Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Geração, Seleção e Combinação de Componentes para Ensembles de Redes Neurais Aplicadas a Problemas de Classificação

Autor: Guilherme Palermo Coelho
Orientador: Prof. Dr. Fernando José Von Zuben

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora

Fernando José Von Zuben, Dr. . . . . DCA/FEEC/Unicamp
Anne Magaly de Paula Canuto, Dra. . . . . DIMAp/CCET/UFRN
Romis Ribeiro de Faissol Attux, Dr. . . . . DECOM/FEEC/Unicamp
Clodoaldo Aparecido de Moraes Lima, Dr. . . . . DCA/FEEC/Unicamp

Campinas, SP
Setembro/2006

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

C65g

Coelho, Guilherme Palermo
Geração, seleção e combinação de componentes para ensembles de redes neurais aplicadas a problemas de classificação / Guilherme Palermo Coelho. --Campinas, SP: [s.n.], 2006.

Orientador: Fernando José Von Zuben
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais (Computação). 2. Sistema imune. 3. Inteligência artificial. I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Generation, selection and combination of components in neural network ensembles applied to classification problems
Palavras-chave em Inglês: Pattern classifiers, Artificial neural networks, Artificial immune systems, Ensembles, Component selection, Component combination
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Anne Magaly de Paula Canuto, Romis Ribeiro de Faissol Attux, Clodoaldo Aparecido de Moraes Lima
Data da defesa: 29/09/2006
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – DISSERTAÇÃO DE MESTRADO

Candidato: Guilherme Palermo Coelho

Data da Defesa: 29 de setembro de 2006

Título da Tese: "Geração, Seleção e Combinação de Componentes para Ensembles de Redes Neurais Aplicadas a Problemas de Classificação"

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

Prof. Dr. Anne Magaly de Paula Canuto: _____

Prof. Dr. Romis Ribeiro de Faissol Attux: _____

Prof. Dr. Clodoaldo Aparecido de Moraes Lima: _____

Resumo

O uso da abordagem *ensembles* tem sido bastante explorado na última década, por se tratar de uma técnica simples e capaz de aumentar a capacidade de generalização de soluções baseadas em aprendizado de máquina. No entanto, para que um *ensemble* seja capaz de promover melhorias de desempenho, os seus componentes devem apresentar bons desempenhos individuais e, ao mesmo tempo, devem ter comportamentos diversos entre si. Neste trabalho, é proposta uma metodologia de criação de *ensembles* para problemas de classificação, onde os componentes são redes neurais artificiais do tipo perceptron multicamadas. Para que fossem gerados bons candidatos a comporem o *ensemble*, atendendo a critérios de desempenho e de diversidade, foi aplicada uma meta-heurística populacional imuno-inspirada, denominada *opt-aiNet*, a qual é caracterizada por definir automaticamente o número de indivíduos na população a cada iteração, promover diversidade e preservar ótimos locais ao longo da busca. Na etapa de seleção dos componentes que efetivamente irão compor o *ensemble*, foram utilizadas seis técnicas distintas e, para combinação dos componentes selecionados, foram adotadas cinco estratégias. A abordagem proposta foi aplicada a quatro problemas de classificação de padrões e os resultados obtidos indicam a validade da metodologia de criação de *ensembles*. Além disso, foi verificada uma dependência entre o melhor par de técnicas de seleção e combinação e a população de indivíduos candidatos a comporem o *ensemble*, assim como foi feita uma análise de confiabilidade dos resultados de classificação.

Palavras-chave: Classificadores de padrões, redes neurais artificiais, sistemas imunológicos artificiais, *ensembles*, seleção de componentes, combinação de componentes.

Abstract

In the last decade, the *ensemble* approach has been widely explored, once it is a simple technique capable of increasing the generalization capability of machine learning based solutions. However, an *ensemble* can only promote performance enhancement if its components present good individual performance and, at the same time, diverse behavior among each other. This work proposes a methodology to synthesize *ensembles* for classification problems, where the components of the *ensembles* are multi-layer perceptrons. To generate good candidates to compose the *ensemble*, meeting the performance and diversity requirements, it was applied a populational and immune-inspired metaheuristic, named *opt-aiNet*, which is characterized as being capable of automatically determining the number of individuals in the population at each iteration, promoting diversity and preserving local optima through the search. In the component selection phase, six distinct techniques were applied and, to combine these selected components, five strategies were adopted. The proposed approach was applied to four pattern classification problems and the obtained results indicated the validity of the methodology to synthesize *ensembles*. It was also verified a dependence of the best pair of selection and combination techniques on the population of candidates to compose the *ensemble*, and it was made an analysis of the confidence of the classification results.

Keywords: Pattern classifiers, artificial neural networks, artificial immune systems, *ensembles*, component selection, component combination.

Aos meus pais Joacir e Stella e à minha irmã Luciana

Agradecimentos

Ao Professor Fernando José Von Zuben, pela orientação neste trabalho, amizade e confiança em mim depositada. Seu senso crítico, o constante estímulo e as valiosas sugestões dadas em nossas reuniões foram essenciais para o meu crescimento como pesquisador e para a conseqüente elaboração deste trabalho.

Aos meus pais Joacir e Stella, pelo constante incentivo e apoio incondicional, não só durante o mestrado, mas em toda a minha vida.

À minha irmã Luciana, pelo estímulo e pelas valiosas sugestões em vários detalhes desta dissertação.

A todos os amigos do Laboratório de Bioinformática e Computação Bio-Inspirada (LBiC), pelo companheirismo e pelas várias discussões, que contribuíram enormemente para o desenvolvimento deste trabalho.

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

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

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Siglas

MLP	-	*Multi-layer Perceptron*
RBF	-	Rede Neural do tipo *Radial Basis Function*
SI	-	Sistema Imunológico Natural
SIA	-	Sistemas Imunológicos Artificiais
RNA	-	Redes Neurais Artificiais
$\sum$	-	Combinação de componentes em um *ensemble*
ECOC	-	*Error-Correcting Output Coding*
CNNE	-	*Cooperative Neural Network Ensemble*
MST	-	*Minimum Spanning Tree*
Cw/oE	-	Construtiva sem Exploração
CwE	-	Construtiva com Exploração
Pw/oE	-	Poda sem Exploração
PwE	-	Poda com Exploração
ARIA	-	Algoritmo de Agrupamento ARIA (*Adaptive Radius Immune Algorithm*)
SS	-	Sem Seleção
MS	-	Média Simples
MP	-	Média Ponderada sem *Bias*
MPB	-	Média Ponderada com *Bias*
VM	-	Voto Majoritário
WTA	-	*Winner-takes-all*

Trabalhos Publicados Pelo Autor

1. P. A. D. de Castro, G. P. Coelho, M. F. Caetano, F. J. Von Zuben. "Designing Ensembles of Fuzzy Classification Systems: An Immune-Inspired Approach". *Fourth International Conference on Artificial Immune Systems* (Icaris), Banff, Alberta, Canada. *Lecture Notes on Computer Science* (LNCS), vol. 3627, pp. 469–482, August 2005.

2. G. P. Coelho, P. A. D. de Castro, F. J. Von Zuben. "The Effective Use of Diverse Rule Bases in Fuzzy Classification". *VII Congresso Brasileiro de Redes Neurais* (CBRN), Natal, Rio Grande do Norte, Brasil, Outubro 2005.

3. G. P. Coelho, F. J. Von Zuben. "The Influence Of The Pool of Candidates on the Performance of Selection and Combination Techniques in Ensembles". *IEEE International Joint Conference on Neural Networks* (IJCNN), Vancouver, British Columbia, Canada, pp. 10588–10595, July 2006.

4. G. P. Coelho, F. J. Von Zuben. "omni-aiNet: An Immune-inspired Approach for Omni Optimization". *Fifth International Conference on Artificial Immune Systems* (Icaris), Oeiras, Portugal. *Lecture Notes on Computer Science* (LNCS), vol. 4163, pp. 294-308, September 2006.

Capítulo 1

Introdução

Quando decisões importantes devem ser tomadas sobre assuntos de grande impacto em nossas vidas, geralmente recorremos a opiniões e considerações de um grupo de pessoas que dominem o assunto em questão, visando maximizar a chance de se tomar uma decisão correta. Essa tendência a organizar comitês para tomar decisões pode ser observada em diversos níveis na sociedade humana, desde em pequenas reuniões familiares para decidir qual o melhor colégio para os filhos, até nas sessões do Congresso Nacional para discutir e votar algum Projeto de Lei.

O objetivo principal por trás da formação desses grupos (também chamados *comitês*) está em reunir pessoas que tenham um certo domínio do assunto em questão, mas que, ao mesmo tempo, tenham opiniões diversas, de forma a levantar discussões que contribuam para a identificação dos principais pontos positivos e negativos que possam estar associados a cada uma das opções de decisão, e com isso levar à escolha da melhor solução para o problema em questão.

No campo da inteligência computacional, mais especificamente em aprendizado de máquina, a idéia de formação de comitês de indivíduos que tenham um bom conhecimento sobre um problema e ao mesmo tempo tenham "opiniões", em certo grau, distintas dos demais indivíduos no comitê foi adotada nos chamados *ensembles* (Hansen & Salamon, 1990).

Um outro ponto com que freqüentemente nos deparamos em nosso cotidiano, mesmo sem que percebamos às vezes, é que constantemente temos a necessidade de *classificar* coisas. Por exemplo, tomemos uma ida ao supermercado para comprar maçãs. Ao chegar lá, a maioria das pessoas não compra as primeiras maçãs que estejam ao seu alcance (apesar de alguns adotarem esta estratégia). Normalmente, analisamos a aparência da fruta, sua rigidez, sua cor, preço e outras características, de forma a levarmos as melhores maçãs, dentre as disponíveis, para casa. Apesar de "melhores maçãs" ser um critério subjetivo, na verdade o que fazemos no supermercado é *classificar* as frutas em duas classes: aquelas que estão e aquelas que não estão em um estado que satisfaça nossas necessidades.

Dessa forma, podemos definir como *problemas de classificação* aqueles em que, a partir de um

conjunto de atributos de um dado objeto de estudo (no caso da maçã: aparência, rigidez, cor e preço), deve-se definir um grupo ao qual este objeto pertença (no caso da maçã: os grupos de maçãs que compensam e que não compensam ser adquiridas). A cada um desses grupos dá-se o nome de *classe* e ao dispositivo responsável pelo processo de classificação dá-se o nome de *classificador* (que corresponde ao comprador no supermercado, no caso das maçãs). Os problemas de classificação estão presentes em diversas áreas de conhecimento, seja por exemplo na determinação do diagnóstico médico de acordo com os resultados de exames de um paciente (se ele está ou não doente, ou qual o tipo de uma dada doença), seja na determinação de um tipo de vinho de acordo com suas características.

Em aprendizado de máquina, existem inúmeras propostas para tratamento desses problemas de classificação, que se baseiam tanto no uso individual dos mais diversos tipos de classificadores (como *redes neurais artificiais*, *classificadores baseados em regras*, *árvores de decisão* etc.) quanto na combinação destes classificadores, como é feito nos *ensembles*.

Neste trabalho, serão então estudados *ensembles* de redes neurais artificiais do tipo *perceptron multi-camadas*, ou *multi-layer perceptron* (Haykin, 1999), para aplicações em problemas de classificação de padrões.

1.1 Motivação

A abordagem de *ensembles* tem sido bastante explorada na última década, por se tratar de uma técnica simples e capaz de aumentar a capacidade de generalização de soluções baseadas em aprendizado de máquina. Para ilustrar esta característica de aumento da capacidade de generalização e conseqüente melhora do desempenho geral do sistema, tomemos o problema de aproximação de funções representado na Figura 1.1 e o problema de classificação de padrões representado na Figura 1.2.

Para o problema pictórico de regressão da Figura 1.1, desejamos aproximar a função dada pela curva tracejada em rosa a partir de três amostras de treinamento, dadas pelos pontos em verde (em uma situação real, seria necessário um número bem maior de amostras de treinamento). A partir destas três amostras, foram obtidos dois regressores, A e B, representados pelas curvas em vermelho e em azul, respectivamente. Como podemos perceber, os dois regressores obtidos são capazes de aproximar com sucesso o valor da função desejada para as amostras utilizadas no treinamento. No entanto, para outros valores de x não vistos na etapa de treinamento, como por exemplo $x = 0.6$, os dois regressores apresentam erros consideráveis, ou seja, sua capacidade de generalização para dados não vistos previamente é baixa.

Suponhamos agora que estes mesmos dois regressores A e B sejam utilizados como componentes de um *ensemble* cuja saída (decisão final do comitê) é dada pela média simples das saídas de

seus componentes individuais. A saída deste *ensemble* está representada na Figura 1.1 pela curva em preto. Como podemos observar, a curva gerada pelo *ensemble* se aproxima bem mais da curva desejada para todos os pontos no intervalo considerado, e não mais apenas para as amostras usadas no treinamento, como nos regressores individuais. Ou seja, o *ensemble* obtido possui uma capacidade de generalização bem maior que a dos regressores individuais, já que ele é capaz de aproximar regiões da função desejada não vistas no treinamento com um erro bem menor que os apresentados pelos seus componentes individuais (vide novamente o ponto $x = 0.6$, representado pela linha pontilhada).

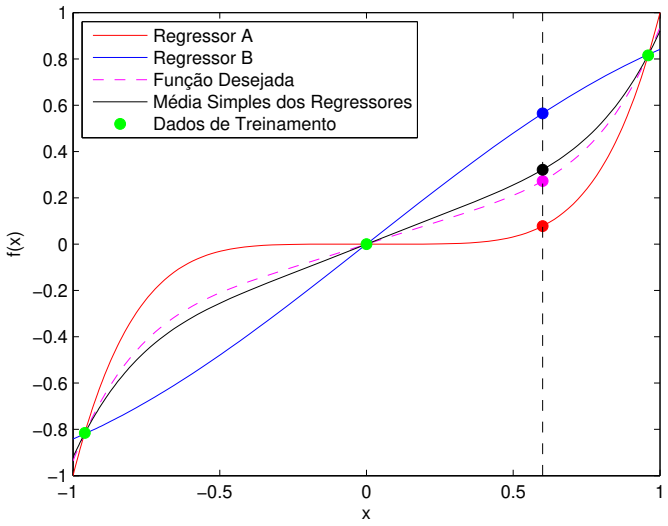


Fig. 1.1: Exemplo pictórico da capacidade de aumento de generalização proporcionada pelo uso de *ensembles*. Aqui deseja-se aproximar a função dada pela curva tracejada em rosa, a partir dos dados de treinamento representados em verde. Foram obtidos dois regressores (curvas em vermelho e azul) que são capazes de aproximar corretamente os dados de treinamento, mas seu desempenho não é tão bom para as demais regiões do problema, ou seja, eles possuem baixa capacidade de generalização (vide ponto $x = 0.6$, por exemplo). Estes dois regressores foram combinados em um *ensemble*, representado pela curva em preto, que como pode ser observado, possui capacidade de generalização maior que os regressores individuais.

Deve-se salientar que, no exemplo da Figura 1.1, os componentes do *ensemble* divergiam de forma equilibrada em torno do valor desejado, o que permitiu um bom desempenho do *ensemble*. No entanto, nem sempre os componentes gerados apresentam diferenças de comportamento que contribuem para a melhora do resultado obtido pelo *ensemble*, o que pode até levar a situações em que não se verifica nenhuma melhora em comparação com o os resultados do melhor componente individual. Diante disso, normalmente o processo de construção de *ensembles* engloba uma etapa de *seleção de componentes*, que tem como objetivo tentar selecionar apenas aqueles componentes gerados que contribuem para a melhora da saída do *ensemble*.

Na Figura 1.2 temos um exemplo do ganho de desempenho que pode ser obtido, com o uso de

ensembles, para um problema de classificação de padrões com três classes. As fronteiras de decisão de três propostas de classificadores são apresentadas nas Figuras 1.2(a)(b)(c). Como pode ser observado, os três classificadores apresentam fronteiras de decisão distintas, o que os leva a acertarem e errarem a classificação de amostras diferentes dos dados. No entanto, como pode ser visto nas Figuras 1.2(d) e (e), a combinação desses três classificadores em um *ensemble*, através de voto majoritário, é capaz de gerar uma nova fronteira de decisão capaz de separar corretamente todas as amostras do conjunto de dados em questão.

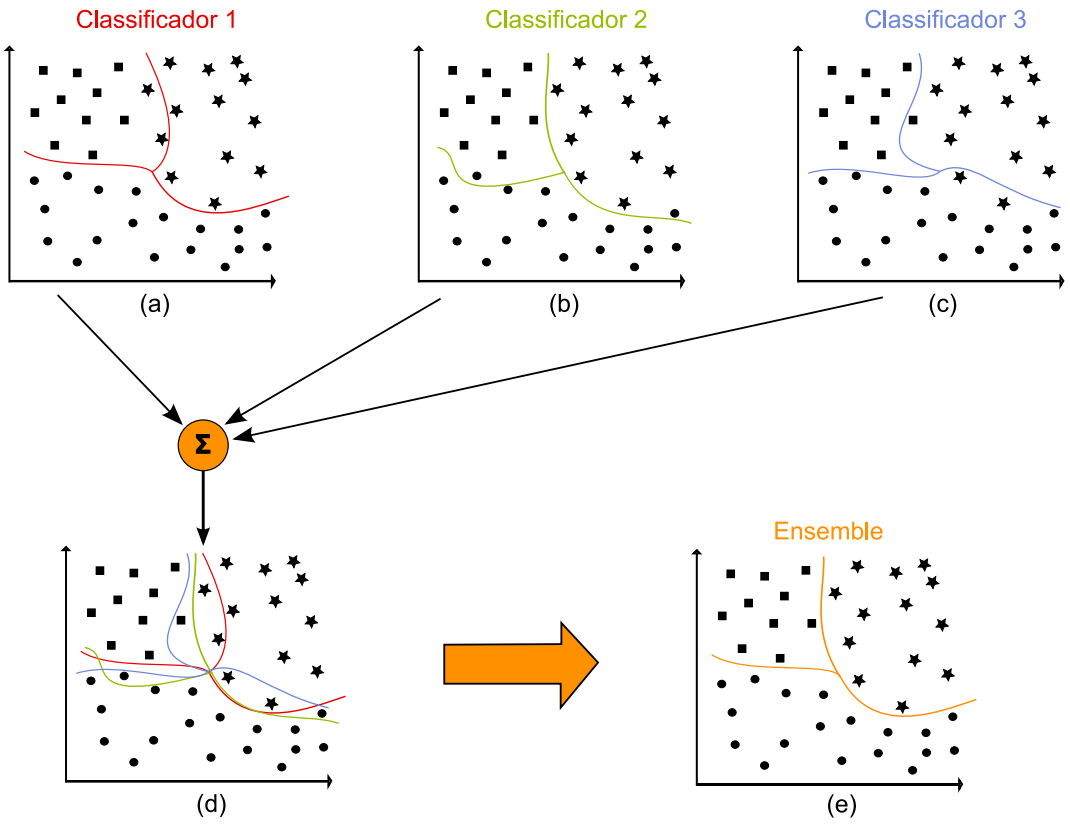


Fig. 1.2: Exemplo pictórico do ganho de desempenho proporcionado pelo uso de *ensembles* para um problema de classificação de padrões. As Figuras (a), (b) e (c) correspondem às fronteiras de decisão de três classificadores propostos para o problema em questão. A Figura (d) apresenta as três fronteiras de decisão juntas e a Figura (e) a fronteira de decisão obtida pela combinação dos três classificadores propostos em um *ensemble*, através de voto majoritário.

Apesar do *ensemble* apresentado da Figura 1.2(e) ter sido capaz de separar todas as amostras deste exemplo corretamente, é importante ressaltar aqui que, na prática, nem sempre os ganhos obtidos com o uso de *ensembles* são tão expressivos, podendo até mesmo não haver melhoras frente ao melhor classificado individual.

Para que a abordagem de *ensembles* seja capaz de promover melhorias de desempenho para um dado problema, os seus componentes devem apresentar bons desempenhos individuais e, ao mesmo

tempo, devem ter comportamentos diversos entre si, ou seja, eles devem apresentar diversidade em seus erros. Esta necessidade de diversidade de erros é, de certa forma, intuitiva, já que se considerarmos componentes que apresentam padrões de erros semelhantes, dificilmente sua combinação será capaz de compensar tais erros, uma vez que eles são inseridos no resultado final do *ensemble* por todos os componentes. Voltando ao exemplo do comitê de pessoas, se todos os indivíduos tiverem as mesmas opiniões e pontos de vista sobre o assunto em questão, ou seja, se eles não forem diversos entre si, dificilmente surgirão discussões que contribuam para uma melhora da decisão final tomada por este comitê. Nas Figuras 1.1 e 1.2 descritas anteriormente, pode-se observar que tanto os regressores quanto os classificadores apresentam comportamentos distintos entre si, o que pode permitir a melhora de desempenho quando combinados em *ensembles*.

Para melhor ilustrar essa diferença de comportamento, consideremos o exemplo de regressão da Figura 1.1. No intervalo $-0.96 < x < 0.00$, a curva gerada pelo regressor A está acima da curva desejada enquanto que a curva gerada pelo regressor B está abaixo. Já para o intervalo $0.00 < x < 0.96$, este comportamento se inverte. Se tivéssemos por exemplo, apenas componentes muito semelhantes ao regressor A (mesmo padrão de erros), dificilmente conseguiríamos obter uma melhora na aproximação da função como a apresentada pelo *ensemble* nesse exemplo.

Como mencionado anteriormente, neste trabalho serão estudados *ensembles* de redes neurais artificiais do tipo *multi-layer perceptron*, para aplicações em problemas de classificação de padrões. Uma das principais motivações para o uso de redes neurais do tipo MLP (*multi-layer perceptron*), além de sua simplicidade de operação, é sua característica de *aproximador universal de funções* (Hornik et al., 1989), ou seja, as redes MLP são capazes de aproximar qualquer função, dado um número suficiente de neurônios em sua camada intermediária.

Chama-se treinamento de uma rede neural o processo de ajuste dos parâmetros dessas redes, para que se tenha o menor erro possível para o mapeamento dos dados de entrada nos dados de saída desejados. Na literatura, existem várias propostas para o treinamento de redes MLP, sendo a mais conhecida o algoritmo de *backpropagation* (Haykin, 1999). No entanto, esse algoritmo se baseia no gradiente da função de erro da rede neural, o que o torna altamente susceptível a mínimos locais. Numa tentativa de contornar esse problema, várias técnicas que não se baseiam em gradiente foram propostas na literatura. Dentre elas, as técnicas evolutivas apresentam um importante papel, uma vez que são capazes de realizar uma ampla exploração do espaço de busca, sendo dessa forma capazes de escapar de mínimos locais ruins.

No campo de computação evolutiva, um paradigma relativamente novo, com aplicações em diversas áreas, são os chamados *Sistemas Imunológicos Artificiais* (SIAs), que surgiram a partir de tentativas de modelagem e aplicação de princípios imunológicos na resolução de problemas (de Castro & Timmis, 2002b). Quando comparados com outras estratégias evolutivas clássicas, os sistemas

imunológicos artificiais apresentam algumas vantagens, tais como:

- Os algoritmos baseados neste paradigma são inerentemente capazes de manter a diversidade entre os indivíduos da população;
- O tamanho da população (número de indivíduos) é dinamicamente ajustado de acordo com a demanda da aplicação;
- As soluções ótimas locais tendem a ser simultaneamente preservadas quando localizadas.

Tais características, em especial os eficientes mecanismos de manutenção de diversidade, foram exploradas neste trabalho visando a obtenção de *ensembles* de alta performance.

1.2 Objetivos do Trabalho

O principal objetivo deste trabalho é explorar os mecanismos de manutenção de diversidade, apresentados pelos sistemas imunológicos artificiais, para a geração de um conjunto de redes neurais artificiais, do tipo MLP, que apresentem um bom desempenho individual e ao mesmo tempo diversidade entre seus erros, para que possam ser consideradas candidatas à constituição de *ensembles*. Para isso foi utilizado o algoritmo *opt-aiNet*, proposto por de Castro & Timmis (2002a) para a geração da população de redes neurais com as características desejadas. Attux et al. (2005) e Pasti & de Castro (2006) já utilizaram esse algoritmo para o treinamento de redes MLP. No entanto, neste trabalho os mecanismos de manutenção de diversidade dentre os indivíduos da população, presentes no *opt-aiNet*, foram adaptados de forma que o algoritmo possa gerar redes neurais que, além de apresentarem bons desempenhos individuais, sejam também diversas em seus padrões de classificação, ou seja, apresentem diversidade em seus erros.

Vencida a etapa de geração de componentes candidatos à constituição dos *ensembles*, foram aplicadas algumas técnicas de seleção de componentes, já descritas na literatura, para selecionar dentre esses candidatos aqueles que realmente contribuem para o desempenho do *ensemble*. Na literatura de *ensembles*, dois paradigmas de seleção de componentes são os mais comuns: o paradigma *construtivo* e o de *poda*. Diante disso, foram utilizadas neste trabalho duas técnicas construtivas (*construtiva sem exploração* e *construtiva com exploração*) e duas técnicas de poda (*poda sem exploração* e *poda com exploração*), além de uma técnica de seleção baseada em um algoritmo de agrupamento de dados (*clusterização*).

Por fim, a última etapa de construção dos *ensembles* é a combinação dos resultados gerados por cada componente selecionado no resultado do *ensemble* propriamente dito. Para isso foram utilizadas

cinco das técnicas de combinação de componentes mais comuns na literatura: *média simples*, *média ponderada com bias*, *média ponderada sem bias*, *voto majoritário* e *winner-takes-all*.

A metodologia proposta neste trabalho foi então aplicada a quatro problemas de classificação de padrões, sendo três problemas reais, e os resultados obtidos pelos *ensembles* foram comparados com os resultados obtidos pelos melhores classificadores individualmente. Foram analisados os índices de acertos, o número de componentes em cada *ensemble* e a diversidade entre estes componentes. Além disso, foi proposta uma métrica de confiança para os resultados de classificação, e buscou-se verificar a influência do uso de *ensembles* sobre essa métrica, quando comparada com os valores obtidos pelos melhores classificadores individuais.

1.3 Organização do Documento

Esta dissertação está dividida em seis capítulos, que cobrem os aspectos teóricos e práticos deste trabalho. No Capítulo 2, serão discutidos mais a fundo o conceito de *ensembles* e a questão da diversidade de seus componentes, além de também ser apresentada uma breve revisão sobre métodos de geração de diversidade presentes na literatura.

O Capítulo 3 tratará dos sistemas imunológicos artificiais e do algoritmo *opt-aiNet*, utilizado neste trabalho para a geração dos componentes para *ensembles*. Neste capítulo, serão apresentados também os principais mecanismos de funcionamento e teorias existentes sobre o sistema imunológico natural, que servem de inspiração aos sistemas imunológicos artificiais, especialmente ao algoritmo *opt-aiNet*.

No Capítulo 4, será feita uma breve apresentação das redes neurais do tipo MLP e será detalhada a proposta deste trabalho. Serão discutidas as modificações feitas no algoritmo *opt-aiNet*, a estrutura das redes neurais utilizadas, e serão detalhadas as técnicas de seleção e combinação de componentes para *ensembles* adotadas neste trabalho.

A metodologia experimental, os problemas tratados aqui e os resultados experimentais obtidos serão descritos e discutidos no Capítulo 5.

Por fim, no Capítulo 6, será feita uma recapitulação do que foi proposto neste trabalho e serão apresentadas as conclusões obtidas, bem como as perspectivas para trabalhos futuros.

Capítulo 2

Ensembles e a Questão da Diversidade

Ensemble é um paradigma de aprendizado em que propostas alternativas de solução para um problema, denominadas *componentes*, têm suas saídas individuais combinadas na obtenção de uma solução final. Esta abordagem tem sido amplamente utilizada na última década, tanto para problemas de regressão quanto para problemas de classificação de padrões, uma vez que os *ensembles* são comprovadamente capazes de aumentar a capacidade de generalização e, conseqüentemente, o desempenho geral do sistema (Hansen & Salamon, 1990; Hashem et al., 1994; Zhou et al., 2000; Zhou & Jiang, 2003; Inoue & Narihisa, 2004; de Castro et al., 2005; Barajas-Montiel & Reyes-García, 2005; Twala & Cartwright, 2005; Lu et al., 2006; Polikar, 2006). Intuitivamente, a combinação de múltiplos componentes é vantajosa, uma vez que componentes diferentes podem implicitamente representar aspectos distintos e, ao mesmo tempo, relevantes para a solução de um dado problema. A Figura 2.1 ilustra a estrutura geral de um *ensemble*. Se considerarmos que temos um *ensemble* de classificadores (o que é o caso deste trabalho), cada componente será um classificador (rede neural, árvore de decisão, classificador baseado em regras, etc...) proposto independentemente e capaz de atuar isoladamente. Para cada conjunto de dados de entrada, os M componentes gerarão M saídas que serão então combinadas para produzir a saída final do *ensemble*.

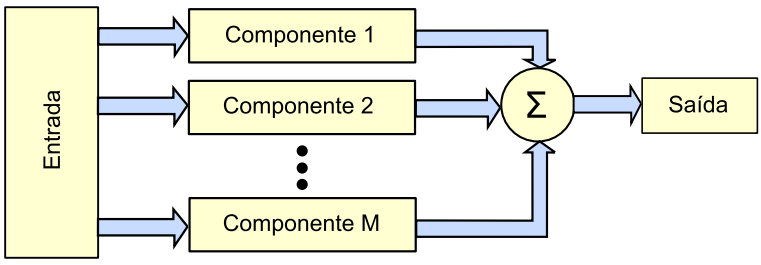


Fig. 2.1: Estrutura geral de um *ensemble*. Os M componentes recebem os mesmos dados de entrada e geram resultados individuais, que são combinados ($\sum$) em uma única saída para o problema.

O paradigma de *ensembles* originou-se do trabalho de Hansen & Salamon (1990), que mostraram que a combinação de várias redes neurais artificiais, treinadas separadamente, pode melhorar significativamente a capacidade de generalização do sistema. O bom desempenho dos *ensembles* estimulou a sua aplicação na tentativa de resolução de diversos problemas, tanto na área de classificação de padrões – como em reconhecimento de face (Huang et al., 2000; Lu et al., 2006), reconhecimento de caracteres (Hansen et al., 1992; Mao, 1998; Liu, 2005), análise de imagens (Cherkauer, 1996), engenharia de software (Twala & Cartwright, 2005) e diagnóstico médico (Zhou et al., 2000; Zhou & Jiang, 2003; Wang et al., 2005) – quanto na área de regressão – como em aproximação de funções (Hashem & Schmeiser, 1995; Lima et al., 2002) e predição de séries temporais (Inoue & Narihisa, 2000; Wichard & Ogorzalek, 2004) – levando inclusive à utilização simultânea de diversos tipos de componentes, além de redes neurais artificiais, formando os chamados *ensembles heterogêneos*.

No entanto, essa melhora proporcionada pelos *ensembles* na capacidade de generalização e, conseqüentemente, no desempenho total do sistema, se apóia na qualidade e *diversidade do erro* apresentado pelos seus componentes (Perrone & Cooper, 1993), ou seja, cada um dos componentes em um *ensemble* deve apresentar um bom desempenho quando aplicado isoladamente ao problema e, ao mesmo tempo, quando comparados com os demais componentes, devem "cometer erros" distintos. Essa necessidade de diversidade do erro dos componentes é de certa forma intuitiva, uma vez que, se combinarmos vários componentes que apresentam um mesmo padrão de erro, claramente não teremos nenhum incremento de desempenho, já que o fato deles errarem para um mesmo subconjunto de estímulos de entrada implica em acertos também coincidentes, o que faz com que sua combinação traga apenas um aumento no custo computacional, sem resultados práticos de desempenho. Esta questão da diversidade de componentes será tratada mais a fundo nas próximas seções deste capítulo.

A construção de um *ensemble* geralmente é feita em três etapas: (i) *geração* de um conjunto de elementos candidatos a fazerem parte do *ensemble*; (ii) *seleção* dos candidatos que realmente são necessários e que contribuem de alguma forma ao serem inseridos no *ensemble*; e (iii) *combinação* das saídas geradas por esses elementos selecionados (componentes) em uma única saída, que corresponderá ao resultado final do *ensemble*. Apesar de alguns autores não incluírem a etapa de seleção na construção de seus *ensembles*, ela é de grande importância, uma vez que Zhou et al. (2002) mostraram que o uso de todos os candidatos disponíveis no *ensemble* pode degradar seu desempenho.

Sob a perspectiva de aprendizagem supervisionada, o conjunto de amostras de entrada e saída disponível para um dado problema pode ser separado em subconjuntos que representam descrições equivalentes da mesma distribuição estatística. Dessa forma, para contribuir ainda mais para o aumento da capacidade de generalização, na construção de *ensembles* o conjunto original de dados deveria ser idealmente subdividido em cinco sub-conjuntos: *treinamento*, *validação*, *seleção*, *combinação* e *teste*. Com isso, na etapa de geração de componentes seriam utilizados os sub-conjuntos de

treinamento e *validação*, na etapa de seleção o sub-conjunto de *seleção* e na etapa de combinação o sub-conjunto de *combinação*. O desempenho do *ensemble* obtido seria então verificado utilizando-se o sub-conjunto de *teste*. No entanto, como normalmente o número de amostras de entrada e saída disponível é limitado, os conjuntos de *seleção* e *combinação* são então fundidos com o conjunto de *validação*, e a construção de *ensembles* passa a utilizar o conjunto de *treinamento* na geração de componentes, o de *validação* nas etapas de seleção e combinação, e o de *teste* na verificação de desempenho do *ensemble* obtido. Esta subdivisão do conjunto de dados original em apenas três sub-conjuntos é muito comum em aprendizado de máquina, principalmente em algoritmos que requerem algum tipo de treinamento. No caso de *ensembles*, essa subdivisão pode ser encontrada na literatura em Lima (2004), e também foi adotada neste trabalho.

No restante deste capítulo, será feito um breve resumo sobre a questão da diversidade, tanto para problemas de regressão quanto para problemas de classificação, e serão abordados alguns dos principais métodos de criação de diversidade apresentados na literatura. Para facilitar a exposição de tais métodos, será parcialmente seguida a taxonomia proposta por Brown et al. (2005).

2.1 A Questão da Diversidade

Os primeiros estudos sobre a combinação de componentes para problemas de regressão foram feitos paralelamente por Perrone (1993) e Hashem (1993), tornando-se um tópico intensamente investigado nos anos subseqüentes. Esse interesse acabou por contribuir muito para o amadurecimento do conceito de diversidade de erros em regressores, levando ao surgimento das teorias de *Ambiguity Decomposition*, proposta por Krogh & Vedelsby (1995), *Bias-variance Decomposition*, proposta por Geman et al. (1992) e *Bias-Variance-Covariance Decomposition*, proposta por Ueda & Nakano (1996).

Por outro lado, no contexto de classificação de padrões, ainda não existe nenhum trabalho que tenha conseguido mostrar como a diversidade entre as saídas de classificadores individuais influenciam no desempenho do *ensemble* como um todo, a não ser que tais problemas de classificação possam ser reformulados como problemas de regressão. Nesse caso, os conceitos disponíveis para regressão se aplicam, como realizado em Tumer & Ghosh (1996), Roli & Fumera (2002) e Fumera & Roli (2003). Quando os classificadores envolvidos são capazes de apresentar apenas saídas discretas, como em Árvores de Decisão, o problema se torna bem mais complicado, já que não é possível estabelecer uma relação de ordem entre as saídas, de forma que o conceito de covariância não pode ser definido (Brown et al., 2005). Diante dessas dificuldades, que ainda não permitiram a criação de uma situação equivalente ao caso de regressão, onde o erro quadrático do *ensemble* pode ser decomposto no erro quadrático de cada componente e em um termo que quantifica sua correlação, vários trabalhos em-

píricos buscaram obter expressões heurísticas, no contexto de classificação, que aproximassem esse termo de diversidade, como pode ser visto em Sharkey & Sharkey (1997a), Cunningham & Carney (2000), Kuncheva et al. (2000), Kuncheva & Whitaker (2001), Zenobi & Cunningham (2001), Kuncheva & Kountchev (2002), Shipp & Kuncheva (2002), Skurichina et al. (2002), Kuncheva (2003), Kuncheva et al. (2003) e Kuncheva & Whitaker (2003).

Como é possível observar pelas referências acima, a grande maioria dos trabalhos envolvendo observações empíricas sobre a questão da diversidade em classificadores se deve a Ludmila I. Kuncheva. Em seus trabalhos, Kuncheva organizou várias métricas de diversidade presentes na literatura e as dividiu em duas classes distintas: aquelas que consistem em tomar a média de uma dada métrica de distância entre todos os classificadores do *ensemble* (medidas do tipo *par-a-par*) e aquelas que se baseiam em *entropia* ou na correlação de cada classificador com a saída média dos classificadores (medidas que não são do tipo *par-a-par*).

Dentre a variedade de métricas par-a-par presentes na literatura, podemos citar a *estatística Q* (Yule, 1900), o *coeficiente de correlação* (Mardia et al., 1979), a *métrica de não-concordância* (Skalak, 1996) e a *medida de dupla-falta* (Giacinto & Roli, 2001) como algumas das mais utilizadas. No entanto, neste trabalho, vamos considerar apenas a *estatística Q*. Seja $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ um conjunto de dados rotulados, $\mathbf{z}_j \in \Re^n$, proveniente de um problema de classificação. A saída de um classificador D_i pode ser representada por um vetor binário N-dimensional $\mathbf{y}_i = [y_{1,i}, \ldots, y_{N,i}]^T$, tal que $y_{j,i} = 1$ se D_i é capaz de classificar corretamente a amostra $\mathbf{z}_j$ e $y_{j,i} = 0$ caso ele erre na classificação dessa amostra (esse vetor binário $\mathbf{y}_i$ é conhecido na literatura como *saída oráculo* do classificador D_i, mas neste trabalho adotaremos o termo *vetor de acertos*, para evitar confusões com a chamada *curva oráculo*, utilizada no Capítulo 5). Dessa forma, para dois classificadores D_i e D_k, a estatística Q é dada pela Expressão 2.1 abaixo:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \tag{2.1}$$

onde N^{11} é o número de amostras corretamente classificadas pelos classificadores D_i e D_k, N^{00} é o número de amostras incorretamente classificadas pelos classificadores D_i e D_k, N^{10} é o número de amostras corretamente classificadas pelo classificador D_i e incorretamente classificadas pelo classificador D_k e N^{01} é o número de amostras incorretamente classificadas pelo classificador D_i e corretamente classificadas pelo classificador D_k (vide Tabela 2.1). Note que $N = N^{11} + N^{00} + N^{01} + N^{10}$, onde N é o número total de dados rotulados.

$Q_{i,k}$ pode assumir valores entre -1 e $+1$. Classificadores que classificam corretamente as mesmas amostras de um problema tendem a produzir valores positivos de Q, enquanto que aqueles que cometem erros em amostras diferentes tendem a produzir valores negativos de Q. Para classificadores *estatisticamente independentes*, $Q_{i,k}$ tende ao valor 0. Considerando um conjunto D de M classi-

Tab. 2.1: Ilustração das relações entre acertos e erros dos classificadores D_i e D_k, utilizadas no cálculo da estatística Q.

	D_k correto (1)	D_k incorreto (0)
D_i correto (1)	N^{11}	N^{10}
D_i incorreto (0)	N^{01}	N^{00}

ficadores, o valor médio da estatística Q tomada em todos os pares de classificadores é dado pela Expressão 2.2:

$$Q_{med} = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} Q_{i,k}. \tag{2.2}$$

Neste trabalho, foi utilizada uma métrica de diversidade par-a-par de classificadores baseada na distância de Hamming dos vetores de acertos, a qual foi proposta por de Castro et al. (2005), com bons resultados, e é descrita na Seção 4.2.3. A estatística Q também foi utilizada nos experimentos para verificação dos resultados (vide Capítulo 5).

Apenas para ilustrar o caso de métricas de diversidade de classificadores que não são do tipo par-a-par, tomemos agora a chamada *Ambigüidade de Classificação*, proposta por Zenobi & Cunningham (2001). Nesta métrica, a ambigüidade média para o i-ésimo classificador, tomada em N amostras, é dada pela Expressão 2.3:

$$A_i = \frac{1}{N} \sum_{n=1}^{N} a_i(x_n), \tag{2.3}$$

onde $a_i(x_n)$ é 1 se a saída do indivíduo i é diferente da saída do *ensemble* e 0 caso contrário. Com isso, a *Ambigüidade de classificação* do *ensemble* (com M componentes) é dada pela Expressão 2.4:

$$\bar{A} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{M} \sum_{i=1}^{M} a_i(x_n). \tag{2.4}$$

Apesar das definições propostas na literatura para métricas de diversidade entre classificadores serem aparentemente intuitivas, ainda não foi provada nenhuma relação formal entre tais métricas e o erro total do *ensemble*, o que mostra que muito ainda deve ser feito nesta área.

2.2 Abordagens de Criação de Diversidade em *Ensembles*

Na etapa de construção de *ensembles*, existem técnicas que tentam explicitamente otimizar uma dada métrica de diversidade, enquanto que outras não. Isto nos permite fazer uma distinção entre essas duas abordagens, classificando cada um dos métodos como métodos com diversidade *implícita* e métodos com diversidade *explícita*.

Durante a aprendizagem, cada componente do *ensemble* segue uma trajetória no *espaço de hipóteses*[1] e, para que se tenha um bom desempenho, é desejável que tais componentes ocupem pontos diferentes neste espaço. Diante disso, os métodos implícitos se baseiam na aleatoriedade para que trajetórias diversas sejam geradas, enquanto que os métodos explícitos escolhem deterministicamente diferentes caminhos neste espaço (Brown et al., 2005).

Além desta dicotomia entre métodos implícitos e métodos explícitos, é possível classificar os métodos de geração de diversidade em *ensembles* baseando-se em outras características além das duas citadas. Nesta seção, utilizaremos a seguinte divisão:

- **Métodos que atuam sobre o ponto de partida no espaço de hipóteses:** os métodos incluídos neste grupo variam os pontos de partida da busca no espaço de hipóteses, influenciando dessa forma o ponto de convergência.

- **Métodos que atuam sobre os dados de treinamento:** através do fornecimento de conjuntos de dados de treinamento diferentes para cada um dos componentes do *ensemble*, estes métodos buscam gerar componentes que produzam mapeamentos diferentes, visto que os estímulos de entrada serão distintos.

- **Métodos que manipulam a arquitetura de cada componente:** estes métodos variam a arquitetura de cada componente no *ensemble*, de maneira que diferentes conjuntos de hipóteses estejam acessíveis para cada componente, ou seja, como os componentes do *ensemble* possuem arquiteturas diferentes, os conjuntos de hipóteses associados a esses componentes também serão distintos, o que pode contribuir para a diversidade.

- **Métodos que atuam sobre a forma de exploração do espaço de hipóteses:** alterando a forma de exploração do espaço de hipóteses, esses métodos levam os diferentes componentes a convergirem para diferentes hipóteses, mesmo tendo um mesmo ponto de partida.

- **Métodos híbridos:** formados por alguma combinação dos métodos acima.

[1] Neste trabalho, chamaremos de *hipótese* cada mapeamento entrada-saída feito por uma rede neural. Diante isso, o conjunto de todos os mapeamentos possíveis de serem representados por uma rede neural multicamadas será definido como *espaço de hipóteses*.

2.2.1 Ponto de Partida no Espaço e Hipóteses

Como este trabalho trata de *ensembles* de redes neurais MLPs, deste ponto em diante nos restringiremos ao tratamento apenas de *ensembles* com este tipo de componente, sendo explicitado no texto quando alguma abordagem utilizar outros tipos de componentes.

Dentro do subconjunto de métodos de geração de diversidade em *ensembles* que atuam sobre o ponto de partida no espaço de hipóteses, podemos novamente fazer uma subdivisão entre métodos *implícitos* e métodos *explícitos*. Por métodos implícitos, compreendem-se aqueles cujo conjunto de pesos iniciais das redes neurais são gerados aleatoriamente. Já os métodos explícitos são aqueles em que as redes neurais são concebidas de modo a amostrar regiões distintas do espaço de hipóteses.

A estratégia mais comum dentre estes métodos é a inicialização de cada rede neural com um conjunto de pesos diferente, gerado aleatoriamente. No entanto, esta técnica tem se mostrado a menos eficiente na obtenção de boa diversidade, como pode ser observado em Sharkey et al. (1995), Partridge & Yates (1996), Yates & Partridge (1996) e Parmanto et al. (1996).

Em relação aos métodos explícitos, existem poucos trabalhos na literatura. Maclin & Shavlik (1995) propuseram uma abordagem de inicialização de pesos em redes neurais que utiliza aprendizagem competitiva para posicionar as redes inicialmente em pontos distantes da origem do espaço de pesos, buscando assim aumentar o conjunto de mínimos locais passíveis de serem atingidos.

2.2.2 Atuação sobre os Dados de Treinamento

Uma das formas de treinamento de *ensembles* mais pesquisadas é aquela que busca produzir diversidade através do fornecimento de conjuntos de treinamento diferentes para cada um dos componentes. Existem várias maneiras de gerar tais conjuntos de treinamento. Uma delas seria fornecer, a cada componente em treinamento, subconjuntos de amostras diferentes, com todos os atributos do conjunto de treinamento original. Outra seria fornecer todas as amostras presentes mas formando-se subconjuntos com atributos diferentes. Uma terceira maneira seria pré-processar os atributos de forma a obter uma representação diferente, como por exemplo aplicar uma análise de componentes principais (Mardia et al., 1979). O espaço dos possíveis conjuntos de treinamento descritos acima pode ser visualizado na Figura 2.2. Às duas primeiras alternativas dá-se o nome de técnicas de *re-amostragem*, enquanto que a terceira é chamada de técnica de *distorção* (Sharkey et al., 2000).

Krogh & Vedelsby (1995) se basearam num dos métodos de re-amostragem mais conhecidos, o *k-fold cross-validation*, para o treinamento de componentes para *ensembles*. Nesse trabalho, o conjunto de dados é dividido aleatoriamente em k subconjuntos disjuntos e os novos conjuntos de treinamento para cada membro do *ensemble* são gerados excluindo-se um desses k subconjuntos e tomando-se os restantes.

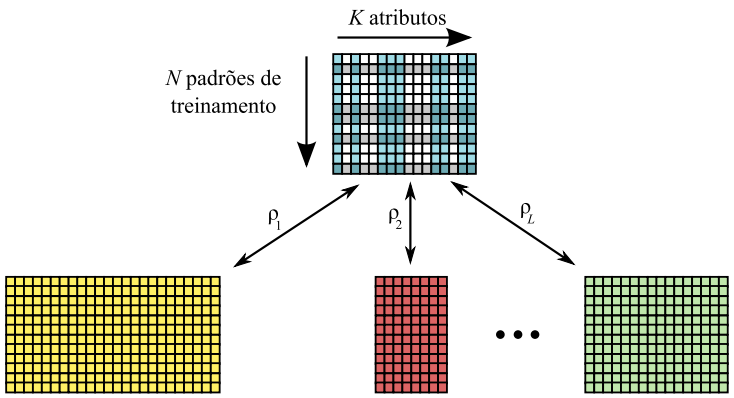


Fig. 2.2: Espaço dos possíveis conjuntos de treinamento para componentes de um *ensemble*. É possível que seja feita uma seleção de um sub-conjunto de amostras, dentre os N padrões de treinamento; seleção de um sub-conjunto dentre os K atributos disponíveis; ou o pré-processamento do conjunto de atributos através da aplicação de transformações ρ_i.

Os algoritmos *Bagging* (*Bootstrap Aggregating*) e *Boosting*, propostos por Breiman (1996) e Schapire (1990), respectivamente, são outros exemplos de re-amostragem. Atualmente, estes dois algoritmos são as abordagens predominantes na geração de componentes para *ensembles*.

O algoritmo *Bagging* se baseia na amostragem *bootstrap* (Efron & Tibshirani, 1993) e gera vários conjuntos de treinamento a partir de amostragem uniforme do conjunto de dados, com reposição, e utiliza cada um desses conjuntos para treinar uma rede neural (ou outro tipo de classificador/regressor). Os conjuntos de treinamento possuem o mesmo número de amostras que o conjunto de dados original, mas algumas amostras podem ser selecionadas mais de uma vez, o que conseqüentemente implica que podem existir amostras que não são selecionadas.

Já o algoritmo *Boosting*, apesar de ter sido proposto por Schapire (1990), foi aperfeiçoado por Freund & Schapire (1995) e Freund (1995). Neste método, os conjuntos de treinamento não são gerados via amostragem uniforme, como no algoritmo *Bagging*. A probabilidade de uma dada amostra ser escolhida depende da contribuição desta para o erro dos componentes já treinados. Tomando como exemplo um problema de classificação, uma amostra que não seja corretamente classificada pelos componentes já treinados terá uma maior probabilidade de ser selecionada que uma amostra que seja corretamente classificada. Com isso, esta amostra terá uma chance maior de compor o conjunto de treinamento do próximo componente a ser treinado. Um dos problemas desta técnica é a necessidade de treinamento seqüencial dos componentes do *ensemble*, uma vez que as probabilidades devem ser redefinidas de acordo com o comportamento dos componentes já treinados. A versão mais popular do algoritmo de *Boosting* é chamada *AdaBoost* (Freund & Schapire, 1996), na qual cada componente é treinado seqüencialmente e a distribuição dos dados de treinamento é feita levando-se em consideração os erros do componente treinado no passo imediatamente anterior. Uma variante do *AdaBoost*

foi proposta por Oza (2003) e, neste algoritmo, a distribuição das amostras se baseia no erro de todos os componentes treinados até o momento, e não apenas no erro do componente treinado no passo imediatamente anterior, como no caso da proposta de Freund & Schapire (1996).

Dentre as técnicas de distorção, Sharkey et al. (1996) e Sharkey & Sharkey (1997b) adicionam aos atributos originais um grupo de novos atributos gerados aplicando-se uma transformação aleatória através da passagem desses atributos originais por uma rede neural não treinada. Esta técnica levou a um desempenho melhor que o obtido com *ensembles* gerados apenas com os dados sem transformação. Uma outra técnica, proposta por Raviv & Intrator (1996), consiste em aplicar uma amostragem como em *Bagging*, mas adicionando ruído gaussiano aos dados de entrada.

A medida de ambigüidade proposta por Zenobi & Cunningham (2001) e dada pela Expressão 2.4 é utilizada em Zenobi & Cunningham (2001) para selecionar subconjuntos de atributos para cada componente e em Melville & Mooney (2003) para determinar se um componente será ou não inserido no *ensemble*. A seleção de atributos de treinamento para cada componente em Zenobi & Cunningham (2001) é feita utilizando uma estratégia de *hill-climbing* baseada no erro individual e na diversidade estimada do *ensemble* até o momento. A construção do *ensemble* é feita através de adições sucessivas dos componentes e um componente é rejeitado se reduzir a diversidade de acordo com um limiar pré-determinado ou se aumentar o erro do *ensemble*. Já em Melville & Mooney (2003), os componentes são treinados com os dados originais acrescidos de um *conjunto diverso* de novos exemplos artificialmente gerados. Os padrões presentes neste novo conjunto primeiramente são submetidos ao *ensemble* e, em seguida, são rotulados com o oposto dos resultados gerados pelo *ensemble*. Dessa forma, um componente treinado com este conjunto terá uma alta discordância com o *ensemble*, aumentando a diversidade e, supostamente, contribuindo para a capacidade de generalização do *ensemble*. Se o erro não for diminuído, um novo conjunto diverso é gerado e um novo componente é treinado, até que se atinja o número desejado de componentes ou o máximo de iterações permitido.

Também através do uso de seleção de atributos, Oza & Tumer (2001) buscaram reduzir a correlação entre os indivíduos. A seleção de atributos foi feita baseada no cálculo da correlação entre cada atributo e cada classe. Com isso, os componentes foram treinados de forma a se tornarem especialistas em cada classe ou grupo de classes.

Liao & Moody (2000) agruparam todas as variáveis de entrada baseando-se em sua informação mútua (Haykin, 1999). Em sua proposta, variáveis estatisticamente semelhantes foram agrupadas e o conjunto de treinamento de cada componente foi formado por variáveis selecionadas a partir de grupos diferentes.

Exceto pelo trabalho de Melville & Mooney (2003), até agora foram discutidos apenas métodos que manipulam os dados de *entrada*. No entanto, alguns métodos atuam sobre os dados de saída. Entre eles, Breiman (1998) propôs a adição de ruído à saída dos dados, o que levou a um desempenho

superior ao obtido pela técnica de *Bagging*, mas a resultados próximos aos obtidos pelo *AdaBoost*. Já Dietterich & Bakiri (1991) manipularam os dados de saída através da técnica de codificação da saída por correção de erro (do inglês *Error-Correcting Output Coding - ECOC*). Nesta técnica, cada classe do problema é representada por um vetor binário, ortogonal à representação das demais classes, e os componentes são treinados. Para definir a qual classe um componente atribui uma dada amostra, caso o vetor de saída produzido não corresponda exatamente à representação de uma dada classe, é atribuído o rótulo da classe que possuir a menor distância de Hamming ao vetor gerado. Em Kong & Dietterich (1995), a técnica ECOC foi analisada e concluiu-se que sua eficácia se dá por reduzir a variância do ensemble e corrigir eventuais polarizações. Na técnica ECOC, a diversidade entre os componentes é gerada pela aproximação feita entre as saídas de cada componente e a representação binária mais próxima de acordo com a distância de Hamming.

2.2.3 Manipulação de Arquitetura

Apesar de ser aparentemente intuitivo que o uso de diferentes tipos de componentes em um *ensemble* contribui para a geração de diversidade do erro, esta abordagem não é muito explorada na literatura.

No caso de trabalhos que manipulam a arquitetura através da variação do número de neurônios na camada intermediária de redes MLP ou do uso de diferentes tipos de redes neurais em um *ensemble*, Partridge & Yates (1996) e Partridge (1996) exploraram estruturas com redes MLP de diferentes números de neurônios na camada intermediária (variando de 8 a 12 neurônios) e concluíram que esta abordagem de geração de diversidade só não é mais ineficiente que a inicialização aleatória de pesos, descrita na seção 2.2.1. No entanto, esses trabalhos se basearam apenas em um único conjunto de dados. Yates & Partridge (1996) também pesquisaram a influência do tipo de rede neural sobre a diversidade, construindo *ensembles* com redes MLP e RBF, e verificando que a mistura de tipos de redes é mais efetiva que a variação das dimensões da camada oculta de redes MLP.

Opitz & Shavlik (1996) utilizaram um algoritmo evolutivo para otimizar as topologias dos componentes. Neste trabalho, cada um dos componentes foi treinado via *backpropagation* e o processo de seleção visou otimizar métricas de diversidade.

Islam et al. (2003) propuseram o algoritmo CNNE (do inglês *Cooperative Neural Network Ensembles*), que gera *ensembles* construtivamente, monitorando a diversidade durante o processo. Durante a execução, o CNNE determina a estrutura do *ensemble*, treina as redes neurais individuais através da aprendizagem com *Correlação Negativa* (Liu, 1998; Brown, 2004) e encoraja a diversidade. Esta técnica determina automaticamente o número de redes neurais no *ensemble* e o número de neurônios na camada oculta.

Ensembles Heterogêneos

Além de *ensembles* cujos componentes são redes neurais de estruturas e tipos diferentes, existem também os *ensembles heterogêneos* (ou *ensembles híbridos*, em algumas publicações) cujos componentes são elementos de paradigmas distintos. Apenas para ilustrar, tomemos como exemplo um problema de classificação. Neste caso, um *ensemble* heterogêneo poderia ser constituído de redes neurais e classificadores baseados em regras, por exemplo.

Apesar da existência de poucos estudos sobre tais *ensembles*, eles mostram que o uso de diferentes paradigmas leva a componentes com diferentes especialidades e precisões, que podem apresentar diferentes desempenhos e, com isso, diferentes padrões de generalização. Esta especialização pode indicar que, em *ensembles* heterogêneos, a *seleção* da saída de um único regressor ou classificador é mais eficiente que a *fusão* das saídas de todos eles, como normalmente é feito em *ensembles* de redes neurais (Brown et al., 2005). Nessa linha, Woods et al. (1997) utilizaram como componentes redes neurais, classificadores do tipo *k-nearest neighbour*, árvores de decisão e classificadores quadráticos bayesianos e, para cada componente, utilizaram a estimativa de sua precisão local no espaço de atributos para escolher qual desses componentes seria responsável pela resposta para uma nova entrada.

Os *ensembles* heterogêneos em si e a escolha de apenas um componente para determinar a saída do *ensemble* podem ser vistos como uma tentativa de exploração do teorema do *No Free Lunch*, proposto por Wolpert & Mcready (1997). Em linhas gerais, para o caso de classificação de padrões, esse teorema nos diz que, considerando-se todos os problemas de classificação possíveis, na média todas as abordagens para classificação apresentarão o mesmo desempenho. Dessa forma, a única maneira de uma dada técnica superar o desempenho de outra é em casos em que há uma especialização junto a um dado problema. No caso de *ensembles* heterogêneos, onde cada componente representa uma abordagem diferente de resolução para o problema sendo tratado, a determinação da saída do *ensemble* através da escolha da saída de um dos componentes nada mais é que a escolha do componente mais especializado para aquele problema em questão, que, de acordo com o teorema do *No Free Lunch*, tende a superar o desempenho dos demais.

Dentre os outros trabalhos nesta linha de *ensembles* heterogêneos, Wang et al. (2000) utilizaram como componentes redes neurais e árvores de decisão, chegando à conclusão de que os melhores desempenhos são obtidos com um número de redes neurais maior que o de árvores de decisão, mas com pelo menos uma árvore no *ensemble*. Langdon et al. (2002) também utilizaram redes neurais e árvores de decisão, mas aplicaram Programação Genética (Koza, 1992) para evoluir uma regra de combinação dos indivíduos. Soares et al. (2006) utilizaram como componentes redes neurais do tipo MLP, redes neurais do tipo *radial basis function* (RBF), classificadores *naïve Bayes*, máquinas de vetores suporte (SVM) e classificadores baseados em aprendizado de regras proposicionais, e

propuseram duas técnicas de seleção de componentes (baseadas em algoritmo de agrupamento e *k-nearest neighbours*) que buscam não apenas reduzir o erro do *ensemble*, mas também aumentar a diversidade de seus componentes.

2.2.4 Exploração do Espaço de Hipóteses

Dado um espaço de busca definido pela arquitetura da rede neural e pelos dados de treinamento, é possível ocuparmos qualquer ponto nesse espaço. No entanto, o tipo de *ensemble* que teremos no final dependerá da maneira como é feita tal exploração. Os métodos de criação de diversidade que atuam sob a forma de exploração do espaço de hipóteses podem ser divididos em duas sub-categorias: *métodos de otimização convencional*, como os chamados *métodos de penalidades*, que adicionam um termo de custo (por ausência de diversidade) ao erro de cada rede neural, buscando a criação de hipóteses diversas, e os *métodos de busca exploratória*, onde estão os *algoritmos evolutivos*, que são métodos de busca populacionais que encorajam a diversidade na população de candidatos.

2.2.4.1 Métodos de Otimização Convencional

Como o objetivo dos métodos convencionais de otimização normalmente é minimizar o erro na saída de uma rede neural, geralmente essas técnicas se baseiam no *gradiente* da função de erro do sistema (como, por exemplo, no algoritmo de *backpropagation* (Haykin, 1999)) e, a partir de um dado ponto inicial, seguem uma "trajetória gulosa" (ou seja, sempre na direção que promover a maior redução da função de erro) até encontrarem um ponto de mínimo. Desconsiderando os problemas característicos desta abordagem, como a susceptibilidade a mínimos locais, o objetivo desses métodos é simplesmente obter o menor erro possível para a rede neural sendo treinada, sem levar em conta nenhuma questão de diversidade. Diante disso, com o objetivo de, além de reduzir o erro do componente sendo treinado, estimular a diversidade deste componente em relação aos demais (já treinados ou em processo de treinamento simultâneo), surgiram os chamados *métodos de penalidade*.

Nos métodos de penalidade, o erro de cada rede i se torna:

$$e_i = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(f_i(n) - d(n))^2 + \frac{1}{N}\sum_{n=1}^{N}\lambda R_i(n), \tag{2.5}$$

onde N é o número de amostras, f_i é a saída da rede neural i, d é a saída desejada e λ é um fator de ponderação do termo de penalidade R_i. Pode-se perceber que, quando o fator de ponderação é $\lambda = 0$, teremos um *ensemble* em que cada componente será treinado pelo algoritmo original de *backpropagation*, e que, com o aumento de λ, aumenta-se a ênfase que deve ser dada na minimização do termo de penalidade escolhido.

Rosen (1996) usou o seguinte termo de penalidade:

$$R_i = \sum_{j=1}^{i-1} c(j,i) p_i, \tag{2.6}$$

onde $c(j,i)$ é uma função de indicação que especifica quais redes i e j devem ser descorrelacionadas entre si e p_i é o produto das polarizações das i-ésima e j-ésima redes, dado pela Expressão 2.7.

$$p_i = (f_i - d)(f_j - d), \tag{2.7}$$

onde f_i é a saída da rede i, f_j é a saída da rede j e d é a saída desejada.

Liu (1998) propôs uma extensão para o trabalho de Rosen (1996), que permitiu o treinamento simultâneo das redes neurais, com a metodologia conhecida por *Correlação Negativa*, onde o termo de regularização é dado por:

$$R_i = (f_i - \bar{f}) \sum_{j \neq i} (f_j - \bar{f}), \tag{2.8}$$

onde $\bar{f}$ é a saída média de todo o *ensemble* no passo anterior.

O método de Correlação Negativa foi aplicado com sucesso em vários trabalhos, como em Liu & Yao (1997), Liu (1998) e Liu & Yao (1999), superando consistentemente o desempenho de outros *ensembles*. Isto se dá pois a técnica de Correlação Negativa controla diretamente o termo de covariância, ajustando assim a diversidade do *ensemble* (Brown, 2004). No entanto, esta técnica foi concebida especificamente para tratar problemas de regressão.

Outro método nesta categoria é o *Root Quartic Negative Correlation Learning (RTQRT)*, proposto por McKay & Abbass (2001b) e McKay & Abbass (2001a), que se baseia no termo de penalidade dado pela Expressão 2.9.

$$R_i = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (f_i - d)^4}, \tag{2.9}$$

onde M é o número de componentes no *ensemble*, f_i é a saída da rede i e d é a saída desejada.

Esta técnica foi aplicada a um sistema que envolve *Programação Genética* (Koza, 1992) e apresentou performance superior às obtidas via Correlação Negativa para *ensembles* com muitos componentes. No entanto, ainda é necessário levantar quais aspectos conceituais sustentam este ganho de performance.

2.2.4.2 Métodos de Busca Exploratória

Dentre os métodos de busca exploratória, os algoritmos evolutivos exercem grande importância nas aplicações atuais. Na literatura de computação evolutiva, o termo *diversidade* possui um conceito diferente do utilizado na literatura de *ensembles* (Brown, 2004). Em computação evolutiva, a *diversidade da população* se refere à presença de indivíduos que exploram regiões distintas do espaço de busca. Dessa forma, a manutenção de uma população diversa de indivíduos permite uma exploração maior do espaço de busca e, conseqüentemente, uma maior eficiência na localização de soluções melhores. Além disso, nas aplicações em computação evolutiva baseadas na abordagem *Michigan*, após um número pré-determinado de gerações/iterações, o melhor indivíduo da população é escolhido como a solução encontrada (Michalewicz, 1996).

Apesar dessas diferenças conceituais, alguns autores exploraram os mecanismos de manutenção de diversidade já desenvolvidos em computação evolutiva junto à questão de *ensembles*. No trabalho de Yao & Liu (1998), uma população de redes neurais é evoluída e, ao final, toda a população é combinada em um *ensemble*. Para estimular a diversidade, foi utilizada a técnica de *fitness sharing*. Já no trabalho de Khare & Yao (2002), tal conceito foi estendido para problemas de classificação, utilizando-se a entropia de Kullback-Leibler (Kullback & Leibler, 1951) como métrica de diversidade a ser otimizada durante a busca.

Dentre os algoritmos evolutivos, os *sistemas imunológicos artificiais* (Dasgupta, 1999a; de Castro & Timmis, 2002b) possuem mecanismos intrínsecos capazes de manter a diversidade da população. Em de Castro et al. (2005), este paradigma de sistemas imunológicos artificiais foi usado para evoluir *ensembles* de sistemas nebulosos para classificação. No trabalho de Coelho et al. (2005), os autores verificaram que tal metodologia gera bases de regras de classificadores nebulosos que exploram regiões diversas do problema.

Neste trabalho, as características de manutenção de diversidade inerentes aos sistemas imunológicos artificiais serão exploradas, e o algoritmo imuno-inspirado *opt-aiNet* (de Castro & Timmis, 2002a), desenvolvido originalmente para problemas de otimização multimodal (problemas com mais de um ótimo global/local), será aplicado na geração de componentes de bom desempenho, e ao mesmo tempo diversos, para *ensembles* de redes neurais artificiais.

2.3 Síntese do Capítulo e Motivação para a Aplicação Pretendida

O objetivo principal deste capítulo foi permitir que o leitor se familiarizasse com o paradigma de *ensembles* e com os mecanismos de geração de diversidade, através de uma compilação das principais técnicas apresentadas na literatura até o momento.

Para isso, no início deste capítulo foram feitas uma introdução sobre os principais conceitos de

ensembles e breves comentários sobre a importância da diversidade do erro dos componentes de *ensembles*, bem como das métricas exploradas até o momento. Foi ressaltado também que, apesar da necessidade de diversidade entre os componentes ser intuitiva, ainda não se conseguiu mostrar formalmente qual a relação entre diversidade de componentes e desempenho do *ensemble*, isso para problemas de classificação cuja saída dos componentes já é o rótulo da classe.

Em seguida, foram apresentadas algumas das principais técnicas de criação de diversidade em componentes de *ensembles*. Para que tais métodos pudessem ser apresentados de forma organizada, foi seguida uma taxonomia parcialmente inspirada na proposta de Brown et al. (2005), que divide tais métodos em quatro grandes classes, baseando-se na atuação que estes têm sobre o espaço de hipóteses e na atuação sobre o conjunto de dados de treinamento e arquitetura dos componentes. São elas as classes: métodos que atuam sobre o ponto de partida no espaço de hipóteses, métodos que atuam sobre o conjunto de dados de treinamento, métodos que variam a arquitetura dos componentes e métodos que atuam na forma de exploração do espaço de hipóteses. Embora não tenham sido tratados explicitamente, combinações dessas quatro grandes classes já vêm sendo propostas na literatura, sendo denominadas aqui de métodos híbridos.

Neste trabalho foi utilizado o paradigma de sistemas imunológicos artificiais para a geração de componentes para *ensembles* que apresentam bom desempenho e, ao mesmo tempo, mantêm um certo grau de diversidade entre si (vide Seção 2.2.4.2). Esta técnica é um método de busca exploratória, que possui mecanismos de controle da diversidade dos indivíduos em sua população. Dessa forma, de maneira semelhante à dos métodos de otimização convencional, a função objetivo vai depender do erro de classificação dos componentes e da diversidade entre eles. No próximo capítulo, o paradigma de sistemas imunológicos artificiais será apresentado, bem como o algoritmo utilizado neste trabalho.

Capítulo 3

Sistemas Imunológicos Artificiais

No campo da computação evolutiva, um paradigma computacional relativamente novo, conhecido como *Sistemas Imunológicos Artificiais* (SIAs), surgiu a partir de tentativas de modelagem e aplicação de princípios imunológicos na resolução de problemas de diversas áreas, tais como otimização, mineração de dados, segurança computacional e robótica (Dasgupta, 1999a; de Castro & Timmis, 2002b).

Quando comparados com as estratégias evolutivas clássicas, os sistemas imunológicos artificiais apresentam algumas vantagens, tais como:

- Os algoritmos baseados neste paradigma são inerentemente capazes de manter a diversidade da população, uma vez que possuem módulos que desempenham funções semelhantes às técnicas de *niching* e *fitness sharing*;

- O tamanho da população nestes algoritmos é dinamicamente ajustado de acordo com a demanda da aplicação;

- As soluções ótimas locais tendem a ser simultaneamente preservadas quando localizadas.

Neste trabalho, tais aspectos são explorados através da utilização do algoritmo imuno-inspirado *opt-aiNet* (de Castro & Timmis, 2002a) para a geração de um conjunto de redes neurais classificadoras de bom desempenho e, ao mesmo tempo, diversas, para serem utilizadas na composição de ensembles.

Na primeira parte deste capítulo, será feita uma breve descrição do funcionamento do Sistema Imunológico Natural. Em seguida, os Sistemas Imunológicos Artificiais serão formalizados, e o algoritmo *opt-aiNet*, aplicado neste trabalho, será descrito.

3.1 O Sistema Imunológico Natural

O sistema imunológico natural (SI) pode ser considerado um dos mais importantes componentes dos organismos vivos superiores, e seus mecanismos de reconhecimento e combate a agentes infecciosos externos (*patógenos*) têm o objetivo de manter tais organismos saudáveis. Os padrões moleculares presentes nesses patógenos invasores (padrões estes denominados *antígenos*) são responsáveis pelo disparo da resposta imunológica, a qual promove mecanismos de reconhecimento de padrões empregando células do sistema imunológico.

Esse sistema de defesa pode ser subdividido em duas linhas: o sistema imunológico *inato* e o sistema imunológico *adaptativo*. As células do sistema inato são capazes de responder a uma ampla gama de agentes invasores sem a necessidade de uma exposição prévia a eles, diferentemente das células do sistema adaptativo (responsáveis pela liberação dos chamados *anticorpos*), que são produzidas em resposta a infecções específicas. Dessa forma, a presença de anticorpos em um dado indivíduo reflete as infecções às quais ele já foi exposto.

O sistema imunológico adaptativo é constituído de um grande número de componentes, dentre os quais os *linfócitos* merecem atenção especial. Existem dois tipos principais de linfócitos: os *linfócitos B* (ou células B) e os *linfócitos T* (ou células T). Estes dois tipos de linfócitos possuem em sua superfície receptores de antígenos com alta especificidade. Tais células participam da resposta imunológica adaptativa e são responsáveis pelo *reconhecimento* e *eliminação* de agentes causadores de doenças (patógenos), e pela chamada *memória imunológica*. A *memória imunológica* corresponde à capacidade das células do sistema adaptativo de reconhecerem um mesmo antígeno (ou um antígeno semelhante) quando houver uma infecção reincidente por parte de um dado patógeno, o que conseqüentemente leva a uma resposta imunológica mais rápida, podendo até mesmo evitar o reestabelecimento da doença no organismo. Com isso, a resposta adaptativa dá ao sistema imunológico a capacidade de aprender e se aprimorar diante de cada infecção sofrida.

Enquanto a resposta adaptativa resulta em imunidade contra re-infecções de um mesmo agente infeccioso, a resposta inata não altera significativamente o seu comportamento com base no histórico de exposição a antígenos. Juntos, os sistemas imunológicos inato e adaptativo contribuem para um mecanismo de defesa extremamente eficiente, que opera em paralelo, recorrendo a uma diversidade de agentes e componentes distribuídos espacialmente e operando em rede.

Apesar da importância do sistema imunológico inato no sistema imunológico natural como um todo, o restante deste capítulo será focado apenas nos mecanismos responsáveis pelo funcionamento da resposta imunológica adaptativa, que é a principal fonte de inspiração para os sistemas imunológicos artificiais. No entanto, salienta-se que há uma forte interação dos sistemas inato e adaptativo em organismos vivos, com evidências claras de que os iniciadores da resposta adaptativa são componentes do sistema imune inato (de Castro & Timmis, 2002b).

3.1.1 A Resposta Imunológica Adaptativa

Para facilitar a compreensão do funcionamento da resposta imunológica adaptativa, tomemos como ponto de partida o momento em que um patógeno é identificado. Neste instante, este patógeno é "engolido" (*fagocitose*) por um grupo de células chamadas *macrófagos* que estão em circulação pelo organismo, e tem sua estrutura fragmentada. Cada um destes macrófagos passa então a exibir em sua superfície peptídeos característicos desse patógeno (os chamados *antígenos*), resultantes do processo de fragmentação, que permitem que as demais células do sistema imunológico percebam a necessidade de reação ao agente invasor (Figura 3.1, passo I). Estes macrófagos se deslocam então para os chamados *linfonodos* (que são nódulos espalhados pelo organismo, com alta concentração de células do sistema imunológico), onde têm os antígenos em sua superfície reconhecidos pelos receptores das chamadas *células T ajudantes* (ou *células T de defesa*) (Figura 3.1, passo II).

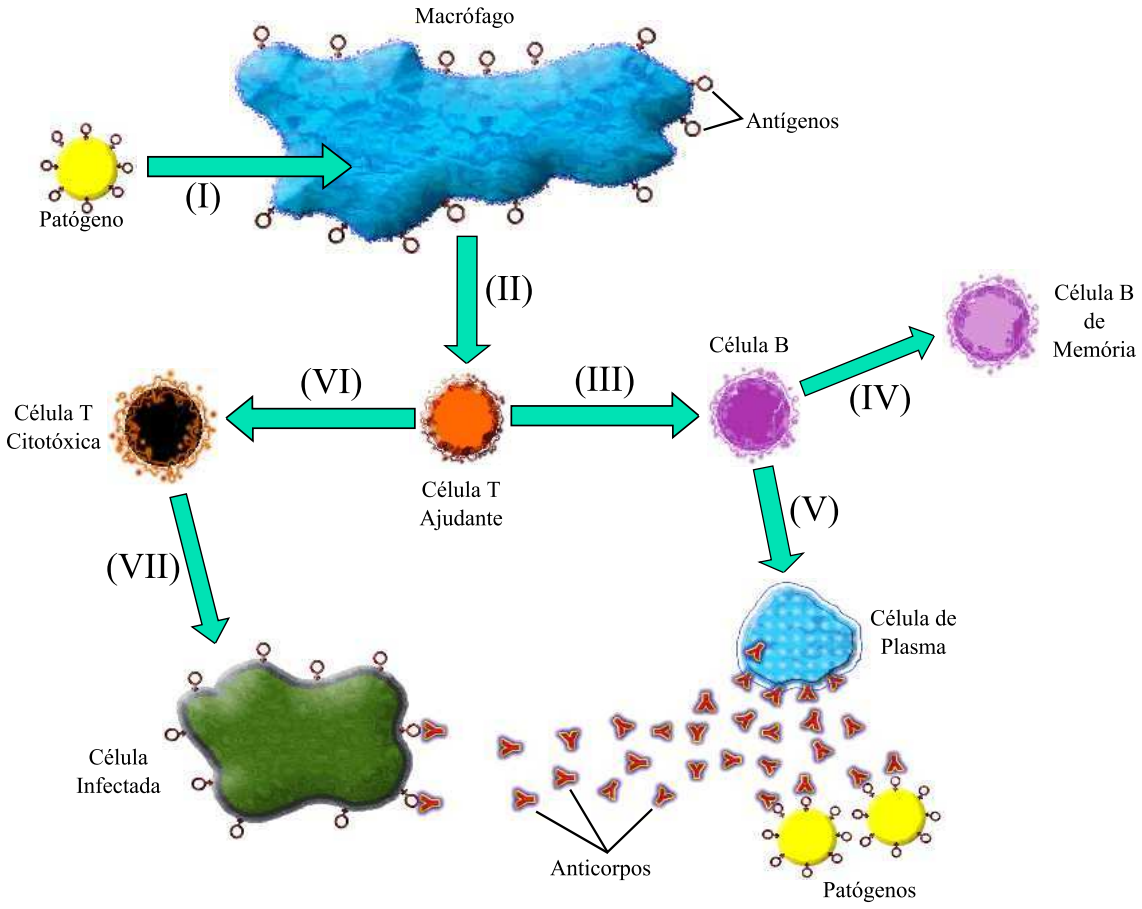


Fig. 3.1: Resposta Imunológica Adaptativa: (I) Fagocitose e quebra do patógeno por macrófagos; (II) reconhecimento dos antígenos pelas células T ajudantes; (III) estímulo da produção de células B; (IV) diferenciação das células B em células de memória; (V) diferenciação das células B em células de plasma; (VI) estímulo da produção de células T citotóxicas; e (VII) eliminação de células do próprio organismo infectadas pelo patógeno. Esta figura ilustra uma situação hipotética em que o patógeno possui apenas um tipo de antígeno.

Essas células T ajudantes, por sua vez, estimulam a reprodução das *células B* (Figura 3.1, passo III) e a diferenciação destas em *células B de memória* (Figura 3.1, passo IV) e *células de plasma* (Figura 3.1, passo V). As células B de memória garantem uma resposta mais rápida a patógenos com antígenos similares que possam vir a invadir o organismo no futuro, enquanto as células de plasma são as principais células secretoras de *anticorpos* do organismo. Anticorpos são moléculas que se ligam aos antígenos presentes tanto nos patógenos quanto nas células infectadas e sinalizam para os demais componentes do sistema imunológico que estes indivíduos devem ser eliminados. Tanto os anticorpos quanto os antígenos possuem composições físico-químicas bem definidas, de forma que, quanto melhor for a complementariedade de suas geometrias, melhor será a qualidade da ligação entre eles (maior *afinidade* - vide Seção 3.2.1).

Além de estimular a reprodução das células B, as células T ajudantes também são responsáveis por estimular a produção das chamadas *células T citotóxicas* (Figura 3.1, passo VI), que são responsáveis pela eliminação de células do próprio organismo que já foram infectadas pelo patógeno (Figura 3.1, passo VII).

O mecanismo de resposta imunológica adaptativa representado na Figura 3.1 ilustra apenas uma situação hipotética em que o patógeno apresenta um único tipo de antígeno. Na verdade, um mesmo patógeno pode apresentar múltiplos antígenos, como representado na Figura 3.2. Com isso, após a fragmentação do patógeno pelos macrófagos, para cada antígeno diferente apresentado na superfície dessas células, uma resposta imunológica como a apresentada na Figura 3.1 pode ser disparada.

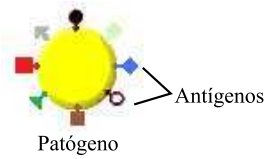


Fig. 3.2: Ilustração de um patógeno que apresenta múltiplos antígenos diferentes em sua superfície.

3.1.2 O Princípio da Seleção Clonal

Quando ocorre o estímulo para produção de células B após a detecção de um antígeno, estas células passam a se reproduzir com maior velocidade, em um processo de *clonagem*. Durante este processo, visando promover a melhora da adaptação dos clones gerados aos antígenos invasores (maior afinidade entre anticorpos e antígenos), esses novos clones sofrem *hipermutação* com taxas de variabilidade genética inversamente proporcionais à sua afinidade com os antígenos em questão (células com maior afinidade com o antígeno sofrem uma variação genética menor, e vice-versa). Dentre as células resultantes, aquelas com menor afinidade e as que porventura se tornaram prejudiciais ao organismo com a etapa de hipermutação são então eliminadas da população.

Este processo de *expansão clonal*, *hipermutação* e *seleção* das células mais adaptadas é conhecido como **Princípio da Seleção Clonal** e foi proposto por Burnet (1959). Este princípio é uma das principais inspirações do algoritmo *opt-aiNet* usado neste trabalho.

3.1.3 A Teoria da Rede Imunológica

Outra teoria imunológica com importância crucial no campo de sistemas imunológicos artificiais é a chamada **Teoria da Rede Imunológica**, proposta por Jerne (1974). Jerne propôs que o sistema imunológico não é apenas um sistema reativo que permanece em repouso até que um patógeno invada o organismo. Segundo ele, algumas partes dos receptores das células imunológicas (chamadas *idiotopos*) seriam reconhecidas pelos receptores de outras células e moléculas. Diante disso, o sistema imunológico poderia então ser visto como uma enorme e complexa rede, que permanece em um estado de equilíbrio dinâmico em que seus elementos sofrem constantemente alterações em suas concentrações. Assim, quando um antígeno altera este equilíbrio, a rede automaticamente altera as concentrações dos indivíduos em reação a este antígeno e acaba convergindo para um novo estado de equilíbrio.

Já que as células imunológicas são capazes de reconhecer células do próprio organismo, foi proposto também que existem mecanismos de *supressão* da resposta imunológica (conhecidos também como *resposta negativa da rede*) para evitar o ataque a células do próprio organismo, e mecanismos de *estímulo* para guiar a resposta imunológica (*resposta positiva da rede*) (Figura 3.3). No entanto, esses mecanismos não foram claramente descritos na teoria proposta nem foram realizados experimentos que sustentem o conjunto de hipóteses em questão. Apesar disso, muitos aspectos dessa teoria são de fácil implementação computacional e permitem reproduzir em computador algumas funcionalidades dos sistemas imunológicos, como será visto nas próximas seções.

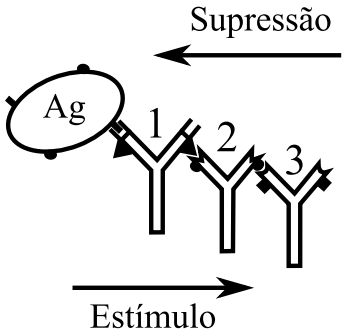


Fig. 3.3: Mecanismos de resposta positiva (estímulo) e negativa (supressão) da rede imunológica. O reconhecimento de um antígeno (Ag) pelos elementos da rede (1, 2 e 3) ativa a resposta imunológica de combate ao agente invasor, enquanto que o reconhecimento de um outro elemento do próprio organismo leva à supressão dessa resposta.

3.2 Sistemas Imunológicos Artificiais

Formalmente, um sistema imunológico artificial pode ser definido como segue (de Castro & Timmis, 2002b):

"Sistemas Imunológicos Artificiais (SIAs) são sistemas adaptativos, inspirados pela imunologia teórica e observações de funções, princípios e modelos imunológicos, que são aplicados à resolução de problemas."

De acordo com esta definição, para um sistema ser caracterizado como um SIA, ele deve no mínimo englobar um modelo de um componente do SI (por exemplo, uma célula, molécula ou órgão), deve ser desenvolvido através da incorporação de idéias da imunologia teórica e/ou experimental, e deve ter o objetivo de resolver problemas. Dessa forma, apenas atribuir uma terminologia imunológica a um dado sistema não é suficiente para caracterizá-lo como um SIA (de Castro & Timmis, 2002b).

À primeira vista, o sistema imunológico pode ser interpretado apenas como um sistema de reconhecimento de padrões, responsável pela identificação e eliminação de elementos perigosos ao organismo. Isso limitaria o escopo de sistemas imunológicos artificiais a tarefas de reconhecimento de padrões, particularmente aplicadas à segurança computacional. No entanto, a grande quantidade de características computacionalmente interessantes presentes no sistema imunológico natural sugerem uma gama de aplicações praticamente ilimitada (de Castro & Timmis, 2002b). Características como o aprendizado e a memória imunológica, auto-organização, capacidade de detecção de anomalias, tolerância a falhas, distributividade e robustez presentes no sistema imunológico natural são indicadores de que os SIAs representam um paradigma que pode ser aplicado em diversas áreas, tanto em inteligência computacional quanto em engenharia e ciência da computação. Apenas para exemplificar, existem aplicações de SIAs em campos como:

- *Reconhecimento de Padrões*: Dasgupta et al. (1999), Carter (2000), Tarakanov et al. (2000), Carvalho & Freitas (2001);
- *Segurança Computacional*: Kephart (1994), Somayagi et al. (1998), Dasgupta (1999b), Kim & Bentley (1999), Hofmeyr & Forrest (2000);
- *Detecção de Anomalias*: Aisu & Mitzutani (1996), Dasgupta (1996), McCoy & Devarajan (1997), Dasgupta & Forrest (1999);
- *Otimização*: Bersini & Varela (1990), Mori et al. (1993), Hajela & Yoo (1999), Toma et al. (1999), Gaspar & Collard (2000), de Castro & Timmis (2002a), Gomes et al. (2003), de França et al. (2005);

- *Aprendizagem de Máquina*: Hunt & Cooke (1996), Bersini (1999), de Castro & Von Zuben (2000), Timmis et al. (2000), Watkins (2001);
- *Robótica*: Mitsumoto et al. (1996), Lee & Sim (1997), Ishiguro et al. (1998), Michelan & Von Zuben (2002), Cazangi & Von Zuben (2006);
- *Controle*: Bersini (1991), Takahashi & Yamada (1997), Ootsuki & Sekiguchi (1999);
- *Memória Associativa*: Gibert & Routen (1994), Abbattista et al. (1996);
- *Ecologia*: Bersini & Varela (1994), Walker (2001);
- *Síntese Sonora*: Caetano (2006).

Neste trabalho, foi utilizado o algoritmo *opt-aiNet* (de Castro & Timmis, 2002a), que é um sistema imunológico artificial baseado nos princípios de Seleção Clonal e Teoria da Rede Imunológica, descritos nas Seções 3.1.2 e 3.1.3, respectivamente, e desenvolvido originalmente para a solução de problemas de otimização *multimodal*, ou seja, em que se busca identificar múltiplos ótimos globais. A família de algoritmos à qual o algoritmo *opt-aiNet* pertence, bem como o algoritmo em si, serão descritos nas Subseções 3.2.2 e 3.2.3, respectivamente.

3.2.1 Espaço de Formas

Um dos pontos principais em sistemas imunológicos artificiais é a questão da *afinidade*, seja entre um anticorpo e um antígeno ou entre dois anticorpos. Com o objetivo de descrever quantitativamente as interações de moléculas do sistema imunológico, Perelson & Oster (1979) introduziram o conceito de *espaço de formas* a partir de estudos teóricos sobre os mecanismos de seleção clonal.

A afinidade entre um anticorpo e um antígeno (tudo o que for mencionado nesta seção para o caso anticorpo/antígeno, a não ser que seja dito o contrário, é válido para o caso anticorpo/anticorpo) envolve vários processos, tais como interações covalentes baseadas em cargas eletrostáticas, pontes de hidrogênio, interações de van der Waals, etc. Diante disso, para que um antígeno seja reconhecido por um anticorpo, é necessário que ambas as moléculas se liguem de maneira *complementar* entre si, por longos trechos de sua superfície, como representado pictoricamente na Figura 3.4. Define-se então como *afinidade* entre um anticorpo e um antígeno esse grau de complementariedade que as duas moléculas apresentam entre si. Caso um anticorpo e um antígeno não sejam tão complementares entre si, mesmo assim eles ainda poderão se ligar, embora com menor afinidade.

A forma e a distribuição de cargas elétricas, assim como a existência de grupos químicos em posições complementares na superfície dos antígenos e anticorpos, podem ser consideradas como

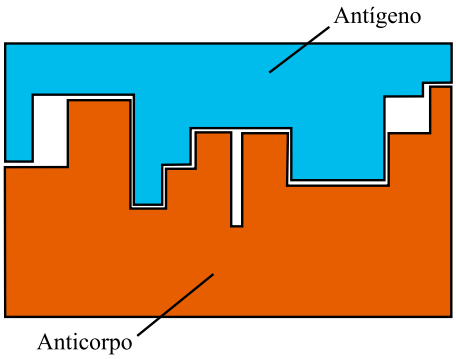


Fig. 3.4: Complementariedade geométrica parcial das formas de anticorpos e antígenos. Quanto melhor for esta complementariedade, maior será a afinidade entre eles. Esta figura apresenta uma visão pictórica de interações físico-químicas de compostos moleculares.

propriedades dessas moléculas e são importantes na determinação de suas interações. A este conjunto de atributos dá-se o nome de *forma generalizada*. Supondo que a forma generalizada de um anticorpo ou antígeno possa ser descrita por um conjunto de L parâmetros, então o espaço L-dimensional, dentro do qual essa forma generalizada corresponde a um ponto, é chamado de *espaço de formas*. Do ponto de vista computacional, a equivalência biológica desses atributos da forma generalizada de anticorpos não é relevante, uma vez que tais atributos serão basicamente definidos pelo domínio da aplicação do sistema imunológico artificial.

Matematicamente, a forma generalizada de qualquer molécula $\mathbf{m}$ em um espaço de formas S pode ser representada como um vetor de atributos de tamanho L, dado por $\mathbf{m} = [m_1,\ m_2\ \ldots\ , m_L]$. Esse vetor pode ser composto por atributos de qualquer tipo, tais como valores reais, inteiros, bits e símbolos. Geralmente, tais atributos são definidos de acordo com o domínio do problema sendo tratado pelo sistema imunológico artificial, e são importantes na definição de quais métricas serão usadas para medir suas interações. Além disso, o tipo dos atributos definirá o tipo do espaço de formas a ser adotado, que pode ser:

- *Espaço de Formas Real*: onde os vetores de atributos são vetores de números reais;
- *Espaço de Formas Inteiro*: onde os vetores de atributos são vetores de números inteiros;
- *Espaço de Formas de Hamming*: onde os vetores de atributos são compostos por elementos pertencentes a um alfabeto finito de tamanho k, e
- *Espaço de Formas Simbólico*: onde os vetores de atributos são geralmente compostos por diferentes tipos de atributos, onde pelo menos um deles é simbólico, como por exemplo um *nome*, uma *cor*, etc.

Dessa forma, se considerarmos, sem perda de generalidade, que as formas generalizadas de um anticorpo e de um antígeno são de mesmo tamanho (o que no caso biológico não ocorre necessariamente), então sob a perspectiva de reconhecimento de padrões, a interação destes dois indivíduos, ou sua *medida de afinidade*, é avaliada a partir de alguma métrica de *distância*, definida no espaço de formas a que pertencem os correspondentes vetores de atributos. Esta medida de afinidade realiza então um mapeamento da interação do anticorpo com o antígeno em um número real não negativo que corresponde à *afinidade* ou *grau de similaridade*. Dessa forma, a afinidade entre anticorpos e antígenos, sendo proporcional ao grau de similaridade entre seus vetores de atributos, não trabalha com o conceito de complementariedade entre moléculas, embora seja possível definir uma relação direta entre ambos.

A expressão "afinidade" é normalmente adotada para quantificar o grau de reconhecimento entre um anticorpo e um antígeno. No entanto, é possível ver a afinidade como um termo geral que indica a qualidade de um elemento do sistema imunológico em relação ao ambiente em que está inserido (de Castro & Timmis, 2002b). Por exemplo, se um sistema imunológico está sendo aplicado a um problema de otimização de funções, então o anticorpo pode corresponder a um ponto relacionado a um dado valor da função objetivo. Dessa forma, sua afinidade corresponderá ao valor da função sendo otimizada para este ponto, o que equivale a um valor de *fitness* em um algoritmo evolutivo.

Neste trabalho, as duas interpretações de afinidade estarão presentes. No algoritmo imuno-inspirado utilizado para treinamento das redes neurais artificiais (*opt-aiNet*), estas redes corresponderão aos anticorpos do SIA e, dessa forma, possuirão uma métrica de *fitness* inversamente proporcional ao erro de classificação cometido por cada rede neural. Além disso, para controle da diversidade da população, será adotada também uma métrica de afinidade entre anticorpos (entre as redes neurais), ou seja, será adotada também uma métrica de afinidade que indica o grau de reconhecimento entre dois indivíduos da população.

Nas próximas seções, serão apresentados a família de algoritmos à qual o algoritmo *opt-aiNet* utilizado neste trabalho pertence e, em seguida, o próprio algoritmo *opt-aiNet*.

3.2.2 A Família de Algoritmos aiNet

A família de algoritmos imuno-inspirados *aiNet* (*Artificial Immune Network*) surgiu em 2000 com o algoritmo de mesmo nome, apresentado e detalhado em de Castro & Von Zuben (2000) e de Castro & Von Zuben (2001). Baseado no princípio da Seleção Clonal e na teoria da Rede Imunológica, o algoritmo aiNet foi proposto para análise e agrupamento de dados (*clustering*). Em um trabalho subseqüente, de Castro & Timmis (2002a) desenvolveram a extensão do algoritmo aiNet para problemas de otimização multimodal e a denominaram *opt-aiNet* (*Artificial Immune Network for Optimization*). O terceiro integrante desta família de algoritmos, a *copt-aiNet* (*Artificial Immune Network for Com-*

binatorial Optimization), foi proposta por Gomes et al. (2003) para otimização combinatória e, em seguida, de França et al. (2005) propuseram uma extensão da *opt-aiNet* para problemas de otimização dinâmica (funções-objetivo variantes no tempo), a qual chamaram de *dopt-aiNet* (*Artificial Immune Network for Dynamic Optimization*). O mais recente algoritmo deste grupo, até o momento, foi proposto por Coelho & Von Zuben (2006), denomina-se *omni-aiNet* (*Artificial Immune Network for Omni Optimization*) e foi desenvolvido para problemas de omni-otimização, ou seja, é um algoritmo que busca resolver problemas com um ou mais objetivos e com um ou mais ótimos globais, sem que seja necessária nenhuma informação prévia sobre o tipo de problema sendo tratado.

Em todos os trabalhos, os autores avaliaram empiricamente a capacidade dos algoritmos propostos para os respectivos tipos de problemas a que se aplicam, através da obtenção de resultados competitivos quando comparados com outras abordagens da literatura.

3.2.3 O Algoritmo opt-aiNet

Para que o algoritmo *opt-aiNet* (de Castro & Timmis, 2002a) possa ser apresentado, vamos considerar a seguinte terminologia:

- *Anticorpo*: cada indivíduo na população. Tanto no algoritmo original quanto na versão utilizada neste trabalho, os anticorpos são codificados em vetores de atributos reais (vide Seção 3.2.1);
- *Antígeno*: conjunto de dados utilizado no treinamento;
- *Fitness ou Afinidade Anticorpo/Antígeno*: medida que indica o quão adaptado está um dado anticorpo da população para detectar um antígeno (segunda interpretação de afinidade, descrita na Seção 3.2.1), ;
- *Afinidade Anticorpo/Anticorpo*: medida de similaridade entre dois anticorpos da população. Esta medida é utilizada na determinação da supressão de indivíduos.

Feitas estas definições, podemos então analisar o funcionamento do algoritmo. A Figura 3.5 mostra o fluxograma com os principais passos do *opt-aiNet*.

O algoritmo começa com a geração aleatória da população inicial de anticorpos e com a verificação do critério de parada, que neste trabalho corresponde a um certo número de gerações, definido pelo usuário. Caso a condição de parada tenha sido satisfeita, o algoritmo encerra sua execução. Caso contrário, o *fitness* de cada indivíduo na população é determinado e, para cada indivíduo na população, um número N_c de clones é gerado (onde N_c é um número fixo para todos os indivíduos na população e definido pelo usuário).

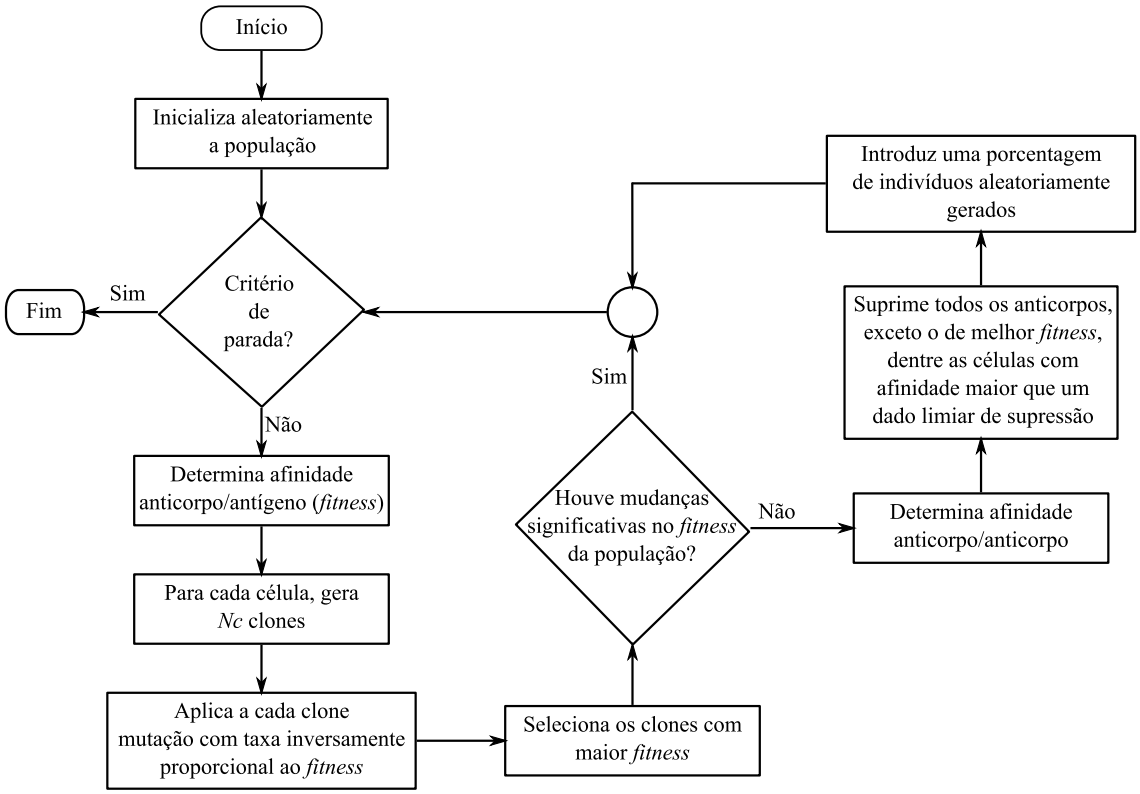


Fig. 3.5: Fluxograma com os principais passos do algoritmo *opt-aiNet*.

Cada um destes clones sofre então um processo de hipermutação com taxa de variabilidade genética inversamente proporcional ao *fitness* do anticorpo que gerou cada clone, ou seja, quanto melhor for o anticorpo gerador dos clones, menor será a variabilidade genética aplicada a estes clones. Esta etapa de hipermutação é feita de acordo com a Expressão 3.1 abaixo:

$$c' = c + \frac{1}{\beta} e^{(-f^*)} N(0,1), \tag{3.1}$$

onde *c'* é o novo anticorpo gerado a partir de *c*, *N*(0,1) é uma variável aleatória gaussiana de média zero e variância 1, f^* é o *fitness* do indivíduo c e β é um parâmetro de controle da amplitude da função exponencial inversa. Este parâmetro β assume os valores $1, 10, 100, 1000$ sucessivamente, de acordo com o número de gerações executadas pelo algoritmo até o momento (a cada 25 gerações do algoritmo, β tem seu valor multiplicado ou dividido por 10, de forma a seguir a seqüência $1 \rightarrow 10 \rightarrow 100 \rightarrow 1000 \rightarrow 100 \rightarrow 10 \rightarrow 1 \rightarrow 10 \ldots$). Dessa forma, o algoritmo alterna etapas de ajuste fino dos indivíduos da população (valores altos de β) com etapas de maior variabilidade genética dos indivíduos (valores baixos de β).

Aplicada a mutação, são então selecionados os N anticorpos de maior *fitness* (considerando-se tanto os indivíduos originais quanto os clones gerados) para permanecerem na população, onde N é

o número de indivíduos no início dessa iteração do algoritmo, e o *fitness* médio dessa população é calculado. Se houve uma melhora nesse *fitness* médio, o algoritmo executa novamente o seu ciclo principal, voltando à etapa de verificação das condições de parada. Caso não tenha ocorrido uma variação significativa neste *fitness* médio (o que indica que a população está estagnada, provavelmente em um ótimo local), a afinidade entre os anticorpos na população é determinada e estes elementos são comparados dois a dois. Caso a afinidade entre dois anticorpos esteja acima de um dado limiar de supressão definido pelo usuário (ou seja, eles são muito semelhantes), o anticorpo de pior *fitness* é eliminado da população. Concluída esta etapa de supressão de anticorpos semelhantes, é introduzida na população uma porcentagem d de novos indivíduos aleatoriamente gerados, o que contribui para o aumento da diversidade, e a execução retorna para a etapa de verificação das condições de parada.

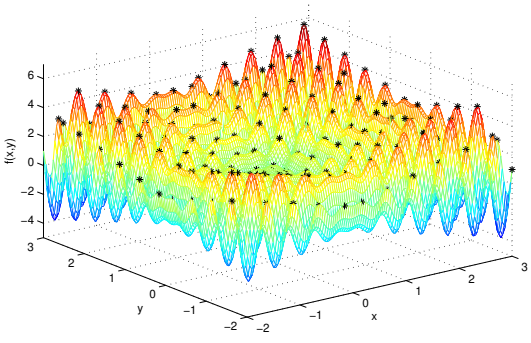
É importante ressaltar aqui que, no algoritmo *opt-aiNet*, o tamanho da população em cada iteração é automaticamente definido de acordo com a demanda do problema (dado um limiar de supressão), ou seja, os mecanismos de supressão e de introdução de diversidade automaticamente eliminam indivíduos semelhantes da população, mas preservando os ótimos locais e globais já identificados. Esta característica de manutenção de diversidade da população, inerente ao algoritmo *opt-aiNet*, foi o principal motivo que levou à sua aplicação na geração de componentes para *ensembles* neste trabalho.

Apenas para ilustrar a capacidade do algoritmo de encontrar e preservar os múltiplos ótimos globais/locais de um problema, o *opt-aiNet* foi aplicado a três problemas de otimização bidimensionais (problemas de maximização) que apresentam multimodalidade. A escolha de apenas duas variáveis de otimização, ou seja, de espaços de busca bidimensionais, se deu aqui para permitir a visualização dos resultados. As superfícies de otimização estão associadas às funções *Multi*, *Roots* e de *Schaffer*, dadas pelas Equações 3.2, 3.3 e 3.4, respectivamente. Os resultados estão apresentados na Figura 3.6.

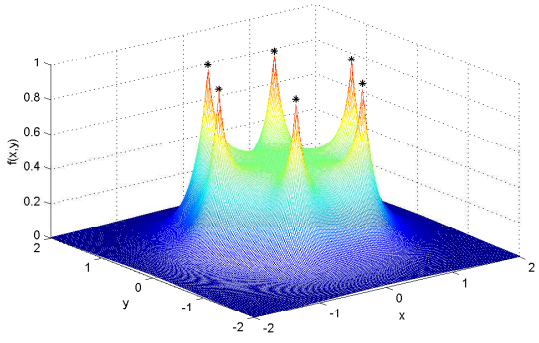
$$f(x,y) = x.\sin(4\pi x) - y.\sin(4\pi y + \pi) \quad x,y \in [-2,3] \tag{3.2}$$

$$f(z) = \frac{1}{1+|z^6-1|}, \quad z = x + \mathbf{i}y, \quad x,y \in [-2,2] \tag{3.3}$$

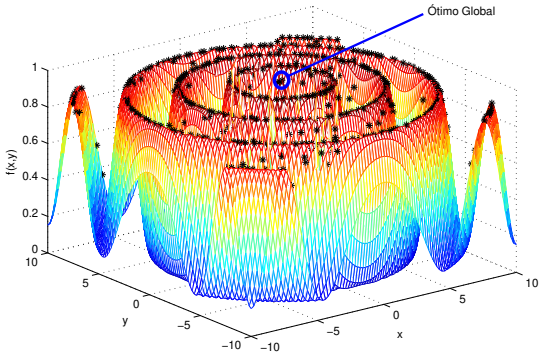
$$f(x,y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{1+0.001(x^2+y^2)}, \quad x,y \in [-10,10] \tag{3.4}$$



(a) Função *Multi*



(b) Função *Roots*



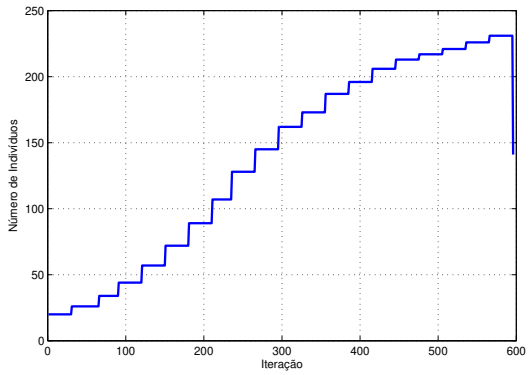
(c) Função *Schaffer*

Fig. 3.6: Resultados da aplicação do algoritmo *opt-aiNet* a três problemas bidimensionais e com múltiplos ótimos: (a) função *Multi*, (b) função *Roots* e (c) função de *Schaffer* (a população final corresponde aos pontos em *).

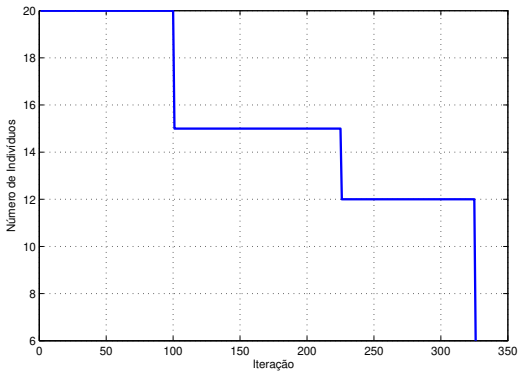
Como pode ser observado na Figura 3.6, o algoritmo *opt-aiNet* foi capaz de identificar ótimos locais e, ao mesmo tempo, manter a diversidade destas soluções. Nas funções *Multi* e *Roots*, todos os ótimos globais foram encontrados e, no caso da função *Multi*, o algoritmo manteve uma solução em cada ótimo local do problema. Já para a função de *Schaffer*, que possui um único ótimo global na região central e infinitos ótimos locais, o algoritmo foi capaz de encontrar este ótimo global e povoar com uma distribuição aproximadamente uniforme as regiões correspondentes aos infinitos ótimos locais. Como já mencionado (e pode ser verificado na Figura 3.7), o tamanho da população varia automaticamente ao longo do processo de otimização e converge para valores que dependem da demanda de cada problema.

É interessante notar que, no final de sua execução, o algoritmo *opt-aiNet* executa uma etapa de higienização das soluções presentes em sua população final, de forma a eliminar eventuais redundâncias que ainda permaneçam e aqueles indivíduos que não convergiram para algum ótimo local ou global do problema, o que provoca a redução brusca no número de indivíduos, ilustrada na Figura

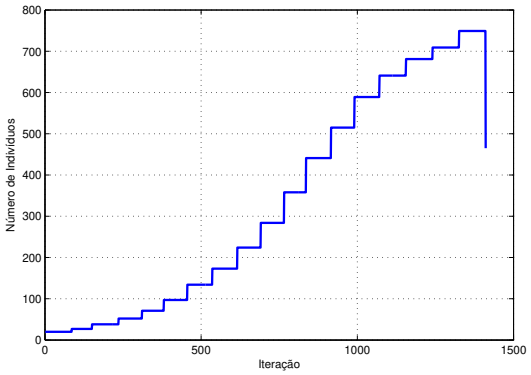
3.7, na última iteração do algoritmo para cada um dos problemas tratados.



(a) Função *Multi*



(b) Função *Roots*



(c) Função *Schaffer*

Fig. 3.7: Evolução do número de indivíduos (anticorpos) na população do algoritmo *opt-aiNet* ao longo das iterações, para os problemas (a) *Multi*, (b) *Roots* e (c) *Schaffer*. O algoritmo foi inicializado com 20 indivíduos para os três problemas.

3.3 Síntese do Capítulo e Motivação para a Aplicação Pretendida

Neste capítulo, o funcionamento básico do sistema imunológico natural foi apresentado, juntamente com as principais teorias e princípios presentes nesta área e relevantes ao contexto deste trabalho. Em seguida, o paradigma de sistemas imunológicos artificiais foi formalizado e alguns exemplos de aplicações deste paradigma nas mais diversas áreas foram citados.

Dentre os sistemas imunológicos artificiais, a família de algoritmos imuno-inspirados conhecida como *aiNet* possui aplicações em áreas como agrupamento de dados e otimização multimodal, combinatória, dinâmica e multiobjetivo, e seus algoritmos apresentam características interessantes, como capacidade de manutenção da diversidade dos indivíduos em suas populações e de definição automá-

tica (dinâmica) do tamanho de suas populações, ou seja, o número de indivíduos se ajusta de acordo com a demanda da aplicação.

Tais características, em especial a capacidade de manutenção de diversidade inerente a estes algoritmos, são de grande importância para a implementação de *ensembles*, já que, para um *ensemble* apresentar boa performance, é necessário que seus componentes apresentem boa performance e sejam, ao mesmo tempo, diversos em relação ao erro (vide Capítulo 2). Com isso, visando explorar essa manutenção de diversidade dos algoritmos imuno-inspirados da família *aiNet*, neste trabalho a versão para otimização multimodal *opt-aiNet* será aplicada à geração de uma população de classificadores (redes neurais artificiais do tipo *multi layer perceptron - MLP*) de boa performance e, ao mesmo tempo, diversos, para formarem um ensemble.

No próximo capítulo, será feita uma breve introdução às redes neurais do tipo MLP. Em seguida, as redes neurais classificadoras utilizadas neste trabalho serão descritas, a proposta de aplicação do algoritmo *opt-aiNet* para geração de componentes para *ensembles* será devidamente formalizada e as técnicas aplicadas nas etapas de seleção e combinação de componentes serão detalhadas.

Capítulo 4

Construção de Ensembles de Classificadores

Como mencionado anteriormente, a construção de um *ensemble* pode ser dividida em três etapas: geração, seleção e combinação de componentes. Para a etapa de geração de componentes, neste trabalho foi utilizado o algoritmo imuno-inspirado *opt-aiNet* para a geração de uma população de redes neurais artificiais do tipo MLP (*multi-layer perceptrons*) voltadas para a solução de problemas de classificação, cujos elementos tenham bom desempenho individual e, ao mesmo tempo, apresentem diversidade em relação aos seus erros.

Vencida a etapa de geração de uma população de candidatos para os *ensembles*, foram aplicadas seis técnicas de seleção de componentes: *Construtiva sem Exploração*, *Construtiva com Exploração*, *Poda sem Exploração*, *Poda com Exploração*, o algoritmo de agrupamento conhecido como *ARIA* e, para fins de comparação, a técnica *Sem Seleção*, onde todos os candidatos são considerados componentes do *ensemble*.

Por fim, para a combinação das saídas de cada componente em uma única saída, correspondente à saída do *ensemble* em si, foram utilizadas neste trabalho cinco técnicas de combinação de componentes: *Média Simples*, *Média Ponderada sem Bias*, *Média Ponderada com Bias*, *Voto Majoritário* e *Winner-takes-all*.

O objetivo deste capítulo é detalhar os algoritmos e técnicas empregadas em cada uma das etapas de construção de um *ensemble*. Para isso, foi feita uma divisão em cinco seções principais. Na Seção 4.1, será feita uma breve introdução às redes neurais artificiais e, em seguida, serão detalhadas as redes neurais utilizadas neste trabalho. Na Seção 4.2, serão apresentadas as principais modificações do algoritmo *opt-aiNet* feitas neste trabalho, para geração de uma população de redes neurais de bom desempenho e, ao mesmo tempo, diversas. As seis técnicas de seleção de componentes empregadas neste trabalho estão descritas na Seção 4.3, e as técnicas de combinação de componentes na Seção 4.4. Por fim, uma tabela com o resumo das siglas adotadas nos próximos capítulos é apresentada na Seção 4.5.

4.1 Redes Neurais Artificiais

Apesar de ser possível utilizar vários outros tipos de componentes em *ensembles*, neste trabalho nos restringimos ao uso de redes neurais artificiais do tipo *multi-layer perceptron* (ou perceptron multicamadas), ou simplesmente redes MLP (Haykin, 1999). Além de sua simplicidade de operação, é sabido que essas redes são *aproximadores universais de funções* (Hornik et al., 1989), ou seja, as redes MLP são capazes de aproximar qualquer função contínua, restrita a uma região compacta do espaço de aproximação, dado um número suficiente de neurônios na camada intermediária.

Os elementos básicos de uma rede neural MLP são unidades de processamento de informação chamadas de *neurônios artificiais*, que na verdade são modelos matemáticos de neurônios biológicos. A Figura 4.1 ilustra o modelo de um neurônio. Nela, podemos identificar três elementos básicos:

- Um conjunto de *sinapses* ou *ligações*, sendo cada uma delas caracterizada por um peso próprio que multiplica o sinal de entrada correspondente;
- Um *combinador linear*, responsável por somar todas as entradas ponderadas pelos pesos sinápticos; e
- Uma *função de ativação*, para limitar a amplitude da saída do neurônio.

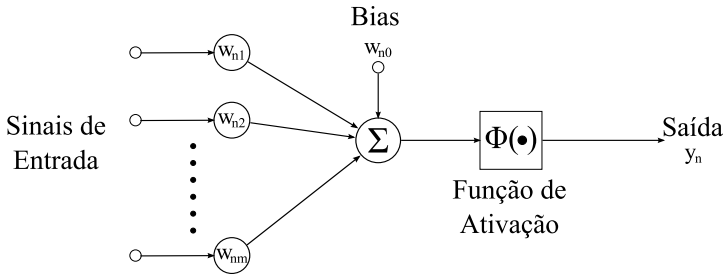


Fig. 4.1: Modelo de um neurônio de índice n: pesos sinápticos w_{nj}, $j = 1, \ldots, m$, combinador linear (Σ) e função de ativação (ϕ).

O modelo representado na Figura 4.1 apresenta ainda o chamado *bias* (ou polarização), que pode ser associado a uma entrada constante. Diante disso, a saída de um neurônio de índice n, representado na Figura 4.1, é dada pela Expressão 4.1:

$$y_n = \phi(\sum_{j=1}^{m} w_{nj}x_j + w_{n0}) \tag{4.1}$$

A *função de ativação* $\phi(.)$ pode assumir várias formas, mas a mais comum em redes neurais é a de uma *função sigmóide*, que possui o formato de "s". Neste trabalho, a função de ativação adotada foi a função *tangente hiperbólica*, dada pela expressão $\phi(.) = \tanh(.)$.

Feita essa definição do modelo básico de um neurônio artificial, uma rede neural artificial do tipo MLP pode ser definida como um grafo direcionado, ordenado em camadas, onde cada nó corresponde a um neurônio artificial, como o da Figura 4.1. A Figura 4.2 ilustra uma rede do tipo MLP. As camadas dessas redes são denominadas, respectivamente, *camada de entrada*, *camada oculta* (ou *camada intermediária*) e *camada de saída*. Uma dada camada é dita *oculta* se não possuir conexões diretas com as entradas e saídas da rede neural. No caso da Figura 4.2, existe uma única camada oculta com n neurônios, embora possam ser implementadas redes neurais MLP com mais de uma camada oculta.

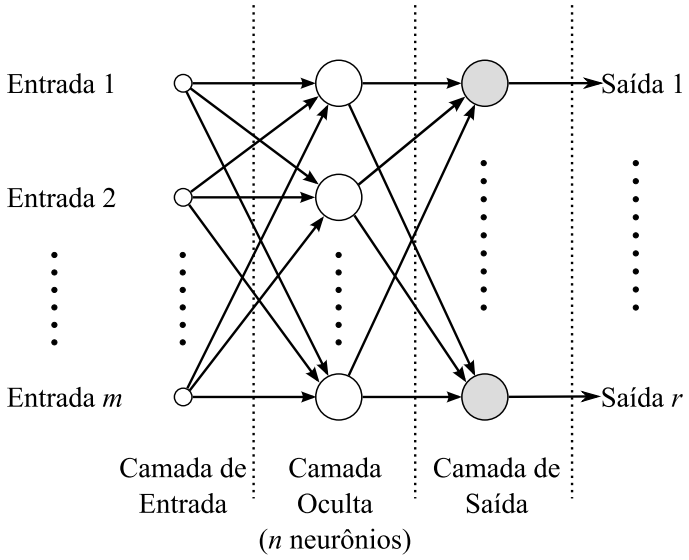


Fig. 4.2: Uma rede neural do tipo MLP (*Multi-Layer Perceptron*) com m entradas, n neurônios na camada oculta e r saídas.

Para um dado par de vetores de entrada e saída, existe um único conjunto de pesos sinápticos para toda a rede neural que minimiza o erro gerado pela rede neural ao aproximar o mapeamento desses dados de entrada nos dados de saída desejados. A obtenção desse conjunto único de pesos corresponde à etapa conhecida como *treinamento* da rede neural. Existem várias propostas para o treinamento de redes MLP, sendo a mais conhecida o algoritmo de *backpropagation* (Haykin, 1999). No entanto, esse algoritmo se baseia no gradiente da função de erro da rede neural, o que o torna altamente susceptível a mínimos locais. Numa tentativa de contornar esse problema, várias técnicas que não se baseiam em gradiente foram propostas na literatura. Dentre elas, as técnicas evolutivas apresentam um importante papel, uma vez que são capazes de realizar uma ampla exploração do espaço de busca, sendo dessa forma capazes de encontrar conjuntos de pesos não vinculados a mínimos locais ruins. Como dito anteriormente, neste trabalho utilizou-se o algoritmo imuno-inspirado *opt-aiNet* para o treinamento das redes neurais MLP, o que será melhor descrito na Seção 4.2.

4.1.1 Formato das Redes Neurais Utilizadas neste Trabalho

Neste trabalho, foram utilizadas redes MLP como a ilustrada na Figura 4.3, ou seja, redes totalmente conectadas, com uma camada oculta com número fixo de neurônios (definidos de acordo com o problema), e número de saídas igual ao número de classes do problema sendo tratado (por exemplo, para um problema com duas classes, cada rede neural terá duas saídas). As funções de ativação tanto dos neurônios da camada oculta quanto da camada de saída são do tipo *tangente hiperbólica*.

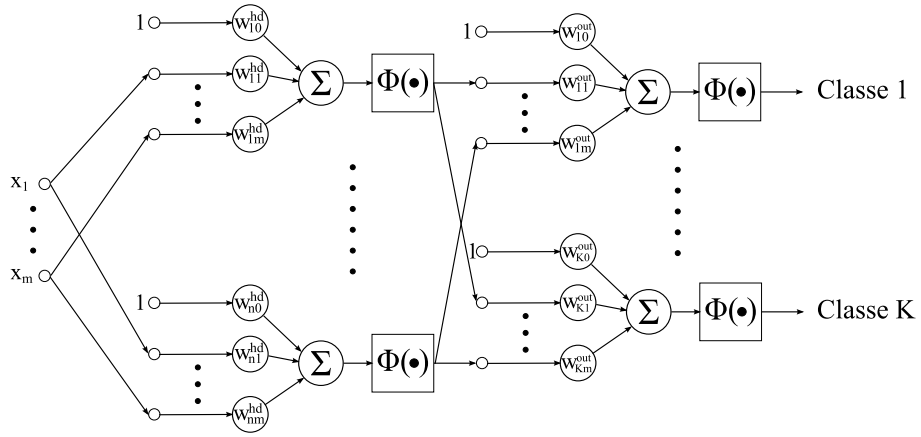


Fig. 4.3: Ilustração das redes MLP utilizadas neste trabalho. Foram adotadas redes com uma camada oculta e número de saídas igual ao número de classes do problema sendo tratado (na figura, um problema com K classes). As camadas oculta e de saída possuem funções de ativação $\Phi(.)$ do tipo tangente hiperbólica e o rótulo da classe de uma dada amostra é dado pelo índice da saída de maior valor. Os pesos sinápticos w_{ij}^{hd}, $i = 1, \ldots, n$ e $j = 1, \ldots, m$, estão associados a neurônios da camada oculta e os pesos sinápticos w_{kl}^{out}, $k = 1, \ldots, K$ e $l = 1, \ldots, n$ a neurônios da camada de saída.

O rótulo da classe, gerado pela rede neural para uma dada amostra do problema, corresponderá ao *índice da saída de maior valor*. Por exemplo, supondo que a amostra em questão pertence a um problema de três classes, e que uma dada rede neural produz para essa amostra um vetor de saída $[+0.1 \ -0.6 \ +0.5]$, como a terceira posição possui o maior valor $(+0.5)$, então o rótulo que essa rede neural atribuiria à amostra em questão seria *"Classe 3"*.

Diante disso, os conjuntos de dados utilizados no treinamento das redes neurais foram alterados, sendo os rótulos de cada amostra substituídos por vetores com número de elementos igual ao número de classes possíveis para o problema, de forma a corresponderem ao número de saídas das redes neurais. Nesses vetores, a posição correspondende à classe (rótulo) dessa amostra recebeu valor $+1$, enquanto as demais receberam valores -1 (vide exemplo na Figura 4.4 para um problema de três classes). No entanto, cabe ressaltar aqui que tais vetores de saída com valores $+1$ e -1 servem apenas para direcionar o aprendizado dos classificadores, uma vez que, para uma dada amostra pertencente à classe i, basta que a i-ésima saída dos classificadores apresente um valor maior que o apresentado

pelas demais para que a classificação dessa amostra seja considerada correta.

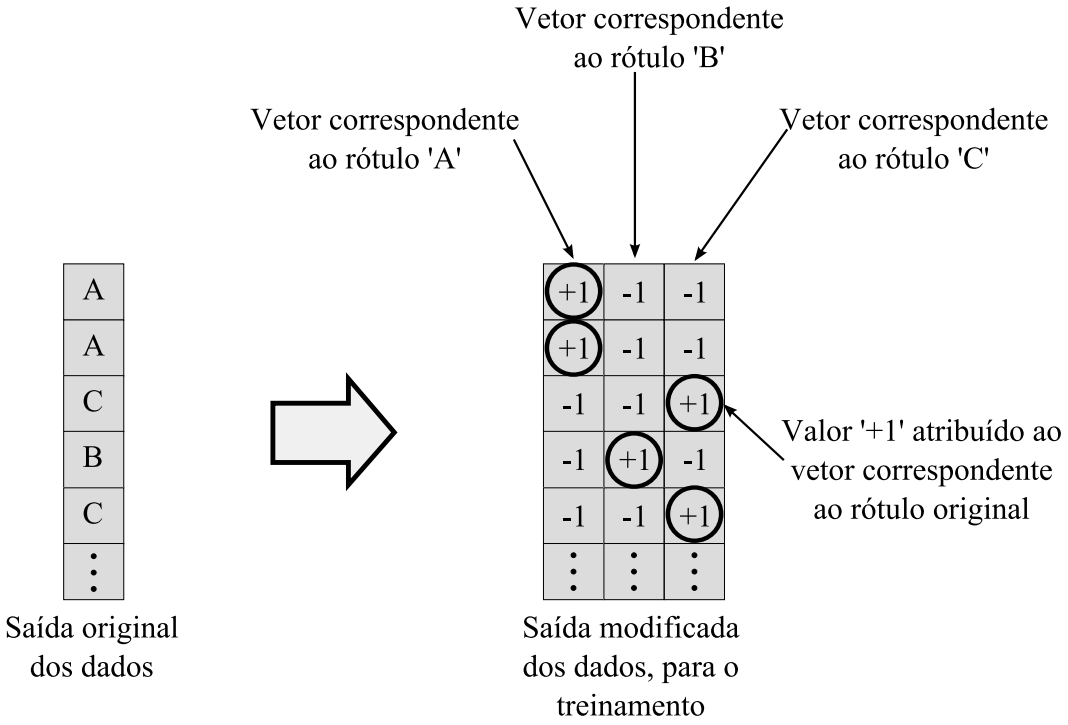


Fig. 4.4: Exemplo de adequação do conjunto de dados de um problema com três classes ao treinamento das redes neurais utilizadas neste trabalho. O rótulo de saída de cada amostra é substituído por um vetor de três posições (número de classes do problema), sendo que a posição correspondente ao rótulo da amostra em questão recebe o valor $+1$ e as demais recebem o valor -1.

Esta decisão de projeto, em relação às redes neurais com uma saída por classe (também utilizada em outros trabalhos na literatura, como em Wesolkowski & Hassanein (1997) e Lazarevic & Obradovic (2001)), foi adotada por ser considerada a mais intuitiva na determinação do rótulo de classificação de uma amostra, independentemente do número de classes que tenha o problema. Se fosse adotada a opção de, por exemplo, uma única saída para a rede neural, com o intervalo de saída (*range* da saída) subdividido em um número de sub-intervalos igual ao número de classes do problema, seriam necessários mecanismos para ajustar o tamanho destes sub-intervalos bem como o tamanho do intervalo total da saída da rede. Caso o intervalo de saída da rede fosse mantido fixo (por exemplo, em $[-1;\ +1]$) para todos os problemas, independentemente do número de classes, o tratamento de um problema de duas classes teria uma flexibilidade muito maior (cada classe estaria alocada em um sub-intervalo de tamanho 1.0) que, por exemplo, um problema com dez classes, em que, para cada classe, estaria alocado um sub-intervalo de tamanho 0.2, o que não seria desejável. Além disso, a divisão em intervalos forçaria uma *ordenação* das classes e, conseqüentemente, o estabelecimento de distâncias diferentes entre cada classe, o que também não é desejável.

4.2 Geração de Componentes

Como dito anteriormente, a etapa de geração dos componentes para os *ensembles*, ou seja, a etapa de treinamento dos classificadores baseados em redes neurais do tipo MLP, foi feita através do algoritmo *opt-aiNet*, descrito na Seção 3.2.3. Apesar da versão do algoritmo *opt-aiNet* utilizado aqui seguir exatamente os passos representados no fluxograma da Figura 3.5, alguns pontos merecem atenção: a codificação das redes neurais nos *anticorpos* do algoritmo; a métrica usada na determinação do *fitness* desses anticorpos; e a métrica de afinidade entre dois anticorpos, que é um dos pontos mais importantes para a manutenção de diversidade dos indivíduos. Estes três pontos serão detalhados nas próximas subseções.

4.2.1 Codificação das Redes Neurais

Na versão original do algoritmo *opt-aiNet*, cada anticorpo na população é representado por um vetor de atributos pertencentes ao conjunto dos números reais. Para evitar alterações nesta representação, os pesos das camadas oculta e de saída também foram codificados em um único vetor de valores reais. Esta codificação é bem simples e consiste basicamente em concatenar o vetor de pesos correspondente aos neurônios da camada intermediária com o vetor de pesos dos neurônios da camada de saída, como ilustrado na Figura 4.5.

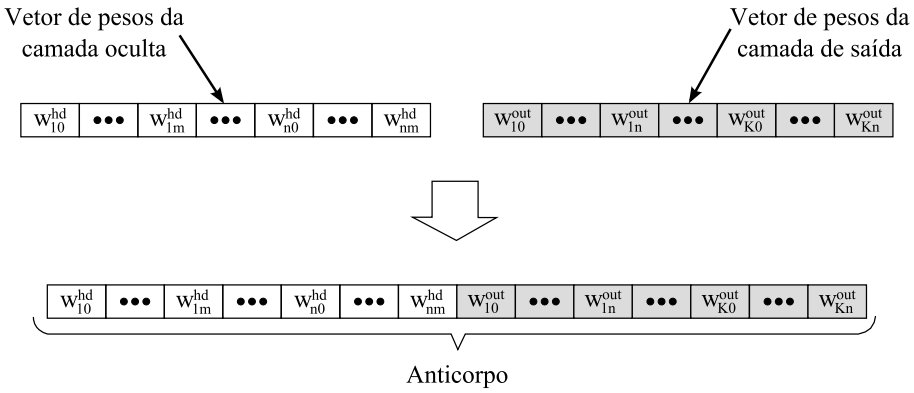


Fig. 4.5: Ilustração do processo de codificação dos vetores de pesos das redes neurais em anticorpos do algoritmo *opt-aiNet*. Na figura, os vetores de pesos correspondem a uma rede neural com m entradas, n neurônios na camada oculta e K saídas.

4.2.2 Afinidade Anticorpo/Antígeno (*fitness*)

Neste trabalho, como um dos objetivos é treinar redes neurais que apresentem baixos erros de classificação, foi adotada a segunda interpretação de *afinidade* entre anticorpos e antígenos, descrita

na Seção 3.2.1. Ou seja, a afinidade (ou *fitness*) de cada rede neural em relação aos dados de treinamento foi tomada como um termo geral que indica a qualidade de um anticorpo (rede neural) em relação ao ambiente em que está inserido.

Dessa forma, o *fitness* das redes neurais nada mais é que o número de amostras corretamente classificadas, dividido pelo número total de amostras presentes no conjunto de treinamento. Com isso, é possível avaliar o desempenho de cada indivíduo na população.

4.2.3 Afinidade Anticorpo/Anticorpo e Supressão

Já para o caso de afinidade entre dois anticorpos, foi adotada a interpretação de reconhecimento entre dois elementos (vide Seção 3.2.1). A métrica de afinidade entre dois anticorpos original do algoritmo *opt-aiNet* é proporcional ao inverso da distância euclidiana entre cada anticorpo. Neste trabalho, caso adotássemos a mesma métrica, o algoritmo geraria redes neurais com vetores de pesos diversos, o que não é interessante para o caso de *ensembles*, já que redes neurais com vetores de peso diferentes podem apresentar um mesmo padrão de classificação, ou seja, elas podem errar e acertar os resultados de classificação para as mesmas amostras. No caso de *ensembles*, é importante que os componentes apresentem diversidade em seu erro, ou seja, que errem a classificação de amostras diferentes do problema. Como não há correlação direta entre a diferença de comportamento de duas redes neurais artificiais e a distância euclidiana de seus vetores de pesos, foi necessário buscar outra métrica diferente da distância euclidiana entre os vetores de pesos.

Diante disso, neste trabalho foi adotada como métrica de afinidade entre dois anticorpos, ou seja, entre duas redes neurais, uma métrica de diversidade par-a-par (vide Seção 2.1), baseada na distância de Hamming normalizada dos *vetores de acertos* de cada classificador. Esta métrica foi proposta originalmente em de Castro et al. (2005).

Os *vetores de acertos*[1] são vetores binários (um para cada classificador na população), de tamanho igual ao número de amostras do conjunto de treinamento, onde o valor 1 na posição i indica que a i-ésima amostra do conjunto de treinamento foi corretamente classificada e o valor 0 indica que essa amostra foi incorretamente classificada (vide Figura 4.6). Calculando-se a distância de Hamming de tais vetores binários, teremos uma medida de quão semelhantes são os padrões de classificação gerados entre os classificadores, ou seja, saberemos apontar a diferença de comportamento par-a-par entre os classificadores.

Idealmente, a medida de afinidade entre dois anticorpos deveria ser dada por $1 - D_H$ (onde D_H é a distância de Hamming), de forma que quanto menor fosse esse valor, mais diversos seriam os padrões de erro entre dois classificadores. No entanto, para manter uma relação direta deste valor com o grau de diversidade entre os componentes, neste trabalho, a avaliação do grau de similaridade

[1] Também conhecidos na literatura como *saídas oráculo*.

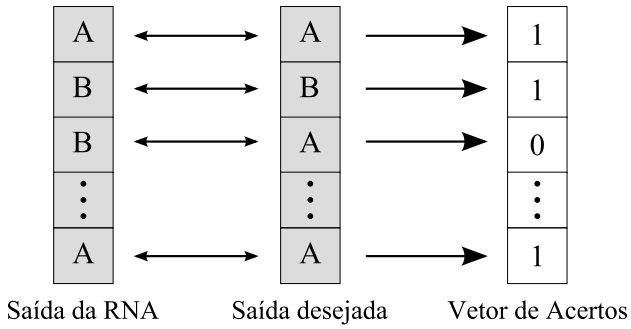


Fig. 4.6: O *vetor de acertos* de um classificador é um vetor binário de tamanho igual ao número de amostras no conjunto de dados sendo avaliado, onde o valor 1 na posição i indica que o classificador classificou corretamente a i-ésima amostra do conjunto, enquanto que um valor 0 indica que a i-ésima amostra foi erroneamente classificada.

entre dois anticorpos foi feita tomando-se a distância de Hamming diretamente (mas normalizada, ou seja, dividida pelo número de amostras no conjunto de treinamento).

Como D_H é normalizada, a medida de afinidade pode assumir valores no intervalo $[0,\ 1]$, sendo que quanto maior for esse valor, mais diversos serão os padrões de erro entre dois classificadores. Além disso, esta normalização melhora a interpretação da métrica, uma vez que, se dois indivíduos da população possuírem afinidade igual a 0.15, por exemplo, isto significa que eles divergem na classificação de 15% das amostras do conjunto de treinamento.

Dessa forma, na etapa de *supressão* do algoritmo *opt-aiNet*, todos os classificadores na população têm sua afinidade comparada e, caso dois deles possuam esta medida inferior a um determinado limiar definido pelo usuário, ou seja, caso dois deles expressem comportamentos de classificação "mais semelhantes que o desejado", o que tiver pior *fitness* é eliminado da população.

Um ponto que deve ser destacado sobre essa métrica de diversidade adotada neste trabalho é que ela é muito semelhante à métrica conhecida como *medida de não-concordância* (do inglês *disagreement measure*), utilizada por Skalak (1996) para caracterizar a diversidade entre um classificador base e um classificador complementar, e por Ho (1998) para medir a diversidade em florestas de decisão. A medida de desacordo é dada pela seguinte expressão:

$$D_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{4.2}$$

onde N^{11} é o número de amostras corretamente classificadas pelos classificadores D_i e D_k, N^{00} é o número de amostras incorretamente classificadas pelos classificadores D_i e D_k, N^{10} é o número de amostras corretamente classificadas pelo classificador D_i e incorretamente classificadas pelo classificador D_k e N^{01} é o número de amostras incorretamente classificadas pelo classificador D_i e corretamente classificadas pelo classificador D_k (vide Tabela 2.1).

Como a distância de Hamming entre dois vetores binários equivale ao número de posições para as quais os valores em cada vetor são diferentes, então, utilizando a notação adotada na Equação 4.2, essa distância de Hamming corresponde a $N^{01} + N^{10}$. Com isso, e sabendo que $N = N^{11} + N^{00} + N^{01} + N^{10}$ (onde N é o número total de amostras), temos que a medida de desacordo dada pela Equação 4.2 corresponde à distância de Hamming normalizada entre dois vetores de acertos, que é a métrica adotada neste trabalho e proposta por de Castro et al. (2005).

4.3 Seleção de Componentes

Com os candidatos a comporem o *ensemble* treinados, a etapa seguinte do processo é escolher, dentre estes candidatos, aqueles que realmente contribuem para o desempenho do ensemble, uma vez que a inclusão de todos os candidatos no *ensemble* pode degradar sua performance (Zhou et al., 2002).

Neste trabalho, foram adotadas seis técnicas de seleção de componentes: *Construtiva sem Exploração*, *Construtiva com Exploração*, *Poda sem Exploração*, *Poda com Exploração*, seleção por algoritmo de agrupamento e, para fins de comparação, a técnica *Sem Seleção*, onde todos os candidatos são considerados como componentes para o *ensemble*. Essas técnicas serão descritas nas subseções a seguir.

4.3.1 Construtiva sem Exploração

As técnicas de seleção construtivas, adotadas neste trabalho, são adaptações da técnica construtiva original, proposta por Perrone & Cooper (1993).

No caso construtivo sem exploração (que será representado nos resultados experimentais pela sigla *Cw/oE*), inicialmente os M candidatos a comporem o *ensemble* são ordenados com base no desempenho individual obtido para o conjunto de dados de validação (que é um sub-conjunto do conjunto de dados original, como será descrito na Seção 5.2), e o candidato com melhor desempenho individual é inserido no *ensemble*. O próximo candidato (o de segundo melhor desempenho individual) é então adicionado ao *ensemble*. Se o desempenho do *ensemble* em relação ao conjunto de dados de validação melhorar, o *ensemble* passa a ter dois componentes. Caso contrário, o elemento recém-inserido é removido. Este processo se repete então para todos os demais candidados restantes na lista ordenada gerada inicialmente, um após o outro. Ao final da execução dessa técnica de seleção, são testados $M - 1$ *ensembles* ao todo.

4.3.2 Construtiva com Exploração

A técnica construtiva com exploração (que será representada nos resultados experimentais pela sigla *CwE*) também inicia-se com a ordenação dos M candidatos, baseada em seu desempenho individual em relação aos dados de validação, e inserção do candidato de melhor desempenho individual no *ensemble*.

No entanto, ao invés de se considerar apenas o segundo candidato de melhor desempenho individual como segundo componente do *ensemble*, todos os $M - 1$ componentes são considerados um a um e o candidato escolhido é aquele que conseguir promover o maior aumento de desempenho do *ensemble*, tomados os dados de validação. Se nenhum dos candidatos conseguir melhorar o desempenho do *ensemble*, quando inserido, então o processo de seleção é encerrado. Caso contrário, o *ensemble* agora passa a ter dois componentes e o processo é repetido para os $M - 2$ candidatos restantes. O número de *ensembles* testados ao final da execução dessa técnica de seleção, considerando-se o pior caso, ou seja, o caso em que sempre há um incremento de desempenho do *ensemble*, é $\frac{M(M-1)}{2}$.

Para ilustrar o comportamento das duas técnicas construtivas já descritas, considere um *ensemble* com três componentes A, B e C, sendo o componente A o de melhor desempenho individual e o componente C o de pior desempenho. Considere também que o componente B melhora o desempenho de um *ensemble* apenas quando em conjunto com o componente C, enquanto que C gera um incremento de desempenho sempre que é inserido. Com isso, a técnica construtiva sem exploração executará os seguintes passos:

- Insere o candidato A no *ensemble* (A é o candidato de melhor desempenho individual);
- Insere o candidato B e compara o desempenho do novo *ensemble* (composto pelos candidatos A e B) com o *ensemble* anterior. Como B só melhora o desempenho quando em conjunto com o candidato C, este novo *ensemble* é descartado;
- Insere o candidato C e compara o desempenho do novo *ensemble* (composto pelos candidatos A e C) com o *ensemble* anterior. Como o novo *ensemble* tem um desempenho melhor que o anterior, o candidato C é mantido e o *ensemble* final obtido possui os elementos A e C.

Já a técnica construtiva com exploração, executaria os seguintes passos:

- Insere o candidato A no *ensemble* (A é o candidato de melhor desempenho individual);
- Verifica qual dos candidatos B e C restantes gera o maior aumento de desempenho quando inserido no *ensemble* obtido no passo anterior. Como B não traz nenhum benefício ao ser inserido com o candidato A, e C sempre melhora o desempenho de um *ensemble* quando inserido, o novo *ensemble* passa a ter os candidatos A e C;

- Verifica se a inserção do candidato B restante traz benefícios ao ser inserido no *ensemble*. Como o candidato C já faz parte do *ensemble* e B sempre melhora o desempenho de um *ensemble* quando em conjunto com C, então o candidato B é inserido no *ensemble*. Com isso, o *ensemble* final obtido por esta técnica possui os candidatos A, B e C, diferentemente do obtido pela técnica sem exploração.

4.3.3 Poda sem Exploração

As técnicas de poda usadas neste trabalho são adaptações da técnica de poda original, proposta por Zhou et al. (2002), e funcionam no sentido oposto ao adotado pelas técnicas construtivas.

No caso de poda sem exploração (que será representado nos resultados experimentais pela sigla *Pw/oE*), os M candidatos também são inicialmente ordenados a partir de seu desempenho individual em relação aos dados de validação. Mas, em vez de apenas o melhor candidato ser inserido no *ensemble*, todos os candidatos o são. Em seguida, o candidato com pior performance individual é removido e o *ensemble* tem seu desempenho avaliado em relação ao conjunto de dados de validação. Se ocorreu uma melhora de seu desempenho em relação à configuração anterior, o *ensemble* passa a contar então com $M-1$ componentes. Caso contrário, o candidato retirado é re-inserido, e o processo então se repete para os demais candidatos, exceto o de melhor desempenho individual, que nunca é removido do *ensemble*.

Como na técnica construtiva sem exploração, o número de *ensembles* avaliados aqui é igual ao número de candidatos candidatos a constituírem o *ensemble*, ou seja, $M - 1$.

4.3.4 Poda com Exploração

Aqui, novamente os M candidatos são ordenados com base em seus desempenhos individuais para os dados de validação, e todos são inseridos no *ensemble*. No entanto, ao invés de escolher o candidato com a pior performance individual para ser o primeiro a ser removido, todos os candidatos, com exceção do de melhor desempenho, são considerados, um após o outro, e a remoção que apresentar maior ganho de desempenho para o *ensemble* em relação aos dados de validação é tomada como definitiva. Se nenhuma remoção resultar em ganho para o *ensemble*, o processo é encerrado. Caso contrário, o processo continua e é repetido para os $M - 2$ candidatos restantes (lembrando que o candidato de melhor desempenho individual nunca é removido do *ensemble*). No pior caso, o número total de *ensembles* avaliados nesta técnica é $\frac{M(M-1)}{2}$.

Esta técnica de poda com exploração será representada nos resultados experimentais pela sigla *PwE*.

Para ilustrar o comportamento das duas técnicas de poda descritas aqui, considere um *ensemble*

com três candidatos A, B e C, sendo o candidato A o de melhor desempenho individual e o candidato C o de pior desempenho. Considere também que, ao ser removido, o candidato C melhora o desempenho de um *ensemble* apenas quando o candidato B não está mais presente, enquanto que B gera um incremento de desempenho sempre que é removido. Com isso, a técnica de poda sem exploração executará os seguintes passos:

- Insere todos os candidatos no *ensemble*, que passa a ter os candidatos A, B e C;
- Remove o candidato C. Como o candidato B permanece no *ensemble*, essa remoção não leva a nenhum ganho de desempenho e C é re-inserido no *ensemble*;
- Remove o candidato B, o que leva a um ganho de desempenho. Como o candidato A (de melhor desempenho individual) nunca é removido, a execução se encerra e o *ensemble* final obtido possui os candidatos A e C.

Já a técnica de poda com exploração, executaria os seguintes passos:

- Insere todos os candidatos no *ensemble*, que passa a ter os candidatos A, B e C;
- Verifica qual dos candidatos B e C, ao ser removido, leva ao maior ganho de desempenho do *ensemble*. Como C não traz nenhum benefício ao ser removido, sendo que o candidato B ainda permanece no *ensemble*, e B sempre melhora o desempenho de um *ensemble* ao ser removido, o novo *ensemble* passa a ter os candidatos A e C;
- Verifica se a remoção do candidato C restante traz benefícios ao *ensemble*. Como o candidato B foi removido no passo anterior, a remoção do candidato C promove um ganho de desempenho ao *ensemble*, o que faz com que o *ensemble* final tenha apenas o candidato A, diferentemente do obtido pela técnica sem exploração.

4.3.5 Algoritmo de Agrupamento - ARIA

Uma outra forma de seleção de componentes utilizada neste trabalho foi a seleção através do uso de ferramentas de agrupamento de dados (ou *clusterização*). A idéia aqui é agrupar os candidatos com características semelhantes e selecionar os representantes de melhor desempenho individual, em cada grupo, para compor o *ensemble*. Uma abordagem análoga foi proposta por Liu et al. (2000), onde foi proposta uma abordagem evolutiva com aprendizado por correlação negativa e agrupamento dos classificadores resultantes via *k-means* (Seber, 1984). Existem duas principais desvantagens do uso do algoritmo *k-means*: a convergência prematura para ótimos locais e a necessidade de definição prévia do número de grupos (ou *clusters*). Neste trabalho, para evitar tais limitações, especialmente

a necessidade de definir o número de grupos (e, conseqüentemente, o número de componentes no *ensemble*), foi aplicado o algoritmo de agrupamento ARIA.

O algoritmo ARIA (*Adaptive Radius Immune Algorithm*), que é capaz de definir automaticamente o número de grupos e possui mecanismos de exploração do espaço de busca mais eficientes, pode ser considerado uma extensão do algoritmo *aiNet* (de Castro & Von Zuben, 2001) e foi proposto por Bezerra et al. (2005) para problemas de agrupamento. O principal diferencial do ARIA em relação a outros algoritmos de agrupamento capazes de automaticamente determinar o número de grupos (e que também poderiam ser empregados aqui), é a capacidade de considerar informações de densidade presentes nos dados, evitando assim distorções nas densidades relativas de anticorpos, quando essas densidades relativas são comparadas àquelas associadas aos antígenos (dados originais).

Para melhor ilustrar essa capacidade de consideração de informações de densidade, tomemos o problema de agrupamento representado na Figura 4.7. Em problemas de agrupamento, geralmente os algoritmos posicionam um certo número de candidatos no espaço dos dados, chamados de *protótipos*, que são responsáveis por representar todos os dados que estejam dentro de um dado raio de atuação deste protótipo. Após o posicionamento desses indivíduos, que passam a representar os dados originais do problema, é aplicada uma técnica capaz de separar esses protótipos em grupos distintos, que corresponderão aos grupos ou *clusters* procurados. No caso do algoritmo ARIA, as informações de densidade dos dados são consideradas nas etapas de posicionamento dos protótipos e determinação de seus raios de atuação, e a separação em grupos é feita pela técnica conhecida como *Árvore Geradora Mínima* (ou *Minimum Spanning Tree* – MST – Gabow et al., 1986).

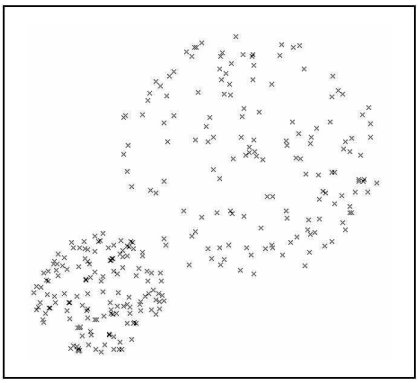
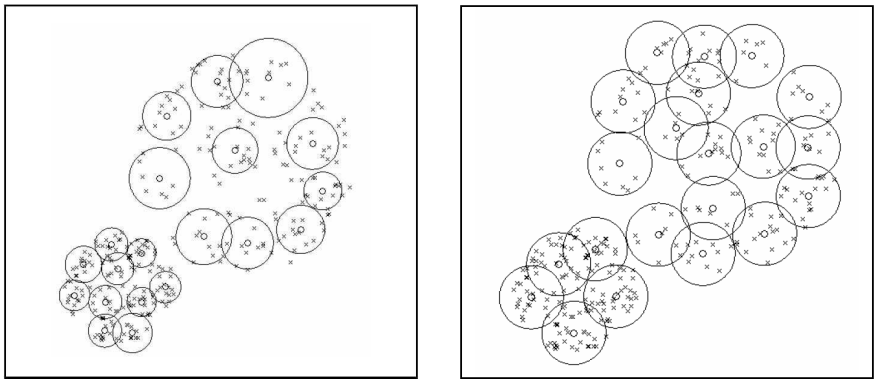


Fig. 4.7: Problema de agrupamento com dois grupos de 150 amostras e diferentes densidades.

Na Figura 4.8(a) temos o posicionamento final dos anticorpos (ou protótipos) após a convergência do ARIA para o problema da Figura 4.7, e na Figura 4.8(b) os resultados obtidos com a aplicação de um algoritmo que não considera a informação de densidade dos dados (nesse caso, o algoritmo *aiNet*).

Como pode ser observado na Figura 4.8, o ARIA foi capaz de posicionar um mesmo número de



(a) ARIA (b) *aiNet*

Fig. 4.8: Posicionamento dos anticorpos após a convergência dos algoritmos ARIA e *aiNet* para o problema representado na Figura 4.7.

protótipos em cada um dos dois grupos de dados do problema, mesmo com estes grupos apresentando densidades diferentes. Já para o algoritmo *aiNet*, isso não ocorreu, uma vez que o número de protótipos posicionados na região de menor densidade foi muito maior que o número de protótipos na região de maior densidade, mesmo com o número de amostras sendo igual para os dois grupos do problema.

O pseudo-código da etapa de posicionamento de protótipos do ARIA está representado no Algoritmo 4.1 abaixo.

Algoritmo 4.1 Posicionamento de protótipos do algoritmo ARIA

```
Inicializa variáveis;
Gera população inicial;
for iteração 1 até gen do
  for cada Antígeno (amostra dos dados) do
    Selecione o anticorpo de melhor afinidade;
    Aplique mutação com taxa mi;
  end for
  Elimine os anticorpos que não forem estimulados;
  Clone os anticorpos que reconheçam antígenos posicionados a uma distância superior ao seu
  raio R;
  Calcule a densidade local para cada anticorpo;
  Calcule o raio de supressão para cada anticorpo (inversamente proporcional à densidade dos
  dados);
  Suprima os anticorpos, favorecendo a eliminação daqueles de maior raio;
  if a atual geração é maior que gen/2 then
    Altere a taxa de mutação da seguinte forma: mi = mi * (decaimento);
  end if
end for
```

O algoritmo ARIA começa com a inicialização das variáveis e geração aleatória da população inicial de anticorpos (protótipos). Como a população do algoritmo varia dinamicamente ao longo de sua execução, o número inicial de anticorpos pode ser pequeno, uma vez que, caso sejam necessários mais indivíduos, o próprio algoritmo se encarregará de inseri-los na população. O ARIA também é capaz de ajustar os raios de cada anticorpo na população, o que permite que os raios iniciais também possam ser gerados aleatoriamente. A taxa de mutação mi é inicializada com o valor 1.

Feita a inicialização, as amostras de dados de treinamento são apresentadas uma a uma e aleatoriamente à população de anticorpos. Nessa etapa, quando um dado anticorpo reconhece uma amostra, ou seja, quando a sua distância à amostra é menor que a de todos os demais anticorpos na população (maior afinidade), esse anticorpo sofre mutação na direção da amostra reconhecida, com taxa mi.

Em seguida, os anticorpos que não foram capazes de reconhecer nenhuma amostra dos dados são eliminados da população, e aqueles que reconheceram amostras fora de seus raios de atuação são *clonados*, sendo que os clones gerados sofrem mutação. Mesmo que um anticorpo tenha reconhecido várias amostras fora de seu raio de atuação, apenas um clone é gerado para este protótipo.

Gerada a nova população de anticorpos, a densidade local dos dados para cada indivíduo, bem como seus respectivos raios de atuação são determinados e passa-se à fase de supressão. Nesta fase, os anticorpos são comparados dois a dois e, caso a distância entre eles seja menor que um dos raios de supressão, o indivíduo de maior raio é suprimido.

Por fim, a taxa de mutação mi é alterada. O valor de mi é mantido constante por $gen/2$ iterações e, em seguida, começa a ser reduzido geometricamente pela constante $decaimento$, de forma a forçar a convergência da rede de anticorpos.

Após o posicionamento dos protótipos e determinação de seus raios de atuação, o algoritmo ARIA aplica um procedimento baseado em Árvores Geradoras Mínimas para definir automaticamente os agrupamentos. Este processo de definição automática de agrupamentos pode ser dividido em duas etapas: (*i*) construção de uma árvore geradora mínima que une todos os anticorpos posicionados na etapa anterior do algoritmo; e (*ii*) determinação e eliminação das arestas inconsistentes da árvore, de forma que as arestas remanescentes e conectadas entre si possam ser consideradas como pertencentes a um mesmo agrupamento.

Para determinar quais arestas da árvore geradora mínima são inconsistentes, as extremidades de cada uma delas são consideradas e a média e o desvio padrão do comprimento das arestas localizadas a p ou menos passos de cada uma dessas extremidades são calculados: se o comprimento da aresta sendo analisada for maior que a média adicionada de dois dos desvios padrões calculados para estas arestas vizinhas às extremidades, ela é considerada inconsistente (Zhan, 1971).

A descrição completa do algoritmo ARIA, bem como de todos os mecanismos empregados em cada etapa, pode ser encontrada em Bezerra et al. (2005).

Neste trabalho, o agrupamento feito pelo algoritmo ARIA foi aplicado aos vetores de saída produzidos pelos classificadores gerados pela execução do algoritmo *opt-aiNet*, e não pelos vetores de pesos dessas redes neurais. Dessa forma, os candidatos a componentes do *ensemble* são agrupados de acordo com seus padrões de classificação, e não pela semelhança entre seus vetores de peso.

Feito este agrupamento, a mesma abordagem utilizada por Liu et al. (2000) foi adotada: o candidato com melhor desempenho individual (para o conjunto de validação) de cada grupo é selecionado para fazer parte do *ensemble* e o *ensemble* é montado com número de componentes igual ao número de grupos obtido automaticamente pelo algoritmo ARIA.

4.3.6 Sem Seleção

Como o próprio nome diz, esta técnica de seleção consiste em adicionar todos os candidatos da população de candidatos ao *ensemble*, independentemente de prejudicarem ou não o desempenho do mesmo. Este critério foi adotado para fins de comparação com as demais técnicas. Nos resultados experimentais, esta técnica de seleção será representada pela sigla *SS*.

4.4 Combinação de Componentes

Outro ponto importante na construção de *ensembles* é a maneira como as saídas de cada componente serão combinadas em uma única saída (a saída do *ensemble*). Neste trabalho, foram utilizadas cinco técnicas de combinação de componentes, que serão descritas a seguir.

4.4.1 Média Simples

Esta primeira estratégia de combinação (que será tratada pela sigla *MS* nos resultados experimentais), também conhecida como *Vector Addition*, é uma técnica bem simples e se aplica tanto para problemas de regressão quanto para problemas de classificação. Aqui, a saída do *ensemble* é dada pela média simples entre as saídas de cada um de seus componentes antes da conversão para os rótulos das classes, ou seja,

$$y^k = \frac{1}{M}\sum_{i=1}^{M} y_i^k, \tag{4.3}$$

onde M é o número de componentes no *ensemble*, y_i^k é a k-ésima saída do i-ésimo componente e y^k é a k-ésima saída do *ensemble*.

4.4.2 Média Ponderada sem *Bias*

Nesta estratégia de combinação, que será representada pela sigla *MP* nos resultados experimentais, a saída do *ensemble* é dada por uma soma ponderada das saídas de cada um de seus componentes (antes da conversão de saída contínua para os rótulos das classes), como representado na Equação 4.4.

$$y^k = p_1^k \times y_1^k + p_2^k \times y_2^k + \cdots + p_M^k \times y_M^k, \tag{4.4}$$

onde *M* é o número de componentes no *ensemble*, y^k é a k-ésima saída do *ensemble*, y_i^k é a k-ésima saída do i-ésimo componente e p_i^k é o peso atribuído à saída k do componente i.

Os pesos para a saída k de cada componente neste método de combinação foram obtidos através da resolução do seguinte problema de minimização, dado pelas Expressões 4.5, 4.6 e 4.7:

$$\min_{p^k} \frac{1}{2} \|C\vec{p}^k - \vec{y}_d^k\|_2^2 \tag{4.5}$$

sujeito a:

$$\sum_i p_i^k = 1 \tag{4.6}$$

$$p_i^k \geq 0, \forall i \tag{4.7}$$

onde $\vec{y}_d^k$ é o vetor de saídas desejadas para o conjunto de dados de validação (vide Seção 5.2), e a matriz C e o vetor de pesos $\vec{p}^k$ são dados por:

$$C = [\vec{y}_1^k \quad \vec{y}_2^k \quad \cdots \quad \vec{y}_M^k] \tag{4.8}$$

$$\vec{p}^k = [p_1^k \quad p_2^k \quad \cdots \quad p_M^k]^T \tag{4.9}$$

onde M é o número de componentes no *ensemble* e $\vec{y}_i^k$ é o vetor de saídas do i-ésimo componente.

A restrição dada pela Expressão 4.7 é necessária, uma vez que, se fossem permitidos pesos negativos, alguns dos classificadores de alto desempenho gerados pelo *opt-aiNet* poderiam ter suas saídas completamente invertidas, o que provocaria uma reversão completa no padrão de classificação do componente. Cabe ressaltar novamente aqui que o rótulo da classe de uma dada amostra é dado pelo índice da saída de maior valor da rede neural.

O problema de otimização dado pelas Expressões 4.5, 4.6 e 4.7 foi resolvido através do *Optimization Toolbox* da ferramenta *Matlab*. O problema foi modelado como um problema de quadrados mínimos com restrições e resolvido através de uma técnica baseada em *Programação Quadrática*, implementada na ferramenta.

4.4.3 Média Ponderada com *Bias*

A técnica de combinação por média ponderada com *bias* (ou polarização), representada pela sigla *MPB*, é bem semelhante à técnica de combinação por média ponderada simples descrita na Seção 4.4.2. No entanto, ao fazer a ponderação das saídas de cada componente e a sua combinação, é adicionado um termo de polarização, como pode ser visto na Expressão 4.10 (termo p_0^k).

$$y^k = p_0^k + p_1^k \times y_1^k + p_2^k \times y_2^k + \cdots + p_M^k \times y_M^k \tag{4.10}$$

onde *M* é o número de componentes no *ensemble*, y^k é a k-ésima saída do *ensemble*, y_i^k é a k-ésima saída do i-ésimo componente e p_i^k é o peso atribuído à k-ésima saída do componente i.

Os pesos para a saída k de cada componente e o termo p_0 de polarização foram obtidos através da resolução do seguinte problema de minimização, dado pelas Expressões 4.11, 4.12 e 4.13:

$$\min_{p^k} \frac{1}{2} \|C\vec{p}^k - \vec{y}_d^k\|_2^2 \tag{4.11}$$

sujeito a:

$$\sum_{i=1}^{M} p_i^k = 1, \tag{4.12}$$

$$p_i^k \geq 0, \forall i \in [1, \ldots, M] \tag{4.13}$$

onde $\vec{y}_d^k$ é o vetor de saídas desejadas para o conjunto de dados de validação (vide Seção 5.2), e a matriz C e o vetor de pesos $\vec{p}^k$ são dados por:

$$C = [\vec{1} \quad \vec{y}_1^k \quad \cdots \quad \vec{y}_M^k] \tag{4.14}$$

$$p^k = [p_0^k \quad p_1^k \quad \cdots \quad p_M^k]^T \tag{4.15}$$

onde M é o número de componentes no *ensemble*, $\vec{y}_i^k$ é o vetor de saídas do i-ésimo componente e $\vec{1}$ é um vetor onde todas as posições têm o valor 1. É importante ressaltar aqui que o termo de polarização p_0^k não está sujeito às restrições dadas pelas Expressões 4.12 e 4.13.

O problema de otimização dado pelas Expressões 4.11, 4.12 e 4.13 foi novamente resolvido através do *Optimization Toolbox* da ferramenta *Matlab*, de maneira análoga à feita para a técnica de média ponderada sem *bias* (Seção 4.4.2).

4.4.4 Voto Majoritário

A técnica de combinação conhecida como voto majoritário (representada pela sigla *VM* nos resultados experimentais) é um método de combinação não-linear onde, dada uma amostra do conjunto de dados, cada componente do *ensemble* apontará uma classe para esta amostra e a classe que receber o maior número de votos (for apontada pelo maior número de componentes), corresponderá ao rótulo atribuído a esta amostra pelo *ensemble*.

Para ilustrar, considere o caso de um problema de duas classes e um *ensemble* com três componentes cujas saídas e respectivos rótulos para uma dada amostra são:

- Componente 1: $\vec{y_1} = [+0.3 \ -0.8]$ - Classe A;
- Componente 2: $\vec{y_2} = [+0.2 \ +0.9]$ - Classe B;
- Componente 3: $\vec{y_3} = [+0.7 \ +0.6]$ - Classe A;

Como dois dos três componentes classificaram a amostra em questão como pertencente à classe "A", então o *ensemble* também atribuirá a esta amostra o rótulo de classe "A".

Caso haja um empate na votação, o rótulo da classe resultante para a amostra em questão é determinado arbitrariamente, dentre aqueles que obtiveram o maior número de votos.

4.4.5 Winner-takes-all

O método *winner-takes-all* (representado pela sigla *WTA*) é uma técnica não-linear e elitista que consiste em selecionar como rótulo da classe na saída do *ensemble* o rótulo gerado pelo componente que apresentar a saída de maior valor dentre todos os componentes do *ensemble*.

Para ilustrar este método, considere o mesmo problema do exemplo dado para a técnica de voto majoritário na Seção 4.4.4. Analisando as saídas dos três classificadores, temos que a de maior valor é a segunda saída do segundo componente (valor $+0.9$), que indica que a amostra em questão pertence à classe "B". Dessa forma, o *ensemble* também atribuirá a esta amostra o rótulo de classe "B".

4.5 Resumo das Siglas Utilizadas para as Técnicas de Seleção e Combinação

Para facilitar a apresentação dos resultados experimentais no próximo capítulo, cada técnica de seleção e combinação de componentes empregada neste trabalho foi denotada por uma sigla, conforme apresentado na Tabela 4.1 abaixo.

Tab. 4.1: Resumo das siglas adotadas para as técnicas de seleção e combinação de componentes empregadas neste trabalho.

Técnicas de Seleção	Sigla
Construtiva sem Exploração	Cw/oE
Construtiva com Exploração	CwE
Poda sem Exploração	Pw/oE
Poda com Exploração	PwE
Algoritmo de Agrupamento	ARIA
Sem Seleção	SS
Técnicas de Combinação	**Sigla**
Média Simples	MS
Média Ponderada sem *Bias*	MP
Média Ponderada com *Bias*	MPB
Voto Majoritário	VM
Winner-takes-all	WTA

4.6 Síntese do Capítulo

Neste capítulo, foi feito o detalhamento da proposta deste trabalho. Inicialmente foi feita uma breve apresentação das redes neurais do tipo MLP e, em seguida, foram apresentadas as técnicas empregadas nas etapas de construção de um *ensemble*.

Basicamente, a construção de *ensembles* envolve três etapas: geração, seleção e combinação de componentes. Na etapa de geração de componentes, foi aplicado o algoritmo imuno-inspirado *opt-aiNet*, que apresenta características de manutenção de diversidade e capacidade de localização de múltiplos ótimos locais e globais e de determinação automática do número de indivíduos na população. Na Seção 4.2, foi detalhado como as redes neurais empregadas neste trabalho (classificadores) foram codificadas em indivíduos da população tratada pelo algoritmo (chamados de anticorpos), como cada um destes classificadores é avaliado (*fitness*) e como o mecanismo de manutenção de diversidade

do algoritmo foi alterado de forma a manter uma população de indivíduos diversos, não em relação aos seus vetores de pesos, mas em relação aos seus padrões de classificação apresentados para um dado conjunto de dados.

Vencida a etapa de geração de um grupo de classificadores candidados a comporem um *ensemble*, é necessário que, dentre estes, sejam selecionados aqueles que realmente contribuem para o desempenho do *ensemble*, eliminando candidatos que tragam redundância desnecessária e que possam ser prejudiciais ao *ensemble*. Neste trabalho, foram adotadas seis técnicas de seleção de componentes: construtiva sem e com exploração, poda sem e com exploração, algoritmo de agrupamento e sem seleção. Estas técnicas foram descritas na Seção 4.3.

A última etapa de construção de um *ensemble* é a determinação da forma como as saídas de cada componente serão combinadas em uma única saída. Aqui, foram aplicadas cinco técnicas de combinação: média simples, média ponderada sem *bias*, média ponderada com *bias*, voto majoritário e *winner-takes-all*, todas descritas na Seção 4.4.

No próximo capítulo, serão apresentados os problemas tratados neste trabalho, a metodologia empregada para avaliar as diversas configurações de *ensembles* e os resultados experimentais obtidos.

Capítulo 5

Resultados Experimentais

Para verificação da validade da metodologia proposta neste trabalho, foi feita uma série de experimentos a partir de conjuntos de dados de problemas reais e artificiais, e os resultados obtidos são apresentados e discutidos neste capítulo. Tais conjuntos de dados utilizados aqui estão descritos na Seção 5.1, e a metodologia empregada nos experimentos, na Seção 5.2.

Os resultados propriamente ditos são apresentados nas três seções subseqüentes: a verificação da capacidade do algoritmo *opt-aiNet* gerar um conjunto de classificadores de bom desempenho e ao mesmo tempo diversos é feita na Seção 5.3; na Seção 5.4 são apresentados resultados que indicam que o desempenho das técnicas de seleção e combinação de componentes não depende apenas do problema em si, mas sim do conjunto de elementos candidatos a constituírem um *ensemble*; e, por fim, na Seção 5.5 os resultados finais dos experimentos são apresentados e discutidos.

Optou-se por essa divisão, em que os resultados de todas as bases de dados são apresentados conjuntamente em cada seção (ou seja, por uma divisão de acordo com os pontos que se deseja ressaltar), para facilitar a compreensão e a comparação dos resultados obtidos para cada problema tratado.

5.1 Conjuntos de Dados

Os experimentos deste trabalho foram feitos utilizando-se quatro conjuntos de dados: um gerado artificialmente e três derivados de problemas reais, disponíveis no *UCI Repository of Machine Learning Databases* (Newman et al., 1998).

O conjunto de dados artificiais é semelhante ao utilizado no trabalho de de Castro et al. (2005), e possui três mil amostras, dois atributos e três classes. As amostras de cada classe desse conjunto artificial foram geradas aleatoriamente, seguindo uma distribuição gaussiana, como representado na Figura 5.1.

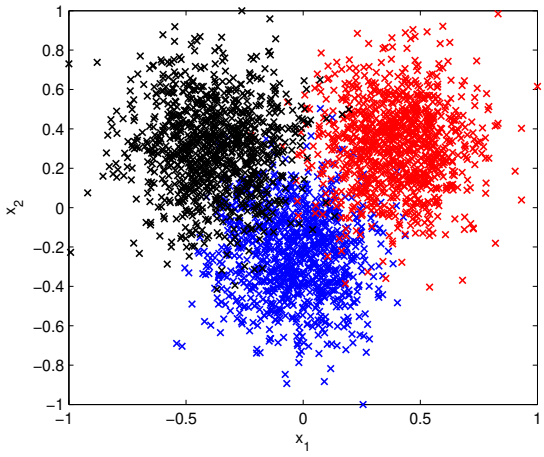


Fig. 5.1: Representação gráfica do conjunto de dados artificiais, com dois atributos x_1 e x_2. Cada classe está representada na figura por uma cor diferente.

Os conjuntos de dados de problemas reais utilizados neste trabalho foram: *Bupa Liver Disorder*, *Wine Recognition* e *Glass Identification*.

O conjunto *Bupa* é um conjunto de dados médicos sobre doenças de fígado, e possui 345 amostras com seis atributos reais e duas classes, sendo que a primeira possui 145 e a segunda 200 amostras.

O conjunto *Wine* possui 178 amostras com 13 atributos reais, que correspondem a resultados de análises químicas de vinhos, e 3 classes que correspondem às regiões de origem de cada vinho testado. Neste conjunto, existem 59 amostras correspondentes à primeira classe, 71 à segunda e 48 à terceira classe.

O terceiro conjunto de dados reais utilizado, o conjunto *Glass*, foi produzido pelo Serviço de Ciências Forenses norte-americano, e consiste em 214 amostras de 9 atributos (também com valores reais) que correspondem à composição química de 6 tipos de vidros, ou seja, existem 6 classes possíveis no problema. O número de amostras em cada classe está distribuído da seguinte forma: 70 amostras da primeira classe, 76 da segunda, 17 da terceira, 13 da quarta, 9 da quinta e 29 amostras da sexta classe.

A Tabela 5.1 apresenta um resumo das principais características dos conjuntos de dados utilizados nos experimentos deste trabalho.

5.2 Metodologia

Para manter a independência entre os processos de treinamento e seleção, e como os conjuntos de dados reais não possuem uma quantidade de amostras suficientes para seguir o particionamento ideal destes conjuntos originais em cinco sub-conjuntos (como descrito na Introdução do Capítulo

Tab. 5.1: Resumo das principais características dos conjuntos de dados utilizados neste trabalho.

Conjunto de Dados	Número de Amostras	Número de Atributos	Número de Classes
Artificial	3000	2	3
Bupa	345	6	2
Wine	178	13	3
Glass	214	9	6

2), foi adotado o particionamento dos conjuntos de dados originais em três sub-conjuntos: *treinamento*, *validação* e *teste*. O sub-conjunto de *treinamento* foi utilizado na etapa de treinamento dos classificadores, o sub-conjunto de *validação* nas etapas de seleção e combinação de componentes e o sub-conjunto de *teste* foi utilizado para verificar o desempenho do *ensemble* e dos classificadores individualmente.

Para que todas as amostras presentes em cada conjunto de dados fossem utilizadas no sub-conjunto de *teste*, para todos os problemas tratados neste trabalho os conjuntos de dados originais foram particionados aleatoriamente em dez sub-conjuntos. Após esse particionamento, o algoritmo proposto foi então executado dez vezes, cada uma utilizando um dos dez sub-conjuntos previamente definidos como conjunto de *teste* e os demais nove conjuntos (restante dos dados) para *treinamento/validação*. Dentre todos os dados nestes nove sub-conjuntos restantes, 40% foi usado na construção do sub-conjunto de *treinamento* e 60% na construção do sub-conjunto de *validação*[1], uma vez que o objetivo principal era obter uma performance melhor no *ensemble* e não nos classificadores individuais.

Outra forma de particionamento dos dados que foi utilizada foi o *particionamento estratificado*, que consiste em manter a proporção das classes do conjunto original em cada sub-conjunto gerado. Para exemplificar, se considerarmos um problema com duas classes, onde 25% das amostras pertencem à classe 1 e 75% à classe 2, então nos sub-conjuntos de treinamento, validação e teste, apesar de termos um número total de amostras menor, a proporção entre as classes se manterá em 25% para a classe 1 e 75% para a classe 2 nos três sub-conjuntos. No entanto, como os resultados observados com o uso do particionamento estratificado não apresentaram diferenças significativas em relação à outra maneira de particionamento de dados descrita anteriormente, nesta dissertação serão apresentados apenas os resultados obtidos com a técnica de particionamento não estratificada.

Para a execução dos experimentos, foi necessário definir alguns parâmetros, principalmente para o algoritmo *opt-aiNet*. Esses parâmetros foram definidos empiricamente, após uma série de ensaios

[1]Esta nova subdivisão também foi feita através de re-amostragem aleatória dos dados. No entanto, ela é repetida ao início de cada execução do algoritmo.

preliminares, de forma a produzirem um bom desempenho para o sistema, com custo computacional relativamente baixo. Os valores adotados estão representados na Tabela 5.2, de acordo com cada conjunto de dados utilizado nos experimentos.

Tab. 5.2: Principais parâmetros utilizados nos experimentos.

	Artificial	Bupa	Wine	Glass
Tamanho da População Inicial (Anticorpos)	10	10	10	10
Número de Clones por Anticorpo	10	10	10	10
Critério de Parada (Número de Gerações)	100	100	100	150
Limiar de Supressão	0.05	0.05	0.05	0.05
Número de Neurônios na Camada Oculta	2	2	2	2

No caso da escolha de dois neurônios na camada oculta das redes neurais, foram feitos testes com um, dois, três, quatro, cinco e dez neurônios, mas não foram observadas diferenças significativas de desempenho para nenhum dos conjuntos de dados tratados, exceto para o caso com um único neurônio, onde os resultados não foram satisfatórios. Dessa forma, optou-se pelo uso de dois neurônios, de forma a manter a configuração mais simples possível.

5.3 Validação da opt-aiNet

O objetivo desta seção é apresentar os resultados da etapa de geração de componentes para os *ensembles*, de forma a verificar a capacidade do algoritmo *opt-aiNet* em gerar um conjunto de classificadores de alto desempenho e ao mesmo tempo diversos. Para isso, vamos considerar uma única execução do algoritmo para o problema *Wine*, escolhido arbitrariamente. Nesta execução, foram gerados onze classificadores, cujos índices de acertos para o conjunto de treinamento estão apresentados na Tabela 5.3 abaixo, após serem ordenados de acordo com seus desempenhos individuais.

Como pode ser verificado na Tabela 5.3, seis dos onze classificadores obtiveram um índice de acertos superior a 90%, e todos eles ficaram acima de 70%. As medidas de diversidade média, mínima e máxima destes classificadores são dadas na Tabela 5.4. Como pode ser observado, os classificadores obtidos ao final dessa execução, além de apresentarem índices de acertos acima de 70%, são significativamente diversos entre si. Vale ressaltar aqui que esta medida de diversidade foi adotada como a métrica de afinidade entre anticorpos (vide Seção 4.2.3).

Os resultados observados nesta execução da etapa de geração de componentes para o problema *Wine* podem ser estendidos para os demais problemas tratados neste trabalho. A Figura 5.2 apresenta o desempenho médio e a Figura 5.3 a diversidade média, para o sub-conjunto de treinamento, entre

Tab. 5.3: Índice de acertos dos 11 classificadores obtidos em uma execução da etapa de geração, para o problema *Wine* (conjunto de treinamento). Este índice de acertos corresponde ao número de classificações corretas dividido pelo número total de amostras e varia entre 0 e 1, sendo que 0 corresponde a 100% de erros e 1 a 100% de acertos.

Classificador	Índice de Acertos	Classificador	Índice de Acertos
1	1.00	7	0.88
2	0.94	8	0.88
3	0.94	9	0.85
4	0.91	10	0.78
5	0.91	11	0.72
6	0.90	-	-

Tab. 5.4: Valores mínimo, médio e máximo da diversidade entre anticorpos para o conjunto de classificadores representados na Tabela 5.3.

Diversidade Mínima	Diversidade Média	Diversidade Máxima
0.06	0.19	0.44

os classificadores obtidos ao final das dez execuções da etapa de geração, para os quatro problemas tratados neste trabalho. Como pode ser observado, os classificadores obtiveram desempenho compatível com o exigido para cada problema e, ao mesmo tempo, mantiveram a diversidade entre si. Na Figura 5.3, é interessante observar para o conjunto *Glass* que, mesmo sendo uma base de dados altamente desbalanceada (algumas classes com poucas amostras e outras com muitas), o *opt-aiNet* foi capaz de gerar um conjunto de classificadores diversos. Para atestar a qualidade dos resultados obtidos, na Tabela 5.5 são apresentados alguns dos melhores resultados obtidos na literatura para os problemas *Bupa*, *Wine* e *Glass*.

Tab. 5.5: Resultados da literatura para os problemas *Bupa*, *Wine* e *Glass*.

Problema	Índice de Acertos	Referência
Bupa	0.74 ± 0.01	de Castro et al. (2005)
Wine	0.97 ± 0.01	Inoue & Narihisa (2004)
Glass	0.69 ± 0.01	Lazarevic & Obradovic (2001)

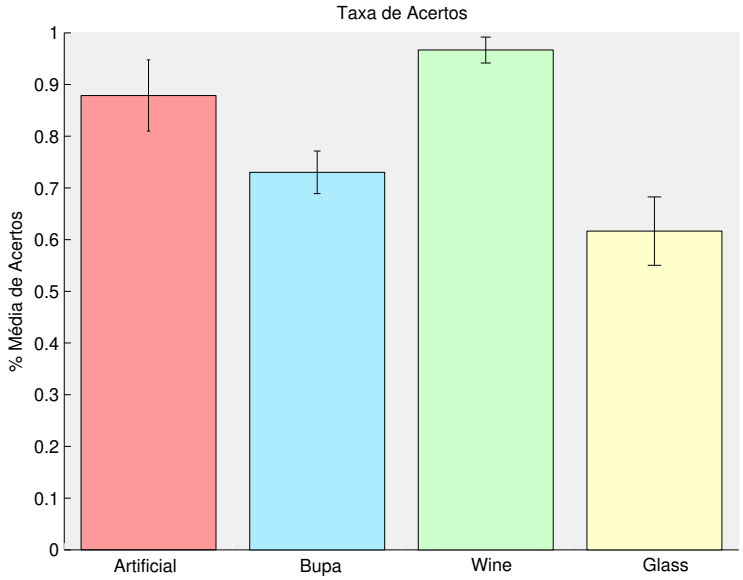


Fig. 5.2: Representação gráfica do desempenho médio obtido pelos classificadores, junto ao sub-conjunto de treinamento, nas dez execuções da etapa de geração, para os problemas *Artificial*, *Bupa*, *Wine* e *Glass*. Cada coluna corresponde ao desempenho médio, enquanto que as barras no topo das colunas correspondem ao desvio padrão obtido.

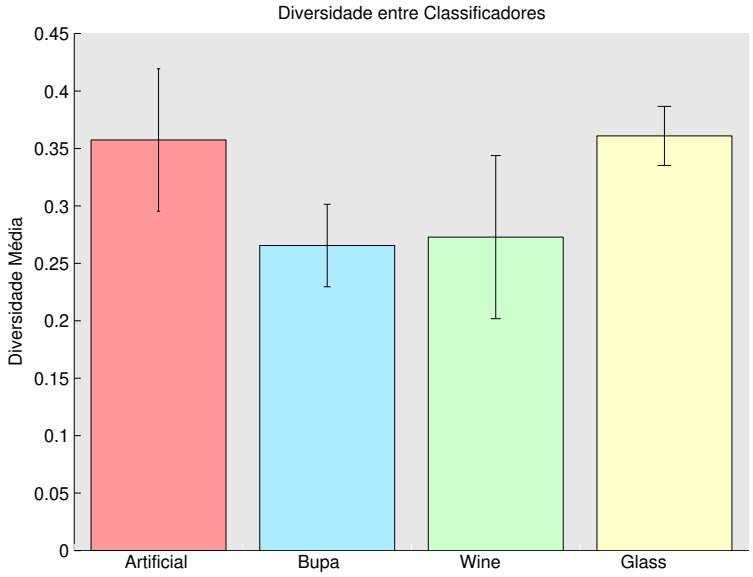


Fig. 5.3: Representação gráfica da métrica de diversidade anticorpo/anticorpo média obtida pelos classificadores, junto ao sub-conjunto de treinamento, nas dez execuções da etapa de geração, para os problemas *Artificial*, *Bupa*, *Wine* e *Glass*. Cada coluna corresponde à diversidade média, enquanto que as barras no topo das colunas correspondem ao desvio padrão obtido.

5.4 Dependência entre Seleção e Combinação e o Conjunto de Candidatos ao *Ensemble*

Uma característica que foi observada nos resultados obtidos nos experimentos foi a dependência entre o desempenho das técnicas de seleção e combinação de componentes e o conjunto de candidatos disponíveis. Intuitivamente, se considerarmos um dado problema e construirmos vários *ensembles* a partir de um conjunto de n_1 candidatos e repetirmos esse experimento para outro conjunto com n_2 candidatos diferentes, imagina-se que o par de técnicas de seleção e combinação que resultarem no *ensemble* de melhor desempenho para o primeiro caso será o mesmo para o segundo. No entanto, o que se observou neste trabalho (e também já foi percebido em outros trabalhos) foi que isso não ocorria, ou seja, para cada um dos problemas tratados, não foi possível identificar um par de técnicas de seleção e combinação que fosse o mais adequado a cada problema em todas as execuções.

Na Figura 5.4, estão representados os resultados (índice de acertos) dos *ensembles* obtidos com cada técnica de seleção e combinação utilizadas neste trabalho, ao longo das dez execuções do algoritmo, para o conjunto de validação do problema *Artificial*. Nestas figuras, cada gráfico corresponde a uma técnica de seleção de componentes e cada linha em cada gráfico a uma das técnicas de combinação empregadas aqui. Além disso, foram inseridas também em cada gráfico curvas denominadas "*Oráculo*". Estas curvas-oráculo são geradas apenas para fins de comparação, e seu índice de acertos é obtido verificando-se, para cada amostra, se pelo menos um dos componentes do *ensemble* tem em sua saída o resultado correto da classificação. Caso tenha, considera-se isso como um "acerto" e caso contrário, como um erro.

O objetivo dessas curvas-oráculo é ilustrar o desempenho que poderia ser obtido por um *ensemble* criado a partir de uma técnica de combinação ideal, capaz de levá-lo a acertar o resultado da classificação de amostras que fossem corretamente classificadas por pelo menos um de seus componentes.

Como pode ser observado nesses gráficos, não é possível identificar claramente um par de técnicas de seleção e combinação de componentes que apresente um desempenho claramente superior aos demais em todas as execuções do algoritmo. Este comportamento não foi observado apenas para o problema *Artificial*. Como pode ser visto nas Figuras 5.5, 5.6 e 5.7, isto se repetiu para os demais conjuntos de dados tratados neste trabalho.

O sub-conjunto de validação está disponível na etapa de construção do *ensemble*, o que nos permite utilizá-lo para selecionar o melhor par de técnicas de seleção e combinação para um dado conjunto de indivíduos gerado na etapa de treinamento, ou seja, é possível selecionar o melhor par de técnicas de seleção e combinação que, aplicados ao conjunto de classificadores disponível, resultará no *ensemble* de melhor desempenho. Neste trabalho, foi feita, a cada execução, uma busca exaustiva dentre todos os pares de técnicas de seleção e combinação aqui empregadas, e aquele capaz de ge-

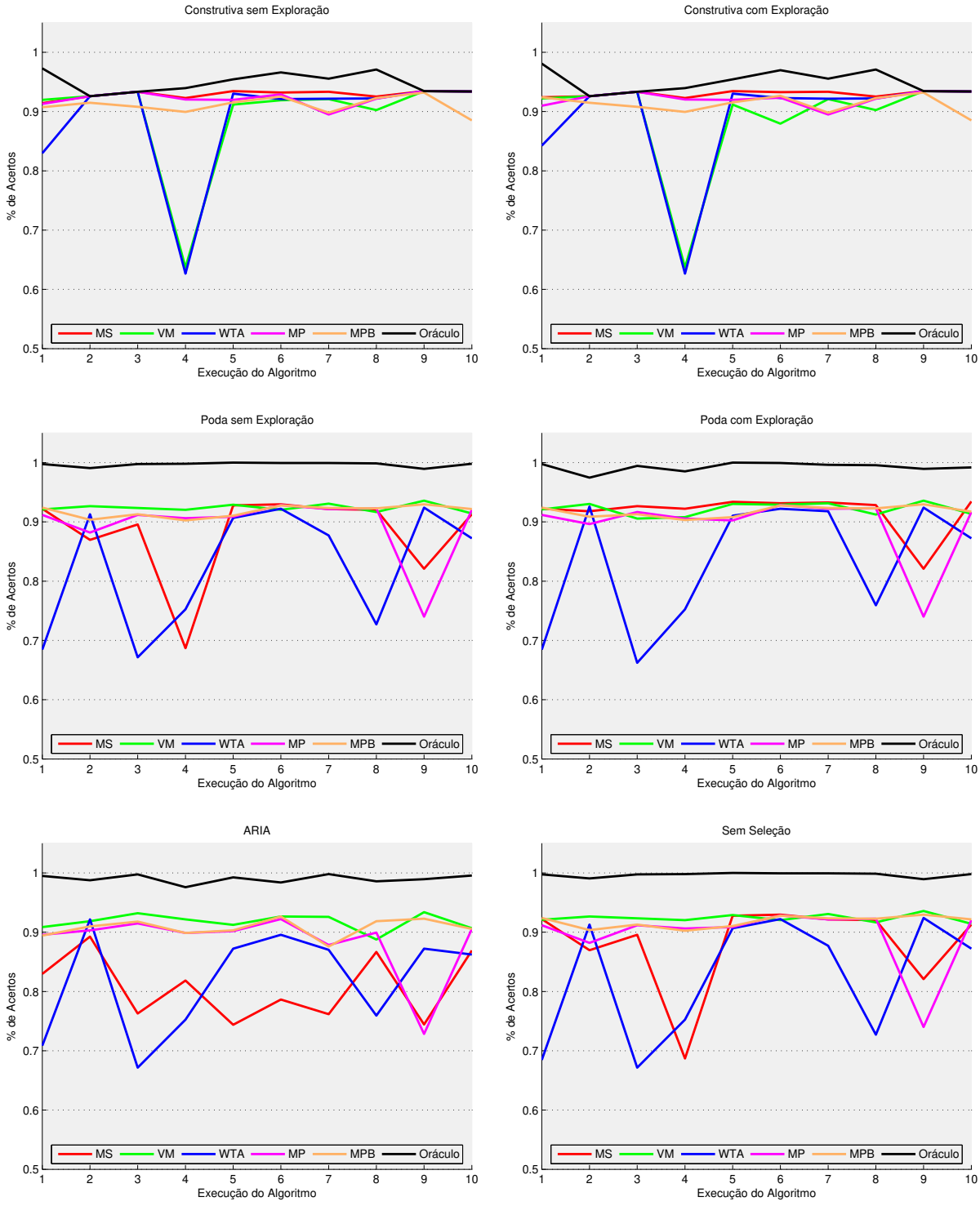


Fig. 5.4: Gráficos de desempenho dos *ensembles* ao longo das dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Artificial*. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

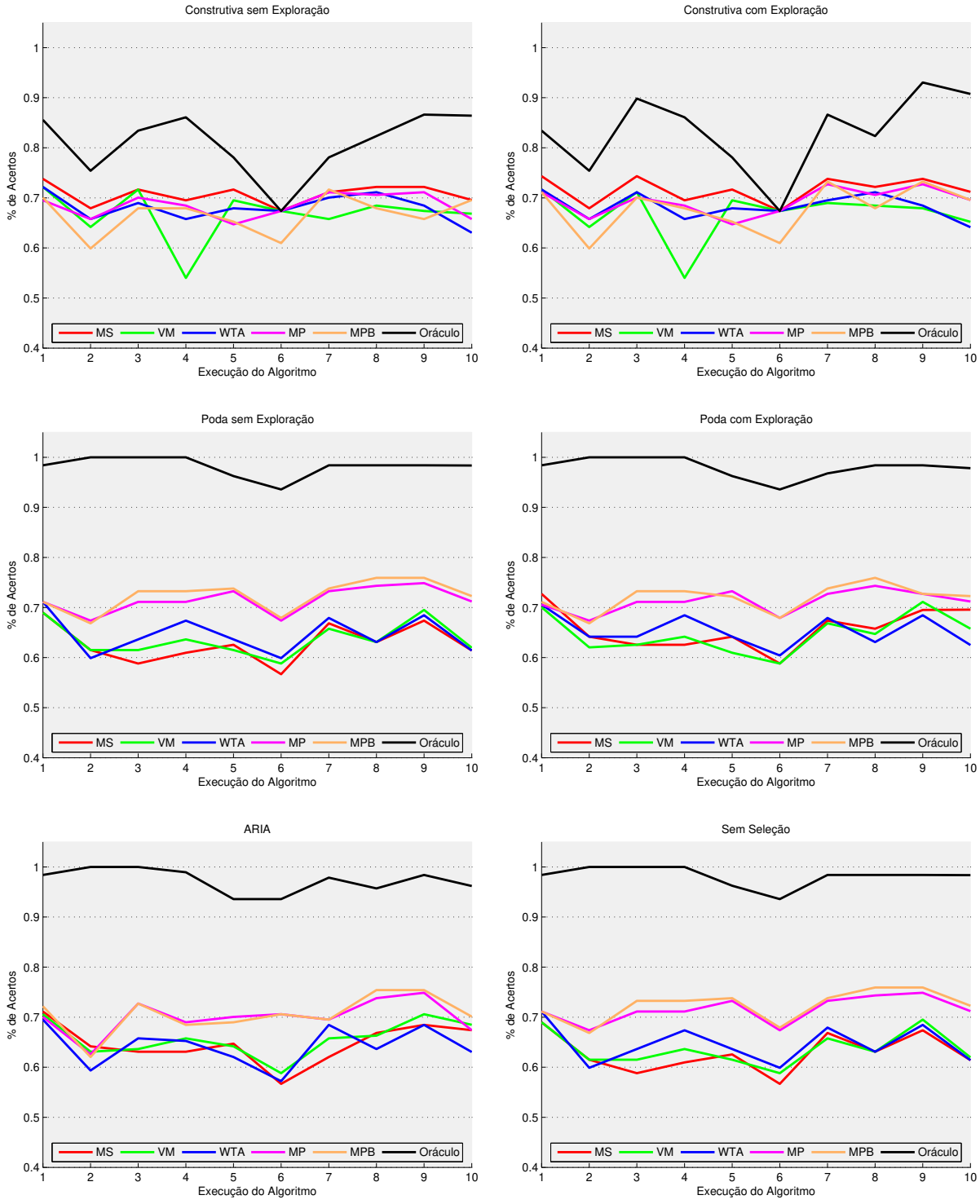


Fig. 5.5: Gráficos de desempenho dos *ensembles* ao longo das dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Bupa*. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

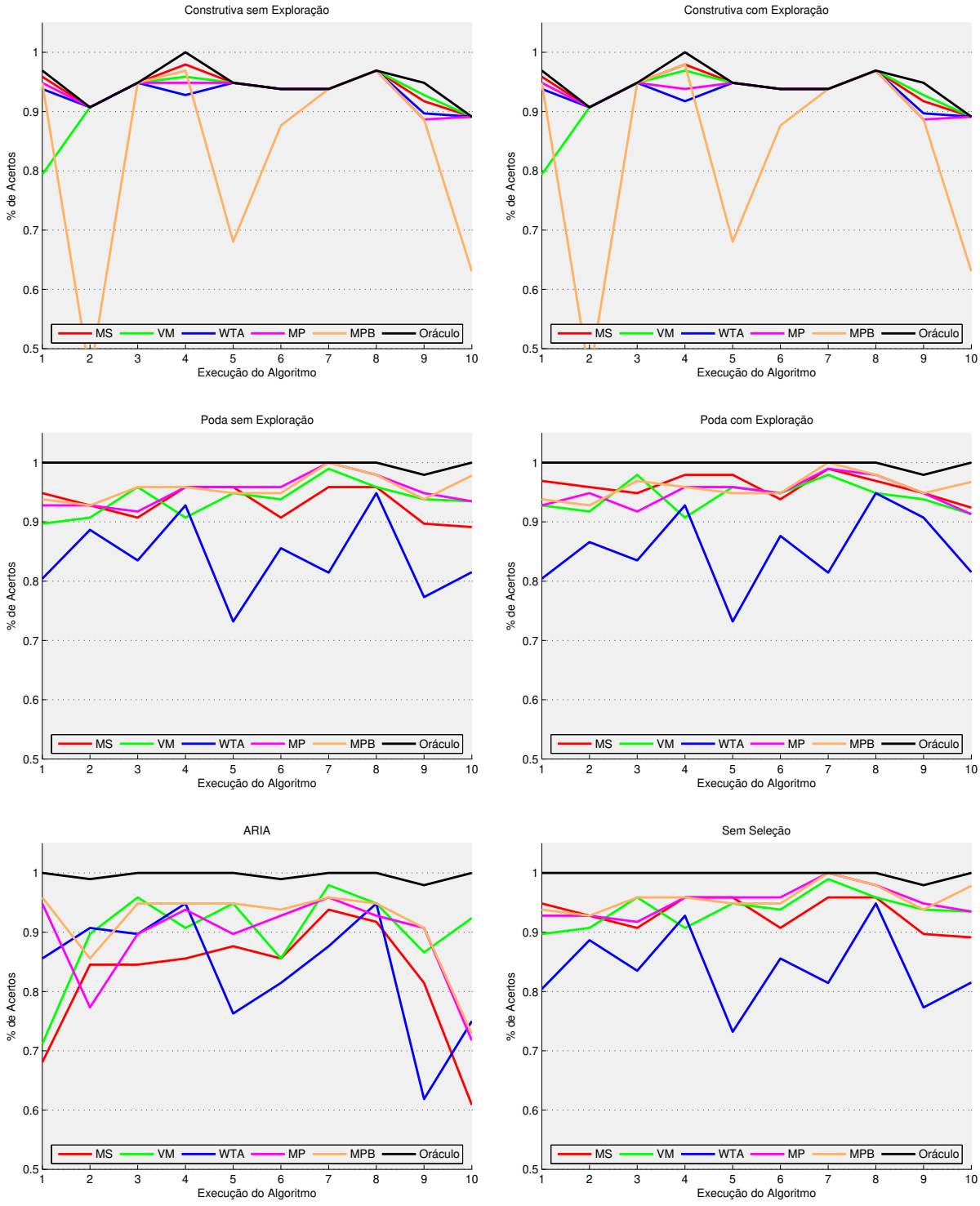


Fig. 5.6: Gráficos de desempenho dos *ensembles* ao longo das dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Wine*. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

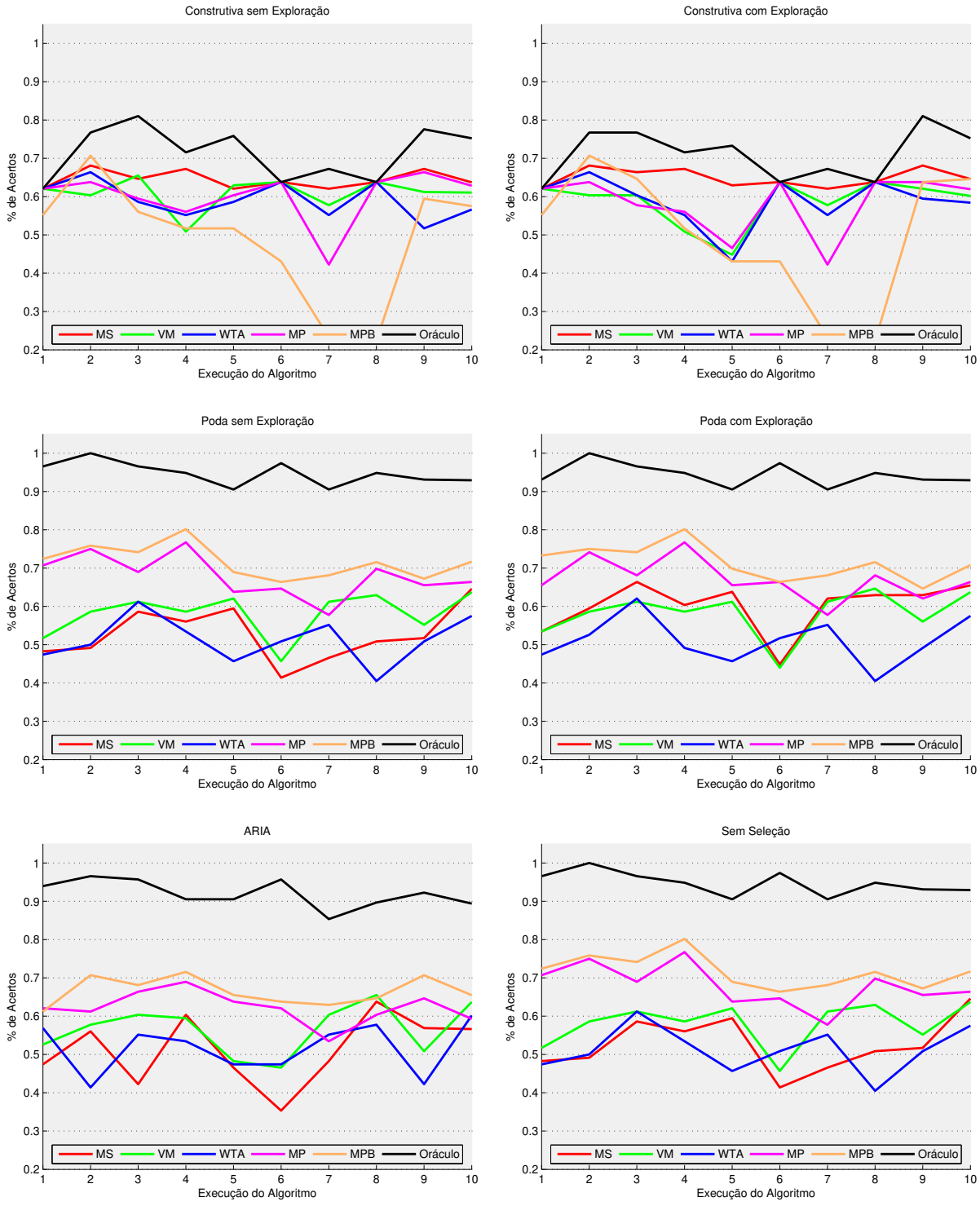


Fig. 5.7: Gráficos de desempenho dos *ensembles* ao longo das dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Glass*. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

rar o melhor *ensemble* para o sub-conjunto de validação, ao ser aplicado ao grupo de classificadores disponíveis, será considerado o par de técnicas vencedoras. Esta seleção a partir da busca exaustiva baseada no desempenho com relação ao sub-conjunto de validação se baseia na hipótese de que o sub-conjunto de validação seja capaz de representar as características do problema como um todo, da mesma maneira que os sub-conjuntos de teste e treinamento. Dessa forma, sustenta-se que os resultados obtidos para o sub-conjunto de validação sejam um indicador claro do que ocorrerá com o conjunto de testes.

Ainda com relação a essa busca exaustiva, pode parecer à primeira vista que seu custo computacional seja elevado. Mas, na verdade, a maior parte do custo computacional total envolvido na construção dos *ensembles* está na etapa de geração dos componentes, feita pelo algoritmo *opt-aiNet*. Em outras abordagens em que a etapa de geração não seja muito custosa, ou em que se tenha um número grande de técnicas de seleção e combinação de componentes a serem analisadas (o que levaria à explosão combinatória da busca exaustiva), seria interessante o desenvolvimento de alguma heurística para esta seleção do melhor par de técnicas, de forma a obter uma solução boa com um custo computacional relativamente baixo.

A fim de verificar se esta dependência entre o desempenho das técnicas de seleção e combinação e o conjunto de classificadores candidatos a constituírem o *ensemble* está relacionada à métrica de diversidade adotada nesta proposta, a métrica de afinidade entre anticorpos, que é a própria medida de diversidade par-a-par, utilizada na etapa de supressão do algoritmo *opt-aiNet*, foi substituída pela *estatística Q*, descrita da Seção 2.1, e os experimentos foram executados novamente. Nesta nova situação, deseja-se que o valor da estatística Q entre dois indivíduos seja o menor possível, diferentemente da métrica baseada em distância de Hamming, que deveria ser a maior possível. Dessa forma, o algoritmo foi modificado para suprimir o indivíduo de pior *fitness* dentre dois que possuam uma afinidade entre si *superior* a um dado limiar de supressão. Os novos limiares de supressão adotados estão representados na Tabela 5.6. Como no caso da distância de Hamming, estes parâmetros foram determinados empiricamente de forma a manter um nível de supressão de indivíduos que não elimine todos os elementos da população, exceto o melhor a cada etapa de supressão, mas que também evite que nenhum indivíduo seja eliminado.

Tab. 5.6: Limiares de supressão adotados nos experimentos com estatística Q como critério de supressão.

	Artificial	Bupa	Wine	Glass
Limiar de Supressão	0.0010	0.0075	0.0140	0.0120

Os resultados encontrados para as simulações utilizando a estatística Q como critério de supressão estão representados nas Figuras 5.8, 5.9, 5.10 e 5.11, para os problemas *Artificial*, *Bupa*, *Wine* e *Glass*,

respectivamente.

Analisando os resultados nas Figuras 5.8, 5.9, 5.10 e 5.11, podemos observar que a ausência de um par de técnicas de seleção e combinação que apresente o melhor desempenho para cada problema se repete aqui, mas de maneira mais sutil. Para o problema *Artificial*, existe um grande número de pares de técnicas de seleção e combinação que apresentam um desempenho semelhante ao longo das dez execuções, com diferenças de desempenho muito pequenas entre si na maioria das execuções, o que já não ocorre para os demais problemas, onde a alternância entre pares vencedores se dá de maneira mais veemente. Para o problema *Glass*, podemos perceber uma tendência de vitória da técnica de combinação por média ponderada com *bias*, quando associada a técnicas de seleção que não sejam construtivas. No entanto ainda há uma pequena alternância entre as técnicas de seleção vitoriosas para este problema.

Estes resultados nos levam a concluir que a métrica de diversidade influi sim na dependência entre o desempenho das técnicas de seleção e combinação e o conjunto de elementos disponíveis para serem utilizados por estas técnicas. No entanto, mais estudos são necessários para se verificar como e por que essa influência ocorre.

Essa dependência entre a métrica de diversidade e a influência do conjunto de classificadores junto ao desempenho das técnicas de seleção e combinação possivelmente está relacionada "à quantidade de informação" sobre os erros de cada componente que cada métrica de diversidade utiliza. Para exemplificar, tomemos as duas métricas empregadas neste trabalho: a estatística Q (dada pela Expressão 2.1) utiliza informação sobre o número de amostras em que ambos os classificadores acertam, em que ambos erram, em que o primeiro acerta e o segundo erra, e vice-versa; enquanto que a métrica baseada na distância de Hamming dos vetores de acertos leva em consideração apenas a informação do número de amostras em que o primeiro classificador acerta e o segundo erra e vice-versa. Dessa forma, a segunda métrica é menos detalhada na indicação de diversidade dos indivíduos gerados, uma vez que desconsidera parte da informação utilizada pela estatística Q, que pode ser relevante para o desempenho das técnicas de seleção e combinação de componentes e, assim, levar a uma maior variabilidade do desempenho de cada técnica de acordo com a população de classificadores candidatos disponíveis.

5.5 Desempenho Geral da Proposta

Nesta seção, serão apresentados os resultados finais para os *ensembles*, aplicados ao sub-conjunto de dados de teste. Todos os resultados apresentados corresponderão aos valores médios e seus respectivos desvios padrões, obtidos ao longo das dez execuções do algoritmo.

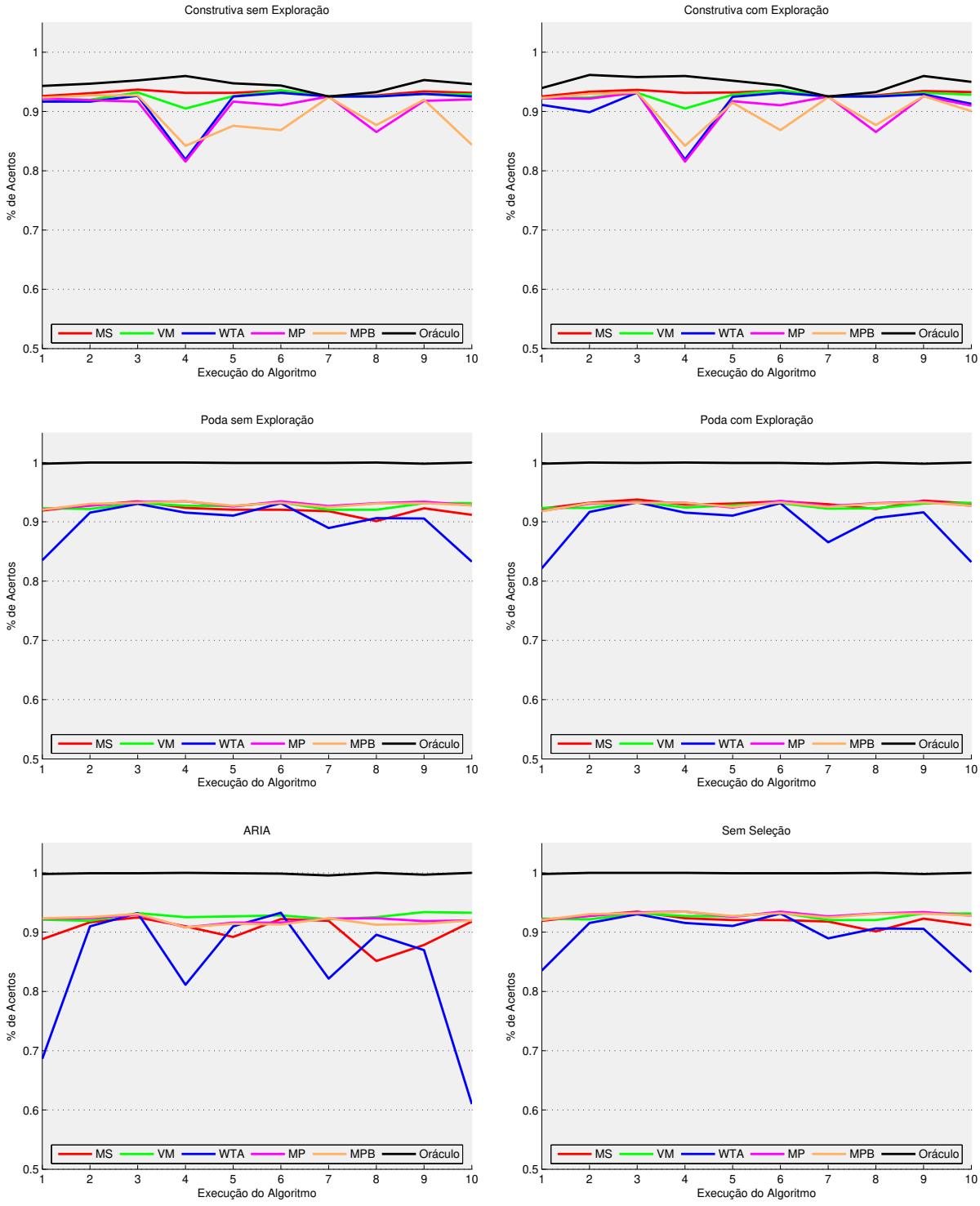


Fig. 5.8: Gráficos de desempenho dos *ensembles* ao longo de dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Artificial*, utilizando-se a estatística Q como métrica de diversidade. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

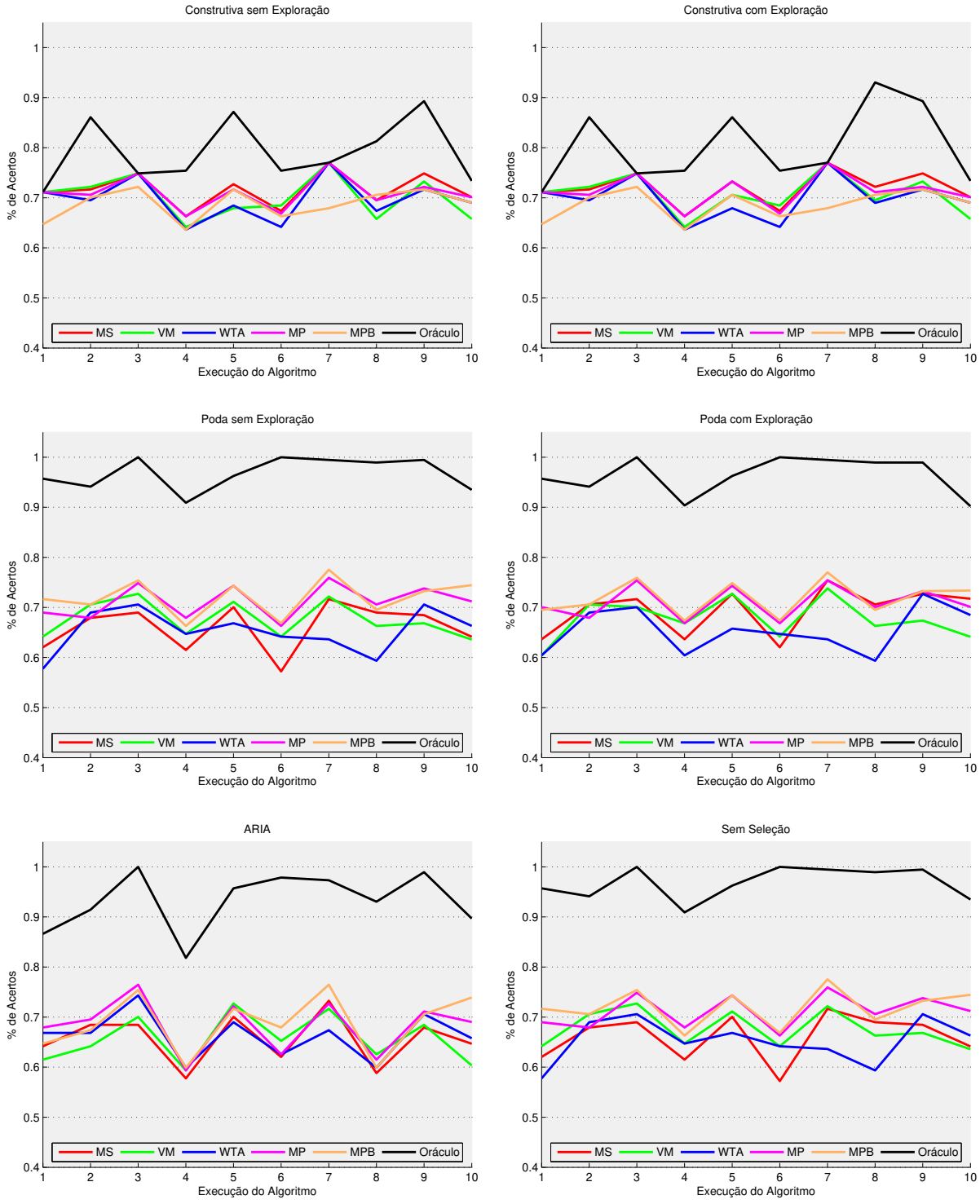


Fig. 5.9: Gráficos de desempenho dos *ensembles* ao longo de dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Bupa*, utilizando-se a estatística Q como métrica de diversidade. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

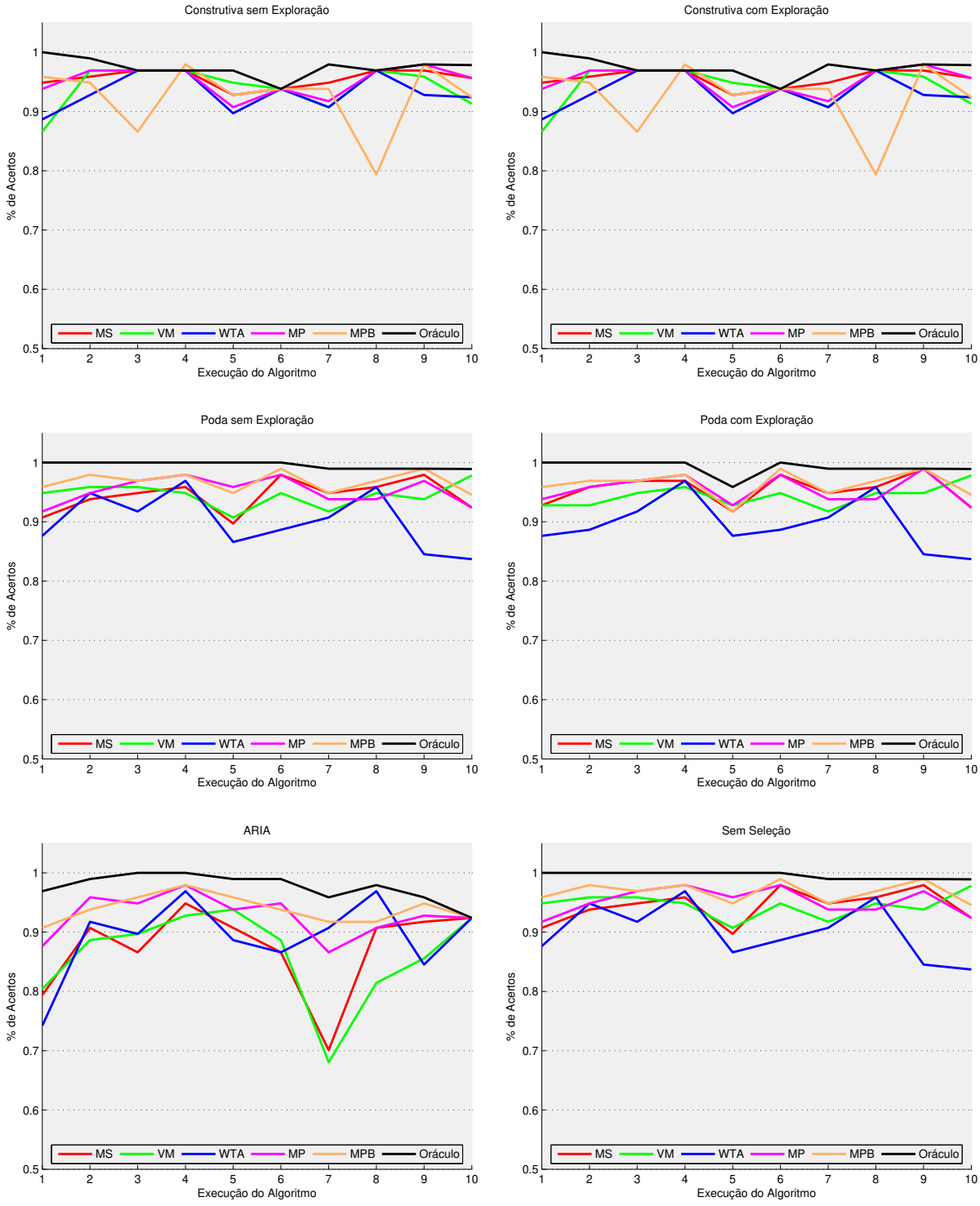


Fig. 5.10: Gráficos de desempenho dos *ensembles* ao longo de dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Wine*, utilizando-se a estatística Q como métrica de diversidade. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

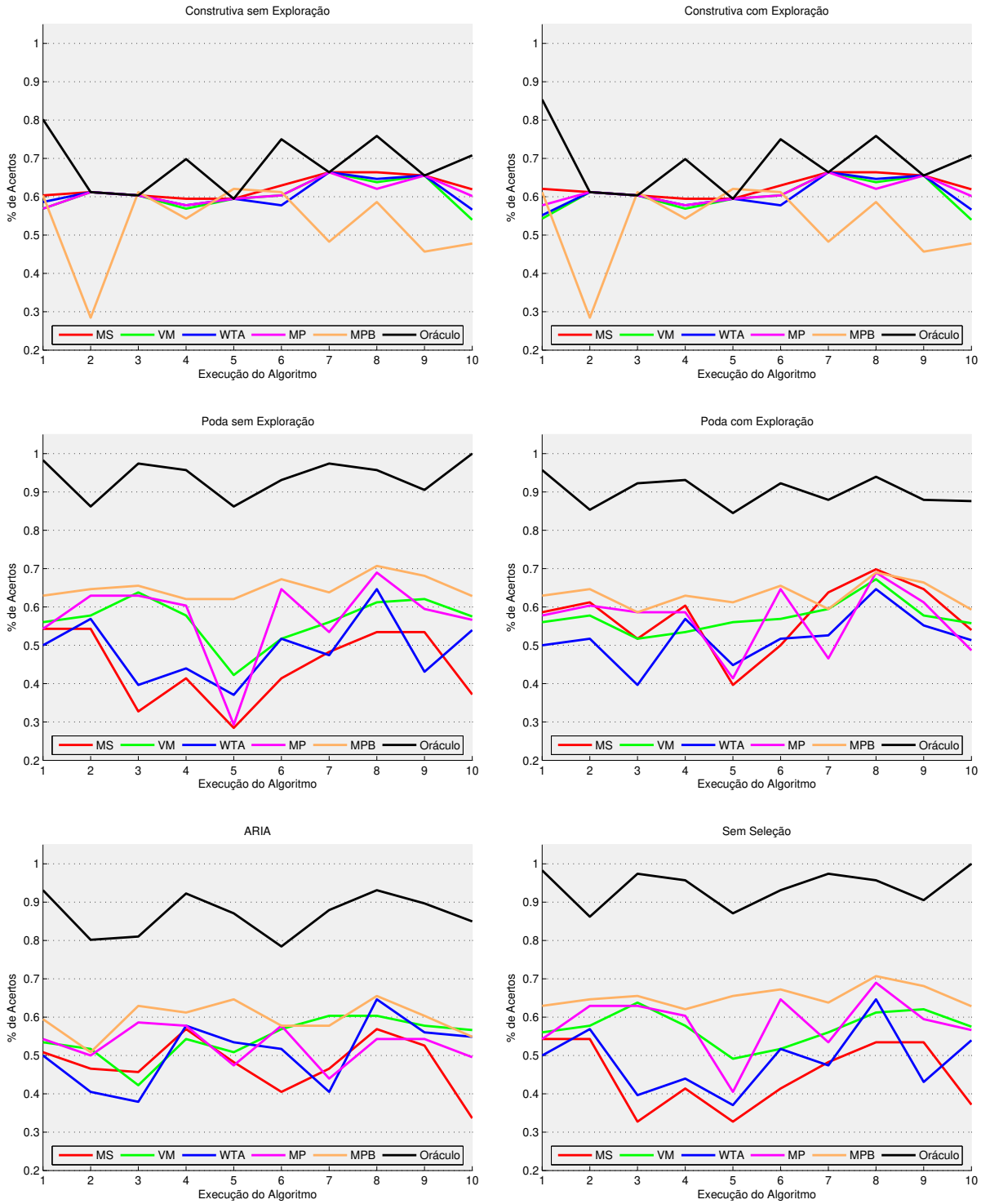


Fig. 5.11: Gráficos de desempenho dos *ensembles* ao longo de dez execuções do algoritmo para o sub-conjunto de dados de validação do problema *Glass*, utilizando-se a estatística Q como métrica de diversidade. Cada curva dos gráficos corresponde a uma das técnicas de combinação de componentes empregadas neste trabalho, além da chamada curva *Oráculo*.

5.5.1 Análise do Índice de Acerto dos *Ensembles*

As Tabelas 5.7, 5.8, 5.9 e 5.10, e as Figuras 5.12, 5.13, 5.14 e 5.15 contêm os índices de acertos médios (média do número de classificações corretas dividido pelo número de amostras) e seus respectivos desvios padrões, para os *ensembles* obtidos com cada par de técnicas de seleção e combinação adotados neste trabalho, para o melhor classificador individual e para o melhor *ensemble* obtido via busca exaustiva[2], aplicados aos **sub-conjuntos de teste** dos problemas *Artificial*, *Bupa*, *Wine* e *Glass*, respectivamente. No entanto, é importante ressaltar aqui que a seleção do melhor classificador individual e a busca exaustiva pelo melhor *ensemble* foram feitas baseadas no desempenho individual dos classificadores e de cada técnica de seleção e combinação junto ao **sub-conjunto de dados de validação**.

Tab. 5.7: Índice médio de classificações corretas dos *ensembles* obtidos com cada par de técnicas de seleção e combinação de componentes, para o sub-conjunto de testes do problema *Artificial*. São apresentados também os resultados do melhor classificador individual e do melhor *ensemble* obtido via busca exaustiva. Os resultados estão apresentados na forma (média) $\pm$ (desvio padrão).

Método de Seleção	Método de Combinação	Índice de Acertos	Método de Seleção	Método de Combinação	Índice de Acertos
Cw/oE	MS	0.92 ± 0.02	PwE	MS	0.91 ± 0.05
	MP	0.92 ± 0.02		MP	0.89 ± 0.07
	MPB	0.91 ± 0.03		MPB	0.92 ± 0.02
	VM	0.89 ± 0.09		VM	0.92 ± 0.02
	WTA	0.88 ± 0.10		WTA	0.84 ± 0.11
CwE	MS	0.92 ± 0.02	ARIA	MS	0.80 ± 0.07
	MP	0.91 ± 0.02		MP	0.89 ± 0.07
	MPB	0.91 ± 0.02		MPB	0.91 ± 0.02
	VM	0.88 ± 0.09		VM	0.92 ± 0.02
	WTA	0.88 ± 0.10		WTA	0.82 ± 0.09
Pw/oE	MS	0.87 ± 0.08	SS	MS	0.87 ± 0.08
	MP	0.89 ± 0.07		MP	0.89 ± 0.07
	MPB	0.92 ± 0.02		MPB	0.92 ± 0.02
	VM	0.92 ± 0.02		VM	0.92 ± 0.02
	WTA	0.82 ± 0.11		WTA	0.82 ± 0.11
Melhor Classificador: 0.92 ± 0.02					
Melhor *Ensemble* (busca exaustiva): 0.92 ± 0.01					

[2]Os resultados associados a esta *busca exaustiva* se referem à média e desvio padrão (junto aos sub-conjuntos de teste) dos índices de acerto dos melhores *ensembles* obtidos junto aos sub-conjuntos de validação, em cada uma das 10 execuções da metodologia proposta. É importante ressaltar que este resultado agrega o desempenho de *ensembles* construídos com diferentes técnicas de seleção e combinação de componentes.

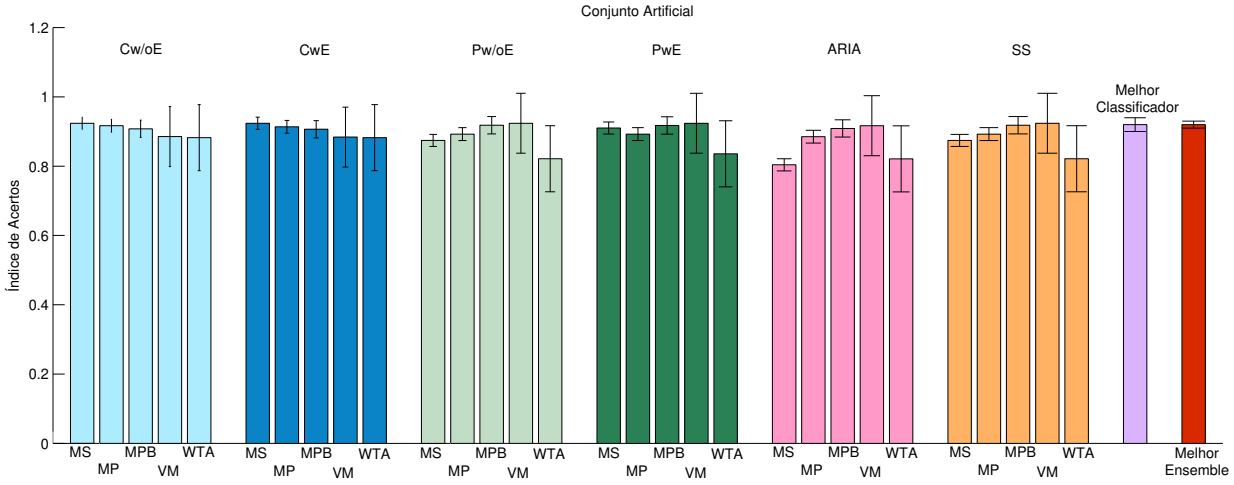


Fig. 5.12: Representação gráfica dos resultados apresentados na Tabela 5.7, para o problema *Artificial*. Cada coluna corresponde ao índice médio de acertos, enquanto que as barras em seus topos correspondem aos respectivos desvios padrões.

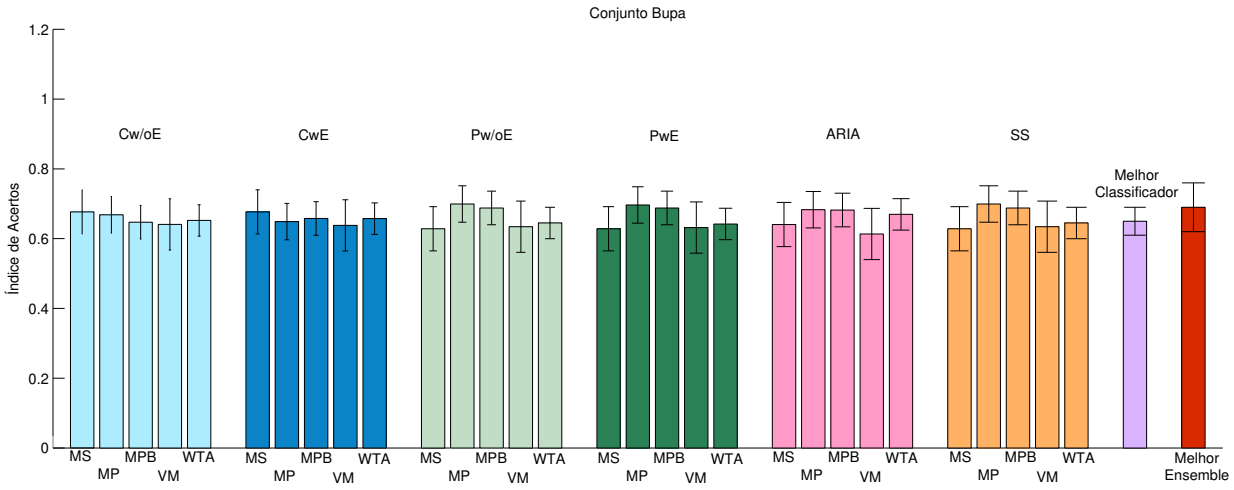


Fig. 5.13: Representação gráfica dos resultados apresentados na Tabela 5.8, para o problema *Bupa*. Cada coluna corresponde ao índice médio de acertos, enquanto que as barras em seus topos correspondem aos respectivos desvios padrões.

Dos resultados apresentados nas Tabelas 5.7, 5.8, 5.9 e 5.10, podemos observar alguns aspectos para cada base de dados utilizada.

Problema Artificial

Para o problema *Artificial*, não existe nenhum *ensemble* capaz de apresentar desempenho médio superior a 0.92 junto aos casos de teste, que corresponde ao desempenho do melhor indivíduo se-

Tab. 5.8: Índice médio de classificações corretas dos *ensembles* obtidos com cada par de técnicas de seleção e combinação de componentes, para o sub-conjunto de testes do problema *Bupa*. São apresentados também os resultados do melhor classificador individual e do melhor *ensemble* obtido via busca exaustiva.

Método de Seleção	Método de Combinação	Índice de Acertos	Método de Seleção	Método de Combinação	Índice de Acertos
Cw/oE	MS	0.68 ± 0.06	PwE	MS	0.63 ± 0.07
	MP	0.67 ± 0.08		MP	0.70 ± 0.06
	MPB	0.65 ± 0.06		MPB	0.69 ± 0.06
	VM	0.64 ± 0.08		VM	0.63 ± 0.07
	WTA	0.65 ± 0.04		WTA	0.64 ± 0.09
CwE	MS	0.66 ± 0.04	ARIA	MS	0.64 ± 0.07
	MP	0.65 ± 0.05		MP	0.68 ± 0.09
	MPB	0.66 ± 0.05		MPB	0.68 ± 0.09
	VM	0.64 ± 0.07		VM	0.61 ± 0.05
	WTA	0.66 ± 0.04		WTA	0.67 ± 0.07
Pw/oE	MS	0.63 ± 0.07	SS	MS	0.63 ± 0.07
	MP	0.70 ± 0.05		MP	0.70 ± 0.05
	MPB	0.69 ± 0.05		MPB	0.69 ± 0.05
	VM	0.63 ± 0.07		VM	0.63 ± 0.07
	WTA	0.65 ± 0.09		WTA	0.65 ± 0.09
Melhor Classificador: 0.65 ± 0.04					
Melhor *Ensemble* (busca exaustiva): 0.69 ± 0.07					

lecionado a partir dos dados de validação. Além disso, com a exceção de alguns casos, a maioria dos *ensembles* apresenta desempenhos muito próximos entre si, e o desempenho do melhor *ensemble*, obtido via busca exaustiva com base nos dados de validação, não é superior a outros *ensembles* representados na tabela.

Comparando-se as técnicas de seleção de componentes com a estratégia sem seleção, esta última foi superada apenas pelas técnicas construtivas (com e sem exploração) nos casos de combinação por média simples, ponderada sem *bias* e *winner-takes-all*, e pela técnica de poda com exploração para os casos de combinação por média simples e *winner-takes-all*. Para as outras técnicas de combinação de componentes, a estratégia sem seleção apresentou desempenho superior ou igual ao das demais técnicas de seleção.

Considerando-se cada técnica de combinação de componentes, a técnica de seleção construtiva sem exploração apresentou o melhor desempenho junto a três delas (Média Simples, Média ponderada sem *bias* e *winner-takes-all*), enquanto que as técnicas construtiva com exploração, poda (com e sem exploração), e sem seleção apresentam os melhores desempenhos junto a duas técnicas de

Tab. 5.9: Índice médio de classificações corretas dos *ensembles* obtidos com cada par de técnicas de seleção e combinação de componentes, para o sub-conjunto de testes do problema *Wine*. São apresentados também os resultados do melhor classificador individual e do melhor *ensemble* obtido via busca exaustiva.

Método de Seleção	Método de Combinação	Índice de Acertos	Método de Seleção	Método de Combinação	Índice de Acertos
Cw/oE	MS	0.89 ± 0.06	PwE	MS	0.92 ± 0.07
	MP	0.88 ± 0.06		MP	0.93 ± 0.06
	MPB	0.80 ± 0.19		MPB	0.92 ± 0.06
	VM	0.88 ± 0.04		VM	0.94 ± 0.06
	WTA	0.88 ± 0.05		WTA	0.86 ± 0.05
CwE	MS	0.89 ± 0.06	ARIA	MS	0.86 ± 0.13
	MP	0.89 ± 0.06		MP	0.90 ± 0.10
	MPB	0.81 ± 0.19		MPB	0.88 ± 0.10
	VM	0.88 ± 0.04		VM	0.91 ± 0.04
	WTA	0.87 ± 0.06		WTA	0.86 ± 0.08
Pw/oE	MS	0.95 ± 0.06	SS	MS	0.95 ± 0.03
	MP	0.93 ± 0.07		MP	0.93 ± 0.07
	MPB	0.92 ± 0.06		MPB	0.92 ± 0.05
	VM	0.94 ± 0.05		VM	0.94 ± 0.05
	WTA	0.85 ± 0.06		WTA	0.85 ± 0.06
Melhor Classificador: 0.89 ± 0.05					
Melhor *Ensemble* (busca exaustiva): 0.95 ± 0.03					

combinação cada.

Com relação aos métodos de combinação, considerando-se cada técnica de seleção, o voto majoritário apresentou o melhor desempenho para quatro dessas técnicas (poda com e sem exploração, sem seleção e ARIA), enquanto que o método de média ponderada com *bias* apresentou o melhor desempenho junto às técnicas de poda (com e sem exploração) e sem seleção.

No entanto, é importante ressaltar que, quando combinadas, as duas técnicas vencedoras individualmente para o maior número de casos (seleção construtiva sem exploração e combinação por voto majoritário) apresentaram um desempenho médio inferior ao resultado obtido por outros pares de técnicas utilizadas neste trabalho.

Problema *Bupa*

Já para o problema *Bupa*, podemos observar alguns *ensembles* com valores médios um pouco superiores aos do melhor classificador individual. Além disso, o melhor *ensemble* (obtido via busca exaustiva empregando o conjunto de validação) também supera o melhor classificador individual,

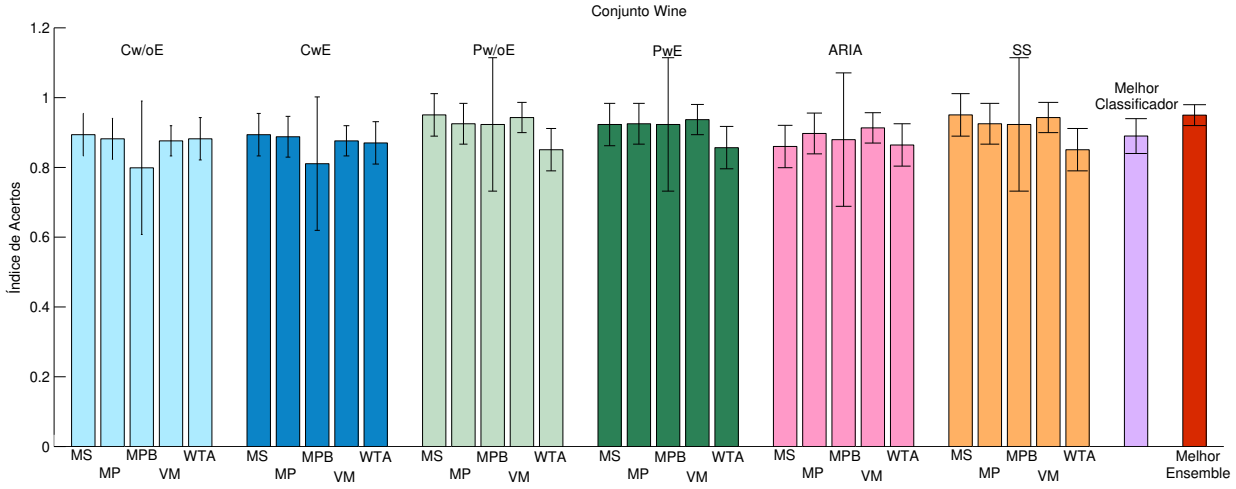


Fig. 5.14: Representação gráfica dos resultados apresentados na Tabela 5.9, para o problema *Wine*. Cada coluna corresponde ao índice médio de acertos, enquanto que as barras em seus topos correspondem aos respectivos desvios padrões.

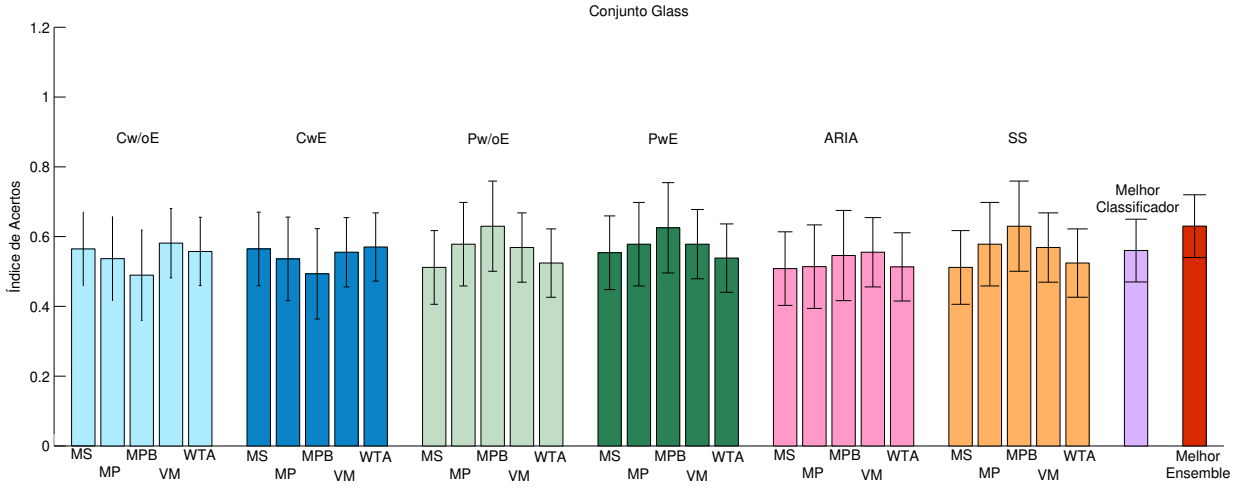


Fig. 5.15: Representação gráfica dos resultados apresentados na Tabela 5.10, para o problema *Glass*. Cada coluna corresponde ao índice médio de acertos, enquanto que as barras em seus topos correspondem aos respectivos desvios padrões.

quando ambos são avaliados junto ao conjunto de teste. Mas, para este problema, o *ensemble* obtido via busca exaustiva chega a ser superado por alguns *ensembles* obtidos via técnicas de poda e sem seleção, e com combinação por média ponderada.

Comparando-se as técnicas de seleção de componentes com a estratégia sem seleção, esta última foi superada pela técnica construtiva sem exploração, nos casos de combinação por média simples e voto majoritário, e pelas técnicas construtiva com exploração e ARIA para os casos de combinação

Tab. 5.10: Índice médio de classificações corretas dos *ensembles* obtidos com cada par de técnicas de seleção e combinação de componentes, para o sub-conjunto de testes do problema *Glass*. São apresentados também os resultados do melhor classificador individual e do melhor *ensemble* obtido via busca exaustiva.

Método de Seleção	Método de Combinação	Índice de Acertos	Método de Seleção	Método de Combinação	Índice de Acertos
Cw/oE	MS	0.56 ± 0.11	PwE	MS	0.55 ± 0.11
	MP	0.54 ± 0.13		MP	0.58 ± 0.07
	MPB	0.49 ± 0.12		MPB	0.63 ± 0.09
	VM	0.58 ± 0.10		VM	0.58 ± 0.11
	WTA	0.56 ± 0.09		WTA	0.57 ± 0.11
CwE	MS	0.58 ± 0.07	ARIA	MS	0.51 ± 0.10
	MP	0.54 ± 0.12		MP	0.51 ± 0.07
	MPB	0.49 ± 0.13		MPB	0.55 ± 0.08
	VM	0.56 ± 0.10		VM	0.56 ± 0.10
	WTA	0.57 ± 0.10		WTA	0.51 ± 0.05
Pw/oE	MS	0.51 ± 0.12	SS	MS	0.51 ± 0.12
	MP	0.58 ± 0.07		MP	0.58 ± 0.07
	MPB	0.63 ± 0.07		MPB	0.63 ± 0.07
	VM	0.57 ± 0.12		VM	0.57 ± 0.12
	WTA	0.52 ± 0.09		WTA	0.52 ± 0.09
Melhor Classificador: 0.56 ± 0.09					
Melhor *Ensemble* (busca exaustiva): 0.63 ± 0.09					

por média simples e *winner-takes-all*. Para as outras técnicas de combinação de componentes, a estratégia sem seleção apresentou desempenho superior ou igual ao das demais técnicas de seleção.

Tomando-se cada técnica de combinação de componentes, as técnicas de seleção construtiva sem exploração, poda (com e sem exploração) e sem seleção, apresentaram o melhor desempenho junto a dois métodos de combinação cada uma, sendo média simples e voto majoritário no caso da técnica construtiva sem exploração, e média ponderada com e sem *bias* para as demais técnicas.

Com relação aos métodos de combinação, considerando-se cada técnica de seleção, a média ponderada sem *bias* apresentou o melhor desempenho para quatro dessas técnicas (poda com e sem exploração, sem seleção e ARIA), enquanto que os métodos de média simples e média ponderada com *bias* apresentaram o melhor desempenho junto a duas técnicas de seleção cada.

Problema *Wine*

Como no caso do problema *Bupa*, para o problema *Wine* (e *Glass*) também foram obtidos *ensembles* com desempenho superior ao do melhor classificador individual, mas que também apresentaram

desempenho médio equivalente ao do melhor *ensemble* obtido via busca exaustiva.

Comparando-se as técnicas de seleção de componentes com a estratégia sem seleção, esta última foi superada apenas pelas técnicas construtiva com e sem exploração, poda com exploração e ARIA, nos casos de combinação por *winner-takes-all*. Para todos os outros métodos de combinação de componentes, a estratégia sem seleção apresentou desempenho superior ou igual ao das demais técnicas de seleção.

Considerando-se cada um dos métodos de combinação de componentes, as técnicas de seleção por poda sem exploração e sem seleção apresentaram os melhores desempenhos junto a quatro métodos de combinação cada uma (média simples, médias ponderadas com e sem *bias* e voto majoritário), enquanto que a técnica de poda com exploração foi a melhor junto aos métodos de média ponderada com e sem *bias* e voto majoritário.

Com relação aos métodos de combinação, considerando-se cada técnica de seleção, os métodos de média simples e voto majoritário apresentaram os melhores desempenhos para três dessas técnicas, sendo as técnicas construtiva com exploração, poda sem exploração e sem seleção para a média simples; e construtiva sem exploração, poda com exploração e ARIA para o voto majoritário.

Problema *Glass*

Para o problema *Glass* também foram obtidos *ensembles* com desempenho superior ao do melhor classificador individual, mas que também apresentaram desempenho médio equivalente ao do melhor *ensemble* obtido via busca exaustiva.

Comparando-se as técnicas de seleção de componentes com a estratégia sem seleção, esta última foi superada pelas técnicas construtiva com e sem exploração e poda com exploração nos casos de combinação por média simples e *winner-takes-all*; e pelas técnicas construtiva sem exploração e poda com exploração para combinação por voto majoritário. Para os outros métodos de combinação de componentes, a estratégia sem seleção apresentou desempenho superior ou igual ao das demais técnicas de seleção.

Considerando-se cada um dos métodos de combinação de componentes, a técnica de seleção por poda com exploração foi a vencedora no maior número de casos, apresentando o melhor desempenho médio junto a quatro métodos de combinação (médias ponderadas com e sem *bias*, voto majoritário e *winner-takes-all*).

Com relação aos métodos de combinação, considerando-se cada técnica de seleção, o método de combinação por média ponderada com *bias* apresentou os melhores desempenhos médios para três dessas técnicas (poda com e sem exploração e sem seleção), e o voto majoritário os melhores resultados junto a duas técnicas de seleção (construtiva sem exploração e ARIA).

Como verificamos na Seção 5.4 que o melhor par de técnicas de seleção e combinação depende do conjunto de classificadores disponíveis, o fato de o melhor *ensemble* obtido via busca exaustiva baseada no sub-conjunto de validação não ser capaz de superar o desempenho médio dos demais *ensembles* quando aplicados ao sub-conjunto de testes indica que o sub-conjunto de validação nem sempre é capaz de representar as mesmas características do problema representadas pelo sub-conjunto de testes. Apenas para verificarmos esta hipótese, aplicamos o mesmo mecanismo de busca exaustiva descrito anteriormente, mas agora baseado na performance de cada *ensemble* com relação ao conjunto de dados de teste. Os melhores *ensembles* obtidos estão representados na Tabela 5.11.

Tab. 5.11: Melhores *ensembles* obtidos via busca exaustiva baseada no desempenho individual avaliado nos dados de teste.

Problema	Índice de Acertos
Artificial	0.93 ± 0.01
Bupa	0.76 ± 0.04
Wine	0.97 ± 0.02
Glass	0.69 ± 0.06

Comparando os resultados presentes na Tabela 5.11 com os valores nas Tabelas 5.7, 5.8, 5.9 e 5.10, podemos perceber que o melhor par de técnicas de seleção e combinação realmente varia a cada execução do algoritmo, uma vez que este novo melhor *ensemble* resultante é capaz de superar a performance média de todos os demais *ensembles* obtidos, sendo este ganho de desempenho mais significativo para os problemas *Bupa* e *Glass*.

Pelos resultados nas Tabelas 5.7, 5.8, 5.9, 5.10 e 5.11, podemos perceber também que este problema encontrado no particionamento dos conjuntos de dados é praticamente inexistente para o problema *Artificial*, já que a diferença dos resultados das buscas exaustivas feitas tanto no conjunto de validação quanto no de testes é insignificante, dados os desvios padrões dos resultados. Isto ocorre pois o problema *Artificial* apresenta 3000 amostras, sendo mil delas para cada classe, o que facilita a obtenção de sub-conjuntos representativos do problema.

A diferença entre os resultados para o problema *Wine* também é menor que a observada para os problemas *Bupa* e *Glass*, mas aqui o número de amostras é bem reduzido. No entanto, o problema *Wine* é reconhecidamente de fácil solução, como pode ser visto nos resultados presentes na literatura (um deles representado na Tabela 5.12), o que facilita a representação das características do problema com um número menor de amostras.

Na Tabela 5.12, estão apresentados alguns resultados descritos na literatura para os problemas tratados neste trabalho. Apesar de serem necessários alguns cuidados em relação a comparações com

resultados da literatura, uma vez que dificilmente dois trabalhos distintos seguem a mesma metodologia, ou usam classificadores/regressores com a mesma estrutura, ou se baseiam na mesma partição do conjunto de dados, elas são válidas para se verificar, sem muito rigor, como os resultados da abordagem sendo proposta se posicionam diante de outras propostas de solução que também empregam *ensembles* de redes neurais. Dessa forma, na Tabela 5.12, buscou-se apresentar resultados obtidos com *ensembles* de redes neurais artificiais (preferencialmente MLPs), de forma que pelo menos a estrutura dos classificadores fosse semelhante.

Tab. 5.12: Resultados da Literatura, melhores resultados obtidos pela metodologia proposta e resultados dos melhores *ensembles* obtidos através de busca exaustiva junto ao conjunto de testes, para os problemas *Bupa*, *Wine* e *Glass*.

Problema	Resultados da Literatura	Melhores Resultados Obtidos	Busca Exaustiva junto ao Conjunto de Teste
Bupa	0.697	0.70 ± 0.05	0.76 ± 0.04
Wine	0.97 ± 0.01	0.95 ± 0.03	0.97 ± 0.02
Glass	0.69 ± 0.01	0.63 ± 0.09	0.69 ± 0.06

Para o problema *Bupa*, os resultados apresentados na Tabela 5.12 (Rocha et al., 2005) correspondem a *ensembles* de redes neurais do tipo MLP, obtidos através de um algoritmo evolutivo. Para o problema *Wine* (Inoue & Narihisa, 2004), esses resultados correspondem a *ensembles* de *self-generating neural networks* e para o problema *Glass* (Lazarevic & Obradovic, 2001), também a *ensembles* de redes neurais MLP.

Comparando os resultados obtidos na literatura, apresentados na Tabela 5.12, com os obtidos pela metodologia proposta aqui, podemos perceber que os *ensembles* gerados aqui são competitivos com os resultados apresentados na literatura (apesar de seus desempenhos médios serem um pouco inferiores para os problemas *Wine* e *Glass*), uma vez que as diferenças entre os valores médios obtidos e os reportados na literatura estão dentro da faixa do desvio padrão.

Apesar de terem sido escolhidos resultados da literatura que foram obtidos com o uso de *ensembles* de redes neurais artificiais, existem diferenças significativas entre esses trabalhos e a metodologia proposta aqui. Para o problema *Bupa*, por exemplo, são utilizadas redes neurais do tipo MLP com número de neurônios na camada intermediária variável, além do particionamento dos dados ter sido feito de forma diferente (20% para teste, 30% para validação e 50% para treinamento). Para o problema *Wine*, o tipo de redes neurais utilizadas é diferente do usado aqui e, para o problema *Glass*, o número de neurônios na camada oculta é igual ao número de atributos do problema, ou seja, são utilizadas redes com 9 neurônios na camada intermediária. Isso sem considerar as diferentes propostas de geração dos *ensembles*. Dessa forma, alguma diferença nos resultados era esperada.

Se considerarmos os melhores *ensembles* obtidos via busca exaustiva junto ao conjunto de testes (também representados na Tabela 5.12), podemos perceber que os valores médios para os problemas *Wine* e *Glass* se igualam aos apresentados na literatura e, para o problema *Bupa*, o índice de acertos obtido para o melhor *ensemble* chega a superar o resultado da literatura. No entanto, os resultados desses *ensembles* obtidos via busca exaustiva junto ao conjunto de testes devem ser vistos como um resultado potencial que a metodologia proposta aqui poderia ter atingido, caso a estratégia de particionamento dos dados utilizada tivesse conseguido obter sub-conjuntos representativos das características dos problemas em questão.

5.5.2 Análise do Número de Componentes e Diversidade

A Figura 5.16 apresenta o número médio de componentes nos *ensembles*, enquanto a Figura 5.17 apresenta o valor médio da métrica de diversidade para os problemas *Artificial*, *Bupa*, *Wine* e *Glass*, de acordo com cada uma das técnicas de seleção adotadas neste trabalho, e para a estratégia sem seleção.

Das Figuras 5.16 e 5.17, podemos observar que as técnicas construtivas tendem a gerar *ensembles* com um menor número de componentes, mais similares entre si, do que as técnicas de poda e o algoritmo de agrupamento ARIA. No entanto, esse menor número de componentes e a menor diversidade desses componentes resultou em uma menor eficiência das técnicas construtivas quanto à precisão na classificação, uma vez que, como pode ser observado nas tabelas 5.7, 5.8, 5.9 e 5.10, essas técnicas foram responsáveis pela geração de *ensembles* que levaram aos melhores resultados apenas para o problema *Artificial*.

O fato das técnicas construtivas levarem a *ensembles* com menor número de componentes e menor diversidade pode ser explicado pelas diferenças inerentes a cada metodologia. As técnicas construtivas começam com apenas um componente no *ensemble* (nenhuma diversidade), e vão incrementando o número de elementos e, conseqüentemente, a diversidade, até atingirem seus critérios de parada. Já as técnicas de poda funcionam de maneira oposta, ou seja, no início o *ensemble* é constituído por todos os componentes (alta diversidade), e este número vai diminuindo ao longo da execução, o que pode levar a uma redução da diversidade. O método baseado no algoritmo ARIA, por sua vez, também tende a gerar um *ensemble* com componentes mais diversos, uma vez que é selecionado um elemento por cada grupo e, quanto mais diversos forem dois indivíduos, maiores serão as chances de pertencerem a grupos diferentes.

A Figura 5.17 também nos mostra que a metodologia proposta atingiu um bom nível de diversidade entre os componentes dos *ensembles*, o que justifica o uso de *ensembles* homogêneos (somente redes neurais MLP) e o foco restrito à manipulação dos pesos das redes neurais. A diversidade obtida com a metodologia proposta é resultado do tratamento simultâneo de múltiplas soluções e não

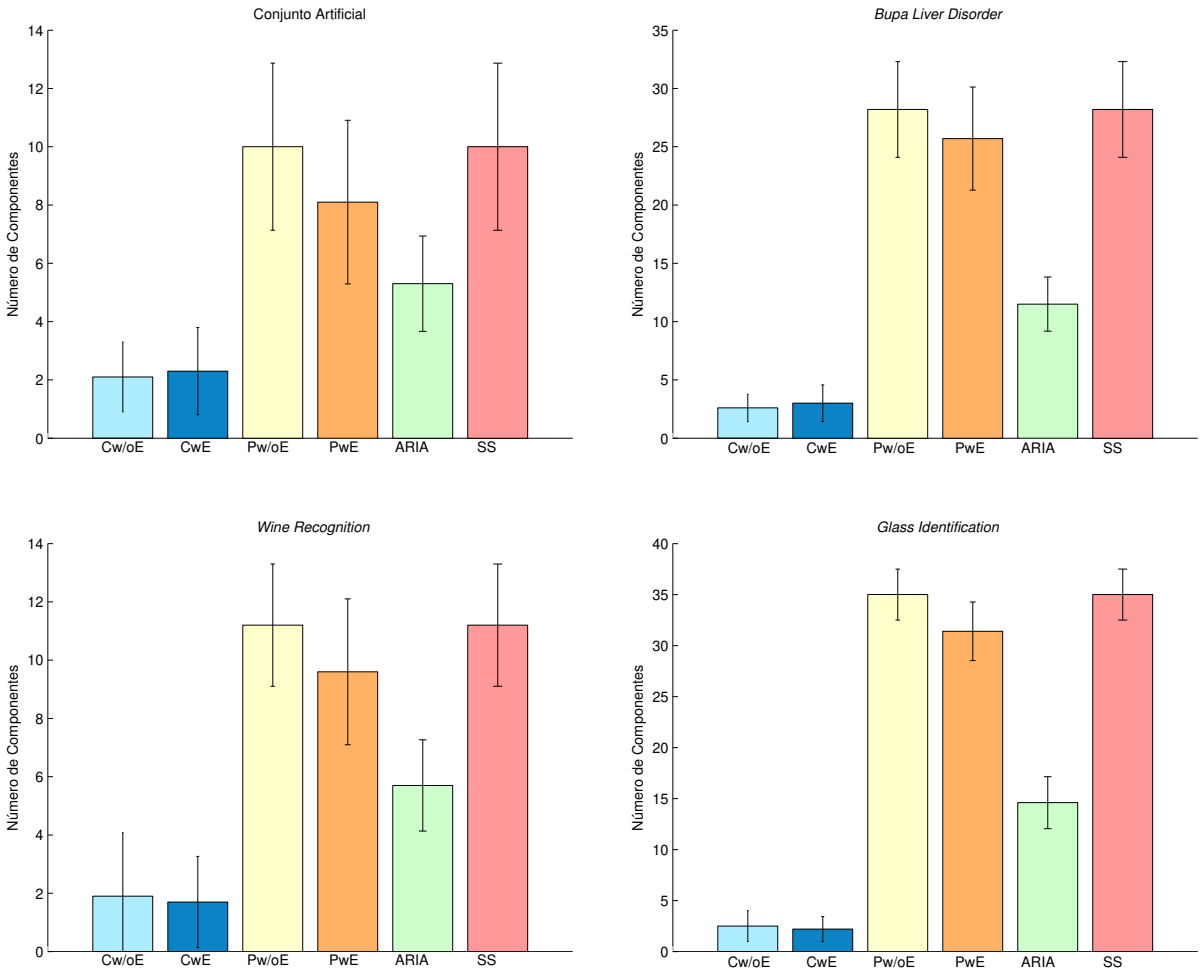


Fig. 5.16: Número médio de componentes nos *ensembles* de acordo com as técnicas de seleção. Cada coluna corresponde ao valor médio do número de componentes e as barras em seus topos correspondem aos respectivos desvios padrões.

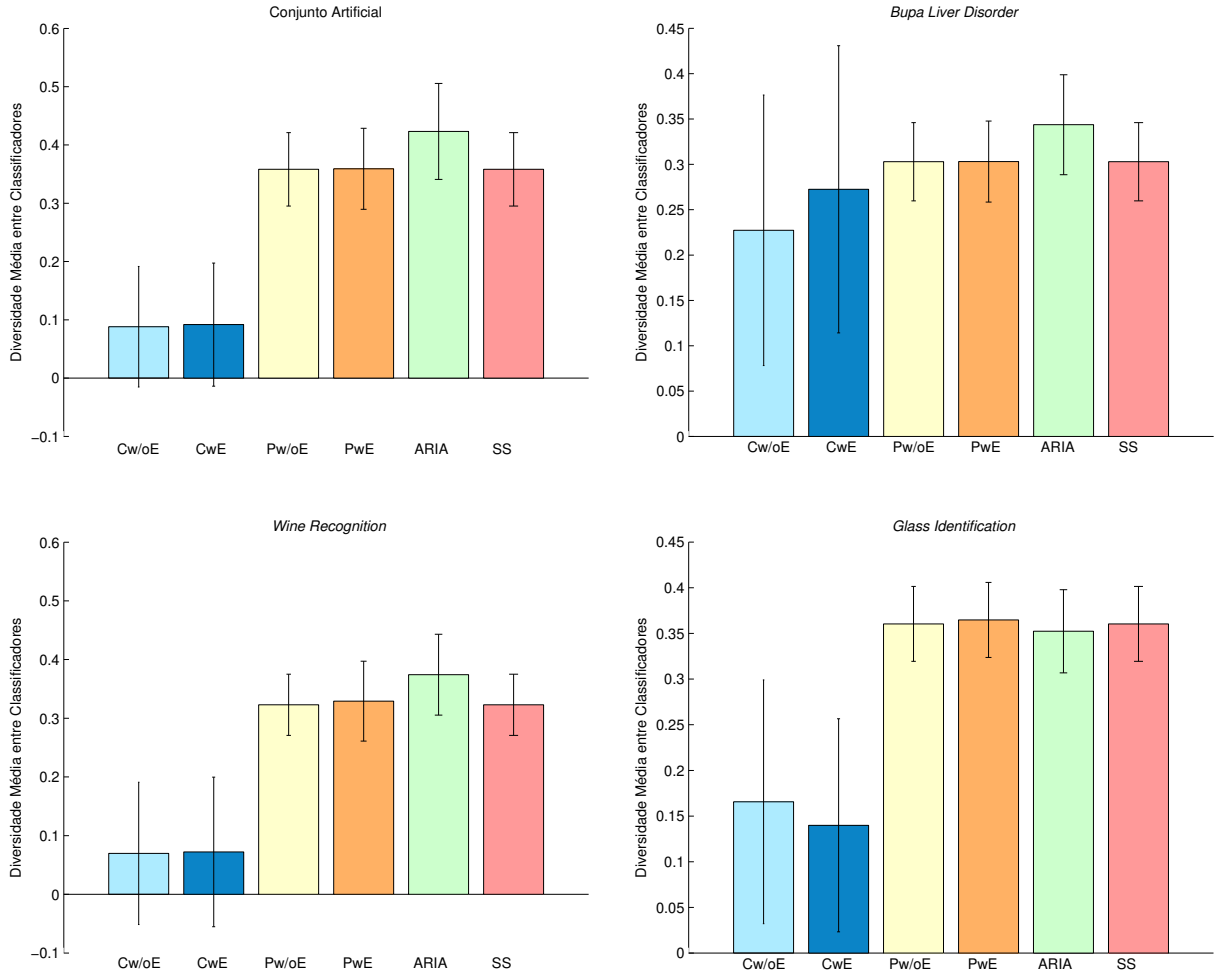


Fig. 5.17: Valor médio da métrica de diversidade (afinidade anticorpo/anticorpo) para os *ensembles*, de acordo com as técnicas de seleção. Cada coluna corresponde ao valor médio do número de componentes e as barras em seus topos correspondem aos respectivos desvios padrões.

poderia ser obtida através de várias execuções de algum algoritmo clássico de treinamento de redes neurais com condições iniciais diferentes, uma vez que a diversidade é baseada no comportamento dos classificadores e não na distância euclidiana dos vetores de pesos das redes neurais.

5.5.3 Confiança das Saídas dos *Ensembles*

Neste trabalho, também foi proposta uma métrica de *confiança* para as saídas dos *ensembles* e dos classificadores individuais, que se aplica às técnicas de combinação lineares, ou seja, à média simples, média ponderada e média ponderada com *bias*.

Esta métrica de *confiança* permite uma avaliação de "quão certo" um dado classificador está de sua saída, e é dada pela diferença média entre a saída de máximo valor (cujo índice dá o rótulo da classe) e a saída de segundo maior valor, tomados apenas os casos de amostras corretamente classificadas. Um exemplo do cálculo desta métrica de confiança para cada amostra de um problema de 3 classes pode ser visto na Figura 5.18.

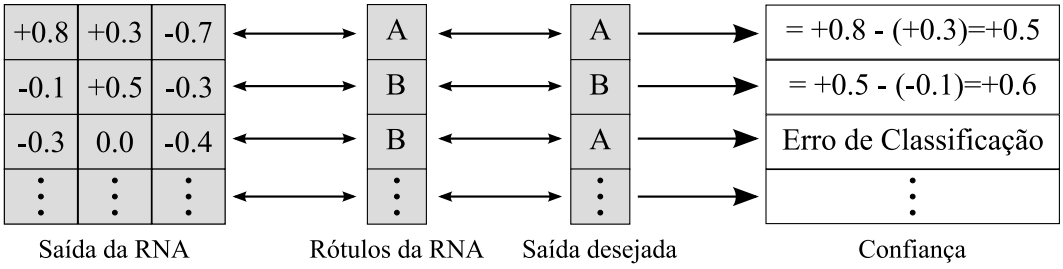


Fig. 5.18: Ilustração do cálculo da medida de *confiança*, para um problema de três classes. Neste exemplo, a rede neural classifica corretamente as duas primeiras amostras do problema, o que leva ao cálculo das respectivas confianças. Já para a terceira amostra, ocorre um erro de classificação o que exclui esta amostra do cálculo. A confiança total para a rede neural exemplificada aqui é dada pela média das confianças obtidas para todas as amostras corretamente classificadas.

As Tabelas 5.13, 5.14, 5.15 e 5.16 apresentam os resultados desta métrica de confiança obtida para os *ensembles* e para o melhor classificador individual (obtido pelo conjunto de validação) para o sub-conjunto de testes.

Analisando os dados sobre a confiança, podemos perceber que, para o problema *Artificial*, o uso de *ensembles* levou a uma redução da variância desta métrica e a um leve incremento de seu valor médio para alguns casos de uso de técnicas de combinação por média ponderada e média ponderada com *bias*. Mas considerando-se os altos desvios padrões envolvidos, este aumento não é significativo. Para o problema *Bupa*, houve uma redução considerável na variância da métrica de confiança para as técnicas de seleção por poda, ARIA e sem seleção, mas também houve uma redução do valor médio em si. Já para o problema *Wine*, as técnicas de combinação por média ponderada com e sem *bias*, principalmente quando associadas às técnicas de seleção por poda, ARIA e sem seleção, levaram a

Tab. 5.13: Valor da confiança obtido para cada *ensemble* com técnica de combinação linear e para o melhor classificador individual, para o sub-conjunto de teste do problema *Artificial*.

Método de Seleção	Método de Combinação	Confiança	Método de Seleção	Método de Combinação	Confiança
Cw/oE	MS	0.66 ± 0.16	PwE	MS	0.35 ± 0.09
	MP	0.82 ± 0.12		MP	0.86 ± 0.18
	MPB	0.85 ± 0.13		MPB	0.89 ± 0.15
CwE	MS	0.63 ± 0.15	ARIA	MS	0.32 ± 0.12
	MP	0.86 ± 0.15		MP	0.71 ± 0.14
	MPB	0.88 ± 0.15		MPB	0.77 ± 0.11
Pw/oE	MS	0.33 ± 0.06	SS	MS	0.33 ± 0.06
	MP	0.91 ± 0.18		MP	0.91 ± 0.18
	MPB	0.94 ± 0.15		MPB	0.94 ± 0.15
Melhor Classificador: 0.86 ± 0.24					

Tab. 5.14: Valor da confiança obtido para cada *ensemble* com técnica de combinação linear e para o melhor classificador individual, para o sub-conjunto de teste do problema *Bupa*.

Método de Seleção	Método de Combinação	Confiança	Método de Seleção	Método de Combinação	Confiança
Cw/oE	MS	0.63 ± 0.31	PwE	MS	0.54 ± 0.08
	MP	0.70 ± 0.24		MP	0.51 ± 0.05
	MPB	0.64 ± 0.21		MPB	0.52 ± 0.04
CwE	MS	0.57 ± 0.26	ARIA	MS	0.49 ± 0.07
	MP	0.66 ± 0.22		MP	0.51 ± 0.06
	MPB	0.60 ± 0.21		MPB	0.50 ± 0.07
Pw/oE	MS	0.55 ± 0.07	SS	MS	0.55 ± 0.07
	MP	0.51 ± 0.05		MP	0.51 ± 0.05
	MPB	0.51 ± 0.04		MPB	0.51 ± 0.04
Melhor Classificador: 0.76 ± 0.31					

Tab. 5.15: Valor da confiança obtido para cada *ensemble* com técnica de combinação linear e para o melhor classificador individual, para o sub-conjunto de teste do problema *Wine*.

Método de Seleção	Método de Combinação	Confiança	Método de Seleção	Método de Combinação	Confiança
Cw/oE	MS	0.59 ± 0.36	PwE	MS	0.60 ± 0.12
	MP	0.63 ± 0.39		MP	0.98 ± 0.21
	MPB	0.74 ± 0.27		MPB	0.97 ± 0.21
CwE	MS	0.57 ± 0.36	ARIA	MS	0.59 ± 0.14
	MP	0.60 ± 0.36		MP	0.90 ± 0.27
	MPB	0.71 ± 0.26		MPB	0.90 ± 0.27
Pw/oE	MS	0.59 ± 0.11	SS	MS	0.59 ± 0.11
	MP	1.01 ± 0.21		MP	1.01 ± 0.21
	MPB	0.99 ± 0.20		MPB	0.99 ± 0.20
Melhor Classificador: 0.59 ± 0.36					

Tab. 5.16: Valor da confiança obtido para cada *ensemble* com técnica de combinação linear e para o melhor classificador individual, para o sub-conjunto de teste do problema *Glass*.

Método de Seleção	Método de Combinação	Confiança	Método de Seleção	Método de Combinação	Confiança
Cw/oE	MS	0.22 ± 0.13	PwE	MS	0.30 ± 0.07
	MP	0.37 ± 0.28		MP	0.49 ± 0.09
	MPB	0.41 ± 0.13		MPB	0.42 ± 0.05
CwE	MS	0.23 ± 0.17	ARIA	MS	0.35 ± 0.11
	MP	0.36 ± 0.28		MP	0.49 ± 0.17
	MPB	0.41 ± 0.14		MPB	0.47 ± 0.09
Pw/oE	MS	0.32 ± 0.06	SS	MS	0.32 ± 0.06
	MP	0.49 ± 0.09		MP	0.49 ± 0.08
	MPB	0.48 ± 0.05		MPB	0.48 ± 0.05
Melhor Classificador: 0.24 ± 0.29					

um significativo aumento do valor médio da confiança e, na maioria dos casos, a uma redução da variância desta medida. Por fim, para o problema *Glass*, novamente houve um aumento da confiança para os *ensembles* com técnicas de combinação por média ponderada e média ponderada com *bias*, e uma significativa redução da variância para os casos de seleção por técnicas de poda e sem seleção. No entanto, como para este último caso a variância da medida para o melhor classificador é bem alta, o aumento do valor médio da confiança não é tão relevante.

Diante isso, podemos perceber uma forte dependência entre os resultados da aplicação da métrica de confiança e os problemas tratados neste trabalho, uma vez que, pelos resultados obtidos, não é possível determinar claramente os efeitos do uso da abordagem de *ensembles* sobre a métrica de confiança, quando comparados com os melhores classificadores individuais.

5.5.4 Análise Geral dos Resultados

De modo geral, os resultados obtidos neste trabalho indicaram uma forte dependência junto aos problemas tratados, sendo observadas apenas algumas tendências gerais, especialmente com relação ao número de componentes e à diversidade desses componentes nos *ensembles* formados. Tal dependência entre os resultados obtidos e os problemas tratados ficou mais evidente nos desempenhos médios e na aplicação da métrica de confiança junto aos *ensembles* e classificadores individuais, sendo que para esse último caso não foi possível perceber claramente qual a influência do uso da abordagem de *ensembles* sobre a métrica de confiança.

Para o caso do desempenho médio, quando analisamos qual a melhor técnica de seleção de componentes para cada um dos métodos de combinação, percebemos que: para o problema *Artificial*, a técnica construtiva sem exploração apresentou melhor desempenho junto ao maior número de métodos de combinação; para o problema *Bupa*, isso ocorreu para as técnicas construtiva sem exploração, poda (tanto com quanto sem exploração) e sem seleção; para o problema *Wine*, as melhores técnicas de seleção junto ao maior número de métodos de combinação foram as técnicas de poda sem exploração e sem seleção; e, para o problema *Glass*, isso ocorreu com a técnica de poda com exploração.

Já para o caso dos melhores métodos de combinação para cada técnica de seleção, obtivemos que: para o problema *Artificial*, o método de voto majoritário foi o vencedor junto ao maior número de técnicas de seleção; para o problema *Bupa*, o método vencedor foi a média ponderada sem *bias*; para o conjunto *Wine*, isso ocorreu para os métodos de média simples e voto majoritário; e para o problema *Glass*, para o método de média ponderada com *bias*.

No entanto, é importante ressaltar aqui que nem sempre o par de técnicas de seleção e combinação citadas acima foi o que levou à geração do *ensemble* de melhor desempenho médio para cada um dos problemas. Para ilustrar, tomemos o problema *Artificial*, em que a seleção por técnica construtiva sem exploração e combinação por voto majoritário (as técnicas vencedoras, de acordo com

os critérios descritos acima) levaram a um índice médio de acertos 0.89 ± 0.09, enquanto que outras técnicas levaram a índices de acertos de 0.92 ± 0.02 (por exemplo, seleção por poda sem exploração e combinação por média ponderada com *bias*).

Por fim, com relação ao número de componentes e diversidade desses componentes em cada *ensemble* gerado, foi possível observar uma tendência geral para todos os problemas tratados neste trabalho. Foi observado que as técnicas construtivas de seleção de componentes (tanto com quanto sem exploração) foram responsáveis pela geração de *ensembles* com o menor número médio de componentes e que, além disso, levaram a *ensembles* com menor diversidade entre seus componentes. No entanto, essas técnicas também levaram a *ensembles* com pior desempenho, quando comparados aos obtidos por outras estratégias, sendo que apenas para o problema *Artificial* um *ensemble* gerado por uma abordagem construtiva esteve entre os que apresentaram os melhores índices médios de acertos junto ao conjunto de testes.

5.6 Síntese do Capítulo

Neste capítulo, foram apresentados e analisados os resultados experimentais obtidos neste trabalho. Os experimentos foram feitos tomando-se por base quatro problemas de classificação, sendo um artificial e três problemas reais, obtidos no *UCI Repository of Machine Learning* (Newman et al., 1998): *Bupa Liver Disorder*, *Wine Recognition* e *Glass Identification*. Os conjuntos de dados desses problemas, bem como a metodologia adotada nos experimentos deste trabalho, foram descritos nas Seções 5.1 e 5.2, respectivamente.

Como mostrado na Seção 5.3, a aplicação do algoritmo *opt-aiNet* foi capaz de gerar uma população de classificadores (redes neurais) de bom desempenho e, ao mesmo tempo, diversas em relação aos seus padrões de classificação. Esta população diversa de componentes foi então submetida às demais etapas de construção do *ensemble* e, como mostrado na Seção 5.4, verificou-se que o par de técnicas de seleção e combinação responsável pelo *ensemble* de melhor desempenho variou a cada execução do algoritmo proposto, considerando-se um mesmo problema. Ou seja, este melhor par de técnicas de seleção e combinação apresentou uma dependência do conjunto de classificadores disponíveis para constituírem o *ensemble*. Nesta mesma seção, os experimentos foram repetidos adotando-se então a chamada estatística Q (Seção 2.1) como medida de diversidade entre classificadores, na tentativa de se verificar se a métrica de diversidade adotada influencia ou não esta dependência do melhor par de técnicas de seleção e combinação. Os resultados obtidos mostraram que existe uma influência da métrica de diversidade mas, para definir-se o porquê e a maneira como esta métrica influencia nesta dependência, são necessários mais estudos.

Para tentar extrair o máximo dessa dependência entre o melhor par de técnicas de seleção e combinação de componentes e o conjunto de classificadores disponíveis, foi proposto então o uso de

uma busca exaustiva, baseada no sub-conjunto de dados de validação, para selecionar o par dessas técnicas que leva ao melhor *ensemble* a cada execução do algoritmo. Na busca exaustiva, o uso do sub-conjunto de validação, disponível durante o treinamento de componentes e criação dos *ensembles*, se baseia na hipótese de que todos os sub-conjuntos em que o conjunto de dados original é particionado (treinamento, validação e teste) são capazes de representar as mesmas características do problema. No entanto, como mostrado na Seção 5.5.1, esta busca exaustiva baseada no sub-conjunto de validação não foi capaz de selecionar técnicas que apresentassem um valor médio de acertos superior ao dos demais *ensembles* quando o conjunto de teste é considerado. Levantou-se então a hipótese de que o particionamento por re-amostragem aleatória adotado neste trabalho não foi capaz de dividir os conjuntos de dados originais em sub-conjuntos capazes de representar igualmente as características de cada problema. Para que isso fosse verificado, a busca exaustiva foi novamente executada, mas levando-se em consideração agora o desempenho de cada par de técnicas de seleção e combinação de componentes para o conjunto de testes, e verificou-se que os novos *ensembles* obtidos superaram os demais, principalmente para os problemas *Bupa* e *Glass*, o que evidencia esta deficiência do particionamento.

Os índices de acertos obtidos para os *ensembles* neste trabalho também foram comparados com resultados previamente apresentados na literatura. Para isso, buscaram-se resultados em trabalhos que apresentassem propostas com alguma semelhança à feita aqui, como o uso de *ensembles* de redes neurais artificiais. Com isso, verificou-se que a metodologia proposta neste trabalho apresentou resultados compatíveis com os apresentados na literatura para os problemas aqui tratados.

Foram analisados também, na Seção 5.5.2, os números médios de componentes dos *ensembles* obtidos e a diversidade média desses componentes, e verificou-se que a metodologia proposta é capaz de gerar um bom grau de diversidade. Verificou-se também que as técnicas construtivas de seleção tendem a gerar *ensembles* com componentes mais semelhantes entre si (menor diversidade) e em menor número quando comparadas às técnicas de poda e baseada em algoritmo de agrupamento, o que é uma característica inerente a estas técnicas de seleção. No entanto, esse menor número de componentes e menor diversidade obtida pelas técnicas construtivas se traduziu em *ensembles* com piores índices de acertos quando comparados às demais estratégias empregadas neste trabalho.

Por fim, na Seção 5.5.3, foi proposta uma métrica (denominada *confiança*) capaz de avaliar o quão certo um dado classificador está de sua saída. Esta métrica só se aplica a técnicas de combinação linear (média simples e média ponderada com e sem *bias*), pois se baseia nas saídas dos classificadores antes da conversão para rótulos das classes. Foi então verificado que o uso de *ensembles* tende a gerar uma diminuição da variância desta métrica, quando comparado aos melhores classificadores individuais, mas de modo geral os resultados variaram muito entre cada problema aqui tratado e não se estabeleceu nenhuma tendência digna de nota.

Capítulo 6

Conclusões e Perspectivas Futuras

Neste trabalho, a construção de *ensembles* de redes neurais artificiais para problemas de classificação de padrões foi tratada como um problema de aprendizado de máquina composto por três fases: a geração de um conjunto de classificadores candidatos à composição do *ensemble*; a seleção dos componentes que realmente contribuem para o aumento da capacidade de generalização do sistema como um todo; e a combinação dos componentes selecionados para se obter a saída do *ensemble*.

Para que um *ensemble* tenha boa performance, seus componentes devem apresentar um bom desempenho individual e, ao mesmo tempo, devem possuir diversidade de comportamento, ou seja, devem atuar de forma diversa quanto aos erros cometidos em suas saídas. Visando atender a esses critérios de desempenho e diversidade, a etapa de geração de componentes foi tratada como um problema de otimização multimodal, resolvido através da aplicação do algoritmo populacional imuno-inspirado conhecido como *opt-aiNet*. Foram treinadas simultaneamente múltiplas redes neurais para classificação de padrões, buscando-se ao mesmo tempo minimizar o erro de classificação cometido por cada uma dessas redes e maximizar a diversidade de padrões de classificação apresentados por elas. A geração de classificadores empregou um conjunto de dados de treinamento.

Gerada essa população de classificadores, sub-conjuntos desses indivíduos foram então selecionados, empregando para tanto um conjunto de dados de validação, e, em seguida, combinados para produzirem os *ensembles*. Foram adotadas aqui seis técnicas de seleção distintas (construtiva sem e com exploração, poda sem e com exploração, algoritmo de agrupamento e sem seleção) e cinco estratégias de combinação de componentes (média simples, média ponderada com e sem *bias*, voto majoritário e *winner-takes-all*).

A metodologia proposta foi então aplicada a quatro problemas distintos de classificação de padrões, sendo um artificialmente gerado e três baseados em dados reais (*Bupa Liver Disorder*, *Wine Recognition* e *Glass Identification*), e verificou-se que os resultados obtidos foram compatíveis com os apresentados na literatura, sendo que o desempenho dos *ensembles* gerados foi medido junto a um

conjunto de dados de teste.

Ao longo dos experimentos, foi observado que, para um mesmo problema, o par de técnicas de seleção e combinação responsável pelo *ensemble* de melhor desempenho variou a cada execução do algoritmo proposto. Ou seja, verificou-se que o desempenho das técnicas de seleção e combinação adotadas aqui apresentou uma dependência em relação ao conjunto de classificadores disponíveis (candidatos a comporem o *ensemble*).

Buscando extrair o máximo dessa dependência entre o melhor par de técnicas de seleção e combinação de componentes e o conjunto de classificadores disponíveis, foi proposto então o uso de uma busca exaustiva, baseada no desempenho dos *ensembles* junto ao sub-conjunto de dados de validação, para selecionar o par dessas técnicas que leva ao melhor *ensemble* a cada execução do algoritmo. O uso do desempenho junto ao sub-conjunto de validação, como critério da busca exaustiva, se baseia na hipótese de que todos os sub-conjuntos em que o conjunto de dados original é particionado (treinamento, validação e teste) são capazes de representar as mesmas características do problema. No entanto, foi verificado que esta busca exaustiva baseada no sub-conjunto de validação não foi capaz de selecionar técnicas cujos *ensembles* apresentassem um valor médio de acertos para o conjunto de testes superior aos obtidos pelos demais *ensembles*. Levantou-se então a hipótese de que o particionamento por re-amostragem aleatória adotada neste trabalho não foi capaz de dividir os conjuntos de dados originais em sub-conjuntos capazes de representar igualmente as características de cada problema. Para que isso fosse verificado, foi feita uma nova execução da busca exaustiva, mas levando-se em consideração agora o desempenho de cada par de técnicas de seleção e combinação de componentes junto ao sub-conjunto de testes. Essa hipótese foi confirmada uma vez que o desempenho médio desse melhor *ensemble* resultante dessa nova busca exaustiva superou o desempenho dos demais, principalmente para os problemas *Bupa* e *Glass*.

Para verificar se a métrica de diversidade adotada neste trabalho teve alguma influência ou não na dependência do melhor par de técnicas de seleção e combinação em relação ao conjunto de classificadores candidatos, os experimentos foram repetidos adotando-se a chamada estatística Q como medida de diversidade entre os classificadores. Os resultados obtidos mostraram que existe essa influência da métrica de diversidade, mas, para definir-se como isso ocorre, são necessários mais estudos.

Por fim, foi feita também uma análise de confiabilidade dos resultados de classificação gerados pelos *ensembles*, e foi verificado que, de modo geral, o uso de *ensembles* tende a gerar uma diminuição da variância da métrica proposta neste trabalho, quando comparados aos melhores classificadores individuais. No entanto, os resultados para os experimentos com esta métrica não apresentaram nenhuma tendência relevante junto aos problemas tratados.

Como perspectivas futuras, uma extensão imediata deste trabalho seria o uso de diversos tipos de classificadores como candidatos, o que levaria aos chamados *ensembles heterogêneos*. Esse uso de

classificadores heterogêneos tende a aumentar a diversidade dos componentes, o que poderia tornar ainda mais relevante a busca pelo melhor par de técnicas de seleção e combinação de componentes.

Pretende-se, como uma outra extensão deste trabalho, tratar o problema de geração de candidatos a comporem o *ensemble* como um problema de otimização multi-objetivo, onde desempenho e grau de diversidade são dois objetivos a serem otimizados de forma explícita e simultaneamente.

Precisam ser feitos também mais estudos sobre essa questão da dependência do desempenho das técnicas de seleção e combinação de componentes em relação aos conjuntos de componentes disponíveis, bem como sobre a influência da métrica de diversidade utilizada sobre essa dependência. Caso seja confirmado que essa dependência é de grande relevância, o desenvolvimento de algumas heurísticas para determinação do melhor par de técnicas de seleção e combinação, dado um conjunto de componentes candidados à composição de um *ensemble*, certamente contribuiria para um ganho de desempenho ainda maior junto ao uso de *ensembles*.

Além disso, a proposta do uso de algoritmos imuno-inspirados para geração de um conjunto de componentes para *ensembles* que atendam aos requisitos de desempenho e diversidade exigidos poderia ser estendido para problemas de regressão, tais como aproximação de funções e predição de séries temporais.

Referências Bibliográficas

Abbattista, F., Di Gioia, G., Di Santo, G., & Farinelli, A. M. (1996). An associative memory based on the immune networks. In *Proceedings of the IEEE International Conference on Neural Networks*, (pp. 519–523).

Aisu, H., & Mitzutani, H. (1996). Immunity-based learning - Integration of distributed search and constraint relaxation. In *Proceedings of the International Conference on Multi-Agent Systems - Workshop on Immune-Based Systems*, (pp. 124–135). Kyoto, Japan.

Attux, R. R. F., Duarte, L. T., Ferrari, R., Panazio, C. M., de Castro, L. N., Von Zuben, F. J., & Romano, J. M. T. (2005). MLP-based equalization and pre-distortion using an artificial immune network. In *Proceedings of the 5th. IEEE Workshop on Machine Learning for Signal Processing*, (pp. 177–182).

Barajas-Montiel, S. E., & Reyes-García, C. A. (2005). Identifying pain and hunger in infant cry with classifiers ensembles. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, (pp. 770–775). Vienna, Austria.

Bersini, H. (1991). Immune network and adaptive control. In *Proceedings of the First European Conference on Artificial Life*, (pp. 217–226).

Bersini, H. (1999). The endogenous double plasticity of the immune network and the inspiration to the drawn for engineering aircrafts. In D. Dasgupta (Ed.) *Artificial Immune Systems and their Applications*, (pp. 22–44). Springer-Verlag.

Bersini, H., & Varela, F. J. (1990). Hints for adaptive problem solving gleaned from immune networks. *Parallel Problem Solving from Nature*, (pp. 343–354).

Bersini, H., & Varela, F. J. (1994). The immune learning mechanisms: Reinforcement, recruitment and their applications. In R. Paton (Ed.) *Computing with Biological Metaphors*. Chapman & Hall.

Bezerra, G. B., Barra, T. V., de Castro, L. N., & Von Zuben, F. J. (2005). Adaptive radius immune algorithm for data clustering. In C. Jacob, M. L. Pilat, P. J. Bentley, & J. Timmis (Eds.) *Proceedings of the 4th. International Conference on Artificial Immune Systems*, vol. 3627 of *Lecture Notes on Computer Science*, (pp. 290–303). Banff, Canada.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (1998). Randomizing outputs to increase prediction accuracy. Tech. rep., Statistics Department, University of California. Disponível em: http://www.boosting.org/papers/Bre98.pdf.

Brown, G. (2004). Diversity in neural network ensembles. Ph.D. thesis, School of Computer Science, University of Birmingham, UK.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorization. *Journal of Information Fusion*, *6*(1), 5–20.

Burnet, F. M. (1959). *The Clonal Selection Theory of Acquired Immunity*. Cambridge Press.

Caetano, M. F. (2006). Síntese sonora auto-organizável através da aplicação de algoritmos bio-inspirados. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e Computação (FEEC), Universidade Estadual de Campinas (Unicamp), Brasil.

Carter, J. H. (2000). The immune system as a model for pattern recognition and classification. *Journal of the American Medical Informatics Association*, *7*(1), 28–41.

Carvalho, D. R., & Freitas, A. A. (2001). An immunological algorithm for discovering small-disjunct rules in data mining. In *Proceedings of the Genetic and Evolutionary Computation Conference - Workshop on Artificial Immune Systems and Their Applications*, (pp. 401–404). San Francisco, USA.

Cazangi, R. . R., & Von Zuben, F. J. (2006). Immune learning classifier networks: Evolving nodes and connections. In *Proceedings of the IEEE Congress on Evolutionary Computation*, (pp. 7994–8001). Vancouver, Canada.

Cherkauer, K. J. (1996). Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks. In P. Chan, S. Stolfo, & D. Wolpert (Eds.) *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, (pp. 15–21). Portland, USA.

Coelho, G. P., de Castro, P. A. D., & Von Zuben, F. J. (2005). The effective use of diverse rule bases in fuzzy classification. In *Anais do VII Congresso Brasileiro de Redes Neurais*. Natal, Brasil.

Coelho, G. P., & Von Zuben, F. J. (2006). omni-aiNet: An immune-inspired approach for omni optimization. In H. Bersini, & J. Carneiro (Eds.) *Proceedings of the 5th. International Conference on Artificial Immune Systems*, vol. 4163 of *Lecture Notes in Computer Science*, (pp. 294–308). Oeiras, Portugal.

Cunningham, P., & Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection. In R. L. de Mántaras, & E. Plaza (Eds.) *Proceedings of the 11th European Conference on Machine Learning*, (pp. 109–116). Barcelona, Spain.

Dasgupta, D. (1996). A new algorithm for anomaly detection in time series data. In *Proceedings of the International Conference on Knowledge-Based Computer Systems*. Bombay, India.

Dasgupta, D. (Ed.) (1999a). *Artificial Immune Systems and their Applications*. Springer-Verlag.

Dasgupta, D. (1999b). Immunity-based intrusion detection system. In *Proceedings of the 22nd. National Information Systems Security Conference*, (pp. 147–160). Arlington, USA.

Dasgupta, D., Cao, Y., & Yang, C. (1999). An immunogenetic approach to spectra recognition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 149–155). Orlando, USA.

Dasgupta, D., & Forrest, S. (1999). Novelty detection in time series data using ideas from immunology. In *Proceedings of The International Conference on Intelligent Systems*.

de Castro, L. N., & Timmis, J. (2002a). An artificial immune network for multimodal function optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, (pp. 699–674). Hawaii, USA.

de Castro, L. N., & Timmis, J. (2002b). *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer.

de Castro, L. N., & Von Zuben, F. J. (2000). An evolutionary immune network for data clustering. In *Anais do Simpósio Brasileiro de Redes Neurais*, (pp. 84–89). Rio de Janeiro, Brasil.

de Castro, L. N., & Von Zuben, F. J. (2001). aiNet: An artificial immune network for data analysis. In H. A. Abbass, R. A. Sarker, & C. S. Newton (Eds.) *Data Mining: A Heuristic Approach*, (pp. 231–259). USA: Idea Group Publishing.

de Castro, P. A. D., Coelho, G. P., Caetano, M., & Von Zuben, F. J. (2005). Ensembles of fuzzy classification systems: An immune-inspired approach. In C. Jacob, M. L. Pilat, P. J. Bentley, &

J. Timmis (Eds.) *Proceedings of the 4th. International Conference on Artificial Immune Systems*, vol. 3627 of *Lecture Notes in Computer Science*, (pp. 469–482). Banff, Canada.

de França, F. O., Von Zuben, F. J., & de Castro, L. N. (2005). An artificial immune network for multimodal function optimization on dynamic environments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 289–296). Washington, DC, USA.

Dietterich, T. G., & Bakiri, G. (1991). Error-correcting output codes: a general method for improving multiclass inductive learning programs. In T. L. Dean, & K. McKeown (Eds.) *Proceedings of the 9th AAAI National Conference on Artificial Intelligence*, (pp. 572–577). Menlo Park, USA.

Efron, E., & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Freund, Y. (1995). Boosting a weak algorithm by majority. *Information and Computation*, *121*(2), 256–286.

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of the on-line learning and an application to boosting. In *Proceedings of the 2nd. European Conference on Computational Learning Theory*, (pp. 23–27). Barcelona, Spain.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In M. Kaufmann (Ed.) *Proceedings of the 13th International Conference on Machine Learning*, (pp. 148–156). Bari, Italy.

Fumera, G., & Roli, F. (2003). Linear combiners for classifier fusion: Some theoretical and experimental results. In T. Windeatt, & F. Roli (Eds.) *Proceedings of the 4th. International Workshop on Multiple Classifier Systems*, (pp. 74–83). Guilford, UK.

Gabow, H. N., Galil, Z., Spencer, T., & Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, *6*(2), 109–122.

Gaspar, A., & Collard, P. (2000). Two models of immunization for time dependent optimization. In *Proceedings of the IEEE System, Man, and Cybernetics Conference*, (pp. 113–118). Nashville, USA.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*(1), 1–58.

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification processes. *Image Vision and Computing Journal*, *19*(9-10), 699–707.

Gibert, C. J., & Routen, T. W. (1994). Associative memory in an immune-based system. In *Proceedings of the 12th. National Conference on Artificial Intelligence*, (pp. 852–857).

Gomes, L. C. T., de Sousa, J. S., Bezerra, G. B., de Castro, L. N., & Von Zuben, F. J. (2003). Copt-aiNet and the gene ordering problem. *Revista Tecnologia da Informação, Universidade Católica de Brasília*, *3*(2), 27–33.

Hajela, P., & Yoo, J. S. (1999). Immune network modelling in design optimization. In D. Corne, M. Dorigo, & F. Glover (Eds.) *New Ideas in Optimization*, (pp. 203–215). McGraw Hill.

Hansen, L. K., Liisberg, L., & Salamon, P. (1992). Ensemble methods for hadwritten digit recognition. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, (pp. 333–342). Helsingoer, France.

Hansen, L. K., & Salamon, P. (1990). Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001.

Hashem, S. (1993). Optimal linear combinations of neural networks. Ph.D. thesis, School of Industrial Engineering, University of Purdue, USA.

Hashem, S., & Schmeiser, B. (1995). Improving model accuracy using optimal linear combinations of trained neural networks. *IEEE Transactions on Neural Networks*, *6*(3), 792–794.

Hashem, S., Schmeiser, B., & Tih, Y. (1994). Optimal linear combinations of neural networks: An overview. In *Proceedings of the IEEE International Conference on Neural Networks*, (pp. 1507–1512). Orlando, USA.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Ho, T. (1998). The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation*, *7*(1), 45–68.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Huang, F. J., Zhou, Z.-H., Zhang, H.-J., & Chen, T. H. (2000). Pose invariant face recognition. In *Proceedings of the 4th. IEEE International Conference on Automatic Face and Gesture Recognition*, (pp. 245–250). Grenoble, France.

Hunt, J. E., & Cooke, D. E. (1996). Learning using an artificial immune system. *Journal of Network and Computer Applications*, *19*(2), 189–212.

Inoue, H., & Narihisa, H. (2000). Predicting chaotic time series by ensemble self-generating. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 231–236). Como, Italy.

Inoue, H., & Narihisa, H. (2004). Effective online pruning method for ensemble self-generating neural networks. In *Proceedings of the 47th Midwest Symposium on Circuits and Systems*, (pp. 85–88). Hiroshima, Japan.

Ishiguro, A., Ichikawa, S., Shibata, T., & Uchikawa, Y. (1998). Moderationism in the immune system: Gait acquisition of a legged robot using the metadynamics function. In *Proceedings of the IEEE System, Man, and Cybernetics Conference*, (pp. 3827–3832).

Islam, M. M., Yao, X., & Murase, K. (2003). A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on Neural Networks*, *14*(4), 820–834.

Jerne, N. K. (1974). Towards a network theory of the immune system. *Ann. Immunol. (Inst. Pasteur)*, *125C*, 373–389.

Kephart, J. O. (1994). A biologically inspired immune system for computers. In R. A. Brooks, & P. Maes (Eds.) *Artificial Live IV - Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, (pp. 130–139).

Khare, V., & Yao, X. (2002). Artificial speciation of neural network ensembles. In J. A. Bullinaria (Ed.) *Proceedings of the Workshop on Computational Intelligence*, (pp. 96–103). Birmingham, UK.

Kim, J., & Bentley, P. (1999). The human immune system and network intrusion detection. In *Proceedings of the European Congress on Intelligent Techniques and Soft Computing*, (pp. CD–ROM).

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In M. Kaufmann (Ed.) *Proceedings of the 12th International Conference on Machine Learning*, (pp. 313–321). Tahoe City, USA.

Koza, J. R. (1992). *Genetic Programming: On the Programming of computers by Means of Natural Selection*. MIT Press.

Krogh, A., & Vedelsby, J. (1995). Neural networks ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, *7*, 231–238.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Stat.*, *22*, 79–86.

Kuncheva, L. (2003). That elusive diversity in classifier ensembles. In J. J. Perales, A. J. C. Campilho, N. P. de la Blanca, & A. Sanfeliu (Eds.) *Proceedings of the First Iberian Conference on Pattern Recognition and Image Analysis*, (pp. 1126–1138). Mallorca, Spain.

Kuncheva, L., & Kountchev, R. (2002). Generating classifier outputs of fixed accuracy and diversity. *Pattern Recognition Letters*, *23*, 593–600.

Kuncheva, L., & Whitaker, C. (2001). Ten measures of diversity in classifiers ensembles: limits for two classifiers. In *Proceedings of the IEEE Workshop on Intelligent Sensor Processing*, (pp. 10/1 – 1010). Birmingham, UK.

Kuncheva, L., & Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*, 181–207.

Kuncheva, L., Whitaker, C., Shipp, C., & Duin, R. (2000). Is independence good for combining classifiers? In *Proceedings of the 15th International Conference on Pattern Recognition*, (pp. 2168–2171). Barcelona, Spain.

Kuncheva, L., Whitaker, C., Shipp, C., & Duin, R. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, *6*(1), 22–31.

Langdon, W. B., Barrett, S. J., & Buxton, B. F. (2002). Combining decision trees and neural networks for drug discovery. In *Proceedings of the 5th European Conference on Genetic Programming*, (pp. 60–70). Kinsale, Ireland.

Lazarevic, A., & Obradovic, Z. (2001). Effective pruning of neural network classifier ensembles. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 796–801). Washington, USA.

Lee, D.-W., & Sim, K.-B. (1997). Artificial immune network-based cooperative control in collective autonomous mobile robots. In *Proceedings of the IEEE International Workshop on Robotics and Human Communications*, (pp. 58–63).

Liao, Y., & Moody, J. (2000). Constructing heterogeneous committees using input feature grouping. *Advances in Neural Information Processing Systems*, *12*.

Lima, C. A. M. (2004). Comitê de máquinas: Uma abordagem unificada empregando máquinas de vetores suporte. Tese de Doutorado, Faculdade de Engenharia Elétrica e Computação (FEEC), Universidade Estadual de Campinas (Unicamp), Brasil.

Lima, C. A. M., Coelho, A. L. V., & Von Zuben, F. J. (2002). Ensembles of support vector machines for regression problems. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 2381–2386). Hawaii, USA.

Liu, Y. (1998). Negative correlation learning and evolutionary neural network ensembles. Ph.D. thesis, University College, The University of New South Wales, Australian Defence Force Academy, Australia.

Liu, Y. (2005). Ensemble methods for handwritten text line recognition systems. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, (pp. 2334–2339). Hawaii, USA.

Liu, Y., & Yao, X. (1997). Negatively correlated neural networks can produce best ensembles. *Australian Journal of Intelligent Information Processing Systems*, *4*(3-4), 176–185.

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, *12*(10), 1399–1404.

Liu, Y., Yao, X., & Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, *4*(4), 380–387.

Lu, J., Plataniotis, K. N., Venetsanopoulos, A. N., & Li, S. Z. (2006). Ensemble-based discriminant learning with boosting for face recognition. *IEEE Transactions on Neural Networks*, *17*(1), 166–178.

Maclin, R., & Shavlik, J. W. (1995). Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, (pp. 524–530). Montreal, Canada.

Mao, J. (1998). A case study on bagging, boosting and basic ensembles of neural networks for OCR. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 1828–1833). Anchorage, Alaska, USA.

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press.

McCoy, D. F., & Devarajan, V. (1997). Artificial immune systems and aerial image segmentation. In *Proceedings of the IEEE System, Man, and Cybernetics Conference*, (pp. 867–872). Orlando, USA.

McKay, R., & Abbass, H. (2001a). Analyzing anticorrelation in ensemble learning. In *Proceedings of the Conference on Artificial Neural Networks and Expert Systems*, (pp. 22–27). Dunedin, New Zealand.

McKay, R., & Abbass, H. (2001b). Anticorrelation measures in genetic programming. In *Proceedings of the Australian-Japan Workshop on Intelligent and Evolutionary Systems*, (pp. 45–51).

Melville, P., & Mooney, R. (2003). Constructing diverse classifier ensembles using artificial training examples. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, (pp. 505–510). Acapulco, Mexico.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.

Michelan, R., & Von Zuben, F. J. (2002). Decentralized control system for autonomous navigation based on an evolved artificial immune network. In *Proceedings of the IEEE Congress on Evolutionary Computation*, (pp. 1021–1026). Honolulu, USA.

Mitsumoto, N., Fukuda, T., Shimojima, K., & Ogawa, A. (1996). Self-organizing multiple robotic system (a population control through biologically inspired immune network architecture). In *Proceedings of the IEEE International Conference on Robotics and Automation*, (pp. 1614–1619). Minneapolis, USA.

Mori, M., Tsukiyama, M., & Fukuda, T. (1993). Immune algorithm with searching diversity and its application to resource allocation problem. *Transactions of the Institute of Electrical Engineers of Japan*, *113-C*(10), 872–878.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. Disponível on-line em: http://www.ics.uci.edu/~mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences.

Ootsuki, J. T., & Sekiguchi, T. (1999). Application of the immune system network concept to sequential control. In *Proceedings of the IEEE System, Man, and Cybernetics Conference*, vol. 3, (pp. 869–874). Tokyo, Japan.

Opitz, D. W., & Shavlik, J. W. (1996). Generating accurate and diverse members of a neural-network ensemble. In *Proceedings of the Neural Information Processing Systems Conference*, (pp. 535–541). Denver, USA.

Oza, N. C. (2003). Boosting with averaged weight vectors. In *Proceedings of the 4th. International Workshop on Multiple Classifier Systems*, (pp. 15–24). Guilford, UK.

Oza, N. C., & Tumer, K. (2001). Input decimation ensembles: Decorrelation through dimensionality reduction. In *Proceedings of the International Workshop on Multiple Classifier Systems*, (pp. 238–247). Cambridge, UK.

Parmanto, B., Munro, P. W., & Doyle, H. R. (1996). Improving committee diagnosis with resampling techniques. *Advances in Neural Information Processing Systems*, *8*, 882–888.

Partridge, D. (1996). Network generalization differences quantified. *Neural Networks*, *9*(2), 263–271.

Partridge, D., & Yates, W. B. (1996). Engineering multiversion neural-net systems. *Neural Computation*, *8*(4), 869–893.

Pasti, R., & de Castro, L. N. (2006). An immune and a gradient-based method to train multi-layer perceptron neural networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 4077–4084). Vancouver, Canada.

Perelson, A. S., & Oster, G. F. (1979). Theorectical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonself discrimination. *J. Theor. Biol.*, *81*, 645–670.

Perrone, M. (1993). Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization. Ph.D. thesis, Institute for Brain and Neural Systems, Brown University, USA.

Perrone, M. P., & Cooper, L. N. (1993). When networks disagree: Ensemble method for neural networks. In R. J. Mammone (Ed.) *Neural Networks for Speech and Image Processing*, (pp. 126–142). Chapman-Hall.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45.

Raviv, Y., & Intrator, N. (1996). Bootstrapping with noise: An effective regularisation technique. *Connection Science*, *8*, 355–372.

Rocha, M., Cortez, P., & Neves, J. (2005). Evolutionary design of neural networks for classification and regression. In *Proceedings of the 7th International Conference on Adaptive and Natural Computing Algorithms*, (pp. 304–307). Coimbra, Portugal.

Roli, F., & Fumera, G. (2002). Analysis of linear and order statistics combiners for fusion of imbalanced classifiers. In F. Roli, & J. Kittler (Eds.) *Proceedings of the 3rd. International Workshop on Multiple Classifier Systems*, (pp. 252–261). Cagliari, Italy.

Rosen, B. E. (1996). Ensemble learning using decorrelated neural networks. *Connection Science - Special Issue on combining Artificial Neural Networks: Ensemble Approaches*, *8*(3-4), 373–384.

Schapire, R. E. (1990). The strengh of weak learnability. *Machine Learning*, *5*, 197–227.

Seber, G. A. F. (1984). *Multivariate Observations*. Wiley.

Sharkey, A., & Sharkey, N. (1997a). Combining diverse neural networks. *The Knowledge Engineering Review*, *12*(3), 231–247.

Sharkey, A., & Sharkey, N. (1997b). Diversity, selection, and ensembles of artificial neural nets. In *Proceedings of the 3rd. International Conference on Neural Networks and their Applications*, (pp. 205–212). Marseille, France.

Sharkey, A., Sharkey, N., & Chandroth, G. (1996). Diverse neural net solutions to fault diagnosis problem. *Neural Computing and Applications*, *4*, 218–227.

Sharkey, A. J. C., Sharkey, N. E., Gerecke, U., & Chandroth, G. O. (2000). The test and select approach to ensemble combination. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, (pp. 30–44). Cagliari, Italy.

Sharkey, N., Neary, J., & Sharkey, A. (1995). Searching weight space for backpropagation solution types. In *Current Trends in Connectionism: Swedish Conference on Connectionism*, (pp. 103–120).

Shipp, C., & Kuncheva, L. (2002). Relationships between combination methods and measures of diversity. *Information Fusion*, *3*, 135–148.

Skalak, D. (1996). The sources of increased accuracy for two proposed boosting algorithms. In *Proceedings of the American Association for Artificial Intelligence (AAAI-96) Integrating Multiple Learned Models Workshop*. Portland, USA.

Skurichina, M., Kuncheva, L., & Duin, R. P. W. (2002). Bagging and boosting for nearest mean classifier: Effects of sample size on diversity and accuracy. In F. Roli, & J. Kittler (Eds.) *Proceedings of the 3rd. International Workshop on Multiple Classifier Systems*, (pp. 62–71). Cagliari, Italy.

Soares, R. G. F., Santana, A., Canuto, A. M. P., & Souto, M. C. P. (2006). Using accuracy and diversity to select classifiers to build ensembles. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 2289–2295). Vancouver, Canada.

Somayagi, A., Hofmeyr, S. A., & Forrest, S. (1998). Principles of a computer immune system. In *Proceedings of the New Security Paradigm Workshop*, (pp. 75–82). Langdale, UK.

Takahashi, K., & Yamada, T. (1997). A self-tuning immune feedback controller for controlling mechanical systems. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Tokyo, Japan.

Tarakanov, A., Sokolova, S., Abramov, B., & Aikimbayev, A. (2000). Immunocomputing of the natural plague foci. In *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 38–39). Las Vegas, USA.

Timmis, J., Neal, M., & Hunt, J. (2000). An artificial immune system for data analysis. *BioSystems*, *55*(1-3), 143–150.

Toma, N., Endo, S., & Yamada, K. (1999). Immune algorithm with immune network and MHC for adaptive problem solving. In *Proceedings of the IEEE System, Man, and Cybernetics Conference*, vol. 4, (pp. 271–276). Tokyo, Japan.

Tumer, K., & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection Science*, *8*(3-4), 385–403.

Twala, B., & Cartwright, M. (2005). Ensemble imputation methods for missing software engineering data. In *Proceedings of the 11th IEEE International Software Metrics Symposium*. Como, Italy.

Ueda, N., & Nakano, R. (1996). Generalization error of ensemble estimators. In *Proceedings of IEEE International Conference on Neural Networks*, (pp. 90–95). Washington, USA.

Walker, B. (2001). Ecosystems and immune systems: Useful analogy or stretching a metaphor. *Conservation Ecology*, *5*(1), 16. Disponível on-line em: http://www.consecol.org/vol5/iss1/art16.

Wang, W., Jones, P., & Partridge, D. (2000). Diversity between neural networks and decision trees for building multiple classifier systems. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, (pp. 240–249). Calgiari, Italy.

Wang, W., Richards, G., & Rea, S. (2005). Hybrid data mining ensemble for predicting osteoporosis risk. In *Proceedings of the IEEE 27th Annual Conference on Engineering in Medicine and Biology*, (pp. 886–889). Shanghai, China.

Watkins, A. (2001). AIRS: A resource limited artificial immune classifier. M. S. Dissertation, Mississippi State University, Mississippi, EUA.

Wesolkowski, S., & Hassanein, K. (1997). A comparative study of combination schemes for an ensemble of digit recognition neural networks. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, (pp. 3534–3539). Orlando, USA.

Wichard, J. D., & Ogorzalek, M. (2004). Time series prediction with ensemble models. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, (pp. 1625–1630). Budapest, Hungary.

Wolpert, D. H., & Mcready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Woods, K., Kegelmeyer, W., & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 405–410.

Yao, X., & Liu, Y. (1998). Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, *28*, 417–425.

Yates, W., & Partridge, D. (1996). Use of methodological diversity to improve neural network generalization. *Neural Computing and Applications*, *4*(2), 114–128.

Yule, G. U. (1900). On the association of attributes in statistics. *Philosophical Transactions*, *A*(194), 257–319.

Zenobi, G., & Cunningham, P. (2001). Using diversity in preparing ensembles of classifiers based on different subsets to minimize generalization error. In L. de Raedt, & P. Flach (Eds.) *Proceedings of the 12th European Conference on Machine Learning*, (pp. 576–587). Freiburg, Germany.

Zhan, T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, *C*(20), 68–86.

Zhou, Z. H., & Jiang, Y. (2003). Medical diagnosis with c4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, *7*(1), 37–42.

Zhou, Z. H., Jiang, Y., Yang, Y. B., & Chen, S. F. (2000). Lung cancer cell identification based on neural network ensembles. *Artificial Intelligence in Medicine*, *24*(1), 25–36.

Zhou, Z. H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, *137*(1-2), 239–263.