Universidade Estadual de Campinas - Unicamp
Faculdade de Engenharia Elétrica e de Computação - FEEC
Departamento de Engenharia de Computação
e Automação Industrial - DCA

Sinergia entre Sistemas Imunológicos Artificiais e Modelos Gráficos Probabilísticos

Pablo Alberto Dalbem de Castro

Orientador: Prof. Dr. Fernando José Von Zuben

Tese de Doutorado apresentada ao Programa de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para obtenção do título de Doutor em Engenharia Elétrica

Área de Concentração: **Engenharia de Computação**

Campinas - São Paulo - Brasil
Julho de 2009

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

C729s

Castro, Pablo Alberto Dalbem
Sinergia entre sistemas imunológicos artificiais e modelos gráficos probabilísticos / Pablo Alberto Dalbem de Castro. --Campinas, SP: [s.n.], 2009.

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

1. Probabilistica. 2. Aprendizado do computador. 3. Metaheuristica. 4. Otimização . 5. Sistemas inteligentes. I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Synergy between artificial immune systems and probabilistic graphical models
Palavras-chave em Inglês: Probability, Machine learning, Metaheuristic, Optimization, Intelligent buildings
Área de concentração: Engenharia da Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora: André Luís Vasconcelos Coelho, Ricardo Hiroshi Caldeira Takahashi, Márcio Luiz de Andrade Netto, Romis Ribeiro de Faissol Attux
Data da defesa: 07/07/2009
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Pablo Alberto Dalbem de Castro

Data da Defesa: 7 de julho de 2009

Título da Tese: "Sinergia entre Sistemas Imunológicos Artificiais e Modelos Gráficos Probabilísticos"

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

Prof. Dr. André Luís Vasconcelos Coelho:

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi:

Prof. Dr. Márcio Luiz de Andrade Netto:

Prof. Dr. Romis Ribeiro de Faissol Attux:

Agradecimentos

Agradeço, primeiramente, ao meu orientador Prof. Dr. Fernando José Von Zuben, pelos ensinamentos, dedicação, competência e pela orientação segura.

Agradeço aos membros da banca examinadora, pelas valiosas contribuições.

Agradeço também aos amigos do Laboratório de Bioinformática e Computação Bio-inspirada (LBiC), pelo companheirismo e por proporcionarem um ambiente agradável para se trabalhar.

Agradeço à Faculdade de Engenharia Elétrica e de Computação (FEEC) da Unicamp, e em particular ao Departamento de Computação e Automação Industrial (DCA), pela oportunidade de realizar o curso de doutorado.

Agradeço aos meus familiares e amigos, pela motivação e incentivo.

Agradeço a todos que contribuíram de forma direta ou indireta para a realização deste trabalho.

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

Resumo

Sistemas imunológicos artificiais (SIAs) e modelos gráficos probabilísticos são duas importantes técnicas para a construção de sistemas inteligentes e têm sido amplamente exploradas por pesquisadores das mais diversas áreas, tanto no aspecto teórico quanto prático. Entretanto, geralmente o potencial de cada técnica é explorado isoladamente, sem levar em consideração a possível cooperação entre elas. Como uma primeira contribuição deste trabalho, é proposta uma metodologia que explora as principais vantagens dos SIAs como ferramentas de otimização voltadas para aprendizado de redes bayesianas a partir de conjuntos de dados. Por outro lado, os SIAs já propostos para otimização em espaços discretos e contínuos correspondem a meta-heurísticas populacionais sem mecanismos para lidarem eficientemente com blocos construtivos, e também com poucos recursos para se beneficiarem do conhecimento já adquirido acerca do espaço de busca. A segunda contribuição desta tese é a proposição de quatro algoritmos que procuram superar estas limitações, em contextos mono-objetivo e multiobjetivo. São substituídos os operadores de clonagem e mutação por um modelo probabilístico representando a distribuição de probabilidades das melhores soluções. Em seguida, este modelo é empregado para gerar novas soluções. Os modelos probabilísticos utilizados são a rede bayesiana, para espaços discretos, e a rede gaussiana, para espaços contínuos. A escolha de ambas se deve às suas capacidades de capturar adequadamente as interações mais relevantes das variáveis do problema. Resultados promissores foram obtidos nos experimentos de otimização realizados, os quais trataram, em espaços discretos, de seleção de atributos e de *ensembles* para classificação de padrões, e em espaços contínuos, de funções multimodais de elevada dimensão.

Palavras-chave: sistemas imunológicos artificiais, redes bayesianas, redes gaussianas, otimização em espaços discretos e contínuos, otimização mono-objetivo e multiobjetivo.

Abstract

Artificial immune systems (AISs) and probabilistic graphical models are two important techniques for the design of intelligent systems, and they have been widely explored by researchers from diverse areas, in both theoretical and practical aspects. However, the potential of each technique is usually explored in isolation, without considering the possible cooperation between them. As a first contribution of this work, it is proposed an approach that explores the main advantages of AISs as optimization tools applied to the learning of Bayesian networks from data sets. On the other hand, the AISs already proposed to perform optimization in discrete and continuous spaces correspond to population-based meta-heuristics without mechanisms to deal effectively with building blocks, and also having few resources to benefit from the knowledge already acquired from the search space. The second contribution of this thesis is the proposition of four algorithms devoted to overcoming these limitations, both in single-objective and multi-objective contexts. The cloning and mutation operators are replaced by a probabilistic model representing the probability distribution of the best solutions. After that, this model is employed to generate new solutions. The probabilistic models adopted are the Bayesian network, for discrete spaces, and the Gaussian network, for continuous spaces. These choices are supported by their ability to properly capture the most relevant interactions among the variables of the problem. Promising results were obtained in the optimization experiments carried out, which have treated, in discrete spaces, feature selection and ensembles for pattern classification, and, in continuous spaces, multimodal functions of high dimension.

Keywords: artificial immune systems, Bayesian networks, Gaussian networks, optimization in discrete and continuous domains, single-objective and multi-objective optimization.

Trabalhos publicados pelo autor

1. CASTRO, P. A. D.; VON ZUBEN, F. J. Multi-objective Bayesian Artificial Immune System: Empirical Evaluation and Comparative Analyses. Journal of Mathematical Modelling and Algorithms, vol. 8, pp. 151-173, 2009.

2. CASTRO, P. A. D.; VON ZUBEN, F. J. BAIS: A Bayesian Artificial Immune System for the effective handling of building blocks. Information Sciences, vol. 179, pp. 1426-1440, 2009.

3. CASTRO, P. A. D.; VON ZUBEN, F. J. Multi-objective Feature Selection Using a Bayesian Artificial Immune System. International Journal of Intelligent Computing and Cybernetics (no prelo).

4. CASTRO, P. A. D.; DE FRANÇA, F. O.; FERREIRA, H. M.; COELHO, G. P.; VON ZUBEN, F. J. Query Expansion Using an Immune-Inspired Biclustering Algorithm. Natural Computing (no prelo).

5. CASTRO, P. A. D.; VON ZUBEN, F. J. Learning Bayesian Networks to Perform Feature Selection. Proceedings of the 2009 International Joint Conference on Neural Networks, June 14-19, pp. 467-473, 2009.

6. CASTRO, P. A. D.; VON ZUBEN, F. J. MOBAIS: A Bayesian Artificial Immune System for Multi-objective Optimization. Proc. of the 7th International Conference on Artificial Immune System. Lecture Notes in Computer Science, Springer, vol. 5132, pp. 48-59, 2008.

7. CASTRO, P. A. D.; VON ZUBEN, F. J. Feature Subset Selection by means of a Bayesian Artificial Immune System. Proceedings of the 8th Conference on Hybrid Intelligent Systems, pp. 561-566, 2008.

8. CASTRO, P. A. D.; DE FRANÇA, F. O.; FERREIRA, H. M.; VON ZUBEN, F. J. Applying Biclustering to Perform Collaborative Filtering. Proc. of the 7th International Conference on Intelligent Systems Design and Applications, pp. 421-426, 2007.

9. CASTRO, P. A. D.; DE FRANÇA, F. O.; FERREIRA, H. M.; VON ZUBEN, F. J. Applying Biclustering to Text Mining: An Immuno-Inspired Approach. Proc. of the 6th International Conference on Artificial Immune System, Lecture Notes in Computer Science, Springer, vol. 4628, pp. 83-94, 2007.

10. CASTRO, P. A. D.; DE FRANÇA, F. O.; FERREIRA, H. M.; VON ZUBEN, F. J. Evaluating the Performance of a Biclustering Algorithm Applied to Collaborative Filtering: A

Comparative Analysis. Proceedings of the 7th International Conference on Hybrid Intelligent Systems, pp. 65-70, 2007.

11. CASTRO, P. A. D.; VON ZUBEN, F. J. Bayesian Learning of Neural Networks by Means of Artificial Immune Systems. Proceedings of the IEEE World Congress on Computational Intelligence, pp. 9885-9892, 2006.

12. LIMA, C. A. M.; CASTRO, P. A. D.; COELHO, A. L. V.; JUNQUEIRA, C.; VON ZUBEN, F. J. Controlling Nonlinear Dynamic Systems with Projection Pursuit Learning. Proceedings of the 3rd IEEE International Conference on Intelligent Systems, pp. 332-337, 2006.

13. CASTRO, P. A. D.; COELHO, G. P.; CAETANO, M. F.; VON ZUBEN, F. J. Designing Ensembles of Fuzzy Classification Systems: An Immune-Inspired Approach. Proceedings of the 4th International Conference on Artificial Immune Systems, Lecture Notes in Computer Science, Springer, vol. 3627, pp. 469-482, 2005.

14. COELHO, G. P.; CASTRO, P. A. D.; VON ZUBEN, F. J. The Effective Use of Diverse Rule Bases in Fuzzy Classification. Anais do VII Congresso Brasileiro de Redes Neurais, pp. 1-6, 2005.

15. CASTRO, P. A. D.; VON ZUBEN, F. J. An Immune-Inspired Approach to Bayesian Networks. Proceedings of the 5th International Conference on Hybrid Intelligent Systems, pp. 23-28, 2005.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Acrônimos

AG Algoritmo genético
AIC Critério de Informação do Akaike, do inglês *Akaike's Information Criterion*

BIC Critério de Informação Bayesiano, do inglês *Bayesian Information Criterion*

MDL Comprimento Mínimo de Descrição, do inglês *Minimum Description Length*
MLP Perceptron multicamadas, do inglês *Multi-layer Perceptron*

PLS Amostragem Probabilística Lógica, do inglês *Probabilistic Logic Sampling*

SIA Sistema Imunológico Artificial

Capítulo 1

Introdução

Aprendizado de máquina é uma das principais frentes numa ampla área de pesquisa denominada inteligência computacional, sendo caracterizada pelo desenvolvimento de algoritmos heurísticos que combinam aspectos de aprendizado, representação do conhecimento e evolução para criar, em computador, mecanismos de processamento de informação mais avançados.

Como já é comum em muitas outras áreas de Engenharia de Computação, também em aprendizado de máquina deve-se lidar com uma questão prática desafiadora: na busca de formas mais eficazes para tratamento de dados e atendimento de critérios de desempenho, as próprias técnicas de aprendizado de máquina, originalmente voltadas para a solução de certos tipos de problemas computacionais, criam problemas computacionais de elevada complexidade. Esta tese está voltada para o tratamento de dois problemas computacionais criados por técnicas de aprendizado de máquina:

Problema 1: No contexto de inferência probabilística, como obter modelos gráficos probabilísticos que melhor explicam as interdependências de variáveis de algum processo em estudo, a partir de dados amostrados?

Problema 2: No contexto de sistemas imunológicos artificiais voltados para a solução de problemas de otimização, como tornar o processo de busca mais eficiente, preservando blocos construtivos?

Duas particularidades estão envolvidas no tratamento que a tese dá a esses dois problemas:

- Serão empregadas técnicas de aprendizado de máquina para lidar com esses problemas derivados da implementação de técnicas de aprendizado de máquina;
- A técnica que trata o Problema 1 é a responsável pela existência do Problema 2, enquanto que a técnica que trata o Problema 2 é a responsável pela existência do Problema 1.

Devido à segunda particularidade acima, poder-se-ia incorrer em uma metodologia cíclica de tratamento: Ao se tentar resolver o Problema 1, cria-se o Problema 2; e ao se tentar resolver o

Problema 2, cria-se o Problema 1. Para evitar este caráter cíclico da metodologia, resolve-se o Problema 1 com sistemas imunológicos artificiais, em uma configuração que tem o Problema 2 como uma questão em aberto; e resolve-se o Problema 2 com modelos gráficos probabilísticos que têm o Problema 1 como uma questão em aberto.

O fundamental é que, mesmo mantendo questões em aberto no tratamento dos dois problemas acima, as soluções obtidas representam claramente um passo adiante em seus processos de tratamento, com ganhos de desempenho expressivos frente a abordagens alternativas existentes na literatura. Cria-se, assim, uma sinergia entre sistemas imunológicos artificiais e modelos gráficos probabilísticos.

A seguir, na Seção 1.1, são apresentados aspectos conceituais dos dois problemas acima e argumentos capazes de evidenciar o mérito individual de cada problema, justificando assim o investimento de esforços no seu tratamento. A Seção 1.2, por sua vez, descreve os objetivos da tese em termos mais específicos e contém motivações para sustentar esta sinergia entre sistemas imunológicos artificiais e modelos gráficos probabilísticos. A Seção 1.3 lista as contribuições alcançadas, e a Seção 1.4 descreve a organização do texto da tese.

1.1 Fundamentação e Motivação

Modelos gráficos probabilísticos compreendem qualquer modelo que utiliza a teoria dos grafos em conjunto com a teoria de probabilidade para representar e manipular conhecimento em domínios em que a complexidade e a incerteza estão presentes.

Um modelo gráfico probabilístico é composto de duas partes, uma qualitativa e outra quantitativa. A parte qualitativa é expressa por um grafo em que os nós representam variáveis aleatórias e os arcos que as relacionam denotam a dependência estatística entre elas. O segundo componente, quantitativo, é a distribuição de probabilidades para essas variáveis.

Dentre os modelos gráficos probabilísticos, o mais conhecido e utilizado é a rede bayesiana, em que a estrutura é um grafo acíclico direcionado e as variáveis são discretas. Elas são aplicadas com sucesso em problemas de bioinformática (Friedman et al., 2000; Imoto et al., 2003; Nariai et al., 2004), classificação de padrões (Androutsopoulos et al., 2000; Raval et al., 2002; Stephenson et al., 2000; Vehtari & Lampinen, 2000), diagnóstico médico (Antal et al., 2003), diagnóstico de falhas em computadores (Horvitz et al., 2001) e aprendizado de máquina (Hruschka Jr et al., 2004; Inza et al., 2000).

Nos últimos anos, além do aumento no interesse por aplicações de redes bayesianas, houve também um crescimento no desenvolvimento de ferramentas capazes de construir automaticamente redes bayesianas a partir de conjunto de dados que representam amostras ou exemplos do problema.

O aprendizado da rede bayesiana pode ser dividido em duas etapas. A primeira é a definição das relações de interdependência entre as variáveis, ou seja, a estrutura da rede. Uma vez que a estrutura da rede tenha sido definida, entra em cena a segunda etapa, que é o cálculo das probabilidades para as variáveis.

As distribuições de probabilidades podem ser estimadas usando o método da máxima probabilidade a *posteriori* ou da máxima verossimilhança. Ambos os métodos são computados facilmente a partir dos dados e os algoritmos existentes na literatura são eficientes (Buntine, 1996). O aprendizado da estrutura da rede consiste em determinar relações entre as variáveis, adicionar ou excluir arcos, estabelecer direções, enfim definir um grafo direcionado acíclico para o qual serão calculadas distribuições de probabilidade. De acordo com Buntine (1996), este é um problema NP-completo.

Um paradigma de aprendizado, conhecido como busca e pontuação, considera a tarefa de construção da rede bayesiana como um problema de busca e otimização, em que um mecanismo de busca é responsável por explorar o espaço de todas as possíveis estruturas de rede, enquanto uma função é responsável pela avaliação de cada solução candidata encontrada.

Embora estratégias de busca não-populacionais, como o subida da encosta (*hill climbing*), tenham sido amplamente empregadas, elas são fortemente sujeitas a mínimos locais, uma vez que elas refinam apenas uma única solução por vez (Heckerman, 1995). Para eliminar este inconveniente, algumas metodologias propõem o uso de algoritmos baseados em população de soluções, como algoritmos genéticos. Entretanto, conforme apontado por Buntine (1996), o espaço de busca no aprendizado de redes bayesianas é amplo e multimodal, e um algoritmo genético básico não é capaz de lidar com otimização multimodal de forma efetiva (Goldberg & Richardson, 1987).

Um novo paradigma de computação bio-inspirada, chamado de sistema imunológico artificial (SIA), foi concebido a partir dos princípios de funcionamento do sistema imunológico dos vertebrados (Dasgupta, 1999; de Castro & Timmis, 2002b). Em um sistema imunológico artificial, as soluções candidatas, chamadas de anticorpos, são codificadas num vetor de caracteres. Aumentando a concentração das melhores soluções e as submetendo a um processo de mutação, o algoritmo visa obter novas e, possivelmente, melhores soluções. Os SIAs diferem de outras meta-heurísticas populacionais nos seguintes aspectos: (*i*) possuem capacidade de ajustar o tamanho da população automaticamente ao longo da busca, em resposta às particularidades do problema; (*ii*) são inerentemente capazes de manter a diversidade na população; e (*iii*) são dotados de um mecanismo eficiente de exploração do espaço de busca, o que lhes confere a competência para encontrar e preservar múltiplos ótimos locais.

Apesar dos sistemas imunológicos artificiais apresentarem bom desempenho para uma vasta gama de problemas, alguns aspectos podem ser melhorados. Por exemplo, muitos problemas de otimização

podem ser decompostos em sub-problemas, os quais são resolvidos separadamente e suas soluções parciais passam a ser combinadas posteriormente para compor a solução global. Nesse contexto, é possível que o vetor de atributos associado a cada solução candidata apresente soluções parciais para o problema. A essas soluções parciais, dá-se o nome de blocos construtivos (do inglês, *building blocks*).

Os sistemas imunológicos artificiais existentes na literatura não possuem recursos para lidarem eficientemente com blocos construtivos e para se beneficiarem do conhecimento já adquirido acerca do espaço de busca. O operador de mutação pode facilmente romper os blocos construtivos, pois ele não considera a interação dos caracteres presentes no anticorpo ao desempenhar sua função.

1.2 Objetivos da Pesquisa

O objetivo deste trabalho é investigar a sinergia entre sistemas imunológicos artificiais e modelos gráficos probabilísticos, procurando identificar em quais aspectos um pode se beneficiar do outro. Para melhor entendimento, o objetivo será decomposto em duas frentes.

O primeiro objetivo desta tese é propor e analisar a viabilidade de um sistema imunológico artificial para a construção de redes bayesianas. Já que o algoritmo é capaz de encontrar várias soluções de boa qualidade numa única execução, serão empregadas também metodologias para combinar, numa única estrutura, as diversas estruturas de redes bayesianas geradas pelo algoritmo. Espera-se que esta combinação leve à obtenção de uma rede que seja capaz de explicar os dados de forma consistente.

O segundo objetivo desta tese é estudar, desenvolver e aplicar sistemas imunológicos artificiais que superem as limitações apresentadas na Seção 1.1, particularmente quando aplicados ao tratamento de problemas de otimização. Essas limitações podem ser amenizadas tomando-se como inspiração a forma como os Algoritmos de Estimação de Distribuição (Mühlenbein & Paass, 1996) evoluem a população de soluções. Primeiramente, é construído um modelo probabilístico que represente a distribuição de probabilidades conjunta das melhores soluções. Em seguida, este modelo é utilizado para gerar novas soluções. Nos algoritmos imuno-inspirados propostos nesta tese, os operadores de clonagem e mutação são substituídos por uma rede bayesiana (otimização em espaços discretos) e uma rede gaussiana (otimização em espaços contínuos), devido à capacidade que elas apresentam de capturar adequadamente as relações mais expressivas das variáveis do problema. O posicionamento dos algoritmos propostos frente aos Algoritmos de Estimação de Distribuição é apresentado ao longo da tese.

Além da manipulação eficiente de blocos construtivos, pode-se citar mais algumas motivações para se utilizar essa nova forma de evoluir a população de soluções. Com a substituição dos

operadores de clonagem e mutação, não é preciso mais se preocupar com a escolha dos operadores e parâmetros mais adequados para cada tipo de problema. É evidente que a construção do modelo probabilístico requer a determinação de alguns parâmetros, mas estes são de mais fácil definição. Outra motivação é a possibilidade de analisar melhor e entender o processo de evolução do algoritmo. Por fim, a busca torna-se mais robusta, no sentido de que ocorre uma redução significativa na variação de desempenho entre buscas derivadas de re-execuções do algoritmo para um mesmo problema, e tende a ocorrer também uma redução acentuada no número de avaliações de soluções candidatas até que se atinja um certo nível de desempenho.

1.3 Contribuições Alcançadas

A seguir, são elencadas as principais contribuições deste trabalho:

- Estudo da viabilidade de se utilizar um sistema imunológico artificial como mecanismo de busca na tarefa de aprendizado de redes bayesianas;
- Apresentação das vantagens de se combinar múltiplas redes bayesianas;
- Proposição e formalização de quatro novos algoritmos imuno-inspirados capazes de lidar eficientemente com blocos construtivos:

    1. *Bayesian Artificial Immune System* (BAIS): um algoritmo imuno-inspirado para otimização mono-objetivo no espaço discreto, que utiliza rede bayesiana como modelo probabilístico para representar o relacionamento entre as variáveis;
    2. *Gaussian Artificial Immune System* (GAIS): um algoritmo imuno-inspirado para otimização mono-objetivo no espaço contínuo que utiliza rede gaussiana como modelo probabilístico;
    3. *Multi-objective Bayesian Artificial Immune System* (MOBAIS): versão do BAIS para problemas de otimização multiobjetivo;
    4. *Multi-objective Gaussian Artificial Immune System* (MOGAIS): versão do GAIS para problemas de otimização multiobjetivo.

- Avaliação dos algoritmos propostos em aplicações práticas, como síntese de *ensembles* de classificadores, seleção de atributos em problemas de classificação e otimização de funções no espaço contínuo.

1.4 Organização do Texto

Esta tese está organizada em 9 capítulos, conforme descrito a seguir.

No Capítulo 1, são discutidos os principais aspectos que levaram ao desenvolvimento desta pesquisa. As motivações, objetivos e contribuições são apresentadas também neste capítulo.

No Capítulo 2, são apresentados os conceitos básicos de redes bayesianas e redes gaussianas, seus componentes e as formas de construí-las, dando ênfase especial ao paradigma de aprendizado baseado em busca e pontuação. No intuito de proporcionar um melhor entendimento dos conceitos e funcionamento desses modelos gráficos probabilísticos, primeiramente é realizada uma breve introdução à teoria de probabilidade.

No Capítulo 3, é realizada uma breve introdução ao sistema imunológico natural visando fundamentar os aspectos biológicos que serviram de inspiração para a criação de um novo paradigma computacional bio-inspirado. Será dada ênfase particular às teorias da seleção clonal e da rede imunológica. Em seguida, são apresentados os principais conceitos de sistemas imunológicos artificiais, suas vantagens e limitações para a solução de problemas de otimização em espaços discretos e contínuos.

No Capítulo 4, é descrita a primeira contribuição desta pesquisa, que consiste em investigar o aprendizado de redes bayesianas por meio de sistemas imunológicos artificiais. São apresentados os componentes do algoritmo, mais especificamente a forma de codificação das soluções candidatas, a função de *fitness* e operadores de clonagem, mutação e supressão. Os resultados obtidos junto a casos de estudo comumente adotados na literatura são então descritos e analisados.

No Capítulo 5, está a maior contribuição deste trabalho. Nele, é proposto um novo paradigma para o desenvolvimento de sistemas imunológicos artificiais capazes de lidar eficientemente com blocos construtivos. Quatro novos algoritmos resultantes deste novo paradigma foram concebidos para resolver problemas de otimização mono-objetivo e multiobjetivo, tanto no espaço discreto como no espaço contínuo. Os principais trabalhos da literatura que estão relacionados às contribuições desta tese são apresentados também.

No Capítulo 6, são descritos os experimentos conduzidos para avaliar a viabilidade do algoritmo BAIS quando aplicado à geração de *ensembles* de redes neurais.

No Capítulo 7, são apresentados os experimentos realizados com os algoritmos BAIS e MOBAIS, aplicados à tarefa de seleção de atributos em problemas de classificação de padrões.

No Capítulo 8, são apresentados os experimentos realizados com os algoritmos GAIS e MOGAIS, aplicados em problemas de otimização no espaço contínuo.

Por fim, no Capítulo 9, são apresentados os comentários finais e as perspectivas futuras da pesquisa.

Capítulo 2

Modelos Gráficos Probabilísticos

2.1 Considerações Iniciais

Modelos gráficos probabilísticos compreendem um conjunto de modelos que utilizam as teorias da probabilidade e dos grafos para representar e manipular conhecimento nas mais diversas áreas, procurando amenizar dois problemas que ocorrem em situações práticas do mundo real: incerteza e complexidade.

Um modelo gráfico probabilístico é composto de duas partes, uma qualitativa e outra quantitativa. A parte qualitativa é expressa por um grafo em que os nós representam variáveis aleatórias e os arcos que as relacionam denotam a dependência estatística entre elas. O segundo componente, quantitativo, é a distribuição de probabilidades para essas variáveis.

Existem dois principais tipos de grafos que são utilizados para representar a dependência entre as variáveis em modelos gráficos probabilísticos: os não-direcionados e os direcionados. Em um grafo não-direcionado, os arcos não possuem direção. Pode-se citar as redes de Markov como exemplos de modelos gráficos que utilizam este tipo de estrutura. Já nos grafos direcionados, os arcos possuem direção. As redes bayesianas e as redes gaussianas são exemplos de modelos gráficos que utilizam este tipo de grafo, os quais devem ser acíclicos. Neste trabalho, serão abordados apenas os modelos gráficos probabilísticos cuja estrutura é um grafo direcionado e acíclico.

Portanto, a finalidade deste capítulo é apresentar os conceitos básicos de redes bayesianas e redes gaussianas, seus componentes e as formas de construí-las, dando ênfase especial ao paradigma de aprendizado baseado em busca e pontuação. Para tanto, a fim de proporcionar um melhor entendimento dos conceitos e funcionamento desses modelos gráficos probabilísticos, primeiramente se faz necessária a introdução de alguns conceitos de probabilidade, como probabilidade conjunta e condicional, variável aleatória e distribuição de probabilidades.

2.2 Conceitos Básicos de Probabilidade

A teoria da probabilidade fornece um conjunto formal de mecanismos para lidar com situações em que a incerteza está presente. Foi a partir desses mecanismos, aliados à teoria dos grafos, que surgiram os modelos gráficos probabilísticos. Nesta seção, são introduzidos os conceitos básicos de probabilidade necessários para melhor compreensão das vantagens oferecidas pelos modelos gráficos probabilísticos. O formalismo a seguir foi baseado em Papoulis & Pillai (2002).

2.2.1 Experimento aleatório, espaço amostral e eventos

Um experimento aleatório é aquele em que o resultado não pode ser predito, mesmo que ele seja repetido várias vezes nas mesmas condições. O conjunto de todos os possíveis resultados de um experimento aleatório é chamado de espaço amostral, denotado por Ω. Se os elementos do espaço amostral são finitos e contáveis, o espaço amostral é dito discreto. Caso os possíveis resultados do experimento sejam infinitos, o espaço amostral é dito ser contínuo. É interessante observar que, para um dado experimento aleatório, o espaço amostral associado a ele pode não ser único e sua elaboração depende do ponto de vista adotado, bem como das questões a serem respondidas. Por exemplo, quando retirada uma carta de um baralho de 52 cartas, o interesse poderia ser pelo valor dela (Ás até o Rei). Por outro lado, o interesse poderia ser pelo naipe (copa, ouro, espada ou paus).

Um subconjunto do espaço amostral é chamado de evento. Dado um experimento aleatório, para cada evento A do espaço amostral é atribuído um número real no intervalo [0,1], chamado de probabilidade e denotado por $P(A)$, o qual fornece uma medida numérica da chance de ocorrer o evento A. A medida de probabilidade possui as seguintes propriedades, chamadas de axiomas da probabilidade:

- Axioma 1: $P(A) \geq 0$;
- Axioma 2: $P(\Omega) = 1$;
- Axioma 3: Para dois eventos que não podem ocorrer simultaneamente, A e B, a probabilidade de ocorrer um ou outro é $P(A \text{ ou } B) = P(A) + P(B)$.

2.2.2 Como obter a probabilidade de algum evento

Existem duas abordagens para definir a probabilidade de um evento, a clássica e a frequência relativa.

A abordagem clássica é empregada quando os eventos são equiprováveis. Portanto, a probabilide $P(A)$ para um evento A é dada por:

$$P(A) = \frac{N_A}{N}. \tag{2.1}$$

em que N_A denota o número de casos favoráveis a A e N é o número de resultados possíveis.

Exemplo 2.1 Considere o experimento aleatório de lançar um dado honesto. O conjunto de todos os possíveis resultados, isto é, o espaço amostral é $\Omega = \{1, 2, 3, 4, 5, 6\}$.

(a) Qual a probabilidade do evento A="obter o número 5 na face voltada para cima"?

Dentre os 6 possíveis resultados para este experimento, somente um único resultado é favorável ao evento de interesse. Portanto, $P(A) = \frac{1}{6}$.

(b) Qual a probabilidade do evento B="obter um número par na face voltada para cima do dado"? Dentre os 6 possíveis resultados, três deles são favoráveis: $\{2,4,6\}$. Portanto, $P(B) = \frac{3}{6}$.

Na maioria das situações práticas, os eventos simples do espaço amostral não são equiprováveis e não se pode calcular probabilidades usando a abordagem clássica. Neste caso, as probabilidades são calculadas como frequência relativa de um evento, ao se repetir o mesmo experimento várias vezes. Na abordagem frequencista, a probabilidade de um evento A é dada por:

$$P(A) = \lim_{n\to\infty} \frac{n_A}{n} \tag{2.2}$$

em que n_A denota o número de ocorrências de A e n é o número de repetições do experimento.

Exemplo 2.2 Uma moeda foi lançada 100 vezes e a face "cara" foi obtida 60 vezes. É razoável estimar a probabilidade de obter "cara" nos lançamentos futuros como sendo 0,60.

2.2.3 Operações com eventos

A teoria da probabilidade faz uso extensivo da teoria dos conjuntos. Tendo em vista que cada evento é considerado um subconjunto do espaço amostral, todas operações relativas a conjuntos, como intersecção união e complemento, podem ser aplicadas a eventos.

Considere o experimento aleatório de jogar um dado honesto e observar o número em sua face voltada para cima. Considere também os eventos A="obter número par", B="obter número maior que 2". O espaço amostral para este experimento é $\Omega = \{1, 2, 3, 4, 5, 6\}$. Pela abordagem clássica de atribuição de probabilidades, $P(A) = \frac{3}{6}$, pois o evento A compreende o conjunto $\{2,4,6\}$. Já a probabilidade para o evento B é $P(B) = \frac{4}{6}$, uma vez que B denota o conjunto $\{3,4,5,6\}$.

A intersecção dos eventos A e B, denotada por $A \cap B$ ou AB, é o conjunto consistindo de todos os elementos comuns a A e B. A intersecção AB representa a ocorrência simultânea dos eventos A e B. A probabilidade de ocorrem os eventos A e B simultaneamente (obter número par e este número

ser maior que 2) é dada por $P(AB) = \frac{2}{6}$, pois a intersecção de A=$\{$2,4,6$\}$ e B=$\{$3,4,5,6$\}$ é o conjunto $\{$4,6$\}$. Logo, existem dois elementos favoráveis dentre os seis possíveis resultados. Se a intersecção de dois eventos é o conjunto vazio, então estes eventos são mutuamente exclusivos. Isto significa que ambos não podem ocorrer simultaneamente.

A união dos eventos A e B, denotada por $A \cup B$, representa a ocorrência de pelo menos um destes eventos. Repare que ambos os eventos não são mutuamente exclusivos, pois possuem elementos em comum, $\{$4,6$\}$. Logo, não é possível calcular $P(A \cup B)$ utilizando o Axioma 3. É preciso subtrair a probabilidade da intersecção, $P(AB)$. Portanto, a probabilidade de obter número par ou número maior que 2 é $P(A \cup B) = P(A) + P(B) - P(AB) = \frac{3}{6} + \frac{4}{6} - \frac{2}{6} = \frac{5}{6}$.

O complemento de um evento A, denotado por $\bar{A}$, consiste de todos os elementos do espaço amostral que não estão em A. Portanto, $P(\bar{A}) = 1 - P(A) = 1 - \frac{3}{6} = \frac{3}{6}$. Da mesma forma, $P(\bar{B}) = 1 - P(B) = 1 - \frac{4}{6} = \frac{2}{6}$.

2.2.4 Probabilidade condicional e probabilidade conjunta

Em alguns casos, a ocorrência de um evento pode ser influenciada pela ocorrência de outro. A probabilidade de um evento A, dado que o evento B ocorreu, é chamada de probabilidade condicional de A dado B, a qual é denotada por $P(A|B)$ e definida como:

$$P(A|B) = \frac{P(AB)}{P(B)}, \tag{2.3}$$

em que $P(B) > 0$ e $P(AB)$ é a probabilidade conjunta de A e B, ou seja, a probabilidade desses dois eventos ocorrerem ao mesmo tempo.

Exemplo 2.3 No experimento de lançar um dado honesto e observar o número da face voltada para cima, suponha os eventos A=“obter número 2” e B=“obter número par”. Sabe-se que $P(A)$=1/6, $P(B)$=1/2 e $P(AB)$=1/6. Pela Equação (2.3) tem-se:

$$P(A|B) = \frac{P(AB)}{P(B)} = \frac{1/6}{1/2} = 1/3.$$

Dado que ocorreu o evento B=“obter número par”, a probabilidade de ocorrer o evento A=“obter número 2” é alterada e passa a ser 1/3.

Repare que, pela Equação (2.3), pode-se expressar a probabilidade conjunta dos eventos A e B como:

$$P(AB) = P(A|B)P(B) = P(B|A)P(A). \tag{2.4}$$

A Equação (2.4) é conhecida como regra do produto. Generalizando a regra do produto para *n* eventos, tem-se:

$$P(A_1A_2...A_n) = P(A_n|A_{n-1}...A_1)...P(A_2|A_1)P(A_1). \tag{2.5}$$

2.2.5 Eventos independentes e independência condicional

Entre todos os conceitos envolvidos na teoria da probabilidade, existem dois que são peças-chave para um modelo gráfico probabilístico, como será visto na seção seguinte: a independência e a independência condicional. Dois ou mais eventos são independentes se a ocorrência de um não afeta a probabilidade de ocorrência dos demais. Por exemplo, dois eventos A e B são independentes se

$$\begin{aligned} P(A|B) &= P(A) \\ P(B|A) &= P(B). \end{aligned} \tag{2.6}$$

Logo, a probabilidade conjunta desses dois eventos independentes é obtida pela combinação das equações (2.4) e (2.6), resultando em:

$$P(AB) = P(A)P(B). \tag{2.7}$$

Agora, considere os três eventos A, B e C. Diz-se que o evento A é condicionalmente independente de B, dado um terceiro evento C, se

$$\begin{aligned} P(AB|C) &= P(A|C)P(B|C) \\ P(A|BC) &= P(A|C). \end{aligned} \tag{2.8}$$

A Equação (2.8) retrata que, conhecendo o evento C, qualquer informação sobre B não altera a probabilidade de ocorrência do evento A.

Exemplo 2.4 **(Eventos independentes)** Considere o experimento de lançar dois dados honestos e observar as suas faces voltadas para cima. Sejam os eventos A="obter o número 2 no primeiro dado" e B="obter um número par no segundo dado". Sabe-se que $P(A)$=1/6 e $P(B)$=1/2 e $P(AB)$=1/12.

É fácil verificar que estes eventos são independentes, pois

$$P(A|B) = \frac{P(AB)}{P(B)} = \frac{1/12}{1/2} = 1/6 = P(A).$$

Da mesma forma,

$$P(B|A) = \frac{P(AB)}{P(A)} = \frac{1/12}{1/6} = 1/2 = P(B).$$

Exemplo 2.5 **(Independência condicional)** No experimento de lançar uma mesma moeda duas vezes, considere os eventos A="resultado do primeiro lançamento" e B="resultado do segundo lançamento". Agora, assuma que existe a possibilidade da moeda estar viciada, favorecendo "cara". Nesse caso, A e B não são independentes. Por exemplo, ao observar que A é "cara" aumentaria a crença que B também será "cara". Os eventos A e B são dependentes de um terceiro evento C="moeda é viciada". Desta forma, A e B tornam-se condicionalmente independentes dado o evento C. O conhecimento de A não pode alterar a crença em B.

2.2.6 Probabilidade Total e Teorema de Bayes

Em alguns casos, a probabilidade de um evento não pode ser calculada diretamente, mas pode ser obtida a partir de uma ponderação de várias probabilidades condicionais. Seja $\{A_1, ...A_n\}$ uma partição de Ω, tal que $\cup A_i = \Omega$, $A_i \cap A_j = \emptyset \ \forall i, j$ e $P(A_i) > 0 \ \forall i$, e B um evento arbitrário, como apresentado na Figura 2.1.

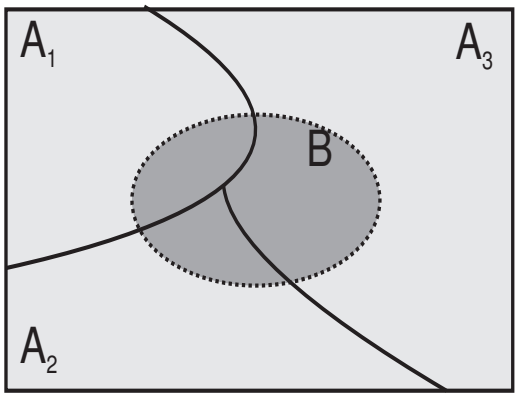


Figura 2.1: Partição do espaço amostral.

Claramente, $B = B \cap \Omega = B \cap (A_1 \cup ... \cup A_n) = (B \cap A_1) \cup ... \cup (B \cap A_n)$. Os eventos $B \cap A_i$ e $B \cap A_j$ são mutuamente exclusivos, uma vez que A_i e A_j o são. Desta forma, a probabilidade de B pode ser expressa como:

$$P(B) = P(BA_1) + ... + P(BA_n).$$

De acordo com a Equação (2.3), os termos $P(BA_i)$ são obtidos da seguinte forma:

$$P(BA_i) = P(B|A_i)P(A_i). \tag{2.9}$$

Logo, pode-se reescrever a probabilidade do evento B como sendo:

$$P(B) = P(B|A_1)P(A_1) + ... + P(B|A_n)P(A_n). \tag{2.10}$$

Este resultado é conhecido como *teorema da probabilidade total*.

Pela Equação (2.4), pode-se reescrever a Equação (2.9) da seguinte forma:

$$P(BA_i) = P(A_i|B)P(B).$$

Com isso, chega-se a um importante resultado do teorema da probabilidade total:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}. \tag{2.11}$$

A Equação (2.11), conhecida como *teorema de Bayes* e desenvolvida pelo matemático inglês Thomas Bayes, provê um mecanismo para ajustar a probabilidade de um evento conforme novas informações são obtidas. Em outras palavras, ele permite combinar novas informações com o conhecimento já existente.

Exemplo 2.6 Suponha que existam quatro caixas. Na caixa 1 há 2000 componentes eletrônicos dos quais 5% são defeituosos. A caixa 2 contém 500 componentes, sendo que 40% apresentam defeitos. As caixas 3 e 4 contêm 1000 componentes cada com 10% sendo defeituosos. Uma caixa é selecionada aleatoriamente e uma peça é retirada aleatoriamente também.

(a) Qual a probabilidade do componente retirado ser defeituoso?

Será denotado por A_i o evento consistindo de todos os componentes da *i*-ésima caixa. O evento consistindo de todos os componentes defeituosos será denotado por B.

Dado que as caixas são escolhidas aleatoriamente, claramente $P(A_1)$=$P(A_2)$=$P(A_3)$=$P(A_4)$= 0,25. Conforme o enunciado, a probabilidade de um componente retirado de uma caixa específica ser defeituoso é :

$$P(B|A_1) = 0,05 \quad P(B|A_2) = 0,4$$
$$P(B|A_3) = 0,1 \quad P(B|A_4) = 0,1.$$

Uma vez que os eventos A_1, A_2, A_3 e A_4 formam uma partição do Ω, pode-se concluir que

$$P(B) = 0,05 \times 0,25 + 0,4 \times 0,25 + 0,1 \times 0,25 + 0,1 \times 0,25 = 0,1625.$$

Logo, a probabilidade do componente escolhido ser defeituoso é 0,1625.

(b) O componente retirado foi examinado e constatou-se que ele é defeituoso. Na presença dessa informação, qual a probabilidade que ele tenha vindo da caixa 2?

O que se deseja calcular é $P(A_2|B)$. Tendo em vista que $P(B)$=0,1625, $P(B|A_2)$=0,4 e $P(A_2)$= 0,25, a probabilidade dele ter vindo da caixa 2 é dada por:

$$P(A_2|B) = 0,4 \times \frac{0,25}{0,1625} = 0,615.$$

2.2.7 Variável aleatória

Uma variável aleatória X é uma função que atribui, para cada elemento do espaço amostral de um experimento aleatório, um número real $x \in \mathbb{R}$, conforme ilustrado na Figura 2.2.

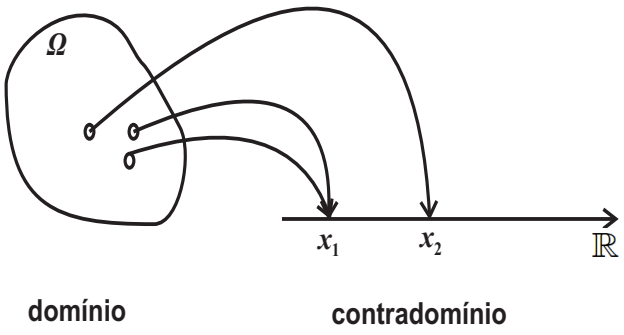


Figura 2.2: Exemplo de variável aleatória mapeando cada elemento do espaço amostral a um número.

O espaço amostral é denominado de domínio da variável X e o conjunto de todos os valores de X é denominado contradomínio da variável. Dois ou mais elementos do domínio podem estar associados a um mesmo valor da variável aleatória. Entretanto, dois ou mais valores do contradomínio não podem estar relacionados com o mesmo elemento do domínio.

Se os possíveis resultados de um experimento aleatório já forem numéricos, pode-se interpretá-los como uma variável aleatória definida pela função identidade.

Dependendo dos valores que a variável pode assumir, ela é classificada em discreta ou contínua. Uma variável aleatória discreta é aquela que assume um conjunto de valores finito ou infinito enumerável, como a do Exemplo 2.7. Caso contrário, ela é dita ser contínua. Por convenção, variáveis aleatórias são representadas por letras maiúsculas, por exemplo X, Y e Z, enquanto os valores que a variável pode assumir são representados por letras minúsculas, como x, y e z.

2.2.8 Função Massa de Probabilidade e Função Densidade de Probabilidade

Se X é uma variável aleatória e x é um número fixado no contradomínio, pode-se definir o evento A_x como o subconjunto de Ω consistindo de todos os elementos para os quais a variável aleatória X associa o número x. Esse evento A_x possui, portanto, uma probabilidade $P(A_x)$, a qual pode ser interpretada como a probabilidade da variável aleatória X assumir o valor x. Para toda variável aleatória, existe uma distribuição de probabilidades que relaciona cada valor da variável com a sua probabilidade de ocorrência. Existem as distribuições discretas e as contínuas.

No caso discreto, a distribuição é caracterizada pela função massa de probabilidade, a qual fornece a probabilidade da variável discreta X assumir o valor x e é denotada por $p(x)$. Essa função possui as seguintes propriedades:

1. $p(x) \geq 0, \forall x \in X$
2. $\sum_x p(x) = 1$

Exemplo 2.7 Considere o experimento aleatório de lançar duas moedas e observar as suas faces voltadas para cima. O espaço amostral é Ω={(cara, coroa), (coroa, cara), (cara, cara), (coroa, coroa)} Pode-se definir, por exemplo, a variável aleatória X que representa o número de "caras" obtidas. Os possíveis valores que X pode assumir são 0, 1 ou 2, dependendo do resultado obtido no experimento. Na Tabela 2.1, é apresentada a probabilidade da variável X assumir o valor x_i. O gráfico da função massa de probabilidade está ilustrado na Figura 2.3.

Tabela 2.1: Distribuição de probabilidade para a variável X.

x_i	$p(x_i) = P(X = x_i)$
0	0,25
1	0,5
2	0,25

Para uma variável aleatória contínua X, as probabilidades são expressas em relação a intervalos de valores. Distribuições contínuas são caracterizadas pela função densidade de probabilidade, denotada por $f(x)$, que deve satisfazer as condições:

1. $f(x) \geq 0, \quad \forall x \in \mathbb{R}$
2. $\int_{-\infty}^{\infty} f(x)dx = 1$
3. $P(a \leq X \leq b) = \int_a^b f(x)dx, \quad \forall a, b$

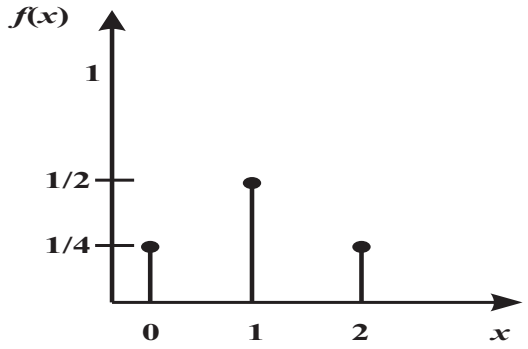


Figura 2.3: Gráfico da função massa de probabilidade para a variável discreta X do Exemplo 2.7.

A função densidade de probabilidade não representa probabilidades. Somente quando ela for integrada entre dois limites ela produzirá probabilidade, que é a área sob a curva da função entre os valores considerados. Portanto, a probabilidade da variável X assumir um valor específico x é zero, pois $\int_x^x f(x)dx = 0$.

Dentre as funções densidade de probabilidade, uma das mais conhecidas e utilizadas para modelar fenômenos da vida real é a gaussiana:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \tag{2.12}$$

em que μ e $\sigma > 0$ são números reais que representam a média e o desvio padrão, respectivamente, da variável aleatória. o termo σ^2 é chamado de variância. Para indicar que uma variável aleatória X possui distribuição gaussiana com média μ e variância σ^2, utiliza-se a notação $X \sim \mathcal{N}(\mu, \sigma^2)$. Na Figura 2.4, é ilustrado o gráfico da função densidade de probabilidade gaussiana com μ=0 e σ=1.

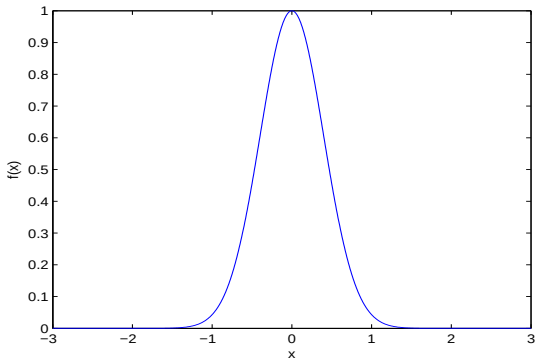


Figura 2.4: Gráfico da função densidade de probabilidade gaussiana.

2.2.9 Múltiplas variáveis aleatórias

Geralmente, em situações práticas, modelos probabilísticos envolvem mais de uma variável aleatória de interesse. Por exemplo, a caracterização do peso e altura numa determinada população e o estudo da temperatura e pressão em um certo experimento. Nesta subseção, serão estendidos os conceitos abordados anteriormente de probabilidade conjunta, probabilidade condicional e independência, no contexto de múltiplas variáveis aleatórias.

Distribuição de probabilidade conjunta

A distribuição de probabilidade conjunta para n variáveis aleatórias $X_1, X_2, ..., X_n$ discretas é caracterizada pela função massa de probabilidade conjunta, definida como:

$$p(\boldsymbol{x}) = p(x_1 x_2 ... x_n) = p(X_1 = x_1, X_2 = x_2, ..., X_n = x_n), \tag{2.13}$$

sendo que $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ e x_i é um possível valor da variável X_i, $\forall i$.

A função massa de probabilidade conjunta para as n variáveis aleatórias discretas possui as seguintes propriedades:

1. $p(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x}$
2. $\sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1$

No caso em que as variáveis $X_1, X_2, ..., X_n$ são contínuas, a distribuição de probabilidades conjunta para é caracterizada pela função densidade de probabilidade conjunta, denotada por $f(\boldsymbol{x}) = f(x_1 x_2 ... x_n)$, a qual satisfaz as condições:

1. $f(\boldsymbol{x}) \geq 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n$
2. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} f(x_1 x_2 ... x_n) dx_1 dx_2 ... dx_n$ = 1
3. $P(a_1 \leq X_1 \leq b_1, a_2 \leq X_2 \leq b_2, ..., a_n \leq X_n \leq b_n) =$
   $\int_{a_1}^{b_1} \int_{a_2}^{b_2} ... \int_{a_n}^{b_n} f(x_1 x_2 ... x_n) dx_1 dx_2 ... dx_n, \quad \forall a_i, b_i, i = 1...n$

Distribuição marginal

Dada uma distribuição de probabilidade conjunta para n variáveis aleatórias, é possível determinar a probabilidade de uma delas sem considerar as demais. Esta distribuição é chamada de marginal.

Para duas variáveis aleatórias discretas X e Y, cuja distribuição conjunta é conhecida, a distribuição marginal de X é obtida somando a probabilidade conjunta levando em consideração todos os possíveis valores de Y, conforme expresso na Equação (2.14).

$$p(x) = p(X = x) = \sum_{y} p(xy). \tag{2.14}$$

Para o caso em que X e Y forem variáveis contínuas, a distribuição marginal de X é obtida integrando a probabilidade conjunta levando em relação a Y, como apresentado na Equação (2.15):

$$f(x) = \int_{-\infty}^{\infty} f(xy)dy. \tag{2.15}$$

Distribuição de probabilidade condicional e independência

Sejam X e Y duas variáveis aleatórias definidas sobre um determinado experimento. A probabilidade condicional de X, dado Y é expressa por:

$$p(x|y) = p(X = x|Y = y) = \frac{p(xy)}{p(y)}. \tag{2.16}$$

Se X e Y forem variáveis contínuas, a probabilidade condicional de X, dado Y é:

$$f(x|y) = \frac{f(xy)}{f(y)}. \tag{2.17}$$

Se no caso discreto $p(xy) = p(x)p(y)\ \forall x, y$, e no caso contínuo $f(xy) = f(x)f(y)\ \forall x, y$, então diz-se que as variáveis X e Y são independentes.

Para efeito de simplicidade, no restante deste capítulo as variáveis discretas e contínuas serão tratadas com uma mesma notação, sem perda de generalidade. Isso significa que todas as referências à natureza da variável (discreta ou contínua) serão omitidas a partir daqui. Com isso, a função massa de probabilidade $p(x)$ e a função densidade de probabilidade $f(x)$ serão substituídas pelo termo mais geral $\rho(x)$, que não reflete a natureza da variável aleatória X.

2.3 Modelos Gráficos Probabilísticos

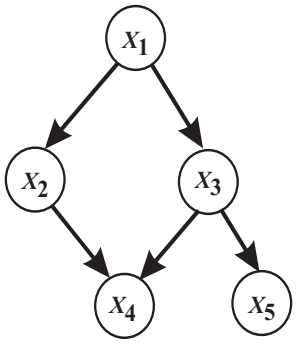
Formalmente, um modelo gráfico probabilístico para um conjunto X=$\{X_1, X_2, ..., X_n\}$ de variáveis aleatórias é uma fatoração gráfica da distribuição de probabilidade conjunta de X. O modelo gráfico consiste de uma estrutura em grafo que representa as dependências condicionais das variáveis em X e de um conjunto de probabilidades para cada variável (Cowell et al., 1999; Jensen, 1996, 2001; Pearl, 1988).

Como exposto anteriormente, o foco desta tese está em modelos gráficos probabilísticos cuja estrutura é um grafo direcionado acíclico, no qual:

- cada nó corresponde a uma variável aleatória;
- os arcos ligando os nós indicam as relações de dependência entre as variáveis. Um arco saindo de uma variável X_i e chegando numa variável X_j significa que "X_i é pai de X_j" e que "X_j é filha de X_i";
- cada variável possui uma distribuição de probabilidades.

Se as variáveis forem discretas, os modelos recebem o nome de rede bayesiana. Se as variáveis forem contínuas, o modelo é chamado de rede gaussiana.

Na Figura 2.5 é apresentado um exemplo de modelo gráfico probabilístico que ilustra os conceitos definidos previamente. O conjunto de variáveis X=$\{X_1, X_2, X_3, X_4, X_5\}$ retrata as variáveis do modelo, que são representadas pelos nós do grafo.



$$\rho(x_1x_2x_3x_4x_5) = \rho(x_1)\rho(x_2|x_1)\rho(x_3|x_1)\rho(x_4|x_2x_3)\rho(x_5|x_3)$$

Figura 2.5: Exemplo de modelo gráfico probabilístico e a probabilidade conjunta definida por este modelo.

Além da dependência condicional explícita entre as variáveis, os modelos gráficos representam implicitamente as independências condicionais entre elas. Uma variável é dita ser condicionalmente independente de outras que não são suas filhas, dados os seus pais. Na rede da Figura 2.5, a variável X_5 é condicionalmente independente de X_1, X_2 e X_4, dado seu pai X_3. Desta forma, $\rho(x_5|x_1x_2x_3x_4)$ é o mesmo que $\rho(x_5|x_3)$.

Portanto, os modelos gráficos probabilísticos são capazes de expressar a probabilidade conjunta das variáveis de uma forma mais compacta. Assim, a distribuição conjunta das variáveis do modelo da Figura 2.5 pode ser expressa por:

$$\rho(x_1x_2x_3x_4x_5) = \rho(x_1)\rho(x_2|x_1)\rho(x_3|x_1)\rho(x_4|x_2x_3)\rho(x_5|x_3). \tag{2.18}$$

De uma forma mais geral, a distribuição conjunta de X é dada por:

$$\rho(x_1x_2x_3x_4x_5) = \prod_{i=1}^{5} \rho(x_i|\boldsymbol{pa_i}). \tag{2.19}$$

em que x_i é um possível valor da variável aleatória X e $\boldsymbol{pa_i}$ é o conjunto de pais da variável X_i. Por exemplo, na Figura 2.5, $\boldsymbol{pa_4}$=$\{X_2, X_3\}$. O termo $\rho(x_i|\boldsymbol{pa_i})$ é a probabilidade condicional de x_i, dados os valores das variáveis contidas no conjunto $\boldsymbol{pa_i}$.

É a independência condicional que permite graficamente decompor a distribuição de probabilidade conjunta generalizada de X em produtos, levando a uma redução no número de parâmetros necessários para especificar o modelo.

2.4 *Markov Blanket*

Para um determinado nó A, existe um conjunto único de nós que exercem influência na distribuição de probabilidades desse nó A. Para esse conjunto de nós, dá-se o nome de *Markov blanket* do nó A. Isso significa que todo o conhecimento necessário para predizer o comportamento de um determinado nó encontra-se exclusivamente no seu *Markov blanket* e os nós que não fazem parte do seu *Markov blanket* não alteram sua distribuição de probabilidade.

O *Markov blanket* de um nó A consiste dos pais e filhos de A e dos demais pais dos filhos de A. Na Figura 2.6, é apresentado o *Markov blanket* (nós cinzas) para o nó A em um modelo gráfico probabilístico fictício.

O nó A é condicionalmente independente de todos os outros nós da rede, dado o seu *Markov blanket*.

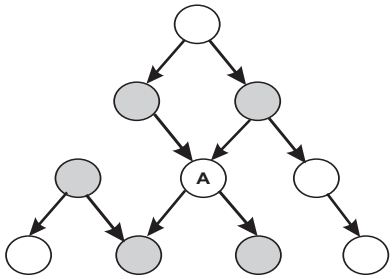


Figura 2.6: Os nós em cinza representam o *Markov blanket* do nó A.

2.5 Aprendizado em Modelos Gráficos Probabilísticos

A construção de modelos gráficos probabilísticos é feita tipicamente por um especialista no domínio do problema que se está tentando modelar. Quanto mais conhecimento sobre o problema

o especialista possuir, mais acurado será o modelo. Entretanto, dependendo da natureza desse problema, tal processo pode ser difícil e demorado ou o especialista pode não estar disponível. Dessa forma, nos últimos anos têm sido utilizadas técnicas automáticas que geram o modelo a partir de dados numéricos que representam amostras ou exemplos do problema. Quanto mais informação sobre o problema os dados representarem, mais acurado será o modelo.

O aprendizado de modelos gráficos probabilísticos pode ser dividido em duas etapas. A primeira é a definição das relações de interdependência entre as variáveis, ou seja, a estrutura da rede. Uma vez que a estrutura da rede foi definida, entra em cena a segunda etapa, que é o cálculo das probabilidades para as variáveis.

As distribuições de probabilidades podem ser estimadas usando o método da máxima probabilidade a *posteriori* ou da máxima verossimilhança. Ambos os métodos são computados facilmente a partir dos dados e os algoritmos existentes na literatura são eficientes (Buntine, 1996).

O aprendizado da estrutura da rede consiste em determinar relações entre as variáveis, adicionar ou excluir arcos, estabelecer direções, enfim, definir um grafo direcionado acíclico para o qual serão calculadas distribuições de probabilidade. Este é um problema NP-completo (Buntine, 1996). Em geral, há dois paradigmas principais de aprendizado a partir de dados (Cheng et al., 1997): independência condicional e busca e pontuação.

2.5.1 Paradigma baseado em independência condicional

Os algoritmos pertencentes ao paradigma de independência condicional tentam descobrir a estrutura da rede por meio da identificação das variáveis que são condicionalmente independentes das demais variáveis do problema. As relações de dependência são avaliadas pelo uso de alguma classe de teste de independência condicional (IC). De acordo com Cheng et al. (1997), são poucos os algoritmos nesta categoria e os existentes são computacionalmente custosos e quase impraticáveis para instâncias de tamanho elevado. Os algoritmos *Wermuth-Lauritzen* (Wermuth & Lauritzen, 1983) e *SRA* (Srinivas et al., 1990) são indicados para grandes conjuntos de dados com poucas variáveis, uma vez que o teste de independência condicional entre as variáveis é custoso computacionalmente e requer número de amostras elevado. Outros algoritmos similares a estes dois podem ser encontrados em Cheng et al. (1997) e Spirtes et al. (1991).

2.5.2 Paradigma baseado em busca e pontuação

Já no paradigma de busca e pontuação, o aprendizado é visto como um problema de otimização. Sob esta perspectiva, tem-se o espaço de busca contendo todos os possíveis grafos acíclicos candidatos para o problema em questão, um mecanismo de busca para explorar este espaço e uma

forma de avaliar cada solução. Normalmente, a busca pelo modelo gráfico probabilístico que melhor representa o conjunto de dados se realiza por meio de heurísticas, dado que explorar todo o espaço de busca é um problema NP-completo (Buntine, 1996).

Pode-se perceber, então, que neste paradigma de aprendizado dois elementos fundamentais são necessários: (*i*) o mecanismo de busca; e (*ii*) a forma de avaliar as redes encontradas.

Com relação ao mecanismo de busca, o mais usual é utilizar o algoritmo subida da encosta (*hill climbing*) (Blanco et al., 2004; Chickering, 2002; Chickering et al., 1995; Friedman, 1997; Heckerman, 1995). Embora mecanismos como o subida da encosta e suas variantes sejam amplamente utilizados, eles são fortemente sujeitos a mínimos locais, dado que eles apenas refinam localmente as soluções-candidatas. Pensando nesse inconveniente, alguns pesquisadores propuseram metodologias que utilizam mecanismos de busca baseados em populações de soluções, como algoritmos genéticos (Etxeberria et al., 1997; Larrañaga et al., 1996; Myers et al., 1999), sistemas imunológicos artificiais (Castro & Von Zuben, 2005) e colônia de formigas (Campos et al., 2002).

Entretanto, de acordo com Buntine (1996), o espaço de busca no problema do aprendizado de modelos gráficos probabilísticos é amplo e multimodal. Portanto, é fundamental que o mecanismo de busca consiga trabalhar eficientemente com espaços multimodais, característica essa que os algoritmos genéticos e colônia de formigas não apresentam. Embora técnicas de *niching*, como *fitness sharing*, possam ser incorporadas nesses algoritmos, visando suprir essas deficiências, eles ainda apresentam desempenho aquém do desejado quando comparados com sistemas imunológicos artificiais, que são algoritmos inerentemente capazes de lidar com espaços de busca amplos e multimodais. O Capítulo 4 se dedica especificamente a este assunto.

Dentre os critérios de pontuação mais utilizados estão as métricas bayesianas (Buntine, 1994; Cooper & Herskovits, 1992; Heckerman et al., 1995; Hruschka Jr & Ebecken, 2003), a métrica baseada na entropia (Acid & Campos, 1996) e a métrica do Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*) (Grünwald, 2000; Lam & Bacchus, 1994; Suzuki, 1996). Outros critérios clássicos para seleção de modelos também podem ser empregados para avaliação das redes bayesianas, como BIC (*Bayesian Information Criterion*) (Schwarz, 1978) e AIC (*Akaike's Information Criterion*) (Akaike, 1974). As métricas dependem do tipo de modelo gráfico probabilístico considerado. Portanto, algumas delas serão detalhadas formalmente após a definição de redes bayesianas e redes gaussianas.

2.6 Redes Bayesianas

No caso em que X=$\{X_1, X_2,...,X_n\}$ é uma variável aleatória discreta *n*-dimensional, o modelo gráfico probabilístico para X é chamado de rede bayesiana (Pearl, 1988). A decomposição em

produtos da distribuição de probabilidade conjunta para X adquire a seguinte forma, quando o modelo utilizado é uma rede bayesiana:

$$p(x) = p(x_1 x_2 \dots x_n) = \prod_{i=1}^{n} p(x_i | \boldsymbol{pa_i}). \tag{2.20}$$

Note que a Equação (2.20) é um caso particular da Equação (2.19), levando em consideração a natureza discreta das variáveis.

As redes bayesianas estão cada vez mais sendo utilizadas em problemas do mundo real. Para a tarefa de mineração de dados em bioinformática, elas são utilizadas para análise de expressões gênicas e para descobrir as relações entre os genes (Friedman et al., 2000), bem como para modelar redes gênicas (Imoto et al., 2003; Nariai et al., 2004). Redes bayesianas são empregadas também em problemas de classificação nos mais diferentes contextos, desde reconhecimento de voz até detecção de *spam* (correspondência, geralmente eletrônica, enviada em massa e que não é de interesse para os destinatários) (Androutsopoulos et al., 2000; Stephenson et al., 2000; Vehtari & Lampinen, 2000). Trabalhos que apresentam aplicações de redes bayesianas em diagnósticos médicos e de falhas de computadores podem ser encontrados em Antal et al. (2003) e Horvitz et al. (2001). As redes bayesianas estão sendo empregadas também na área de aprendizado de máquina para seleção de atributos relevantes em problemas de classificação de padrões (Hruschka Jr et al., 2004; Inza et al., 2000).

2.6.1 Métricas para avaliar redes bayesianas

Existem três principais medidas de avaliação, a saber: as métricas bayesianas, a métrica MDL e aquelas baseadas na máxima verossimilhança com termo de penalização.

- **Métricas bayesianas**

As métricas bayesianas medem a qualidade da rede computando a sua verossimilhança com respeito aos dados, permitindo incluir conhecimento a *priori* sobre a estrutura e os parâmetros numéricos. Sua forma geral é:

$$P(B|D) = \frac{P(D|B)P(B)}{P(D)}, \tag{2.21}$$

em que B é a rede bayesiana sendo avaliada e D é o conjunto de dados disponível. Como $P(D)$ é constante para todas as redes, pode ser omitido. Portanto, as métricas maximizam $P(D|B)P(B)$.

Heckerman et al. (1995) propuseram uma métrica que calcula a verossimilhança marginal e usa conhecimento a *priori* que segue uma distribuição de *Dirichlet*. A essa métrica, eles deram o nome de *Bayesian Dirichlet* (BD):

$$P(D|B) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(N'_{ij})!}{(N'_{ij} + N_{ij})!} \prod_{k=1}^{r_i} \frac{(N'_{ijk} + N_{ijk})!}{(N'_{ijk})!}, \tag{2.22}$$

em que D representa o conjunto de dados, B é a rede bayesiana sendo avaliada, n é o número de variáveis, q_i denota o número de possíveis valores que os pais de X_i podem assumir, r_i é o número de possíveis valores de X_i, N_{ijk} é o número de casos em que X_i assume o *k*-ésimo valor com seus pais assumindo o *j*-ésimo valor, e $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Os termos N'_{ij} e N'_{ijk} representam conhecimento a *priori* sobre as estatísticas N_{ij} e N_{ijk}.

A métrica BD foi generalizada a partir da métrica K2, proposta por Cooper & Herskovits (1992), a qual não considera conhecimento a *priori* sobre a rede bayesiana:

$$P(D|B) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2.23}$$

- **Métrica MDL**

A métrica MDL parte do princípio de que qualquer conjunto de dados pode ser representado por uma sequência finita de símbolos de um alfabeto. A ideia fundamental é procurar por modelos que permitam representar os dados usando a menor sequência de símbolos. A métrica MDL é expressa por:

$$log\ P(D|B) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk}\ log \frac{N_{ijk}}{N_{ij}} - \frac{1}{2} logN * dim(B) - DL(B), \tag{2.24}$$

em que N é a quantidade de instâncias no conjunto de dados, $DL(B)$ é o número de bits necessários para representar a estrutura do grafo e $dim(B)$ é a quantidade de parâmetros requiridos para especificar o modelo. Logo,

$$dim(B) = \sum_{i=1}^{n} q_i(r_i - 1). \tag{2.25}$$

- **Máxima verossimilhança com termo de penalização**

Esse tipo de métrica calcula o log da função de verossimilhança junto com uma função para penalizar redes muito complexas. Ela assume um ponto de vista frequencista, uma vez que se baseia na frequência de ocorrência das instâncias e não permite incluir conhecimento a *priori* acerca do processo de seleção da rede bayesiana. Essa métrica é expressa por:

$$log\ P(D|B) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk}\ log \frac{N_{ijk}}{N_{ij}} - \varphi(N) * dim(B). \tag{2.26}$$

O termo $\varphi(N)$ é uma função que visa penalizar modelos complexos. Se $\varphi(N)$=1, então a métrica é chamada de *Akaike's Information Criterion* (AIC). Por outro lado, se $\varphi(N) = 1/2\ log\ N$, recebe o nome de *Bayesian Information Criterion* (BIC).

2.7 Redes Gaussianas

No caso em que X=$\{X_1, X_2,...,X_n\}$ é uma variável aleatória contínua *n*-dimensional, o modelo gráfico probabilístico para X é chamado de rede gaussiana. Ela recebe esse nome porque a probabilidade conjunta para X é representada por uma distribuição gaussiana *n*-dimensional com vetor de médias $\boldsymbol{\mu}$ e matriz de covariância $\boldsymbol{\Sigma}$, da forma:

$$f(x) = f(x_1 x_2 \dots x_n) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-n/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-1/2(x-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})}, \tag{2.27}$$

em que $|\Sigma|$ denota o determinante da matriz de covariância e Σ^{-1} é a inversa da matriz de covariância, também chamada de matriz de precisão e algumas vezes denotada por W.

A decomposição em produtos da distribuição de probabilidade conjunta para X adquire a seguinte forma, quando o modelo utilizado é uma rede gaussiana:

$$f(x) = \prod_{i=1}^{n} f(x_i | \boldsymbol{pa_i}), \tag{2.28}$$

em que

$$f(x_i | \boldsymbol{pa_i}) = \mathcal{N}(\mu_i + \sum_{x_k \in \boldsymbol{pa_i}} b_{ki}(x_k - \mu_k), \sigma_i^2), \tag{2.29}$$

sendo que μ_i denota a média de X_i, σ_i^2 é a variância de X_i, condicionada aos pais de X_i e b_{ki} é o coeficiente de regressão linear refletindo o relacionamento entre as variáveis X_k e X_i. Se $b_{ki} = 0$, então não existe relacionamento entre as variáveis X_k e X_i.

Para tornar evidente a relação entre redes gaussianas e distribuições gaussianas multivariadas, Shachter & Kenley (1989) mostraram a transformação de b_{ki} e σ_i^2 da rede gaussiana em matriz de precisão W da distribuição gaussiana equivalente. A transformação é realizada por meio da seguinte fórmula recursiva:

$$W(i+1) = \begin{pmatrix} W(i) + \frac{b_{i+1} b_{i+1}^t}{\sigma_{i+1}^2} & \frac{-b_{i+1}}{\sigma_{i+1}^2} \\ \frac{-b_{i+1}^t}{\sigma_{i+1}^2} & \frac{1}{\sigma_{i+1}^2}, \end{pmatrix} \tag{2.30}$$

em que $W(i)$ denota a submatriz do canto superior esquerdo, $W(1) = \frac{1}{\sigma_i^2}$, b_i é o vetor coluna

$(b_{1i}, ..., b_{(i-1)i})^t$ e b_i^t é o seu vetor transposto.

2.7.1 Métrica para avaliar redes gaussianas

A métrica mais utilizada para medir a qualidade de redes gaussianas foi derivada da métrica BIC, a qual utiliza o log da verossimilhança junto com uma função para penalizar redes muito complexas:

$$p(D|G) = \left[\prod_{l=1}^{N} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi v_i}} e^{-1/2v_i(x_{li} - \mu_i - \sum_{x_k \in \boldsymbol{pa_i}} b_{ki}(x_{lk} - \mu_k))^2} \right] - f(N) * dim(G), \tag{2.31}$$

em que D representa o conjunto de dados, G é a rede gaussiana sendo avaliada, N denota a quantidade de instâncias, n é o número de variáveis, b_{ki} é o coeficiente de regressão linear refletindo o relacionamento entre X_k e X_i, $\boldsymbol{pa_i}$ é o conjunto de pais da variável X_i e v_i é a variância condicional de X_i dados $X_1, ..., X_{i-1}$ $\forall i, k$. A função $f(N) = \frac{1}{2}\, ln\, N$ é responsável por penalizar modelos complexos e $dim(G) = 2n + \sum_{i=1}^{n} \boldsymbol{pa_i}$ é a quantidade de parâmetros a serem estimados.

2.8 Considerações Finais

Neste capítulo, foram introduzidos alguns conceitos básicos de probabilidade, como probabilidade conjunta, probabilidade condicional, eventos independentes, teorema de Bayes, variável aleatória e distribuições de probabilidade. Em seguida, foi apresentada a definição de modelos gráficos probabilísticos, particularmente as redes bayesianas e as redes gaussianas. Foi dada ênfase ao aprendizado da estrutura dos modelos pelo paradigma de busca e pontuação, em que um mecanismo de busca percorre o espaço contendo todos os grafos acíclicos direcionados enquanto uma métrica avalia a qualidade de cada solução encontrada. Por fim, as métricas mais importantes foram descritas, tanto para as redes bayesianas quanto para as redes gaussianas.

Capítulo 3

Sistemas Imunológicos Artificiais

3.1 Considerações Iniciais

O sistema imunológico natural é composto de um complexo conjunto de células e moléculas que formam um mecanismo de defesa rápido e efetivo contra a invasão de agentes infecciosos no nosso organismo. Sob a perspectiva da computação e engenharia, o sistema imunológico possui várias características importantes, tais como: reconhecimento de padrões, aprendizado, memória, adaptação e auto-organização. Essas características o tornam uma fonte de inspiração bastante interessante para o desenvolvimento de ferramentas computacionais para a solução de problemas do mundo real, denominadas sistemas imunológicos artificiais (SIAs) (Dasgupta, 1999; de Castro & Timmis, 2002b).

Diferentes algoritmos imuno-inspirados têm sido propostos na literatura, dependendo de qual princípio imunológico foi utilizado como inspiração. Neste trabalho, será abordada a classe de algoritmos que se baseiam nos princípios da seleção clonal e teoria da rede. Computacionalmente implementados, esses dois princípios são eficazes na manutenção de diversidade na população, na localização de múltiplas soluções ótimas locais e no ajuste automático do tamanho da população de acordo com a demanda.

Neste capítulo, é realizada uma breve introdução ao sistema imunológico natural visando fundamentar os aspectos biológicos que serviram de inspiração para a criação de um novo paradigma computacional bio-inspirado. Será dada ênfase particular às teorias da seleção clonal e da rede imunológica. Em seguida, são apresentados os principais conceitos de sistemas imunológicos artificiais, suas características e limitações para solução de problemas.

3.2 Sistema Imunológico Natural

Frequentemente, nosso organismo é invadido por diferentes agentes infecciosos (patógenos), capazes de causar danos ao nosso corpo. Quando isso ocorre, o sistema imunológico precisa agir rapidamente para não comprometer a vida do organismo infectado.

O sistema imunológico apresenta uma arquitetura de múltiplas camadas, com mecanismos de defesa espalhados por vários os níveis, como é apresentado na Figura 3.1 (de Castro & Timmis, 2002b).

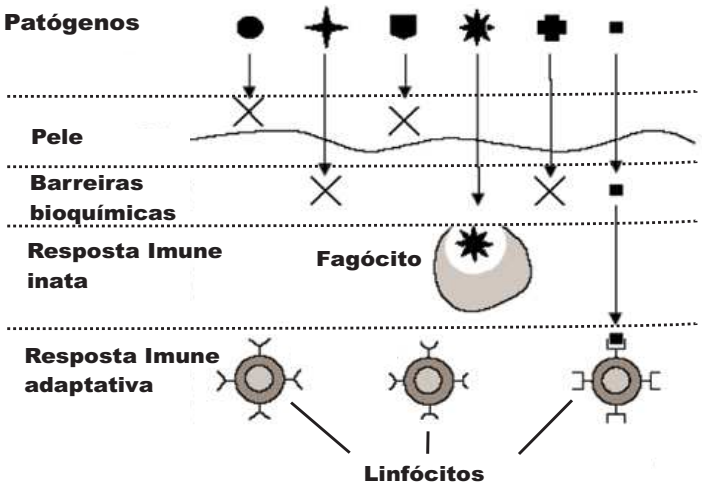


Figura 3.1: Estrutura em camadas do sistema imunológico.

A pele representa uma barreira física e funciona como uma espécie de escudo contra os invasores. As barreiras bioquímicas são os fluidos como a saliva, lágrima e suor. Os ácidos estomacais também ajudam a eliminar grande parte dos microorganismos ingeridos junto com os alimentos. Além disso, o pH e a temperatura do corpo podem apresentar condições de vida desfavoráveis para alguns invasores.

O sistema imunológico inato contém células que estão imediatamente disponíveis para o combate contra uma ampla quantidade de agentes invasores e atuam sempre do mesmo modo. Alguns autores, como Scroferneker & Pohlmann (1998), consideram a pele e as barreiras bioquímicas como parte do sistema imunológico inato. As células mais importantes do sistema imunológico inato são os fagócitos, como os monócitos, macrófagos e neutrófilos. Estes fagócitos possuem receptores que reconhecem padrões moleculares encontrados em microorganismos. Quando este reconhecimento ocorre, os fagócitos englobam (“fagocitam”) os invasores, não importando de qual tipo eles sejam. Além da fagocitose, ocorre também o estímulo para a produção de citocinas. As citocinas tem como objetivos emitir sinais para outras células do sistema imunológico, induzir uma resposta inflamatória e elevar a temperatura do corpo, provocando a chamada febre. Acredita-se que a febre, até certo ponto, é benéfica, pois a atividade de alguns agentes infecciosos é reduzida conforme a temperatura

do corpo aumenta. A resposta imune inata desempenha, portanto, um papel crucial na iniciação e posterior direcionamento das respostas imunes adaptativas, principalmente devido ao fato de que as respostas adaptativas demoram um certo período de tempo para exercer seus efeitos.

Já a resposta imune adaptativa tem como característica principal o reconhecimento de um patógeno específico, tendo como células principais os linfócitos T e B. Os linfócitos T recebem este nome porque são gerados no timo, uma glândula linfática localizada na cavidade torácica. Já os linfócitos B são gerados na medula óssea (*bone marrow*, em inglês) e por isto recebem este nome. Tanto as células B quanto as T possuem moléculas receptoras (reconhecedoras) em suas superfícies capazes de reconhecer os patógenos. As células T têm por finalidade reconhecer um invasor previamente fagocitado pelas células da resposta imunológica inata e estimular a produção de linfócitos B, que, por sua vez, se encarregam de produzir anticorpos específicos para aquele invasor. No decorrer deste processo, são formadas células de memória, que respondem de forma mais rápida e eficiente a uma reinfecção. Embora as células T não produzam anticorpos, elas são muitos importantes para a estimulação das células B.

Enquanto a resposta imune adaptativa resulta na imunidade contra a re-infecção ao mesmo agente infectante, a resposta imune inata não sofre mudanças expressivas ao longo da vida de um indivíduo, independente da exposição a antígenos. Esta é uma importante diferença entre a resposta adaptativa e a resposta inata. Em conjunto, os sistemas inato e adaptativo contribuem para uma defesa notavelmente eficaz, garantindo que, embora passemos nossas vidas cercados por germes potencialmente patogênicos, apresentemos resistência às enfermidades.

Apesar da importância da defesa inata no sistema imunológico natural como um todo, o restante deste capítulo será focado apenas nos mecanismos responsáveis pelo funcionamento da defesa adaptativa, que é a principal fonte de inspiração para os sistemas imunológicos artificiais.

3.2.1 Resposta Imunológica Adaptativa

Uma vez que um patógeno invadiu um organismo, ele pode ser fagocitado por um grupo de células, chamadas macrófagos, que estão em circulação pelo organismo. Cada um destes macrófagos fragmenta o patógeno e passa então a exibir em sua superfície características desse patógeno, os chamados antígenos. Estes macrófagos se deslocam então para os linfonodos, nódulos espalhados pelo organismo com alta concentração de células do sistema imunológico, onde têm os antígenos em sua superfície reconhecidos pelas células T. Essas células T, por sua vez, estimulam a reprodução de células B, as quais se diferenciam em células B de memória e células de plasma, também chamadas de plasmócitos. As células B de memória possuem um período de vida mais longo que as outras e ficam circulando pelo sangue, vasos linfáticos e tecidos. Elas garantem uma resposta mais rápida a patógenos com antígenos similares que possam vir a invadir o organismo no futuro. Já os plasmócitos

são as células responsáveis por secretar anticorpos no organismo em altas taxas. Cada célula B produz um único tipo de anticorpo, propriedade esta conhecida como monoespecificidade. Na Figura 3.2(a) (de Castro & Timmis, 2002b), está apresentada uma célula B com destaque para o tipo de anticorpo que ela é capaz de produzir.

Anticorpos, ou imunoglobinas, 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 esses indivíduos devem ser eliminados. A região do antígeno em que ocorre a ligação com o anticorpo é chamada de epítopo. Vale salientar aqui que, embora um anticorpo possua apenas um tipo de receptor de antígeno, um mesmo antígeno pode apresentar múltiplos epítopos. Isto significa que deferentes anticorpos podem se ligar a um único antígeno (veja Figura 3.2(b)).

Tanto os anticorpos quanto os antígenos possuem composições físico-químicas bem definidas, de forma que, quanto melhor forem as afinidades físico-químicas, melhor será a qualidade da ligação entre eles.

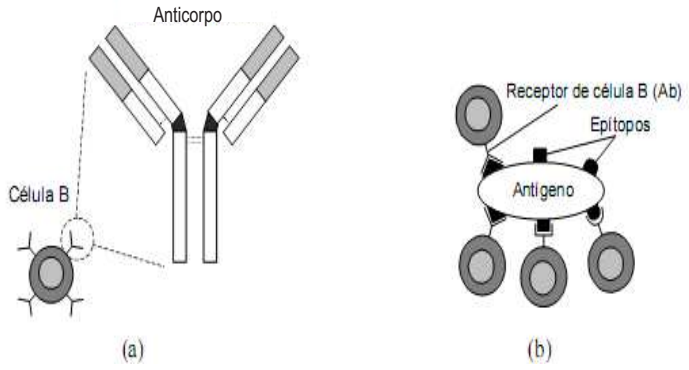


Figura 3.2: Linfócito B. (a) Destaque para o anticorpo em sua superfície. (b) Anticorpos reconhecendo um antígeno.

3.2.2 Princípio da Seleção Clonal

Foi dito anteriormente que as células B e T possuem moléculas receptoras em suas superfícies capazes de reconhecer antígenos. Entretanto, cada célula apresenta um padrão distinto de receptor antigênico, fazendo com que a quantidade de células que podem se ligar a um determinado antígeno seja restrita. A fim de produzir células suficientes para combater a infecção, a célula ativada (aquela que reconheceu o antígeno) deve se proliferar, gerando cópias idênticas (clones) de si mesma.

Durante este processo, para melhorar a adaptação entre os clones gerados e os 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 com o antígeno e as que por ventura se tornaram prejudiciais ao organismo com a etapa de hipermutação são então eliminadas da população.

Este processo é conhecido como Princípio da Seleção Clonal e foi proposto por Burnet (1959). A seleção clonal ocorre tanto no caso dos linfócitos B quanto dos linfócitos T. Maiores detalhes sobre este princípio imunológico podem ser encontrados em (Ada & Nossal, 1987). Na Figura 3.3, é apresentado um esquema ilustrativo do funcionamento da seleção clonal (de Castro & Timmis, 2002b).

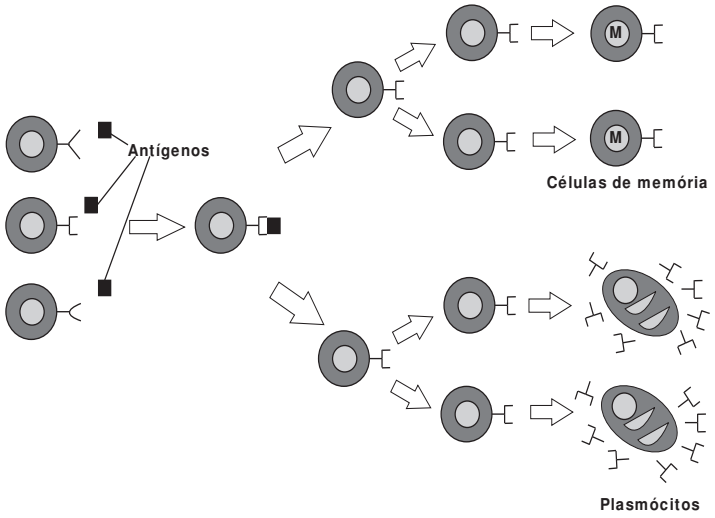


Figura 3.3: Princípio da seleção clonal.

3.2.3 Teoria da Rede Imunológica

Outra teoria importante na imunologia é a teoria da rede, proposta por Jerne (1974). Essa teoria diz que o sistema imunológico não é apenas um sistema reativo que permanece em repouso até que um patógeno invada o organismo. Segundo a teoria da rede, as células imunológicas são capazes de reconhecer antígenos e reconhecer umas às outras. Diante disto, o sistema imunológico pode ser visto como uma enorme e complexa rede, que permanece em um estado de equilíbrio dinâmico em que seus elementos sofrem constantes alterações em suas concentrações. Quando um antígeno altera este equilíbrio, a rede automaticamente altera as concentrações de indivíduos de forma a combater este antígeno e convergir para um novo estado de equilíbrio.

Já que as células do sistema imunológico são capazes também de reconhecer células do próprio organismo, a teoria da rede afirma que os linfócitos poderiam responder positivamente ou negativamente ao sinal de algum reconhecimento. Uma resposta positiva é disparada

no reconhecimento de antígenos, estimulando a secreção de anticorpos para guiar a resposta imunológica. Já a resposta negativa ocorre no reconhecimento de outra célula da rede, levando à supressão da resposta imunológica para evitar o ataque a células do próprio organismo.

Embora a teoria da rede tenha aceitação reduzida na imunologia moderna, seus conceitos têm conduzido a algoritmos eficazes, quando explorados em mecanismos computacionais.

3.3 Sistema Imunológico Artificial

Os sistema imunológico artificiais (SIAs) surgiram como uma ferramenta computacional inspirada no sistema imunológico natural para solução de problemas nas mais diversas áreas, como otimização (Chen & Mahfouf, 2006; Chun et al., 1998; Coelho & Von Zuben, 2006; de França et al., 2005), agrupamento de dados (Castro et al., 2007a,b; de Castro & Von Zuben, 2001), reconhecimento de padrões (Dasgupta, 1999; Watkins et al., 2004), segurança computacional (Aickelin et al., 2004; Kephart, 1994) e aprendizado de máquina (Castro & Von Zuben, 2006; Castro et al., 2005; Hunt & Cooke, 1996). Mais aplicações de SIAs em distintas áreas são abordadas em de Castro & Timmis (2002b).

O interesse pelo desenvolvimento de sistemas computacionais inspirados no sistema imunológico natural se deve às características apresentadas por estes, as quais permitem a obtenção de um algoritmo robusto, escalável e flexível, quando implementadas. A seguir, são listadas algumas dessas características:

- **Reconhecimento de padrões:** as células e moléculas que não pertencem ao organismo são reconhecidas e eliminadas;
- **Detecção de anomalia:** o sistema imunológico pode reconhecer e reagir a agentes infecciosos sem que o organismo nunca tenha tido contato com tais agentes;
- **Detecção imperfeita (tolerância a ruído):** um reconhecimento absoluto do patógeno não é necessário para que o sistema imunológico reaja a ele;
- **Diversidade:** existem vários tipos de elementos (células, moléculas, proteínas, etc.), cada um com sua função, que, juntos, formam a resposta imunológica;
- **Aprendizado e memória:** o sistema imunológico pode aprender a estrutura dos patógenos e se lembrar delas quando o organismo for infectado novamente pelo mesmo patógeno. Esta característica permite que a resposta imunológica seja mais rápida;

- **Descentralização:** as células do sistema imunológico estão espalhadas pelo nosso corpo e, mais importante, não existe um elemento central que as controla;
- **Auto-organização:** quando um patógeno entra em contato com o organismo, não existe informação a respeito de como as células e moléculas devem agir. A seleção clonal e a maturação de afinidade são responsáveis por selecionar as células e expandir as mais aptas a combater o patógeno;
- **Dinâmica:** o sistema imunológico possui um elevado número de elementos, porém finito. Esses elementos não reconhecem todos os possíveis patógenos existentes. Por isso, o sistema imunológico promove mudanças no repertório de células.

Muitos sistemas imunológicos artificiais têm sido propostos na literatura, seguindo diferentes princípios imunológicos como inspiração. Como exposto anteriormente, neste trabalho será abordada a classe de algoritmos que se baseiam nos princípios da seleção clonal e teoria da rede, por apresentarem características interessantes que possibilitam a implementação de algoritmos eficazes na manutenção de diversidade na população, na localização de múltiplas soluções ótimas locais e no ajuste automático do tamanho da população de acordo com a demanda.

Qualquer SIA que se baseia nesses dois princípios possui a estrutura apresentada no Algoritmo 3.1, adaptada de de Castro & Timmis (2002b).

Algoritmo 3.1 Sistema Imunológico Artificial

Início
Iniciar a população de anticorpos aleatoriamente;
Enquanto condição de parada não for satisfeita **faça**
Avaliar a população;
Clonar anticorpos;
Aplicar mutação nos clones;
Eliminar os clones com *fitness* menor que um limiar;
Eliminar anticorpos semelhantes;
Inserir aleatoriamente anticorpos na população;
Fim enquanto
Fim

Primeiramente, é criada aleatoriamente uma população de indivíduos que representam possíveis soluções para o problema, chamados anticorpos. Em seguida, o critério de parada do algoritmo é verificado. Caso ainda não tenha sido alcançado, o *fitness* de cada indivíduo é calculado. Cada anticorpo dará origem a um número de clones. Esse número é proporcional ao *fitness* do anticorpo.

Quanto maior, mais clones ele irá gerar. Cada clone, então, passará por um processo de mutação. A taxa de mutação é inversamente proporcional ao *fitness* do clone. Um novo cálculo do *fitness* é realizado e os clones com valor de *fitness* menor que um determinado limiar são eliminados da população. Por fim, os anticorpos similares, dado um limiar, serão excluídos da população, ficando somente aquele com maior *fitness*. Por fim, novos anticorpos são inseridos aleatoriamente na população, a fim de manter diversidade. O fluxo de execução retorna, então, para a etapa de verificação da condição de parada. Espera-se que, atingido o critério de parada, o algoritmo tenha encontrado boas soluções para o problema. Dentre os principais critérios de parada, cabe destacar: número máximo de gerações e número de gerações em que não ocorre aperfeiçoamento do melhor indivíduo.

Para ilustrar a capacidade dos algoritmos imunológicos de encontrar os múltiplos ótimos locais de um problema, mantendo a diversidade das soluções, um algoritmo imuno-inspirado chamado opt-aiNet (de Castro & Timmis, 2002a), que possui estrutura de funcionamento semelhante à apresentada no Algoritmo 3.1, foi aplicado a três problemas de otimização bidimensional (problemas de maximização) que apresentam multimodalidade. As funções otimizadas foram *Multi* (Equação 3.1), *Roots* (Equação 3.2) e *Schaffer* (Equação 3.3).

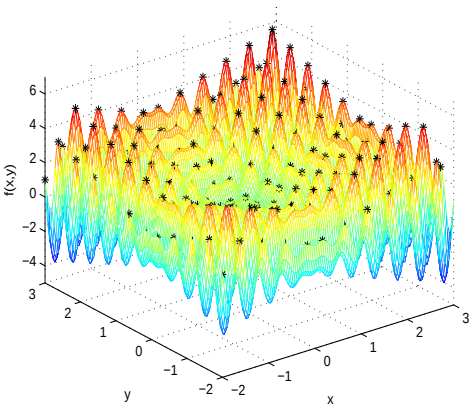
$$f(x,y) = x\text{sen}(4\pi x) - y\text{sen}(4\pi y + \pi), \quad x,y \in [-2,3]; \tag{3.1}$$

$$f(z) = \frac{1}{1+|z^6-1|}, \quad z \in \mathbb{C}, z = x+iy, \quad x,y \in [-2,2]; \tag{3.2}$$

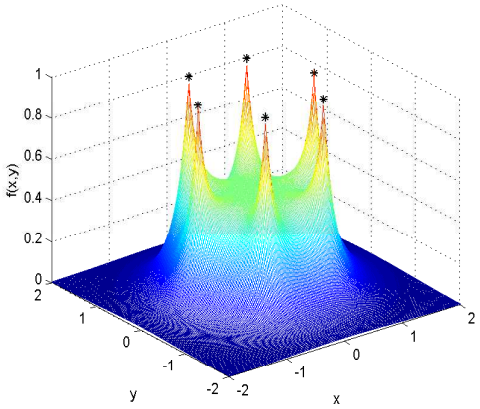
$$f(x,y) = 0,5 + \frac{\text{sen}^2(\sqrt{x^2+y^2}) - 0,5}{1+0,001(x^2+y^2)}, \quad x,y \in [-10,10]. \tag{3.3}$$

Os resultados podem ser observados na Figura 3.4. Como pode ser visto, o algoritmo foi capaz de gerar um conjunto de soluções boas e, ao mesmo tempo, manter a diversidade destas soluções. Para as funções *Multi* e *Roots*, todos os ótimos locais foram encontrados. Já para a função *Schaffer*, a qual possui um único ótimo global e infinitos ótimos locais, o algoritmo foi capaz de encontrar esse ótimo global e vários outros locais.

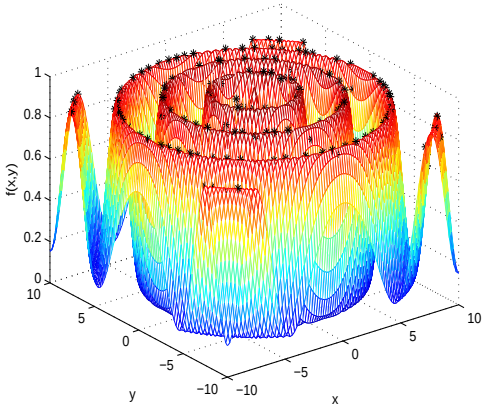
O tamanho da população varia automaticamente ao longo do processo de otimização e converge



(a) função *Multi*



(b) função *Roots*

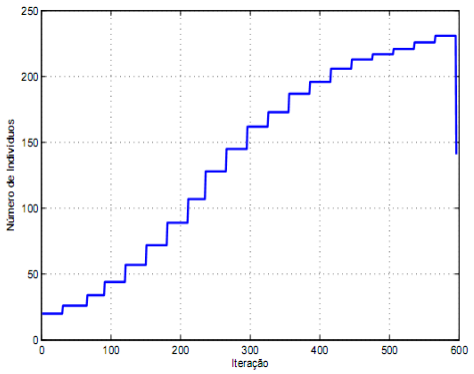


(c) função *Schaffer*

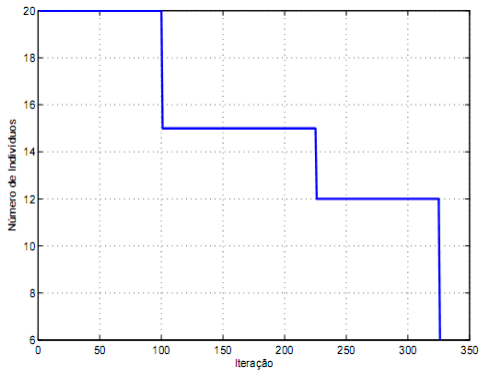
Figura 3.4: Resultado da aplicação de SIAs em otimização multimodal. Os asteriscos (*) representam os indivíduos da população (anticorpos) ao final da execução.

para valores que dependem da demanda de cada problema. Para os três casos, a população inicial foi criada com 20 indivíduos. É interessante notar que, no final de sua execução, o algoritmo opt-aiNet elimina aqueles indivíduos que não convergiram para algum ótimo local ou global do problema, o que provoca a redução brusca no número de indivíduos, como ilustrado na Figura 3.5.

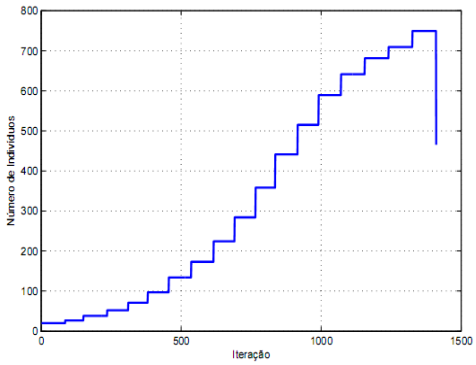
É evidente que para o algoritmo apresentar um bom desempenho, dentre outros fatores, o operador de mutação precisa ser escolhido com cuidado, bem como a taxa de mutação. Esses parâmetros, se mal definidos, podem levar à geração de soluções ruins. Outro inconveniente em sistemas imunológicos artificiais, em relação ao operador de mutação, é o fato dele não considerar a interação dos atributos presentes no anticorpo na hora de desempenhar sua função.



(a) função *Multi*



(b) *Roots*



(c) *Schaffer*

Figura 3.5: Tamanho da população ao longo das iterações, para as três funções.

3.4 Sistemas Imunológicos Artificiais e Blocos Construtivos

Apesar dos algoritmos imunológicos artificiais apresentarem bom desempenho para uma vasta gama de problemas, alguns aspectos podem ser melhorados. O sistema imunológico artificial, assim como diversos outros algoritmos de busca e otimização, representa as soluções candidatas para um determinado problema como um vetor de atributos. É sabido que muitos problemas de otimização podem ser decompostos em sub-problemas, os quais são resolvidos separadamente e suas soluções parciais serão combinadas posteriormente para compor a solução global. Nesse contexto, é possível que o vetor de atributos associado a cada solução candidata apresente soluções parciais para o problema.

Por exemplo, considere um problema binário de otimização combinatória consistindo de dez variáveis. Na Figura 3.6 são apresentados cinco vetores que supostamente representam cinco soluções de boa qualidade para este problema fictício. Suponha que a linha pontilhada indica a existência de uma solução parcial para o problemas, dizendo que as posições 3 e 4 dos vetores devem conter o valor “1” e a posição 5 deve conter o valor “0”. Já a linha tracejada denota uma segunda solução

parcial, dizendo que as posições 7 e 10 dos vetores devem conter o valor “0” enquanto a posição 9 deve possuir o valor “1”.

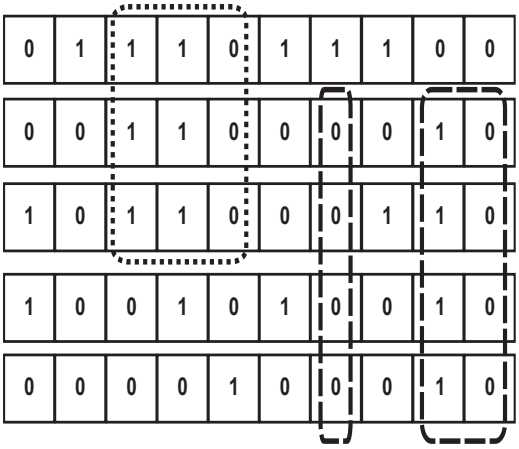


Figura 3.6: Exemplo de blocos construtivos.

A essas soluções parciais dá-se o nome de blocos construtivos (do inglês *building blocks*). Os blocos construtivos podem ser vistos, então, como pequenos conjuntos de posições do vetor de soluções que se relacionam e, juntos, representam sub-soluções para o problema. Esses blocos construtivos contribuem de forma significativa para garantir a qualidade das boas soluções e devem ser sempre preservados.

Entretanto, os algoritmos imunológicos artificiais existentes na literatura são incapazes de determinar diretamente quais elementos do vetor se relacionam para formar um bloco construtivo. Mesmo que ele seja descoberto, o operador de mutação pode facilmente rompê-lo, pois ele não considera a interação dos caracteres presentes no anticorpo na hora de desempenhar sua função.

3.5 Considerações Finais

Neste capítulo, foi realizada uma breve descrição do sistema imunológico natural, junto com as duas principais teorias que servem de inspiração para o desenvolvimento de sistemas computacionais que atuam como meta-heurísticas para a solução de problemas nas mais diversas áreas. Em seguida, foram apresentados o conceito dos sistemas imunológicos artificiais e suas características mais relevantes, como a capacidade de manter diversidade na população e o controle automático do tamanho da população em resposta às particularidades do problema que está sendo tratado. Essas características conferem aos sistemas imunológicos artificiais a capacidade de solucionar problemas de otimização em que o espaço de busca é amplo e multimodal, encontrando os ótimos globais e locais.

Foi apresentada também uma limitação dos sistemas imunológicos artificiais: a ausência de mecanismos para considerar as interações das variáveis do problema no momento de resolvê-lo.

No Capítulo 5, será apresentada uma abordagem promissora que confere aos algoritmos bio-inspirados a capacidade de lidar eficientemente com blocos construtivos. Em seguida, serão propostos e descritos quatro sistemas imunológicos artificiais que utilizam essa abordagem para resolver problemas de otimização no espaço discreto e contínuo.

Capítulo 4

Aprendizado em Redes Bayesianas por meio de Sistemas Imunológicos Artificiais

4.1 Considerações Iniciais

Conforme detalhado no Capítulo 2, o aprendizado de redes bayesianas a partir de conjuntos de dados tem recebido grande atenção nos últimos anos. Dentre os paradigmas de aprendizado, o mais utilizado é aquele baseado na busca e pontuação, em que um algoritmo de busca é responsável por explorar o espaço contendo todos os grafos acíclicos direcionados, enquanto uma métrica de pontuação avalia cada estrutura, em termos de quão bem ela representa os dados.

De acordo com Buntine (1996), os mecanismos de busca mais indicados para esta tarefa devem ser baseados em população de soluções e serem capazes de realizar busca multimodal. Um forte candidato a essa tarefa é um sistema imunológico artificial, uma vez que é baseado em população e é capaz de controlar o tamanho e a diversidade da população. Com este eficiente mecanismo de exploração do espaço de busca, um sistema imunológico pode encontrar e preservar as múltiplas soluções de boa qualidade num problema multimodal.

Neste capítulo, é apresentada uma contribuição específica da tese, que é a investigação de uma nova metodologia baseada em sistemas imunológicos artificiais destinada ao aprendizado de redes bayesianas (Castro & Von Zuben, 2005). Na Seção 4.2, são apresentados os componentes do algoritmo, como codificação, função de avaliação, operadores de clonagem e mutação. Os resultados obtidos junto a casos de estudo comumente adotados na literatura são então descritos e analisados. Na Seção 4.3, são descritos experimentos de aprendizado de redes bayesianas agora voltados para a tarefa de seleção de atributos, por meio da identificação do *Markov blanket* do nó que denota a variável de saída.

Embora experimentos com seleção de atributos sejam apresentados neste capítulo, os conceitos

associados ao assunto não são abordados aqui, mas sim no Capítulo 7, por ser exclusivo sobre seleção de atributos.

4.2 Aprendizado de Redes Bayesianas por meio de um Sistema Imunológico Artificial

Nesta seção, é descrita a aplicação de um sistema imunológico artificial baseado nas teorias da seleção clonal e da rede imunológica, para o aprendizado de redes bayesianas. A partir de um conjunto de dados, o algoritmo tenta encontrar um rede bayesiana que seja capaz de representá-los de forma apropriada.

A seguir, os componentes do algoritmo serão detalhados.

4.2.1 Codificação

Na implementação do algoritmo, cada rede candidata é um anticorpo enquanto o conjunto de dados representa os antígenos. Para um domínio específico com *n* variáveis, as soluções candidatas são codificadas como uma matriz binária $n \times n$, em que os elementos da matriz, c_{ij}, i,j=1,...,n, podem assumir os valores:

$$c_{ij} = \begin{cases} 1 & \text{se o nó } j \text{ é pai do nó } i; \\ 0 & \text{caso contrário.} \end{cases} \tag{4.1}$$

Por exemplo, considere a rede bayesiana apresentada na Figura 4.1(a). O anticorpo que representa tal rede é codificado conforme mostra a matriz da Figura 4.1(b).

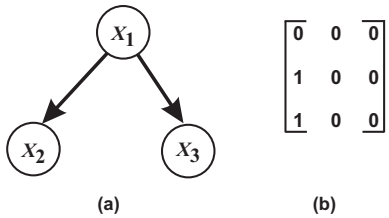


Figura 4.1: Codificação de uma possível solução candidata.

A população inicial é gerada aleatoriamente. Ela é formada por matrizes cujos elementos podem assumir valores 0 ou 1, com o cuidado de não gerar ciclos.

4.2.2 Função de aptidão

A função de aptidão utilizada para avaliar a qualidade de uma rede candidata B, dado o conjunto de dados D, é a pontuação BIC (*Bayesian Information Criterion*), expressa na Equação (4.2):

$$\log P(D|B) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk}\, log \frac{N_{ijk}}{N_{ij}} - \frac{1}{2} \log N * dim(B), \tag{4.2}$$

sendo

$$dim(B) = \sum_{i=1}^{n} q_i (r_i - 1) \tag{4.3}$$

em que n é o número de variáveis, q_i denota o número de possíveis instâncias pais de x_i, r_i é o número de possíveis valores de x_i, N_{ijk} é o número de casos em que x_i assume o *k*-ésimo valor com seus pais assumindo o *j*-ésimo valor, e $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. O termo $dim(B)$ é a dimensão da rede, isto é, a quantidade de parâmetros necessários para especificar o modelo.

4.2.3 Clonagem e Mutação

Durante a etapa de clonagem, cópias idênticas de todos os anticorpos são criadas. Em seguida, cada clone passa por um processo de mutação, sendo que a taxa de mutação para cada clone é inversamente proporcional ao seu valor de aptidão. A taxa de mutação é calculada pela Equação (4.4).

$$P_{mut}(Ab_i) = Max\_P_{mut} * \frac{(Max\_Fit - Fit(Ab_i))}{(Max\_Fit - Min\_Fit)} \tag{4.4}$$

em que $P_{mut}(Ab_i)$ é a taxa de mutação para o clone *i*; Max_P_{mut} é o maior valor que a taxa de mutação pode alcançar; $Fit(Ab_i)$ é o valor de aptidão do clone i; Max_Fit e Min_Fit são o maior e menor valor de aptidão encontrado na população corrente, respectivamente.

O operador de mutação é responsável por adicionar, remover e inverter a ordem dos arcos da rede bayesiana codificada no anticorpo. Isso é feito alterando o bit de “0” para “1” e vice-versa. Durante a mutação, assim como na etapa de criação da população inicial, é preciso evitar a criação de grafos cíclicos. Para cada ciclo identificado, uma aresta escolhida aleatoriamente é removida ou invertida. Aqui, um processo de busca local poderia ser executado para definir qual conexão deveria ser removida. Entretanto, a exclusão aleatória foi adotada neste trabalho por questão de simplicidade.

4.2.4 Supressão

Nesta fase, anticorpos idênticos são eliminados da população para evitar redundância e, assim, manter a diversidade na população. Se dois ou mais anticorpos apresentam a mesma estrutura, isso é, matrizes idênticas, todos com exceção de um são excluídos da população. Soluções candidatas que apresentam valor de aptidão muito baixo também são removidas.

4.2.5 Resultados

Inicialmente, o algoritmo proposto foi testado em dois *benchmarks* muito conhecidos na literatura: ALARM (Beinlich et al., 1989) e ASIA (Lauritzen & Spiegelhalter, 1988). O primeiro refere-se a uma rede que contém 37 nós e 46 arestas. A rede referente ao *benchmark* ASIA contém 8 nós e 8 arestas. Na Figura 4.2, são apresentadas essas duas redes.

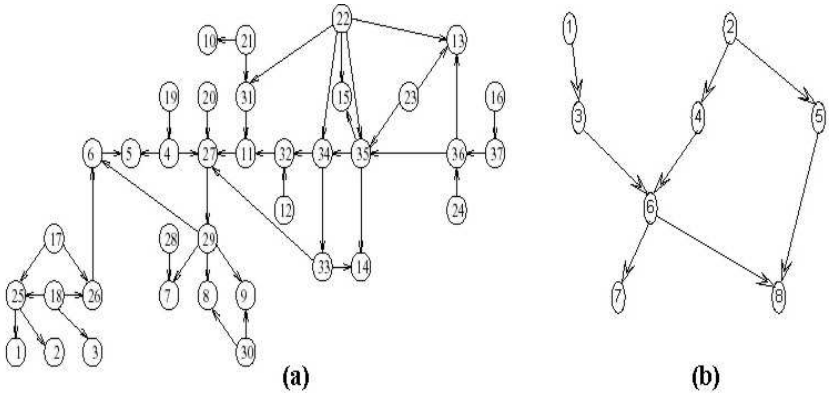


Figura 4.2: *Benchmarks* considerados durante os experimentos: (a) ALARM e (b) ASIA.

Os experimentos consistiram em amostrar conjuntos de dados a partir das redes originais e, a partir destes dados, tentar recuperar a rede original usando o algoritmo imunológico proposto. Para ambos os problemas, os conjuntos de dados possuíam 1000 amostras. Esse número foi obtido empiricamente a partir de experimentos preliminares que visavam verificar qual o tamanho ideal do conjunto de dados. Quando utilizaram-se poucos dados, em torno de 100, o algoritmo não apresentou bom desempenho, gerando redes de má qualidade.

A população inicial do sistema imunológico artificial contém 50 anticorpos, o número de clones por anticorpo é 3 e a taxa máxima de mutação é 0,4. O critério de parada é o número máximo de gerações, definido como 100.

Para cada *benchmark*, foram realizadas 10 execuções do algoritmo. Os resultados foram comparados com aqueles obtidos por dois outros algoritmos: o K2 (Cooper & Herskovits, 1992) e um algoritmo genético, ambos utilizando também a métrica BIC. Para o algoritmo genético, o tamanho da população foi definido como 800 e o número máximo de iterações foi 500.

Na Tabela 4.1, são apresentados os resultados médios obtidos em 10 execuções. Para as redes originais, o valor de BIC é -76304 e -6074, para ALARM e ASIA, respectivamente.

Tabela 4.1: Resultados comparativos entre os 3 algoritmos, para cada *benchmark*.

Benchmark	SIA	AG	K2
ALARM	**-77720,6**	-76812,4	-75306,5
ASIA	-5618,8	**-5803,3**	-5584,0

Pela Tabela 4.1, pode-se observar que os três algoritmos obtiveram desempenhos semelhantes, em termos da métrica BIC.

Dado que a estrutura das redes originais é conhecida, a estrutura das redes obtidas foram comparadas com a estrutura das redes obtidas com a rede original, para cada *benchmark*. Na Tabela 4.2, são apresentados o número total de arcos (TA), o número de arcos extras (AE), o número de arcos ausentes (AA) e o número de arcos invertidos (AI), para cada algoritmo em cada *benchmark*. Esses números representam uma média em 10 execuções.

Tabela 4.2: Comparação da estrutura das redes obtidas pelos algoritmos com a rede original, para cada *benchmark*.

	ALARM				ASIA			
Algoritmo	TA	AE	AA	AI	TA	AE	AA	AI
SIA	**47**	**2**	**1**	**2**	**8**	**0**	**0**	**1**
AG	44	2	4	4	8	0	0	2
K2	52	7	1	3	9	1	0	1

Pode-se observar pela Tabela 4.2 que as redes encontradas pelo sistema imunológico artificial possuem poucos erros, quando comparadas com as redes originais. Para o *benchmark* ALARM, os arcos extras foram $37 \rightarrow 24$ e $23 \rightarrow 33$, o único arco ausente foi aquele ligando os nós 12 e 32 ($12 \rightarrow 32$). De acordo com Cooper & Herskovits (1992), este relacionamento é realmente difícil de ser obtido a partir dos dados. Os arcos invertidos foram $33 \rightarrow 34$ e $2 \rightarrow 25$.

Conforme esperado, numa única execução o algoritmo gerou várias soluções boas, como mostrado na Figura 4.3 para o *benchmark* ASIA. Estas foram as três melhores redes numa execução do algoritmo. Apesar delas divergirem quanto à orientação e existência de alguns arcos, os valores de suas verossimilhanças são muito semelhantes e próximos ao valor da verossimilhança da rede

original. Isso indica que as soluções obtidas são eficazes na representação/explicação do conjunto de dados.

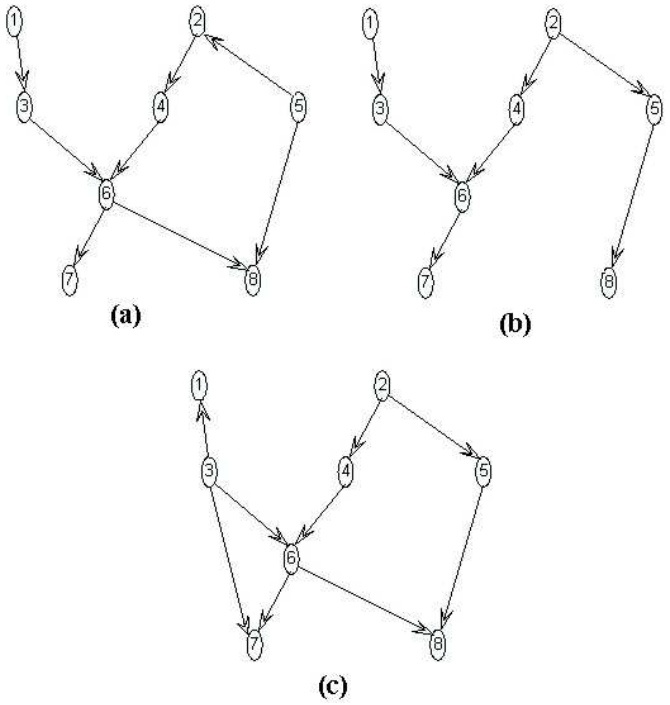


Figura 4.3: Três redes geradas numa única execução para o *benchmark* ASIA.

Uma vez que é difícil escolher apenas uma rede como solução final, pode-se combinar as diversas estruturas de rede numa única estrutura que represente todas elas da melhor forma. Para fazer isso, algum método de consenso de árvore (McMorris & Neumann, 1983) pode ser adaptado e aplicado às redes bayesianas.

A vantagem dessa combinação é clara: as chances de recuperar a rede original são grandes. Como exemplo, se for utilizado um esquema de voto majoritário nas redes da Figura 4.3, uma rede bayesiana de consenso seria criada levando-se em conta a presença e a orientação dos arcos que aparecem na maioria das redes. A estrutura da rede bayesiana de consenso é apresentada na Figura 4.4.

É importante ressaltar aqui que a rede obtida do consenso das três soluções apresentadas na Figura 4.3 é idêntica à rede ASIA original. É evidente que as redes das quais sairá a rede de consenso deverão ser de boa qualidade e que a determinação de quais e quantas redes serão escolhidas para gerar a rede de consenso exerce um papel fundamental nesta tarefa.

Alguns trabalhos já realizaram a combinação de diversas redes numa só usando uma abordagem diferente. Primeiro, é feita a combinação das distribuições de probabilidades das diversas redes e, então, sobre esta distribuição é construída a estrutura da rede de consenso (del Sagrado & Moral, 2003; Pennock & Wellman, 1999). A desvantagem dessa metodologia é o custo computacional

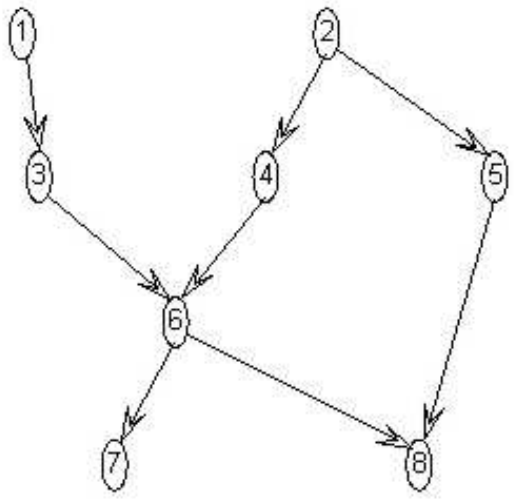


Figura 4.4: Rede bayesiana de consenso.

associado com os diversos cálculos para agregar as distribuições de probabilidades, além da rede resultante não necessariamente apresentar uma tendência de se aproximar da rede original. E como já ocorria com os dados originais, nada garante que a busca por uma rede bayesiana a partir dos dados agregados resulte em apenas uma rede bayesiana de boa qualidade, recriando, assim, a falta de consenso.

4.3 Gerando Redes Bayesianas para Efetuar Seleção de Atributos

Dentre as várias abordagens para seleção dos atributos relevantes para representar um conceito, uma com expressivo destaque na literatura consiste em gerar uma rede bayesiana a partir do conjunto de dados e identificar o *Markov blanket* do nó que representa o conceito. Conforme explicado no Capítulo 2, para um determinado nó A da rede bayesiana, existe um conjunto único de nós que exercem influência na distribuição de probabilidades desse nó A. Para esse conjunto de nós, dá-se o nome de *Markov blanket* do nó A.

Por exemplo, para tarefas de classificação de padrões, é possível utilizar o *Markov blanket* do nó que representa a variável de saída (classe) como critério para seleção de atributos. Isso significa que todo o conhecimento necessário para predizer a classe das instâncias encontra-se exclusivamente no *Markov blanket* do nó que representa a classe, e os nós que não fazem parte do seu *Markov blanket* não alteram sua distribuição de probabilidade.

Nesta seção, são descritos os experimentos realizados com seleção de atributos usando redes bayesianas, em problemas de classificação de padrões. Primeiramente, o sistema imunológico artificial é executado para gerar a rede bayesiana que melhor representa o conjunto de dados. Em seguida, o classificador naïve Bayes (Theodoridis & Koutroumbas, 2006) é treinado usando somente os atributos apontados pelo *Markov blanket* do nó que representa a classe. Os atributos contínuos foram discretizados usando um procedimento proposto por Dougherty et al. (1995), o qual aplica a métrica Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*). Esse pré-processamento foi realizado para que o classificador utilizado (naïve Bayes) pudesse manipular os dados de forma mais eficiente. Embora existam versões do naïve Bayes para lidar tanto com atributos discretos como contínuos, melhores desempenhos são obtidos quando os dados são discretizados (Dougherty et al., 1995).

Os principais objetivos dos experimentos são: (*i*) analisar a escalabilidade do algoritmo imuno-inspirado quando aplicado a conjunto de dados volumosos, e (*ii*) comparar os resultados obtidos com aqueles produzidos por outros algoritmos de seleção de atributos.

Por meio de experimentos preliminares, os parâmetros do algoritmo foram ajustados como segue. A população inicial contém 10 anticorpos, o número de clones por anticorpo é 3, a taxa máxima de mutação é 0,4 e o critério de parada é o número máximo de iterações, definido como 500.

Nas duas subseções seguintes, são descritos os conjuntos de dados empregados e os resultados obtidos, respectivamente.

4.3.1 Descrição dos conjuntos de dados

Dez conjuntos de dados obtidos do *UCI Repository of Machine Learning Databases* (Asunción & Newman, 2007) foram considerados para este experimento: Wisconsin, Bupa, Ionosphere, Pima, Wine, Chess, Mushroom, Sonar, Card e Vote. Estes conjuntos de dados foram adotados porque são amplamente utilizados na literatura e, assim, é possível comparar o algoritmo proposto para seleção de atributos com os já existentes. Na Tabela 4.3, são apresentadas as características de cada um, com número total de instâncias, número de atributos e de classes.

Os conjuntos de dados foram testados utilizando a validação cruzada de 10 pastas. O conjunto de dados é dividido em 10 subconjuntos disjuntos de igual tamanho, sendo 9 subconjuntos para treinamento e 1 para teste. Esse procedimento é realizado dez vezes, permutando-se todos os subconjuntos. A capacidade de generalização do classificador é estimada tomando-se a média da taxa de classificação correta sobre os 10 subconjuntos de teste.

Tabela 4.3: Características dos conjuntos de dados.

Conjunto de dados	# Instâncias	# Atributos	# Classes
Wisconsin	683	9	2
Bupa	345	6	2
Ionosphere	350	34	2
Pima	768	8	2
Wine	178	13	3
Chess	3196	36	2
Mushroom	8124	22	2
Sonar	208	60	2
Card	690	15	2
Vote	435	16	2

4.3.2 Resultados

Nesta seção, são apresentados os resultados obtidos, bem como é realizada uma comparação com dois algoritmos da literatura: RELIEF-E (Kononenko, 1994) e K2χ^2 (Hruschka Jr et al., 2004).

O RELIEF-E é uma extensão do RELIEF para permitir sua execução junto a problemas de classificação que apresentam mais de duas classes. Ele segue a abordagem filtro e atribui um peso para cada atributo, refletindo a habilidade desse atributo para discriminar as classes. O critério de seleção escolhe aqueles atributos que possuem o peso superior a um limiar definido pelo usuário. O segundo algoritmo também seleciona os atributos usando o conceito de *Markov blanket*. A rede bayesiana é construída usando o algoritmo K2 (Cooper & Herskovits, 1992) e o teste do χ^2. Para todos os três algoritmos, os parâmetros foram ajustados visando obter o menor número possível de atributos, sem degradar o desempenho do classificador.

Na Tabela 4.4, são apresentados os resultados médios para os 10 conjuntos de dados, quando foram usados todos os atributos e quando foram utilizados apenas os selecionados pelo algoritmo imunológico. O número entre parênteses indica a quantidade média de atributos que foi selecionada. Para verificar se a diferença nos resultados possui significado estatístico, o *teste-t* foi aplicado com nível de significância de 95%. Se o *p*-valor for inferior a 0,05 então a diferença nos resultados é estatisticamente significativa. Da mesma forma, nas Tabelas 4.5 e 4.6, são apresentados os resultados comparativos entre o sistema imunológico e os algoritmos RELIEF-E e K2χ^2, respectivamente.

Observando as Tabelas 4.4, 4.5 e 4.6, pode-se perceber que o desempenho dos classificadores melhorou de forma significativa quando empregaram apenas os atributos relevantes, para a maioria dos conjuntos de dados.

O número de atributos selecionados pelo algoritmo proposto e pelo K2χ^2 é similar. Entretanto,

Tabela 4.4: Resultados comparativos usando todos os atributos e aqueles selecionados pelo SIA.

Conjunto de dados	Todos os atributos	Selecionados pelo SIA	*p*-valor
Wisconsin	96,6 %	98,1 % (6,9)	0,000
Bupa	71,6 %	74,9 % (3,4)	0,000
Ionosphere	84,8 %	87,7 % (18,6)	0,000
Pima	73,7 %	74,6 % (4,9)	0,410
Wine	96,3 %	97,8 % (6,7)	0,000
Chess	94,5 %	98,4 % (11,8)	0,000
Mushroom	94,7 %	96,2 % (14,7)	0,000
Sonar	71,8 %	73,4 %(23,2)	0,000
Card	79,2 %	83,1 % (13,7)	0,000
Vote	91,3 %	93,6 % (10,7)	0,000

Tabela 4.5: Resultados comparativos entre SIA e RELIEF-E.

Conjunto de dados	SIA	RELIEF-E	*p*-valor
Wisconsin	98,1 % (6,9)	95,3 % (7,6)	0,000
Bupa	74,9 % (3,4)	70,8 % (1,9)	0,000
Ionosphere	87,7 % (18,6)	85,2 % (15,3)	0,036
Pima	74,6 % (4,9)	73,2 % (4,8)	0,064
Wine	97,8 % (6,7)	96,6 % (5,4)	0,114
Chess	98,4 % (11,8)	98,5 % (19,2)	0,827
Mushroom	96,2 % (14,7)	94,9 % (16,1)	0,000
Sonar	73,4 %(23,2)	72,7 % (31,2)	0,058
Card	83,1 % (13,7)	80,1 % (9,1)	0,000
Vote	93,6 % (10,7)	90,7 % (10,3)	0,000

o classificador que usou os atributos selecionados pelo algoritmo imuno-inspirado apresentou desempenho ligeiramente superior para a maioria dos conjuntos de dados. Isso se deve ao eficiente mecanismo de exploração do espaço de busca do SIA. Já o K2χ^2 está sujeito a ótimos locais, uma vez que ele utiliza um mecanismo de busca não-populacional.

Na Tabela 4.7, são apresentados os melhores conjuntos de atributos selecionados pelo algoritmo imunológico ao longo de 10 execuções.

Tabela 4.6: Resultados comparativos entre SIA e K2χ^2.

Conjunto de dados	SIA	K2χ^2	*p*-valor
Wisconsin	98,1 % (6,9)	97,2 % (6,6)	0,030
Bupa	74,9 % (3,4)	73,6 % (3,9)	0,022
Ionosphere	87,7 % (18,6)	85,4 % (17,8)	0,000
Pima	74,6 % (4,9)	73,2 % (4,8)	0,041
Wine	97,8 % (6,7)	97,1 % (7,9)	0,060
Chess	98,4 % (11,8)	97,5 % (14,6)	0,230
Mushroom	96,2 % (14,7)	98,7 % (17,3)	0,000
Sonar	73,4 %(23,2)	78,1 % (19,1)	0,000
Card	83,1 % (13,7)	83,7 % (10,7)	0,977
Vote	93,6 % (10,7)	93,1 % (11,5)	0,864

Tabela 4.7: Melhor conjunto de atributos encontrado ao longo das 10 execuções do algoritmo.

Conjunto de dados	Atributos selecionados
Wisconsin	2, 3, 5, 7, 8, 9
Bupa	1, 3, 4
Ionosphere	1, 3, 4, 6, 8, 16, 18, 33
Pima	2, 6, 7
Wine	1, 3, 8, 9, 12, 13
Chess	3,10,15,16,21,32,33
Mushroom	3, 5, 8, 9, 12, 16, 22
Sonar	5, 9, 10, 11, 28, 35, 44, 45, 46, 48, 54
Card	2, 4, 6, 8, 9, 10, 14, 15
Vote	7, 10, 12, 13, 15, 16

4.3.3 Redes de consenso

Sob a perspectiva do processo de busca, é interessante notar que o algoritmo imuno-inspirado foi capaz de gerar não somente uma, mas várias soluções de boa qualidade em uma só execução. Na Figura 4.5, são apresentadas três redes bayesianas alternativas obtidas para o conjunto de dados Wisconsin. Embora as redes possuam algumas poucas divergências com relação à estrutura, todas elas são capazes de representar os dados muito bem e possuem valores de verossimilhança parecidos. O nó com o rótulo “C” é a classe e seu *Markov blanket* é representado pelos nós em cinza.

Usando o naïve Bayes com os três conjuntos de dados selecionados pelas redes das Figura 4.5, foram obtidas as seguintes taxas de classificação correta: 98,1% usando os atributos da Figura 4.5(a),

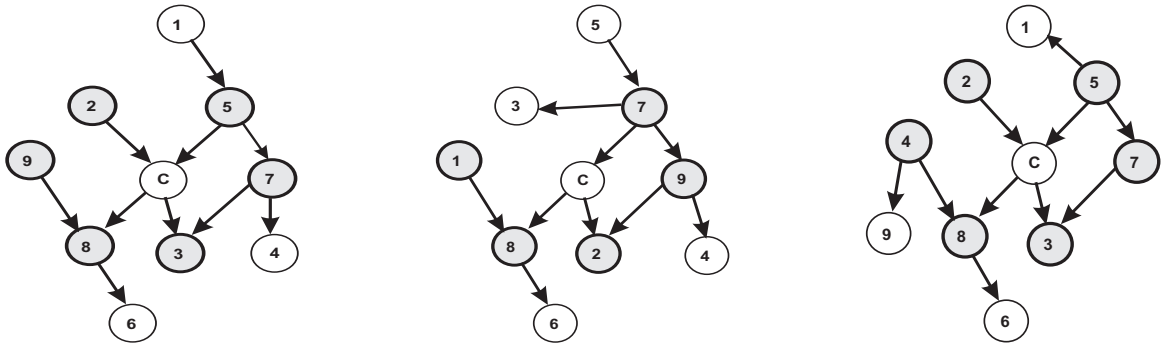


Figura 4.5: Três redes bayesianas alternativas para o conjunto de dados Wisconsin.

97,4% com os atributos da Figura 4.5(b), e 97,2% usando os atributos da Figura 4.5(c).

Novamente, pode-se tomar como consenso o voto majoritário, o que neste caso devolveria a própria seleção de atributos realizada pela rede bayesiana da Figura 4.5(a).

4.4 Considerações Finais

Neste capítulo, foi descrita uma contribuição específica na tese, que consiste na investigação de uma nova metodologia para o aprendizado de redes bayesianas por meio de sistemas imunológicos artificiais. Durante os experimentos preliminares conduzidos nos *benchmarks* ASIA e ALARM, resultados promissores foram obtidos, indicando a viabilidade da metodologia.

Em seguida, a tarefa de seleção de atributos em problemas de classificação de padrões foi tema dos experimentos. O sistema imunológico artificial proposto foi aplicado aos conjuntos de dados, a fim de gerar uma rede bayesiana. Subsequentemente, os nós que compõem o *Markov blanket* do nó que representa a classe foram tomados como os atributos selecionados. Os resultados mostraram que subconjuntos reduzidos de atributos foram obtidos, sem degradar o desempenho do classificador e, até mesmo, melhorando a taxa de classificação.

Capítulo 5

Sistemas Imunológicos Artificiais Baseados em Modelos Gráficos Probabilísticos

5.1 Considerações Iniciais

Conforme exposto no capítulo 3, os sistemas imunológicos artificiais existentes na literatura não possuem a capacidade de lidar eficientemente com blocos construtivos. Dado que o processo de maturação de afinidade requer clonagem dos indivíduos seguida da mutação das novas células geradas, o operador de mutação não consegue identificar os relacionamentos entre as variáveis do problema, levando ao rompimento dos blocos construtivos que porventura foram encontrados.

Uma alternativa bastante promissora, e que será explorada nesse capítulo, é deixar que o próprio algoritmo aprenda as relações existentes entre as variáveis usando informações estatísticas extraídas a partir do conjunto das melhores soluções encontradas até então. O sistema imunológico artificial vai ser capaz, assim, de estimar a distribuição de probabilidades das melhores soluções e, posteriormente, utilizá-la para gerar novas soluções candidatas que apresentem características similares àquelas contidas no conjunto de melhores soluções.

Neste capítulo, são propostos e apresentados quatro novos sistemas imunológicos artificiais que, em vez de evoluírem as soluções candidatas por meio de clonagem e mutação, utilizam um modelo probabilístico que representa a distribuição de probabilidade das melhores soluções encontradas até então. O modelo probabilístico exerce grande influência no comportamento do algoritmo, de modo que sua escolha precisa ser realizada adequadamente. Nos algoritmos propostos neste capítulo, os modelos probabilísticos adotados foram as redes bayesianas (caso discreto) e as redes gaussianas (caso contínuo), por apresentarem a capacidade de representar adequadamente interações complexas entre as variáveis do problema.

Na Seção 5.2, é descrito em detalhes o funcionamento dessa abordagem para manipulação

de blocos construtivos. Em seguida, na Seção 5.3, é apresentado o *Bayesian Artificial Immune System* (BAIS), um sistema imuno-inspirado para otimização mono-objetivo no espaço discreto, que utiliza rede bayesiana como modelo probabilístico para representar o relacionamento entre as variáveis. Posteriormente, foi desenvolvida uma versão do BAIS para otimização no espaço contínuo, denominada *Gaussian Artificial Immune System* (GAIS) por utilizar rede gaussiana, que será descrita na Seção 5.4.

Extensões dos dois algoritmos previamente citados foram desenvolvidas para lidar com problemas de otimização multiobjetivo. Uma breve introdução à otimização multiobjetivo é apresentada na Seção 5.5. Nas Seções 5.6 e 5.7 são descritos o *Multi-objective Bayesian Artificial Immune System* (MOBAIS) e o *Multi-objective Gaussian Artificial Immune System* (MOGAIS), respectivamente.

Por fim, os principais trabalhos da literatura que estão relacionados às contribuições desta tese são apresentados na Seção 5.9. Este capítulo não contém exemplos de aplicação dos quatro algoritmos propostos, sendo que esse aspecto será tratado nos três próximos capítulos.

5.2 Possível Abordagem para Manipular Blocos Construtivos

O conceito e o problema da descoberta e preservação de blocos construtivos não são exclusividades dos algoritmos imuno-inspirados. São assuntos que já foram abordados, por exemplo, com relação aos algoritmos genéticos (Goldberg, 1989; Holland, 1975; Mitchell, 1996). A partir de então, várias tentativas para evitar o rompimento dessas soluções parciais foram realizadas, incluindo a mudança na representação das soluções e o desenvolvimento de operadores evolutivos específicos. Infelizmente, abordagens deste tipo são altamente dependentes do problema, levando ao desenvolvimento de algoritmos muito específicos. Além disso, elas requerem que o usuário possua um conhecimento prévio no domínio do problema. Entretanto, em muitos problemas do mundo real, este conhecimento não está disponível previamente.

Nesse sentido, uma maneira mais eficiente de lidar com os blocos construtivos é deixar que o próprio algoritmo aprenda as relações existentes entre as variáveis, usando informações estatísticas extraídas a partir do conjunto das melhores soluções.

As melhores soluções podem ser encaradas como amostras geradas a partir de uma distribuição de probabilidades desconhecida. Em estatística, o termo distribuição de probabilidade para um conjunto de dados refere-se à probabilidade desses dados serem observados. O algoritmo bio-inspirado poderia, então, estimar a distribuição de probabilidades das melhores soluções e, posteriormente, utilizá-la para gerar novas soluções candidatas que apresentassem características similares àquelas contidas no conjunto de melhores soluções.

Dessa maneira, um algoritmo com esse princípio de funcionamento consegue manter os blocos

construtivos do problema, uma vez que foram identificados, de forma eficiente sem qualquer informação prévia, evitando o rompimento das soluções parciais.

Na Figura 5.1, está ilustrado o esquema de evolução de soluções candidatas para o problema fictício usando um modelo probabilístico, em vez do tradicional operador de mutação.

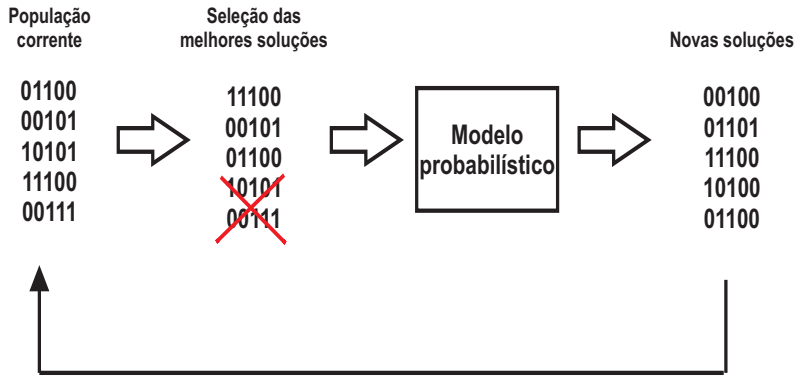


Figura 5.1: Exemplo do uso de um modelo probabilístico para gerar novas soluções candidatas.

A incorporação de um mecanismo semelhante nos algoritmos imunológicos artificiais pode ajudá-los a identificar e manipular automaticamente os blocos construtivos. Dessa forma, problemas de otimização mais desafiadores, que apresentam interações complexas das variáveis, podem ser tratados mais facilmente.

Além da eficiente manipulação de blocos construtivos, pode-se citar mais duas motivações para se utilizar essa nova forma de evoluir a população. Com a substituição dos operadores de clonagem e mutação, não é preciso mais se preocupar com a escolha dos operadores e parâmetros mais adequados para cada tipo de problema. É evidente que a construção do modelo probabilístico requer a determinação de alguns parâmetros, mas estes são de mais fácil definição. A outra motivação é a possibilidade de analisar melhor e entender o processo de evolução do algoritmo. Nas seções seguintes são apresentados quatro sistemas imunológicos artificiais que utilizam modelos probabilísticos para evoluir a população de soluções.

5.3 *Bayesian Artificial Immune System* (BAIS)

O primeiro algoritmo imuno-inspirado concebido utilizando um modelo gráfico probabilístico foi o *Bayesian Artificial Immune System* (BAIS) (Castro & Von Zuben, 2009a). Ele foi desenvolvido para resolver problemas de otimização mono-objetivo no espaço discreto.

No lugar dos operadores de clonagem e mutação, entram em cena as etapas de construção do modelo probabilístico e subsequente amostragem de novas soluções a partir do modelo gerado. O modelo probabilístico utilizado por BAIS é a rede bayesiana, conforme ilustrado na Figura 5.2.

Vale salientar que as aplicações não estão restritas a problemas em que as soluções candidatas são representadas na forma de vetores de bits, como as Figuras 5.1 e 5.2 podem, eventualmente, sugerir.

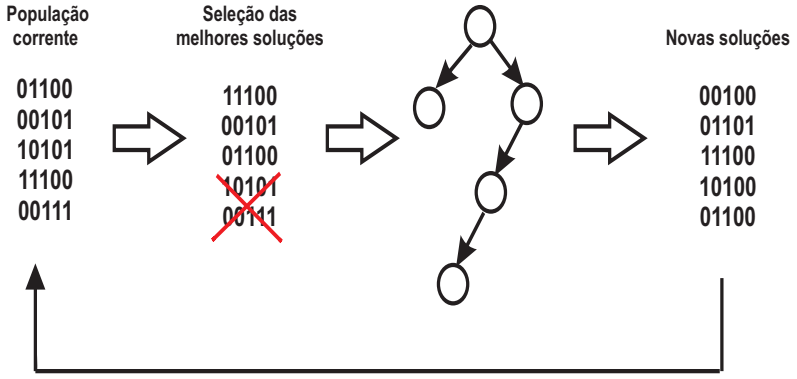


Figura 5.2: SIA usando um modelo probabilístico no lugar dos operadores de clonagem e mutação.

A estrutura desse novo sistema imunológico artificial é apresentada no Algoritmo 5.1. A população inicial é gerada aleatoriamente. Enquanto a condição de parada não for satisfeita, o algoritmo avalia cada solução candidata e seleciona as *n* melhores, a partir das quais será gerada a rede bayesiana. Em seguida, novas soluções são geradas a partir da rede obtida. É realizado, então, o processo de eliminação das soluções semelhantes e das que apresentam baixo valor de *fitness*. Para ajudar a manter diversidade na população, novas soluções são geradas aleatoriamente. Se a inserção de novas soluções fosse realizada também por meio da amostragem da rede bayesiana, essas novas soluções tenderiam a ser muito semelhantes àquelas já existentes na população, levando o algoritmo a explorar o espaço de busca de forma polarizada.

Algoritmo 5.1 *Bayesian Artificial Immune System* (BAIS)

Início
Iniciar a população de anticorpos aleatoriamente;
Enquanto condição de parada não for satisfeita **faça**
Avaliar a população;
Selecionar os melhores anticorpos;
Construir a rede bayesiana;
Amostrar novos anticorpos;
Eliminar os anticorpos com *fitness* menor que um limiar;
Eliminar anticorpos semelhantes;
Inserir aleatoriamente anticorpos na população;
Fim enquanto
Fim

Uma vez definido qual modelo utilizar, é necessário encontrar uma forma de construí-lo. Na

proposta apresentada aqui, foi utilizado o paradigma da busca e pontuação, já descrito no capítulo 2. Inicia-se o aprendizado com um grafo sem arcos. Em seguida, um arco é adicionado ao grafo. O próximo passo consiste em avaliar se a nova estrutura é melhor que a anterior. Se for melhor, o novo arco adicionado é mantido e tenta-se adicionar outro. Este processo continua até que nenhuma nova estrutura seja melhor que a anterior.

Na Figura 5.3, está ilustrado o esquema desta busca para um problema com 4 variáveis. O algoritmo começa sem ligação entre os nós (a) e uma avaliação desta rede não conectada é feita. Em seguida, adiciona-se um arco e a rede resultante é avaliada novamente (b). Se a nova estrutura for melhor, mantém-se o arco. Caso contrário, ele é retirado. Este processo continua até que nenhuma estrutura seja melhor que a anterior. No exemplo, a rede (e) é retornada como solução.

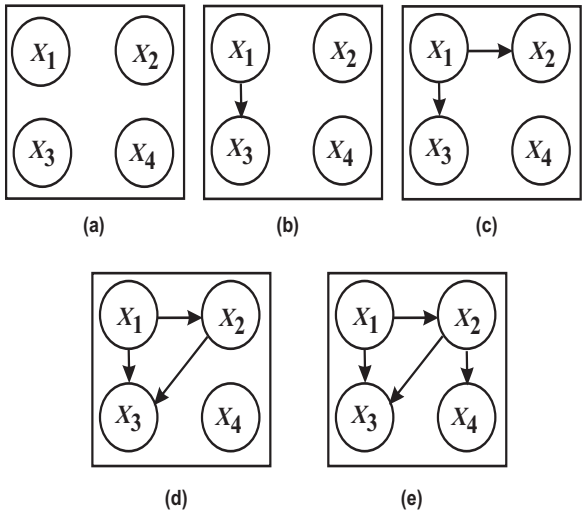


Figura 5.3: Processo iterativo para construção do modelo gráfico probabilístico.

Para avaliar a qualidade das redes geradas, pode-se utilizar qualquer uma das métricas apresentadas no capítulo 2. BAIS utiliza a métrica K2, dada pela Equação (5.1):

$$p(D|B) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{5.1}$$

em que D representa o conjunto de dados, B é a rede bayesiana sendo avaliada, n é o número de variáveis, q_i denota o número de possíveis valores que os pais de X_i podem assumir, r_i é o número de possíveis valores de X_i, N_{ijk} é o número de casos em que X_i assume o *k*-ésimo valor com seus pais assumindo o *j*-ésimo valor, e $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Para amostrar novos indivíduos, é utilizado o método da Amostragem Probabilística Lógica (PLS, do inglês *Probabilistic Logic Sampling*) (Henrion, 1988). Esse método amostra primeiramente todas as variáveis que não são dependentes de nenhuma outra. Em seguida, amostram-se as variáveis-pais

para depois amostrar as variáveis-filhas.

5.3.1 Exemplo de aplicação

Para ilustrar a capacidade de lidar com blocos construtivos de forma eficaz, BAIS foi aplicado ao problema Trap-5 (Deb & Goldberg, 1993) e comparado com um sistema imunológico artificial tradicional. Este problema consiste em dividir o vetor de soluções (vetor binário n-dimensional) em partições disjuntas de 5 bits cada. Este particionamento permanece fixo durante todo o processo de otimização e o algoritmo não possui nenhum conhecimento sobre como foi feito tal particionamento. Em seguida, uma função (equação 5.2) é aplicada para cada grupo de 5 bits.

$$trap_5(u) = \begin{cases} 5, & \text{se } u = 5 \\ 4 - u, & \text{se } u < 5 \end{cases} \tag{5.2}$$

em que u denota a quantidade de vezes que aparece o símbolo 1 no bloco de 5 bits. Na Figura 5.4 é possível visualizar esta função para um bloco de 5 bits.

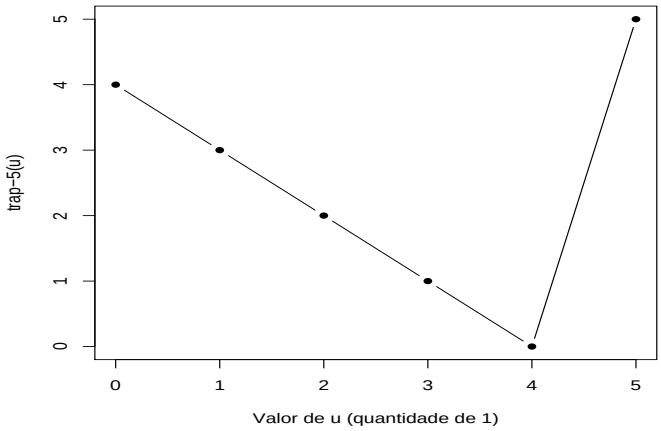


Figura 5.4: Exemplo do problema Trap-5 para um bloco de 5 bits

O objetivo é maximizar a função. Para um vetor de soluções n-dimensional, esta função possui seu ótimo global quando todos os seus bits são iguais a "1", e possui $2^{n/5} - 1$ ótimos locais. A contribuição de todas as funções são combinadas para formar o *fitness* do anticorpo. Portanto, o *fitness* de um anticorpo n-dimensional Ab_i é dado por:

$$fitness(Ab_i) = \sum_{i=1}^{n/5} trap_5(S_i) \tag{5.3}$$

em que S_i são blocos de 5 bits.

O algoritmo BAIS foi comparado com um sistema imunológico artificial tradicional para $n = 100$. O tamanho da população inicial foi estabelecido como 50 para ambos os algoritmos. Os dois algoritmos foram executados 30 vezes e ambos foram capazes de encontrar o ótimo global. Entretanto, os resultados médios junto às 30 execuções mostraram que BAIS foi capaz de encontrar o ótimo global e múltiplos ótimos locais com um número menor de avaliações de *fitness* que o concorrente. Enquanto BAIS avaliou, em média, 645,8 soluções até encontrar o ótimo global, o sistema imunológico artificial avaliou 1073,2 soluções.

O intuito deste experimento comparativo é mostrar que o algoritmo imuno-inspirado tradicional não é capaz de manipular blocos construtivos, pois seu operador de mutação não leva em consideração a interação das variáveis do problema, causando o rompimento dos blocos construtivos. Análises mais exaustivas neste sentido serão abordadas em trabalhos futuros. Maiores detalhes sobre os resultados deste experimento podem ser encontrados em Castro & Von Zuben (2009a).

5.4 *Gaussian Artificial Immune System* (GAIS)

Para lidar com problemas de otimização no espaço contínuo, foi proposto o *Gaussian Artificial Immune System* (GAIS). Seu funcionamento é similar ao do BAIS, porém o modelo probabilístico utilizado para expressar o relacionamento entre as variáveis é a rede gaussiana. A estrutura do GAIS é apresentada no Algoritmo 5.2.

Algoritmo 5.2 *Gaussian Artificial Immune System* (GAIS)

Início
  Iniciar a população de anticorpos aleatoriamente;
  Enquanto condição de parada não for satisfeita **faça**
    Avaliar a população;
    Selecionar os melhores anticorpos;
    Construir a rede gaussiana;
    Amostrar novos anticorpos;
    Eliminar os anticorpos com *fitness* menor que um limiar;
    Eliminar anticorpos semelhantes;
    Inserir aleatoriamente anticorpos na população;
  Fim enquanto
Fim

O aprendizado da rede gaussiana também utiliza o paradigma da busca e pontuação. A métrica de avaliação utilizada é a *Bayesian Information Criterion* (BIC) para redes gaussianas, previamente

apresentada no capítulo 2:

$$p(D|G) = \left[\prod_{l=1}^{N} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi v_i}} e^{-1/2v_i(x_{li} - m_i - \sum_{x_k \in \boldsymbol{pa_i}} b_{ki}(x_{lk} - m_k))^2} \right] - f(N) * dim(G), \tag{5.4}$$

em que D representa o conjunto de dados, G é a rede gaussiana sendo avaliada, N denota a quantidade de instâncias, n é o número de variáveis, b_{ki} é o coeficiente de regressão linear refletindo o relacionamento entre X_k e X_i, $\boldsymbol{pa_i}$ é o conjunto de pais da variável X_i e v_i é a variância condicional de X_i dados $X_1, ..., X_{i-1}$ $\forall i, k$. A função $f(N) = \frac{1}{2}\, ln\, N$ é responsável por penalizar modelos complexos e $dim(G) = 2n + \sum_{i=1}^{n} \boldsymbol{pa_i}$ é a quantidade de parâmetros a serem estimados.

Para amostrar novas soluções, GAIS utiliza um método introduzido por Ripley (1987), semelhante ao Amostragem Probabilística Lógica. Primeiramente, são amostradas as variáveis que não são dependentes de nenhuma outra. Em seguida, amostram-se as variáveis-pais para depois amostrar as variáveis-filhas.

GAIS assume que os dados disponíveis para construir o modelo probabilístico foram gerados por uma única distribuição gaussiana multivariada. Para problemas com funções unimodais, o algoritmo consegue estimar corretamente a distribuição de probabilidade para os dados. Entretanto, para funções multimodais, em que os ótimos locais não estão concentrados em uma única região, mas sim espalhados, uma única distribuição gaussiana multivariada não é capaz de modelar os dados de forma satisfatória.

Para sanar esta deficiência, uma técnica de agrupamento (*clustering*) foi incorporada ao GAIS. Agora, o algoritmo agrupa as soluções selecionadas e processa cada grupo separadamente. Isso significa que, para cada grupo, uma rede gaussiana é gerada usando a mesma métrica, como anteriormente. Para agrupar as soluções, uma grande diversidade de algoritmos de agrupamento pode ser empregada. GAIS utiliza o *k-means*, por ser um algoritmo simples e apresentar bom desempenho. Um inconveniente do *k-means* é o fato dele exigir previamente o número de grupos (*clusters*), denotado por k. No GAIS, esse parâmetro é especificado empiricamente.

Com esta modificação, a função densidade de probabilidade conjunta das melhores soluções é dita ser uma mistura de funções densidade de probabilidade gaussianas, expressa por:

$$f(x) = \sum_{i=1}^{k} \alpha_i f_i(x), \quad \sum_{i=1}^{k} \alpha_i = 1, \;\; \alpha_i \geq 0, \tag{5.5}$$

em que $f_i(x)$ é uma única função densidade de probabilidade gaussiana multivariada, k é quantidade de $clusters$ e, consequentemente, a quantidade de componentes do modelo de mistura, e α_i é o coeficiente do i-ésimo componente da mistura. O valor de cada α_i é proporcional à quantidade de

pontos no i-ésimo *cluster*:

$$\alpha_i = \frac{c_i}{\sum_{j=1}^{k} c_j}, \quad i = 1, ..., k, \tag{5.6}$$

em que c_i denota a quantidade de elementos no *cluster* i.

5.5 Otimização Multiobjetivo

Um problema de otimização multiobjetivo consiste em otimizar simultaneamente duas ou mais funções-objetivo, possivelmente conflitantes, no sentido de que a melhora no valor de um dos objetivos pode degradar o valor dos outros (Deb, 2001; Zitzler et al., 2000).

Formalmente, um problema de otimização multiobjetivo consiste em (Coello Coello, 1998):

$$\left.\begin{array}{rll} \text{minimizar/maximizar} & f_m(x), & m = 1, 2, ..., M; \\ \text{sujeito a} & g_j(x) \geq 0, & j = 1, 2, ..., J; \\ & h_k(x) = 0, & k = 1, 2, ..., K; \\ & x_i^{(I)} \leq x_i \leq x_i^{(S)}, & i = 1, 2, ..., n. \end{array}\right\} \tag{5.7}$$

em que $x = [x_1, ..., x_n]$ é o vetor de variáveis de decisão, também chamado de solução. Os valores x_i^I e x_i^S representam o valor mínimo e o valor máximo, respectivamente, para a variável x_i, i=1,2...n. Estes limites definem o espaço de variáveis de decisão ou, simplesmente, espaço de decisão. Associado ao problema existem *J* restrições de desigualdade e *K* restrições de igualdade, definidas pelas funções $g_j(x)$ e $h_k(x)$, respectivamente. Uma solução que satisfaz todas as restrições do problema e todos os seus $2n$ limites é chamada de solução factível. O conjunto de todas as soluções factíveis formam a região factível ou espaço de busca S.

Cada uma das M funções-objetivo $f_1(x), f_2(x), ..., f_M(x)$ pode ser minimizada ou maximizada. Entretanto, para facilitar o processo de otimização, pode-se converter todas para serem apenas minimizadas ou apenas maximizadas. O princípio da dualidade permite converter um problema de maximização em um problema de minimização pela multiplicação da função objetivo por -1. As funções objetivo constituem um espaço multidimensional chamado de espaço de objetivos, *Z*. Para cada solução no espaço de decisão, existe um ponto correspondente no espaço de objetivos, denotado por $f(x) = (f_1(x), f_2(x), ..., f_M(x)) = z = (z_1, z_2, ..., z_M)$. O mapeamento ocorre, então entre um vetor x *n*-dimensional e um vetor $f(x)$ *M*-dimensional, conforme ilustrado na Figura 5.5.

Quando o problema possui apenas um único objetivo, a tarefa de avaliar as soluções candidatas e encontrar a solução ótima (ou as soluções ótimas no caso multimodal) é mais fácil, se comparada com a otimização multiobjetivo. Em problemas em que várias funções-objetivo são consideradas ao

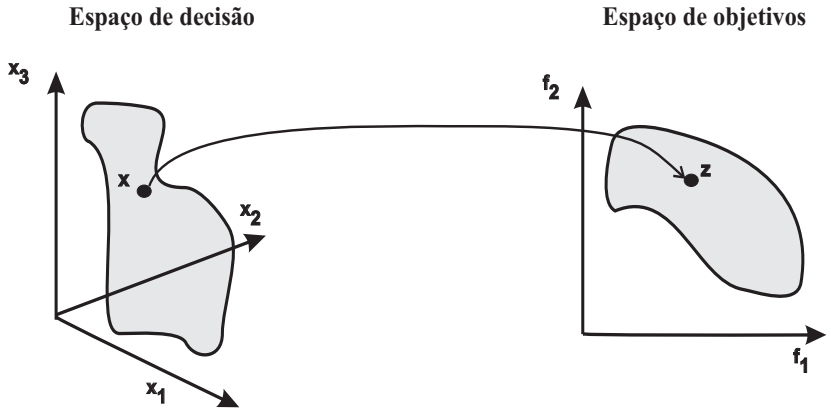


Figura 5.5: Representação ilustrativa do espaço de decisão em 3 dimensões e o correspondente espaço de objetivos em 2 dimensões.

mesmo tempo, o processo de avaliar as soluções torna-se mais complexo, pois não existe uma única solução ótima para todas as funções-objetivo simultaneamente. Nesse caso, a noção de otimalidade comumente adotada na literatura é aquela associada com a otimalidade de Pareto, a qual pode ser expressa usando o conceito de dominância.

Considere um problema com M funções-objetivo $f_i(x)$, i=1,2,...,M, que, sem perda de generalidade, devem ser minimizadas. Os seguintes conceitos são introduzidos (Deb, 2001):

Definição 5.1. (Dominância): uma solução x domina uma solução y (denotado por $x \preceq y$) se e somente se x não é pior que y em nenhum objetivo e x é melhor que y em pelo menos um objetivo. Formalmente,

$$\forall i \in \{1,2,...,M\}\text{: } f_i(x) \leq f_i(y) \bigwedge \exists i \in \{1,2,...,M\}\text{: } f_i(x) < f_i(y).$$

Definição 5.2. (Solução ótima de Pareto): Uma solução x é dita ser ótima de Pareto se, e somente se, x não é dominada por nenhuma outra solução factível do espaço de busca S, ou seja:

$$\nexists y \in S : y \preceq x.$$

Definição 5.3. (Conjunto ótimo de Pareto): é o conjunto *PS* de todas as soluções consideradas ótimas de Pareto, ou seja:

$$PS = \{x| \ \nexists y \in S : y \preceq x \ \}.$$

Portanto, as soluções contidas no conjunto *PS* são todas não-dominadas e são as soluções do problema.

Definição 5.4. (Fronteira de Pareto): é o conjunto *PF* dos valores das funções-objetivo de todas as soluções ótimas de Pareto

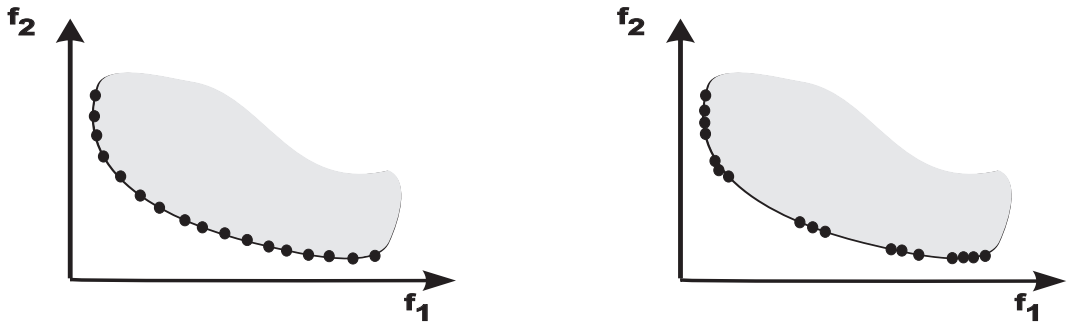
$$PF=\{f(x) = (f_1(x),..., f_M(x)) \ | x \in PS\}$$

A fronteira de Pareto é, portanto, formada pelos valores das funções-objetivo correspondentes a cada solução que compõe o conjunto ótimo de Pareto (soluções não-dominadas). Na ausência de informações adicionais sobre a relevância dos objetivos, todas essas soluções são igualmente importantes. Nesse sentido, duas metas práticas devem ser alcançadas por um algoritmo de otimização multiobjetivo:

1. Encontrar um conjunto de soluções que pertença ou que pelo menos esteja o mais próximo possível da fronteira de Pareto;

2. Encontrar um conjunto de soluções com a maior diversidade possível, de preferência uniformemente distribuídas ao longo da fronteira de Pareto, como no exemplo ilustrado na Figura 5.6(a). Já na Figura 5.6(b), é apresentado um exemplo de distribuição não desejável ao longo da fronteira de Pareto, pois embora possua a mesma quantidade de soluções que a proposta mostrada na Figura 5.6(a), as soluções possuem pouca diversidade entre si, ficando concentradas em algumas regiões da fronteira.

A primeira meta é bastante óbvia, já que a fronteira de Pareto satisfaz as condições de otimalidade. Já a segunda meta é necessária para evitar qualquer polarização em favor de uma função-objetivo específica. Somente com um conjunto diverso de soluções cuja distribuição se aproxima de uma distribuição uniforme pela fronteira de Pareto é que se pode assegurar um "equilíbrio" no que diz respeito a satisfazer os objetivos. A diversidade pode ser promovida nos espaços de decisão e de objetivos, uma vez que a proximidade de duas soluções no espaço de decisão não implica proximidade no espaço de objetivos, e vice-versa.



(a) (b)

Figura 5.6: Exemplos de resultados em otimização multiobjetivo (no espaço dos objetivos): (a) várias soluções distribuídas uniformemente e (b) soluções concentradas em algumas regiões.

Ao tratar problemas de otimização multiobjetivo, métodos clássicos de busca e otimização que não são baseados em população apresentam resultados aquém do desejado (Coello Coello, 1998; Zitzler

et al., 2000). Isso porque: (*i*) muitos deles não podem encontrar múltiplas soluções em uma única execução, requerendo muitas execuções para obter um resultado satisfatório; (*ii*) múltiplas execuções do método não garantem encontrar soluções diversas; e (*iii*) muitos deles não podem manipular problemas em que a fronteira de Pareto é côncava e/ou descontínua.

Como alternativa, uma variedade de meta-heurísticas baseadas em população têm sido propostas para resolver esta classe de problemas, como algoritmos genéticos (Deb et al., 2000; Zitzler & Thiele, 1999), sistemas imunológicos artificiais (Coelho & Von Zuben, 2006; Coello Coello & Cortés, 2005) e algoritmos de estimação de distribuição (Khan et al., 2002; Laumanns & Ocenasek, 2002). Esses algoritmos são menos suscetíveis ao formato ou descontinuidade da fronteira de Pareto e são inerentemente capazes de encontrar várias soluções próximas ou pertencentes à fronteira de Pareto em apenas uma execução. São essas vantagens que se pretende explorar ao longo das próximas seções, as quais estendem respectivamente os resultados das Seções 5.3 e 5.4 para o contexto multiobjetivo.

5.6 *Multi-Objective Bayesian Artificial Immune System* (MOBAIS)

Foi desenvolvida uma versão do algoritmo BAIS para lidar com problemas de otimização multiobjetivo no espaço discreto, denominada *Multi-Objective Bayesian Artificial Immune System* (MOBAIS) (Castro & Von Zuben, 2008b, 2009c), apresentada no Algoritmo 5.3.

Algoritmo 5.3 *Multi-objective Bayesian Artificial Immune System* (MOBAIS)

Início
  Iniciar a população de anticorpos aleatoriamente;
  Enquanto condição de parada não for satisfeita **faça**
    Avaliar a população;
    Classificar os anticorpos usando o conceito de dominância;
    Calcular a distância entre os anticorpos (espaço de objetivos), baseado na densidade;
    Selecionar os melhores anticorpos;
    Construir a rede bayesiana;
    Amostrar novos anticorpos;
    Eliminar anticorpos semelhantes;
    Inserir aleatoriamente anticorpos na população;
  Fim enquanto
Fim

Assim como seu antecessor, MOBAIS também adota rede bayesiana como modelo probabilístico para expressar as interações das variáveis. A forma de construí-la também é igual, conforme explicado

na Seção 5.3. A diferença está na forma de avaliar os anticorpos e nos mecanismos de seleção e supressão, os quais serão descritos a seguir.

A forma de avaliar cada solução candidata utiliza o conceito de dominância, já explicado na seção anterior. Em seguida, MOBAIS divide as soluções em classes. Na classe 1 estão todas as soluções que não são dominadas por nenhuma outra. Na classe 2 estão as soluções que são dominadas por pelo menos uma solução da classe 1 e não são dominadas por nenhuma outra solução da população restante. Esse processo continua até que todas as soluções estejam ordenadas em classes. Na Figura 5.7 está ilustrado um exemplo desse procedimento de classificação para um problema de otimização com dois objetivos.

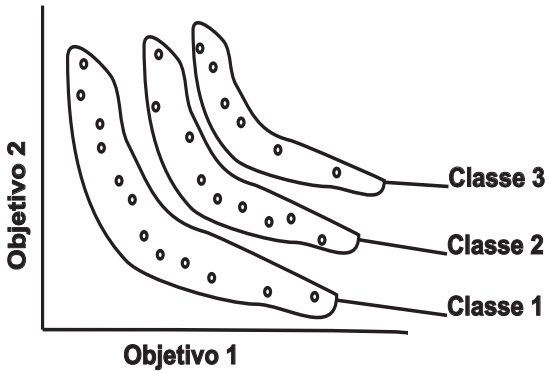


Figura 5.7: Exemplo ilustrativo do procedimento de classificação para um problema com 2 objetivos.

Na etapa de seleção, o algoritmo seleciona as *n* melhores soluções, dando prioridade àquelas pertencentes à classe 1 (soluções não-dominadas por nenhuma outra). Se a quantidade de soluções na classe 1 não for suficiente, o algoritmo passa a selecionar soluções da classe seguinte.

Além de encontrar soluções perto da fronteira de Pareto, é também essencial obter boa cobertura da fronteira, ou seja, que as soluções estejam espalhadas ao longo da fronteira. Desta forma, é desejável um mecanismo para manter diversidade no espaço de objetivos. MOBAIS utiliza um mecanismo que calcula a distância de cada solução para seus vizinhos adjacentes, denotada por $dist$. Esse cálculo da distância, que leva em consideração a densidade da vizinhança, é chamado de *crowding distance* e foi proposto por Deb et al. (2000).

O valor $dist$ é calculado para cada classe, separadamente. As soluções pertencentes à mesma classe são ordenadas de acordo com cada função-objetivo e, então, calcula-se a distância de cada solução para os seus vizinhos adjacentes. As soluções com menor e maior valores da função-objetivo não participam do cálculo da distância. A essas soluções, é atribuído um alto valor para $dist$, a fim de garantir que elas sejam sempre selecionadas. Quanto maior o valor da distância de uma solução, menos densa é a sua vizinhança. Dessa forma, MOBAIS dá preferência para soluções que possuem maior valor de $dist$.

No Algoritmo 5.4, é apresentado o cálculo dessa distância. Para uma solução $P(i)$, $f_m(P(i))$ denota o valor da função-objetivo m para essa solução.

Algoritmo 5.4 Cálculo da distância baseado na densidade *crowding distance*

Início
Para cada classe c **faça**
P = soluções da classe c;
Q = quantidade de soluções em P;
Para cada objetivo m **faça**
R = ordenar P usando o objetivo m;
$dist(R(1)) = \infty$;
$dist(R(Q)) = \infty$;
Para i=2 até Q-1 **faça**
$dist(P(i)) = dist(P(i)) + f_m(P(i+1)) - f_m(P(i-1))$
Fim para
Fim para
Fim para
Fim

Para manter a diversidade na população, é realizado um processo de supressão das soluções semelhantes, tanto no espaço de variáveis quanto no espaço de objetivos. Primeiramente, a distância euclidiana no espaço de variáveis é calculada entre todos os indivíduos da população. Em seguida, essa distância é normalizada com respeito à máxima distância encontrada. As soluções semelhantes, dado um certo limiar, são eliminadas, mantendo-se apenas aquela que possui a menor classe. Caso as classes sejam iguais, mantém-se a solução com maior valor $dist$.

MOBAIS elimina também as soluções que possuem o valor $dist$ menor que um limiar e aquelas que pertencem à classe mais alta (soluções dominadas por todas as outras). Em seguida, é realizada a inserção de novas soluções geradas aleatoriamente.

5.7 *Multi-Objective Gaussian Artificial Immune System* (MOGAIS)

Para resolver problemas de otimização multiobjetivo no espaço contínuo, foi criado o *Multi-Objective Gaussian Artificial Immune System* (MOGAIS), apresentado no Algoritmo 5.5.

Este algoritmo foi concebido tendo como inspiração os algoritmos GAIS e MOBAIS. Similar ao GAIS, MOGAIS adota a rede gaussiana como modelo probabilístico para expressar as interações das variáveis. A forma de construí-la é pelo paradigma de busca e pontuação, igual aos outros algoritmos.

Algoritmo 5.5 *Multi-objective Gaussian Artificial Immune System* (MOGAIS)

Início
Iniciar a população de anticorpos aleatoriamente;
Enquanto condição de parada não for satisfeita **faça**
Avaliar a população;
Classificar os anticorpos usando o conceito de dominância;
Calcular a distância entre os anticorpos (espaço de objetivos), baseado na densidade;
Selecionar os melhores anticorpos;
Construir a rede gaussiana;
Amostrar novos indivíduos;
Eliminar anticorpos semelhantes;
Inserir aleatoriamente anticorpos na população;
Fim enquanto
Fim

Para avaliar cada solução candidata, MOGAIS classifica as soluções usando o conceitos de dominância, assim como MOBAIS. Na etapa de seleção, a prioridade é para aquelas pertencentes à classe 1 (soluções não-dominadas por nenhuma outra). Se a quantidade de soluções na classe 1 não for suficiente, o algoritmo passa a selecionar soluções da classe seguinte. Visando manter diversidade no espaço de objetivos, GAIS também utiliza *crowding distance*.

Para manter a diversidade na população, é realizado um processo de supressão das soluções semelhantes, tanto no espaço de variáveis quanto no espaço de objetivos. Primeiramente, a distância euclidiana no espaço de variáveis é calculada entre todos os indivíduos da população. Em seguida, essa distância é normalizada com respeito à máxima distância encontrada. As soluções semelhantes, dado um certo limiar, são eliminadas, mantendo-se apenas aquela que possui a menor classe. Caso as classes sejam iguais, mantém-se a solução com maior valor para a *crowding distance*. As soluções com *crowding distance* menor que um limiar e aquelas que pertencem à classe mais alta (soluções dominadas por todas as outras) são eliminadas da população. Em seguida, é realizada a inserção de novas soluções geradas aleatoriamente.

Para resolver problemas de otimização multiobjetivo e multimodais, a função densidade de probabilidade conjunta das melhores soluções é um modelo de mistura de gaussianas. O modelo de mistura é estimado usando um algoritmo de agrupamento, como o *k-means*. O número de *clusters* é definido empiricamente e o valor do coeficiente de cada componente da mistura é proporcional à quantidade de pontos no *cluster* associado a ele.

5.8 Caracterização, plausibilidade imunológica e limitações dos algoritmos

Existem diversas características em comum entre os algoritmos propostos nesta tese. Dentre elas, merecem destaque: habilidade para identificar e preservar os blocos construtivos, tamanho da população automaticamente ajustável, manutenção de diversidade na população e capacidade de realizar busca multimodal. Como uma consequência direta destas características, tem-se: a busca torna-se mais robusta, no sentido de que não há muita variação de desempenho entre buscas derivadas de re-execuções do algoritmo para um mesmo problema; e tende a ocorrer uma redução acentuada no número de avaliações de soluções candidatas até que se atinja um certo nível de desempenho.

É importante notar que a seleção clonal e teoria da rede permanecem como inspirações imunológicas nos algoritmos propostos. O mecanismo que implementa a teoria da rede continua o mesmo de um sistema imunológico artificial tradicional, como aquele do Algoritmo 3.1. O que mudou foi a forma de executar a seleção clonal (seleção dos melhores indivíduos seguida de um processo de mutação). Agora, a seleção clonal é realizada por meio dos modelos probabilísticos.

Embora os algoritmos apresentem todas estas características vantajosas e os experimentos descritos nos capítulos seguintes comprovem isto, eles apresentam algumas limitações. A primeira está relacionada ao custo computacional para implementá-los. Já que a cada iteração é necessário construir o modelo probabilístico, os algoritmos não são indicados para problemas de otimização muito simples, pois existem técnicas mais eficientes para resolvê-los. Nos capítulos de experimentos e na seção de perspectivas futuras, são abordadas duas formas para aliviar o custo computacional dos algoritmos.

A outra limitação está na criação da rede bayesiana ou da rede gaussiana para problemas de dimensão elevada. Nestes casos, a qualidade do modelo probabilístico gerado pode não ser suficiente para promover ganho de desempenho.

5.9 Trabalhos Relacionados

Muitos trabalhos têm sido propostos para a manipulação eficiente de blocos construtivos sob a perspectiva de estimação de distribuição. Esta seção visa apresentar algumas das principais propostas reportadas na literatura. Uma extensa revisão pode ser encontrada em Pelikan (2005) e Pelikan et al. (2002).

Na área de sistemas imunológicos artificiais, esta tese representa a primeira contribuição na área. Entretanto, a ideia de evoluir a população de soluções amostrando novos indivíduos a partir de uma distribuição de probabilidade já foi aplicada antes em algoritmos genéticos (Larrañaga et al.,

2000a; Pelikan et al., 2000). Os operadores de cruzamento e mutação foram substituídos por modelos probabilísticos. Esses algoritmos recebem o nome de Algoritmos de Estimação de Distribuição ou Algoritmos Genéticos Baseados em Modelos Probabilísticos (Mühlenbein & Paass, 1996).

Algoritmos de estimação de distribuição representam um paradigma computacional relativamente novo e o interesse por essa classe de algoritmos tem crescido muito nos últimos anos. Resultados interessantes têm sido obtidos com tal metodologia quando aplicados em problemas simples de otimização. Para problemas mais complexos, diversos ajustes nos algoritmos precisam ser feitos para se chegar a uma solução satisfatória. Isso porque os algoritmos genéticos possuem algumas deficiências, particularmente no que diz respeito à manutenção de diversidade e busca multimodal.

São várias as propostas de algoritmos nesse contexto. Dependendo do modelo probabilístico utilizado para expressar o relacionamento das variáveis, o algoritmo pode ser classificado em um dos três grupos: sem dependência, dependência aos pares e múltiplas dependências. A seguir, serão apresentados os algoritmos mais relevantes da literatura.

5.9.1 Sem dependência entre as variáveis

Os algoritmos pertencentes a este grupo são os mais simples e não consideram nenhuma relação entre as variáveis, ou seja, assumem que elas são independentes. Devido à simplicidade do modelo probabilístico utilizado, os algoritmos nesta categoria não requerem custo computacional elevado e apresentam bom desempenho quando aplicados a problemas para os quais o relacionamento entre as variáveis não é tão significativo.

O primeiro algoritmo proposto nesta categoria foi o *Population Based Incremental Learning* (PBIL) (Baluja, 1994). Nesse algoritmo, as soluções candidatas são representadas por vetores binários e chamadas de cromossomos. Existe um vetor de probabilidades, da mesma dimensão do cromossomo, denotando a probabilidade de cada gene do cromossomo receber o valor "1". Inicialmente, o vetor de probabilidades contém o mesmo valor para todas as posições, 50%. A cada iteração, as melhores soluções são selecionadas e o vetor de probabilidades é atualizado seguindo uma regra baseada no aprendizado hebbiano. Esse vetor é utilizado, então, para amostrar novas soluções a cada iteração. Sebag & Ducoulombier (1998) desenvolveram uma versão desse algoritmo para lidar com problemas de otimização no espaço contínuo, denominada PBIL$_c$, na qual cada variável é modelada por um função densidade de probabilidade gaussiana. A cada iteração, a média e a variância de cada gaussiana são estimadas usando as melhores soluções.

Posteriormente, Mühlenbein & Paass (1996) propuseram o *Univariate Marginal Distribution Algorithm* (UMDA). Seu funcionamento é similar ao do PBIL, com a exceção de que o vetor de probabilidades é atualizado calculando-se apenas a frequência com que aparece o número "1" nas melhores soluções selecionadas.

O *Compact Genetic Algorithm* (cGA), proposto por Harik et al. (1999), também utiliza um vetor de probabilidades, o qual é iniciado contendo o valor 50% para todas as posições. Em seguida, duas soluções candidatas são amostradas a partir desse vetor e avaliadas pela função de *fitness*. O vetor de probabilidades é atualizado utilizando o valor dos genes do cromossomo com melhor *fitness*. Se os cromossomos possuírem o mesmo valor num mesmo gene, o vetor de probabilidades não é atualizado para aquela posição. Esse procedimento continua até que o vetor de probabilidades tenha convergido.

O *Reinforcement Learning Estimation of Distribution Algorithm* (RELEDA), proposto por Paul & Iba (2003), é similar aos outros três algoritmos já apresentados. A diferença é que o vetor de probabilidades é atualizado de acordo com um método de aprendizado por reforço. Os autores compararam RELEDA com PBIL e UMDA e mostraram que seu algoritmo converge mais rapidamente para a solução ótima que os concorrentes. Os autores também desenvolveram o *Real-Coded Estimation of Distribution Algorithm* (RECEDA) para otimização de funções com variáveis contínuas. A partir das melhores soluções, a média e a matriz de covariância são calculadas.

O *Distribution Estimation Using MRF* (DEUM) (Shakya et al., 2004) pode ser visto como uma adaptação dos outros quatro algoritmos. Em vez de calcular a frequência para cada gene do cromossomo, ele atualiza o vetor de probabilidades utilizando um campo aleatório de Markov (MRF, do inglês *Markov Random Field*). Os autores mostraram empiricamente que DEUM apresenta bons resultados para uma série de problemas de otimização (Shakya et al., 2005).

5.9.2 Dependência entre pares de variáveis

Essa classe de algoritmos assume dependência entre pares de variáveis. Dessa forma, é preciso definir como representar a estrutura do modelo probabilístico, a fim de expressar o condicionamento das variáveis.

O *Mutual Information Maximization for Input Clustering* (MIMIC) (De Bonet et al., 1997) utiliza uma estrutura em cadeia entre as variáveis para poder representar o relacionamento entre elas, conforme ilustrado na Figura 5.8(a). Para encontrar a melhor estrutura de cadeia de variáveis, o algoritmo utiliza um mecanismo de busca que visa maximizar a informação mútua.

Baluja & Davies (1997) propuseram o *Combining Optimizers with Mutual Information Trees* (COMIT), um algoritmo que utiliza uma estrutura em árvore para modelar a dependência entre as variáveis. Nesse tipo de estrutura, todas as variáveis precisam ter um pai, com exceção da variável-raiz. Este algoritmo é mais geral que o MIMIC, pois duas ou mais variáveis podem ser condicionadas a uma outra variável em comum. Na Figura 5.8(b) está um exemplo gráfico do modelo probabilístico utilizado pelo COMIT. Para obter a estrutura em árvore, os autores empregaram o algoritmo *Maximum Weight Spanning Tree* (MWST) (Chow & Liu, 1968).

Já o *Bivariate Marginal Distribution Algorithm* (BMDA) (Pelikan & Mühlenbein, 1999) pode ser

considerado como uma extensão do COMIT. A diferença é que o modelo em árvore utilizado por ele permite que existam mais de um nó-raiz, como mostrado na Figura 5.8(c). Para criar as sub-árvores e decidir quais variáveis devem ser conectadas ou não, o algoritmo emprega o teste do Qui-quadrado.

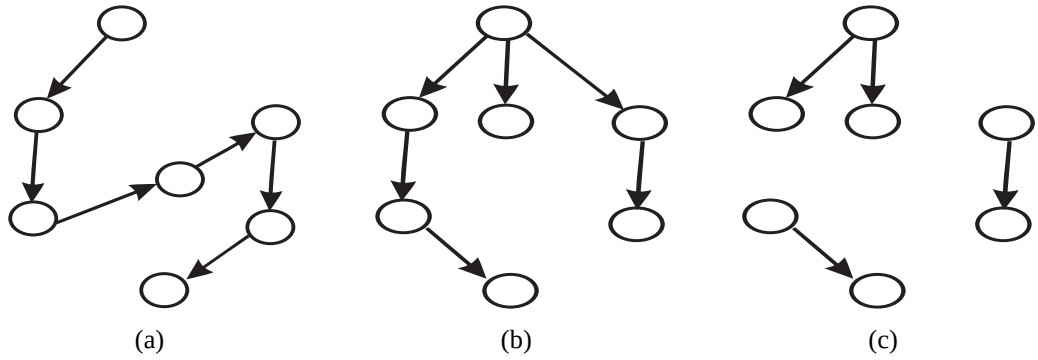


Figura 5.8: Modelos gráficos que consideram dependência entre pares de variáveis: (a) MIMIC, (b) COMIT e (c) BMDA.

5.9.3 Múltiplas dependências

Pelikan et al. (1999) mostraram que, em problemas complexos que possuem dependências entre múltiplas variáveis, os algoritmos pertencentes às duas categorias anteriores não são capazes de solucioná-los de forma satisfatória. Nesse sentido, várias propostas surgiram recentemente utilizando modelos capazes de expressar o relacionamento entre diversas variáveis. Como resultado, o modelo probabilístico é mais poderoso e sua obtenção torna-se mais complexa.

O *Extended Compact Genetic Algorithm* (ECGA) (Harik et al., 2006) é uma extensão do cGA. A idéia básica é agrupar as variáveis em grupos independentes e, então, aplicar o cGA nesses grupos (veja Figura5.9(a)). O algoritmo de agrupamento primeiramente considera a existência de *n* grupos, sendo uma variável em cada grupo. Em seguida, tenta-se unificar pares de grupos usando medidas de entropia e a métrica Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*). Problemas que podem ser decompostos em sub-problemas sem sobreposição são tratados de forma satisfatória pelo ECGA. Por outro lado, se existir sobreposição, o algoritmo falha, pois o problema não pode ser modelado de forma acurada pela simples divisão das variáveis em classes distintas.

Mühlenbein & Mahnig (1999) propuseram o *Factorized Distribution Algorithm* (FDA). O algoritmo utiliza uma estrutura em grafo definida previamente e que se mantém fixa durante todo o processo de otimização, sendo atualizados apenas os parâmetros numéricos (probabilidades). Portanto, é preciso ter conhecimento sobre o problema a ser tratado para se construir um grafo mais

adequado. O FDA utiliza a distribuição de Boltzmann para amostrar novos indivíduos. Um exemplo de grafo utilizado pelo FDA pode ser visto na Figura 5.9(b).

O *Bayesian Optimization Algorithm* (BOA) (Pelikan et al., 1999) utiliza rede bayesiana para modelar as dependências entre as variáveis, conforme ilustrado na Figura 5.9(c). A cada iteração, uma rede bayesiana é gerada a partir das melhores soluções, tentando refletir da melhor forma as relações entre as variáveis. Após gerada, a rede é utilizada para amostrar novas soluções candidatas. Para construir a rede bayesiana, BOA começa com um estrutura sem arcos e os vai adicionando até que não ocorra melhora na métrica utilizada para avaliar a qualidade do modelo. Usualmente, BOA emprega a métrica *Bayesian Dirichlet* (BD). O *real-coded Bayesian Optimization Algorithm* (rBOA) (Ahn et al., 2006) é uma extensão do BOA para otimização no espaço contínuo que utiliza modelo de mistura gaussiana.

Outro algoritmo que também utiliza redes bayesianas foi proposto posteriormente por Larrañaga et al. (2000a), o *Estimation of Bayesian Networks Algorithm* (EBNA). Seu funcionamento é similar ao BOA e diversas versões do algoritmo foram criadas com diferentes métricas para avaliar os modelos. Posteriormente, os autores desenvolveram a versão do algoritmo para lidar com otimização no espaço contínuo, denominada *Estimation of Gaussian Networks Algorithm* (EGNA) (Larrañaga et al., 2000b). EGNA utiliza redes gaussianas para modelar a dependência entre as variáveis.

Bosman & Thierens (2000) propuseram o *Iterated Density Evolutionary Algorithms* (IDEA), que é uma arquitetura geral de algoritmos para serem aplicados em problemas de otimização, tanto no domínio discreto quanto contínuo. No caso contínuo, os algoritmos utilizam funções densidade de probabilidade gaussianas multivariadas.

Uma proposta mais recente é o *Markov Network Estimation of Distribution Algorithms* (MN-EDA) (Santana, 2005), que utiliza rede de Markov como modelo probabilístico, como mostrado na Figura 5.9(d). Essa rede é gerada usando um procedimento estatístico conhecido como aproximação de Kikuchi (Kikuchi, 1951). Novas soluções são amostradas usando amostragem de Gibbs.

Dos algoritmos propostos nesta tese, BAIS e MOBAIS se assemelham muito com BOA e EBNA, enquanto GAIS e MOGAIS são mais parecidos com EGNA. Todos são capazes de expressar múltiplas dependências entre as variáveis por meio de modelos gráficos probabilísticos cuja estrutura é um grafo direcionado.

Embora BOA, EBNA e EGNA, junto com os outros EDAs, tenham sido aplicados de forma satisfatória na maioria dos problemas, eles apresentam algumas desvantagens quando comparados com sistemas imunológicos artificiais. Uma delas está relacionada com a perda de diversidade na população. Dado que tais algoritmos geram novas soluções com base na distribuição de probabilidade das melhores soluções selecionadas previamente, eles tendem a explorar o espaço de busca de uma

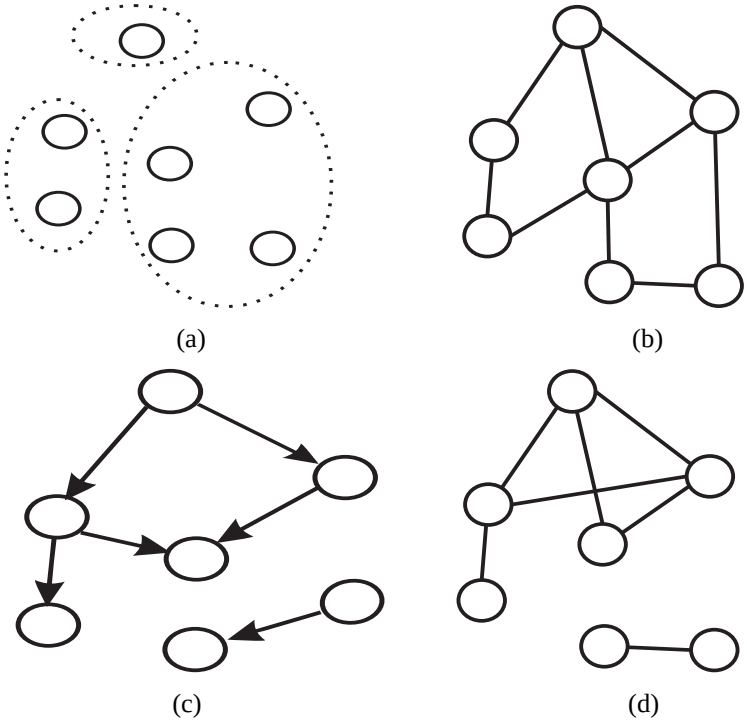


Figura 5.9: Modelos gráficos que consideram múltiplas dependências entre as variáveis: (a) ECGA, (b) FDA, (c) BOA e EBNA e (d) MN-EDA.

forma polarizada. O algoritmo tende a visitar uma região do espaço de busca já visitada anteriormente, levando-o a mínimos locais facilmente. Dessa forma, qualquer algoritmo de estimação de distribuição básico falha quando é aplicado para resolver problemas de otimização multimodal.

Apesar da incorporação de mecanismos para lidar com otimização multimodal nesses algoritmos (Sastry et al., 2005), eles ainda tendem a apresentar desempenho inferior quando comparado com os sistemas imunológicos artificiais, que são inerentemente capazes de lidar com problemas amplos, complexos e multimodais, mantendo diversidade na população.

5.10 Considerações Finais

Neste capítulo, foram descritas quatro novas propostas para que o sistema imunológico artificial seja capaz de manipular eficientemente blocos construtivos. A ideia fundamental é permitir que algoritmo construa um modelo probabilístico usando as melhores soluções e, posteriormente, utilizar este modelo para gerar novas soluções, as quais irão possuir características similares àquelas contidas no conjunto das melhores soluções.

Os quatro novos sistemas imunológicos artificiais, em vez de evoluirem as soluções candidatas por meio de clonagem e mutação, utilizam um modelo probabilístico que representa a distribuição

de probabilidade das melhores soluções encontradas até então. Os modelos probabilísticos adotados foram as redes bayesianas (caso discreto) e as redes gaussianas (caso contínuo), por apresentarem a capacidade de representar adequadamente interações complexas das variáveis do problema.

Por fim, os principais trabalhos relacionados foram apresentados e as quatro propostas desta tese foram devidamente posicionadas frente à literatura. Nos próximos três capítulos, essas novas propostas serão aplicadas junto a diversos problemas de interesse prático.

No Capítulo 6, o algoritmo BAIS é aplicado para a geração de *ensembles* de redes neurais. No Capítulo 7, são apresentados os experimentos realizados com os algoritmos BAIS e MOBAIS na tarefa de seleção de atributos em problemas de classificação de padrões. No Capítulo 8, são apresentados os experimentos realizados com os algoritmos GAIS e MOGAIS, no contexto de problemas de otimização no espaço contínuo.

Capítulo 6

Experimentos com *Ensembles* de Redes Neurais

6.1 Considerações Iniciais

Ensembles de redes neurais compreendem uma linha de pesquisa em aprendizado de máquina que visa combinar as saídas de duas ou mais redes neurais, as quais são treinadas separadamente e recebem o nome de componentes do *ensemble*, para formar a solução final do problema. A vantagem de combinar propostas alternativas de solução está no fato de que cada uma pode representar um aspecto diferente e, ao mesmo tempo, relevante para o problema. O bom desempenho de um *ensemble* depende da capacidade de generalização dos seus componentes e do nível de diversidade entre eles. É preciso que os erros cometidos por eles sejam descorrelacionados, uma vez que combinar modelos que generalizam de forma idêntica não traz benefícios.

Neste capítulo, são descritos os experimentos conduzidos para avaliar a viabilidade de BAIS quando aplicado à geração de *ensembles* de redes neurais. Espera-se que BAIS se mostre uma ferramenta eficaz para cumprir esta tarefa, uma vez que ele é capaz de produzir várias e diversas soluções de boa qualidade, bem como identificar e preservar as interações das variáveis. A aplicação é restrita a problemas de classificação e *ensembles* de redes neurais MLP (*Multilayer Perceptron*), embora problemas de regressão e outros tipos de componentes poderiam ser considerados com pequenas modificações no algoritmo.

Os conceitos básicos de *ensembles* e as etapas para sua geração são apresentados na Seção 6.2. Na Seção 6.3, é descrita a metodologia utilizada por BAIS para a construção dos *ensembles*. Em seguida, as características dos conjuntos de dados utilizados durante os experimentos são apresentados na Seção 6.4. Por fim, são apresentados e discutidos os resultados obtidos por BAIS e outros algoritmos da literatura.

6.2 *Ensemble* de Redes Neurais

Ensemble é um paradigma de aprendizado em que propostas alternativas capazes de gerar a solução para um determinado problema são combinadas para obter a solução final do problema (Hansen & Salamon, 1990; Krogh & Vedelsby, 1994). A essas propostas dá-se o nome de componentes do *ensemble*. Este paradigma originou-se do trabalho de Hansen & Salamon (1990) e foi aplicado primeiramente em problemas de classificação por meio de redes neurais. A combinação de múltiplos componentes é potencialmente vantajosa, pois diferentes componentes podem representar aspectos distintos e relevantes para a solução do problema. Várias redes neurais são treinadas independentemente e combinadas posteriormente para classificar os padrões de entrada. Os resultados mostram que a capacidade de generalização do sistema pode melhorar de forma significativa. Embora as redes neurais tenham sido as mais utilizadas como componentes de um *ensemble*, alguns trabalhos propuseram a combinação de outras técnicas, como sistemas *fuzzy* (Castro et al., 2005) e Máquinas de Vetores-Suporte (SVM, do inglês *Support Vector Machine*) (Zhang et al., 2005).

Na Figura 6.1, é apresentado um exemplo didático adaptado de Polikar (2006) que ilustra a vantagem do uso de *ensembles* para um problema de classificação de padrões. As fronteiras de decisão de três classificadores são apresentadas nas Figuras 6.1(a)(b)(c). Como pode ser observado, as três fronteiras de decisão são diferentes, o que leva os classificadores a acertarem e errarem a classificação de forma distinta. Entretanto, como mostrado na Figura 6.1(d), a combinação desses três classificadores, por meio do voto majoritário, possibilita gerar uma nova fronteira de decisão capaz de separar corretamente todas as amostras do conjunto de dados (Figura 6.1(e)). Apesar do *ensemble* da Figura 6.1(e) ter classificado corretamente todas as amostras, é importante ressaltar que, na prática, nem sempre os ganhos obtidos são tão expressivos.

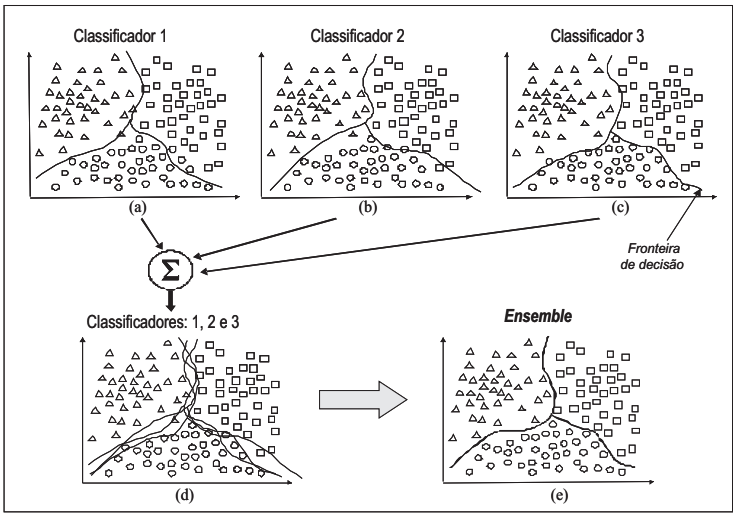


Figura 6.1: Exemplo ilustrativo do ganho de desempenho de um *ensemble* em problemas de classificação.

Técnicas de *ensemble* têm sido aplicadas com sucesso a problemas de classificação, regressão e agrupamento (*clustering*). Entretanto, neste capítulo serão considerados apenas problemas de classificação, em que os componentes do *ensemble* são redes neurais.

Na Figura 6.2 é ilustrado o esquema de funcionamento de um *ensemble* de redes neurais. Cada componente do *ensemble* é um classificador construído independentemente dos demais e pode atuar isoladamente. Para cada vetor de entrada $x = (x_1, x_2, ..., x_n)$, as saídas y_i, *i*=1,...,*M*, geradas pelos *M* componentes são combinadas para produzir a saída do *ensemble*, y_e.

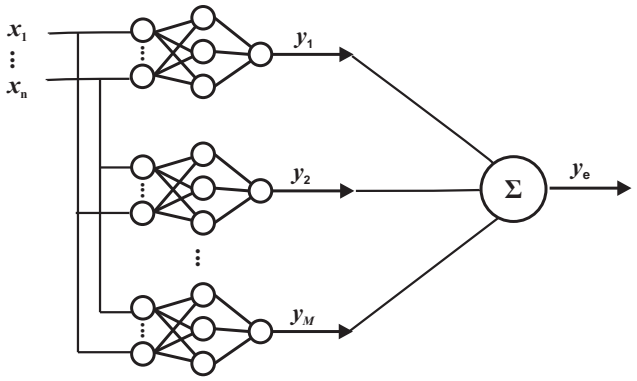


Figura 6.2: *Ensemble* de redes neurais.

O desempenho de um *ensemble* está vinculado à qualidade e diversidade do erro de seus componentes, isto é, cada componente do *ensemble* deve apresentar bom desempenho quando aplicado isoladamente e deve cometer erros distintos dos demais componentes para um mesmo padrão de entrada. É evidente que duas redes neurais de boa qualidade devem produzir a mesma saída para a maioria dos padrões. As redes devem, então, discordar quando elas tomam decisões erradas (Perrone & Cooper, 1993).

De forma geral, a construção de um *ensemble* envolve duas etapas sequenciais, a geração dos componentes candidatos e a combinação das saídas individuais de cada componente. Alguns trabalhos sugerem uma outra etapa, que consiste em selecionar, dentre os componentes gerados, somente alguns componentes. Cada etapa é brevemente descrita seguir:

- **Geração dos componentes candidatos**

Esta etapa se encarrega de gerar individualmente cada componente candidato a fazer parte do *ensemble*. Como descrito anteriormente, os componentes precisam apresentar bom desempenho e serem diversos entre si. Técnicas para gerar componentes diversos e de bom desempenho podem ser divididas em duas categorias: (*i*) geração pelo pré-processamento dos dados de treinamento e (*ii*) geração pelo pré-processamento dos parâmetros e estrutura dos componentes candidatos.

O objetivo das técnicas pertencentes à primeira categoria consiste em produzir dados de treinamento distintos, os quais podem levar à geração de componentes com capacidade de generalização diferente um dos outros. Entre essas técnicas, pode-se citar *Bagging* e *Boosting*. *Bagging* foi proposta por Breiman (1996) e é baseada na amostragem *bootstrap* (Efron & Tibshirani, 1993). Ela gera vários subconjuntos de dados de treinamento distintos a partir do conjunto original e então usa cada um desses conjuntos de dados para treinar uma rede neural.

Na técnica *Boosting*, proposta por Schapire (1990), os conjuntos de treinamento são reamostrados de forma adaptativa, de modo que amostras que mais contribuem para o erro de treinamento dos componentes já treinados têm sua probabilidade aumentada de comporem o conjunto de treinamento a ser empregado para a geração do próximo componente. Como se pode perceber, aqui a geração dos componentes é sequencial.

O pré-processamento dos parâmetros e estrutura, por sua vez, visa construir redes a partir de pontos de partida distintos. A abordagem mais comum é adotar redes neurais com diferentes topologias e diferentes valores para os vetores de pesos iniciais. Dessa forma, é esperado que diversas redes neurais sejam obtidas, uma vez que o desempenho da rede é altamente dependente do seu vetor de pesos iniciais e sua topologia.

- **Combinação**

Após a geração dos componentes do *ensemble*, o próximo passo é definir como as saídas individuais de cada um serão combinadas para formar a saída do *ensemble*. Existem muitas maneiras de se fazer isso, dependendo do tipo de problema abordado. Para tarefas de classificação, a abordagem mais usual é o voto majoritário (Hansen & Salamon, 1990), em que a classe indicada pela maioria dos componentes é a saída do *ensemble*, de acordo com a equação:

$$y_e(x) = k \quad se\ k = \arg\max_k \left(\sum_{i=1}^{M} G_i^k(x) \right), \tag{6.1}$$

em que x é o padrão de entrada, $y_e(.)$ é a saída do *ensemble*, M é a quantidade de componentes e G_i^k é dado pela Equação (6.2), com $y_i(.)$ sendo a saída do i-ésimo componente:

$$G_i^k(x) = \begin{cases} 1, & \text{se } y_i(x) = k \\ 0, & \text{se } y_i(x) \neq k. \end{cases} \tag{6.2}$$

- **Seleção dos componentes**

Recentemente, muitas propostas têm sugerido e provado empiricamente que adicionar todos os componentes gerados na etapa anterior no *ensemble* pode degradar seu desempenho, não importando

qual a técnica de combinação que está sendo empregada. Desta forma, é preciso realizar um processo de seleção dos componentes mais úteis para o *ensemble*, de acordo com um critério apropriado (Zhou et al., 2002).

O critério mais apropriado é o desempenho do *ensemble* como um todo. Nesse contexto, existem duas abordagens principais: construtiva e poda. Na abordagem construtiva, os candidatos a comporem o *ensemble* são ordenados de acordo com seus desempenhos individuais. Em seguida, o melhor candidato é considerado o primeiro componente do *ensemble*. O próximo candidato, que possui o segundo melhor desempenho, é inserido no *ensemble*. Se o desempenho do *ensemble* for melhorado, então ele passa a ter dois componentes agora. Caso contrário, o componente recém-inserido é removido. O mesmo procedimento é realizado considerando os candidatos restantes, um após o outro.

Na abordagem de poda, de partida, todos os candidatos são inseridos no *ensemble*. O candidato com pior desempenho, então, é removido. Se o desempenho do *ensemble* não melhorar, o candidato retorna para o *ensemble*. Este procedimento é realizado com os candidatos restantes, em ordem crescente de desempenho individual.

As abordagens construtiva e de poda requerem um alto custo computacional quando existem muitos candidatos. Por isso, alguns trabalhos sugerem o emprego de meta-heurísticas para realizar esta tarefa.

Dados os requisitos impostos para a construção de um *ensemble* (componentes diversos e com bom desempenho), fica claro que as etapas de geração e seleção exercem um papel fundamental para o sucesso do *ensemble*. Recentemente, muitos trabalhos têm sido propostos para realizar essas tarefas, particularmente no campo da computação bio-inspirada (Coelho & Von Zuben, 2006; García-Pedrajas & Fyfe, 2007; García-Pedrajas et al., 2005; Liu et al., 2001, 2000; Opitz & Shavlik, 1996; **?**; Zhang et al., 2005; Zhou et al., 2002). Algoritmos bio-inspirados são estratégias de busca capazes de lidar com espaços de buscas amplos e complexos, evitando mínimos locais.

Dentre os algoritmos bio-inspirados, os sistemas imunológicos artificiais têm sido utilizados com êxito e melhores resultados são obtidos quando comparados com algoritmos genéticos, por exemplo (Castro et al., 2005; Coelho & Von Zuben, 2006; García-Pedrajas & Fyfe, 2007; Pasti & de Castro, 2009, 2007).

6.3 BAIS Aplicado na Geração de *Ensembles* de Redes Neurais

O objetivo desta seção é apresentar BAIS sendo utilizado para treinar redes neurais e selecionar as mais efetivas para compor o *ensemble*. É esperado que BAIS se mostre uma ferramenta eficaz para cumprir estas tarefas, umas vez que ele é capaz de produzir várias e diversas soluções de boa

qualidade (característica desejada durante a etapa de geração dos componentes), bem como ele pode identificar e preservar as interações das variáveis (conhecimento bastante relevante a ser explorado durante as etapas de geração e seleção).

Diferentemente de abordagens coevolutivas, que simultaneamente evoluem as redes neurais e realizam a seleção por meio de duas populações, a abordagem aqui proposta realiza estas tarefas separadamente. Desta forma, BAIS é executado de forma sequencial e, para cada etapa, são adotadas diferentes formas de codificação, função de aptidão e supressão, conforme será explicado nas Seções 6.3.1 e 6.3.2.

Antes de dar início à execução do algoritmo, o conjunto de dados é particionado em três subconjuntos: treinamento, seleção e teste. A partir dos dados de treinamento, BAIS gera as redes neurais candidatas. Tão logo esta etapa termina, BAIS é aplicado novamente utilizando o subconjunto de seleção para escolher quais as redes que farão parte do *ensemble*. Finalmente, o desempenho do *ensemble* é verificado usando o subconjunto de teste. A aplicação é restrita a problemas de classificação e *ensembles* de redes neurais MLP (*Multilayer Perceptron*), embora problemas de regressão e outros tipos de componentes poderiam ser considerados com pequenas modificações no algoritmo.

6.3.1 Aprendizado das redes neurais

Nesta primeira etapa, BAIS é responsável por evoluir as topologias das MLPs mais apropriadas para um dado problema, enquanto o vetor de pesos será ajustado por um algoritmo de segunda ordem, especificamente o gradiente conjugado escalonado (Moller, 1993).

Muitas das metodologias propostas se concentram no aprendizado de redes neurais considerando apenas funções de ativação pré-determinadas e do mesmo tipo (Yao, 1999). Entretanto, se uma função de ativação adequada para cada neurônio puder ser definida automaticamente, melhores taxas de acerto podem ser obtidas (Liu & Yao, 1996; Iyoda & Von Zuben, 2002). Nesse trabalho, não somente o número de neurônios na camada intermediária é ajustado, mas também o tipo de função de ativação para cada neurônio. Dessa forma, espera-se que as redes neurais geradas possuam capacidades distintas de generalização.

Em seguida, a codificação, a função de aptidão e o mecanismo de supressão do algoritmo serão detalhados.

- **Codificação**

Cada rede candidata é um anticorpo enquanto o subconjunto de treinamento representa um antígeno. As topologias das redes são restritas a possuírem apenas uma camada intermediária

completamente conectada. A primeira e a última camada são constantes, uma vez que elas são definidas usando o número de variáveis de entrada e de saída do problema, respectivamente.

A camada intermediária possui, no mínimo, um neurônio e, no máximo, F neurônios com funções de ativação possivelmente distintas. As funções de ativação são escolhidas a partir de um conjunto de funções candidatas, apresentadas na Tabela 6.1. O uso da função nula, f_0, permite reduzir o número de neurônios na camada intermediária. A justificativa para adotar as funções de ativação candidatas presentes na Tabela 6.1 está no fato de estas funções serem frequentemente adotadas na literatura de redes neurais artificiais, particularmente no caso de redes neurais MLP.

Tabela 6.1: Possíveis funções de ativação.

Identificador	Função	Expressão
f_0	Função nula	y=0
f_1	Tangente hiperbólica	y=tanh(x)
f_2	Cosseno	y=cos(x)
f_3	Seno	y=sin(x)
f_4	Seno hiperbólico	y=sinh(x)

Os anticorpos são codificados com números inteiros, denotando o índice da função de ativação. Por exemplo, suponha que para um determinado problema o número máximo de neurônios na camada intermediária seja cinco, isto é, F=5. Na Figura 6.3 é apresentado um possível anticorpo para este caso. A rede neural representada por ele está ilustrada na Figura 6.4.

Figura 6.3: Exemplo de um anticorpo codificando uma rede neural.

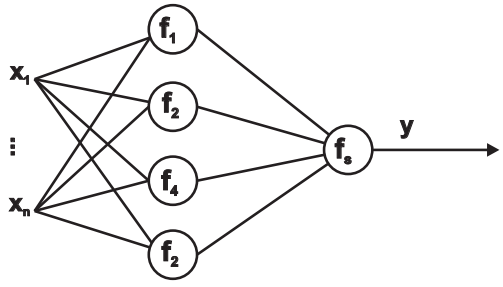


Figura 6.4: Rede neural com topologia representada pelo anticorpo da Figura 6.3.

Dado que a segunda função de ativação do anticorpo da Figura 6.3 é a função nula, a rede é composta apenas de quatro neurônios na camada intermediária.

A função de ativação do neurônio de saída, f_s, é sempre a função logística e, por isso, não faz parte do anticorpo. Da mesma forma, os vetores de peso não são considerados na codificação porque eles são ajustados empregando-se uma técnica de otimização não-linear denominada gradiente conjugado escalonado (Moller, 1993).

- **População inicial**

Para iniciar a população, todas as funções de ativação possuem a mesma probabilidade de serem escolhidas, exceto a função nula que possui probabilidade maior. Assim, espera-se que redes mais parcimoniosas sejam encontradas.

- **Função de aptidão**

A função de aptidão é definida com base no desempenho das redes neurais quando aplicadas junto ao conjunto de treinamento. Ela é expressa por:

$$Fit(Ab_i) = \frac{NPCC(Ab_i)}{NTP} \tag{6.3}$$

em que *NPCC(*Ab_i*)* é o número de padrões corretamente classificados pela rede neural codificada no anticorpo Ab_i e *NTP* é o número total de padrões no conjunto de treinamento.

- **Supressão**

Nessa fase, anticorpos similares são eliminados da população para evitar redundância e, assim, manter diversidade. O grau de similaridade entre as redes neurais que estão sendo evoluídas não é expresso baseado no vetor de pesos e nem na topologia, mas sim no comportamento de entrada-saída de cada rede. Assim, não ocorre o problema de permutação (Angeline et al., 1994), o qual é caracterizado pelo fato de que a ordem de aparecimento dos neurônios na camada intermediária não afeta o comportamento de entrada-saída da rede neural.

Para tanto, a medida de discordância (*disagreement measure*) (Skalak, 1996) foi aplicada aos pares de classificadores. Para dois classificadores, Ab_i e Ab_m, a medida de discordância é calculada como:

$$Dis_{i,m} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{6.4}$$

em que N^{ab} é o número de instâncias para as quais a saída da rede Ab_i é igual a *a* e a saída da rede Ab_j é igual a *b*, conforme ilustrado na Tabela 6.2. Repare que o denominador é igual ao número total de instâncias. Este índice de discordância varia de 0 a 1, indicando que os classificadores são similares (valor perto de 0) ou diversos (valor perto de 1).

Tabela 6.2: Saídas para dois classificadores de índices i e m.

	Ab_m correcto	Ab_m incorreto
Ab_i correto	N^{11}	N^{10}
Ab_i incorrecto	N^{01}	N^{00}

Indivíduos que apresentam entre si um grau de similaridade abaixo de um certo limiar serão eliminados da população, sendo mantido apenas aquele dentre eles com maior valor de aptidão.

A diversidade total para *L* classificadores é dada por:

$$Dis = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{m=i+1}^{L} Dis_{i,m} \tag{6.5}$$

A Equação (6.5) será utilizada para medir a diversidade do *ensemble* como um todo, conforme será descrito na Seção 6.4.

6.3.2 Seleção de componentes

Tão logo termine o processo de geração das redes neurais, BAIS é aplicado novamente para encontrar vários subconjuntos de redes neurais, os quais representam os *ensembles*. Este é um problema de otimização combinatória e certamente possui muitos blocos construtivos.

No que segue, a codificação, a função de aptidão e o mecanismo de supressão utilizados no algoritmo dessa segunda etapa serão detalhados.

- **Codificação**

Nesta etapa, cada anticorpo $Ab_i = (net_{i1}, net_{i2}, ..., net_{ik})$ é codificado como uma sequência binária de tamanho k, sendo k o número de redes neurais geradas na etapa anterior. Cada posição do vetor, net_{ij}, está associada a uma rede neural. Se net_{ij}=1, então o *ensemble i* irá conter a rede neural *j*. Note que cada anticorpo representa de fato um subconjunto específico de classificadores.

- **População inicial**

Nesta etapa, um anticorpo é gerado contendo todas as redes neurais obtidas, isto é, este anticorpo é um vetor contendo o bit “1” em todas as posições. Os demais anticorpos são gerados aleatoriamente, com igual probabilidade para os bits “0” e “1”.

- **Função de aptidão**

A função de aptidão utilizada agora leva em consideração o desempenho do *ensemble* associado ao anticorpo, aplicado junto ao conjunto de dados de seleção. Se o *ensemble* possui um número par de componentes e ocorrer um empate, o melhor classificador, com relação aos dados de treinamento, dará o voto de desempate.

- **Supressão**

Neste estágio, a distância de Hamming é utilizada para calcular o grau de similaridade entre os anticorpos. Indivíduos com um grau de similaridade acima de um certo limiar serão eliminados da população.

6.4 Experimentos Computacionais

Nesta seção, são descritos os experimentos realizados para avaliar a metodologia proposta. Nas subseções seguintes, serão apresentados os conjuntos de dados testados, os parâmetros utilizados e os resultados obtidos.

6.4.1 Descrição dos conjuntos de dados

Foram considerados oito problemas de classificação, obtidos do *UCI Repository of Machine Learning Databases* (Asunción & Newman, 2007): WDBC, Bupa, Ionosphere, Pima, Sonar, Card, Vote e Hepatitis. Na Tabela 6.3, são apresentadas as características de cada um deles, com o número total de instâncias, número de atributos e número de classes.

Tabela 6.3: Características dos conjuntos de dados.

Conjunto de dados	# Instâncias	# Atributos	# Classes
WDBC	569	30	2
Bupa	345	6	2
Ionosphere	350	34	2
Pima	768	8	2
Sonar	208	60	2
Card	690	15	2
Vote	435	16	2
Hepatitis	155	19	2

Estes conjuntos de dados são amplamente utilizados na literatura e, assim, é possível comparar o desempenho de nossa metodologia com outras existentes.

Os conjuntos de dados foram particionados aleatoriamente da seguinte forma: 50% para treinamento, 25% para seleção e 25% para teste. Este particionamento foi realizado 30 vezes e os resultados a serem apresentados nas próximas subseções são as médias dos resultados obtidos junto ao conjunto de teste nas 30 execuções do algoritmo.

6.4.2 Parâmetros

Os parâmetros usados nos experimentos são os mesmos para os oito conjuntos de dados. A população inicial contém 80 anticorpos e o critério de parada é o número máximo de gerações, definido como 50. O número máximo de neurônios na camada intermediária, *F*, foi definido como 10. O ajuste dos pesos é realizado pelo gradiente conjugado escalonado (Moller, 1993).

A síntese da rede bayesiana utilizada pelo BAIS é realizada conforme a descrição realizada na Seção 5.3. No intuito de penalizar modelos complexos, uma restrição com relação ao número de pais que os nós da rede bayesiana podem ter foi estabelecida. Cada nó só pode ter, no máximo, dois pais. Uma vez que a rede bayesiana foi gerada, novas soluções candidatas são geradas. Em cada geração, o número de amostras geradas é metade do tamanho da população atual. Por exemplo, se a população total contém N anticorpos, o algoritmo irá amostrar $N/2$ indivíduos. A quantidade de indivíduos inseridos aleatoriamente para promover diversidade é 3% do tamanho da população atual.

6.4.3 Desempenho individual das redes neurais

Primeiramente, foi investigada a capacidade do algoritmo de gerar redes neurais que possuíssem bom desempenho e que fossem diversas entre si. Na Tabela 6.4, é apresentada a média dos resultados obtidos junto ao conjunto de teste para a melhor rede neural em 30 execuções do algoritmo. A quarta coluna da refere-se ao número médio de neurônios na camada intermediária.

Conforme pode ser observado na Tabela 6.4, o algoritmo foi apto a gerar redes neurais parcimoniosas com poucos neurônios na camada intermediária e boa capacidade de generalização. Isso se deve ao uso de diferentes funções de ativação para cada neurônio aliado ao fato de BAIS conseguir identificar e preservar os blocos construtivos. Em seguida, são apresentadas representações gráficas para algumas redes neurais obtidas. Essas são as melhores redes geradas ao longo das 30 execuções do algoritmo. Nas Figuras 6.5, 6.6 e 6.7 estão as redes neurais obtidas para os conjuntos de dados Pima, Bupa e Ionosphere, respectivamente.

Foram verificadas também as funções de ativação que aparecem com mais frequência. Nas redes neurais para Bupa, Pima, Ionosphere e Card, a tangente hiperbólica aparece regularmente. Para

Tabela 6.4: Desempenho médio da melhor rede neural em 30 execuções do algoritmo, considerando o conjunto de teste.

Conjunto de dados	Taxa de classificação (%)	# de neurônios
WDBC	$98{,}2 \pm 0{,}6$	5,2
Bupa	$73{,}8 \pm 1{,}7$	3,9
Ionosphere	$93{,}8 \pm 1{,}3$	5,2
Pima	$79{,}1 \pm 1{,}4$	3,6
Sonar	$84{,}7 \pm 2{,}1$	6,1
Card	$83{,}4 \pm 1{,}2$	5,7
Vote	$97{,}7 \pm 0{,}9$	7,3
Hepatitis	$82{,}5 \pm 1{,}2$	8,5

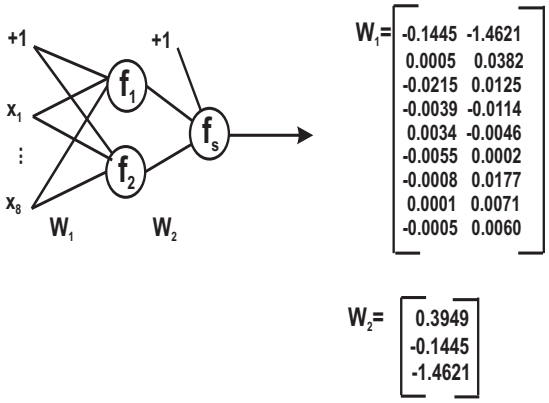


Figura 6.5: Melhor rede neural para o conjunto de dados Pima.

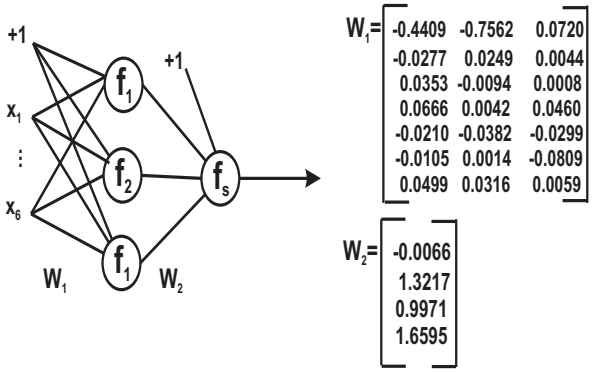


Figura 6.6: Melhor rede neural para o conjunto de dados Bupa.

WDBC, Hepatitis e Vote, a função cosseno é a mais comum. A função seno aparece frequentemente para o Sonar. Já a função seno hiperbólico é a função que menos aparece, considerando todos os

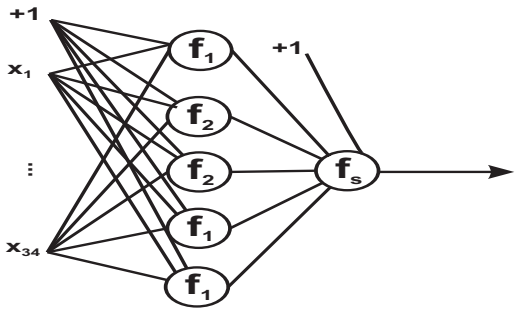


Figura 6.7: Melhor rede neural para o conjunto de dados Ionosphere. Não foram apresentados os pesos sinápticos devido ao elevado número de entradas.

conjuntos de dados.

6.4.4 Desempenho dos *ensembles*

Conforme apresentado na seção anterior, múltiplas redes neurais de boa qualidade foram geradas em cada execução do algoritmo. Agora, BAIS pode selecionar algumas delas para compor o *ensemble*, visando obter uma melhora na taxa de classificação. O método utilizado para combinar as saídas individuais das redes neurais foi o voto majoritário, explicado na Seção 6.2.

O desempenho médio dos *ensembles* sobre 30 execuções do algoritmo pode ser visto na Tabela 6.5. O número médio de componentes no ensemble é apresentado na quarta coluna da tabela. Na última coluna, é mostrado o ganho percentual em desempenho sobre a melhor rede neural.

Tabela 6.5: Desempenho médio do *ensemble* em 30 execuções.

Conjunto de dados	Taxa de classificação (%)	Componentes gerados	Componentes selecionados	Ganho
WDBC	$98{,}7 \pm 1{,}1$	8,6	3,4	0,5
Bupa	$76{,}2 \pm 2{,}1$	15,2	9,7	2,4
Ionosphere	$97{,}1 \pm 2{,}3$	19,7	10,7	3,3
Pima	$81{,}8 \pm 1{,}7$	12,1	7,3	2,7
Sonar	$86{,}4 \pm 2{,}9$	17,4	12,8	1,7
Card	$87{,}6 \pm 0{,}9$	16,9	11,6	4,2
Vote	$96{,}2 \pm 1{,}6$	11,3	7,7	0,9
Hepatitis	$84{,}1 \pm 2{,}2$	13,7	6,9	1,6

Observando a Tabela 6.5, pode-se perceber que o ensemble de redes neurais melhorou a taxa de classificação, superando a melhor rede neural em todos os casos (veja Tabela 6.4). Assim como para os resultados com redes individuais, os resultados dos *ensembles* também foram obtidos empregando

o conjunto de teste. Desta forma, não se conhece a *priori* qual é a melhor rede neural individual, o que torna os resultados obtidos ainda mais expressivos.

O pequeno ganho em desempenho obtido para o conjunto de dados WDBC provavelmente vem do fato dele ser um problema bastante fácil e, como conseqüência, todas as redes neurais geradas pelo BAIS no primeiro estágio possuem capacidade de generalização muito boa mesmo quando aplicadas individualmente. Com isso, o grau de diversidade entre os componentes é baixo (veja Tabela 6.6) e o uso de um *ensemble* com estas características não contribui para melhorar a taxa de classificação.

O grau de diversidade entre os componentes dos *ensembles* foi calculado usando a Equação (6.5). Na Tabela 6.6, o grau de diversidade médio é apresentado. Como esperado, pode-se ver que a metodologia obteve bons níveis de diversidade entre os componentes dos *ensembles*.

Tabela 6.6: Grau de diversidade entre os componentes dos *ensembles*.

Conjunto de dados	Grau de diversidade
WDBC	0,08
Bupa	0,28
Ionosphere	0,24
Pima	0,29
Sonar	0,35
Card	0,31
Vote	0,33
Hepatitis	0,26

6.4.5 Resultados comparativos

Nesta seção, são apresentados os resultados comparativos entre BAIS e mais quatro algoritmos bio-inspirados: o *Evolutionary Ensembles with Negative Correlation Learning* (EENCL), um algoritmo coevolutivo, o *Bayesian Optimization Algorithm* (BOA) e um sistema imunológico artificial (SIA).

O algoritmo EENCL (Liu et al., 2000) evolui redes neurais MLP usando programação evolutiva. O número de neurônios na camada intermediária é especificado pelo usuário. Para manter diversidade na população, *fitness sharing* e aprendizado por correlação negativa foram utilizados para forçar a formação de espécies diferentes. Essas espécies são determinadas usando o algoritmo *k-means* para agrupar os indivíduos da população. Os grupos resultantes correspondem a diferentes espécies. O melhor indivíduo de cada espécie é escolhido para compor o *ensemble*. Os parâmetros utilizados são os mesmos adotados pelos proponentes do algoritmo: tamanho da população: 25, número máximo

de gerações: 200, número de grupos que o *k-means* gera: 25, número de neurônios na camada intermediária: 10.

O algoritmo coevolutivo (García-Pedrajas et al., 2005) evolui duas populações separadamente. Uma população é responsável por evoluir os vetores de pesos e a topologia de MLPs usando programação evolutiva. A outra população utiliza algoritmo genético para evoluir *ensembles* formados por subconjuntos de elementos da primeira população. A população de MLPs contém 30 indivíduos e a de *ensembles* contém 100, cada um representando um *ensemble* composto de 10 MLPs.

O algoritmo BOA (Pelikan & Mühlenbein, 1999), já descrito no Capítulo 5, é um algoritmo de estimação de distribuição que utiliza redes bayesianas como modelo probabilístico. Assim como BAIS, as redes bayesianas são geradas pelo algoritmo K2. O tamanho da população foi definido como 200 e o número máximo de gerações como 100.

Finalmente, foi implementado um sistema imunológico artificial (SIA) baseado na seleção clonal e teoria da rede imunológica, seguindo o pseudo-código do Algoritmo 3.1. A população inicial contém 50 anticorpos e o número máximo de gerações foi definido como 200. Cada anticorpo dá origem a 2 clones.

Para realizar uma comparação mais justa, BOA e o algoritmo imunológico evoluem também os tipos de função de ativação para cada neurônio, além do número de neurônios na camada intermediária. O número máximo de neurônios na camada intermediária foi definido como 10 para os três algoritmos (BAIS, BOA e SIA).

Na Tabela 6.7 são apresentadas as médias dos resultados obtidos em 30 execuções de cada algoritmo. O número entre parênteses indica o número médio de componentes no *ensemble*.

Tabela 6.7: Taxa de classificação média ao longo de 30 execuções. O número entre parênteses indica o número médio de componentes no *ensemble*.

	BAIS	EENCL	Coevolução	BOA	SIA
WDBC	98,7 (3,4)	97,2 (9,6)	96,6 (10)	98,1 (11,3)	97,7 (7,5)
BUPA	76,2 (9,7)	75,7 (17,8)	76,1 (10)	74,4 (12,5)	75,3 (8,2)
Ionosphere	97,1 (10,7)	94,5 (23,4)	93,8 (10)	94,9 (10,4)	95,6 (9,6)
Pima	81,8 (7,3)	77,6 (16,7)	78,3 (10)	78,4 (13,1)	79,2 (11,3)
Sonar	86,4 (12,8)	86,3 (19,2)	79,6 (10)	84,9 (11,8)	85,7 (14,8)
Card	87,6 (11,6)	86,1 (13,4)	86,2 (10)	84,5 (15,2)	87,2 (10,6)
Vote	96,3 (7,7)	94,7 (14,3)	93,4 (10)	93,7 (14,6)	95,9 (12,3)
Hepatitis	84,1 (6,9)	83,8 (11,6)	82,5 (10)	81,8 (9,7)	82,8 (10,4)

Pela Tabela 6.7, pode-se perceber que os *ensembles* gerados por BAIS possuem melhor

desempenho para todos os conjuntos de dados. Outro fato a ser destacado é o baixo número de componentes nos *ensembles* gerados por BAIS.

Quando comparado com EENCL e o algoritmo coevolutivo, BAIS pode ter gerado melhores resultados em função da sua capacidade de evoluir também o tipo de função de ativação para cada neurônio e também devido à sua capacidade de identificar e preservar os blocos construtivos.

Quando comparado com BOA e com o SIA, que também evoluem o tipo de função de ativação, BAIS produziu melhores resultados devido ao seu eficiente mecanismo de explorar o espaço de busca. O SIA, com seu operador de mutação, não pode identificar o relacionamento entre as variáveis, sendo, assim, suscetível a romper soluções parciais encontradas ocasionalmente. Por outro lado, BOA possui a capacidade de lidar com blocos construtivos, mas existe uma limitação com relação à manutenção de diversidade na população.

Da Figura 6.8 à Figura 6.15, são apresentados os *boxplots* comparando os cinco algoritmos para cada um dos conjuntos de dados.

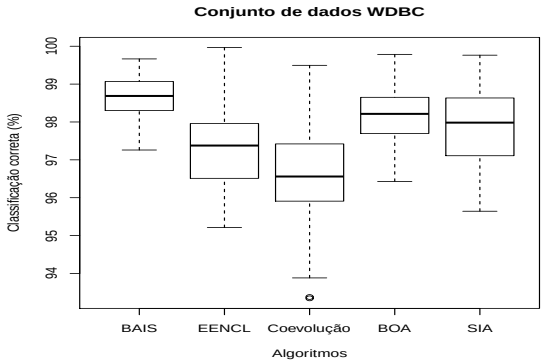


Figura 6.8: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados WDBC.

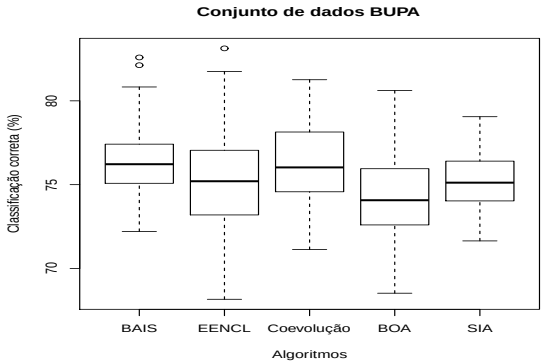


Figura 6.9: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Bupa.

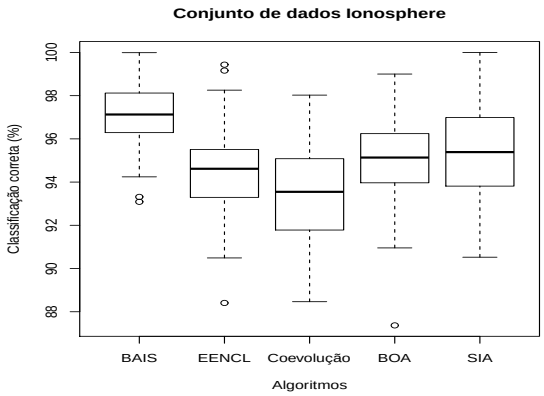


Figura 6.10: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Ionosphere.

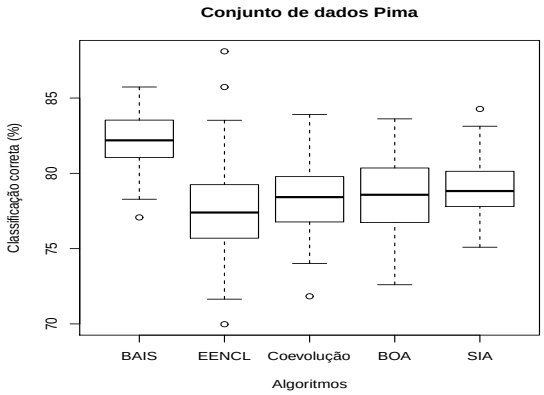


Figura 6.11: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Pima.

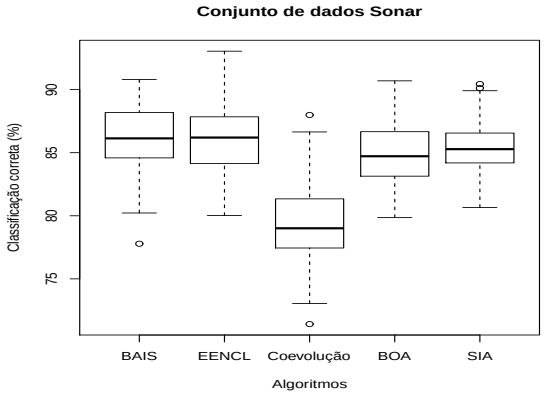


Figura 6.12: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Sonar.

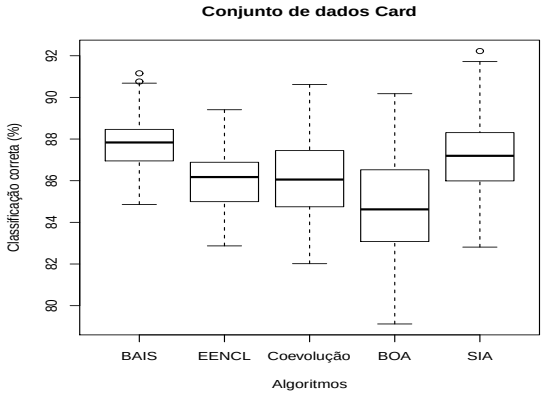


Figura 6.13: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Card.

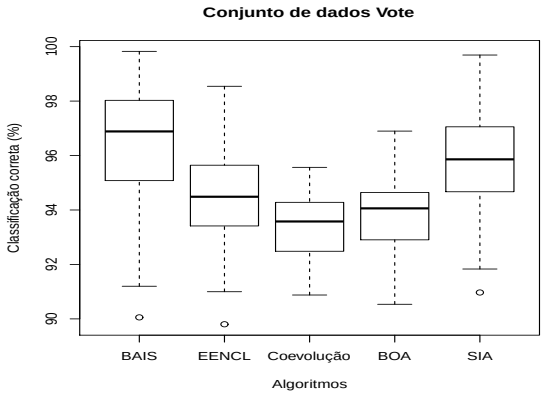


Figura 6.14: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Vote.

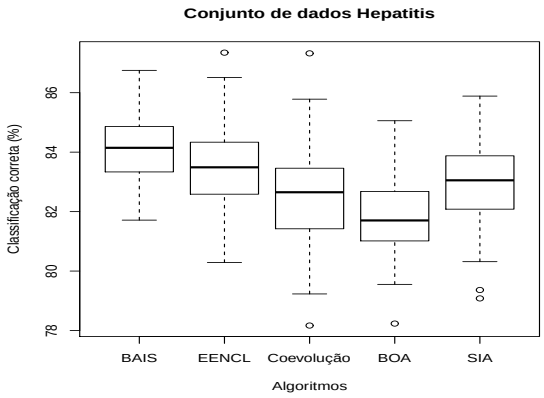


Figura 6.15: *Bloxpot* para os cinco algoritmos junto ao conjunto de dados Hepatitis.

6.5 Considerações Finais

Este capítulo teve por finalidade avaliar o desempenho de BAIS ao aplicá-lo na tarefa de geração de *ensembles* de redes neurais.

Primeiramente, foram apresentados os conceitos básicos de *ensembles* e as etapas para a sua construção. Em seguida, foi descrita a metodologia empregada por BAIS para gerar *ensembles* de redes neurais MLP para problemas de classificação. O algoritmo é executado duas vezes, de forma sequencial. Na primeira execução, ele gera as redes neurais candidatas a compor o *ensemble*. Na segunda execução, BAIS seleciona, dentre as soluções geradas na primeira etapa, um subconjunto de redes neurais para compor de fato o *ensemble*. Para cada etapa, foram adotadas diferentes formas de codificação, função de avaliação e supressão.

Os resultados mostraram que BAIS foi capaz de gerar um conjunto de classificadores de bom desempenho e, ao mesmo tempo, diversos entre si. Isso se deve à otimização não apenas do número de neurônios na camada intermediária, mas também da função de ativação mais adequada para cada neurônio. Quando o algoritmo foi executado para selecionar quais classificadores iriam compor o *ensemble*, soluções acuradas e parcimoniosas foram obtidas.

Quando comparado com outros algoritmos da literatura, BAIS apresentou desempenho igual ou superior para a maioria dos casos.

Capítulo 7

Experimentos com Seleção de Atributos

7.1 Considerações Iniciais

Neste capítulo, são apresentados os experimentos realizados com os algoritmos BAIS e MOBAIS, aplicados para a tarefa de seleção de atributos em problemas de classificação de padrões.

A seleção de atributos é um problema de busca e otimização multimodal que apresenta vários blocos construtivos. Por exemplo, considere um conjunto de dados com 10 atributos. Suponha também que foi verificado que qualquer subconjunto que possuir os atributos 3, 4 e 7, dentre outros, é tido como boa solução. Neste caso, a presença dos atributos 3, 4 e 7 no subconjunto forma um bloco construtivo (solução parcial). Como um segundo exemplo de bloco construtivo, pode-se constatar que sempre que os atributos 1 e 9 estão presentes, o atributo 2 não está.

Desta forma, espera-se que BAIS e MOBAIS produzam bons resultados para este tipo de tarefa, já que ambos combinam as vantagens dos sistemas imunológicos artificiais (eficiente mecanismo para explorar o espaço de busca) com as vantagens dos algoritmos de estimação de distribuição (eficiente mecanismo para lidar com blocos construtivos).

Dado que BAIS e MOBAIS possuem finalidades diferentes no que se refere à quantidade de funções-objetivo a serem otimizadas, os experimentos são abordados de duas formas. Num primeiro momento, a seleção de atributos é vista como um problema de otimização mono-objetivo, em que BAIS procura soluções que maximizem a acuidade de um classificador. De outra forma, a seleção de atributos é encarada como um problema de otimização multiobjetivo, com MOBAIS procurando gerar soluções que maximizem a acuidade do classificador e, ao mesmo tempo, minimize o número de atributos selecionados.

Essa separação na forma de apresentar os experimentos é justificada pelo intuito de evidenciar as potencialidades de cada algoritmo e facilitar a apresentação dos resultados e a comparação com outras propostas.

Primeiramente, os conceitos básicos de seleção de atributos e as três principais abordagens para realizar esta tarefa são introduzidos na Seção 7.2. Em seguida, na Subseção 7.3.1 são descritas as características dos conjuntos de dados utilizados ao longo dos experimentos. Na Subseção 7.3.2, são apresentados e discutidos os resultados obtidos por BAIS. Por fim, os resultados obtidos por MOBAIS encontram-se na Subseção 7.3.2.

7.2 Conceitos Básicos em Seleção de Atributos

Para manipular eficientemente conjuntos de dados volumosos, em que as instâncias são descritas por muitos atributos, é conveniente realizar um pré-processamento dos dados para identificar e remover os atributos irrelevantes e redundantes que possivelmente possam existir no conjunto de dados original. A esta tarefa dá-se o nome de seleção de atributos.

As técnicas de seleção de atributos permitem reduzir a quantidade de atributos necessários para representar os dados, possivelmente conduzindo a uma melhor visualização e interpretação destes. Quando o conjunto de dados contendo apenas os atributos relevantes é submetido a um algoritmo de aprendizado de máquina (treinamento de classificadores ou regressores, por exemplo), este algoritmo consegue atuar de forma mais rápida e eficiente. Mais ainda, é possível ocorrer um ganho de desempenho e compreensibilidade do modelo induzido.

A tarefa de selecionar um subconjunto de atributos relevantes pode ser modelado como um problema de busca e otimização, em que o espaço de busca contém todas as possíveis combinações de atributos. Na Figura 7.1, é ilustrado um exemplo de espaço de busca para um problema com três atributos, adaptado de Blum & Langley (1997). Todas as possíveis combinações de atributos para formarem subconjuntos denotam o espaço de busca, representado na figura pelos oito retângulos. Os atributos são representados por círculos. Círculos pintados indicam que os atributos associados estão incluídos no subconjunto.

De acordo com Blum & Langley (1997), os métodos para seleção de atributos precisam se concentrar em três componentes: ponto de partida no espaço de busca, estratégia de busca e métrica de avaliação.

O ponto de partida determina a região inicial de exploração do espaço de busca. É possível começar a busca com um conjunto vazio de atributos e adicioná-los sucessivamente até preencher todo o conjunto (seleção *forward*). Outra possibilidade é começar com um conjunto contendo todos os atributos e removê-los até chegar no conjunto vazio (seleção *backward*). Existem ainda a busca bidirecional, a qual parte de ambas as direções, e a busca aleatória, que escolhe a ação (adição ou remoção) aleatoriamente.

A estratégia de busca define como o espaço de busca será percorrido. A busca exaustiva avalia

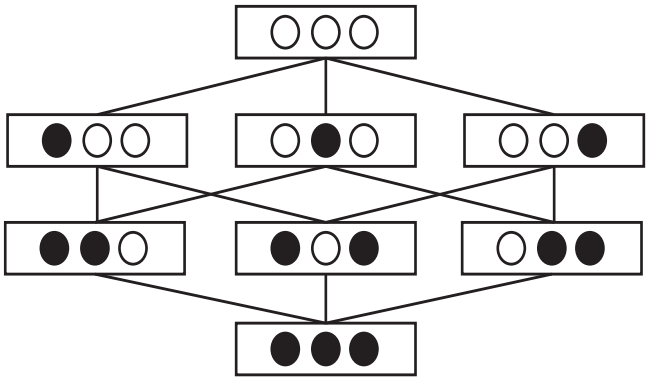


Figura 7.1: Exemplo de espaço de busca (8 subconjuntos) para um conjunto de 3 atributos. Círculo pintado indica que o atributo está incluído no subconjunto.

todos os possíveis subconjuntos de atributos do espaço de busca e apresenta aquele com melhor desempenho na avaliação. Essa estratégia garante encontrar o subconjunto ótimo, mas pode ser muito custosa se o número de atributos for grande. Nessas situações, o interessante é empregar estratégias de busca que utilizam heurísticas, as quais apontam subconjuntos de atributos mais promissores para serem avaliados seguindo alguma regra determinística. Dessa forma, evita-se a avaliação de todos os possíveis subconjuntos de atributos. Essa estratégia é mais rápida, mas não garante encontrar o subconjunto ótimo. Por fim, as estratégias que incluem alguma etapa de decisão não-determinística, diferentemente das duas outras, escolhe com graus variados de aleatoriedade qual será o próximo subconjunto a ser avaliado. Isso implica que múltiplas execuções desse tipo de busca não geram, necessariamente, as mesmas soluções. A principal motivação para seu uso é fazer com que o algoritmo escape de mínimos locais. Exemplos de algoritmos de busca com etapas de decisão não-determinísticas utilizados com sucesso para seleção de atributos são: os algoritmos genéticos (Siedlecki & Sklansky, 1989), sistemas imunológicos artificiais (Castro & Von Zuben, 2008a) e algoritmos de estimação de distribuição (Inza et al., 2000; Cantú-Paz, 2002).

A métrica de avaliação é utilizada para medir quão bom é cada subconjunto de atributos explorado durante a busca. O conceito de "subconjunto-bom de atributos" varia conforme a aplicação. Usualmente, as métricas avaliam se um subconjunto ajuda a melhorar o desempenho de um modelo (classificador ou regressor) e/ou se o subconjunto ajuda a deixar o conjunto de dados mais compacto e compreensível, sem perder informações importantes. Existem métricas que avaliam os subconjuntos baseado em mais de um objetivo ao mesmo tempo.

7.2.1 Paradigmas para seleção de atributos

Algoritmos que realizam seleção de atributos podem ser classificados em três principais categorias, dependendo deles utilizarem ou não o algoritmo que constrói o modelo de predição no

momento de selecionar os atributos (Kohavi & John, 1997): (*i*) abordagem filtro; (*ii*) abordagem *wrapper*; e (*iii*) métodos embutidos (do inglês, *embedded*). Existem também abordagens híbridas, envolvendo mais de uma categoria ao mesmo tempo (Das, 2001; Liu & Yu, 2005).

- **Abordagem Filtro**

Os algoritmos pertencentes à primeira categoria são independentes do modelo que será empregado posteriormente. Esses algoritmos trabalham com propriedades estatísticas intrínsecas dos dados, como a correlação, informação mútua e entropia cruzada, na tentativa de identificar a relevância e/ou redundância de cada atributo.

Mesmo sem considerar o impacto do uso efetivo do subconjunto de atributos selecionados, a seleção é realizada com a expectativa de melhorar o desempenho e facilitar o treinamento do modelo. Por ela ser considerada como uma etapa de filtragem do conjunto de dados, vem a denominação filtro. O sucesso desse tipo de método geralmente decorre do fato de se utilizar uma medida de correlação linear/não-linear robusta, assim como uma estratégia de busca capaz de fugir de mínimos locais.

Uma vantagem dos algoritmos na categoria filtro é que a seleção de atributos precisa ser executada apenas uma vez e, então, diferentes modelos podem ser empregados. Outras vantagens são seu baixo custo computacional e sua capacidade de lidar com conjuntos de dados de dimensão elevada (Saeys et al., 2007). Algoritmos pertencentes a essa abordagem são muito utilizados atualmente, principalmente na estatística.

Uma desvantagem é que a abordagem filtro negligencia a interação com o modelo, o que não garante bons resultados finais para qualquer tipo de classificador ou regressor.

Desta forma, a abordagem filtro é mais indicada quando existe um interesse maior na obtenção de um conjunto de dados simplificado e quando os recursos computacionais disponíveis são limitados em comparação com a dimensão do problema.

Exemplos clássicos de algoritmos nessa categoria são o FOCUS (Almuallim & Dietterich, 1991), o RELIEF (Kira & Rendell, 1992) e aqueles baseados no conceito de *Markov blanket* (Castro & Von Zuben, 2009b).

- **Abordagem *Wrapper***

A abordagem *wrapper* seleciona atributos de acordo com sua utilidade para um dado modelo de predição. Para cada subconjunto de atributos a ser avaliado é treinado um classificador ou regressor, representando o modelo, e seu desempenho é utilizado como medida de qualidade desse subconjunto. O nome vem do fato do algoritmo envolver (do inglês, *wrap*) o modelo (classificador ou regressor) como parte do processo de seleção dos atributos.

Usualmente, os modelos empregados para avaliar subconjuntos de atributos, os quais exercem o papel de classificador ou regressor, são árvores de decisão (Bala et al., 1996), k-vizinhos mais próximos (Punch et al., 1993), sistemas de regras (Vafaie & DeJong, 1993), naïve Bayes (Cantú-Paz, 2002), redes neurais (Brotherton & Simpson, 1995) e sistemas *fuzzy* (Castro et al., 2004).

Essa estrutura permite que as técnicas *wrappers* busquem atributos que sejam relevantes para um modelo específico. Desta forma, os resultados de um *wrapper* não possuem garantias de bom desempenho quando aproveitados para um outro tipo de modelo.

Esta abordagem requer maior esforço computacional que as técnicas da abordagem filtro, especialmente quando a construção do modelo é computacionalmente intensiva. Entretanto, resultados reportados na literatura indicam que ela apresenta melhor desempenho quando comparada com a abordagem filtro (Cantú-Paz, 2002). Um outro ponto desfavorável é a tendência de sobre-ajuste (*overfitting*) dos dados, principalmente quando o número de instâncias é pequeno.

Apesar das desvantagens, o uso da abordagem *wrapper* justifica-se devido ao seu objetivo primário: o de obter um subconjunto de atributos que seja o mais indicado para o modelo a ser empregado na resolução do problema. É importante ressaltar que, além de selecionar o subconjunto de atributos, a abordagem *wrapper* também permite obter o próprio modelo já treinado como consequência do processo de seleção de atributos.

O uso de *wrappers* é indicado para casos em que o número de amostras é reduzido e quando o modelo a ser empregado já está bem definido.

- **Métodos embutidos**

Os métodos embutidos envolvem a seleção de atributos como parte integrante de algum algoritmo de aprendizado de máquina. Nesses algoritmos, o processo de seleção não é claramente diferenciável do treinamento do modelo e, assim como no caso da abordagem *wrapper*, os resultados obtidos são calibrados em relação a um classificador ou regressor específico.

Esta classe de algoritmos é comparativamente pouco discutida na literatura geral de seleção de atributos. Os algoritmos possuem poucas propriedades em comum e, por isso, são melhor explorados na literatura de aprendizado de máquina. Exemplos de métodos embutidos são árvores de decisão, como o CART (*Classification and Regression Trees*), proposto por Breiman et al. (1984), e as Máquinas de Vetores-Suporte (Rakotomamonjy, 2003).

7.3 Experimentos Computacionais

Nesta seção, são descritos os experimentos realizados para avaliar a viabilidade dos algoritmos BAIS e MOBAIS na tarefa de seleção de atributos para problemas de classificação de padrões.

Os experimentos estão divididos e serão apresentados em duas partes (Subseções 7.3.2 e 7.3.3). Em ambos os experimentos, foi utilizada a abordagem *wrapper*, ou seja, cada solução candidata foi avaliada mediante a acuidade de um classificador. A diferença é que, na primeira parte dos experimentos, a seleção de atributos é uma tarefa de otimização mono-objetivo (maximizar acuidade do classificador) e, portanto, o BAIS foi aplicado. Já na segunda parte, ela é multi-objetivo (maximizar acuidade do classificador e minimizar número de atributos selecionados) e, por isso, MOBAIS foi utilizado.

Os principais objetivos dos experimentos são: (*i*) analisar a escalabilidade dos algoritmos propostos quando aplicados a conjuntos de dados de alta dimensão, e (*ii*) comparar os resultados obtidos com os resultados produzidos por outros algoritmos da literatura.

Nas subseções seguintes, são apresentados os conjuntos de dados utilizados e os resultados obtidos.

7.3.1 Descrição dos conjuntos de dados

Foram considerados 10 problemas de classificação, obtidos do *UCI Repository of Machine Learning Databases* (Asunción & Newman, 2007): WDBC, Bupa, Ionosphere, Pima, Wine, Iris, Mushroom, Sonar, Card e Vote. Na Tabela 7.1, são apresentadas as características de cada um deles, com número total de instâncias, número de atributos e número de classes. Estes conjuntos de dados são amplamente utilizados na literatura e, por isso, foram utilizados nos experimentos.

Tabela 7.1: Características dos conjuntos de dados.

Conjunto de dados	# Instâncias	# Atributos	# Classes
WDBC	569	30	2
Bupa	345	6	2
Ionosphere	350	34	2
Pima	768	8	2
Wine	178	13	3
Iris	150	4	3
Mushroom	8124	22	2
Sonar	208	60	2
Card	690	15	2
Vote	435	16	2

Os atributos contínuos foram discretizados usando um procedimento proposto por Dougherty et al. (1995), o qual aplica a métrica Comprimento Mínimo de Descrição (MDL, do inglês *Minimum Description Length*). Esse pré-processamento foi realizado para que o classificador utilizado (naïve

Bayes) pudesse manipular os dados de forma mais eficiente. Embora existam versões do naïve Bayes para lidar tanto com atributos discretos como contínuos, melhores desempenhos são obtidos quando os dados são discretizados (Dougherty et al., 1995).

Os conjuntos de dados foram testados utilizando a validação cruzada de 10 pastas. O conjunto de dados é dividido em 10 subconjuntos disjuntos de igual tamanho, sendo 9 subconjuntos para treinamento e 1 para teste. Esse procedimento é realizado dez vezes, permutando-se todos os subconjuntos. A capacidade de generalização do classificador é estimada tomando-se a média da taxa de classificação correta sobre os 10 subconjuntos de teste.

A validação cruzada de 10 pastas foi realizada 10 vezes. Portanto, os algoritmos foram executados 100 vezes e o resultado da classificação é a média sobre essas 100 execuções usando os conjuntos de teste.

7.3.2 Seleção de atributos usando uma abordagem mono-objetivo

Nesta subseção, são apresentados e analisados os resultados obtidos por BAIS, o qual é aplicado ao problema de seleção de atributos visando otimizar um único objetivo: a acuidade do classificador. Antes de apresentar os resultados, serão descritos os detalhes sobre o funcionamento do algoritmo nesse tipo de tarefa.

Cada anticorpo representa uma solução candidata, ou seja, um subconjunto de atributos. Os anticorpos são codificados com dígitos binários e possuem comprimento igual à quantidade de atributos no conjunto de dados original. Cada posição do vetor está associado com um atributo, sendo que o número “1” indica que o respectivo atributo foi selecionado e o número “0” (zero) indica que ele não foi selecionado.

A cada iteração do algoritmo, os anticorpos são avaliados individualmente calculando-se a utilidade dos subconjuntos de atributos neles codificados para determinado classificador (abordagem *wrapper*). Para estes experimentos, foi utilizado o classificador naïve bayes devido a sua rapidez, simplicidade e bom desempenho. Portanto, para cada solução, um classificador naïve bayes foi treinado utilizando validação cruzada de 5 pastas. O *fitness* do anticorpo é a acuidade média do classificador sobre os 5 conjuntos de teste. Repare que a metodologia emprega a validação cruzada em dois níveis, um externo (para verificar a capacidade de generalização do classificador com o subconjunto de atributos gerado como solução final pelo BAIS) e esse mais interno (para treinar o classificador e avaliar cada solução candidata). Na validação cruzada do nível interno, os dados disponíveis são os de treinamento, fornecidos pela validação cruzada externa.

Ao final de cada execução de BAIS, o melhor anticorpo da população é retornado como solução. O naïve Bayes é, então, aplicado sobre o conjunto de teste usando os atributos codificados nesse anticorpo.

Foi imposta uma restrição no número de pais que os nós da rede bayesiana pode ter, no intuito de penalizar modelos muito complexos. Desta forma, cada nó pode ter, no máximo, 3 pais. O número de novas soluções geradas a partir da rede é metade do número de indivíduos na população corrente. A população inicial foi definida como 100, porém este número pode aumentar ou diminuir ao longo da busca. O critério de parada é quando o algoritmo alcançar o número máximo de iterações, definido como 100. O limiar de supressão varia conforme o conjunto de dados. Esses parâmetros foram obtidos empiricamente por meio de experimentos preliminares.

Na Tabela 7.2, são apresentados os resultados médios obtidos para cada conjunto de dados. Na segunda e terceira colunas da tabela estão os resultados médios quando foram utilizados todos os atributos disponíveis e quando foram utilizados somente os atributos selecionados por BAIS, respectivamente. Os números entre parênteses indicam a quantidade média de atributos contidos na melhor solução obtida por BAIS.

Para verificar se a diferença nos resultados é estatisticamente significativa, o *teste-t* foi empregado no nível de 95% de confiança e o *p*-valor é mostrado na quarta coluna da tabela. Se o *p*-valor for inferior a 0,05 então a diferença é realmente significativa.

Tabela 7.2: Taxa de classificação média usando validação cruzada de 10 pastas para os dez conjuntos de dados.

Conjunto de dados	Todos os atributos (%)	BAIS (%)	*p*-valor
WDBC	96,7±0,8	97,2±0,6 (14,3±0,4)	0,361
Bupa	71,6±2,3	75,2±1,9 (2,6±0,1)	0,003
Ionosphere	84,8±1,4	91,2±1,7 (13,4±0,4)	0,000
Pima	74,7±2,7	75,4±2,1 (3,2±0,2)	0,442
Wine	96,3±0,7	98,4±0,4 (7,8±0,8)	0,027
Iris	94,6±1,1	96,3±0,8 (2,7±0,2)	0,018
Mushroom	94,7±1,3	98,1±1,4 (11,5±1,6)	0,000
Sonar	71,8±3,2	76,3±1,8 (23,6±2,0)	0,000
Card	79,2±2,5	83,6±1,9 (9,3±0,7)	0,002
Vote	91,3±1,6	94,7±0,9 (7,3±0,6)	0,013

Analisando a Tabela 7.2, é possível verificar que BAIS conseguiu encontrar subconjuntos reduzidos de atributos para todos os casos. Além disso, uma melhora expressiva no desempenho do classificador foi obtida, com destaque para os conjuntos de dados Bupa, Ionosphere, Mushroom, Sonar e Card.

Do ponto de vista do mecanismo de busca, é interessante ressaltar uma importante característica de BAIS: o tamanho da população cresce e diminui dinamicamente ao longo da busca, permitindo melhor exploração do espaço de busca. Esse comportamento exerce papel fundamental na capacidade

do algoritmo de realizar busca multimodal. Apenas para comprovar essa capacidade, na Tabela 7.3 são apresentados subconjuntos alternativos de atributos de boa qualidade encontrados pelo algoritmo em uma única execução. Foram selecionados alguns casos apenas para os conjuntos de dados WDBC, Ionosphere, Pima e Wine, por serem mais expressivos.

Tabela 7.3: Soluções alternativas de boa qualidade encontradas por BAIS em um única execução.

Conjunto de dados	Atributos selecionados	Acuidade (%)
WDBC	3, 5, 8, 12, 18, 20, 21, 22, 26, 27, 28, 29	98,1
	5, 7, 8, 10, 11, 16, 21, 22, 24, 25, 27, 29	97,7
	1, 6, 7, 8, 9, 10, 12, 16, 18, 21, 25, 26, 29, 30	97,2
Ionosphere	3, 5, 6, 7, 8, 16, 18, 24, 30, 31, 33, 34	91,8
	3, 5, 6, 7, 8, 18, 22, 24, 25, 31, 32, 33, 34	91,3
Pima	2, 6, 7, 8	75,6
	1, 2, 4, 6, 8	75,4
	3, 6, 7, 8	75,2
	1, 6, 7	74,7
Wine	1, 7, 10, 11, 12, 13	98,6
	1, 5, 7, 8, 10, 13	97,8
	1, 5, 6, 7, 10, 12, 13	97,4
	1, 6, 7, 8, 9, 10, 12, 13	97,1

Ainda sob a perspectiva do processo de busca, foi observado que BAIS consegue identificar e manter os blocos construtivos logo nas primeiras iterações. Na Figura 7.2 são apresentadas as redes bayesianas geradas em algumas iterações do algoritmo, quando aplicado ao conjunto de dados Pima. Embora as redes sejam ligeiramente diferentes, alguns nós (atributos) permanecem sempre relacionados. Por exemplo, a partir da geração 40 os atributos 2, 6, 8 e 7 se mantêm relacionados. Repare que, de fato, esses atributos aparecem com frequência para o conjunto de dados Pima (veja Tabela 7.3). O mesmo comportamento foi observado para os outros conjuntos de dados.

Em seguida, os resultados produzidos por BAIS foram comparados com aqueles gerados por 4 algoritmos: RELIEF-E, FSS-EBNA, um sistema imunológico artificial (SIA) e um algoritmo genético (AG). Todos foram avaliados seguindo o mesmo procedimento de validação cruzada de 10 pastas executada 10 vezes.

O algoritmo RELIEF-E (Kononenko, 1994) é um algoritmo pertencente à abordagem filtro e foi desenvolvido a partir do algoritmo original RELIEF (Kira & Rendell, 1992) para possibilitar a manipulação de problemas de classificação com mais de duas classes. O algoritmo atribui um peso para cada atributo refletindo a suposta habilidade do atributo de discriminar as classes. O critério de seleção escolhe os atributos cujos pesos excedem um determinado limiar definido pelo usuário. Uma

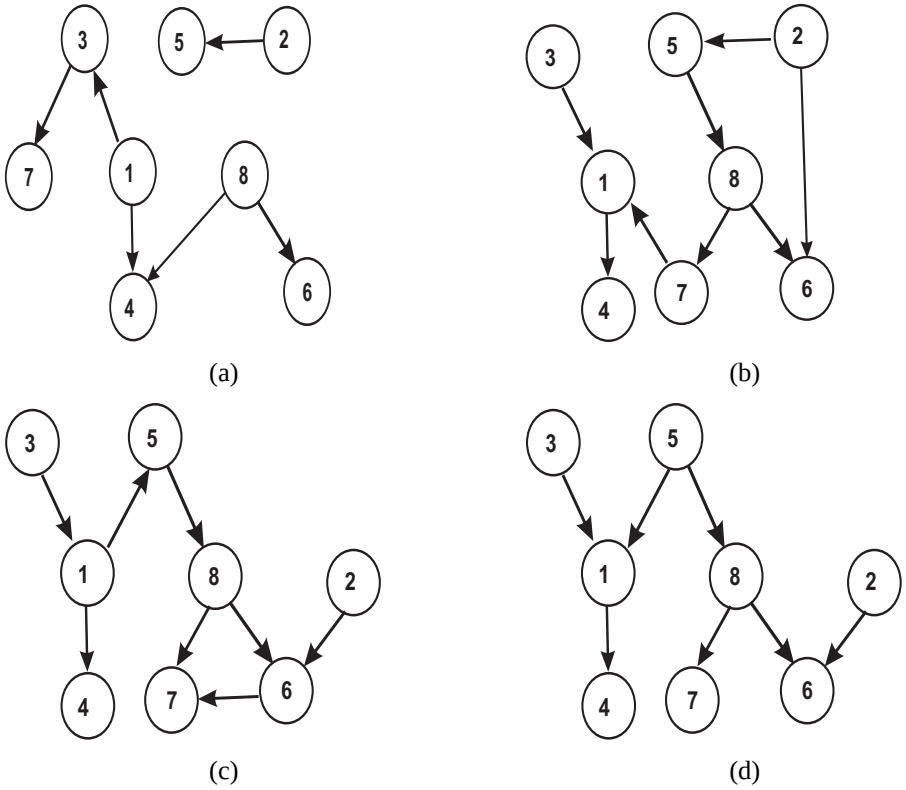


Figura 7.2: Redes bayesianas geradas por BAIS a partir das soluções candidatas para o conjunto de dados Pima: (a) iteração 10, (b) iteração 40, (c) iteração 75 e (d) iteração 100.

característica do RELIEF-E é que ele procura por todos os atributos relevantes para a classe, mesmo sendo redundantes.

O FSS-EBNA (Inza et al., 2000) é um algoritmo de estimação de distribuição que utiliza uma rede bayesiana para representar as dependências entre as variáveis do problema, assim como BAIS. A população inicial foi definida como 1000 e permanece fixa durante todo o processo de busca. O algoritmo para quando alcançar o número máximo de iterações, estabelecido como 800. Para gerar a rede bayesiana, as 250 melhores soluções foram utilizadas.

Um SIA foi implementado usando os princípios da seleção clonal e teoria da rede imunológica. Cada anticorpo da população dá origem a 2 clones. A população inicial possui 100 indivíduos e o número máximo de iterações foi definido como 500.

Por fim, o AG implementado também utiliza uma população com 1000 indivíduos, taxas de cruzamento e mutação de 0,8 e 0,1, respectivamente. O operador de seleção é o torneio sem reposição. O número máximo de iterações é 800.

Os parâmetros utilizados foram obtidos a partir da literatura ou empiricamente, procurando-se maximizar o desempenho de cada um dos algoritmos. Para os três algoritmos bio-inspirados, foi utilizada a abordagem *wrapper*, com o naïve Bayes como classificador.

Na Tabela 7.4, são apresentados os resultados comparativos entre BAIS e RELIEF-E. Entre parênteses, está a quantidade média de atributos encontrada por cada proposta. Os resultados mostram que ambos os algoritmos conseguiram selecionar um número reduzido de atributos. Entretanto, o classificador utilizando os atributos selecionados por BAIS apresentou melhor desempenho quando comparado ao classificador que utilizou os atributos selecionados por RELIEF-E. Para os conjuntos de dados WDBC, Bupa e Vote, o desempenho do classificador utilizando os atributos selecionados por RELIEF-E foi pior que os resultados obtidos ao utilizar todos os atributos. Isso deve ter ocorrido, provavelmente, pelo fato do RELIEF-E ser um filtro e não interagir com o classificador no momento da seleção dos atributos.

Tabela 7.4: Taxa de classificação média obtida por BAIS e RELIEF-E.

Conjunto de dados	BAIS (%)		RELIEF-E (%)		*p*-valor
WDBC	97,2±0,6	(14,3±0,4)	95,3±0,9	(12,6±0,7)	0,021
Bupa	75,2±1,9	(2,6±0,1)	70,8±1,1	(1,9±1,3)	0,000
Ionosphere	91,2±1,7	(13,4±0,4)	85,4±0,8	(15,3±0,5)	0,008
Pima	75,4±2,1	(3,2±0,2)	73,2±1,6	(4,8±0,8)	0,074
Wine	98,4±0,4	(7,8±0,8)	96,6±0,8	(5,4±1,1)	0,105
Iris	96,3±1,2	(2,7±0,2)	94,8±1,1	(2,9±0,6)	0,046
Mushroom	98,1±1,4	(11,5±1,6)	94,9±1,3	(16,1±2,2)	0,000
Sonar	76,3±1,8	(23,6±2,0)	72,7±1,6	(31,2±3,4)	0,000
Card	83,6±1,9	(9,3±0,7)	80,1±0,9	(9,1±0,9)	0,000
Vote	94,7±1,1	(7,3±0,6)	90,7±1,8	(10,3±0,6)	0,000

Na Tabela 7.5, são apresentados os resultados comparativos entre BAIS e FSS-EBNA. Resultados similares foram obtidos por ambos os algoritmos, em termos de acuidade, com uma ligeira vantagem a favor de BAIS para a maioria dos conjuntos de dados. Vale ressaltar que para o FSS-EBNA conseguir resultados similares ao do BAIS, ele precisou de um número muito maior de iterações (800 contra 100 de BAIS).

O SIA também foi apto a selecionar um conjunto reduzido de atributos. Quando o classificador utilizou tais atributos, houve uma melhora no desempenho, em termos de taxa de classificação correta. Entretanto, se comparado ao BAIS, o sistema imunológico artificial sem a abordagem bayesiana apresenta resultados inferiores para alguns casos, inclusive para o número de atributos selecionados. Na Tabela 7.6, são reportados os resultados obtidos por BAIS e pelo SIA.

Tabela 7.5: Taxa de classificação média obtida por BAIS e FSS-EBNA.

Conjunto de dados	BAIS (%)	FSS-EBNA (%)	*p*-valor
WDBC	97,2±0,6 (14,3±0,4)	97,4±1,2 (15,1±1,2)	0,480
Bupa	75,2±1,9 (2,6±0,1)	74,1±2,3 (2,8±0,5)	0,092
Ionosphere	91,2±1,7 (13,4±0,4)	90,3±1,6 (11,9±1,1)	0,173
Pima	75,4±2,1 (3,2±0,2)	75,1±2,4 (5,2±1,3)	0,215
Wine	98,4±0,4 (7,8±0,8)	98,6±0,8 (7,1±1,7)	0,771
Iris	96,3±1,2 (2,7±0,2)	95,7±1,4 (3,1±0,9)	0,538
Mushroom	98,1±1,4 (11,5±1,6)	96,3±2,9 (13,6±2,4)	0,036
Sonar	76,3±1,8 (23,6±2,0)	74,2±3,1 (19,5±3,4)	0,000
Card	83,6±1,9 (9,3±0,7)	82,2±2,4 (12,6±1,7)	0,094
Vote	94,7±1,1 (7,3±0,6)	93,1±1,7 (13,2±2,3)	0,045

Tabela 7.6: Taxa de classificação média obtida por BAIS e SIA.

Conjunto de dados	BAIS (%)	SIA (%)	*p*-valor
WDBC	97,2±0,6 (14,3±0,4)	96,8±1,2 (17,2±3,6)	0,008
Bupa	75,2±1,9 (2,6±0,1)	74,9±2,1 (3,4±1,8)	0,127
Ionosphere	91,2±1,7 (13,4±0,4)	87,7±1,6 (18,6±4,1)	0,000
Pima	75,4±2,1 (3,2±0,2)	74,6±2,8 (4,9±1,3)	0,0466
Wine	98,4±0,4 (7,8±0,8)	97,8±1,2 (6,7±1,7)	0,0251
Iris	96,3±1,2 (2,7±0,2)	95,9±0,9 (2,3±0,9)	0,470
Mushroom	98,1±1,4 (11,5±1,6)	96,2±3,4 (14,7±4,4)	0,000
Sonar	76,3±1,8 (23,6±2,0)	73,4±2,3 (23,2±3,7)	0,000
Card	83,6±1,9 (9,3±0,7)	83,1±1,8 (13,7±1,6)	0,848
Vote	94,7±1,1 (7,3±0,6)	93,6±1,6 (10,7±3,4)	0,0182

Das abordagens bio-inspiradas, o algoritmo genético foi o que apresentou piores resultados. Para os conjuntos de dados WDBC e Pima, houve piora no desempenho do classificador quando comparado à utilização de todos os atributos originais. Esses resultados sugerem que o algoritmo genético ficava preso em mínimos locais, escolhendo subconjuntos de dados de baixa qualidade. Mesmo alterando os parâmetros do AG, seu comportamento permanecia o mesmo. Os resultados obtidos pelo algoritmo genético são apresentados na Tabela 7.7.

Tabela 7.7: Taxa de classificação média obtida por BAIS e AG.

Conjunto de dados	BAIS (%)		AG (%)		*p*-valor
WDBC	97,2±0,6	(14,3±0,4)	96,2±1,6	(12,8±3,2)	0,020
Bupa	75,2±1,9	(2,6±0,1)	73,7±2,5	(4,4±0,8)	0,000
Ionosphere	91,2±1,7	(13,4±0,4)	86,2±1,4	(18,5±3,3)	0,000
Pima	75,4±2,1	(3,2±0,2)	73,6±3,2	(5,7±0,6)	0,620
Wine	98,4±0,4	(7,8±0,8)	96,7±0,7	(9,2±1,3)	0,053
Iris	96,3±1,2	(2,7±0,2)	94,9±1,6	(3,1±0,6)	0,104
Mushroom	98,1±1,4	(11,5±1,6)	95,6±1,8	(23,1±3,7)	0,000
Sonar	76,3±1,8	(23,6±2,0)	72,1±2,3	(31,7±4,3)	0,000
Card	83,6±1,9	(9,3±0,7)	81,5±1,4	(9,1±2,1)	0,006
Vote	94,7±1,1	(7,3±0,6)	92,1±1,8	(11,7±1,4)	0,026

Da Figura 7.3 à Figura 7.12, são apresentados os *boxplots* comparando os algoritmos para cada um dos conjuntos de dados. Estes *boxplots* foram elaborados levando em consideração as 10 execuções da validação cruzada de 10 pastas. Pode-se perceber que houve melhora no desempenho do classificador para todos os casos.

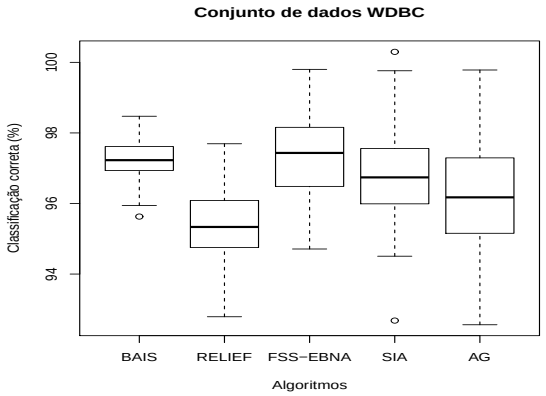


Figura 7.3: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados WDBC.

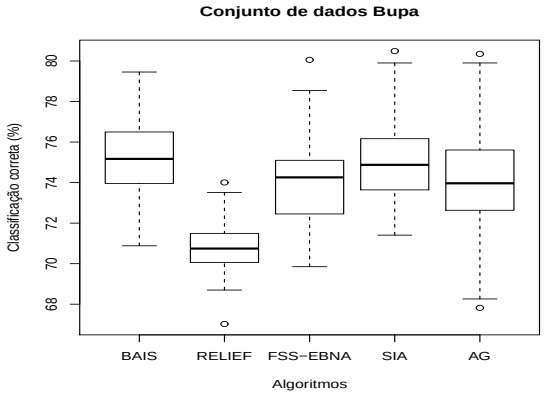


Figura 7.4: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Bupa.

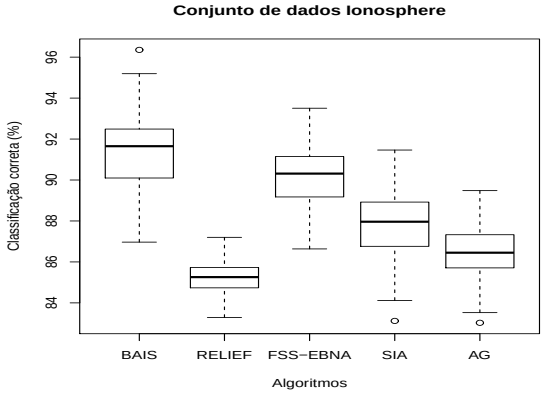


Figura 7.5: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Ionosphere.

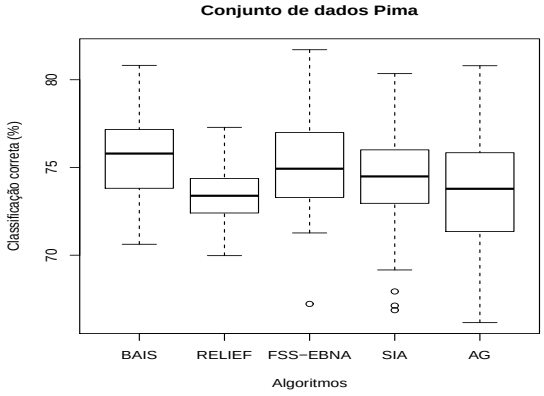


Figura 7.6: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Pima.

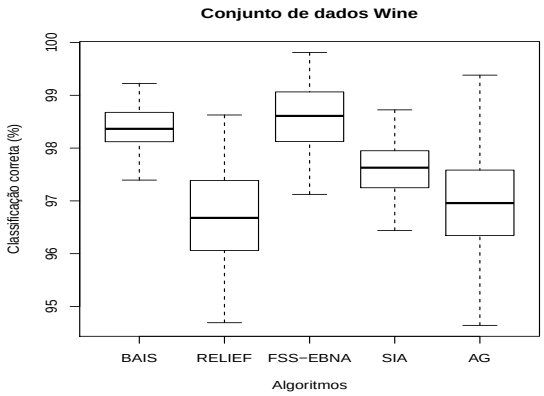


Figura 7.7: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Wine.

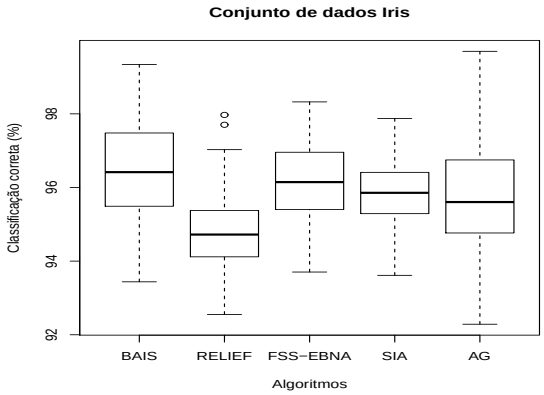


Figura 7.8: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Iris.

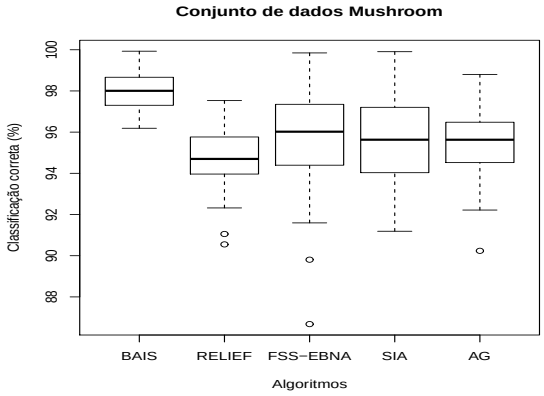


Figura 7.9: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Mushroom.

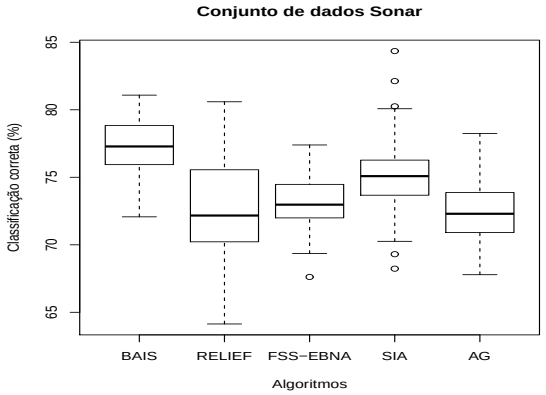


Figura 7.10: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Sonar.

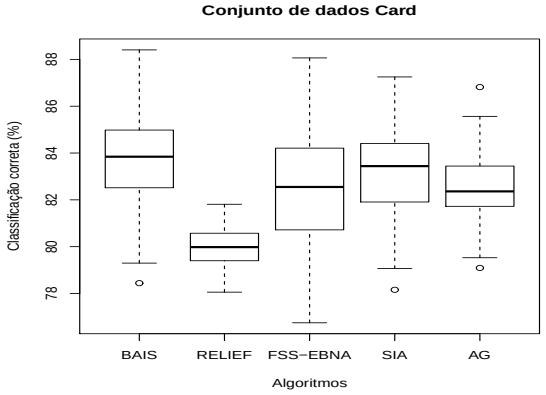


Figura 7.11: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Card.

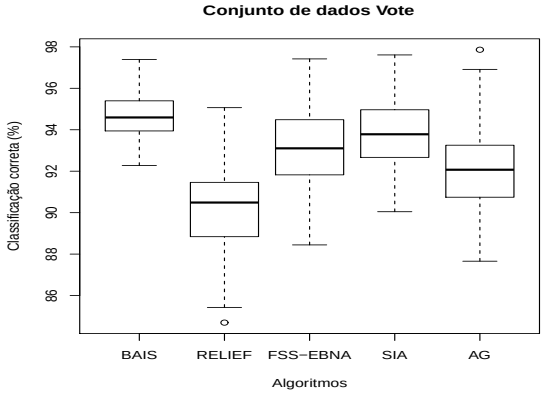


Figura 7.12: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Vote.

7.3.3 Seleção de atributos usando uma abordagem multiobjetivo

Embora soluções de boa qualidade tenham sido obtidas usando BAIS, subconjuntos de atributos mais parcimoniosos e que proporcionam melhores taxas de classificação corretas poderiam ser obtidos se fosse utilizado um algoritmo de otimização multiobjetivo, visando um compromisso entre acuidade e número de atributos.

Nesse sentido, foram realizados experimentos utilizando o MOBAIS para efetuar a seleção de atributos. Primeiramente, serão apresentados os resultados obtidos por ele, os quais serão comparados com os resultados produzidos por BAIS. Em seguida, esses resultados são confrontados com aqueles gerados por três outros algoritmos bio-inspirados que realizam otimização multiobjetivo.

Os parâmetros utilizados por MOBAIS foram iguais aos do BAIS. O tamanho da população inicial é 100, o critério de parada foi 100 iterações, o número máximo de pais que um nó da rede bayesiana pode ter é 3 e o limiar de supressão varia de acordo com o conjunto de dados. O mesmo procedimento adotado anteriormente para avaliar o algoritmo foi adotado aqui também: 10 execuções da validação cruzada de 10 pastas. Para sintetizar o naïve Bayes, foi utilizada validação cruzada de 5 pastas.

Após a execução do algoritmo, um conjunto de soluções não-dominadas (fronteira de Pareto) é retornado como solução. É preciso, portanto, adotar uma estratégia de pós-otimização para tentar selecionar uma única solução da fronteira de Pareto. Neste trabalho, foi empregada a técnica chamada *Compromise Programming* (Zeleny, 1973), que consiste em calcular a distância de todas as soluções alternativas para um ponto de referência ideal. A solução com menor distância é considerada o melhor compromisso entre os objetivos. Desta forma, é necessário definir o ponto de referência e como calcular a distância de todas as soluções para ele. O ponto de referência foi estabelecido como sendo aquele localizado nos valores mínimos dos dois objetivos (taxa de erro e cardinalidade do subconjunto de atributos) e a medida de distância utilizada foi a euclidiana.

A taxa de classificação média obtida por MOBAIS para os 10 conjuntos de dados é apresentada na Tabela 7.8. São apresentados também, a título de comparação, os resultados obtidos por BAIS e aqueles referentes à utilização de todos os atributos do conjunto de dados original. Novamente, o número entre parênteses indica o tamanho médio do subconjunto de atributos.

Pela Tabela 7.8, é possível verificar que MOBAIS gerou subconjuntos de atributos menores que os encontrados por BAIS, sem degradar o desempenho do classificador, com exceção para o conjunto de dados Bupa. Porém, essa diferença não é estatisticamente significante. Estes resultados mostram que a abordagem multiobjetivo para seleção de atributos é efetiva.

Na Tabela 7.9, são apresentados os subconjuntos de atributos produzidos por MOBAIS para cada conjunto de dados. Cada subconjunto corresponde à melhor solução encontrada ao longo das 100 execuções do algoritmo (10 validações cruzadas de 10 pastas).

Análises comparativas foram realizadas levando-se em conta três algoritmos de otimização

Tabela 7.8: Resultados obtidos por MOBAIS e a comparação com BAIS.

Conjunto de dados	Todos os atributos	BAIS (%)		MOBAIS (%)	
WDBC	96,7±0,8	97,2±0,6	(14,3±0,4)	97,9±0,5	(8,7±0,7)
Bupa	71,6±2,3	75,2±1,9	(2,6±0,1)	74,9±0,7	(2,4±0,1)
Ionosphere	84,8±1,4	91,2±1,7	(13,4±0,4)	93,4±1,3	(9,9±0,2)
Pima	74,7±2,7	75,4±2,1	(3,2±0,2)	77,6±1,8	(2,8±0,1)
Wine	96,3±0,7	98,4±0,4	(7,8±0,8)	98,2±0,6	(6,1±0,2)
Iris	94,6±1,1	96,3±0,8	(2,7±0,2)	96,5±1,2	(2,2±0,3)
Mushroom	94,7±1,3	98,1±1,4	(11,5±1,6)	98,1±1,0	(8,9±1,3)
Sonar	71,8±3,2	76,3±1,8	(23,6±2,0)	77,1±2,1	(14,2±2,4)
Card	79,2±2,5	83,6±1,9	(9,3±0,7)	83,8±1,5	(6,8±1,1)
Vote	91,3±1,6	94,7±0,9	(7,3±0,6)	97,5±0,7	(6,2±0,8)

Tabela 7.9: Melhor subconjunto de atributos selecionados por MOBAIS para cada conjunto de dados ao longo das 100 execuções.

Conjunto de dados	Atributos selecionados
WDBC	12, 18, 20, 21, 22, 27, 28
Bupa	1, 2, 4
Ionosphere	1, 3, 5, 6, 7, 8, 16, 18, 33, 34
Pima	1, 6, 7
Wine	2, 5, 8, 9, 12, 13
Iris	3, 4
Mushroom	4, 5, 8, 10, 12, 15, 16, 19, 22
Sonar	4, 5, 9, 10, 11, 21, 28, 35, 44, 45, 46, 48, 51, 54
Card	4, 6, 9, 10, 14, 15
Vote	8, 10, 11, 12, 14

multiobjetivo. O primeiro é o NSGA-II (Deb et al., 2000), que emprega seleção por *rank* e *crowding distance*. O outro algoritmo é o *Multi-objective Immune System Algorithm* (MISA) (Coello Coello & Cortés, 2005), e que usa uma população secundária para implementar elitismo. Finalmente, o *Multi-objective Bayesian Optimization Algorithm* (mBOA) (Khan et al., 2002) é um algoritmo de estimação de distribuição que também utiliza rede bayesiana.

Para o NSGA-II, as taxas de cruzamento e mutação são 0,8 e 0,01, respectivamente, junto com seleção por torneio. No MISA, o operador de mutação uniforme foi empregado. Para o mBOA, a métrica K2 foi utilizada para construir a rede bayesiana. A quantidade de amostras geradas é o tamanho da população. Para os três algoritmos concorrentes, o tamanho da população foi definido

como 1000.

Os resultados são apresentados na Tabela 7.10, Tabela 7.11 e Tabela 7.12, junto com o número médio de atributos selecionados (entre parênteses). O *teste-t* foi aplicado com nível de confiança igual a 95% e o *p*-valor é reportado na quarta coluna de cada tabela.

Tabela 7.10: Resultados comparativos entre MOBAIS e NSGA-II.

Conjunto de dados	MOBAIS (%)		NSGA-II (%)		*p*-valor
WDBC	97,9±0,5	(8,7±0,7)	97,1±0,8	(11,6±1,2)	0,230
Bupa	74,9±0,7	(2,4±0,1)	72,8±1,7	(3,8±0,5)	0,000
Ionosphere	93,4±1,3	(9,9±0,2)	90,7±0,9	(12,9±1,1)	0,000
Pima	77,6±1,8	(2,8±0,1)	75,1±2,6	(6,1±1,3)	0,000
Wine	98,2±0,6	(6,1±0,2)	98,4±0,9	(8,2±1,7)	0,368
Iris	96,5±1,2	(2,2±0,3)	93,6±1,1	(2,9±0,6)	0,003
Mushroom	98,1±1,0	(8,9±1,3)	97,9±0,9	(14,7±3,3)	0,764
Sonar	77,1±2,1	(14,2±2,4)	74,8±2,8	(24,6±2,7)	0,001
Card	83,8±1,5	(6,8±1,1)	82,9±1,7	(10,6±2,4)	0,104
Vote	97,5±0,7	(6,2±0,8)	94,4±1,5	(9,2±1,7)	0,000

Tabela 7.11: Resultados comparativos entre MOBAIS e MISA.

Conjunto de dados	MOBAIS (%)		MISA (%)		*p*-valor
WDBC	97,2±0,5	(8,7±0,7)	96,6±1,1	(10,2±0,9)	0,190
Bupa	74,9±0,7	(2,4±0,1)	74,4±0,5	(4,1±1,2)	0,764
Ionosphere	93,4±1,3	(9,9±0,2)	93,8±1,4	(14,7±2,6)	0,832
Pima	77,6±1,8	(2,8±0,1)	72,1±2,1	(5,6±1,9)	0,000
Wine	98,2±0,6	(6,1±0,2)	96,4±1,3	(7,4±0,8)	0,000
Iris	96,5±1,2	(2,2±0,3)	94,6±0,9	(3,1±0,4)	0,000
Mushroom	98,1±1,0	(8,9±1,3)	95,8±2,2	(13,8±2,7)	0,000
Sonar	77,1±2,1	(14,2±2,4)	73,8±2,5	(31,3±4,1)	0,000
Card	83,8±1,5	(6,8±1,1)	81,7±3,2	(11,2±2,8)	0,000
Vote	97,5±0,7	(6,2±0,8)	95,2±0,6	(11,5±1,4)	0,000

Os resultados mostram que MOBAIS e mBOA foram capazes de encontrar soluções mais parcimoniosas que o NSGA-II e MISA para a maioria dos casos. Entretanto, o desempenho do classificador utilizando os atributos selecionados por MOBAIS é melhor que o do classificador que utilizou os atributos selecionados por mBOA. Isso se deve ao fato do MOBAIS possuir a capacidade

Tabela 7.12: Resultados comparativos entre MOBAIS e mBOA.

Conjunto de dados	MOBAIS (%)		mBOA (%)		*p*-valor
WDBC	97,9±0,5	(8,7±0,7)	97,4±0,6	(13,5±1,2)	0,005
Bupa	74,9±0,7	(2,4±0,1)	70,9±2,3	(2,7±1,1)	0,000
Ionosphere	93,4±1,3	(9,9±0,2)	87,9±1,9	(12,6±0,7)	0,000
Pima	77,6±1,8	(2,8±0,1)	75,7±2,3	(4,2±1,4)	0,028
Wine	98,2±0,6	(6,1±0,2)	96,7±0,9	(6,7±1,5)	0,010
Iris	96,5±1,2	(2,2±0,3)	95,3±1,3	(2,7±0,7)	0,046
Mushroom	98,1±1,0	(8,9±1,3)	96,2±1,4	(10,2±2,8)	0,000
Sonar	77,1±2,1	(14,2±2,4)	73,7±2,6	(12,8±3,1)	0,000
Card	83,8±1,5	(6,8±1,1)	83,9±1,8	(8,5±1,9)	0,827
Vote	97,5±0,7	(6,2±0,8)	96,8±2,2	(6,4±2,7)	0,035

de manipular blocos construtivos eficientemente, aliado ao seu mecanismo robusto de exploração do espaço de busca.

Os resultados comparativos entre os algoritmos, em termos de acuidade do classificador, são apresentados em forma de *boxplots* para todos os conjunto de dados, da Figura 7.13 à Figura 7.22. Estes resultados representam a média dos resultados obtidos sobre 10 execuções da validação cruzada de 10 pastas. Pode-se perceber que houve melhora no desempenho do classificador para todos os casos.

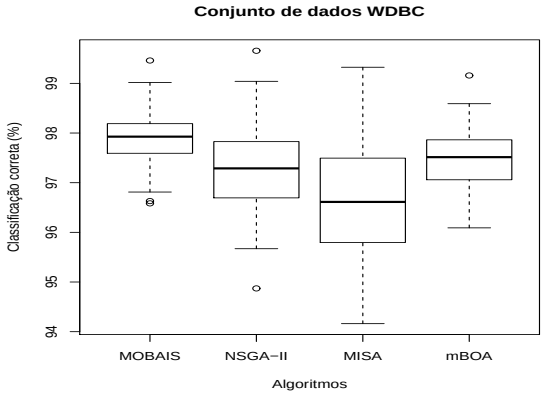


Figura 7.13: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados WDBC.

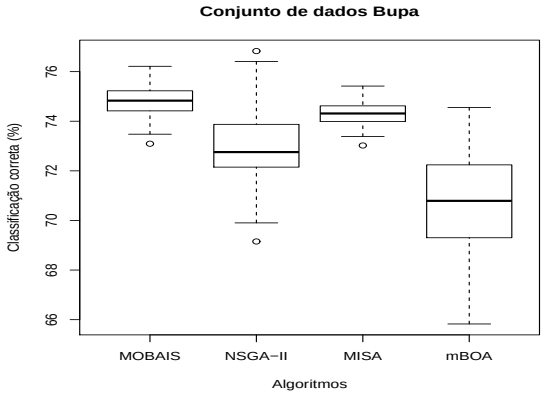


Figura 7.14: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Bupa.

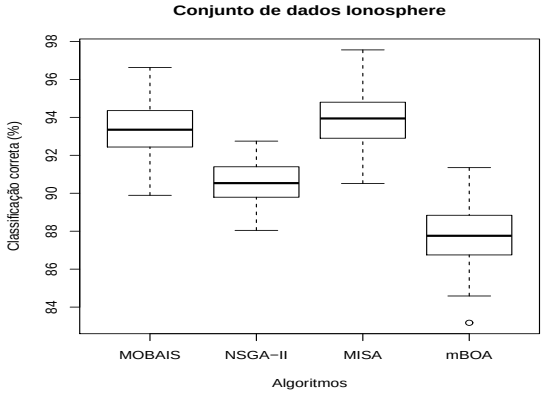


Figura 7.15: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Ionosphere.

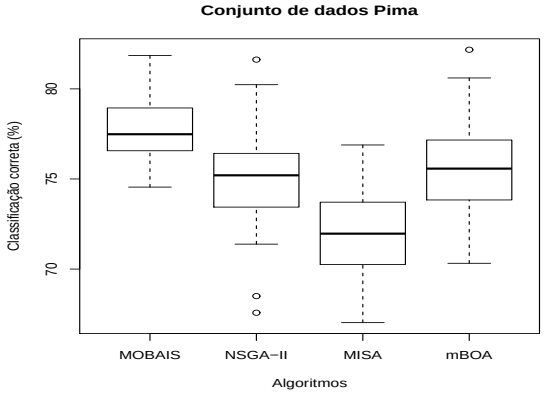


Figura 7.16: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Pima.

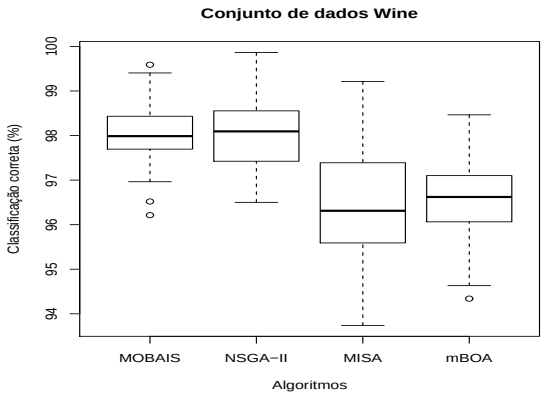


Figura 7.17: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Wine.

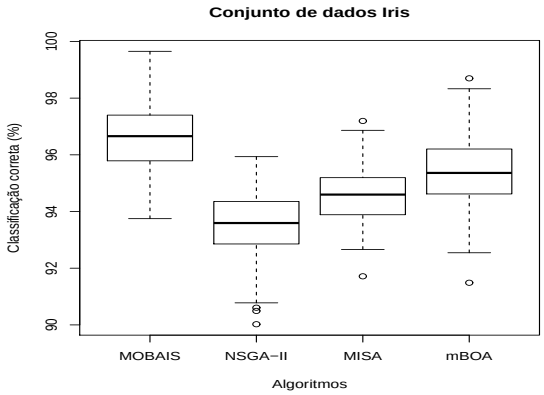


Figura 7.18: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Iris.

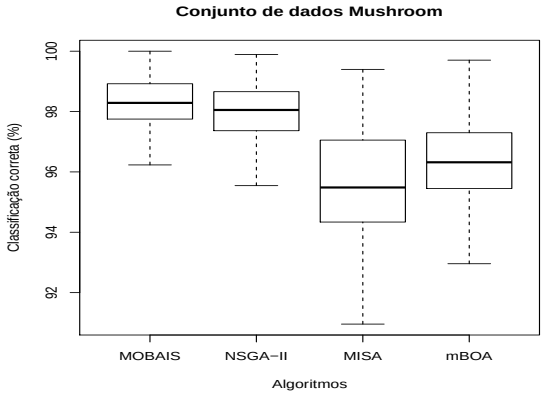


Figura 7.19: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Mushroom.

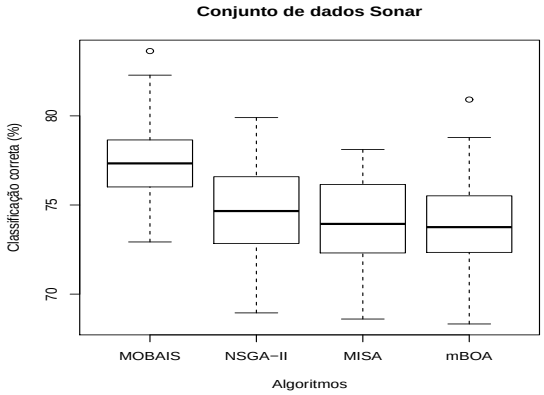


Figura 7.20: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Sonar.

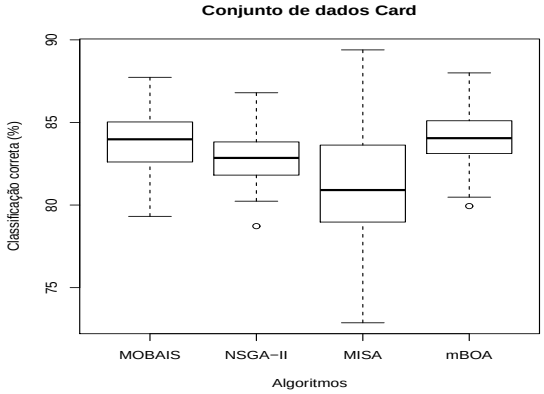


Figura 7.21: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Card.

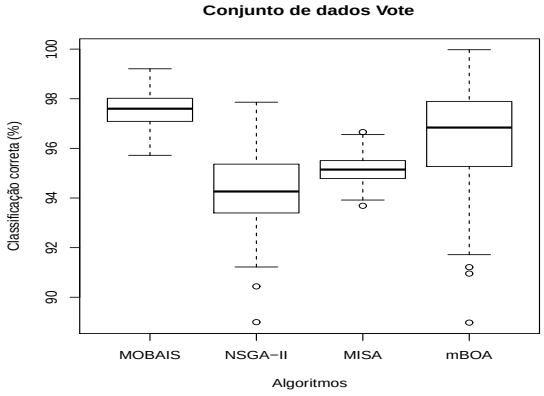


Figura 7.22: *Bloxpot* para os quatro algoritmos junto ao conjunto de dados Vote.

Para confirmar a viabilidade de MOBAIS e sua superioridade frente aos outros algoritmos na tarefa de seleção de atributos multiobjetivo, os algoritmos foram avaliados utilizando duas métricas. A primeira é a Cobertura de Dois Conjuntos (*Two Set Coverage*), proposta por Zitzler et al. (2000), que mostra que porcentagem de soluções de um algoritmo é dominada pelas soluções geradas por outro algoritmo concorrente. A métrica é dada por:

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X' : a' \preceq a''|}{|X''|} \tag{7.1}$$

em que $X', X'' \subseteq X$ são dois conjuntos de soluções. O resultado da Equação (7.1) é um número real no intervalo [0,1]. Isso significa que C=1 quando X' domina ou é igual a X''. Repare que ambos os cálculos $C(X', X'')$ e $C(X'', X')$ precisam ser considerados, uma vez que $C(X', X'')$ não é necessariamente igual a $1 - C(X'', X')$.

Na Tabela 7.13 são reportados os valores médios dessa métrica somente para o conjunto de dados Sonar. Entretanto, comportamento similar foi observado em todos os outros domínios.

Tabela 7.13: Valores médios da métrica Cobertura de Dois Conjuntos (A $\preceq$ B) para o conjunto de dados Sonar.

Algoritmo A	Algoritmo B	Cobertura
MOBAIS	NSGA-II	16,3%
	MISA	91,2%
	mBOA	100%
NSGA-II	MOBAIS	12,4%
	MISA	21,7%
	mBOA	84,4%
MISA	MOBAIS	6,2%
	NSGA-II	19,5%
	mBOA	90,8%
mBOA	MOBAIS	0%
	NSGA-II	12,8%
	MISA	7,1%

Pela Tabela 7.13, pode-se observar que as soluções geradas por MOBAIS obtiveram boa cobertura sobre as soluções geradas pelos outros algoritmos. Em contraste, as soluções geradas por ele são dominadas por algumas poucas soluções. Apenas para ressaltar, as soluções encontradas por MOBAIS dominam, em média, 16% das soluções produzidas pelo NSGA-II, 91,2% das soluções obtidas pelo MISA e 100% das soluções geradas mBOA. Por outro lado, as soluções do NSGA-II dominam 12,4% das soluções encontradas por MOBAIS. Já as soluções obtidas pelo MISA dominam 6,2% das soluções encontradas por MOBAIS. Por fim, as soluções obtidas pelo mBOA não dominam

nenhuma solução gerada por MOBAIS.

A segunda métrica, Ocupação da Fronteira (Van Veldhuizen, 1999), mede o número de soluções na fronteira de Pareto. Na Tabela 7.14 é apresentado o número médio de soluções na fronteira de Pareto para cada algoritmo, considerando o conjunto de dados Sonar.

Tabela 7.14: Número médio de soluções na fronteira de Pareto para o conjunto de dados Sonar.

Algoritmo	Ocupação da fronteira
MOBAIS	62,3
NSGA-II	48,6
MISA	57,2
mBOA	39,1

7.4 Considerações Finais

Neste capítulo, foram descritos alguns experimentos realizados com BAIS (abordagem mono-objetivo) e MOBAIS (abordagem multiobjetivo) para a tarefa de seleção de atributos. Para cada caso, foram realizadas também comparações com outros algoritmos conhecidos na literatura. Pelos resultados obtidos, pode-se perceber as vantagens de BAIS e MOBAIS sobre as outras propostas.

A primeira é referente ao mecanismo efetivo para manipulação de blocos construtivos. Com essa capacidade, os algoritmos propostos neste trabalho evitam o rompimento de soluções parciais de boa qualidade encontradas até então. Embora FSS-EBNA e mBOA também possuam a capacidade de lidar com blocos construtivos, seus mecanismos de exploração do espaço de busca não são tão eficientes como os do BAIS e MOBAIS, levando-os a apresentarem desempenho inferior.

BAIS e MOBAIS encontraram diversas soluções de boa qualidade em poucas iterações, enquanto os outros algoritmos precisaram de uma quantidade maior para alcançar soluções similares. Além disso, o tamanho da população inicial não é crucial para BAIS e MOBAIS devido à capacidade que ambos apresentam de ajustar automaticamente o tamanho da população ao longo da busca, diferentemente dos outros algoritmos.

Capítulo 8

Experimentos com Otimização no Espaço Contínuo

8.1 Considerações Iniciais

Neste capítulo, são apresentados os experimentos realizados com os algoritmos GAIS e MOGAIS, aplicados em problemas de otimização no espaço contínuo.

Na Seção 8.2, são apresentados e discutidos os resultados obtidos por GAIS em sete problemas de otimização mono-objetivo: esfera, Griewank, Rosenbrock, Rastrigin, Schwefel, Michalewicz e Ackley. É apresentada também a comparação do algoritmo sem e com a utilização do modelo de mistura de gaussiana. Por fim, os resultados são comparados com aqueles produzidos por outros dois algoritmos da literatura.

Na Seção 8.3, são apresentados os resultados obtidos por MOGAIS em três problemas de otimização multiobjetivo: ZDT1, ZDT3 e ZDT6. Comparações foram realizadas com três algoritmos e os resultados são apresentados graficamente (fronteira de Pareto) e também numericamente por meio da métrica Cobertura de Dois Conjuntos.

8.2 Otimização Mono-objetivo

Para os experimentos com otimização mono-objetivo no espaço contínuo, foram selecionadas sete funções frequentemente utilizadas na literatura. No que segue, é apresentada a definição de cada uma delas. Para cada função, é apresentado também o gráfico para duas dimensões, no intuito de fornecer uma impressão da complexidade dos problemas.

- **Esfera**: esta função é unimodal e corresponde a uma hiperparábola definida a seguir:

$$F(x) = \sum_{i=1}^{d} x_i^2, \quad x_i \in [-5; 5]^d. \tag{8.1}$$

O valor do mínimo global, para qualquer número de dimensões, é zero, o qual é obtido quando todas as variáveis assumem valor zero. Na Figura 8.1, é apresentado o gráfico da função esfera para duas dimensões.

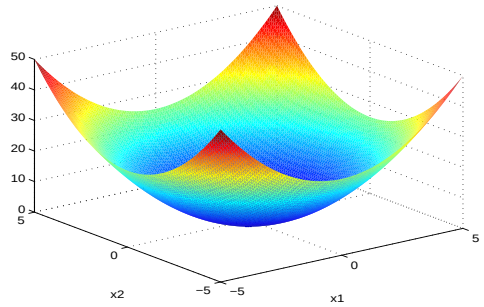


Figura 8.1: Gráfico da função esfera para duas dimensões.

- **Griewank**: é uma função que possui muitos ótimos locais. Ela é definida como:

$$F(x) = 1 + \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}), \quad x_i \in [-600; 600]^d. \tag{8.2}$$

O valor do mínimo global desta função, para qualquer número de dimensões, é zero, o qual é obtido quando todas as variáveis assumem valor zero. Na Figura 8.2, é ilustrado o gráfico da função Griewank para duas dimensões.

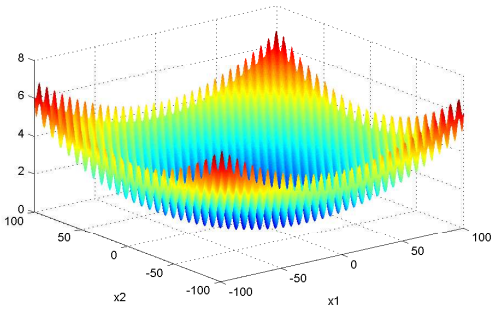


Figura 8.2: Gráfico da função Griewank para duas dimensões.

- **Rosenbrock**: é uma função bastante desafiadora para muitos algoritmos de otimização. Ela possui um vale longo, estreito e curvo, no qual está o ótimo global. A função é definida como:

$$F(x) = \sum_{i=1}^{d-1} 100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2, \quad x_i \in [-5, 12; 5, 12]^d. \tag{8.3}$$

O valor do ótimo global, para qualquer número de dimensões, é zero, o qual é obtido quando todas as variáveis assumem o valor um. O gráfico desta função é apresentado na Figura 8.3.

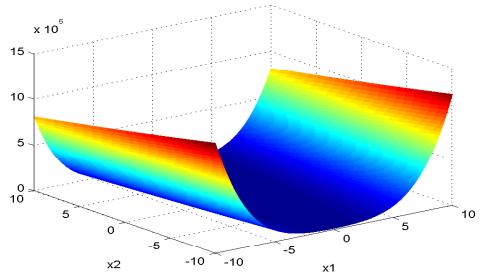


Figura 8.3: Gráfico da função Rosenbrock para duas dimensões.

- **Rastrigin**: esta função é baseada na função esfera, a qual recebeu um termo modulador para permitir a existência de muitos ótimos locais. A função Rastrigin é definida por:

$$F(x) = 10d \sum_{i=1}^{d-1} (x_i^2 - 10\cos(2\pi x_i)), \quad x_i \in [-5, 12; 5, 12]^d. \tag{8.4}$$

O valor do ótimo global, para qualquer número de dimensões, é zero, o qual é obtido quando todas as variáveis assumem o valor zero. Na Figura 8.4, é ilustrado o gráfico da função Rastrigin.

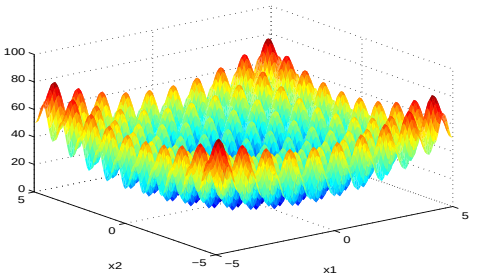


Figura 8.4: Gráfico da função Rastrigin para duas dimensões.

- **Schwefel**: é uma função composta por um grande número de picos e vales. O mínimo global se encontra distante dos mínimos locais, fazendo com que os algoritmos tendam a ficar presos nesses mínimos. Esta função é definida como:

$$F(x) = \sum_{i=1}^{d} -x_i \,\text{sen}(\sqrt{|x_i|}), \quad x_i \in [-500; 500]^d. \tag{8.5}$$

O valor do ótimo global é -418,9829d, o qual é obtido quando todas as variáveis assumem o valor 420,9687. O gráfico desta função para duas dimensões é apresentado na Figura 8.5.

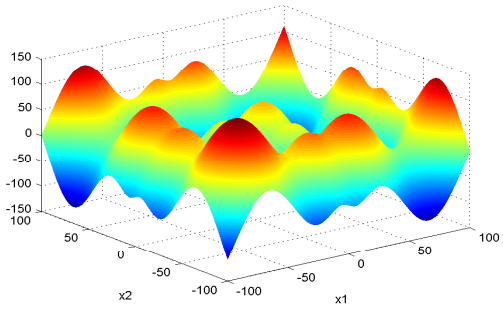


Figura 8.5: Gráfico da função Schwefel para duas dimensões.

- **Michalewicz**: esta função possui muitos ótimos locais e é definida por:

$$F(x) = -\sum_{i=0}^{d-1} \text{sen}(x_i)\text{sen}^{20}\left(\frac{(i+1)x_i^2}{\pi}\right), \quad x_i \in [0; \pi]^d. \tag{8.6}$$

O ótimo global depende do número de dimensões. Na Figura 8.6, é ilustrado o gráfico da função Michalewicz para duas dimensões.

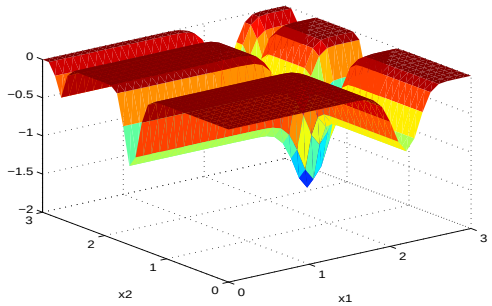


Figura 8.6: Gráfico da função Michalewicz para duas dimensões.

- **Ackley**: é uma função considerada de dificuldade moderada, a qual é definida por:

$$F(x) = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i}} - e^{\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)} + a + e^1, \quad x_i \in [-32,768; 32,768]^d. \tag{8.7}$$

O criador da função sugere utilizar os seguintes valores para os parâmetros: a = 20, b = 0,2 e c = 2π. O valor do ótimo global, para qualquer número de dimensões, é zero, o qual é obtido quando todas as variáveis assumem o valor zero. Na Figura 8.7, é ilustrado o gráfico da função Ackley para duas dimensões.

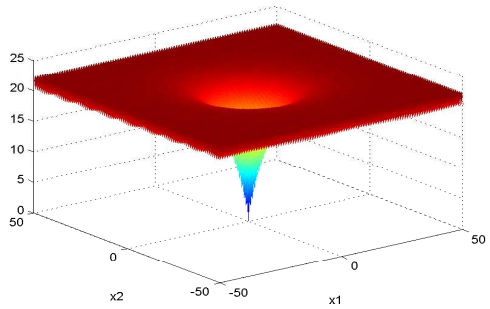


Figura 8.7: Gráfico da função Ackley para duas dimensões.

8.2.1 Resultados obtidos

Esta seção apresenta os resultados obtidos por GAIS quando aplicado na minimização das sete funções previamente apresentadas. Para todas as funções, foi utilizada a dimensão 30, com exceção da função Michalewicz, para a qual utilizou-se dimensão 10.

Os parâmetros do algoritmo são os mesmos para todos os casos. A população inicial contém 100 anticorpos e o critério de parada adotado foi o número máximo de iterações, definido como 200.

A versão do GAIS que utiliza modelo de mistura gaussiana, denotada aqui por GAIS$_M$, também foi avaliada. O algoritmo *k-means* foi utilizado para separar as melhores soluções em grupos e, em seguida, para cada grupo foi gerada uma rede gaussiana. O número de componentes da mistura (e, portanto, também o número de grupos criados pelo *k-means*) foi estabelecido empiricamente como 3. Os coeficientes de mistura foram definidos de acordo com a densidade de cada grupo (*cluster*). Para GAIS$_M$, a população inicial contém 300 anticorpos.

O *Iterated Density Estimation Algorithm* (IDEA), proposto por Bosman & Thierens (2000), é um algoritmo de estimação de distribuição que também utiliza modelo de mistura de gaussianas. Diferentemente do GAIS$_M$, o IDEA utiliza o algoritmo de agrupamento *Leader* (Hartigan, 1975).

Conforme sugerido pelo autor, o número de grupos foi definido como 4 e o tamanho da população como 2000.

Por fim, experimentos comparativos também foram realizados com um sistema imunológico artificial, chamado de *Artificial Immune Network for Optimization* (opt-aiNet) (de Castro & Timmis, 2002a). A população inicial contém 100 anticorpos e cada anticorpo gera 2 clones. O critério de parada é o número máximo de iterações, estabelecido como 200.

Os resultados médios obtidos em 30 execuções são apresentados na Tabela 8.1.

Tabela 8.1: Resultados médios obtidos por GAIS, GAIS$_M$, IDEA e opt-aiNet, para as sete funções em 30 execuções.

Função	GAIS	GAIS$_M$	IDEA	opt-aiNet
Esfera	**0** $\pm$ 0,000	**0** $\pm$ 0,000	**0** $\pm$ 0,000	**0** $\pm$ 0,000
Griewank	0,0058 $\pm$ 0,002	**0** $\pm$ 0,000	0,008 $\pm$ 0,022	0,082 $\pm$ 0,066
Rosenbrock	1,083 $\pm$ 0,005	0,9025 $\pm$ 0,018	**0,8420** $\pm$ 0,058	9,67 $\pm$ 1,322
Rastrigin	0,0028 $\pm$ 0,001	**0,0007** $\pm$ 0,004	0,0032 $\pm$ 0,003	1,96 $\pm$ 0,907
Schwefel	-12569,87 $\pm$ 0,048	**-12569,5** $\pm$ 0,000	-12569,32 $\pm$ 0,046	-8903,24 $\pm$ 112,773
Michalewicz	-9,6539 $\pm$ 0,016	**-9,6604** $\pm$ 0,027	-9,6257 $\pm$ 0,019	-9,0841 $\pm$ 0,086
Ackley	0,0011 $\pm$ 0,003	**0** $\pm$ 0,000	0,0008 $\pm$ 0,007	0,096 $\pm$ 0,002

Observando a Tabela 8.1, pode-se notar que os quatro algoritmos obtiveram resultados semelhantes, com uma ligeira vantagem para GAIS$_M$. Entretanto, frente ao custo computacional elevado que este apresenta, é preferível utilizar o GAIS.

8.3 Otimização Multiobjetivo

Para os experimentos com otimização multiobjetivo no espaço contínuo, foram escolhidos três problemas introduzidos por Zitzler et al. (2000). Os três problemas possuem a mesma estrutura:

$$\begin{array}{lrcl} \text{minimizar} & F(x) & = & [F_1(x_1), F_2(x)], \\ \text{sujeito a} & F_2(x) & = & G(x_2, ..., x_m)H(F_1(x_1), G(x_2, ..., x_m)), \\ \text{em que} & x & = & (x_1, ..., x_m). \end{array} \tag{8.8}$$

A função F_1 é uma função aplicada apenas na primeira variável, G é uma função aplicada nas variáveis restantes e os argumentos de H são os valores das funções F_1 e G. Os problemas diferem nessas funções (F_1, G e H), bem como no número de variáveis m e nos valores que as variáveis podem assumir.

O primeiro problema (ZDT1) possui uma fronteira de Pareto convexa e é dado por:

$$\begin{aligned} F_1(x_1) &= x_1, \\ G(x_2, ..., x_m) &= 1 + 9\sum_{i=2}^{m} x_i/(m-1), \\ H(F_1, G) &= 1 - \sqrt{F_1/G}. \end{aligned} \tag{8.9}$$

em que m=30 e $x_i \in [0, 1]$.

O segundo problema (ZDT3) possui uma fronteira de Pareto não-contínua e é dado por:

$$\begin{aligned} F_1(x_1) &= x_1 \\ G(x_2, ..., x_m) &= 1 + 9\sum_{i=2}^{m} x_i/(m-1) \\ H(F_1, G) &= 1 - \sqrt{F_1/G} - (F_1/G)\text{sen}(10\pi F_1) \end{aligned} \tag{8.10}$$

em que m=30 e $x_i \in [0, 1]$.

Por fim, o terceiro problema (ZDT6) é definido como:

$$\begin{aligned} F_1(x_1) &= 1 - e^{-4x_1}\text{sen}^6(6\pi x_1), \\ G(x_2, ..., x_m) &= 1 + 9((\sum_{i=2}^{m} x_i/(m-1))^{0,25}, \\ H(F_1, G) &= 1 - \sqrt{F_1/G}^2. \end{aligned} \tag{8.11}$$

em que m=10 e $x_i \in [0, 1]$.

8.3.1 Resultados

Com relação aos experimentos em problemas de otimização multiobjetivo, são apresentados apenas os resultados para MOGAIS sem modelo de mistura. A razão para isso é que ambos os algoritmos apresentaram resultados muito similares e, nesse caso, o algoritmo mais simples foi escolhido.

A população inicial de MOGAIS e o número máximo de iterações foram estabelecidos como 100. Os resultados obtidos por MOGAIS são comparados com aqueles gerados por três outros algoritmos: MIDEA, MISA e NSGA-II. O MIDEA é uma extensão do IDEA para lidar com problemas de otimização multiobjetivo. O tamanho da população é 700 e o número máximo de iterações é 500. O outro algoritmo é o *Multi-objective Immune System Algorithm* (MISA), proposto por Coello Coello & Cortés (2005), e que usa uma população secundária para implementar elitismo. O operador de mutação não-uniforme foi aplicado com alta probabilidade, mas diminui ao longo do processo de

otimização (de 0,9 para 0,3). A população contém 150 anticorpos e permanece fixa. O número de iterações é 500. O *Non-dominated Sorting Genetic Algorithm* (NSGA-II) (Deb et al., 2000) emprega seleção por *rank* e *crowding distance*. As taxas de cruzamento e mutação utilizadas foram 0,8 e 0,01, respectivamente, junto com seleção por torneio. A população contém 100 cromossomos e o número máximo de iterações foi definido como 250.

As fronteiras de Pareto encontradas pelos 4 algoritmos para os problemas ZDT1, ZDT3 e ZDT6 são apresentadas nas Figuras 8.8, 8.9 e 8.10, respectivamente. Para cada algoritmo, a fronteira de Pareto foi formada usando as soluções finais geradas em 10 execuções independentes do algoritmo. É possível verificar que todos os algoritmos produziram resultados similares para todos os casos, com soluções bem distribuídas ao longo da fronteira de Pareto. Foi observado que a identificação e preservação de blocos construtivos leva à uma rápida convergência. MOGAIS precisou de um menor número de iterações que seus concorrentes. Embora MIDEA também seja capaz de manipular blocos construtivos, seu desempenho foi inferior ao do MOGAIS. Provavelmente, isso aconteceu porque MOGAIS possui melhor mecanismo para explorar o espaço de busca.

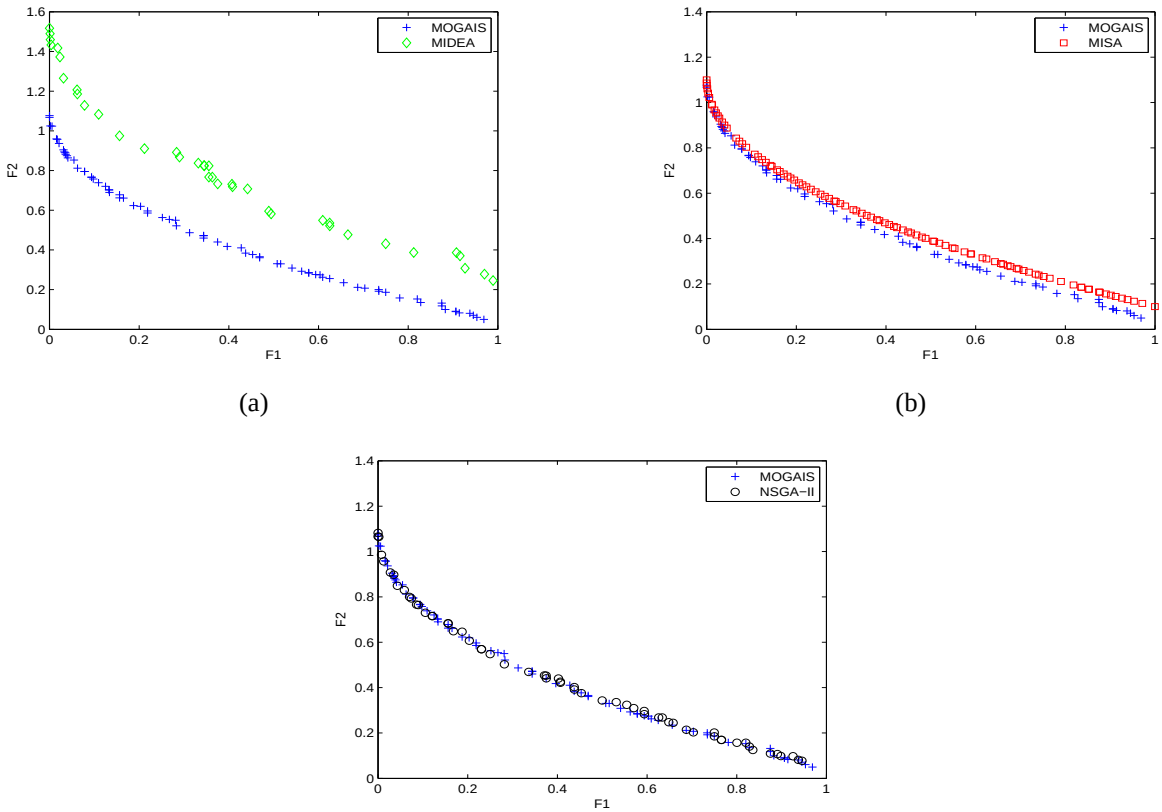
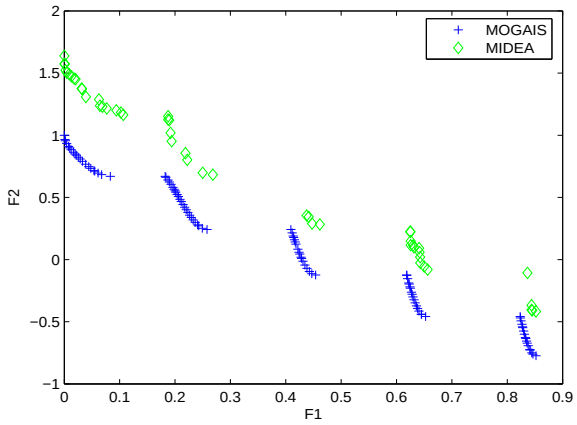
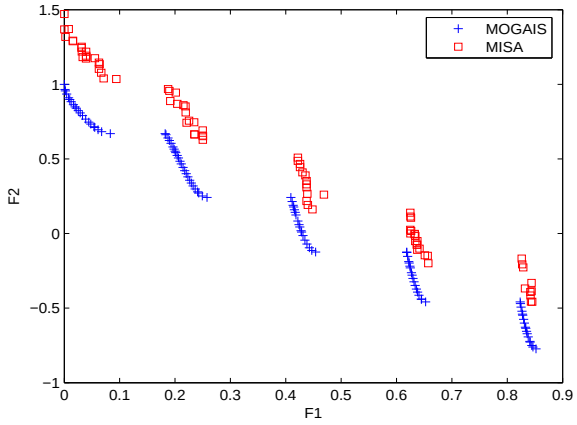


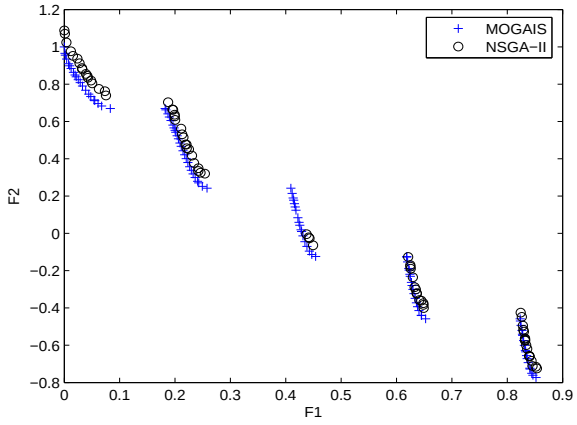
Figura 8.8: Comparação da fronteira de Pareto gerada pelos quatro algoritmos para o problema ZDT1: (a) MOGAIS e MIDEA, (b) MOGAIS e MISA e (c) MOGAIS e NSGA-II.



(a)

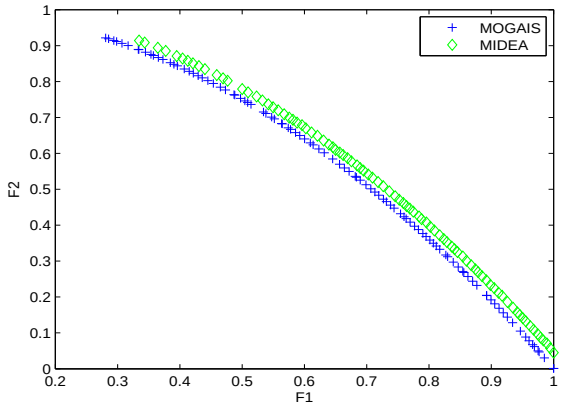


(b)

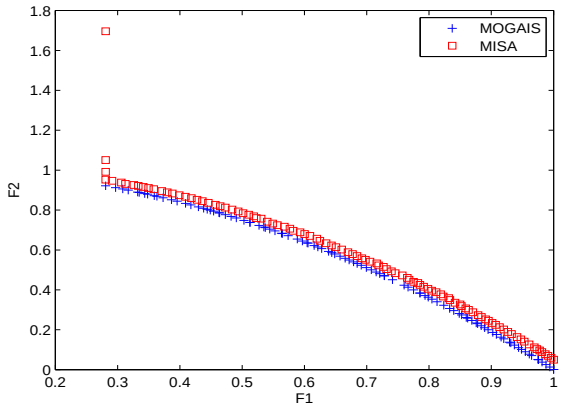


(c)

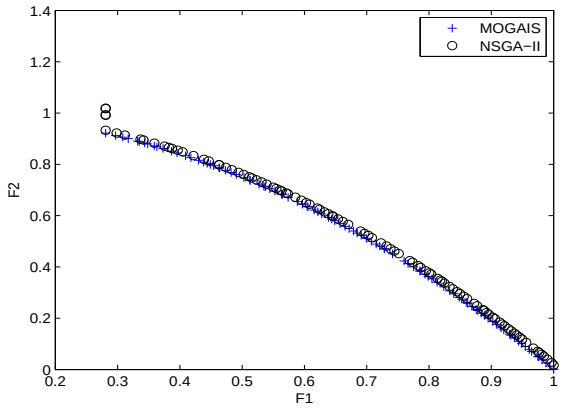
Figura 8.9: Comparação da fronteira de Pareto gerada pelos quatro algoritmos para o problema ZDT3: (a) MOGAIS e MIDEA, (b) MOGAIS e MISA e (c) MOGAIS e NSGA-II.



(a)



(b)



(c)

Figura 8.10: Comparação da fronteira de Pareto gerada pelos quatro algoritmos para o problema ZDT6: (a) MOGAIS e MIDEA, (b) MOGAIS e MISA e (c) MOGAIS e NSGA-II.

Além dos resultados gráficos, os algoritmos foram avaliados utilizando a métrica Cobertura de Dois Conjuntos (*Two Set Coverage*), proposta por Zitzler et al. (2000), a qual mostra que porcentagem de soluções de um algoritmo é dominada pelas soluções geradas por outro algoritmo concorrente. A métrica é dada por:

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X' : a' \preceq a''|}{|X''|} \tag{8.12}$$

em que $X', X'' \subseteq X$ são dois conjuntos de soluções. O resultado da Equação (8.12) é um número real no intervalo [0,1]. Isso significa que C=1 quando X' domina ou é igual a X''. Repare que ambos os cálculos $C(X', X'')$ e $C(X'', X')$ precisam ser considerados, uma vez que $C(X', X'')$ não é necessariamente igual a $1 - C(X'', X')$.

Na Tabela 8.2 são reportados os valores dessa métrica para cada problema.

Tabela 8.2: Valores para a métrica Cobertura de Dois Conjuntos (A $\preceq$ B).

Algoritmo A	Algoritmo B	ZDT1	ZDT3	ZDT6
MOGAIS	MIDEA	100%	100%	100%
	MISA	92%	100%	96%
	NSGA-II	17%	34%	8%
MIDEA	MOGAIS	0%	0%	0%
	MISA	0%	32%	9%
	NSGA-II	0%	0%	3%
MISA	MOGAIS	8%	0%	6%
	MIDEA	100%	52%	78%
	NSGA-II	21%	0%	1%
NSGA-II	MOGAIS	12%	21%	0%
	MIDEA	100%	100%	89%
	MISA	86%	100%	62%

Pela Tabela 8.2, pode-se observar que as soluções geradas por MOGAIS e NSGA-II obtiveram boa cobertura sobre as soluções geradas pelos outros algoritmos. Em contraste, as soluções geradas por eles são dominadas por algumas poucas soluções.

8.4 Considerações Finais

Neste capítulo, foram descritos alguns experimentos realizados com GAIS e MOGAIS em problemas de otimização mono-objetivo e multiobjetivo, respectivamente, no espaço contínuo. Para cada caso, foram realizadas também comparações com outros algoritmos conhecidos na literatura.

GAIS e MOGAIS produziram resultados de boa qualidade em todos os problemas de otimização testados em poucas iterações, enquanto que os outros algoritmos precisaram de uma quantidade maior de iterações para alcançar soluções similares.

A versão do GAIS que utiliza modelo de mistura de gaussianas, denotado por GAIS$_M$, também foi avaliada em problemas de otimização mono-objetivo. Embora bons resultados tenham sido obtidos, a diferença para os resultados produzidos pelo GAIS não foi tão significativa, ao ponto de justificar seu uso. Entretanto, são necessárias maiores investigações sobre o uso de modelo de mistura nos algoritmos propostos.

Capítulo 9

Conclusões e Perspectivas Futuras

9.1 Considerações Finais

No campo da inteligência computacional, os sistemas imunológicos artificiais e os modelos gráficos probabilísticos representam duas importantes técnicas disponíveis para a construção de sistemas inteligentes. Por isso elas têm sido amplamente exploradas por pesquisadores das mais diversas áreas, tanto no aspecto teórico quanto prático.

Entretanto, na maioria das vezes o potencial de cada técnica é abordado isoladamente, sem levar em consideração a possível cooperação entre eles. Muito poucos trabalhos se dedicaram a estudar em quais aspectos dos modelos gráficos probabilísticos os sistemas imunológicos artificias poderiam contribuir. Por outro lado, não surgiu nenhuma contribuição no sentido inverso, além das propostas apresentadas nesta tese.

Com isto exposto, o objetivo desta tese foi investigar a sinergia existente entre modelos gráficos probabilísticos e sistemas imunológicos artificiais. Para facilitar o entendimento, é conveniente decompor o objetivo em duas frentes. Em um primeiro momento, o objetivo foi empregar um sistema imunológico artificial como mecanismo de busca na tarefa de aprendizado de redes bayesianas a partir de dados amostrados. Na segunda parte da tese, os modelos gráficos probabilísticos são aplicados para melhorar o desempenho dos sistemas imunológicos artificiais, permitindo que eles lidem eficientemente com blocos construtivos. Isso foi realizado pela substituição dos operadores de clonagem e mutação pela etapa de aprendizado de um modelo gráfico probabilístico que representa a distribuição de probabilidade das melhores soluções, seguida da amostragem de novas soluções a partir do modelo obtido.

As principais contribuições desta tese estão descritas ao longo dos capítulos que a compõem.

No Capítulo 4, foi descrita e analisada a metodologia proposta para aprendizado de redes bayesianas por meio de sistemas imunológicos artificiais. Durante os experimentos preliminares

conduzidos nos *benchmarks* ASIA e ALARM, resultados promissores foram obtidos, indicando a viabilidade da metodologia. Em seguida, a tarefa de seleção de atributos em problemas de classificação de padrões foi tema dos experimentos. O sistema imunológico artificial proposto foi aplicado aos conjuntos de dados, a fim de gerar uma rede bayesiana. Subsequentemente, os nós que compõem o *Markov blanket* do nó que representa a classe foram tomados como os atributos selecionados. Os resultados mostraram que subconjuntos reduzidos de atributos foram obtidos, sem degradar o desempenho do classificador e, até mesmo, melhorando a taxa de classificação.

A maior contribuição deste trabalho foi descrita no Capítulo 5. Nele, foram apresentados quatro novos sistemas imunológicos artificiais para resolver problemas de otimização mono-objetivo e multiobjetivo, tanto no espaço discreto como no espaço contínuo. Estes algoritmos, em vez de evoluir as soluções candidatas por meio de clonagem e mutação, utilizam modelos probabilísticos que representam a distribuição de probabilidade das melhores soluções encontradas até então. Os modelos probabilísticos adotados foram as redes bayesianas (caso discreto) e as redes gaussianas (caso contínuo), por apresentarem a capacidade de representar adequadamente interações complexas das variáveis do problema, permitindo a manipulação eficiente de blocos construtivos. Dentre as características dos quatro algoritmos propostos, merecem destaque: habilidade para identificar e preservar os blocos construtivos, tamanho da população automaticamente ajustável, manutenção de diversidade na população, capacidade de realizar busca multimodal. Como uma consequência direta destas características, têm-se: a busca torna-se mais robusta, no sentido de que não há muita variação de desempenho entre buscas derivadas de re-execuções do algoritmo para um mesmo problema; e tende a ocorrer uma redução acentuada no número de avaliações de soluções candidatas até atingir um certo nível de desempenho. Os principais trabalhos relacionados foram apresentados e as quatro propostas desta tese foram devidamente posicionadas frente à literatura.

Nos Capítulos 6, 7 e 8, os algoritmos foram avaliados junto a problemas práticos, como geração de *ensembles* de redes neurais, seleção de atributos em problemas de classificação de padrões e otimização no espaço contínuo. Os resultados obtidos foram comparados com aqueles produzidos por outros algoritmos encontrados na literatura.

9.2 Trabalhos Futuros

Todas as propostas apresentadas nesta tese abrem possibilidades de investigação futura.

No que diz respeito ao aprendizado de redes bayesianas por meio de sistemas imunológicos artificiais, as seguintes sugestões são apresentadas:

- Durante a criação da população inicial e na etapa de mutação, é preciso evitar a criação de redes cíclicas. Atualmente, para cada ciclo identificado, uma aresta escolhida aleatoriamente

é removida ou invertida. Entretanto, um processo de busca local poderia ser executado para definir otimamente qual aresta deve ser removida;

- Realizar experimentos com conjuntos de dados que possuem muitas variáveis, como por exemplo, dados de bioinformática;
- Investigar o uso de outras métricas para avaliação das soluções candidatas;
- Permitir que o algoritmo seja capaz de aprender redes bayesianas que apresentem não apenas correlações entre as variáveis, mas também relações de causa-efeito;
- Realizar mais experimentos em diferentes áreas e estabelecer comparações com outros algoritmos da literatura.

Referente aos sistemas imunológicos artificiais baseados em modelos gráficos probabilísticos, as seguintes melhorias e variações são sugeridas como trabalhos futuros:

- A construção do modelo probabilístico a cada iteração do algoritmo representa um custo computacional relativamente alto, embora isso seja compensado pela vantagens proporcionadas. Uma possível maneira de amenizar este custo computacional seria gerar a estrutura do modelo probabilístico a cada n iterações, mantendo a atualização dos parâmetros numéricos a cada iteração;
- Utilização de outros modelos probabilísticos, como redes de Markov;
- Utilização de outras métricas para gerar o modelo probabilístico, bem como outras técnicas para realizar a amostragem de novas soluções a partir do modelo obtido;
- Extensão dos algoritmos para lidar com problemas de otimização dinâmica.

Referências Bibliográficas

Acid, S. & Campos, L. M. (1996). "An algorithm for finding minimum d-separating sets in belief networks". In *Proc. of 12th Conference of Uncertainty in Artificial Intelligence*, volume 1 (pp. 3–10).

Ada, G. L. & Nossal, G. J. V. (1987). "The clonal selection theory". *Scientific American*, 257(2), 50–57.

Ahn, C., Ramakrishna, R., & Goldberg, D. (2006). "Real-coded Bayesian optimization algorithm". In J. A. Lozano, P. Larrañaga, I. Inza, & E. Bengoetxea (Eds.), *Towards a New Evolutionary Computation* (pp. 51–73). Springer.

Aickelin, U., Greensmith, J., & Twycross, J. (2004). "Immune system approaches to intrusion detection - A review". In G. Nicosia, V. Cutello, P. J. Bentley, & J. Timmis (Eds.), *Proceedings of the Third International Conference on Artificial Immune Systems* (pp. 316–329).: Springer.

Akaike, H. A. (1974). "A new look at the statistical model identification". *IEEE Transactions on Automatic Control*, AC-19, 716–723.

Almuallim, H. & Dietterich, T. G. (1991). "Learning with many irrelevant features". In *Proc. of the 9th National Conf. on Artificial Intelligence*, volume 2 (pp. 547–552).

Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., & Spyropoulos, C. (2000). "An evaluation of naive Bayesian anti-spam filtering". In *Proc. of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning* (pp. 9–17).

Angeline, P. J., Saunders, G. M., & Pollack, J. P. (1994). "An evolutionary algorithm that constructs recurrent neural networks". *IEEE Transactions on Neural Networks*, 5(1), 54–65.

Antal, P., Fannes, G., Timmerman, D., De Moor, B., & Moreau, Y. (2003). "Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor 24 classification with rejection". *Artificial Intelligence in Medicine*, 29, 39–60.

Asunción, A. & Newman, D. (2007). UCI Machine Learning Repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Bala, J., DeJong, K., Huang, J., Vafaie, H., & Wechsler, H. (1996). "Using learning to facilitate the evolution of features for recognizing visual concepts". *Evolutionary Computation*, 4(3), 297–311.

Baluja, S. (1994). "*Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*". Technical report, Carnegie Mellon University, Pittsburgh, PA, USA.

Baluja, S. & Davies, S. (1997). "Using optimal dependency-trees for combinational optimization". In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 30–38). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Beinlich, I., Suermondt, H., Chavez, R., & Cooper, G. (1989). "The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks". In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine* (pp. 247–256).: Springer Verlag.

Blanco, R., Inza, I., & Larrañaga, P. (2004). "Learning Bayesian networks by floating search methods". In J. Gámez, S. Moral, & A. Salmerón (Eds.), *Advances in Bayesian Networks* (pp. 181–200). Physica Verlag.

Blum, A. L. & Langley, P. (1997). "Selection of relevant features and examples in machine learning". *Artificial Intelligence*, 97(1-2), 245 – 271.

Bosman, P. & Thierens, D. (2000). "Expanding from discrete to continuous estimation of distribution algorithms: The IDEA". In *Parallel Problem Solving From Nature - PPSN*, volume 6 (pp. 767–776).: Springer.

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, California, U.S.A.: Wadsworth Publishing Company.

Brotherton, T. & Simpson, P. (1995). "Dynamic feature set training of neural nets for classification". In J. R. McDonnell, R. G. Reynolds, & D. B. Fogel (Eds.), *Evolutionary Programming*, volume 4 (pp. 83–94). MIT Press.

Buntine, W. (1994). "Operations for learning with graphical models". *Journal of Artificial Intelligence Research*, 2, 159–225.

Buntine, W. (1996). "A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8, 195–210.

Burnet, F. (1959). *The Clonal selection theory of acquired immunity*. Nashville TN: Vanderbilt Univ. Press.

Campos, L. M., Fernández-Luna, J., Gámez, J., & Puerta, J. (2002). "Ant colony optimization for learning bayesian networks". *International Journal on Artificial Reasoning*, 3(31), 291–311.

Cantú-Paz, E. (2002). "Feature subset selection by Estimation of Distribution Algorithms". In *Proc. of the Genetic and Evolutionary Computation Conference* (pp. 303–310).

Castro, P. A. D., Coelho, G. P., Caetano, M. F., & Von Zuben, F. J. (2005). "Designing ensembles of fuzzy classification systems: An immune-inspired approach". In *Proceedings of the 4th International Conference on Artificial Immune Systems* (pp. 469–482).

Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007a). "Applying biclustering to perform collaborative filtering". In *Proc. Seventh International Conference on Intelligent Systems Design and Applications ISDA 2007* (pp. 421–426).

Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007b). "Evaluating the performance of a biclustering algorithm applied to collaborative filtering - A comparative analysis". In *Proc. 7th International Conference on Hybrid Intelligent Systems HIS 2007* (pp. 65–70).

Castro, P. A. D., Santoro, D. M., Camargo, H. A., & Nicoletti, M. C. (2004). "Improving a Pittsburgh learnt fuzzy rule base using feature subset selection". In *Proc. of the 4th Int. Conf. on Hybrid Intelligent Systems* (pp. 180–185).

Castro, P. A. D. & Von Zuben, F. J. (2005). "An immune-inspired approach to Bayesian networks". In *Proceedings of the 5th International Conference on Hybrid Intelligent Systems* (pp. 23–28).: IEEE Computer Society.

Castro, P. A. D. & Von Zuben, F. J. (2006). "Bayesian learning of neural networks by means of artificial immune systems". In *Proc. of the International Joint Conference on Neural Networks IJCNN '06* (pp. 4831–4838).

Castro, P. A. D. & Von Zuben, F. J. (2008a). "Feature subset selection by means of a Bayesian artificial immune system". In *Proc. Eighth International Conference on Hybrid Intelligent Systems HIS '08* (pp. 561–566).

Castro, P. A. D. & Von Zuben, F. J. (2008b). "MOBAIS: A Bayesian artificial immune system for multi-objective optimization". In P. J. Bentley, D. Lee, & S. Jung (Eds.), *ICARIS*, volume 5132 of *Lecture Notes in Computer Science* (pp. 48–59).: Springer.

Castro, P. A. D. & Von Zuben, F. J. (2009a). "BAIS: A Bayesian artificial immune system for the effective handling of building blocks". *Information Sciences*, 179(10), 1426 – 1440.

Castro, P. A. D. & Von Zuben, F. J. (2009b). "Learning Bayesian networks to perform feature selection". In *Proc. of the IEEE 2009 International Joint Conference on Neural Networks* (pp. 467–473).

Castro, P. A. D. & Von Zuben, F. J. (2009c). "Multi-objective Bayesian artificial immune system: Empirical evaluation and comparative analyses". *Journal of Mathematical Modelling and Algorithms*, 1, 151–173.

Chen, J. & Mahfouf, M. (2006). "A population adaptive based immune algorithm for solving multi-objective optimizaton problems". In H. Bersini & J. Carneiro (Eds.), *Proceedings of the 5th International Conference on Artificial Immune Systems* (pp. 280–293).: Springer.

Cheng, J., Bell, D. A., & Liu, W. (1997). "Learning belief networks from data: An information theory based approach". In *Proceedings of the sixth international conference on Information and knowledge management* (pp. 325–331). New York, NY, USA: ACM Press.

Chickering, D. M. (2002). "Learning equivalence classes of Bayesian network structures". *Journal of Machine Learning Research*, 2, 445–498.

Chickering, D. M., Geiger, D., & Heckerman, D. (1995). "Learning Bayesian networks: Search methods and experimental results". In *Preliminary papers of the 5th International Workshop on Artificial Intelligence and Statistics* (pp. 112–128).

Chow, C. J. & Liu, C. N. (1968). "Approximating discrete probability distributions with dependence trees". *IEEE Trans. on Information Theory*, 14(3), 462–467.

Chun, J. S., Jung, H. K., & Hahn, S. Y. (1998). "A study on comparison of optimization performances between immune algorithm and other heuristic algorithms". *IEEE Transactions on Magnetics*, 34(5), 2972–2975.

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* (pp. 294–308).: Springer.

Coello Coello, C. A. (1998). "A comprehensive survey of evolutionary-based multiobjective optimization techniques". *Knowledge and Information Systems*, 1, 269–308.

Coello Coello, C. A. & Cortés, N. C. (2005). "Solving multiobjective optimization problems using an artificial immune system". *Genetic Programming and Evolvable Machines*, 6(2), 163–190.

Cooper, G. & Herskovits, E. (1992). "A Bayesian method for the induction of probabilistic networks from data". *Machine Learning*, 9, 309–347.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. Springer.

Das, S. (2001). "Filters, wrappers and a boosting-based hybrid for feature selection". In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 74–81). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

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

De Bonet, J. S., Isbell, C. L., & Viola Jr., P. (1997). "MIMIC: Finding optima by estimating probability densities". In *Advances in Neural Information Processing Systems* (pp. 424).

de Castro, L. N. & Timmis, J. (2002a). "An artificial immune network for multimodal function optimization". In *Proc. of the 2002 Congress on Evolutionary Computation (CEC '02)* (pp. 699–704).

de Castro, L. N. & Timmis, J. (2002b). *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag.

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

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 2005 Conference on Genetic and Evolutionary Computation* (pp. 289–296).: ACM Press.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc.

Deb, K., Agrawal, S., Pratab, A., & Meyarivan, T. (2000). "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". In *Proceedings of the Parallel Problem Solving from Nature VI Conference* (pp. 849–858).

Deb, K. & Goldberg, D. E. (1993). "Analysing deception in trap functions". *Foundations of Genetic Algorithms*, 2, 93–108.

del Sagrado, J. & Moral, S. (2003). "Qualitative combination of Bayesian networks". *International Journal of Intelligent Systems*, 18(2), 237–249.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). "Supervised and unsupervised discretization of continuous features". In *Proceedings of the 12th International Conference on Machine Learning* (pp. 194–202).

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

Etxeberria, R., Larrañaga, P., & Picaza, J. M. (1997). "Analysis of the behaviour of genetic algorithms when learning bayesian network structure from data". *Pattern Recogn. Lett.*, 18(11-13), 1269–1273.

Friedman, N. (1997). "Learning belief networks in the presence of missing values and hidden variables". In *Proc. 14th International Conference on Machine Learning* (pp. 125–133).: Morgan Kaufmann.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). "Using Bayesian networks to analyze expression data". *Proc. Fourth Annual International Conference on Computational Molecular Biology*, 1, 127–135.

García-Pedrajas, N. & Fyfe, C. (2007). "Immune network based ensembles". *Neurocomputing*, 70(7-9), 1155–1166.

García-Pedrajas, N., Hervás-Martínez, C., & Ortiz-Boyer, D. (2005). "Cooperative coevolution of artificial neural network ensembles for pattern classification". *IEEE Transactions on Evolutionary Computation*, 9(3), 271–302.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Goldberg, D. E. & Richardson, J. J. (1987). "Genetic algorithms with sharing for multimodal function optimization". In *Proceedings of the 2nd International Conference on Genetic Algorithms* (pp. 41–49).

Grünwald, P. (2000). "Model selection based on minimum description length". *Journal of Mathematical Psychology*, 44, 133–152.

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

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). "The compact genetic algorithm". *IEEE Trans. on Evolutionary Computation*, 3(4), 287–297.

Harik, G. R., Lobo, F. G., & Sastry, K. (2006). "Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA)". In M. Pelikan, K. Sastry, & E. Canty-Paz (Eds.), *Scalable Optimization via Probabilistic Modeling* (pp. 39–61). Springer.

Hartigan, J. A. (1975). *Clustering algorithms*. Wiley New York.

Heckerman, D. (1995). *"A Tutorial on Learning Bayesian Networks"*. Technical Report MSR-TR-95-06.

Heckerman, D., Geiger, D., & Chickering, D. (1995). "Learning Bayesian networks: The combination of knowledge and statistical data". *Machine Learning*, 20, 197–243.

Henrion, M. (1988). "Propagating uncertainty in Bayesian networks by probabilistic logic sampling". In J. F. Lemmer & L. N. Kanal (Eds.), *Uncertainty in Artificial Intelligence*, volume 2 (pp. 149–163). New York, USA: Elsevier Science Publishing Company, Inc.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The MIT Press.

Horvitz, E., Ruan, Y., Gomes, C. P., Kautz, H., Selman, B., & Chickering, D. M. (2001). "A Bayesian approach to tackling hard computational problems". In *Proc. of 17th Conference on Uncertainty and Artificial Intelligence* (pp. 235–244).

Hruschka Jr, E. R. & Ebecken, N. F. F. (2003). "Variable ordering for Bayesian networks learning from data". In *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA 2003* (pp. 1–9). Vienna.

Hruschka Jr, E. R., Hruschka, E. R., & Ebecken, N. F. F. (2004). "Feature selection by Bayesian networks". In *Proceedings of the 7h Canadian Conference on Artificial Intelligence*, volume 3060 (pp. 370–379).: Springer-Verlag.

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

Imoto, S., Kim, S., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003). "Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network". *Journal of Bioinformatics and Computational Biology*, 1(2), 231–252.

Inza, I., Larrañaga, P., Etxeberria, R., & Sierra, B. (2000). "Feature subset selection by Bayesian network-based optimization". *Artificial Intelligence*, 123, 157–184.

Iyoda, E. M. & Von Zuben, F. J. (2002). "Hybrid neural networks: An evolutionary approach with local search". *Integrated Computer-Aided Engineering*, 9(1), 57–72.

Jensen, F. V. (1996). *"An Introduction to Bayesian Networks"*. London: UCL press.

Jensen, F. V. (2001). *"Bayesian Networks and Decision Graphs"*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

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 *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems* (pp. 130–139). Cambridge, MA, US: MIT Press.

Khan, N., Goldberg, D. E., & Pelikan, M. (2002). *Multi-Objective Bayesian Optimization Algorithm*. Technical report, Illigal Report 2002009.

Kikuchi, R. (1951). "A theory of cooperative phenomena". *Physical Review*, 81(6), 988.

Kira, K. & Rendell, L. A. (1992). "A practical approach to feature selection". In *Proc. of the 9th Int. Workshop on Machine Learning* (pp. 249–256).

Kohavi, R. & John, G. H. (1997). "Wrappers for feature subset selection". *Artificial Intelligence*, 97(1-2), 273–324.

Kononenko, I. (1994). "Estimating attributes: Analysis and extensions of RELIEF". In *Proc. of the European Conference on Machine Learning* (pp. 171–182).

Krogh, A. & Vedelsby, J. (1994). "Neural network ensembles, cross validation, and active learning". In D. T. G. Tesauro & T. Leen (Eds.), *Advances in Neural Information Processing System*, volume 7 (pp. 231–238).: MIT Press.

Lam, W. & Bacchus, F. (1994). "Learning Bayesian belief networks: An approach based on the MDL principle". *Computational Intelligence*, 10(4), 269–293.

Larrañaga, P., Etxeberria, R., Lozano, J. A., & Peña, J. M. (2000a). "Combinatorial optimization by learning and simulation of Bayesian networks". In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (pp. 343–352).

Larrañaga, P., Etxeberria, R., Lozano, J. A., & Peña, J. M. (2000b). "Optimization in continuous domains by learning and simulation of Gaussian networks". In *Proceedings of the GECCO-2000 Workshop in Optimization by Building and Using Probabilistic Models* (pp. 201–204).

Larrañaga, P., Kuijpers, C., Murga, R., & Yurramendi, Y. (1996). "Learning bayesian network structures by searching for the best ordering with genetic algorithms". *IEEE Transactions on System, Man and Cybernetics*, 26(4), 487–493.

Laumanns, M. & Ocenasek, J. (2002). "Bayesian optimization algorithms for multi-objective optimization". In *Parallel Problem Solving From Nature - PPSN*, volume 7 (pp. 298–307).

Lauritzen, L. S. & Spiegelhalter, D. J. (1988). "Local computations with probabilities on graphical structures and their application to expert systems". *Journal of Royal Statistics Society B*, 50, 157–194.

Liu, H. & Yu, L. (2005). "Toward integrating feature selection algorithms for classification and clustering". *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 491–502.

Liu, Y. & Yao, X. (1996). "Evolutionary design of artificial neural networks with different node transfer functions". In *Proc. of the 3rd IEEE Intern. Conf. on Evolutionary Computation* (pp. 670–675).

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

Liu, Y., Yao, X., Zhao, Q., & Higuchi, T. (2001). "Evolving a cooperative population of neural networks by minimizing mutual information". In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001* (pp. 384–389).: IEEE Press.

McMorris, F. & Neumann, D. (1983). "Consensus functions defined on trees. *Mathmatical Social Sciences*, 4(2), 131–136.

Mühlenbein, H. & Mahnig, T. (1999). "The factorized distribution algorithm for additively decomposed functions". In *proc. of the 1999 Congress on Evolutionary Computation* (pp. 752–759).

Mühlenbein, H. & Paass, G. (1996). "From recombination of genes to the estimation of distributions i. binary parameters". In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature* (pp. 178–187). London, UK: Springer-Verlag.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.

Moller, M. F. (1993). "A scaled conjugate gradient algorithm for fast supervised learning". *Neural Networks*, 6(4), 525–533.

Myers, J. W., Laskey, K. B., & DeJong, K. A. (1999). "Learning Bayesian networks from incomplete data using evolutionary algorithms". In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, & R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1 (pp. 458–465). Orlando, Florida, USA: Morgan Kaufmann.

Nariai, N., Kim, S., Imoto, S., & Miyano, S. (2004). "Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks". *Proc. of Symposium on Biocomputation*, 9, 336–347.

Opitz, D. & Shavlik, J. (1996). "Generating accurate and diverse members of a neural network ensemble". In *Advances in Neural Information Processing Systems* (pp. 535–541).: MIT Press.

Papoulis, A. & Pillai, S. U. (2002). *Probability, Random Variables, and Stochastic Processes*. Boston, MA: McGraw-Hill, 4th edition.

Pasti, R. & de Castro, L. N. (2007). "The influence of diversity in an immune-based algorithm to train MLP networks". In *Proc. of the 6th International Conference on Artificial Immune Systems* (pp. 71–82).

Pasti, R. & de Castro, L. N. (2009). "Bio-inspired and gradient-based algorithms to train MLPs: The influence of diversity". *Information Sciences*, 179(10), 1441–1453.

Paul, T. K. & Iba, H. (2003). "Reinforcement learning estimation of distribution algorithm". In *Proceedings of the Genetic and Evolutionary Computation Conference 2003 (GECCO2003)* (pp. 1259–1270).: Springer-Verlag.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Pelikan, M. (2005). *Probabilistic Model-Building Genetic Algorithms*. Springer.

Pelikan, M., Goldberg, D., & Cantú-Paz, E. (2000). "Linkage problem, distribution estimation and bayesian networks". *Evolutionary Computation*, 8(2), 311–340.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). "BOA: The Bayesian optimization algorithm". In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, & R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I (pp. 525–532). Orlando, FL: Morgan Kaufmann Publishers, San Fransisco, CA.

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (2002). "A survey of optimization by building and using probabilistic models". *Comput. Optim. Appl.*, 21(1), 5–20.

Pelikan, M. & Mühlenbein, H. (1999). "The bivariate marginal distribution algorithm". In R. Roy, T. Furuhashi, & P. K. Chawdhry (Eds.), *Advances in Soft Computing - Engineering Design and Manufacturing* (pp. 521–535). London: Springer-Verlag.

Pennock, D. M. & Wellman, M. P. (1999). "Graphical representations of consensus belief". In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence* (pp. 531–540).

Perrone, M. & Cooper, L. N. (1993). "When networks disagree: Ensemble method for neural networks". In R. 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.

Punch, W. F., Goodman, E. D., Pei, M., Chia-Shun, L., Hovland, P., & Enbody, R. (1993). "Further research on feature selection and classification using genetic algorithms". In *Proc. of the 5th Int. Conf. on Genetic Algorithms* (pp. 557–564).

Rakotomamonjy, A. (2003). "Variable selection using SVM based criteria". *J. Mach. Learn. Res.*, 3, 1357–1370.

Raval, A., Ghahramani, Z., & Wild, D. L. (2002). "A Bayesian network model for protein fold and remote homologue recognition". *Bioinformatics*, 18(6), 88–780.

Ripley, B. D. (1987). *Stochastic simulation*. New York, NY, USA: John Wiley & Sons, Inc.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). "A review of feature selection techniques in bioinformatics". *Bioinformatics*, 23(19), 2507–2517.

Santana, R. (2005). "Estimation of Distribution Algorithms with Kikuchi approximations". *Evolutionary Computation*, 13(1), 67–97.

Sastry, K., Abbass, H. A., Goldberg, D. E., & Johnson, D. D. (2005). "Sub-structural niching in estimation of distribution algorithms". In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO)* (pp. 671–678). New York, NY, USA: ACM.

Schapire, R. (1990). "The strength of weak learnability". *Machine Learning*, 5(2), 197–227.

Schwarz, G. (1978). "Estimating the dimension of a model". *Annals of Statistics*, 6, 461–465.

Scroferneker, M. L. & Pohlmann, P. R. (1998). *Imunologia Básica e Aplicada*. Sagra Luzzato.

Sebag, M. & Ducoulombier, A. (1998). "Extending population-based incremental learning to continuous search spaces". In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature* (pp. 418–427). London, UK: Springer-Verlag.

Shachter, R. D. & Kenley, C. R. (1989). "Gaussian influence diagrams". *Management Science*, 35(5), 527–550.

Shakya, S., McCall, J., & Brown, D. (2004). "Updating the probability vector using MRF technique for a univariate eda". In *Proceedings of the 4th European Starting AI Researcher Symposium* (pp. 15–25).

Shakya, S., McCall, J., & Brown, D. (2005). "Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis". In *Proc. of the 2005 Conference on Genetic and Evolutionary Computation (GECCO)* (pp. 727–734).: ACM.

Siedlecki, W. & Sklansky, J. (1989). "A note on genetic algorithms for large-scale feature selection". *Pattern Recogn. Lett.*, 10(5), 335–347.

Skalak, D. (1996). "The sources of increased accuracy for two proposed boosting algorithms". In *AAAI '96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*.

Spirtes, P., Glymour, C., & Scheines, R. (1991). "An algorithm for fast recovery of sparse causal graphs". *Social Science Computer Review*, 9, 62–72.

Srinivas, S., Russell, S., & Agogino, A. (1990). "Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information". In *Proc. of the 5th Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)* (pp. 295–308). New York, NY: Elsevier Science Publishing Company, Inc.

Stephenson, T., Bourlard, H., Bengio, S., & Morris, A. (2000). "Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables". In *International Conference on Spoken Language Processing, October 2000, vol. II* (pp. 951–954).

Suzuki, J. (1996). "Learning Bayesian belief networks based on the MDL principle: An efficient algorithm using the branch and bound technique". *Proc. of International Conference on Machine Learning*, 1, 462–470.

Theodoridis, S. & Koutroumbas, K. (2006). *Pattern Recognition, Third Edition*. Orlando, FL, USA: Academic Press, Inc.

Vafaie, H. & DeJong, K. (1993). "Robust feature selection algorithms". In *Proc. 5th Intl. Conf. on Tools with Artificial Intelligence* (pp. 356–363).

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright-Patterson AFB, OH.

Vehtari, A. & Lampinen, J. (2000). "Bayesian MLP neural networks for image analysis". *Pattern Recognition Letters*, 21(13-14), 1183–1191.

Watkins, A., Timmis, J., & Boggess, L. (2004). "Artificial immune recognition system (AIRS): An immune-inspired supervised machine learning algorithm". *Genetic Programming and Evolvable Machines*, 5(3), 291–317.

Wermuth, N. & Lauritzen, S. (1983). "Graphical and recursive models for contingency tables". *Biometrika*, 72, 537–552.

Yao, X. (1999). "Evolving artificial neural networks". *Proceedings of the IEEE*, 87, 1423–1447.

Zeleny, M. (1973). "Compromise programming". In J. Cochrane & M. Zeleny (Eds.), *Multiple Criteria Decision Making* (pp. 262–301). Columbia: University of South Carolina Press.

Zhang, X., Wang, S., Shan, T., & Jiao, L. (2005). "Selective SVMs ensemble driven by immune clonal algorithm". In *Proceedings of EvoWorkshops* (pp. 325–333).

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

Zitzler, E., Deb, K., & Thiele, L. (2000). "Comparison of multiobjective evolutionary algorithms: Empirical results". *Evolutionary Computation*, 8, 173–195.

Zitzler, E. & Thiele, L. (1999). "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach". *IEEE Trans. on Evolutionary Computation*, 3(4), 257–271.

Índice Remissivo de Autores

Dougherty, J. 46, 98, 99
Ducoulombier, A. 67

Ebecken, N. F. F. 22
Efron, B. 76
Etxeberria, R. 22, 67, 70

Fannes, G. 2, 23
Fernández-Luna, J. 22
Friedman, J. H. 97
Friedman, N. 2, 22, 23
Fyfe, C. 77

García-Pedrajas, N. 77, 87
Geiger, D. 22, 23
Ghahramani, Z. 2
Glymour, C. 21
Goldberg, D. E. 3, 52, 56, 62, 66, 69, 70, 110
Goldberg, D.E. 67
Goodman, E. D. 97
Greensmith, J. 32
Grünwald, P. 22

Hansen, L. K. 74, 76
Harik, G. R. 68, 69
Hartigan, John A. 123
Heckerman, D. 3, 22, 23
Henrion, M. 55
Herskovits, E. 22, 24, 42, 43, 47
Hervás-Martínez, C. 77, 87
Holland, J. H. 52
Horvitz, E. 2, 23
Hruschka, E. R. 2, 23, 47
Hruschka Jr, E. R. 2, 22, 23, 47
Hunt, J. E. 32

Iba, H. 68
Imoto, S. 2, 23
Inza, I. 2, 22, 23, 95, 96, 102
Isbell, C. L. 68
Iyoda, E. M. 78

Jensen, F. V. 18
Jerne, N. K. 31
John, G. H. 96
Jung, H. K. 32

Kenley, C. R. 25
Kephart, J. O. 32
Khan, N. 62, 110
Kikuchi, R. 70
Kim, S. 2, 23
Kira, K. 96, 101
Kohavi, R. 46, 96, 98, 99
Kononenko, I. 47, 101
Koutroumbas, K. 46
Koutsias, J. 2, 23
Krogh, A. 74
Kuijpers, C. 22

Lam, W. 22
Lampinen, J. 2, 23
Langley, P. 94
Larrañaga, P. 2, 22, 23, 67, 70, 95, 102
Laskey, K. B. 22
Laumanns, M. 62
Lauritzen, L. S. 42
Lauritzen, S. 21
Linial, M. 2, 23
Liu, C. N. 68
Liu, H. 96
Liu, Y. 77, 78, 86
Lobo, F. G. 68, 69

Mahfouf, M. 32

Mahnig, T. 69
McCall, J. 68
McMorris, F.R. 44
Mühlenbein, H. 4, 67–69, 87
Mitchell, M. 52
Moller, M. F. 78, 80, 83
Moral, S. 44
Myers, J. W. 22

Nariai, N. 2, 23
Neumann, D. 44
Newman, D.J. 46, 82, 98
Nossal, G. J. V. 31

Ocenasek, J. 62
Opitz, D.W. 77

Paass, G. 4, 67
Papoulis, A. 8
Pasti, R. 77
Paul, T. K. 68
Pearl, J. 18, 22
Pelikan, M. 66–70, 87
Pennock, D. M. 44
Perrone, M.P. 75
Pillai, S. U. 8
Pohlmann, P. R. 28
Polikar, R. 74
Punch, W. F. 97

Rakotomamonjy, A. 97
Ramakrishna, R. 70
Raval, A. 2
Rendell, L. A. 96, 101
Richardson, J. J. 3
Ripley, B. D. 58
Ruan, Y. 2, 23
Russell, S. 21

Saeys, Y. 96
Salamon, P. 74, 76
Santana, R. 70
Santoro, D. M. 97
Sastry, K. 71
Saunders, G. M. 80
Schapire, R.E. 76
Schwarz, G. 22
Scroferneker, M. L. 28
Sebag, M. 67
Shachter, R. D. 25
Shakya, S. 68
Shavlik, J.W. 77
Siedlecki, W. 95
Simpson, P.K. 97
Skalak, D. 80
Sklansky, J. 95
Spiegelhalter, D. J. 42
Spirtes, P. 21
Srinivas, S. 21
Stephenson, T. 2, 23
Suermondt, H. 42
Suzuki, J. 22

Theodoridis, S. 46
Thiele, L. 62
Thierens, D. 70, 123
Tibshirani, R. 76
Timmis, J. 3, 27, 28, 30–34, 124

Vafaie, H. 97
Van Veldhuizen, D. A. 117
Vedelsby, J. 74
Vehtari, A. 2, 23

Von Zuben, F. J. 22, 32, 39, 53, 57, 62, 77, 78, 95, 96

Wang, S. 74, 77
Watkins, A. 32
Wellman, M. P. 44
Wermuth, N. 21
Wu, J. 77

Yao, X. 77, 78, 86
Yu, L. 96

Zeleny, M. 109
Zhang, X. 74, 77
Zhou, Z. H. 77
Zitzler, E. 59, 62, 116, 124, 129