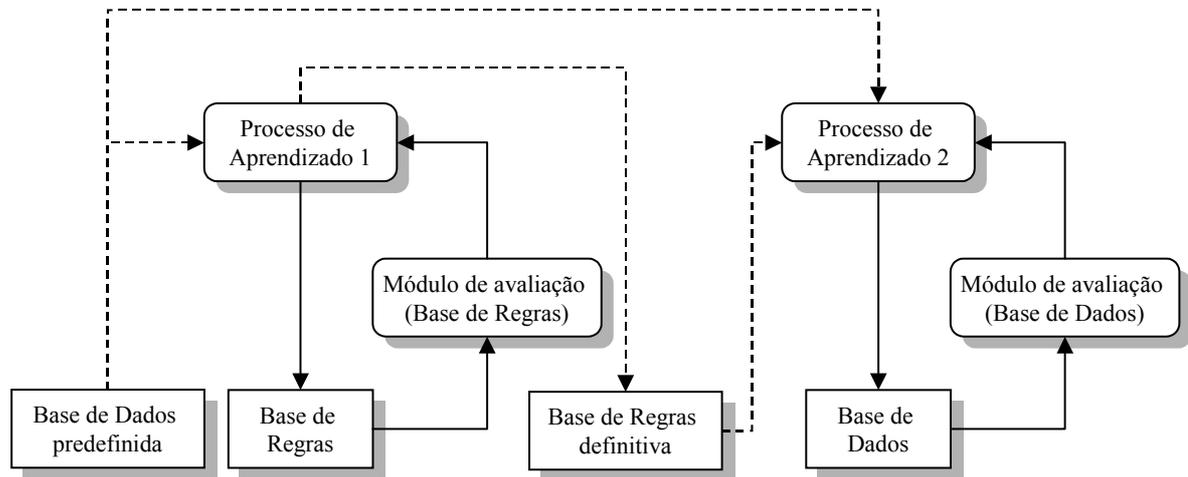


Figura 4.2: Processo de ajuste de uma BD.

Aprendizado Genético da Base de Conhecimento

Outras abordagens propõem a construção dos dois componentes da base de conhecimento ao mesmo tempo. Trabalhando desta forma, há possibilidades de gerar melhores definições da BC. No entanto, o espaço de busca aumenta e o processo de aprendizado torna-se difícil e lento [Carse et al. 1996] [Cordón e Herrera 2001] [Ishibuchi e Murata 1996] [Filipic e Juricic 1996] [Lee e Takagi 1993] [Magdalena e Monasterio 1997] [Parodi e Bonelli 1993] [Velasco 1998]. Na Figura 4.3 é ilustrado este tipo de aprendizado.

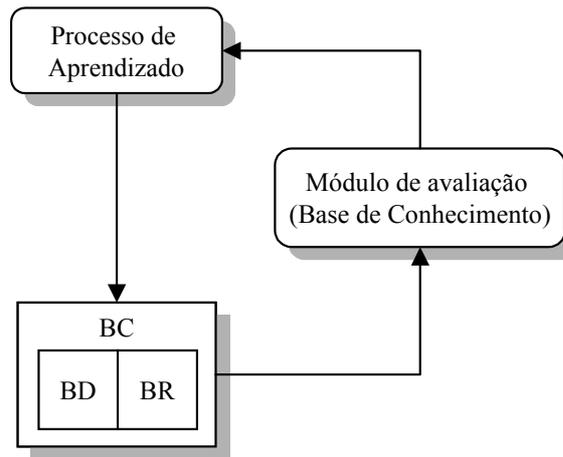


Figura 4.3: Aprendizagem genética dos componentes da BC simultaneamente.

Construção Genética da BD

Uma tendência mais recente propõe a construção genética da BD, considerando dois diferentes processos para ambos os componentes (BR e BD), baseado na aprendizagem da BD *a priori*. Esta abordagem utiliza um algoritmo genético para aprender a BD e um outro método para a construção da BR, como por exemplo o método de Wang e Mendel [Wang e Mendel 1992] (detalhes do algoritmo no Apêndice A). Na Figura 4.4 é mostrada esta abordagem.

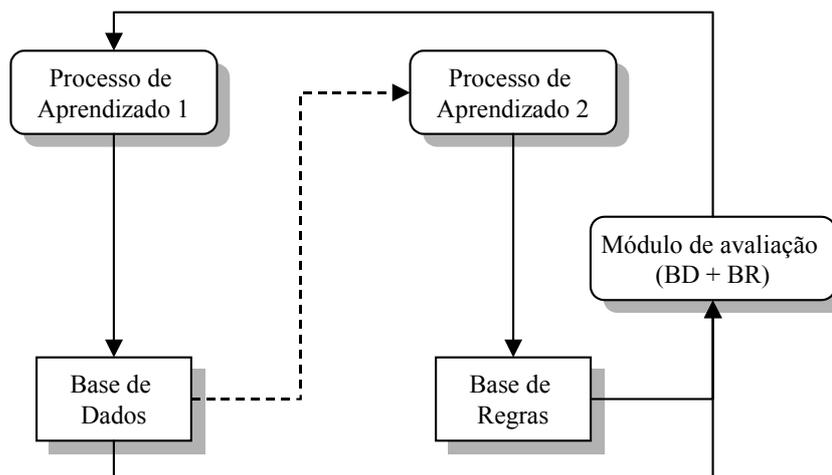


Figura 4.4: Definição genética da BD.

Note que este tipo de operação envolve um particionamento do problema da aprendizagem da BC, pois é composto por processos independentes onde a busca é feita em espaços mais simples, ou seja, um processo define a BD e um outro processo gera a BR.

Entre as diferentes formas de aprendizado genético da BC de SBRN apresentadas, a otimização e construção genética da BD serão os tópicos abordados neste trabalho, os quais serão detalhados na próxima seção.

4.3 Aprendizado Genético de Funções de Pertinência na Modelagem Nebulosa

O principal foco de pesquisa deste trabalho é investigação das abordagens de modelagens automáticas de sistemas nebulosos através de algoritmos genéticos para a definição e sintonia de conjuntos nebulosos que compõem as partições nebulosas dos domínios envolvidos, considerando a manutenção da interpretabilidade dos valores lingüísticos.

São investigadas as abordagens que aplicam algoritmos genéticos apenas nas funções de pertinência, tanto pela maneira mais tradicional, de sintonizar as funções já definidas depois da definição das regras, como pela abordagem mais recente, de construir as funções de pertinência antes da definição das regras. Esse tipo de investigação supõe o uso de outros métodos para a geração das regras, sejam elas geradas antes ou depois da aplicação do algoritmo genético.

Em ambas as abordagens, deve ser considerada a questão do significado semântico dos termos lingüísticos. Durante o processo de aprendizado genético, caso não haja nenhum tipo de restrição, o formato dos conjuntos pode ser transformado de maneira a comprometer o significado dos termos lingüísticos. Para contornar esse problema, este trabalho considera a aplicação dos algoritmos genéticos com restrições para evitar situações onde, por exemplo, o ponto de pico de um conjunto fique em cima do ponto de pico de um outro conjunto que se encontra à direita dele na partição.

A seguir serão detalhadas as abordagens mencionadas, e posteriormente, os elementos do algoritmo genético utilizado por ambas abordagens serão apresentados. Também serão descritos o formato das regras e o raciocínio nebuloso empregado.

4.3.1 Otimização Genética das Funções de Pertinência

Nesta seção será descrito o processo de otimização genética das funções de pertinência *a posteriori*, ou seja, a base de regras é previamente definida, para que depois as funções de pertinência sejam ajustadas. Este processo consiste em dois passos: (1) Geração das regras por um método dirigido por dados para a construção rápida da base de regras, enfocando a simplicidade e transparência das regras e (2) otimização genética do desempenho do sistema por meio do ajuste das funções de pertinência.

A construção da BR é realizada a partir de uma BD inicial predefinida, através de um método rápido e de fácil implementação. Geralmente a definição da BD inicial é feita escolhendo um número de rótulos para cada variável lingüística (um número ímpar entre três e nove) e distribuindo uniformemente os valores lingüísticos em seus respectivos universos de discurso. O número de rótulos para cada variável é definido empiricamente no início do algoritmo pelo usuário e permanece fixo até o fim. Funções de pertinência triangulares são consideradas devido a sua simplicidade de manipulação e implementação. Neste trabalho, a construção das regras está sendo feita pelo método de Wang e Mendell [Wang e Mendel 1992] (para mais detalhes ver Apêndice A).

Após a construção das regras, o segundo passo a ser realizado é encontrar uma BD que melhor se adapta a esta BR, através de um processo de aprendizagem genético. Isto é feito pelo ajuste dos parâmetros das funções de pertinência, que definem os conjuntos nebulosos associados a cada variável lingüística do conjunto de dados em questão. Uma restrição imposta à aplicação dos operadores genéticos é com relação à manutenção da interpretabilidade das variáveis lingüísticas. Em outras palavras, os operadores genéticos devem respeitar o universo de discurso das variáveis, para que valores absurdos, isto é, fora do domínio, não sejam atribuídos a elas.

Cada cromossomo representa a BD completa do sistema através da codificação dos parâmetros das funções de pertinência. Sendo assim, um cromossomo é avaliado medindo o desempenho da BC, obtida a partir da BD representada por ele mais a BR construída na primeira etapa do processo. Com a obtenção da BC completa (BD + BR), um valor é medido para avaliar o desempenho do sistema utilizando um conjunto de teste. O processo de ajuste é executado até que a condição de parada do algoritmo seja satisfeita. Portanto, ao final do mesmo, a última BD codificada será a resposta encontrada. Na Figura 4.5 é mostrado o

fluxograma do algoritmo genético para o ajuste da BD, onde Max_Gen é uma constante que define o número máximo de gerações e $P(t)$ é a população corrente.

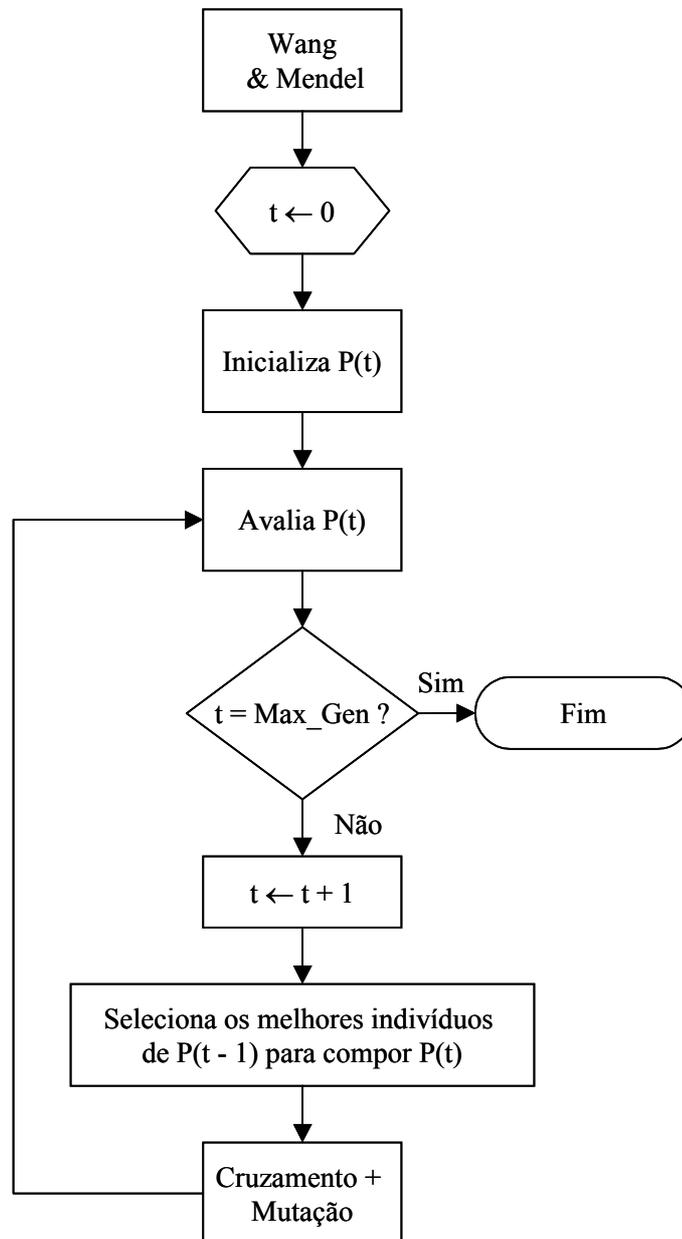


Figura 4.5: Fluxograma do AG que realiza o ajuste da BD.

4.3.2 Construção Genética das Funções de Pertinência

Nesta seção será descrito o processo de construção genética das funções de pertinência *a priori*, ou seja, a base de regras é definida após a geração das funções de pertinência. Para o tratamento desta abordagem, será seguida a estratégia composta de dois métodos com diferentes finalidades: (1) Processo de aprendizado genético da base de dados que permite

definir a forma dos conjuntos nebulosos. O número de termos lingüísticos de cada partição nebulosa é definido no início do algoritmo pelo usuário e permanece fixo até o fim; (2) método dirigido por dados para a construção rápida da base de regras.

Nesta abordagem é utilizado um processo de aprendizado genético para definir a BD e um método rápido para a construção da BR – o método de Wang e Mendel [Wang e Mendel 1992] – os quais trabalham de uma forma cooperativa. Toda vez que uma BD é obtida pelo processo genético, o método de Wang e Mendel é executado para cada indivíduo da população, obtendo, portanto, a BC completa do sistema.

Da mesma forma que a abordagem anterior, cada cromossomo representa a BD completa do sistema através da codificação dos parâmetros das funções de pertinência e o número de rótulos para cada variável lingüística é um número ímpar entre três e nove, definido empiricamente pelo usuário no início do programa, que permanece fixo até o fim do algoritmo, ao contrário do trabalho de [Cordón et al. 2001c], que permite a alteração deste número durante o processo genético. As funções de pertinência também são triangulares.

Para avaliar um cromossomo, a BR é construída a partir da BD contida neste cromossomo, obtendo a BC completa. Portanto, um valor de desempenho é medido a partir de um conjunto de teste. A função que avalia o desempenho do sistema é igual para ambas abordagens e será detalhada posteriormente.

O processo de aprendizado da BD será executado até que a condição do algoritmo seja satisfeita e ao final do mesmo, a última BD definida é a melhor BD encontrada, segundo o algoritmo genético aplicado. Na Figura 4.6 é mostrado o fluxograma do algoritmo genético para a construção genética da BD, onde *Max_Gen* é uma constante que define o número máximo de gerações e *P(t)* é a população corrente.

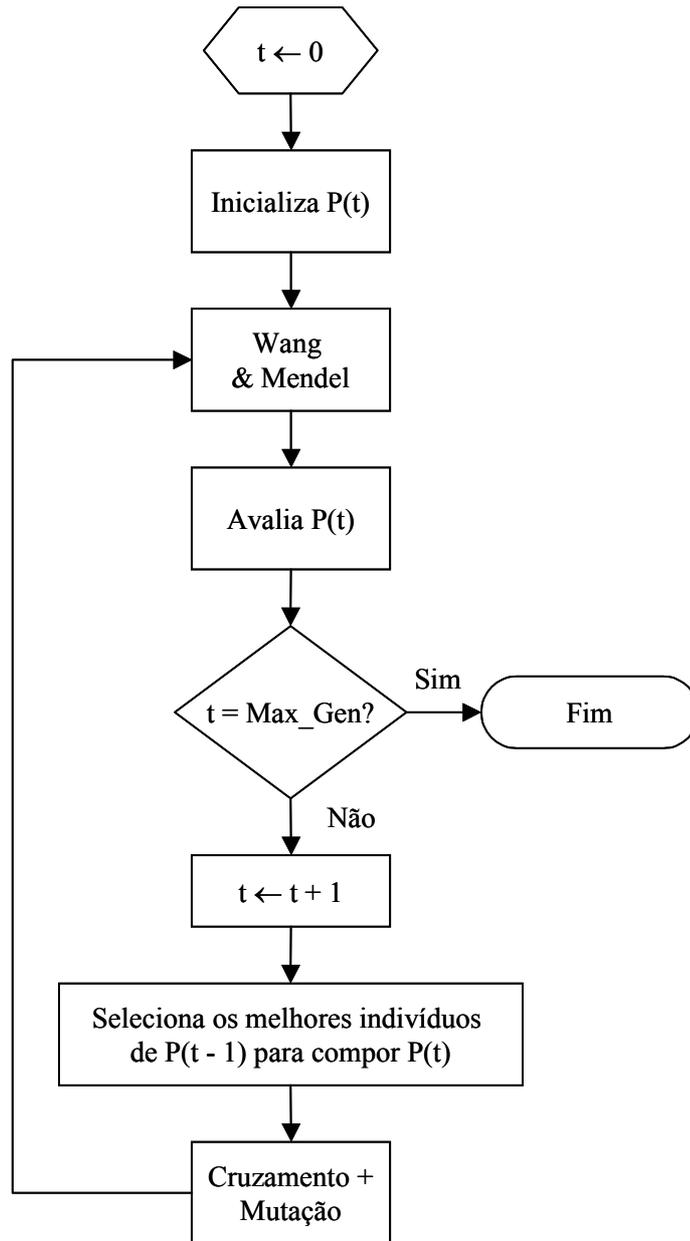


Figura 4.6: Fluxograma do AG que realiza a construção da BD.

A seguir serão detalhados os principais componentes do algoritmo genético utilizado pelas duas abordagens consideradas neste trabalho – o ajuste e a construção genética da BD –, tais como, codificação do cromossomo, formação da população inicial, operadores genéticos, função de avaliação e os parâmetros utilizados pelo algoritmo genético. Também serão descritos o formato das regras e o raciocínio nebuloso empregado.

4.3.3 Representação

Cada cromossomo representa a definição completa da BD pela codificação dos parâmetros das funções de pertinência que definem a semântica das variáveis linguísticas. Como estão sendo consideradas funções triangulares (ver seção 3.2.1), um cromossomo de números reais de $N \times 9 \times 3$ posições é usado para armazenar as funções de pertinência (N variáveis, com nove sendo o número máximo de conjuntos por variável e cada conjunto definido por três valores reais).

Se m_i é a granularidade da variável $i \in [0..N-1]$, $P_{ij}^1, P_{ij}^2, P_{ij}^3$ são os pontos de definição do conjunto j da variável i , portanto, a informação sobre a partição nebulosa da variável i é codificada no cromossomo, como uma seqüência de triplas números reais, onde cada tripla possui três números que descrevem o formato dos conjuntos nebulosos associados a variável i . Uma representação gráfica do cromossomo é mostrada na Figura 4.7 para N variáveis.

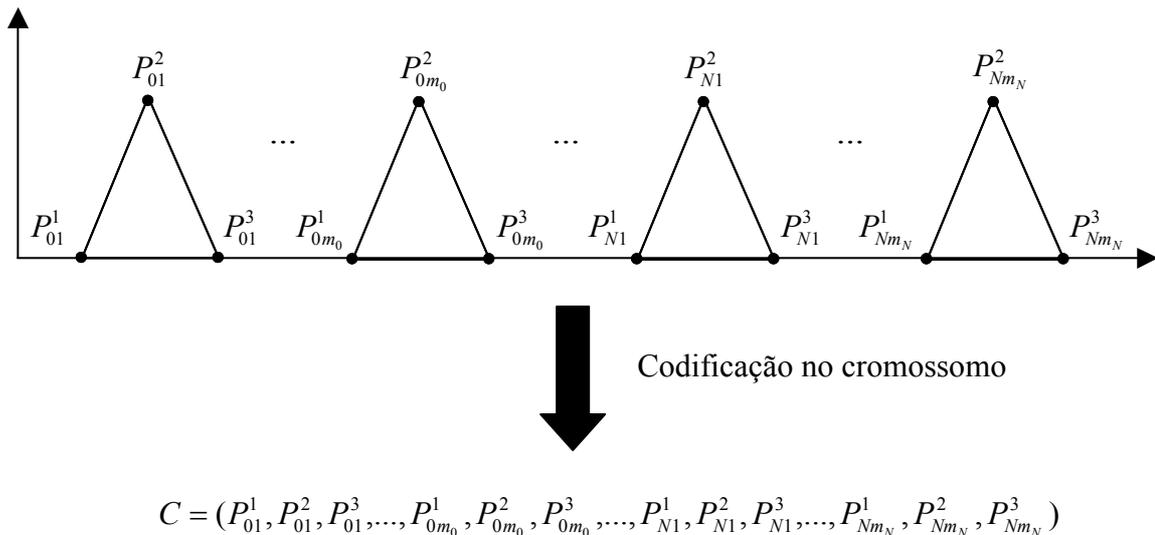


Figura 4.7: Representação cromossômica da BD.

Com o objetivo de evitar situações indesejáveis, tais como o pico de um conjunto sobrepor o pico do próximo conjunto, a variação do intervalo de cada ponto dos conjuntos é definida a partir da partição nebulosa uniforme de cada variável. Os intervalos de variação de cada ponto do conjunto j da variável i , $(P_{ij}^1, P_{ij}^2, P_{ij}^3)$, são definidos em (4.1), baseados em [Cordón et al. 2001c].

Os pontos A, B e C representam os parâmetros de uma função triangular, conforme é ilustrado na Figura 4.8.

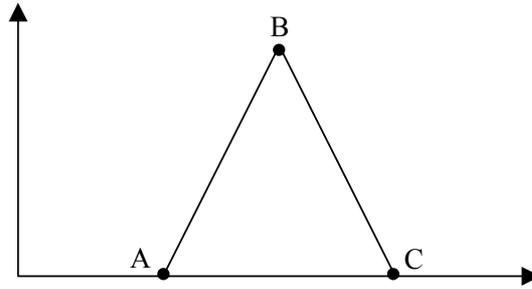


Figura 4.8: Parâmetros de uma função triangular.

$$\begin{aligned}
 P_{ij}^1 \in [A_L, A_R] &= \left[A - \frac{B-A}{2}, A + \frac{B-A}{2} \right] \\
 P_{ij}^2 \in [B_L, B_R] &= \left[B - \frac{B-A}{2}, B + \frac{C-B}{2} \right] \\
 P_{ij}^3 \in [C_L, C_R] &= \left[C - \frac{C-B}{2}, C + \frac{C-B}{2} \right]
 \end{aligned} \tag{4.1}$$

Na Figura 4.9 é ilustrado o intervalo de variação dos pontos das funções de pertinência, conforme definido pela equação (4.1).

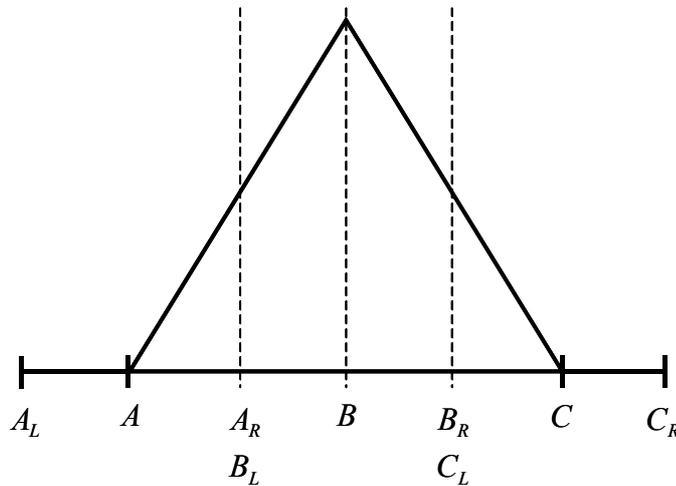


Figura 4.9: Intervalo de variação dos pontos das funções de pertinência.

4.3.4 População Inicial

Sejam $C(i)$, onde $i \in [0..T]$ e T é o número total de indivíduos da população, os cromossomos que constituem a população inicial do algoritmo genético em questão.

A formação do cromossomo $C(0)$, é definida a partir da partição nebulosa uniforme de cada variável do conjunto de dados. Por exemplo, seja uma variável V contínua de um

conjunto de dados J , definida em um domínio pertencente ao intervalo $[0,27]$, e que cinco conjuntos nebulosos foram definidos para esta variável. A codificação dos parâmetros das funções de pertinência associadas aos cinco conjuntos, possuem os valores especificados na Tabela 4.1.

Tabela 4.1: Valores dos parâmetros das funções de pertinência da variável V .

| Conjunto | Parâmetros | | |
|----------|------------|------|------|
| | A | B | C |
| 1 | 0 | 0 | 9 |
| 2 | 4,5 | 9 | 13,5 |
| 3 | 9 | 13,5 | 18 |
| 4 | 13,5 | 18 | 22,5 |
| 5 | 18 | 27 | 27 |

Na Figura 4.10, é mostrada graficamente a partição nebulosa uniforme da variável V , de acordo com valores especificados na Tabela 4.1.

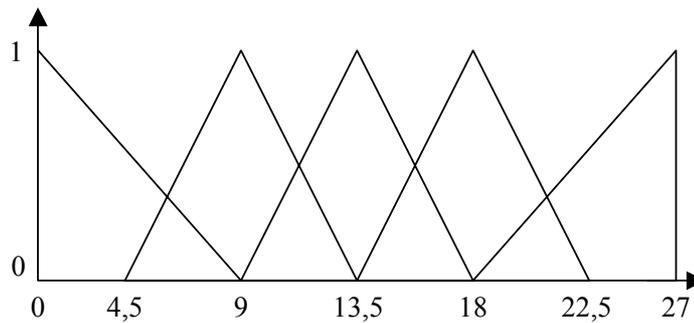


Figura 4.10: Partição nebulosa uniforme da variável V .

A codificação da parte do cromossomo $C(0)$ correspondente à variável V será, portanto, da seguinte forma:

$$(0; 0; 9; 4,5; 9; 13,5; 9; 13,5; 18; 13,5; 18; 22,5; 18; 27; 27).$$

O cromossomo $C(0)$ será formado de concatenações de partes como esta, uma para cada variável.

Após ter definido o cromossomo $C(0)$, os demais são formados a partir dele, ou seja, para cada ponto, é gerado um valor aleatório entre os intervalos definidos na equação (4.1). A Figura 4.11 ilustra um exemplo de uma partição nebulosa da variável V , gerada a partir da sequência correspondente no cromossomo $C(0)$. Os conjuntos desenhados por linhas pontilhadas são os obtidos a partir do $C(0)$.

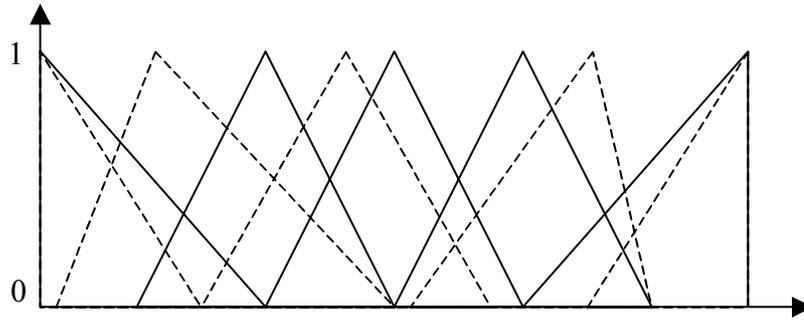


Figura 4.11: Exemplo de partição nebulosa gerada a partir do cromossomo $C(0)$.

Assim, a seqüência obtida possui a seguinte codificação:

$$(0; 0; 7,8; 0,3; 7,4; 13,5; 7,8; 12,6; 17; 13,8; 17,7; 22,5; 17,6; 27; 27).$$

Porém, deve-se notar que há uma condição especial para dois conjuntos pertencentes a esta partição. Como a variável V possui uma partição nebulosa uniforme conforme ilustrada na Figura 4.10, o primeiro e o último conjunto possuem pontos que não devem ser alterados. No caso do primeiro conjunto, os pontos A e B permanecem fixos. Em relação ao último conjunto da partição, os pontos B e C não são alterados. Isto se deve ao fato, de que tais pontos ao serem alterados, podem ultrapassar o limite inferior e superior do domínio da variável. Sendo assim, uma definição da variação dos intervalos de cada ponto para o primeiro e último conjunto da partição são definidos nas equações (4.2) e (4.3), respectivamente.

$$\begin{aligned} P_{ij}^1 \in [A_L, A_R] &= [A, A] \\ P_{ij}^2 \in [B_L, B_R] &= [B, B] \\ P_{ij}^3 \in [C_L, C_R] &= \left[C - \frac{C-B}{2}, C + \frac{C-B}{2} \right] \end{aligned} \quad (4.2)$$

$$\begin{aligned} P_{ij}^1 \in [A_L, A_R] &= \left[A - \frac{B-A}{2}, A + \frac{B-A}{2} \right] \\ P_{ij}^2 \in [B_L, B_R] &= [B, B] \\ P_{ij}^3 \in [C_L, C_R] &= [C, C] \end{aligned} \quad (4.3)$$

Estas definições são usadas para todas as variáveis contínuas do conjunto de dados em questão. Para as variáveis discretas, apenas os conjuntos unitários (*singleton*) são associados a elas. Da mesma forma, o ponto que define o conjunto unitário permanecerá fixo.

4.3.5 Operadores Genéticos

A seleção dos cromossomos (indivíduos) que sofrerão a ação dos operadores genéticos, é feita pelo método da Roleta (ver seção 2.4.2). Para garantir que a aplicação dos operadores genéticos não destrua indivíduos com boas aptidões, será usado em conjunto com o método da Roleta, a seleção Elitista.

Durante a fase de reprodução do algoritmo genético são empregados dois operadores genéticos clássicos, a mutação não uniforme proposta por Michalewicz [Michalewicz 1996] e o cruzamento aritmético (ver seção 2.4.3).

Conforme já comentado anteriormente, em ambas as abordagens, deve ser considerada a questão do significado semântico dos termos lingüísticos. Durante o processo de sintonia ou construção das funções de pertinência, caso não haja nenhum tipo de limitação, o formato dos conjuntos pode ser transformado de maneira a comprometer o significado dos termos lingüísticos. Para contornar esse problema, este trabalho considera a aplicação dos operadores genéticos com restrições, para evitar situações onde o ponto de pico de um conjunto fique em cima do ponto de pico de um outro conjunto que se encontra à direita dele na partição. Estas restrições foram definidas nas equações (4.1), (4.2) e (4.3), as quais estabelecem a variação dos intervalos dos pontos do conjuntos nebulosos associados às variáveis lingüísticas.

4.3.6 Formato das Regras Nebulosas e Método de Raciocínio Nebuloso

Classificação é uma importante tarefa encontrada em vários campos como o reconhecimento de padrões, tomada de decisão, mineração de dados e modelagem [Berg et al. 2002]. Um problema típico de classificação nebulosa pode ser descrito como segue: um universo de objetos ou casos $U = \{u\}$ é descrito por uma coleção de atributos $A = \{A_1, \dots, A_k\}$. Cada atributo A_k mede alguma característica importante de um objeto e é limitado a um conjunto de termos lingüísticos $T(A_k) = \{T_{1k}^k, \dots, T_{sk}^k\}$. $T(A_k)$ é, em outras palavras, o domínio do atributo A_k . Cada objeto u no universo U é classificado por um conjunto de classes $C = \{C_1, \dots, C_L\}$. Em um sistema de classificação todos os objetos podem ser classificados por um conjunto de regras $R = \{r_1, \dots, r_m\}$. Uma regra de classificação pode ser descrita da seguinte forma:

$$r: \text{SE } (A_1 \text{ é } T_{i1}^1) \text{ E...E } (A_k \text{ é } T_{ik}^k) \text{ ENTÃO } (C \text{ é } C_j), \quad (4.4)$$

onde A_1, \dots, A_k são atributos dos objetos, $T_{i1}^1, \dots, T_{ik}^k$ são conjuntos nebulosos e C_j é a classe - possivelmente nebulosa - a que o objeto pertence.

Em um sistema de classificação nebuloso, o método de raciocínio é baseado na lógica nebulosa. Ele deriva conclusões de um conjunto de regras nebulosas e um padrão. Esta seção apresenta o método de raciocínio nebuloso usado nos experimentos, chamado **método de raciocínio nebuloso baseado na regra vencedora**.

Seja $e_p = \{a_{p1}, a_{p2}, \dots, a_{pn}\}$ um objeto a ser classificado, $\{a_{p1}, \dots, a_{pn}\}$ os atributos do objeto e $R = \{R_1, R_2, \dots, R_S\}$ o conjunto de regras nebulosas. O método de raciocínio baseado na regra vencedora segue os seguintes passos, adaptado de [Cordón et al. 2001d]:

- **Passo 1:** Calcular o grau de compatibilidade, $\text{Compat}(R_k, e_p)$, entre o objeto e_p e cada regra R_k , $k=1 \dots S$, aplicando uma T-norma [Klir e Yuan 1995] sobre o grau de pertinência dos valores do objeto nos correspondentes conjuntos nebulosos da regra.

$$\text{Compat}(R_k, e_p) = T(\mu_{A_{i1}}(a_{p1}), \dots, \mu_{A_{in}}(a_{pn})) \quad (4.5)$$

- **Passo 2:** Encontrar a regra com maior grau de compatibilidade com o objeto.

$$\text{Max}\{\text{Compat}(R_k, e_p)\}, k=1 \dots S \quad (4.6)$$

- **Passo 3:** O objeto e_p será classificado na classe C_j , tal que C_j é a classe da regra R_k que possui o maior grau de compatibilidade com o objeto.

Se duas ou mais regras apresentarem o mesmo grau de compatibilidade com o padrão, mas possuírem classes diferentes, a regra que aparecer primeiro será escolhida. Se a classe é nebulosa, o objeto é considerado classificado corretamente se a classe a qual foi designada a ele é a classe tal que o objeto tem maior grau de pertinência.

4.3.7 Função de Avaliação

Da mesma forma que os demais componentes do algoritmo genético, a função de avaliação utilizada é a mesma tanto para a otimização genética da BD, quanto para a construção genética da BD.

A função de avaliação $E(C_i)$ é baseada no desempenho da base de regras que foi gerada a partir das informações contidas nos cromossomos, calculado pelo número de padrões classificados corretamente, usando o método de raciocínio nebuloso apresentado anteriormente. A função de avaliação é expressa por:

$$E(C_i) = NPC(C_i) \quad (4.7)$$

onde $NPC(C_i)$ é o número de padrões classificados corretamente pela base de regras gerada pelo cromossomo C_i .

No entanto, um importante fato em relação à aplicação da função nas abordagens deve ser ressaltado. Na construção genética da BD, necessariamente a função de avaliação é aplicada após a execução do algoritmo de Wang e Mendel, pois é quando uma nova BR é construída, conforme está esquematizado no fluxograma do algoritmo (ver Figura 4.6). Já na abordagem que consiste em ajustar a BD, isto não acontece. Pelo simples fato de que é gerada uma única BR no início do algoritmo e que permanece fixa até o fim do processo (ver Figura 4.5).

4.3.8 Outros Parâmetros

Os parâmetros do algoritmo genético, tais como, número máximo de gerações, tamanho da população, as probabilidades de mutação e cruzamento e porcentagem do elitismo foram definidos empiricamente e estão relacionados na Tabela 4.2.

Tabela 4.2: Parâmetros do AG.

| Parâmetro | Valor |
|-----------------------------|--------------|
| Número máximo de gerações | 1000 |
| Tamanho da população | 150 |
| Probabilidade da mutação | 0,01 |
| Probabilidade de cruzamento | 0,7 |
| Porcentagem do Elitismo | 10% |

4.4 Considerações Finais

Neste capítulo foi apresentada a proposta de trabalho desenvolvida, a qual teve por principais objetivos a aplicação de algoritmos genéticos no ajuste da BD *a posteriori* e na construção da BD *a priori*. Os principais componentes do algoritmo genético utilizado por ambas abordagens foram detalhados, assim como a forma de raciocínio nebuloso empregado.

No próximo capítulo serão apresentados os experimentos e resultados obtidos, aplicados à problemas de classificação de padrões.

Capítulo 5

Experimentos e Resultados

5.1 Considerações Iniciais

O sucesso dos SBRN na resolução de problemas dos mais diversos tipos vem sendo comprovado através da crescente pesquisa que se encontrada na literatura. Em paralelo, o uso de algoritmos genéticos no projeto automático da base de conhecimento dos SBRN torna-se cada vez maior, devido ao aumento no desempenho dos SBRN em comparação com outros métodos de aprendizado de máquina [Yuan e Zhuang 1996] [Ishibuchi et al. 1999b] [Ishibuchi et al. 1999a]. Porém, uma observação que deve ser feita, é o fato de que a maioria destes trabalhos aplica os SBRN em domínios de problemas de controle e aproximação de funções, enquanto que o domínio de problemas de classificação de padrões é explorado com menos frequência, pelo fato de que tais problemas apresentam uma dimensionalidade maior em relação aos problemas de controle, e conseqüentemente, a construção de classificadores nebulosos é mais complexa do que a construção de controladores nebulosos.

Sendo assim, os experimentos realizados neste trabalho focalizam os problemas de classificação de padrões, os quais são representados por conjuntos de dados numéricos que descrevem para cada padrão, seus atributos e classes. Os conjuntos de dados usados para realizar os experimentos foram *Auto-Mpg*, *Boston Housing*, *Wine* e *Iris Plants*, disponíveis em [Blake e Merz 1998].

Neste capítulo, na seção 5.2 são detalhadas as características dos conjuntos de dados utilizados nos experimentos. Na seção 5.3 são discutidos os objetivos dos experimentos. Nas seções 5.4, 5.5, 5.6 e 5.7 são apresentados os resultados obtidos para os conjuntos *Auto-Mpg*, *Iris Plants*, *Wine* e *Boston Housing*, respectivamente.

5.2 Detalhamento dos Conjuntos de Dados

Nesta seção serão detalhados os conjuntos de dados utilizados nos experimentos, descrevendo as características de cada atributo, a quantidade de amostras e o tipo de problema a que se refere cada conjunto.

O conjunto de dados Auto-Mpg possui 398 amostras e se refere ao consumo de combustível de carros, na cidade, em milhas por galão, sendo expresso por três atributos discretos multi-valores e cinco contínuos (incluindo o atributo de saída) conforme definido originalmente. Porém, seis amostras foram perdidas do atributo “cavalos de potência”, sendo assim, apenas 392 amostras são consideradas. Na Tabela 5.1, está a relação dos atributos:

Tabela 5.1: Informações dos atributos do conjunto de dados Auto-Mpg.

| Atributo | Característica |
|------------------------|------------------------|
| Cilindros | Discreto multi-valores |
| Deslocamento | Contínuo |
| Cavalos de potência | Contínuo |
| Peso | Contínuo |
| Aceleração | Contínuo |
| Ano modelo | Discreto multi-valores |
| Origem | Discreto multi-valores |
| Mpg (milhas por galão) | Contínuo |

Alguns atributos relacionados na Tabela 5.1 estão sendo tratados de uma forma diferente neste trabalho. Os atributos “cilindros” e “ano modelo” são considerados contínuos e não discretos, por apresentarem um intervalo de valores, onde é possível ser realizado este tipo de tratamento.

O conjunto de dados Boston Housing descreve o valor de casas do subúrbio de Boston, de acordo com as características representadas pelos atributos da Tabela 5.2, com um total de 506 exemplos.

Tabela 5.2: Informações dos atributos do conjunto de dados Boston Housing.

| Atributo | Característica |
|--|----------------|
| CRIM (taxa de criminalidade) | Contínuo |
| ZN (proporção de zonas residenciais divididas em lotes acima de 25.000 sq.ft.) | Contínuo |
| INDUS (Proporção de acres de estabelecimentos não varejistas por cidade) | Contínuo |
| CHAS (Variável fictícia do Rio Charles, =1 se o terreno está na margem do rio, 0 caso contrário) | Discreto |
| NOX (concentração de óxido nítrico) | Contínuo |
| RM (número médio de quartos por residência) | Contínuo |

| | |
|--|----------|
| AGE (proporção de unidades construídas antes de 1940) | Contínuo |
| DIS (distância a cinco centros de emprego de Boston) | Contínuo |
| RAD (acessibilidade a rodovias) | Contínuo |
| TAX (Proporção da taxa de propriedade completa por \$10,000) | Contínuo |
| PTRATIO (relação aluno-professor) | Contínuo |
| B (população de negros) | Contínuo |
| LSTAT (porcentagem de população pobre) | Contínuo |
| MEDV (valor médio de casas) | Contínuo |

Os atributos descritos na Tabela 5.3 pertencem ao conjunto de dados Wine, que possui 178 amostras divididas em três classes de vinhos diferentes, das quais 59 pertencem a classe 1, 71 da classe 2 e 48 amostras pertencem a classe 3. Estes dados são os resultados de uma análise química de vinhos de uma mesma região da Itália, mas os quais derivaram de três cultivadores diferentes. Esta análise determinou 13 componentes em cada um dos três tipos de vinhos.

Tabela 5.3: Informações dos atributos do conjunto de dados Wine.

| Atributo | Característica |
|--------------------------------|-----------------------|
| Álcool | Contínuo |
| Ácido Málico | Contínuo |
| Ash | Contínuo |
| Alcalinidade de ash | Contínuo |
| Magnésio | Contínuo |
| Total de Fenóis | Contínuo |
| Flavanóides | Contínuo |
| Fenóis não-flavonóides | Contínuo |
| Proanthocyanins | Contínuo |
| Intensidade de cor | Contínuo |
| Cor | Contínuo |
| OD280/OD315 de vinhos diluídos | Contínuo |
| Proline | Contínuo |
| Classe do vinho | Discreto |

Finalmente, o conjunto de dados Iris Plants é um dos mais conhecidos conjuntos de dados da literatura que trata de problemas de classificação de padrões. Ele contém três classes de 50 amostras cada, onde cada classe se refere a um tipo de planta íris (Íris Setosa, Íris Virginica e Íris Versicolor). A Tabela 5.4 descreve as informações dos atributos deste conjunto.

Tabela 5.4: Informações dos atributos do conjunto de dados Íris.

| Atributos | Característica |
|-----------------------|----------------|
| Comprimento da sépala | Contínuo |
| Largura da sépala | Contínuo |
| Comprimento da pétala | Contínuo |
| Largura da pétala | Contínuo |
| Classe da planta | Discreto |

5.3 Objetivos dos Experimentos

Um dos principais objetivos na construção da base de conhecimento de SBRN aplicados na resolução de problemas de classificação de padrões multidimensionais é a obtenção de uma base de regras compacta e com um alto poder de classificação [Ishibuchi et al. 1995] [Ishibuchi et al. 1997] [Kbir et al. 2000] [Castro et al. 2003]. Portanto, um dos objetivos dos experimentos foi constatar o impacto do tamanho da BR, em termos de números de variáveis de entrada, na taxa de classificação dos exemplos. Para isto, testes foram realizados alterando o número de variáveis para todos os conjuntos de dados, sendo que a redução do número de variáveis foi feita de forma empírica.

Os experimentos realizados neste trabalho tiveram também a finalidade de comparar os resultados obtidos por ambas as abordagens apresentadas nas seções 4.3.1 e 4.3.2, com o objetivo de determinar qual delas apresenta melhor desempenho em relação aos problemas de classificação relacionados na seção 5.1. Influenciado pelo trabalho de [Cordón et al. 2000], o qual analisa a importância da granularidade da partição nebulosa das variáveis lingüísticas no desempenho de SBRN, testes variando o número de conjuntos por variável lingüística foram considerados, sendo que este número assumiu valores ímpares entre três e nove.

Os dados que serão mostrados nas tabelas nas próximas seções são: ECC (exemplos classificados corretamente), Regras (número de regras geradas), Desvio padrão (desvio padrão calculado sobre o ECC) e Granularidade (número de conjuntos nebulosos por variável lingüística).

O algoritmo genético utilizado pelas duas abordagens foi executado independentemente cinco vezes para cada tipo de teste, sendo assim, os resultados de ECC e Regras expressam a média aritmética destes cinco resultados obtidos das execuções do algoritmo. Por outro lado, o Desvio padrão expressa o desvio padrão sobre os cinco resultados obtidos para ECC. As tabelas e gráficos serão somente aqueles em que foi considerado

granularidade igual a três, pois estes apresentaram resultados melhores quando comparados com as outras granularidades (cinco, sete e nove). Contudo, os resultados das demais granularidades são apresentados nos Apêndices B, C, D e E, para os conjuntos Auto-Mpg, Iris Plants, Wine e Boston Housing, respectivamente.

5.4 Resultados para o Conjunto Auto-Mpg

Nesta seção serão apresentados os resultados obtidos pelo algoritmo genético quando aplicado ao problema Auto-Mpg, através das abordagens de construção e ajuste da BD. O conjunto de dados Auto-Mpg, o qual possui 392 amostras, foi dividido aleatoriamente em dois conjuntos: 60% dos dados compõem o conjunto de treinamento (235 amostras) e os 40% restantes (157 amostras) compõem o conjunto de teste.

Na Tabela 5.5 e na Tabela 5.6, estão os resultados obtidos para a construção e otimização genética da BD, respectivamente.

Tabela 5.5: Resultados da construção genética da BD (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 140,8 | 145,4 | 149,2 | 156,6 | 156,8 |
| Regras | 45,8 | 39 | 38 | 20,4 | 13,8 |
| Desvio padrão | 1,4832 | 1,9493 | 0,4472 | 0,5477 | 0,4472 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

Tabela 5.6: Resultados da otimização genética da BD (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 137,4 | 144,4 | 145 | 152 | 152,4 |
| Regras | 39,6 | 36,2 | 33 | 20 | 12 |
| Desvio padrão | 5,4129 | 3,2863 | 3,3911 | 2,1213 | 1,8165 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

Na Figura 5.1 é ilustrada a comparação do desempenho do algoritmo genético nas duas abordagens considerando oito variáveis com granularidade três. Este desempenho é com relação ao ECC ao longo das gerações, ou seja, a busca por uma base de conhecimento com uma boa capacidade de classificação.

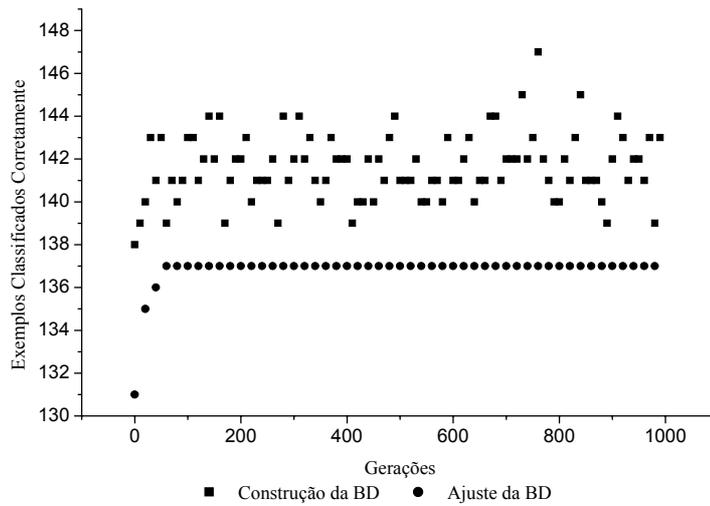


Figura 5.1: Comparação entre as abordagens do desempenho do AG com 8 variáveis e granularidade 3 (Auto-Mpg).

A Figura 5.2 mostra a evolução do melhor e do pior indivíduo do AG e também do valor médio dos indivíduos ao longo das gerações, na construção genética da BD, com oito variáveis e granularidade três. Conforme descrito na seção 4.3.3, um indivíduo representa uma BC completa do SBRN. Assim, a Figura 5.2 também retrata a busca por uma BC com alto poder de classificação. A Figura 5.3 mostra dados semelhantes para a abordagem de otimização genética da BD.

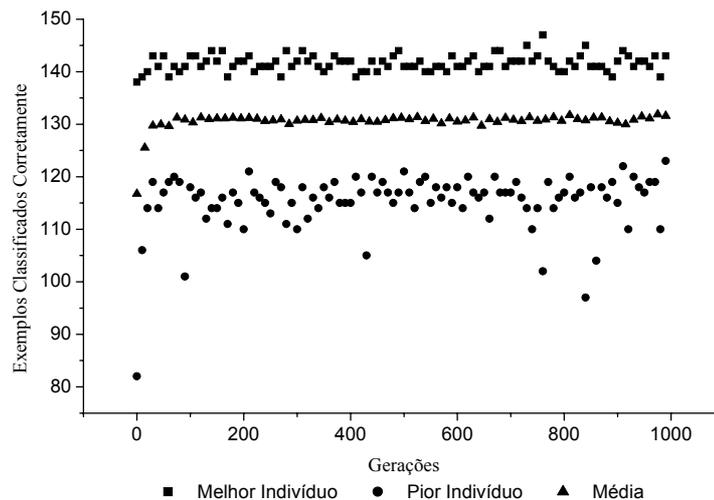


Figura 5.2: Desempenho dos indivíduos na construção da BD (Auto-Mpg).

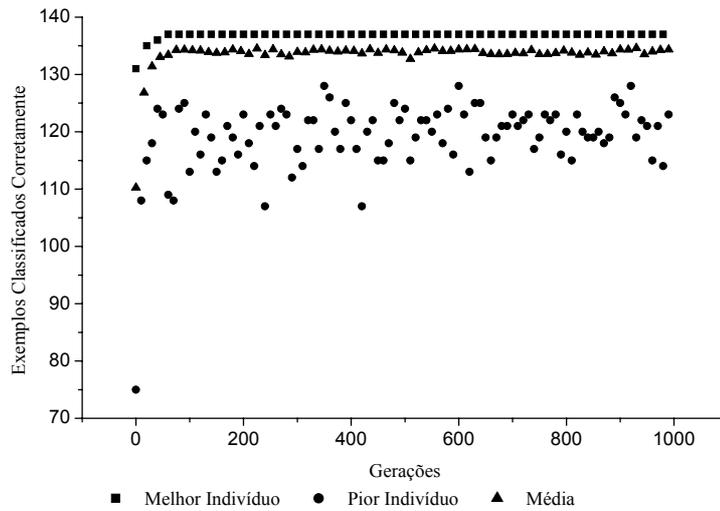


Figura 5.3: Desempenho dos indivíduos na otimização da BD (Auto-Mpg).

Analisando a Figura 5.2 e a Figura 5.3, pode-se verificar que a linha do gráfico que representa o melhor indivíduo do AG apresenta maior oscilação, quando a abordagem de construção da BD é considerada. Isto porque, a cada geração do AG, uma nova BR é construída e avaliada. Por outro lado, na abordagem da otimização da BD, a BR é única para todo o processo genético.

Na Figura 5.4 e na Figura 5.5, é ilustrado a relação da taxa de classificação com a quantidade de regras, para construção e otimização da BD, respectivamente. Esta relação tem o objetivo de mostrar se a taxa de classificação aumenta ou diminui de acordo com a quantidade de regras, sendo que o número de regras pode variar em função do número de variáveis.

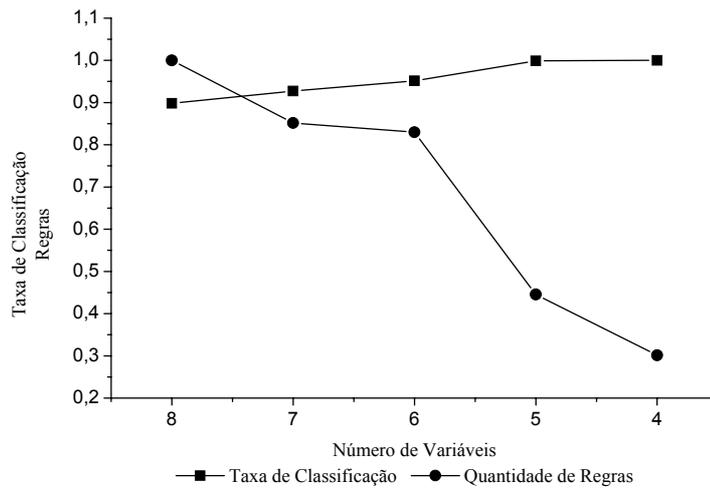


Figura 5.4: Relação taxa de classificação vs. quantidade de regras na construção da BD (Auto-Mpg).

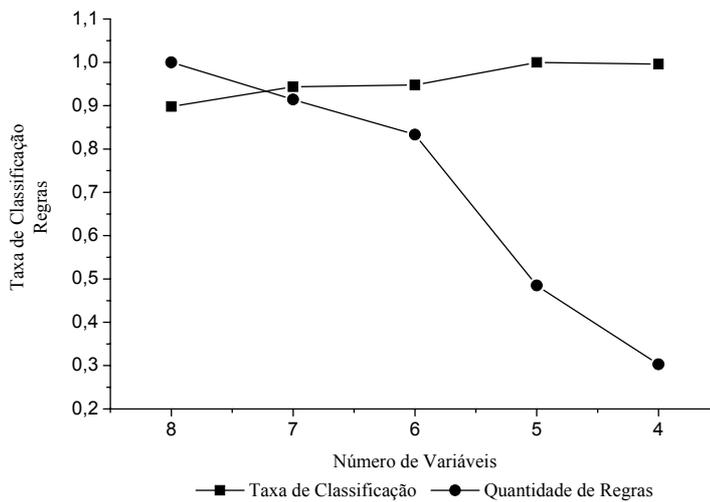


Figura 5.5: Relação taxa de classificação vs. quantidade de regras na otimização da BD (Auto-Mpg).

A partir da exposição dos gráficos, pode-se constatar que a abordagem que constrói a BD apresentou um desempenho melhor que a abordagem que ajusta a BD, porém, esta apresentou uma desvantagem no que se refere ao tamanho da BR, ou seja, gerou um número de regras um pouco maior do que a outra abordagem. Outro fato já esperado é com relação à diminuição no número de regras quando menos variáveis do conjunto de dados são consideradas.

5.5 Resultados para o Conjunto Iris Plants

Nesta seção serão apresentados os resultados obtidos para o problema Iris Plants. Da mesma forma como no Auto-Mpg, o conjunto Iris Plants foi dividido aleatoriamente em dois conjuntos: conjunto de treinamento (60% que equivale a 90 amostras) e conjunto de testes (40% que é igual a 60 amostras). Nos experimentos realizados com o conjunto Iris Plants não houve variação no número de variáveis, por ser um conjunto com poucos atributos.

Os resultados obtidos para a construção e otimização genética da BD estão expressos nas tabelas 5.7 e 5.8, respectivamente.

Tabela 5.7: Resultados da construção genética da BD (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 59,8 |
| Nº de regras | 14,4 |
| Desvio padrão | 0,4472 |
| Granularidade | 3 |

Tabela 5.8: Resultados da otimização genética da BD (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 56,6 |
| Nº de regras | 13 |
| Desvio padrão | 1,8165 |
| Granularidade | 3 |

É ilustrado graficamente uma comparação do desempenho do algoritmo genético utilizando granularidade três nas abordagens de construção e sintonia da BD na Figura 5.6:

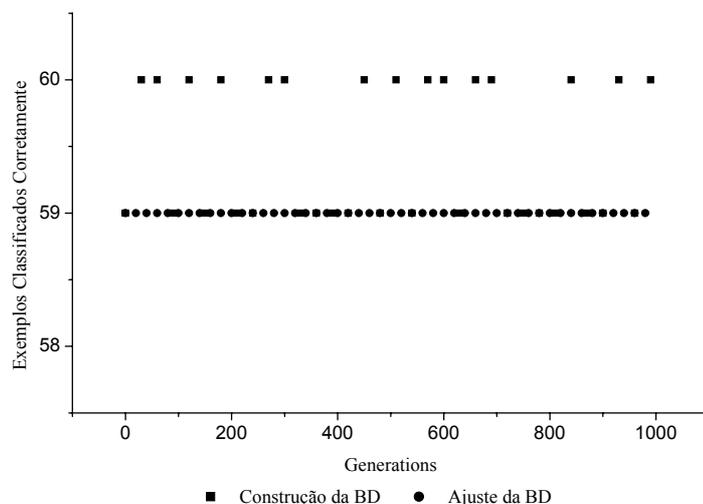


Figura 5.6: Comparação entre as abordagens do desempenho do AG com granularidade 3 (Iris Plants).

A evolução do melhor, do pior e da média dos indivíduos do algoritmo genético ao longo das gerações é ilustrada na Figura 5.7 para a construção genética da BD e na Figura 5.8 para a abordagem de otimização genética da BD.

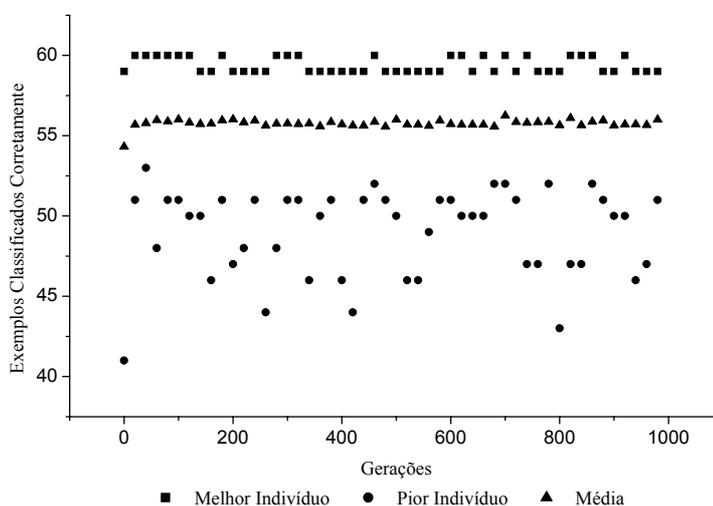


Figura 5.7: Desempenho dos indivíduos na construção da BD (Iris Plants).

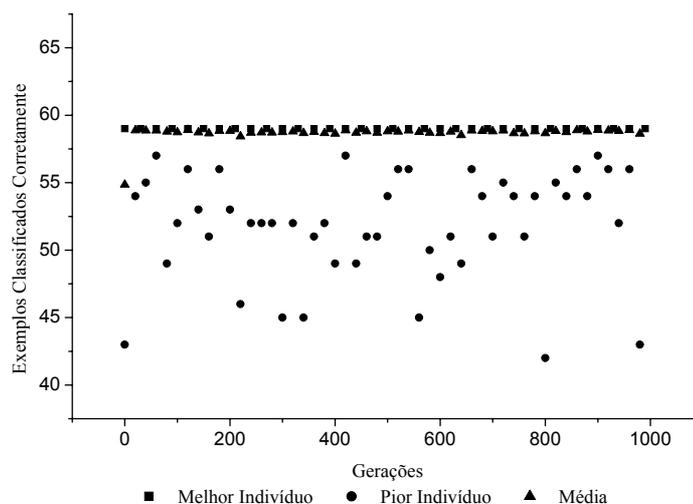


Figura 5.8: Desempenho dos indivíduos na otimização da BD (Iris Plants).

Os resultados apresentados mostraram que ambas as abordagens atingiram uma média de classificação e de regras geradas muito próximas, no entanto, a abordagem que utiliza o aprendizado *a priori* da BD teve melhor desempenho.

5.6 Resultados para o Conjunto Wine

Nesta seção serão apresentados os resultados obtidos para o conjunto de dados Wine. O conjunto de treinamento possui 106 amostras (60%) e o conjunto de testes possui 72 amostras (40%). Na Tabela 5.9 são descritos os resultados obtidos para a construção genética da BD e na Tabela 5.10 para a sintonia da BD.

Tabela 5.9: Resultados da construção genética da BD (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|------|--------|--------|--------|------|
| ECC | 67 | 66,8 | 68 | 69,4 | 70 |
| Regras | 41,6 | 42,2 | 40 | 40,6 | 39,8 |
| Desvio padrão | 1 | 1,7888 | 0,7071 | 0,5477 | 1 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

Tabela 5.10: Resultados da otimização genética da BD (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|------|--------|
| ECC | 68 | 66,8 | 68,6 | 69 | 67,4 |
| Regras | 37,2 | 37,4 | 38,2 | 38,6 | 37 |
| Desvio padrão | 0,7071 | 1,7888 | 0,5477 | 1 | 0,5477 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

A comparação entre as abordagens é ilustrada na Figura 5.9:

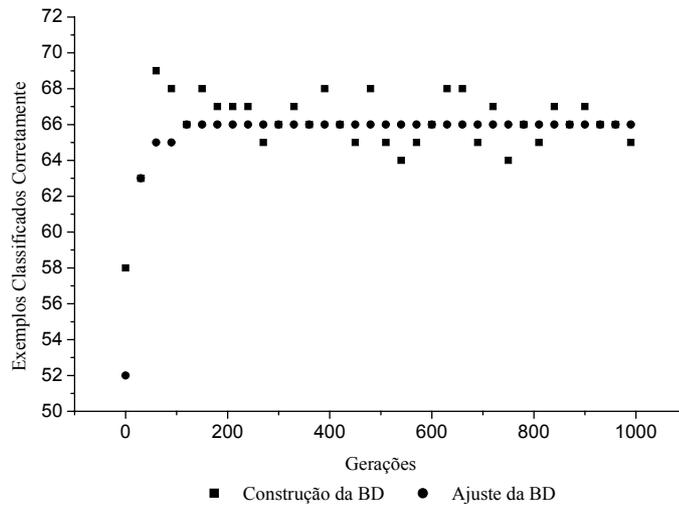


Figura 5.9: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 3 (Wine).

Nas figuras 5.10 e 5.11 são mostrados os comparativos evolutivos dos indivíduos (pior, melhor e valor médio) do algoritmo genético aplicado nas duas abordagens.

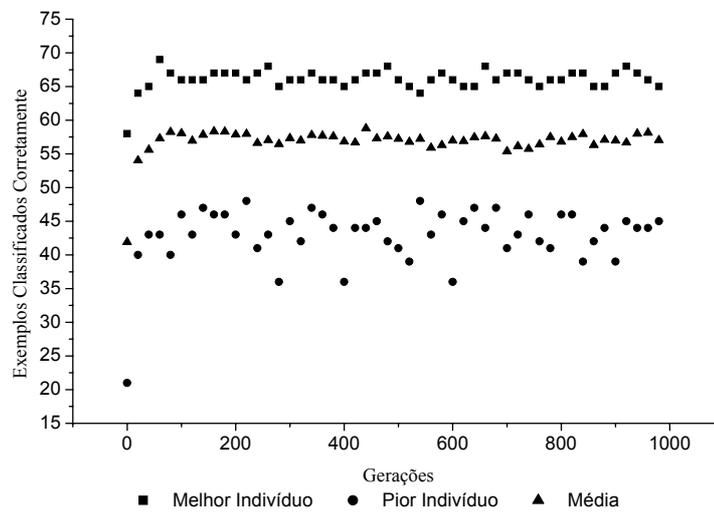


Figura 5.10: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 3 (Wine).

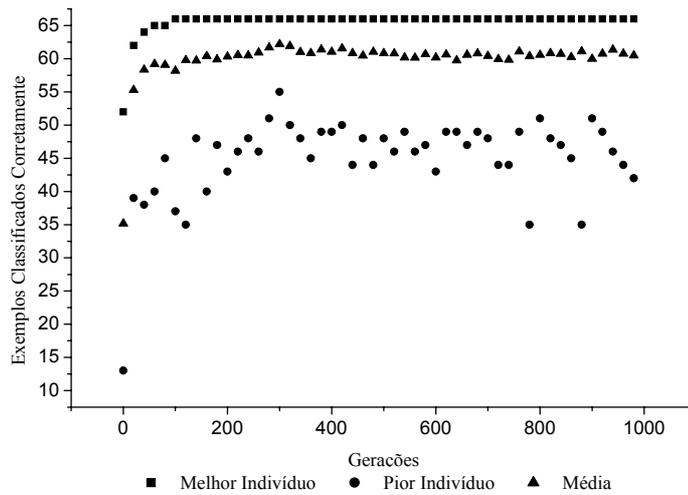


Figura 5.11: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 3 (Wine).

A variação no tamanho da base de regras é influenciada pela redução no número de variáveis consideradas nos testes, conseqüentemente, a taxa de classificação também é influenciada por esta variação. Este comportamento pode ser observado pelas figuras 5.12 e 5.13.

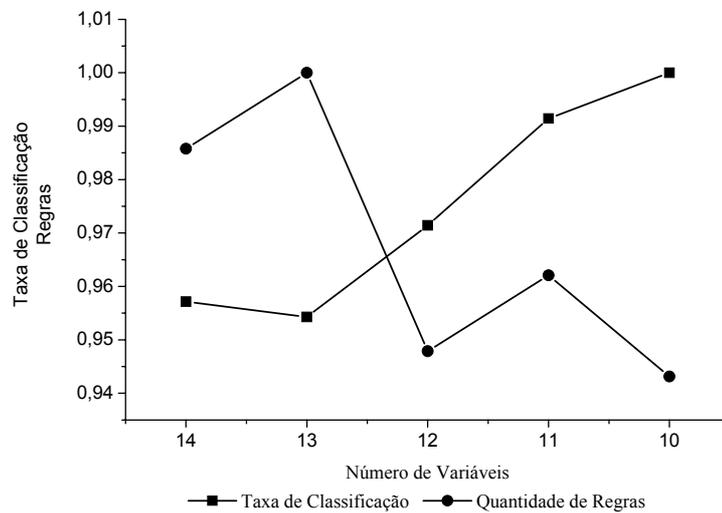


Figura 5.12: Relação taxa de classificação vs. quantidade de regras na construção da BD (Wine).

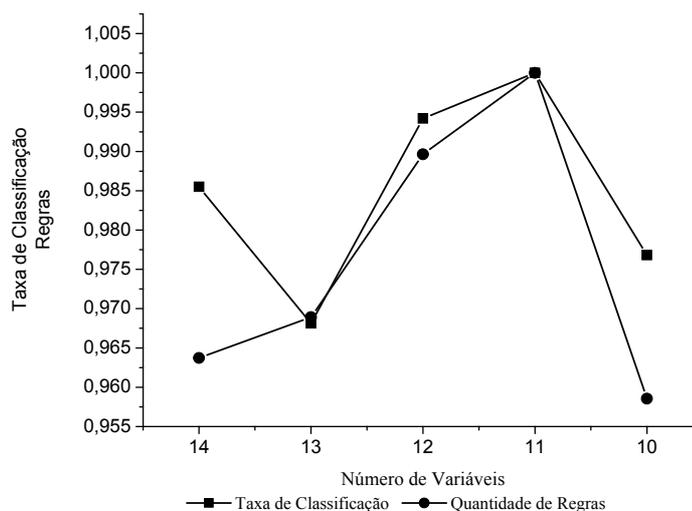


Figura 5.13: Relação taxa de classificação vs. quantidade de regras na otimização da BD (Wine).

Os resultados apresentados mostraram quase um empate entre as duas abordagens, em que alguns casos somente o número de regras geradas foram diferentes. Da mesma forma como ocorreu nos testes anteriores ao Wine, a abordagem de construção da BD é a que apresenta um maior número de regras.

5.7 Resultados para o Conjunto Boston Housing

Os resultados obtidos para o conjunto de dados Boston Housing serão apresentados nesta seção. Este conjunto possui um total de 506 amostras, sendo que 303 (60%) são utilizadas para o treinamento do algoritmo e 203 (40%) para teste.

Na Tabela 5.11 são descritos os resultados obtidos para a construção genética da BD.

Tabela 5.11: Resultados da construção genética da BD (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 50,6 | 185 | 187,8 | 194,8 | 196,6 |
| Regras | 27,8 | 73,4 | 65,2 | 69 | 54,8 |
| Desvio padrão | 2,0736 | 3,8729 | 0,8366 | 1,3038 | 1,3416 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

Na Tabela 5.12 são mostrados os resultados para a sintonia genética da BD.

Tabela 5.12: Resultados da otimização genética da BD (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 49 | 184,2 | 188 | 194,2 | 197,4 |
| Regras | 22,2 | 63 | 54,6 | 60,2 | 48,6 |
| Desvio padrão | 6,3639 | 2,3874 | 2,3452 | 2,3874 | 1,9493 |
| Granularidade | 3 | 3 | 3 | 3 | 3 |

A comparação entre as abordagens é ilustrada na Figura 5.14:

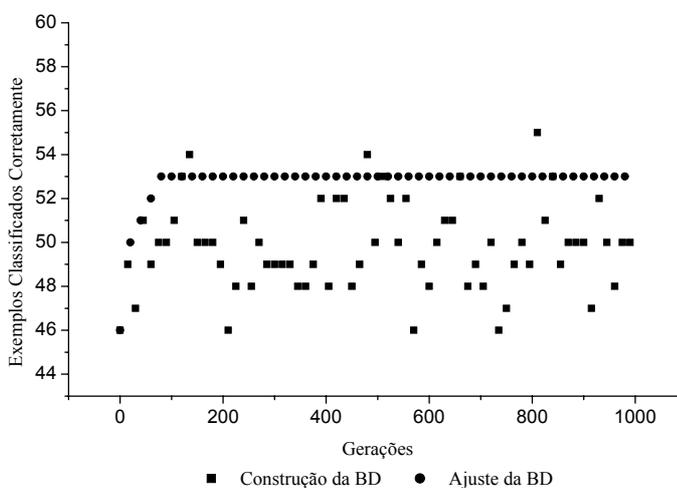


Figura 5.14: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 3 (Boston Housing).

A Figura 5.15 mostra a evolução do melhor e do pior indivíduo do AG, e também do valor médio dos indivíduos ao longo das gerações, na construção genética da BD com 14 variáveis e granularidade três. A Figura 5.16 mostra dados semelhantes para a abordagem de otimização genética da BD.

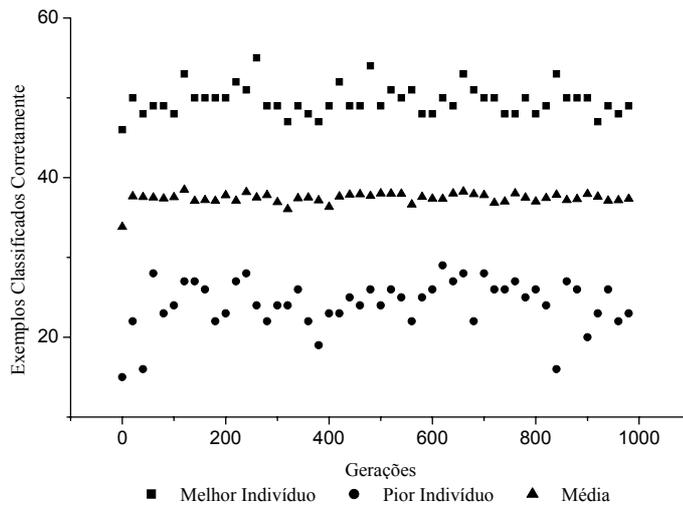


Figura 5.15: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 3 (Boston Housing).

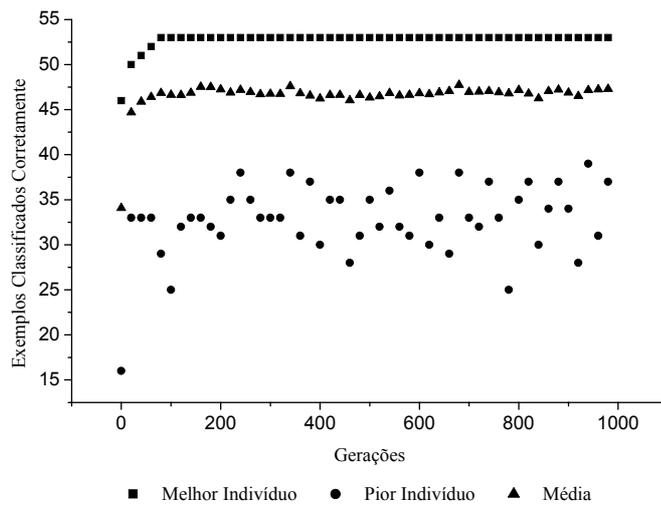


Figura 5.16: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 3 (Boston Housing).

Nas figuras 5.17 e 5.18 são ilustradas as comparações entre as taxas de classificação com o tamanho da base de regras do SBRN.

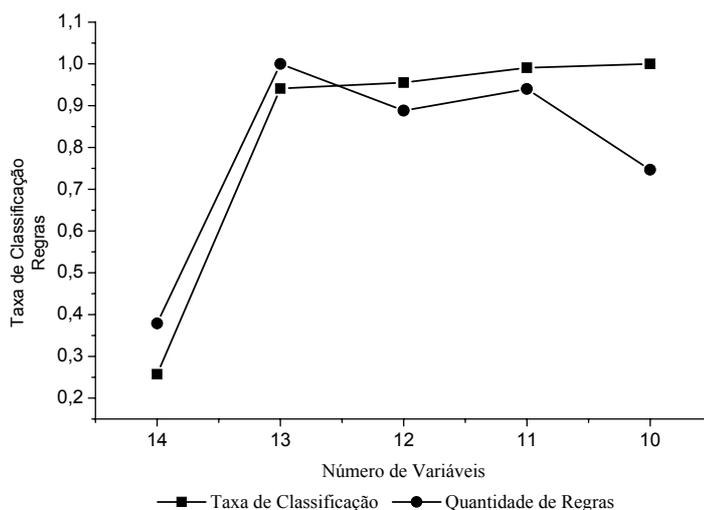


Figura 5.17: Relação taxa de classificação vs. quantidade de regras na construção da BD (Boston Housing).

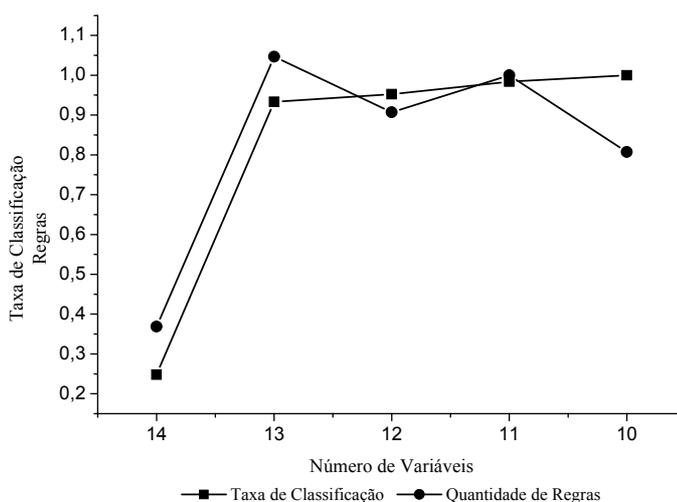


Figura 5.18: Relação taxa de classificação vs. quantidade de regras na otimização da BD (Boston Housing).

Novamente os resultados mostraram uma vantagem não muito expressiva da construção genética em relação ao ajuste da BD.

Um fato que deve ser ressaltado é o mau resultado da classificação apresentado, quando o número total de variáveis do conjunto é considerado nos experimentos. A partir de testes preliminares, pode-se constatar que o atributo ZN (ver Tabela 5.2) quando considerado nos experimentos, prejudica o desempenho do SBRN.

5.8 Considerações Finais

Neste capítulo foram apresentados os resultados obtidos pelas abordagens de construção e otimização genética da BD, aplicados aos problemas de classificação de padrões representados pelos conjuntos Auto-Mpg, Iris Plants, Wine e Boston Housing. No próximo capítulo serão apontadas as conclusões e trabalhos futuros.

Capítulo 6

Conclusões

Ao contrário da computação tradicional, a qual é baseada em modelos matemáticos precisos, a Computação Flexível é tolerante a imprecisão, incerteza e verdade parcial. Isso significa que, os paradigmas que formam a base da Computação Flexível, tais como, Lógica Nebulosa, Redes Neurais e Algoritmos Genéticos, podem ser uma alternativa para a resolução de problemas que atualmente não possuem solução por meio dos métodos tradicionais ou que as soluções encontradas por estes não apresentem bons resultados. Como exemplo, podemos citar a utilização da lógica nebulosa e de algoritmos genéticos em Sistemas Genéticos Nebulosos, a qual se tornou uma importante área de pesquisa durante a última década [Cordón et al. 2004].

Seguindo esta linha de pesquisa, este trabalho teve como objetivo a investigação da aplicação de algoritmos genéticos no aprendizado da base de conhecimento de sistemas nebulosos para a resolução de problemas de classificação de padrões, através de duas abordagens diferentes: aprendizado genético da BD *a priori* e o aprendizado genético da BD *a posteriori*. Esta investigação focalizou em alguns aspectos, tais como, comparar o desempenho das duas abordagens, averiguar o impacto do tamanho da base de regras nos resultados de classificação do sistema e verificar a influência da granularidade das variáveis no desempenho do sistema.

No geral, ambas as abordagens apresentaram resultados muito próximos, sendo que em grande parte dos casos, a abordagem de construção genética da BD gera um número maior de regras em relação à abordagem de sintonia. Isto pode ser atribuído ao fato de que nesta abordagem, a base de regras não é fixa, ou seja, a cada iteração do algoritmo genético, uma nova base de regras é construída e avaliada. Sendo que isto não ocorre na outra abordagem,

onde apenas uma única base de regras previamente definida é considerada durante todo o processo genético.

Com relação ao tamanho da base de regras, este sempre esteve relacionado com o número de variáveis presentes nos testes, ou seja, quanto menor o número de variáveis selecionadas, menor foi o tamanho da base de regras. O motivo disto ocorrer pode ser explicado pelo seguinte fato. Com a redução no número de variáveis dos conjuntos de dados nos experimentos, tornamos estes conjuntos mais simples e menores, conseqüentemente, um número menor de regras é suficiente para expressar as informações presentes nestes conjuntos. Ao mesmo tempo, a taxa de classificação do sistema melhora quando menos variáveis são usadas, isto porque, o tamanho do espaço de busca do algoritmo genético é proporcional ao número de variáveis, ou seja, quanto menor for o número de variáveis, menor é o espaço de busca e mais fácil é a sua exploração.

A influência da granularidade das variáveis lingüísticas no desempenho do sistema foi verificada através da variação no número de conjuntos nebulosos por variável. O que se constatou foi sempre uma redução na classificação do sistema à medida que a granularidade das variáveis aumentava.

Trabalhos Futuros

Através deste trabalho, vários outros aspectos relacionados ao projeto de sistemas nebulosos surgiram, mas que não foram abordados. Contudo, algumas modificações no algoritmo são propostas para que estes aspectos sejam tratados em trabalhos futuros:

- Permitir no processo de aprendizado genético a alteração no número de conjuntos nebulosos por variável lingüística. Trabalhos encontrados na literatura, como em [Cordón et al. 2000], mostra que nem sempre uma partição nebulosa uniforme é a melhor opção;
- Incluir um processo de seleção de atributos, a fim de excluir atributos que não possuem um grau de importância relevante para o problema de classificação em questão, tornando o processo de busca mais fácil;
- Testar novos operadores genéticos;
- Testar a cooperação com um outro algoritmo genético responsável pela construção da base de regras e realizar experimentos em problemas de controle.

Apêndice A

Método de Aprendizado de Wang e Mendel

O processo de geração de regras proposto por Wang e Mendel é amplamente conhecido devido a sua simplicidade e bom desempenho. A geração das regras é feita através dos seguintes passos:

1. *Considere uma partição nebulosa para os espaços das variáveis de entrada e saída:* Esta partição pode ser obtida de um especialista (se este é disponível) ou por um processo de normalização. Se a segunda opção for escolhida, defina uma partição nebulosa para os espaços das variáveis de entrada e saída dividindo cada universo de discurso em um número igual ou não de regiões, selecione um tipo de função de pertinência e designe um conjunto nebuloso a cada sub-espaço.
2. *Gere um conjunto preliminar de regras lingüísticas:* Este conjunto será formado pela melhor regra ao cobrir cada exemplo (dados de entrada-saída) contido no conjunto de dados. A estrutura destas regras é obtida por um exemplo específico, isto é, um vetor real de dimensão $n + 1$ (n entradas e 1 saída), e atribuindo a cada uma das variáveis o melhor rótulo lingüístico que cobre todo o vetor.
3. *Designe um grau de importância as regras:* Seja $R_1 = \text{Se } x_1 \text{ é } A_1 \text{ E } \dots \text{ E } x_n \text{ é } A_n \text{ Então } y \text{ é } B$ uma regra lingüística gerada de um exemplo $e_1 = (x_1^l, \dots, x_n^l, y^l)$. O grau de importância será obtido como se segue:

$$G(R_1) = \mu_{A_1}(x_1^l) \dots \mu_{A_n}(x_n^l) \cdot \mu_b(y^l).$$

Obtenha uma BR final a partir do conjunto preliminar de regras nebulosas: A regra com maior grau de importância é escolhida entre as regras que possuem os mesmos antecedentes.

Apêndice B

Resultados para o Conjunto de Dados Auto-Mpg

Neste apêndice serão relatados os resultados obtidos ao problema Auto-Mpg, para os casos onde a granularidade das variáveis assumiram valores cinco, sete e nove. Os dados mostrados nas tabelas são **ECC** (exemplos classificados corretamente), **Regras** (número de regras geradas), **Desvio Padrão** (desvio padrão calculado sobre o ECC) e **Granularidade** (número de conjuntos nebulosos por variável lingüística). O algoritmo genético utilizado pelas duas abordagens foi executado independentemente cinco vezes para cada tipo de teste, sendo assim, os resultados de ECC e Regras expressam a média aritmética destes cinco resultados. Por outro lado, o Desvio Padrão expressa o desvio padrão sobre os resultados de ECC.

Tabela B.1: Resultados para a construção da BD com granularidade 5 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 125,8 | 135,4 | 136,6 | 152 | 154,2 |
| Regras | 67,6 | 61,8 | 61,6 | 38,4 | 30,6 |
| Desvio Padrão | 3,4205 | 2,3021 | 1,1401 | 1,4142 | 1,0954 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

Tabela B.2: Resultados para a otimização da BD com granularidade 5 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 126,6 | 138,6 | 143,4 | 151 | 152,6 |
| Regras | 66,4 | 61,6 | 61,6 | 40 | 31,6 |
| Desvio Padrão | 7,1274 | 3,2863 | 1,8165 | 2,3452 | 3,2093 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

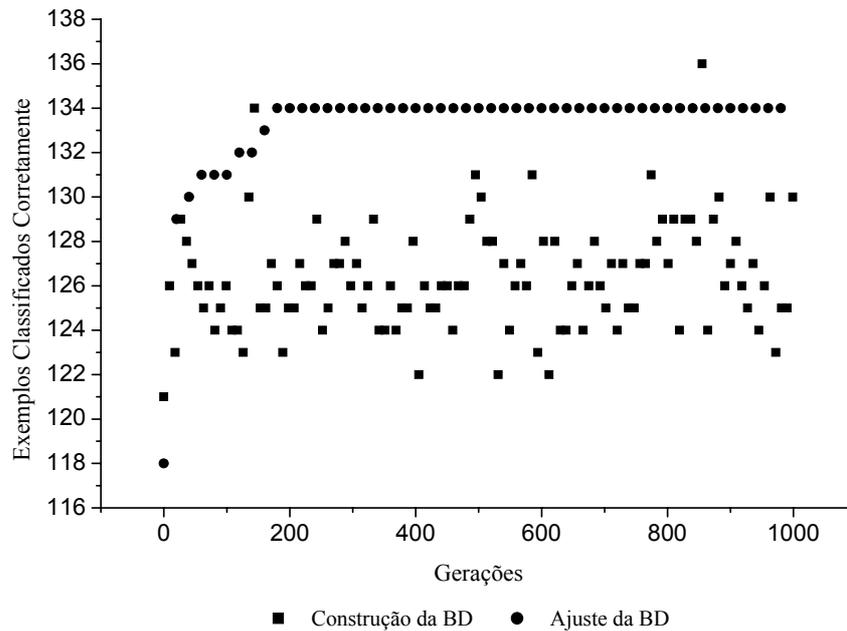


Figura B.1: Comparação entre as abordagens do desempenho do AG com 8 variáveis e granularidade 5 (Auto-Mpg).

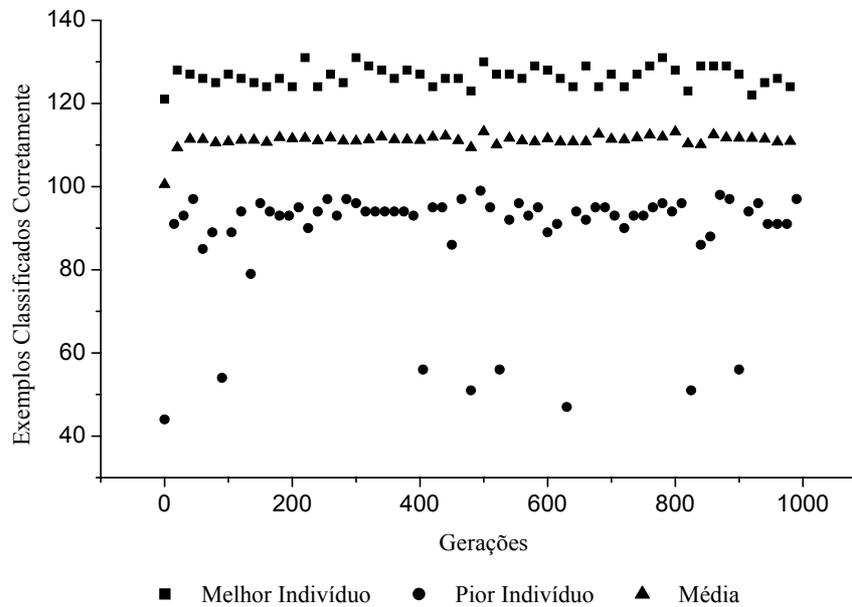


Figura B.2: Desempenho dos indivíduos na construção da BD com 8 variáveis e granularidade 5 (Auto-Mpg).

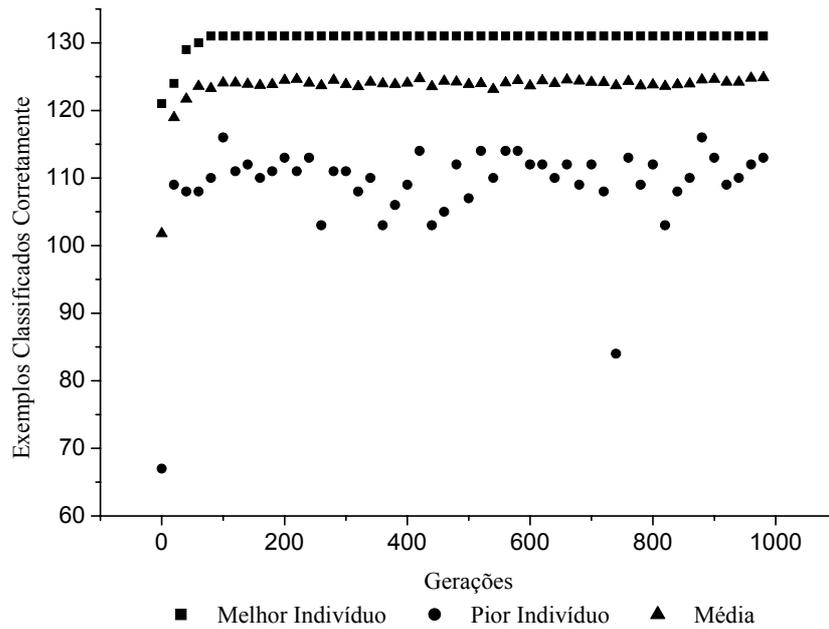


Figura B.3: Desempenho dos indivíduos na otimização da BD com 8 variáveis e granularidade 5 (Auto-Mpg).

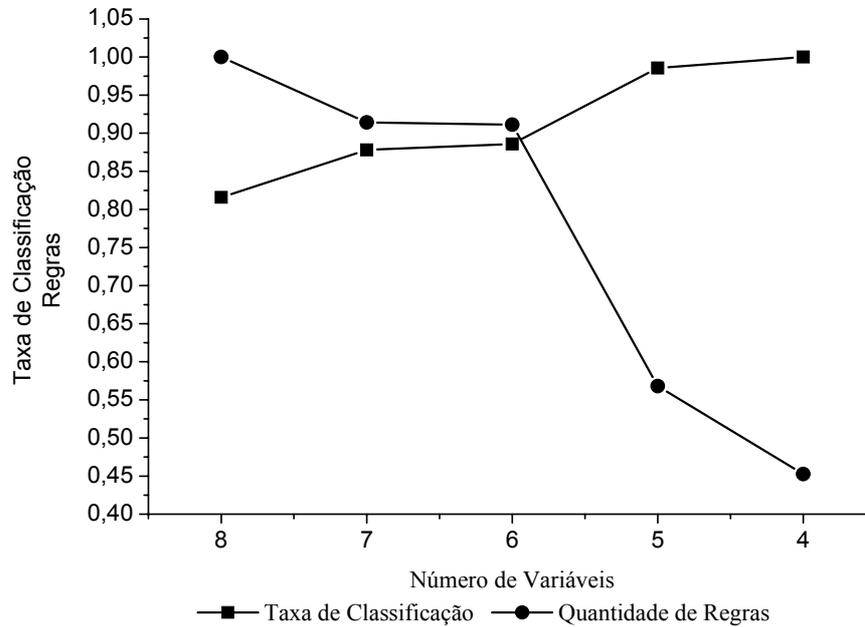


Figura B.4: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 5 (Auto-Mpg).

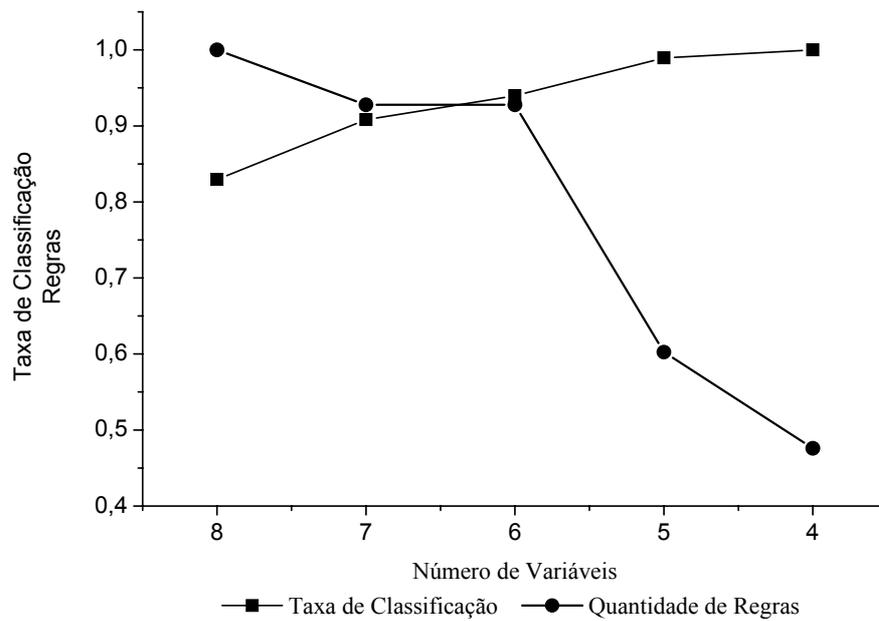


Figura B.5: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 5 (Auto-Mpg).

Tabela B.3: Resultados para a construção da BD com granularidade 7 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 107,2 | 121,4 | 126,2 | 145,2 | 146,6 |
| Regras | 72 | 75,4 | 74,2 | 58,4 | 47,8 |
| Desvio Padrão | 2,1679 | 3,6469 | 4,8166 | 4,4944 | 4,0373 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

Tabela B.4: Resultados para a otimização da BD com granularidade 7 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 121 | 132,2 | 138 | 141,2 | 144,6 |
| Regras | 66 | 68,2 | 74,6 | 55,6 | 44,8 |
| Desvio Padrão | 4,5276 | 3,9623 | 4,2426 | 6,0991 | 3,6469 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

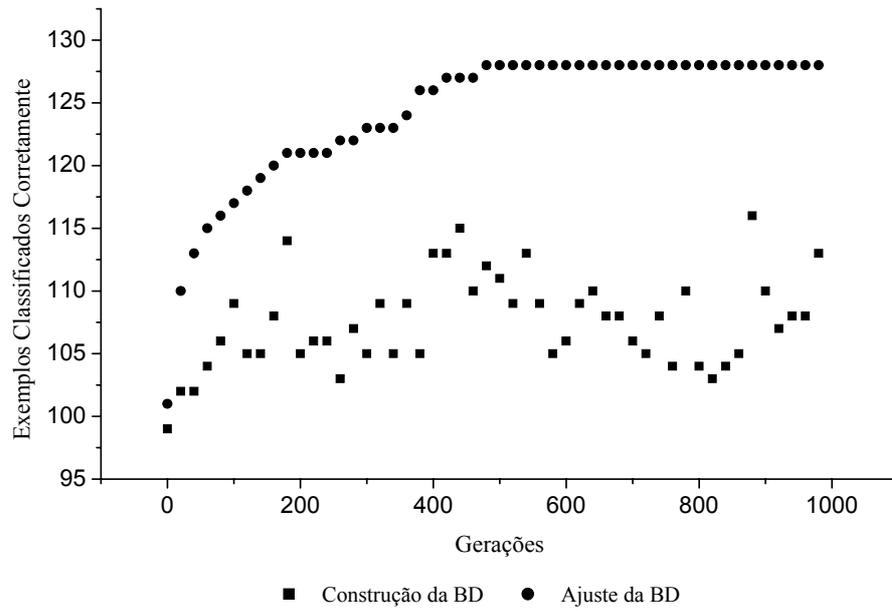


Figura B.6: Comparação entre as abordagens do desempenho do AG com 8 variáveis e granularidade 7 (Auto-Mpg).

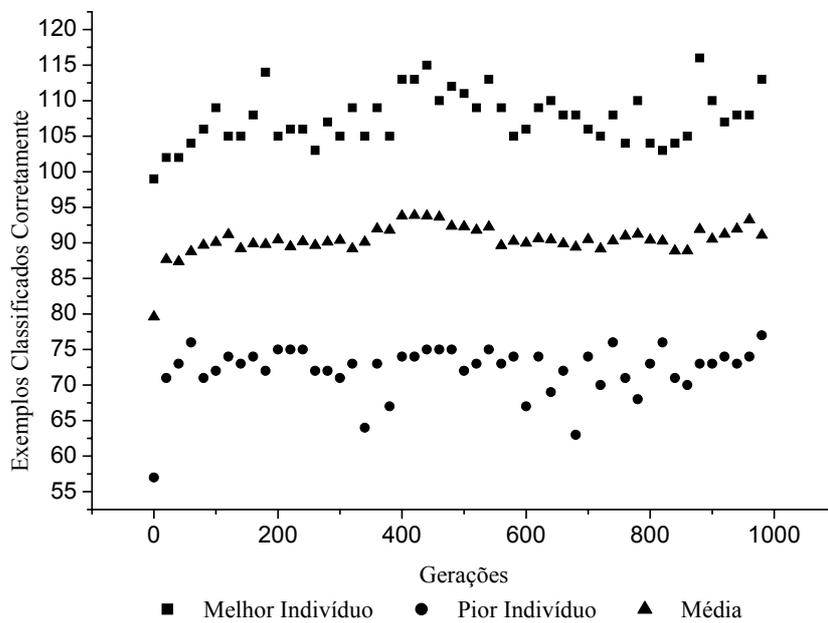


Figura B.7: Desempenho dos indivíduos na construção da BD com 8 variáveis e granularidade 7 (Auto-Mpg).

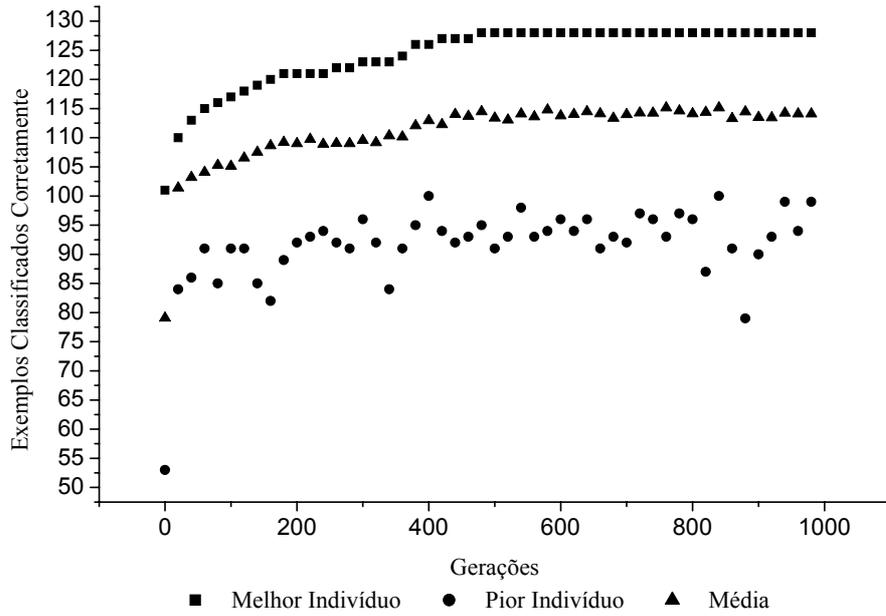


Figura B.8: Desempenho dos indivíduos na otimização da BD com 8 variáveis e granularidade 7 (Auto-Mpg).

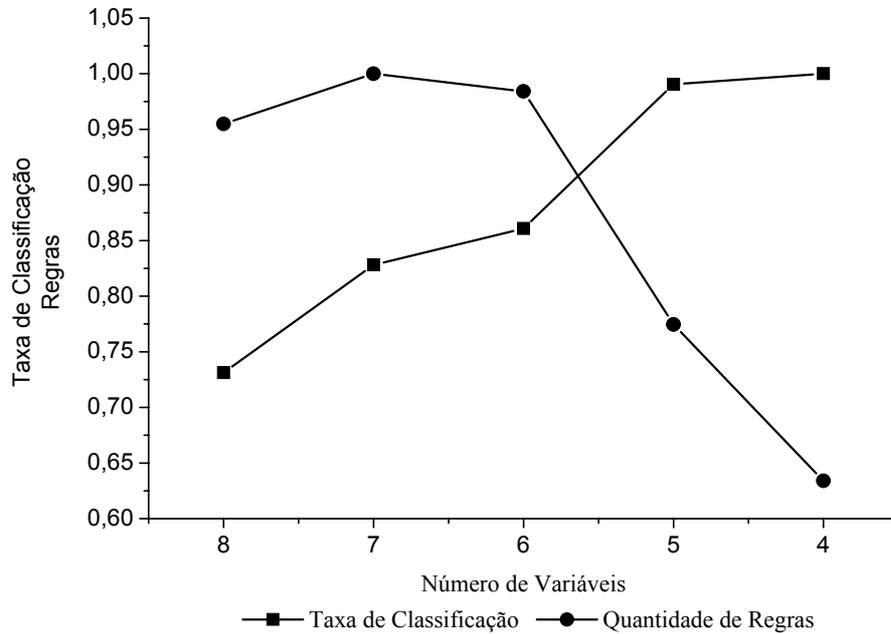


Figura B.9: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 7 (Auto-Mpg).

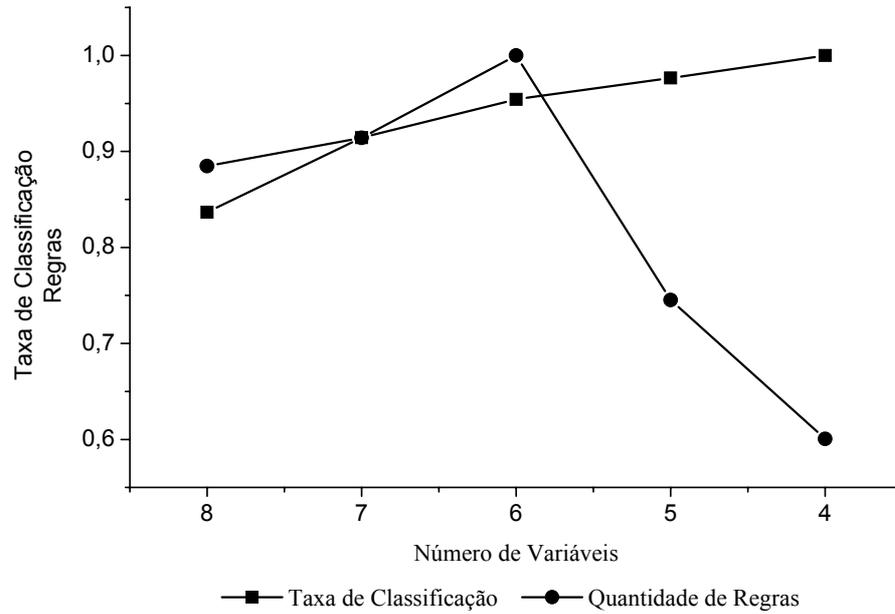


Figura B.10: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 7 (Auto-Mpg).

Tabela B.5: Resultados para a construção da BD com granularidade 9 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 88 | 102 | 108,8 | 134,8 | 136 |
| Regras | 64,4 | 71 | 70,2 | 66,8 | 56 |
| Desvio Padrão | 2,9154 | 2,3452 | 3,1144 | 6,3796 | 2,9154 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

Tabela B.6: Resultados para a otimização da BD com granularidade 9 (Auto-Mpg).

| Variáveis | 8 | 7 | 6 | 5 | 4 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 111,2 | 120,4 | 125,4 | 138,6 | 140,2 |
| Regras | 59,2 | 72,2 | 70,8 | 63,2 | 53,6 |
| Desvio Padrão | 6,2209 | 5,5497 | 5,9413 | 1,5165 | 3,7013 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

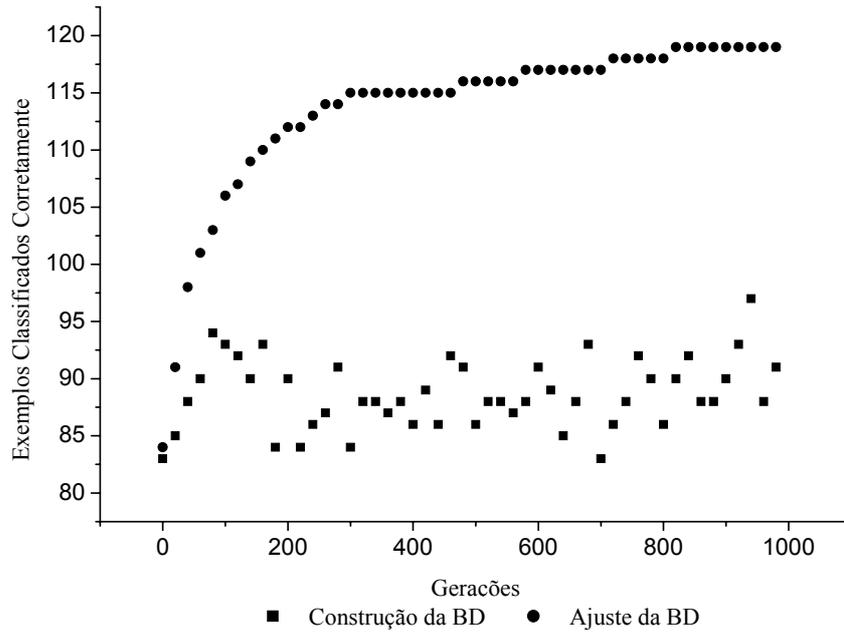


Figura B.11: Comparação entre as abordagens do desempenho do AG com 8 variáveis e granularidade 9 (Auto-Mpg).

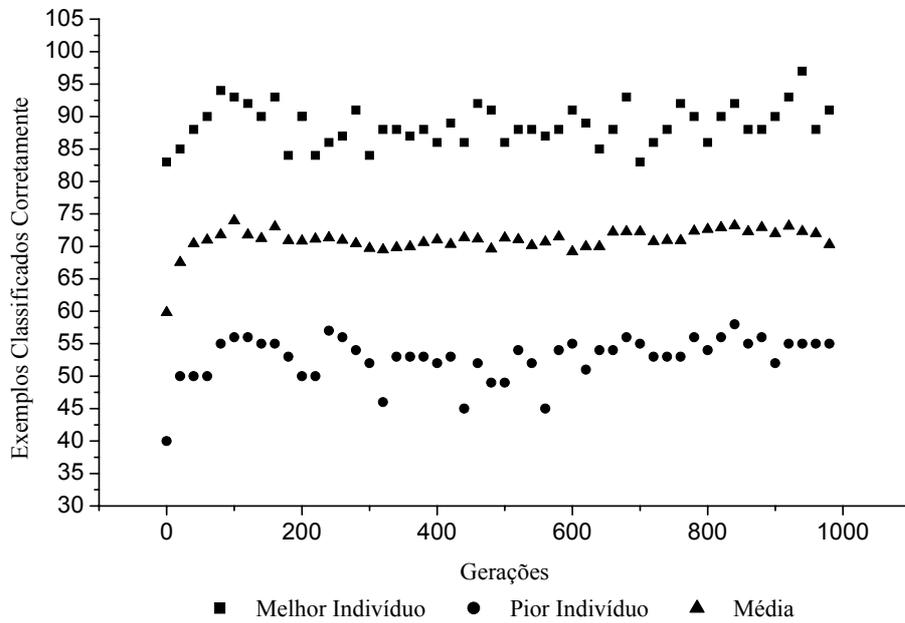


Figura B.12: Desempenho dos indivíduos na construção da BD com 8 variáveis e granularidade 9 (Auto-Mpg).

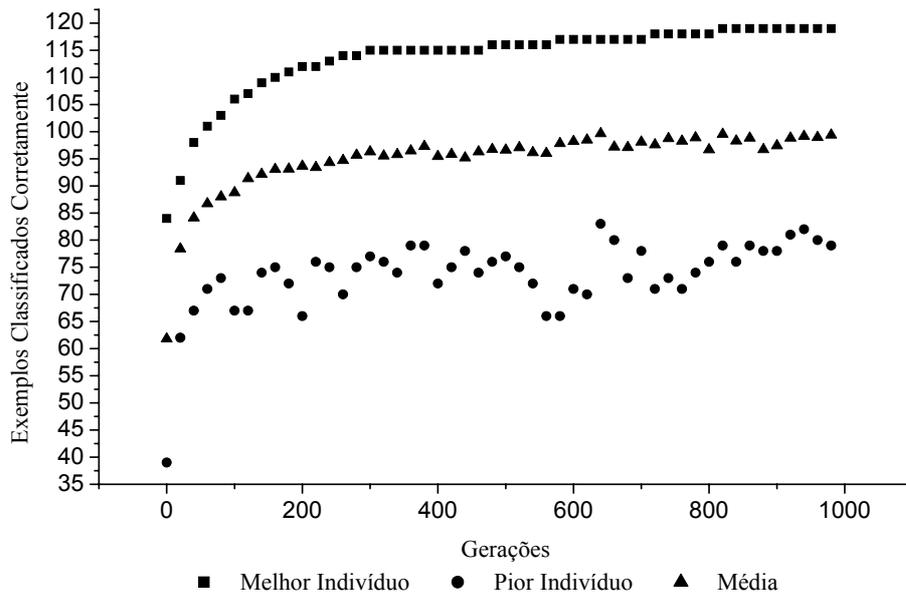


Figura B.13: Desempenho dos indivíduos na otimização da BD com 8 variáveis e granularidade 9 (Auto-Mpg).

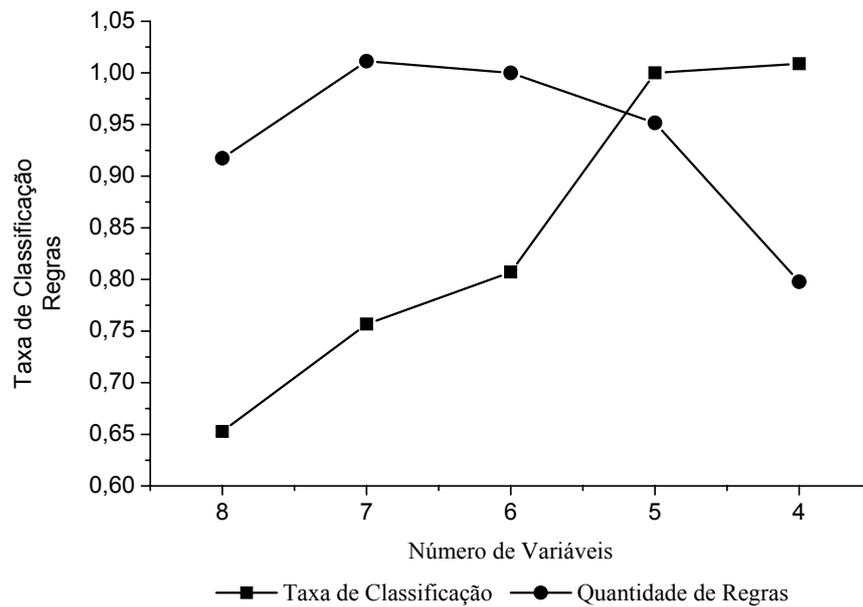


Figura B.14: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 9 (Auto-Mpg).

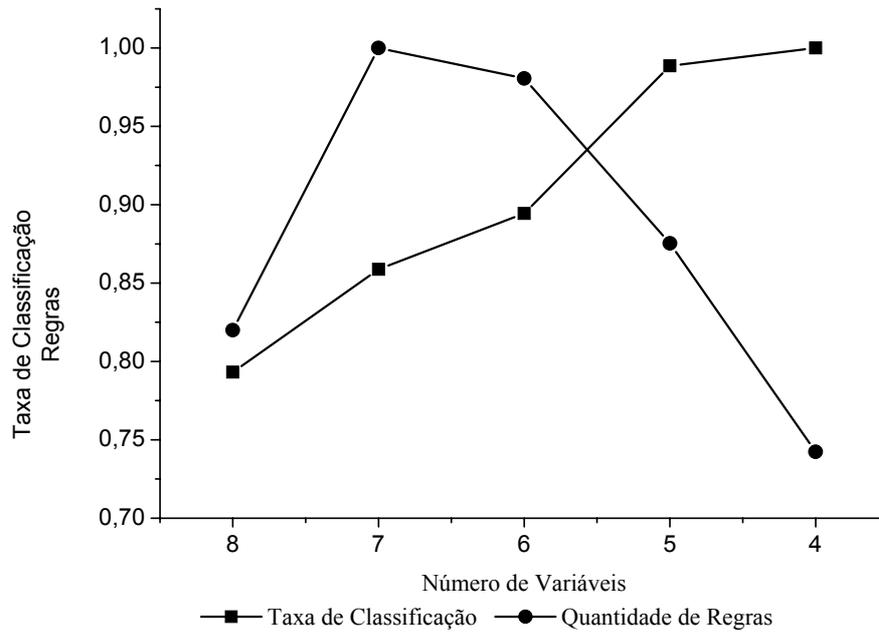


Figura B.15: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 9 (Auto-Mpg).

Apêndice C

Resultados para o Conjunto de Dados Iris Plants

Neste apêndice serão relatados os resultados obtidos ao problema Iris Plants, para os casos onde a granularidade das variáveis assumiram valores cinco, sete e nove. Os dados mostrados nas tabelas são **ECC** (exemplos classificados corretamente), **Regras** (número de regras geradas), **Desvio Padrão** (desvio padrão calculado sobre o ECC) e **Granularidade** (número de conjuntos nebulosos por variável lingüística). O algoritmo genético utilizado pelas duas abordagens foi executado independentemente cinco vezes para cada tipo de teste, sendo assim, os resultados de ECC e Regras expressam a média aritmética destes cinco resultados. Por outro lado, o Desvio Padrão expressa o desvio padrão sobre os resultados de ECC.

Tabela C.1: Resultados para a construção da BD com granularidade 5 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 59,2 |
| Regras | 26 |
| Desvio Padrão | 0,8366 |
| Granularidade | 5 |

Tabela C.2: Resultados para a otimização da BD com granularidade 5 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 57,8 |
| Regras | 24 |
| Desvio Padrão | 1,4832 |
| Granularidade | 5 |

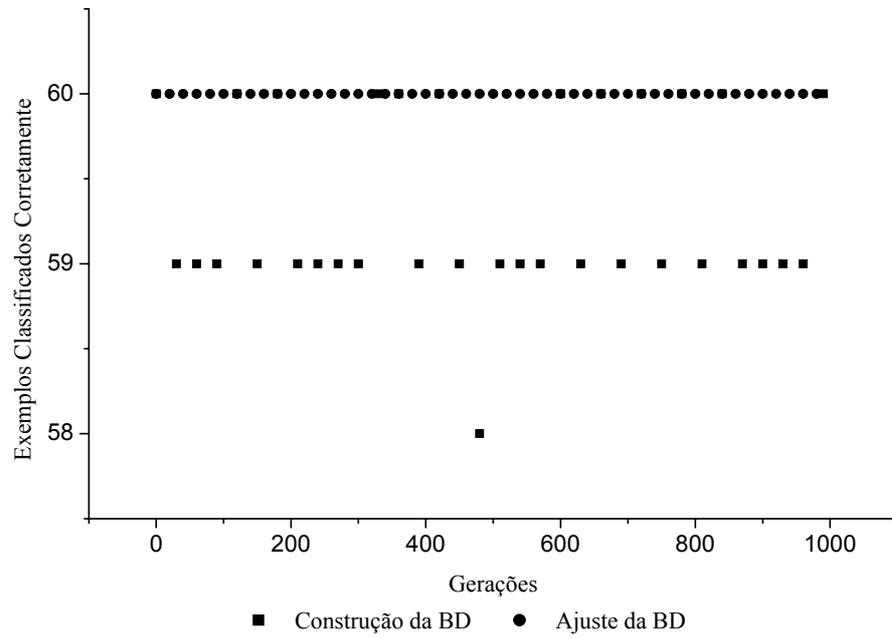


Figura C.1: Comparação entre as abordagens do desempenho do AG com granularidade 5 (Iris Plants).

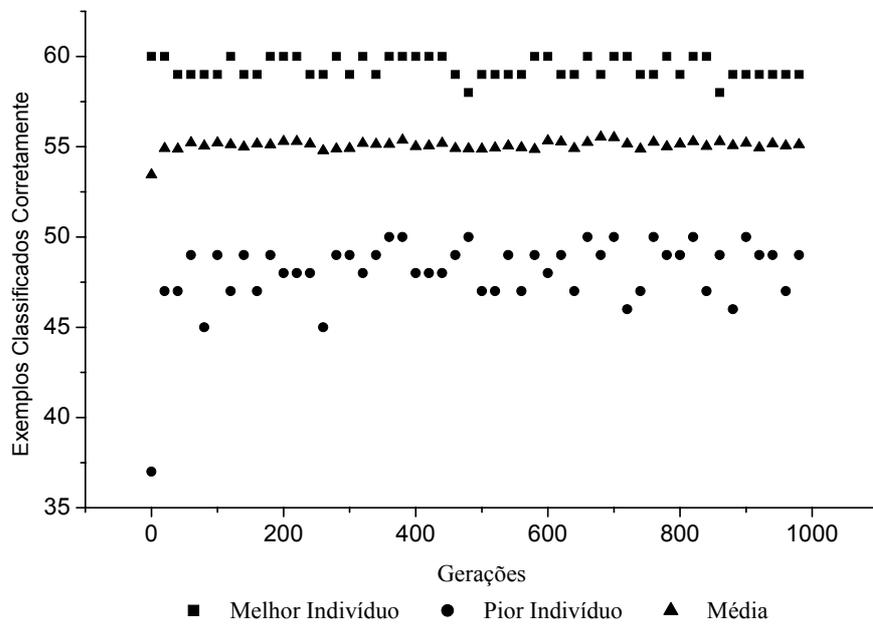


Figura C.2: Desempenho dos indivíduos na construção da BD com granularidade 5 (Iris Plants).

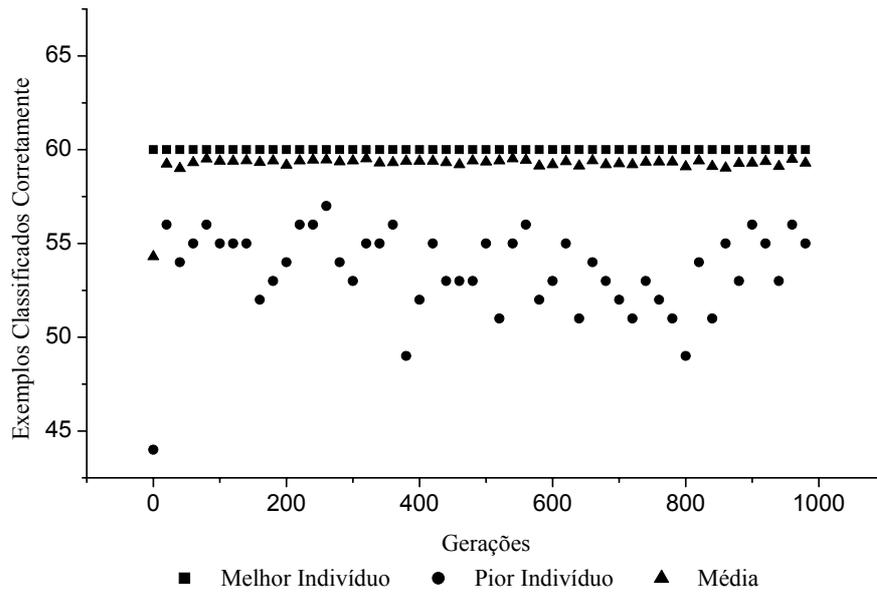


Figura C.3: Desempenho dos indivíduos na otimização da BD com granularidade 5 (Iris Plants).

Tabela C.3: Resultados para a construção da BD com granularidade 7 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 58,2 |
| Regras | 31,2 |
| Desvio Padrão | 0,8366 |
| Granularidade | 7 |

Tabela C.4: Resultados para a otimização da BD com granularidade 7 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 55,6 |
| Regras | 31,4 |
| Desvio Padrão | 1,1401 |
| Granularidade | 7 |

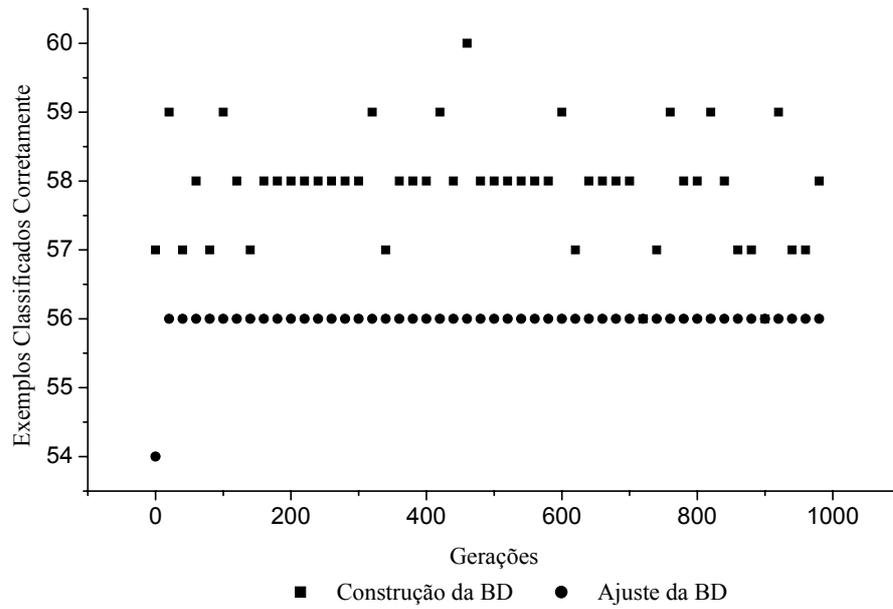


Figura C.4: Comparação entre as abordagens do desempenho do AG com granularidade 7 (Iris Plants).

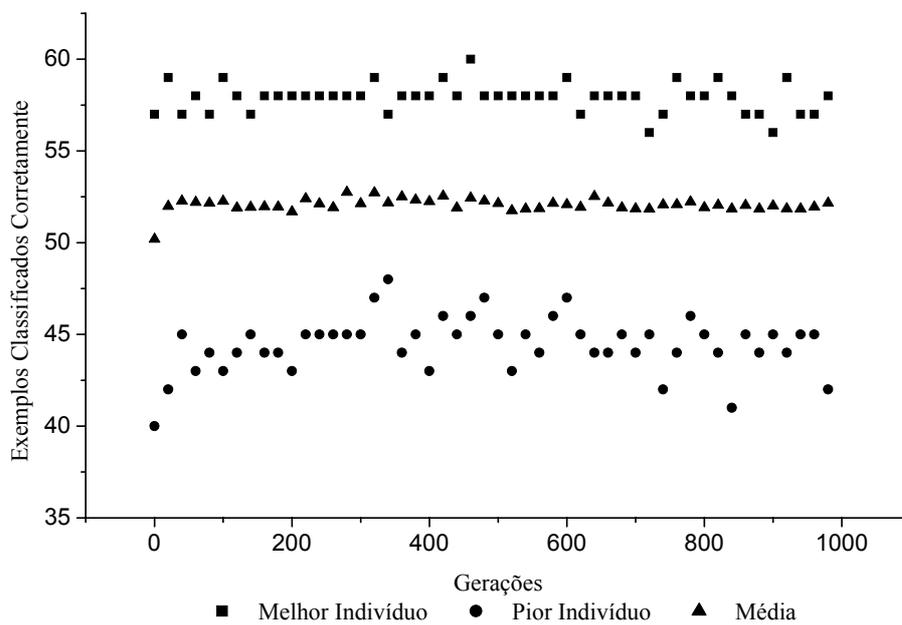


Figura C.5: Desempenho dos indivíduos na construção da BD com granularidade 7 (Iris Plants).

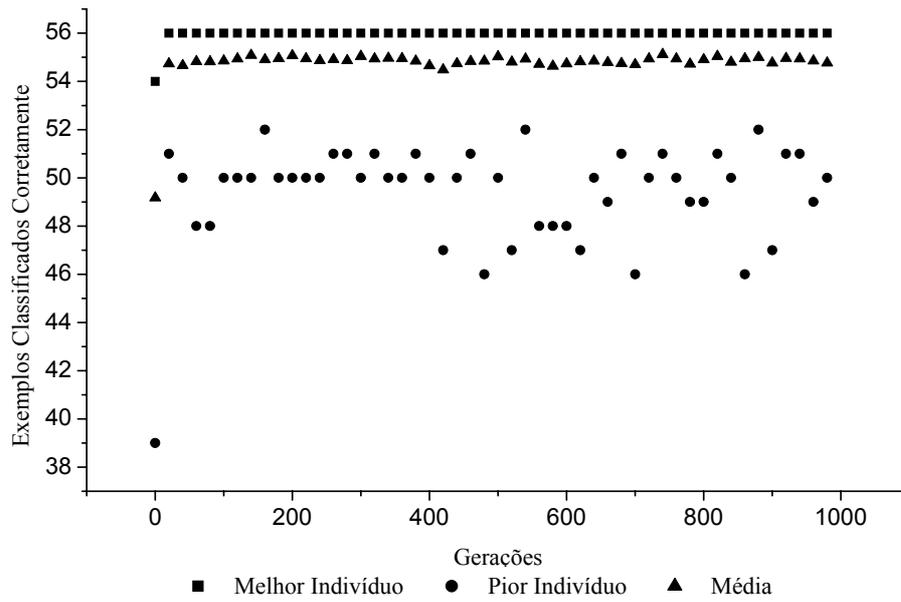


Figura C.6: Desempenho dos indivíduos na otimização da BD com granularidade 7 (Iris Plants).

Tabela C.5: Resultados para a construção da BD com granularidade 9 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 54,8 |
| Regras | 31,6 |
| Desvio Padrão | 0,8366 |
| Granularidade | 9 |

Tabela C.6: Resultados para a otimização da BD com granularidade 9 (Iris Plants).

| | |
|---------------|--------|
| Variáveis | 5 |
| ECC | 55,6 |
| Regras | 32,6 |
| Desvio Padrão | 1,5165 |
| Granularidade | 9 |

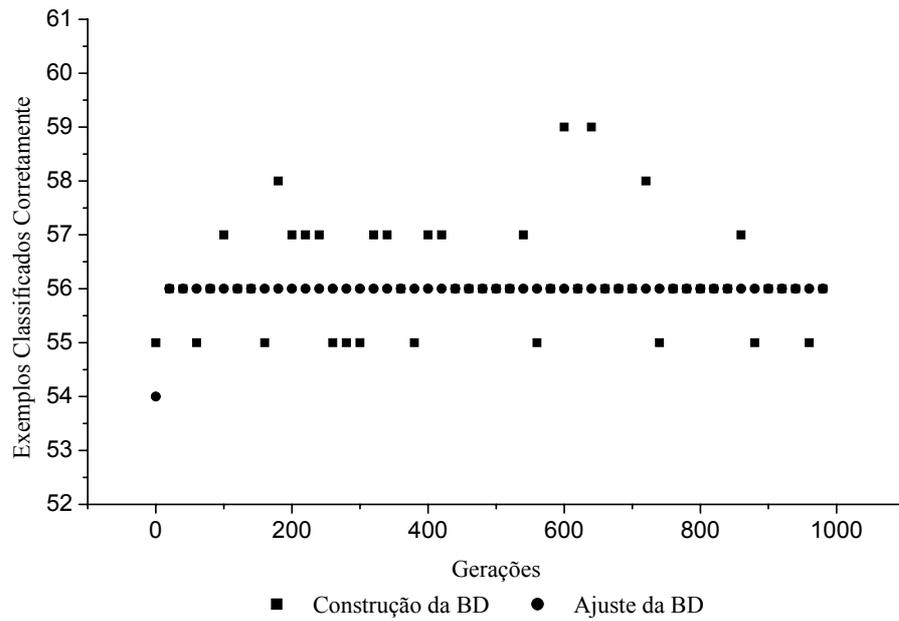


Figura C.7: Comparação entre as abordagens do desempenho do AG com granularidade 9 (Iris Plants).

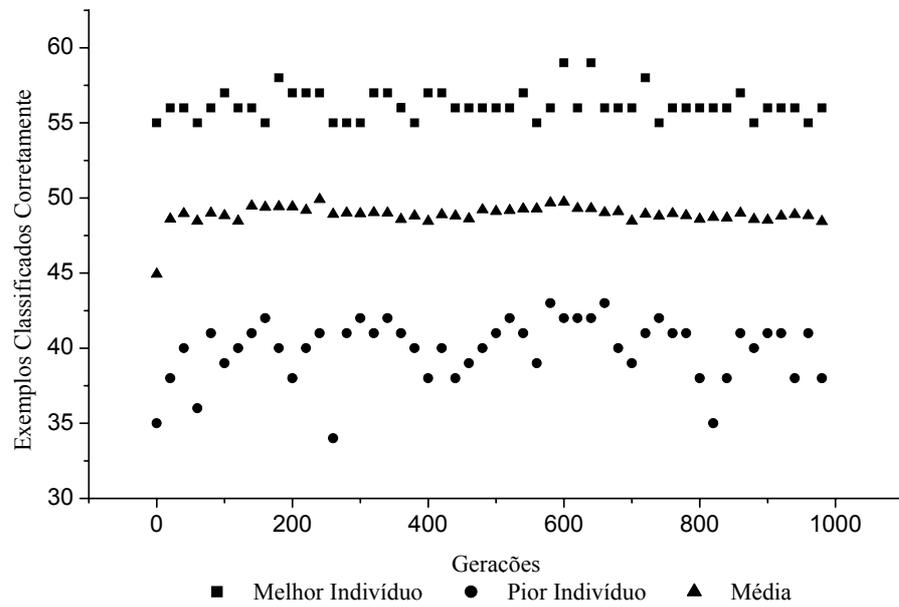


Figura C.8: Desempenho dos indivíduos na construção da BD com granularidade 9 (Iris Plants).

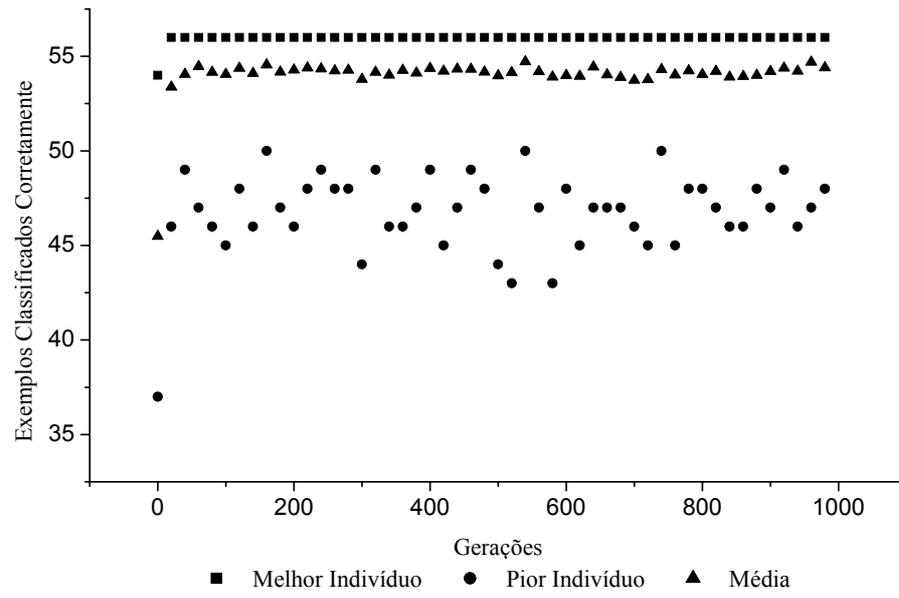


Figura C.9: Desempenho dos indivíduos na otimização da BD com granularidade 9 (Iris Plants).

Apêndice D

Resultados para o Conjunto de Dados Wine

Neste apêndice serão relatados os resultados obtidos ao problema Wine, para os casos onde a granularidade das variáveis assumiram valores cinco, sete e nove. Os dados mostrados nas tabelas são **ECC** (exemplos classificados corretamente), **Regras** (número de regras geradas), **Desvio Padrão** (desvio padrão calculado sobre o ECC) e **Granularidade** (número de conjuntos nebulosos por variável lingüística). O algoritmo genético utilizado pelas duas abordagens foi executado independentemente cinco vezes para cada tipo de teste, sendo assim, os resultados de ECC e Regras expressam a média aritmética destes cinco resultados. Por outro lado, o Desvio Padrão expressa o desvio padrão sobre os resultados de ECC.

Tabela D.1: Resultados para a construção da BD com granularidade 5 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 46,8 | 49,8 | 53,8 | 54,2 | 58,8 |
| Regras | 31,6 | 34 | 36 | 38,4 | 36,6 |
| Desvio Padrão | 1,7888 | 0,8366 | 2,9495 | 2,8635 | 0,8366 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

Tabela D.2: Resultados para a otimização da BD com granularidade 5 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 56,6 | 60,8 | 61 | 59,2 | 65 |
| Regras | 31,2 | 32,4 | 36,2 | 33,8 | 37,6 |
| Desvio Padrão | 5,0793 | 3,2710 | 2,3452 | 1,6431 | 1,2247 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

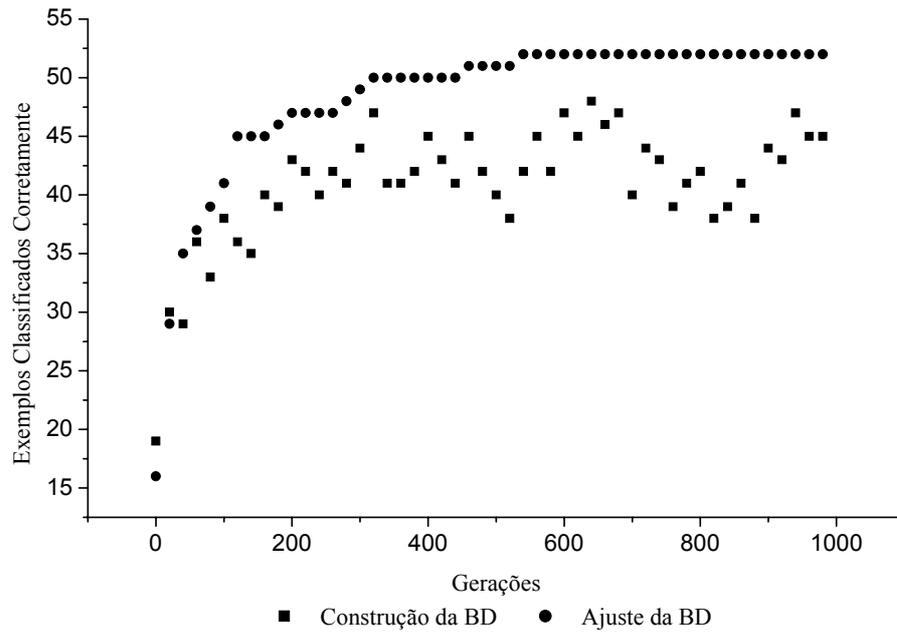


Figura D.1: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 5 (Wine).

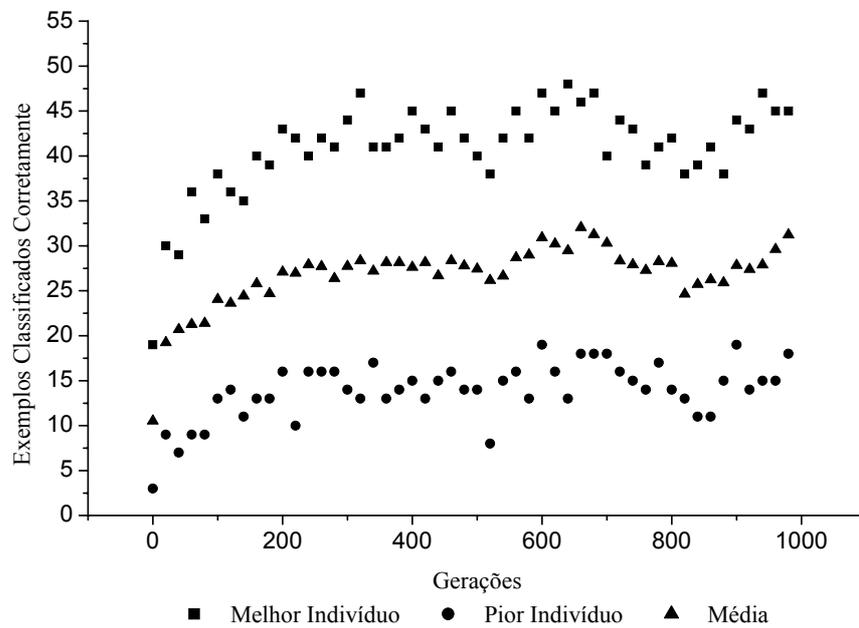


Figura D.2: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 5 (Wine).

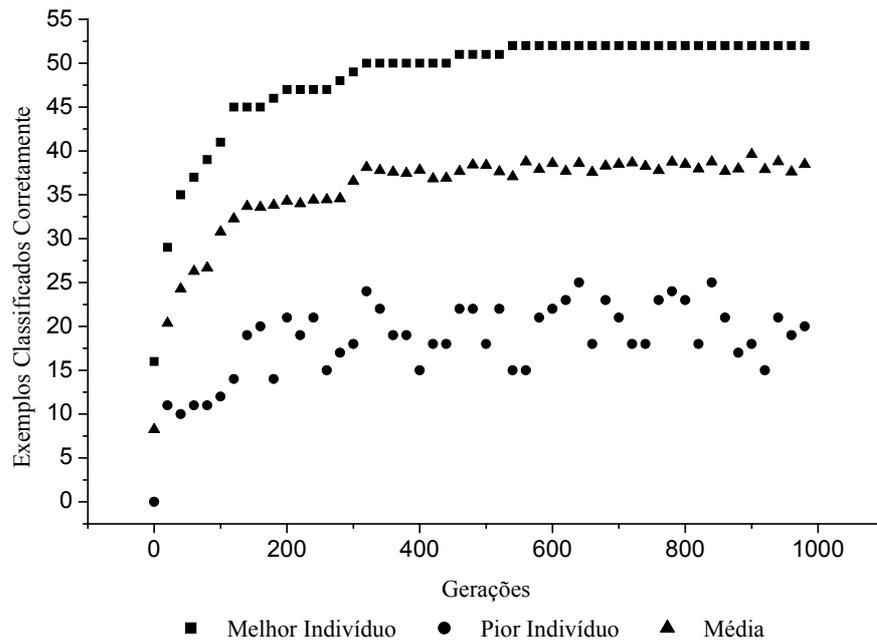


Figura D.3: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 5 (Wine).

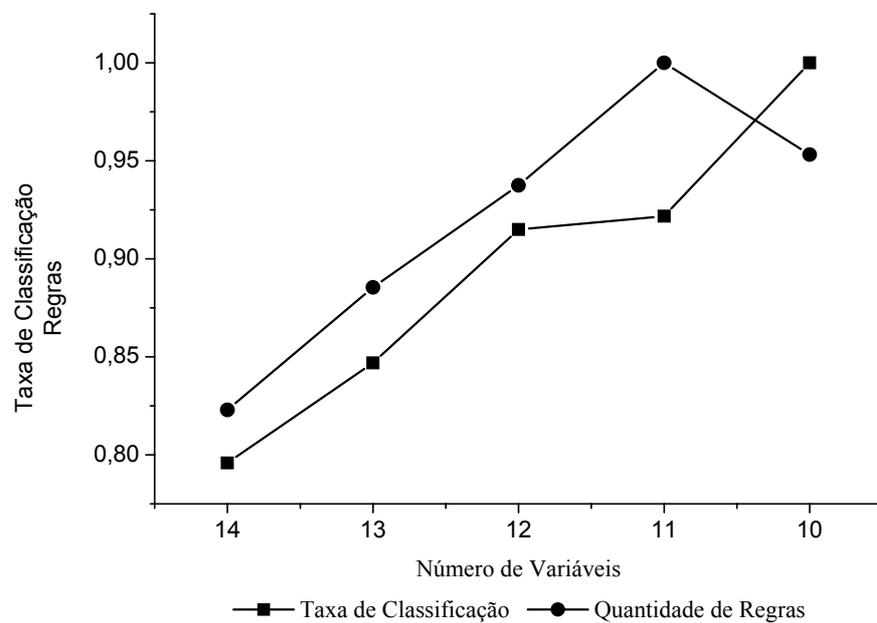


Figura D.4: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 5 (Wine).

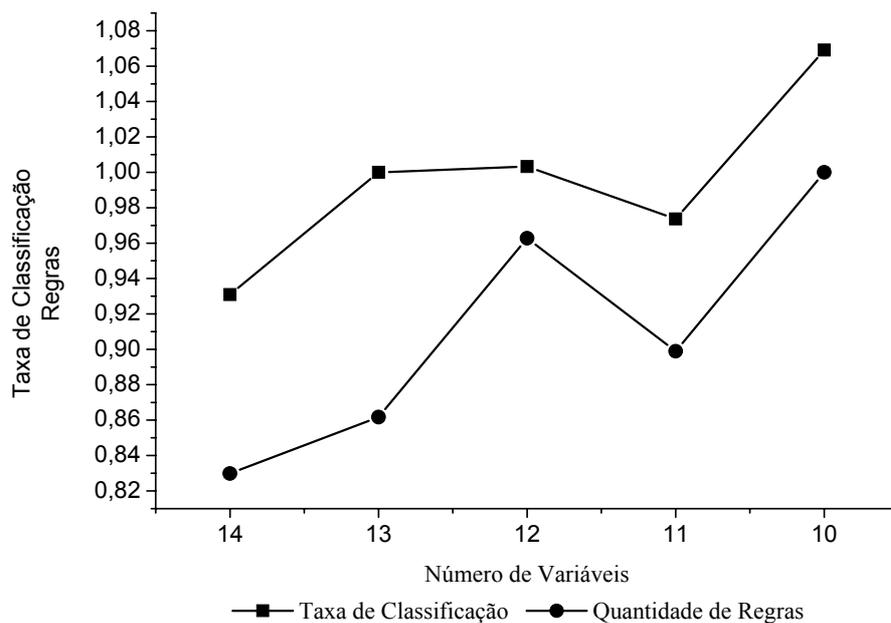


Figura D.5: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 5 (Wine).

Tabela D.3: Resultados para a construção da BD com granularidade 7 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 8,4 | 13,2 | 18,8 | 24,8 | 37,2 |
| Regras | 6,2 | 11,8 | 14,6 | 18 | 27 |
| Desvio Padrão | 0,8944 | 1,4832 | 0,8366 | 0,8366 | 3,9623 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

Tabela D.4: Resultados para a otimização da BD com granularidade 7 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 21,6 | 28,4 | 36,8 | 43,6 | 54,6 |
| Regras | 10,2 | 9,8 | 18,6 | 24,8 | 23,8 |
| Desvio Padrão | 2,0736 | 4,8270 | 0,8366 | 2,0736 | 2,5099 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

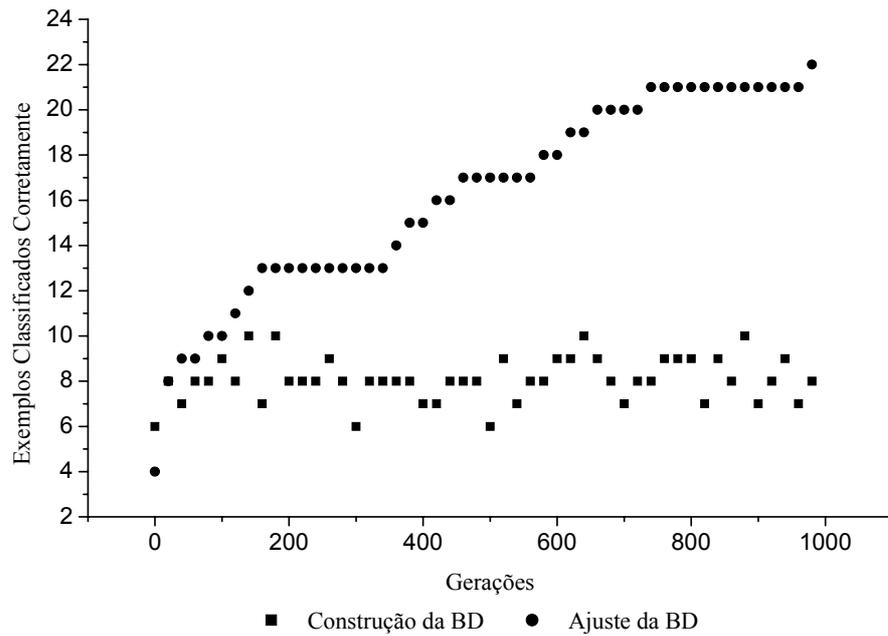


Figura D.6: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 7 (Wine).

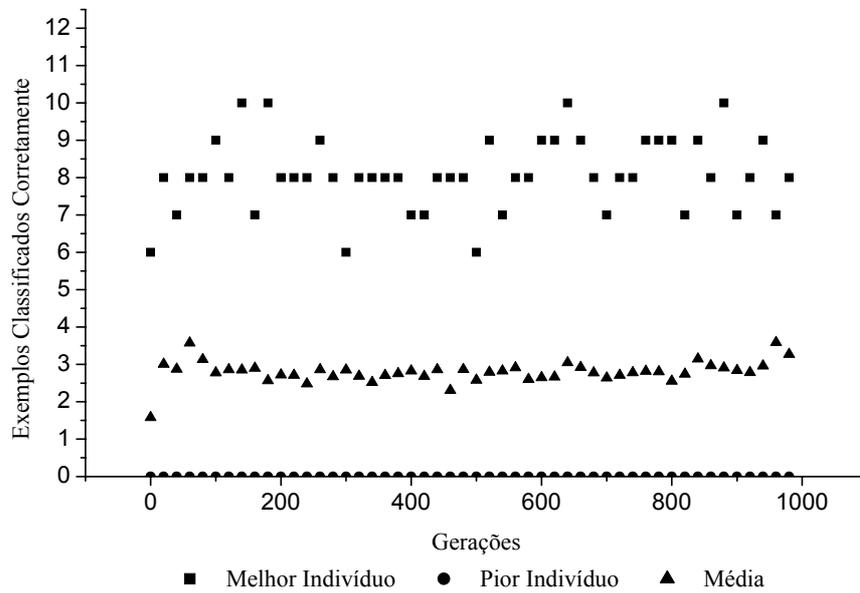


Figura D.7: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 7 (Wine).

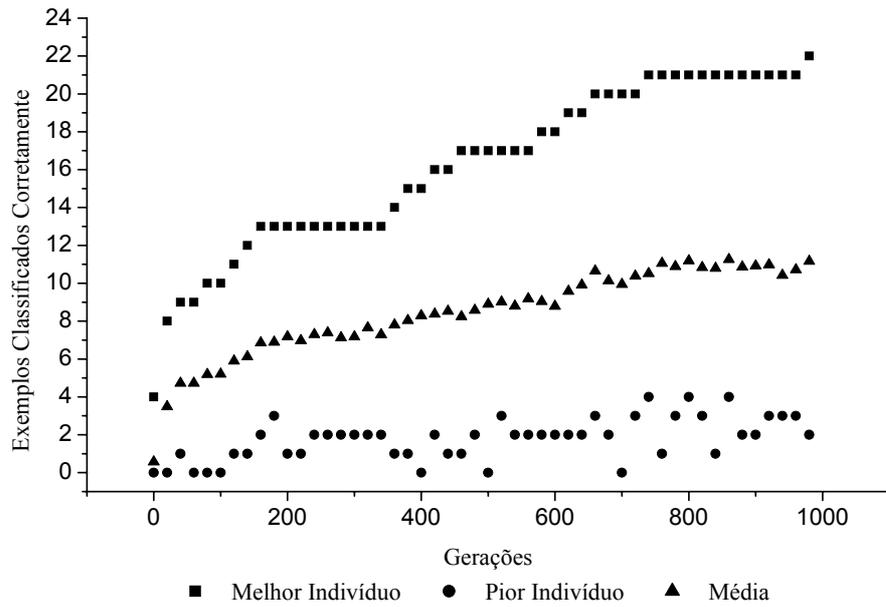


Figura D.8: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 7 (Wine).

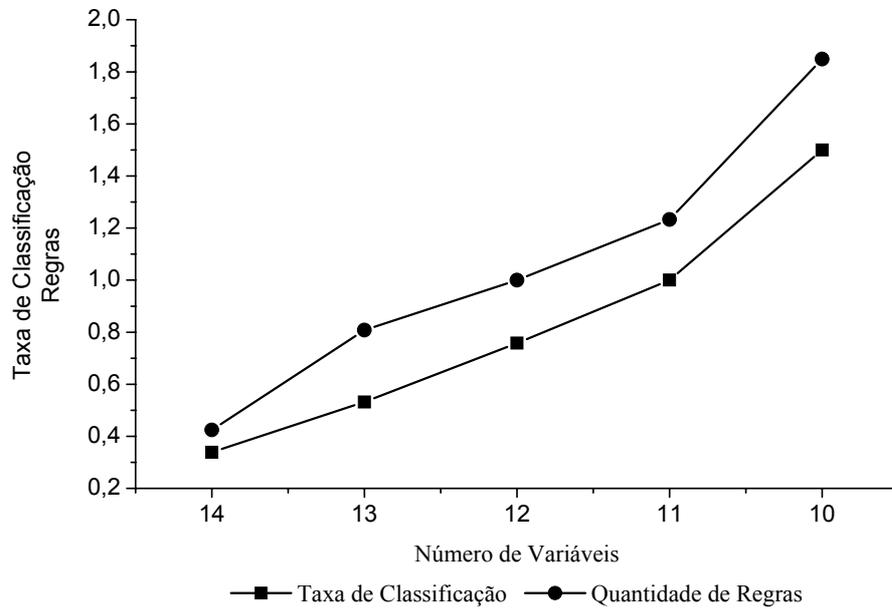


Figura D.9: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 7 (Wine).

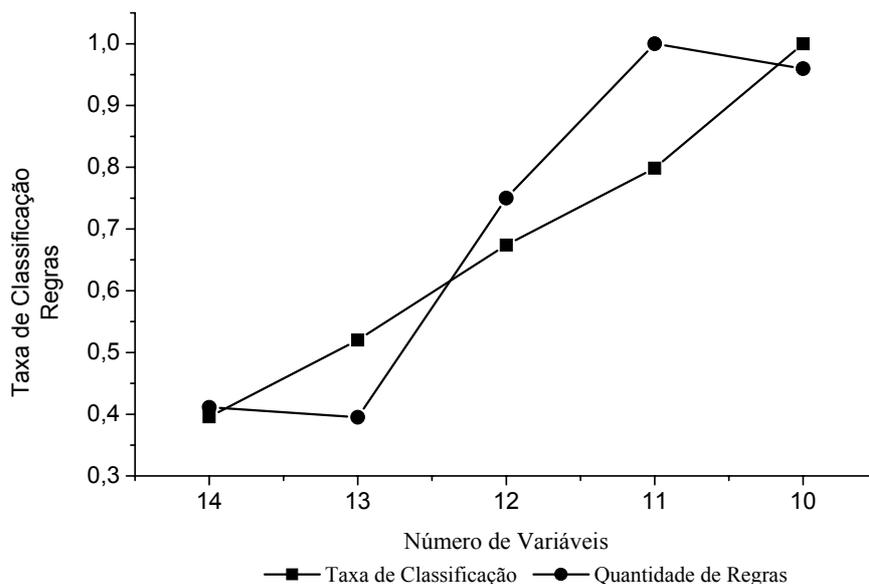


Figura D.10: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 7 (Wine).

Tabela D.5: Resultados para a construção da BD com granularidade 9 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|-----|--------|
| ECC | 2,2 | 4,2 | 6,2 | 9 | 13,4 |
| Regras | 2,2 | 4,2 | 6,2 | 8,4 | 13,2 |
| Desvio Padrão | 0,4472 | 0,4472 | 0,8366 | 1 | 1,3416 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

Tabela D.6: Resultados para a otimização da BD com granularidade 9 (Wine).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 6,8 | 11,2 | 15,8 | 20,2 | 27 |
| Regras | 2,2 | 5,4 | 4,4 | 12,8 | 13,4 |
| Desvio Padrão | 1,7888 | 1,3038 | 2,5884 | 2,2803 | 2,2360 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

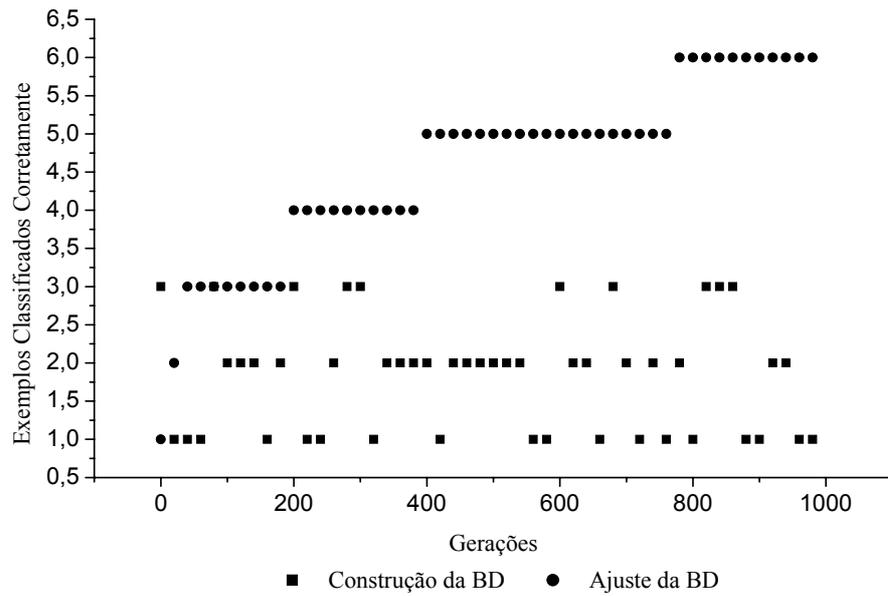


Figura D.11: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 9 (Wine).

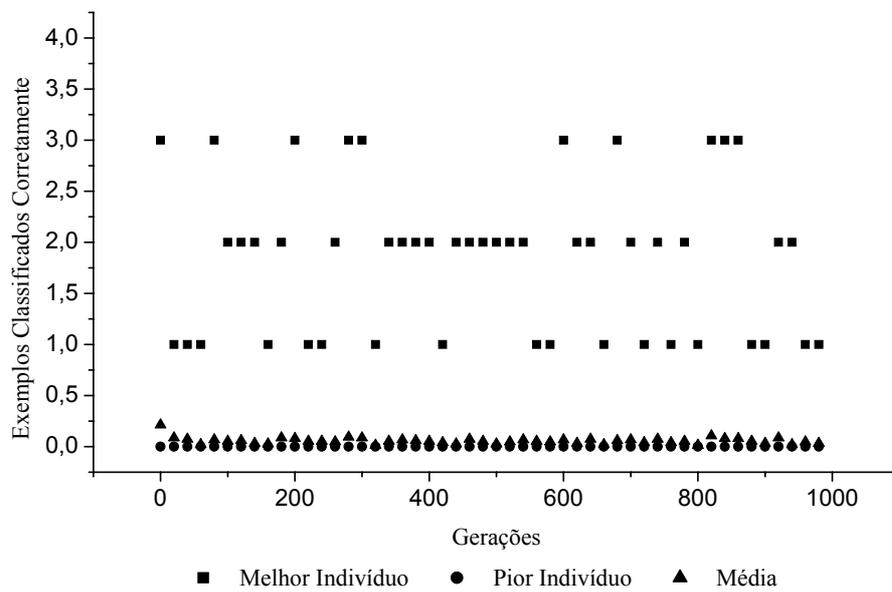


Figura D.12: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 9 (Wine).

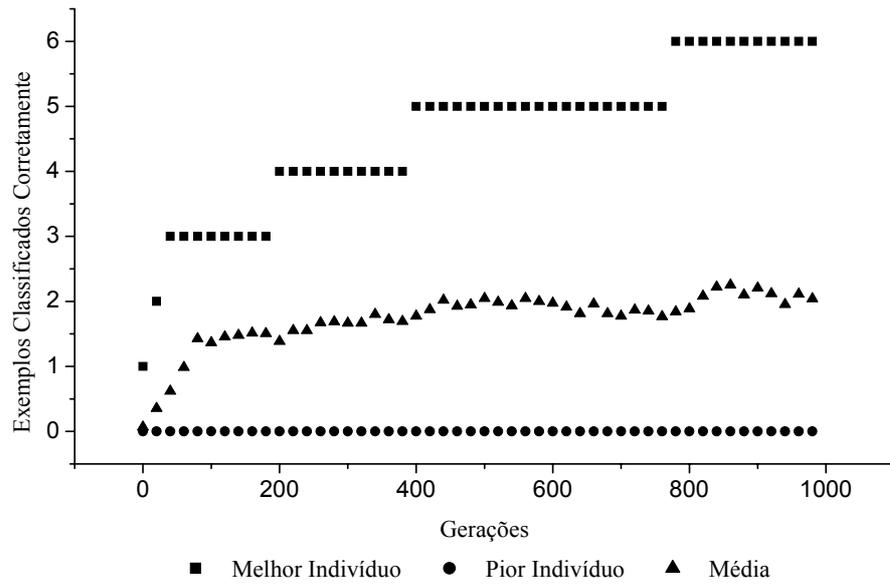


Figura D.13: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 9 (Wine).

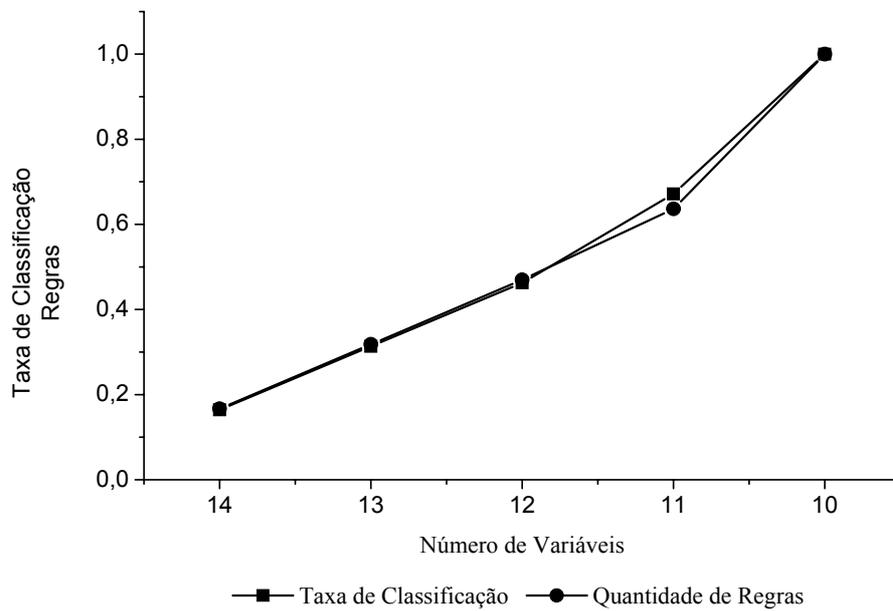


Figura D.14: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 9 (Wine).

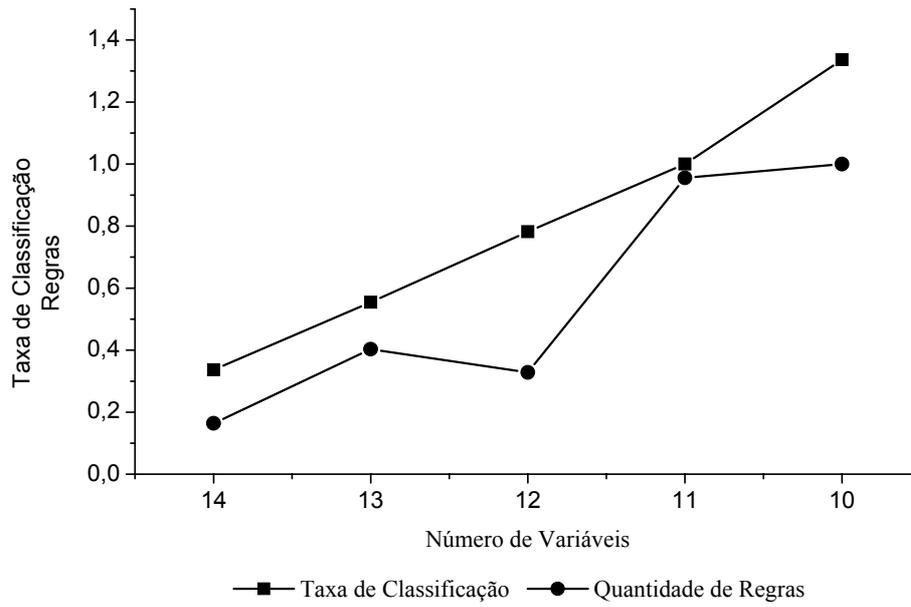


Figura D.15: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 9 (Wine).

Apêndice E

Resultados para o Conjunto Boston Housing

Neste apêndice serão relatados os resultados obtidos ao problema Boston Housing, para os casos onde a granularidade das variáveis assumiram valores cinco, sete e nove. Os dados mostrados nas tabelas são **ECC** (exemplos classificados corretamente), **Regras** (número de regras geradas), **Desvio Padrão** (desvio padrão calculado sobre o ECC) e **Granularidade** (número de conjuntos nebulosos por variável). O algoritmo genético utilizado pelas duas abordagens foi executado independentemente cinco vezes para cada tipo de teste, sendo assim, os resultados de ECC e Regras expressam a média aritmética destes cinco resultados. Por outro lado, o Desvio Padrão expressa o desvio padrão sobre os resultados de ECC.

Tabela E.1: Resultados para a construção da BD com granularidade 5 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 38,8 | 162,8 | 174,4 | 181 | 179,6 |
| Regras | 29 | 93,2 | 101,8 | 101,4 | 91,8 |
| Desvio Padrão | 2,1679 | 1,6431 | 1,5165 | 7,9056 | 3,4351 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

Tabela E.2: Resultados para a otimização da BD com granularidade 5 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 40,6 | 166,4 | 174,8 | 183,2 | 185,2 |
| Regras | 22,4 | 88,6 | 93,2 | 92,2 | 82,6 |
| Desvio Padrão | 3,5777 | 6,1886 | 4,6043 | 1,7888 | 1,3038 |
| Granularidade | 5 | 5 | 5 | 5 | 5 |

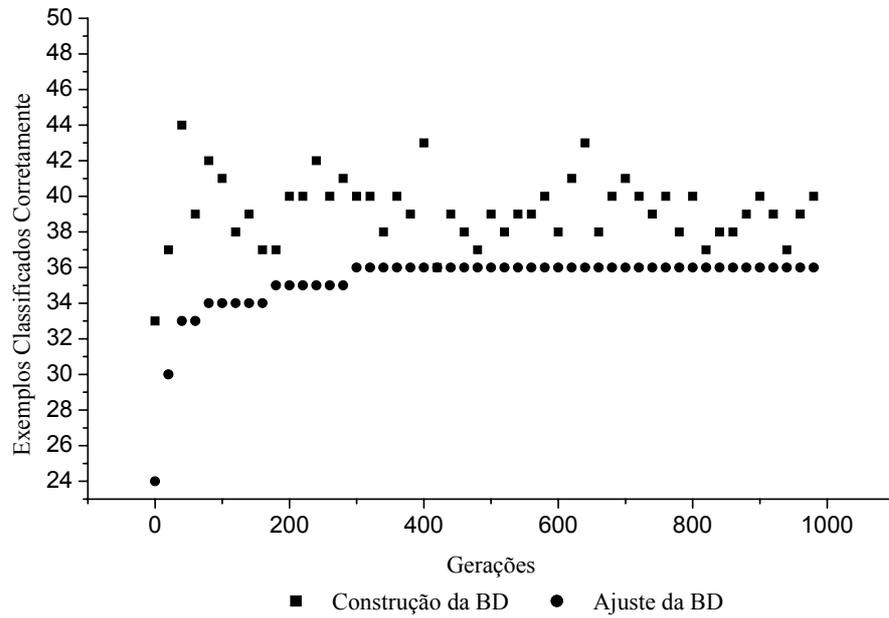


Figura E.1: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 5 (Boston Housing).

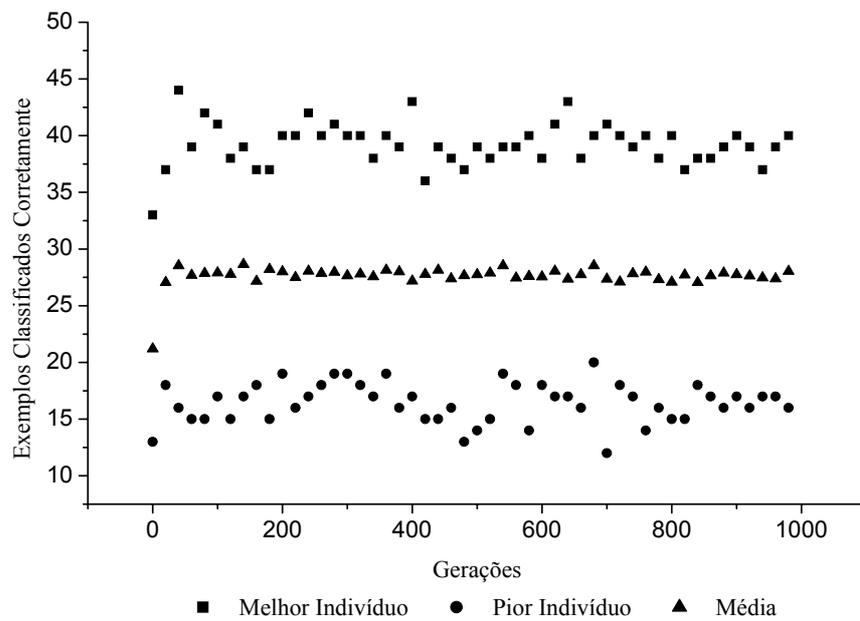


Figura E.2: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 5 (Boston Housing).

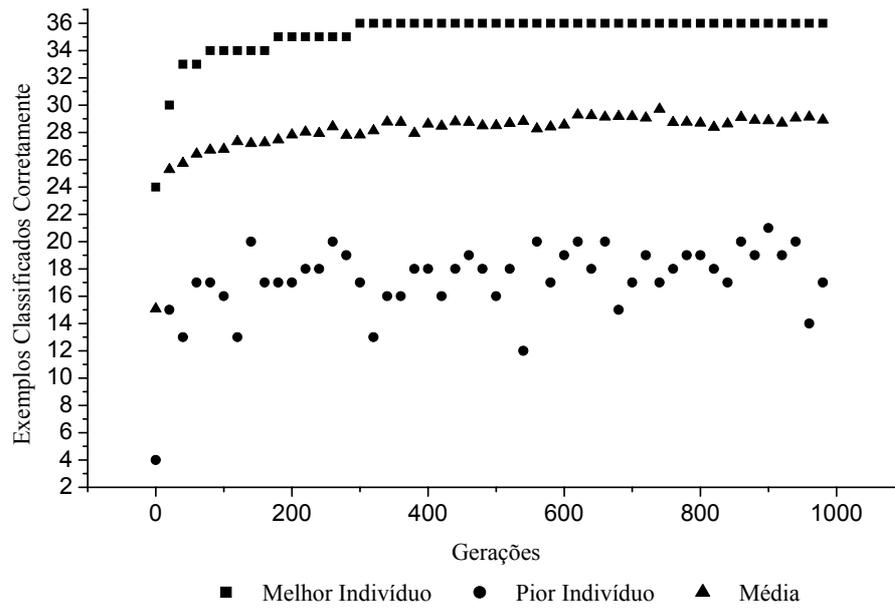


Figura E.3: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 5 (Boston Housing).

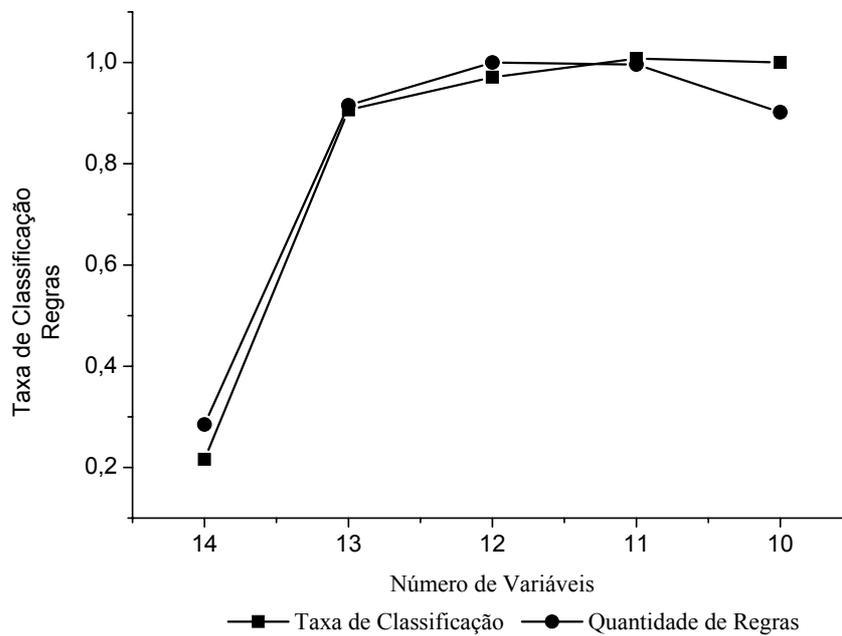


Figura E.4: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 5 (Boston Housing).

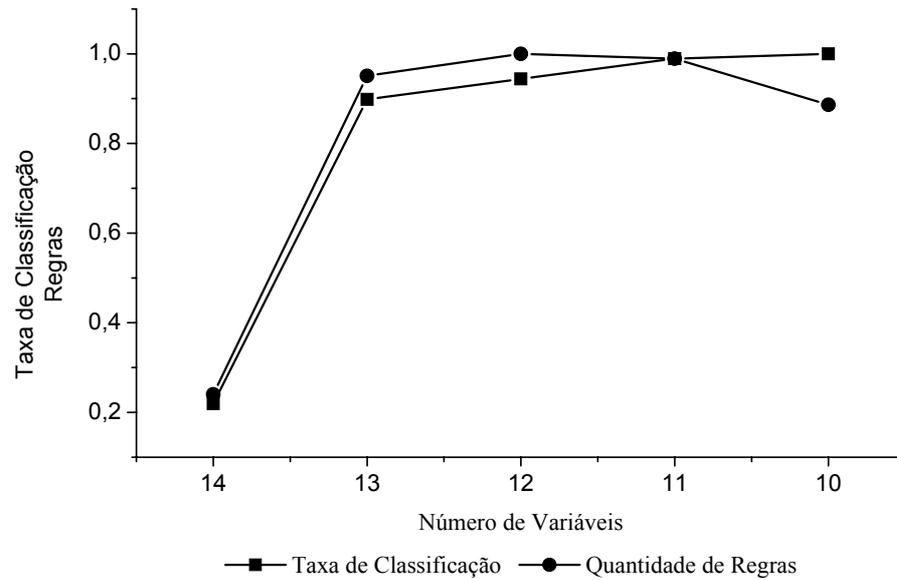


Figura E.5: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 5 (Boston Housing).

Tabela E.3: Resultados para a construção da BD com granularidade 7 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 26 | 126,2 | 133 | 137,6 | 155 |
| Regras | 20,6 | 91,6 | 89,4 | 94,4 | 96,8 |
| Desvio Padrão | 1,4142 | 4,0865 | 2,9154 | 3,7815 | 3,5355 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

Tabela E.4: Resultados para a otimização da BD com granularidade 7 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 30,6 | 140,6 | 147,6 | 158 | 175,6 |
| Regras | 18 | 79,6 | 87,2 | 82,6 | 89,8 |
| Desvio Padrão | 4,3931 | 7,3348 | 5,5497 | 3,3911 | 7,6354 |
| Granularidade | 7 | 7 | 7 | 7 | 7 |

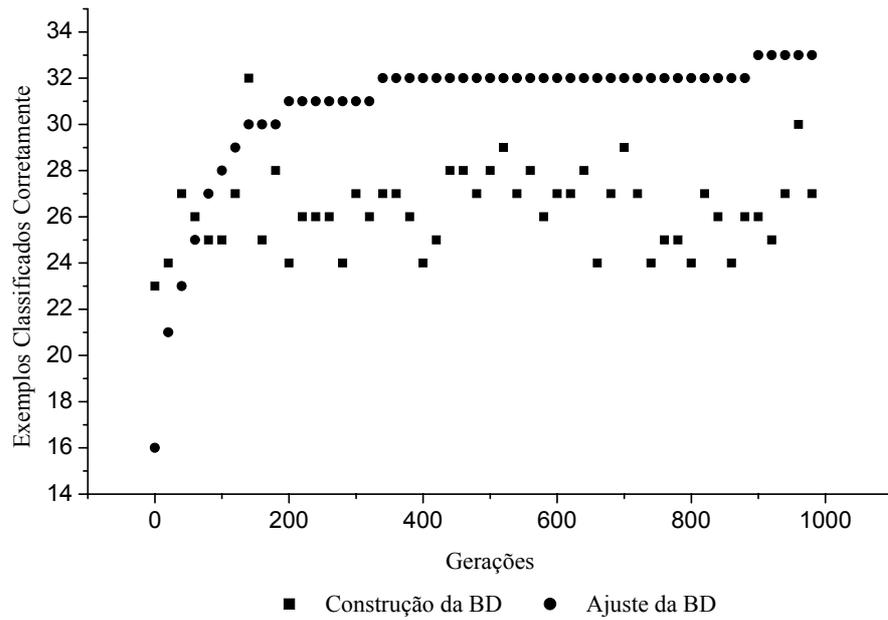


Figura E.6: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 7 (Boston Housing).

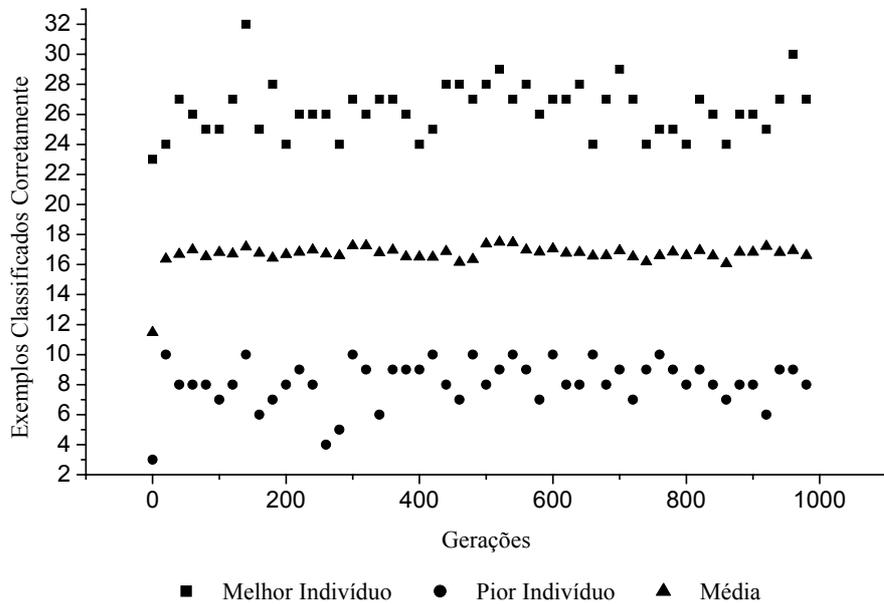


Figura E.7: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 7 (Boston Housing).

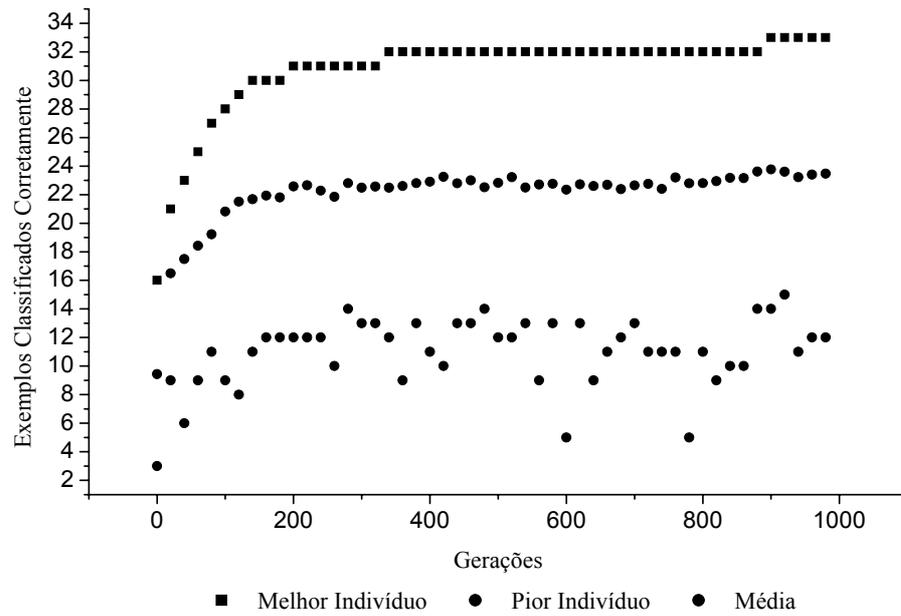


Figura E.8: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 7 (Boston Housing).

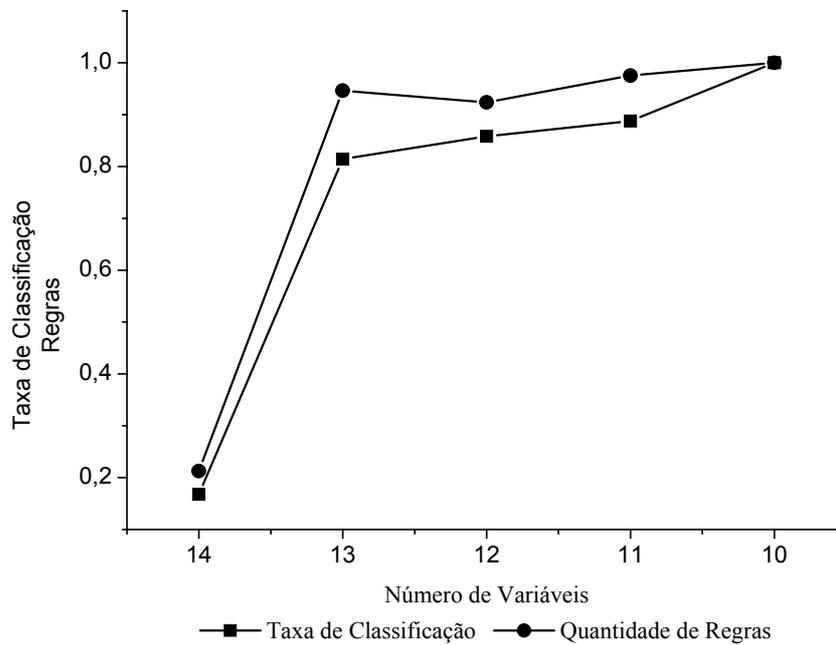


Figura E.9: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 7 (Boston Housing).

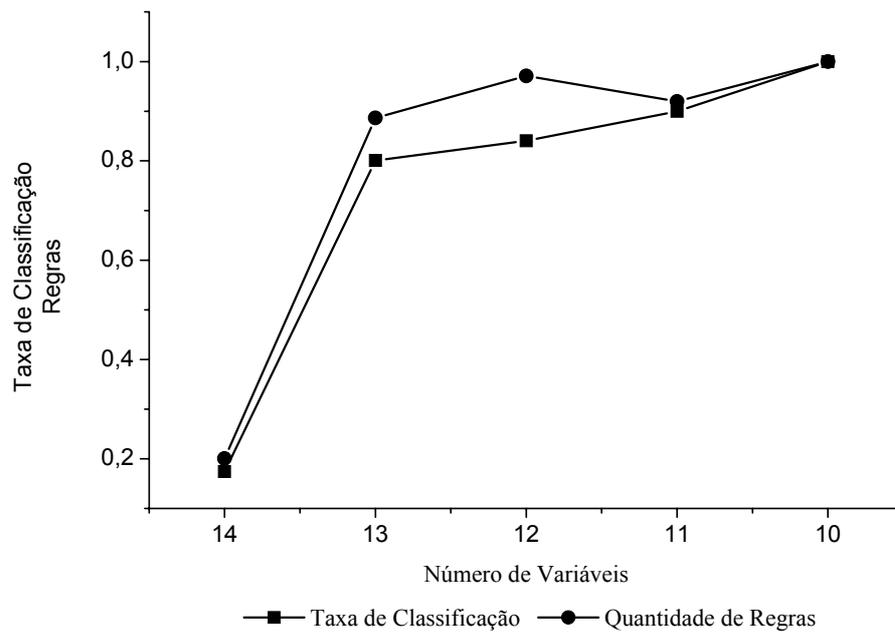


Figura E.10: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 7 (Boston Housing).

Tabela E.5: Resultados para a construção da BD com granularidade 9 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|--------|
| ECC | 16,6 | 85,8 | 89 | 95,6 | 126,6 |
| Regras | 13,6 | 65,4 | 67,2 | 70,4 | 87,6 |
| Desvio Padrão | 0,5477 | 1,9235 | 8,2462 | 7,4699 | 5,6833 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

Tabela E.6: Resultados para a otimização da BD com granularidade 9 (Boston Housing).

| Variáveis | 14 | 13 | 12 | 11 | 10 |
|---------------|--------|--------|--------|--------|---------|
| ECC | 21,6 | 107,6 | 115,8 | 122,6 | 151,6 |
| Regras | 13,4 | 59,2 | 67,8 | 72 | 84,4 |
| Desvio Padrão | 3,5071 | 4,0987 | 6,7970 | 5,9833 | 10,5735 |
| Granularidade | 9 | 9 | 9 | 9 | 9 |

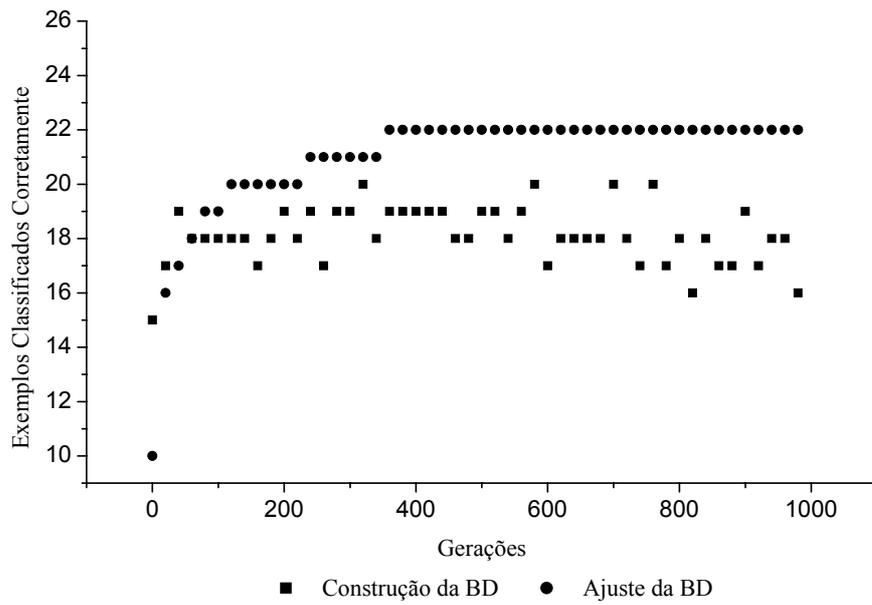


Figura E.11: Comparação entre as abordagens do desempenho do AG com 14 variáveis e granularidade 9 (Boston Housing).

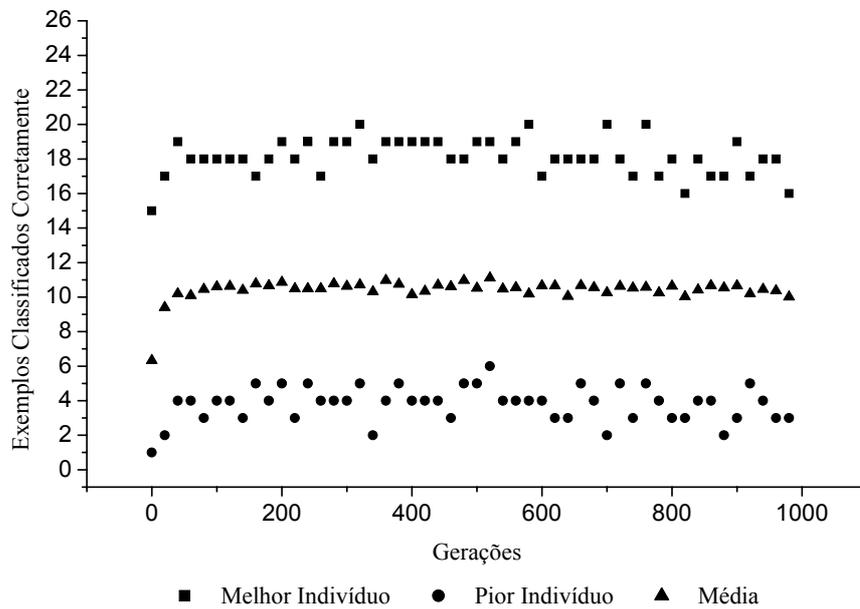


Figura E.12: Desempenho dos indivíduos na construção da BD com 14 variáveis e granularidade 9 (Boston Housing).

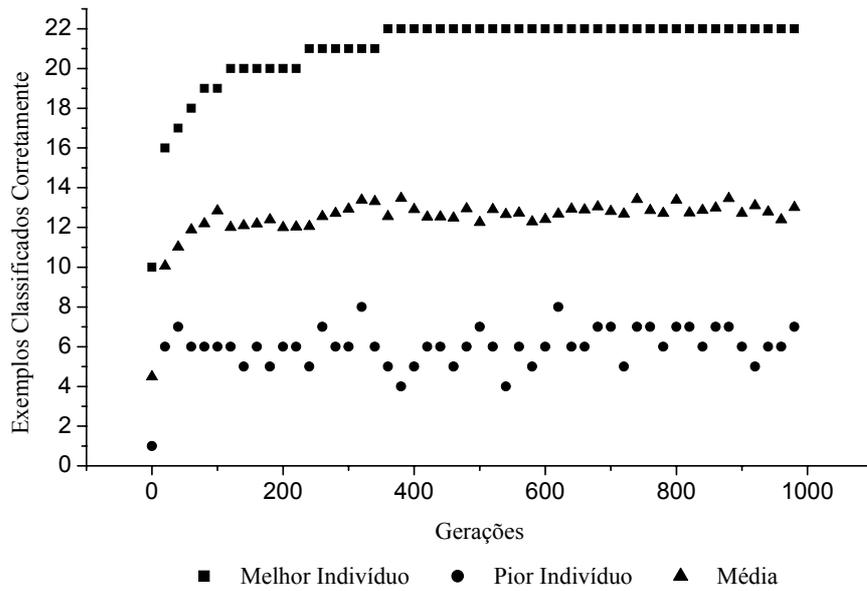


Figura E.13: Desempenho dos indivíduos na otimização da BD com 14 variáveis e granularidade 9 (Boston Housing).

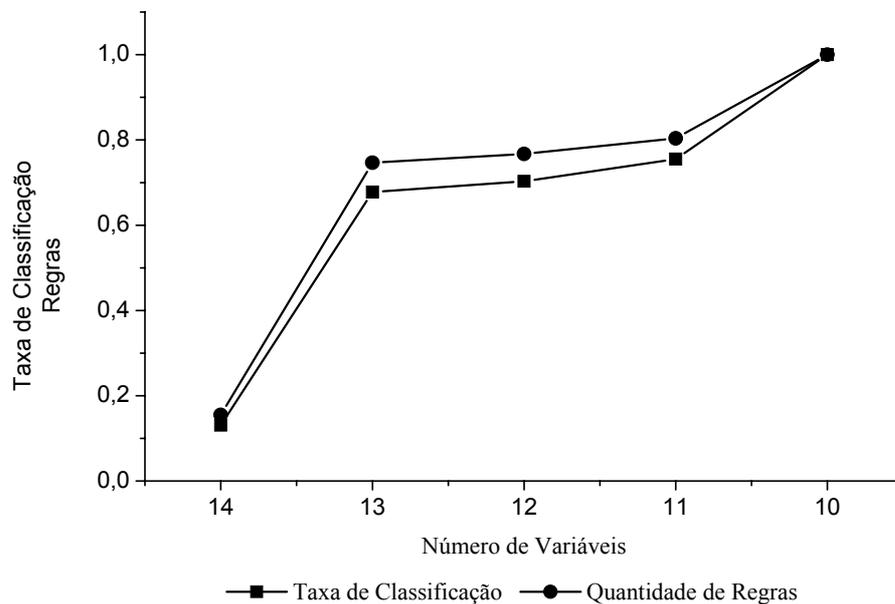


Figura E.14: Relação taxa de classificação vs. quantidade de regras na construção da BD com granularidade 9 (Boston Housing).

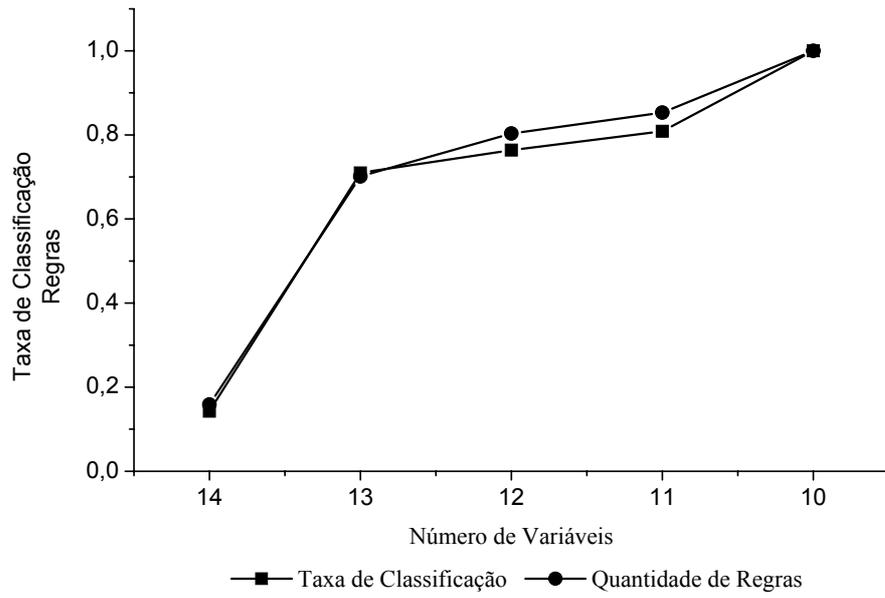


Figura E.15: Relação taxa de classificação vs. quantidade de regras na otimização da BD com granularidade 9 (Boston Housing).

Referências Bibliográficas

- [Bäck et al. 1991] Bäck, T.; Hoffmeister, F.; Schwefel, H.-P., "*A Survey of Evolution Strategies*", **Proceedings of the Fourth International Conference on Genetic Algorithms**, In R. K. Belew and L. B. Booker, eds., Morgan Kaufmann, 1991.
- [Berg et al. 2002] Berg, J. V. D.; Kaymak, U.; Bergh, W. M. V. D., "*Fuzzy Classification using Probability-based Rule Weighting*", **Proceedings of the 11th IEEE International Conference on Fuzzy Systems**, Hawaii - USA, 2002.
- [Bittencourt 1998] Bittencourt, G., **Inteligência Artificial: Ferramentas e Teorias**, Editora da UFSC, 1998.
- [Blake e Merz 1998] Blake, C. L.; Merz, C. J., **UCI Repository of machine learning databases**, Irvine, CA. University of California, Department of Information and Computer Science, <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 1998.
- [Bonarini 1996] Bonarini, A., "*Evolutionary Learning of Fuzzy Rules: Competition and Cooperation*", In **W. Pedrycz, editor, Fuzzy Modelling: Paradigms and Practice**, pp.265-284, Kluwer Academic Press, Norwell, MA, 1996.
- [Bonissone et al. 1999] Bonissone, P.; Chen, Y.-T.; Goebel, K.; Khedkar, P. S., "*Hybrid Soft Computing Systems: Industrial and Commercial Applications*", **Proceedings of the IEEE**, vol.87, no.9, pp.1641-1667, 1999.
- [Bonissone et al. 1996] Bonissone, P.; Khedkar, P. S.; Chen, Y.-T., "*Genetic Algorithms for Automated Tuning of Fuzzy Controllers, a Transportation Application*", **Proceedings of the 5th International Conference Fuzzy Systems (FUZZ-IEEE'96)**, pp.674-680, 1996.
- [Bramlette 1991] Bramlette, M. F., "*Initialization, Mutation and Selection Methods in Genetic Algorithms for Functions Optimization*", **Proceedings of the Fourth International Conference on Genetic Algorithms**, In R. K. Belew and L. B. Booker, eds., Morgan Kaufmann, 1991.

- [Carse et al. 1996] Carse, B.; Forgarty, T. C.; Munro, A., “*A Envolving Fuzzy Rule Based Controllers using Genetic Algorithms*”, **Fuzzy Sets and Fuzzy Systems**, vol.80, pp.273-293, 1996.
- [Casillas et al. 2000] Casillas, J.; Cordon, O.; Herrera, F., “*A Methodology to Improve ad hoc Data-Driven Linguistic Rule Learning Methods by Inducing Cooperation Among Rules*”, **Technical Report #DECSAI-000 101**, University of Granada - Spain, 2000.
- [Castro et al. 2003] Castro, P. A. D.; Pires, M. G.; Camargo, H. A., “*Aprendizado e Seleção de Regras Fuzzy usando Algoritmos Genéticos*”, **VI Simpósio Brasileiro de Automação Inteligente - SBAI**, Bauru - SP, 2003.
- [Cordon et al. 2004] Cordon, O.; Gomide, F.; Herrera, F.; Hoffmann, F.; Magdalena, L., “*Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends*”, **Fuzzy Sets and Fuzzy Systems**, vol.141, pp.5-31, 2004.
- [Cordon e Herrera 1997] Cordon, O.; Herrera, F., “*A Three-stage Evolutionary Process for Learning Descriptive and Approximate Fuzzy Logic Controller Knowledge Bases from Examples*”, **International Journal of Approximate Reasoning**, vol.7, no.4, pp.369-407, 1997.
- [Cordon e Herrera 2000] Cordon, O.; Herrera, F., “*A Proposal for Improving the Accuracy of Linguistic Modeling*”, **IEEE Transactions on Fuzzy Systems**, vol.8, pp.335-344, 2000.
- [Cordon e Herrera 2001] Cordon, O.; Herrera, F., “*Hybridizing Genetic Algorithms with Sharing Scheme and Evolution Strategies for Designing Approximate Fuzzy Rule-Based Systems*”, **Fuzzy Sets and Fuzzy Systems**, vol.118, no.2, pp.235-255, 2001.
- [Cordon et al. 2001a] Cordon, O.; Herrera, F.; Hoffmann, F.; Gomide, F.; Magdalena, L., “*Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends*”, **Proceedings of the 9th IFSA World Congress**, pp.1241-1246, Vancouver, Canada, July 25-28, 2001.
- [Cordon et al. 2001b] Cordon, O.; Herrera, F.; Hoffmann, F.; Magdalena, L., “*Recent Advances in Genetic Fuzzy Systems*”, **Journal of Information Sciences**, vol.136, pp.1-5, 2001.
- [Cordon et al. 2000] Cordon, O.; Herrera, F.; Villar, P., “*Analysis and Guidelines to Obtain a Good Uniform Fuzzy Partition Granularity for Fuzzy Rule-Based System Using Simulated Annealing*”, **International Journal of Approximate Reasoning**, vol.25, no.3, pp.187-215, 2000.
- [Cordon et al. 2001c] Cordon, O.; Herrera, F.; Villar, P., “*Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base*”, **IEEE Transactions on Fuzzy Systems**, vol.9, no.4, pp.667-674, 2001.

- [Cordón et al. 2001d] Cordón, O.; Jesus, M. J.; Herrera, F., “*A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems*”, **International Journal of Approximate Reasoning**, vol.20, pp.21-45, 2001.
- [Delgado 2002] Delgado, M. R., **Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva**, Tese de Doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2002.
- [Filipic e Juricic 1996] Filipic, B.; Juricic, D., “*A Genetic Algorithm to Support Learning Fuzzy Control Rules from Examples*”, **Genetic Algorithms and Soft Computing**, pp.403-418, F. Herrera and J.L. Verdegay, Eds: Physica-Verlag, 1996.
- [Fogel et al. 1966] Fogel, L. J.; Owens, A. J.; Walsh, M. J., **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.
- [Goldberg 1989] Goldberg, D. E., **Genetic Algorithms in Search, Optimization and Machine Learning**, Addison-Wesley, 1989.
- [González e Pérez 1999] González, A.; Pérez, R., “*SLAVE: A Genetic Learning System Based on an Iterative Approach*”, **IEEE Transaction on Fuzzy Systems**, vol.7, pp.176-191, April 1999.
- [Grefenstette 1986] Grefenstette, J. J., “*Optimization of Control Parameters for Genetic Algorithms*”, **IEEE Transaction on Systems, Man and Cybernetics**, vol.16, no.1, pp.122-128, 1986.
- [Haykin 1994] Haykin, S., **Neural Networks - A Comprehensive Foundation**, Prentice-Hall, 1994.
- [Herrera et al. 1995] Herrera, F.; Lozano, M.; Verdegay, J. L., “*Tuning Fuzzy Logic Controllers by Genetic Algorithms*”, **International Journal of Approximate Reasoning**, vol.12, pp.299-315, 1995.
- [Hoffmann e Pfister 1997] Hoffmann, F.; Pfister, G., “*Evolutionary Design of a Fuzzy Knowledge Base for a Mobile Robot*”, **International Journal of Approximate Reasoning**, vol.17, no.4, pp.447-469, 1997.
- [Holland 1975] Holland, J. H., **Adaptation in Natural and Artificial Systems**, Ann Arbor: The University of Michigan Press, 1975.
- [Hwang e Thompson 1994] Hwang, W. R.; Thompson, W. E., “*Design of Fuzzy Logic Controllers Using Genetic Algorithms*”, **Proceedings of the 3th International Conference Fuzzy Systems (FUZZ-IEEE'94)**, pp.1383-1388, Orlando - FL, 1994.
- [Ishibuchi e Murata 1996] Ishibuchi, H.; Murata, T., “*A Genetic-Algorithm-Based Fuzzy Partition Method for Pattern Classification Problems*”, **Genetic Algorithms and Soft Computing**, pp.555-578, Berlin, Germany: Physica-Verlag, 1996.

- [Ishibuchi et al. 1997] Ishibuchi, H.; Murata, T.; Türksen, I. B., “*Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems*”, **Fuzzy Sets and Fuzzy Systems**, vol.89, pp.135-150, 1997.
- [Ishibuchi et al. 1999a] Ishibuchi, H.; Nakashima, T.; Morisawa, T., “*Voting in Fuzzy Rule-Based Systems for Pattern Classification Problems*”, **Fuzzy Sets and Fuzzy Systems**, vol.103, pp.223-238, 1999.
- [Ishibuchi et al. 1999b] Ishibuchi, H.; Nakashima, T.; Murata, T., “*Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems*”, **IEEE Transaction on Systems, Man and Cybernetics**, vol.29, no.5, pp.601-618, 1999.
- [Ishibuchi et al. 1994] Ishibuchi, H.; Nozaki, K.; Tanaka, H.; Hosaka, Y.; Matsuda, M., “*Empirical Study on Learning in Fuzzy Systems by Rice Analysis*”, **Fuzzy Sets and Fuzzy Systems**, vol.64, pp.129-144, 1994.
- [Ishibuchi et al. 1995] Ishibuchi, H.; Nozaki, K.; Yamamoto, N.; Tanaka, H., “*Selecting Fuzzy If-Then Rules for Classification Problems using Genetic Algorithms*”, **IEEE Transaction on Fuzzy Systems**, vol.3, no.3, pp.260-270, 1995.
- [Kandel 1986] Kandel, A., **Fuzzy Mathematical Techniques with Applications**, MA: Addison-Wesley, 1986.
- [Karr 1991] Karr, C. L., “*Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm*”, **Proceedings of the 4th International Conference Genetic Algorithms**, pp.450-457, University of California, San Diego, July 13-16, 1991.
- [Kbir et al. 2000] Kbir, M. A.; Benkirane, H.; Maalmi, K.; Benslimane, R., “*Hierarchical Fuzzy Partition for Pattern Classification with Fuzzy If-Then Rules*”, **Pattern Recognition Letters**, vol.21, pp.503-509, 2000.
- [Klir e Folger 1988] Klir, G.; Folger, T. A., **Fuzzy Sets, Uncertainty and Information**, Prentice Hall, 1988.
- [Klir e Yuan 1995] Klir, G.; Yuan, B., **Fuzzy Sets and Fuzzy Logic: Theory and Applications**, Prentice-Hall, 1995.
- [Lacerda e Carvalho 1980] Lacerda, E. G. M.; Carvalho, A. C. P. L. F., “*Introdução aos Algoritmos Genéticos*”, **Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação**, vol.2, pp.51-126, 1980.
- [Lee e Takagi 1993] Lee, M. A.; Takagi, H., “*Embedding Apriori Knowledge into an Integrated Fuzzy System Design Method Based on Genetic Algorithm*”, **Proceedings of the 5th IFSA World Congress (IFSA'93)**, pp.1293-1296, Seoul, Korea, 1993.

- [Liao et al. 1997] Liao, T. W.; Celmins, A. K.; Hammell, R. J., “*A Fuzzy C-Means Variant for the Generation of Fuzzy Term Sets*”, **Fuzzy Sets and Fuzzy Systems**, vol.135, pp.241-257, 1997.
- [Luger e Stubblefield 1989] Luger, G. F.; Stubblefield, W. A., **Artificial Intelligence and the Design of Expert Systems**, The Benjamin Cummings Publishing Company, 1989.
- [Lukasiewicz 1970] Lukasiewicz, J., “*Philosophical Remarks on Many-Valued Systems of Propositional Logic*”, **In Studies in Logic and the Foundations of Mathematics**, North Holland, 1970.
- [Magdalena e Monasterio 1997] Magdalena, L.; Monasterio, F., “*A Fuzzy Logic Controller with Learning Through the Evolution of its Knowledge Base*”, **International Journal of Approximate Reasoning**, vol.16, pp.335-358, 1997.
- [Mamdani e Assilian 1975] Mamdani, E. H.; Assilian, S., “*An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*”, **International Journal of Man-Machine Studies**, vol.7, no.1, pp.1–13, 1975.
- [Mendel 1995] Mendel, J. M., “*Fuzzy Logic Systems for Engineering: A Tutorial*”, **IEEE Special Issue on Fuzzy Logic**, vol.83, no.3, pp.345-377, 1995.
- [Michalewicz 1996] Michalewicz, Z., **Genetic Algorithms + Data Structures = Evolution Programs**, Springer, 1996.
- [Mitchell 1996] Mitchell, M., **An Introduction to Genetic Algorithms**, The MIT Press, 1996.
- [Mizumoto e Zimmermann 1982] Mizumoto, M.; Zimmermann, H.-J., “*Comparison of Fuzzy Reasoning Methods*”, **FSS**, vol.8, pp.253-283, 1982.
- [Nauck e Cruse 1997] Nauck, D.; Cruse, R., “*A Neuro-Fuzzy Method to Learn Fuzzy Classification Rules from Data*”, **Fuzzy Sets and Systems**, vol.89, pp.277-288, 1997.
- [Nomura et al. 1992] Nomura, H.; Hayashi, L.; Wakami, N., “*A Learning Method of Fuzzy Inference Rules by Descent Method*”, **Proceedings of the 1st IEEE International Conference on Fuzzy Systems**, pp.203-210, San Diego - USA, 1992.
- [Parodi e Bonelli 1993] Parodi, A.; Bonelli, P., “*A New Approach to Fuzzy Classifier Systems*”, **Proceedings of the 5th International Conference of Genetic Algorithms (ICGA'93)**, pp.223-230, 1993.
- [Pedrycz 1998] Pedrycz, W., **Computational Intelligence - An Introduction**, CRC Press, 1998.

- [Pedrycz e Gomide 1998] Pedrycz, W.; Gomide, F., **An Introduction to Fuzzy Sets: Analysis and Design**, The MIT Press, 1998.
- [Rechenberg 1965] Rechenberg, I., **Cybernetic Solution Path of an Experimental Problem**, (Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.). Farnborough Hants: Ministry of Aviation, Royal Aircraft Establishment, 1965.
- [Rechenberg 1973] Rechenberg, I., **Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution**, Frommann-Holzboog (Stuttgart), 1973.
- [Rescher 1969] Rescher, N., **Many-Valued Logic**, Mc Graw Hill, 1969.
- [Rezende 2003] Rezende, S. O., **Sistemas Inteligentes - Fundamentos e Aplicações**, Manoele, 2003.
- [Schwefel 1975] Schwefel, H.-P., **Evolutionsstrategie und Numerische Optimierung**, Ph.D. thesis, Technische Universität Berlin, 1975.
- [Schwefel 1977] Schwefel, H.-P., **Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie**, Basel: Birkhäuser, 1977.
- [Takagi e Sugeno 1983] Takagi, T.; Sugeno, M., "*Derivation of Fuzzy Control Rules from Human Operator's Control Actions*", **Proceedings of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis**, pp.55–60, Marseilles, France, 1983.
- [Tsukamoto 1979] Tsukamoto, Y., "*An Approach to Fuzzy reasoning Methods*", **Advances in Fuzzy Set Theory**, pp.137-149, In Ragade, R. & Yager, R. editors, Amsterdam - North-Holland, 1979.
- [Velasco 1998] Velasco, J. R., "*Genetic-Based On-line Learning for Fuzzy Process Control*", **International Journal of Intelligence Systems**, vol.13, no.10-11, pp.891-903, 1998.
- [Wang e Mendel 1992] Wang, L. X.; Mendel, J. M., "*Generating Fuzzy Rules by Learning from Examples*", **IEEE Transaction on Systems, Man and Cybernetics**, vol.22, pp.1414-1427, 1992.
- [Yuan e Zhuang 1996] Yuan, Y.; Zhuang, H., "*A Genetic Algorithm for Generating Fuzzy Classification Rules*", **Fuzzy Sets and Fuzzy Systems**, pp.1-19, 1996.
- [Zadeh 1965] Zadeh, L. A., "*Fuzzy Sets*", **Information and Control**, vol.8, pp.338-353, 1965.

- [Zadeh 1975] Zadeh, L. A., “*The Concept of Linguistic Variable and its Application to Approximate Reasoning*”, **Information Sciences**, vol.8, pp.199-249 (part I), 301-57 (part II), 1975.
- [Zadeh 1997] Zadeh, L. A., “*What is Soft Computing?*”, **Soft Computing**, vol.1, no.1, pp.1, 1997.
- [Zadeh 1998] Zadeh, L. A., “*Some Reflections on Soft Computing, Granular Computing and Their Roles in the Conception, Design and Utilization of Information/Intelligent Systems*”, **Soft Computing**, vol.2, pp.23-25, Springer-Verlag, 1998.
- [Zimmermann 1990] Zimmermann, H.-J., **Fuzzy Set Theory and its Applications**, 2ª edição, Kluwer Academic Publishers, 1990.