

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E**  
**AUTOMAÇÃO INDUSTRIAL**

**Inteligência Computacional na Síntese de Meta-heurísticas para  
Otimização Combinatória e Multimodal**

**Autora: Lalinka de Campos Teixeira Gomes**

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

**Co-Orientador: Prof. Dr. Leandro Nunes de Castro – Unisantos/SP**

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica.

Área de Concentração: Engenharia de Computação.

**Comissão Examinadora:**

Fernando José Von Zuben – DCA/FEEC/Unicamp – Presidente

Myriam Regattieri De Biase da Silva Delgado – DAINF/CEFET/PR – Membro Externo

Ney Lemke – IBB/Unesp – Botucatu/SP – Membro Externo

Clodoaldo Aparecido de Moraes Lima – DCA/FEEC/Unicamp – Membro Interno

Márcio Luiz de Andrade Netto – DCA/FEEC/Unicamp – Membro Interno

Rafael Santos Mendes – DCA/FEEC/Unicamp – Membro Interno

Campinas – São Paulo – Brasil

Junho de 2006

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

G585i Gomes, Lalinka de Campos Teixeira  
Inteligência computacional na síntese de meta-heurísticas  
para otimização combinatória e multimodal / Lalinka de  
Campos Teixeira Gomes. --Campinas, SP: [s.n.], 2006.

Orientadores: Fernando José Von Zuben, Leandro Nunes  
de Castro

Tese (doutorado) - Universidade Estadual de Campinas,  
Faculdade de Engenharia Elétrica e de Computação.

1. Otimização combinatória. 2. Heurística. 3. Problema  
do caixeiro viajante. 4. Algoritmos genéticos. I. Von  
Zuben, Fernando José. II. Castro, Leandro Nunes de. III.  
Universidade Estadual de Campinas. Faculdade de  
Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês: Computational intelligence applied to the synthesis of metaheuristics  
for combinatorial and multimodal optimization

Palavras-chave em Inglês: Computational optimization, Multimodal optimization,  
Metaheuristic, Vehicle routing problem, Traveling salesman  
problem, Tabu search, Artificial immune systems,  
Evolutionary computation, Artificial intelligence, Copt-  
aiNet

Área de concentração: Engenharia de Computação

Titulação: Doutora em Engenharia Elétrica

Banca examinadora: Myriam Regattieri de Biasi da Silva Delgado, Ney Lemke,  
Clodoaldo Aparecido de Moraes Lima, Márcio Luiz de Andrade  
Neto e Rafael Santos Mendes

Data da defesa: 12/06/2006

Programa de Pós-Graduação: Engenharia Elétrica

# BANCA EXAMINADORA

Prof. Dr. Fernando José Von Zuben (Matr. 263958): Fernando José Von Zuben

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

Prof. Dr. Ney Lemke: Ney Lemke

Prof. Dr. Clodoaldo Aparecido de Moraes Lima: Clodoaldo Aparecido de Moraes Lima

Prof. Dr. Márcio Luiz de Andrade Netto: Márcio Luiz de Andrade Netto

Prof. Dr. Rafael Santos Mendes: Rafael Santos Mendes

Secretário(a) Soraia Cecília Montagner: Soraia Cecília Montagner

Coordenador de PG Prof. Dr. Michel Daoud Yacoub: Michel Daoud Yacoub

Este trabalho contou com o apoio financeiro do CNPq, processo 141882/2001-8.

# AGRADECIMENTOS

Primeiramente gostaria de expressar meus agradecimentos a meu orientador, Fernando José Von Zuben, pela oportunidade e pela competência com que sempre orientou este trabalho de pesquisa.

A meu co-orientador, Leandro Nunes de Castro, pelas importantes discussões e sugestões na área de engenharia imunológica.

À banca examinadora, pelas valiosas sugestões que contribuíram para o enriquecimento deste trabalho.

A meus pais, Eustáquio e Vera, e ao meu irmão Leandro, pelo apoio e constante estímulo.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo suporte financeiro.

# RESUMO

Problemas de otimização combinatória apresentam grande relevância prática e surgem em uma ampla gama de aplicações. Em geral, a otimização combinatória está associada a uma explosão de candidatos à solução, inviabilizando a aplicação de métodos exatos. Frente à intratabilidade desta classe de problemas via métodos exatos, nos últimos anos tem havido um crescente interesse por métodos heurísticos capazes de encontrar soluções de alta qualidade, não necessariamente ótimas. Considerando o notório sucesso empírico de meta-heurísticas concebidas através da inspiração biológica e na natureza, essas abordagens vêm ganhando cada vez mais atenção por parte de pesquisadores. É fato conhecido que não existe uma única metodologia capaz de sempre produzir os melhores resultados para todas as classes de problemas, ou mesmo para todas as instâncias de uma mesma classe. Assim, a busca de solução para problemas de natureza combinatória constitui uma linha de pesquisa desafiadora. Nesta tese são considerados problemas de otimização combinatória multicritério e multimodal. Como principal contribuição, destaca-se a concepção de novas meta-heurísticas para a solução de problemas combinatórios de elevada complexidade, tendo sido propostas duas classes de ferramentas computacionais. A primeira envolve um método híbrido fundamentado em mapas auto-organizáveis de Kohonen e inferência nebulosa, em que um conjunto de regras guia o processo de treinamento do mapa de modo a permitir o tratamento de problemas com restrições e múltiplos objetivos. A segunda abordagem baseia-se em sistemas imunológicos artificiais. Em particular, a abordagem imunológica levou à proposição de meta-heurísticas capazes de encontrar e manter diversas soluções de alta qualidade, viabilizando o tratamento de problemas multimodais. Como casos de estudo, foram consideradas duas classes de problemas de otimização combinatória multimodal: o problema de roteamento de veículos capacitados e o problema do caixeiro viajante simétrico. As técnicas propostas foram também adaptadas para a solução de problemas de bioinformática, em particular ao problema de análise de dados de expressão gênica, produzindo resultados diferenciados e indicando um elevado potencial para aplicações práticas.

# ABSTRACT

Combinatorial optimization problems possess a high practical relevance and emerge on a wide range of applications. Usually, combinatorial optimization is associated with an explosion of candidates to the solution, making exact methods unfeasible. Before the unfeasibility of exact methods when dealing with this class of problems, lately there has been an increasing interest in heuristic methods capable of finding high-quality solutions, not necessarily the optimal one. Considering the widely known empirical success of metaheuristics conceived with inspiration on biological systems and on the nature itself, such approaches are receiving more and more attention from the scientific community. Evidently, there is no single methodology able to always produce the best results for all classes of problems, or even for all instances of one specific class. That is why the search for solutions to combinatorial problems remains a challenging task. This thesis considers multicriteria and multimodal combinatorial optimization problems. As the main contribution, one can emphasize the conception of new metaheuristics designed to the solution of high-complexity combinatorial optimization problems, and two classes of computational tools have been proposed. The first one involves a hybrid method based on Kohonen self-organizing maps and fuzzy inference, in which a set of rules guides the training of the self-organizing maps in order to allow the handling of problems with constraints and multiple objectives. The second approach is based on artificial immune systems. Particularly, the immune-inspired approach leads to the proposal of metaheuristics capable of finding out and maintaining multiple high-quality solutions, making it possible to deal with multimodal problems. As case studies, the capacitated vehicle routing problem and the symmetric traveling salesman problem are considered, giving rise to combinatorial and multimodal problems. The proposed techniques were also adapted to the solution of problems in the field of bioinformatics, specifically the analysis of gene expression data, leading to distinguished results and indicating a high potential for practical applications.

# ÍNDICE

Capítulo 1. Introdução	1
1.1. Motivação	1
1.2. Objetivos e Contribuições	5
1.3. Organização da Tese	6
Capítulo 2. Meta-Heurísticas e Otimização Combinatória	11
2.1. Introdução	12
2.2. Problemas de Otimização Combinatória	15
2.2.1. O Problema do Caixeiro Viajante (TSP)	15
2.2.2. O Problema de Roteamento de Veículos Capacitados (CVRP – Capacitated Vehicle Routing Problem)	17
2.3. Técnicas de Solução para Problemas Combinatórios	19
2.3.1. Classes de Problemas de Otimização	19
2.3.2. Meta-Heurísticas para Problemas Combinatórios	19
2.3.3. Métodos Exatos para Espaços Discretos	21
2.3.4. Meta-Heurísticas via Abordagens Não-Populacionais	22
2.3.5. Meta-Heurísticas via Abordagens Populacionais	28
2.4. Redes Neurais Artificiais	38
2.4.1. Redes de Hopfield	39
2.4.2. Mapas Auto-Organizáveis	43
2.5. Abordagens Híbridas	44
Capítulo 3. Mapas Auto-Organizáveis de Kohonen	47
3.1. Introdução	47

3.2.	Arquitetura do Mapa Auto-Organizável de Kohonen _____	50
3.3.	Treinamento de Mapas Auto-Organizáveis _____	52
3.4.	Inicialização Sináptica _____	54
3.5.	Processo Competitivo _____	54
3.6.	Processo Cooperativo _____	56
3.7.	Processo Adaptativo _____	60
3.8.	Algoritmo de Treinamento _____	62
3.8.1.	Algoritmo de Treinamento em Modo Seqüencial _____	63
3.8.2.	Algoritmo de Treinamento em Lote ou Batelada _____	64
3.9.	Outros Modelos de Redes Neurais Auto-Organizáveis _____	65
3.9.1.	Adaptive Resonance Theory (ART) _____	65
3.9.2.	Modelos Construtivos _____	67
3.9.3.	Redes Neurais Competitivas Hierárquicas _____	68
3.9.4.	Neocognitron _____	69
3.9.5.	Mapas com Dimensão Elevada _____	70
3.10.	Algumas Áreas de Aplicação _____	71
3.10.1.	Agrupamento (Clusterização) e Classificação _____	72
3.10.2.	O Problema do Caixeiro Viajante e o Problema de Roteamento de Veículos _____	75
3.10.3.	Regressão para Identificação de Sistemas Dinâmicos _____	79
3.10.4.	Organização e Seleção de Documentos _____	81
3.10.5.	Segmentação de Imagens _____	82
Capítulo 4. Solução de Problemas Combinatórios Utilizando Redes Neurais Auto-Organizáveis _____		83
4.1.	Introdução _____	83
4.2.	Aplicação de Mapas Auto-Organizáveis ao Problema do Caixeiro Viajante Simétrico _____	85
4.2.1.	Inicialização Sináptica _____	85
4.2.2.	O Processo Competitivo _____	86
4.2.3.	O Processo Cooperativo _____	87
4.2.4.	Atualização Sináptica _____	88

4.2.5.	Convergência do Mapa	88
4.2.6.	O Algoritmo	91
4.3.	Mapas Auto-Organizáveis Aplicados ao Problema de Roteamento de Veículos	92
4.3.1.	Abordagens Existentes Empregando Mapas Auto-Organizáveis	92
4.3.2.	Uma Abordagem Neuro-Nebulosa para o Problema de Roteamento de Veículos Capacitados	95
4.3.3.	Resultados Computacionais	108
4.4.	Síntese do Capítulo	117
Capítulo 5. Sistemas Imunológicos Artificiais		121
5.1.	Introdução	121
5.2.	Conceitos Básicos	123
5.2.1.	Terminologia	123
5.2.2.	O Espaço de Formas	124
5.3.	CLONALG (Clonal Selection Algorithm): Implementação Computacional do Princípio da Seleção Clonal	125
5.3.1.	Aspectos Gerais	125
5.3.2.	Notação	126
5.3.3.	Descrição do CLONALG	127
5.3.4.	O Algoritmo	127
5.3.5.	Resultados do CLONALG	131
5.4.	ABNET (Antibody Network)	136
5.4.1.	Aspectos Gerais	136
5.4.2.	Notação	136
5.4.3.	Descrição da ABNET e Algoritmo	137
5.5.	aiNet (Artificial Immune Network)	141
5.5.1.	Aspectos Gerais	141
5.5.2.	Notação	141
5.5.3.	Descrição da aiNet	142
5.5.4.	Extração de Conhecimento da aiNet Treinada	145

5.6.	opt-aiNet (Artificial Immune Network for Optimization)	146
5.6.1.	Aspectos Gerais	146
5.6.2.	Notação e Terminologia	146
5.6.3.	Algoritmo para a opt-aiNet	148
5.6.4.	Exemplos de Comportamento do Algoritmo	149
Capítulo 6. copt-aiNet: Uma Abordagem Baseada em Sistemas Imunológicos Artificiais para Problemas Combinatórios		155
6.1.	Introdução	155
6.2.	A copt-aiNet (Artificial Immune Network for Combinatorial Optimization)	156
6.2.1.	Aspectos Gerais	156
6.2.2.	Notação e Terminologia	157
6.2.3.	Algoritmo para a copt-aiNet	158
6.2.4.	Exemplos de Comportamento do Algoritmo	163
6.3.	Síntese do Capítulo	166
Capítulo 7. Análise de Dados de Expressão Gênica		167
7.1.	Introdução	167
7.2.	Microarrays e Expressão Gênica	168
7.3.	Organizando Dados de Expressão Gênica	169
7.4.	Técnicas de Agrupamento e de Ordenação	171
7.5.	Aplicação da copt-aiNet ao Problema de Ordenação de Dados de Expressão Gênica	175
7.5.1.	Motivação	175
7.5.2.	Resultados Computacionais	176
Capítulo 8. Conclusões e Perspectivas		179
8.1.	Contribuições	179
8.2.	Perspectivas e Pesquisas Futuras	182
Apêndice A. Complexidade Computacional		185
A.1.	Dificuldade de Problemas: Problemas P, NP, NP-Completo e Reduções	185
Apêndice B. O Sistema Imunológico		189

B.1.	Conceitos Básicos_____	189
B.2.	Estrutura do Sistema Imunológico, Elementos Constituintes e Mecanismos de Defesa	190
B.2.1.	Estrutura do Sistema Imunológico_____	190
B.2.2.	Principais Células do Sistema Imunológico _____	192
B.2.3.	Os Órgãos do Sistema Imunológico _____	196
B.2.4.	Mecanismos Básicos de Defesa_____	198
B.3.	Reconhecimento de Padrões_____	202
B.4.	A Teoria da Seleção Clonal _____	203
B.5.	Diversidade do Receptor _____	205
B.6.	Teoria da Rede Imunológica _____	207
B.7.	Maturação de Afinidade _____	208
Apêndice C. Expressão Gênica _____		209
C.1.	Introdução_____	209
C.2.	Conceitos Básicos de Biologia Molecular _____	210
C.2.1.	O Genoma e os Genes _____	210
C.2.2.	DNA_____	211
C.2.3.	RNA_____	212
C.2.4.	Síntese Protéica _____	213
C.3.	Análise de Expressão Gênica _____	214
C.3.1.	Hibridização Comparativa _____	214
C.3.2.	Repositórios de Dados de Expressão Gênica_____	219
C.3.3.	Softwares para Visualização e Análise de Dados de Expressão Gênica _____	221
Apêndice D. Sistemas Nebulosos_____		223
D.1.	Introdução_____	223
D.2.	Conjuntos Nebulosos x Conjuntos Convencionais_____	224
D.2.1.	Funções de Pertinência _____	226
D.3.	Sistemas baseados em Inferência Nebulosa _____	226

D.3.1.	O Conceito de Variáveis Lingüísticas _____	227
D.3.2.	Computação com Regras Nebulosas _____	227
D.3.3.	Funções de Pertinência _____	229
D.3.4.	Métodos de <i>Defuzzificação</i> _____	231
Apêndice E. Publicações _____		233
Referências Bibliográficas _____		235

# CAPÍTULO 1

## INTRODUÇÃO

---

---

### 1.1 Motivação

A *pesquisa operacional* teve sua origem como uma disciplina formalizada na Inglaterra por volta de 1940, com o objetivo de otimizar o uso de recursos militares e realizar planejamentos logísticos durante a II Guerra Mundial (KIRBY, 2003). Desde então, com a intensificação das pesquisas e a crescente disponibilidade de recursos computacionais, a área da pesquisa operacional tem evoluído constantemente e atualmente nos oferece uma ampla gama de métodos destinados à solução de variadas classes de problemas de otimização. A opção por um método ou outro depende de diversos fatores, que abrangem desde o tipo de problema a ser tratado, se este envolve funções-objetivo e restrições lineares ou não-lineares, se as variáveis de decisão podem assumir valores discretos ou contínuos, até a dimensão do espaço de busca.

Problemas de otimização são compostos por um conjunto de variáveis de decisão, por um conjunto de restrições a serem respeitadas, as quais dividem o espaço de busca em regiões factíveis e infactíveis, e por uma função-objetivo a ser otimizada (minimizada ou maximizada), de modo que cada ponto do espaço de busca admita um valor numérico associado ao seu desempenho como solução candidata do problema. Dependendo das características do problema, a sua formulação leva a um processo de solução que pode ser expresso algebricamente, de forma fechada. Assim, a solução ótima pode ser expressa diretamente e de forma explícita:

$$\mathbf{x}^* = f(\mathbf{p}),$$

onde  $f$  é função de um vetor de parâmetros  $p$ . Dessa forma, observa-se que nenhum processo iterativo é necessário para se obter a solução do problema.

Entretanto, a obtenção de uma solução fechada raramente é possível, pois grande parte dos problemas de otimização apresenta características (por exemplo, não-linearidades e manipulação de variáveis que assumem apenas valores discretos) que inviabilizam a obtenção de uma solução analítica direta, requerendo a aplicação de processos iterativos em um espaço de busca  $X$  de dimensão elevada e/ou sujeito a uma explosão combinatória de candidatos à solução. Nesse ponto, outra questão a ser considerada é o comportamento de convergência do processo iterativo. Três condições são desejáveis: garantia de convergência, tempo de convergência compatível com as necessidades de cada aplicação e convergência para a solução ótima. Basicamente, quanto mais complexo o problema de otimização e quanto maior seu tamanho (número de variáveis envolvidas, que corresponde à dimensão do espaço  $X$ ), menor a chance de se terem atendidas estas três condições simultaneamente, principalmente quando o método de otimização empregado não é suficientemente poderoso na execução do processo de busca da solução.

Neste trabalho serão abordados problemas combinatórios, em particular problemas de caminho mínimo mediante um conjunto de restrições, tais como o problema do caixeiro viajante (REINELT, 1991) e problemas de roteamento de veículos (GOLDEN & ASSAD, 1988). Embora problemas de caminho mínimo possam ser resolvidos na otimalidade através de métodos exatos derivados da programação inteira, tais como *branch-and-bound* e *branch-and-cut* (LAWLER & WOOD, 1966), estes são proibitivos quando aplicados a problemas de grande porte, acarretando sérias dificuldades computacionais. Dessa forma, observa-se que existem problemas para os quais as abordagens clássicas da pesquisa operacional não são tratáveis computacionalmente, requerendo o emprego de métodos denominados *heurísticos*. Os métodos heurísticos produzem soluções de boa qualidade, não necessariamente ótimas, para problemas computacionalmente intratáveis através de métodos exatos, cuja formulação matemática não é completamente conhecida ou cujos requisitos são difíceis de serem atendidos via abordagens de otimização convencionais. Existe uma série de dificuldades relacionadas a processos de otimização global, as quais são geradas sobretudo em virtude do

espaço de solução ser extremamente amplo e da ausência de condições que permitiriam definir se um ponto de ótimo local é global ou o quanto se aproxima dele. Dessa forma, mostra-se necessária a criação de técnicas que permitam uma exploração eficiente do espaço de busca, considerando limitações relacionadas a esforço computacional. Tais técnicas são denominadas *meta-heurísticas*. O termo *heurística* deriva da palavra grega *heuristiken*, que significa encontrar ou descobrir. Na área da inteligência computacional, os métodos chamados heurísticos buscam por soluções subótimas de boa qualidade, provendo um compromisso entre a qualidade das soluções obtidas e o custo computacional para exploração do espaço de busca. Os métodos heurísticos podem não apresentar garantia de factibilidade e/ou otimalidade das soluções encontradas.

Uma meta-heurística pode ser definida como um ou mais métodos heurísticos combinados de forma a operar em sinergia na busca por soluções de alta qualidade. As meta-heurísticas são geralmente aplicadas a problemas para os quais não existe um algoritmo exato ou quando estes requerem grandes quantidades de recursos computacionais, tornando inviável sua aplicação prática.

Com o crescente aumento da capacidade de processamento dos computadores, novas propostas para métodos heurísticos cada vez mais sofisticados voltadas para problemas de complexidade elevada vêm sendo apresentadas.

Neste trabalho são propostas algumas meta-heurísticas voltadas para o tratamento de problemas com elevado grau de complexidade computacional, espaço de busca discreto e de dimensão elevada, e possivelmente apresentando múltiplos objetivos devidamente ponderados e múltiplas restrições. Estas propriedades levam a uma explosão combinatória de candidatos à solução, de modo que uma busca exaustiva pela solução ótima, dentre as soluções candidatas, representa um procedimento computacionalmente intratável (GAREY & JOHNSON, 1979). Em termos de complexidade computacional, uma significativa parcela dos problemas combinatórios é classificada como problemas do tipo *NP-difíceis* (definidos no Apêndice A desta tese), o que impossibilita o uso de algoritmos exatos. Diversas abordagens eficientes para tratar problemas combinatórios NP-difíceis encontram-se disponíveis para prover

soluções aproximadas de alta qualidade com custos computacionais razoáveis. Nesse sentido, a Inteligência Computacional vem exercendo um importante papel na área de otimização combinatória, explorando o potencial de se criarem técnicas através de mecanismos característicos de organismos inteligentes, tais como capacidade de evolução, aprendizado e memória. Encontra-se na literatura uma variedade de métodos provenientes de diferentes metodologias da Inteligência Computacional, como abordagens populacionais, métodos baseados em memória, inferência nebulosa e redes neurais artificiais (YAGER & ZADEH, 1994). As abordagens populacionais, cujos principais representantes são os algoritmos genéticos (HOLLAND, 1992; BÄCK *et al.*, 1997) têm se mostrado eficientes na solução de problemas de otimização de caráter combinatório. Os métodos baseados em memória, tais como a busca tabu (GLOVER & LAGUNA, 1997; GLOVER & LAGUNA, 1998; TAILLARD *et al.*, 1997), mantêm um histórico da busca com o objetivo de explorar de forma eficiente o espaço de busca com base na experiência passada.

Técnicas de inferência nebulosa, introduzidas por ZADEH (1965; 1996), têm sido extensivamente aplicadas a diversas áreas para lidar com informações imprecisas. São comumente usadas em conjunto com outras técnicas. Neste cenário, sistemas nebulosos têm sido úteis em diversos domínios de aplicação, como robótica (FUKUDA, 1999), tomada de decisão, classificação, reconhecimento (KUNG *et al.*, 1999) e mineração de dados (HIROTA & PEDRYCZ, 1999).

Experimentos realizados mostram que associações de algoritmos populacionais com busca local não somente conduzem a soluções de melhor qualidade como também tendem a reduzir o número de iterações (HART, 1994).

O sucesso das meta-heurísticas baseadas na hibridização de variadas técnicas é consequência da sinergia de variados métodos de busca, podendo envolver diversos paradigmas da área de Inteligência Computacional.

## 1.2 Objetivos e Contribuições

Esta tese apresenta uma visão abrangente do estado da arte das meta-heurísticas destinadas à solução de problemas combinatórios, oferecendo uma perspectiva panorâmica de alguns dentre os principais métodos e apontando vantagens e limitações de cada um.

A principal contribuição desta pesquisa reside na proposição de novas meta-heurísticas voltadas para a solução de problemas de caráter combinatório. Como objeto de estudo das abordagens propostas, elegemos o problema de roteamento de veículos capacitados (TOTH & VIGO, 2002), freqüentemente referenciado como CVRP (acrônimo provindo da terminologia em inglês *Capacitated Vehicle Routing Problem*), tanto por sua complexidade elevada como por sua relevância prática. Além do CVRP, será abordado o *problema do caixeiro viajante simétrico*, que pode ser visto como uma instância simples do CVRP e constitui um importante campo de provas para validação de novas metodologias propostas pela comunidade científica.

A primeira abordagem proposta nesta tese baseia-se em redes neurais artificiais, em particular nas redes auto-organizáveis de Kohonen (Kohonen, 1997). A idéia de utilizar redes neurais artificiais na solução de problemas combinatórios tem suas origens no trabalho de HOPFIELD & TANK (1982;1985), quando os autores resolveram o problema do caixeiro viajante simétrico através de uma rede de Hopfield, a qual implementa um sistema dinâmico não-linear que converge para pontos de equilíbrio em estado estacionário. Entretanto, devido a características inerentes ao método, tais como a existência de atratores espúrios (pontos de equilíbrio indesejados), as soluções produzidas muitas vezes eram ineficazes ou longe da otimalidade. Além disso, a rede de Hopfield aplicada a problemas combinatórios em geral requer ajustes não triviais de diversos parâmetros. Uma outra abordagem baseada em redes neurais artificiais faz uso de redes neurais auto-organizáveis, as quais têm sido aplicadas com sucesso a uma variedade de problemas de caminho mínimo.

Um dos principais fatores que motivaram a proposição de uma meta-heurística baseada em mapas auto-organizáveis nesta tese é que redes neurais vêm sendo relativamente pouco aplicadas a problemas combinatórios, em particular para o problema de roteamento de veículos, e, em geral, as abordagens apresentadas na literatura são consideravelmente simples.

Assim, faz-se necessário um estudo mais detalhado do comportamento de mapas auto-organizáveis quando aplicados a problemas combinatórios.

A outra meta-heurística proposta baseia-se em sistemas imunológicos artificiais. Dada a inerente habilidade dos sistemas imunológicos artificiais em encontrar e manter diversas soluções quando aplicadas à otimização de funções multimodais definidas em um espaço contínuo, a técnica proposta em DE CASTRO & TIMMIS (2002) foi estendida para problemas combinatórios.

As técnicas propostas foram também utilizadas na solução de problemas de bioinformática, produzindo resultados diferenciados e provando serem de grande utilidade para aplicações práticas. Em particular, foi tratado o problema de ordenação de dados de expressão gênica, que objetiva estabelecer relações entre genes. Nesta tese foi proposta uma abordagem baseada em ordenação linear, de forma similar ao problema do caixeiro viajante. Foi constatado que a ordenação linear é uma abordagem eficiente, sendo capaz de evidenciar novas relações entre genes, que não puderam ser observadas através de procedimentos de clusterização. As relações entre genes tendem a ser bastante complexas, de modo que uma única configuração em geral não é capaz de prover um entendimento completo acerca de todas as associações relevantes presentes nos dados. Assim, com o intuito de evidenciar múltiplas relações entre genes, foi desenvolvida uma ferramenta computacional imuno-inspirada.

### **1.3 Organização da Tese**

Esta tese encontra-se organizada em 8 capítulos e 5 apêndices, conforme especificado a seguir:

- Capítulo 1: Introdução.
- Capítulo 2: Meta-heurísticas e Problemas Combinatórios.
- Capítulo 3: Mapas Auto-Organizáveis de Kohonen.
- Capítulo 4: Mapas Auto-Organizáveis de Kohonen aplicados a problemas combinatórios.

- Capítulo 5: Sistemas Imunológicos Artificiais.
- Capítulo 6: copt-aiNet: Solução de Problemas Combinatórios através de Sistemas Imunológicos Artificiais.
- Capítulo 7: Aplicação ao Problema de Ordenação de Dados de Expressão Gênica.
- Capítulo 8: Conclusões e Perspectivas.
- Apêndice A: Complexidade Computacional e Teoria da NP-completude.
- Apêndice B: O Sistema Imunológico Biológico.
- Apêndice C: Introdução a Microarrays e ao Problema de Ordenação Gênica.
- Apêndice D: Introdução à Lógica Nebulosa.
- Apêndice E: Publicações.

O Capítulo 2 oferece uma descrição detalhada acerca dos problemas combinatórios tratados neste trabalho: o problema do caixeiro viajante e o problema de roteamento de veículos, além de outros problemas que admitem uma formulação equivalente. Ainda neste capítulo, são apresentadas as principais meta-heurísticas utilizadas na solução de problemas combinatórios. Serão enfocadas diversas técnicas de otimização: busca tabu (GLOVER & LAGUNA, 1997; 1998), algoritmos genéticos (BÄCK *et al.*, 1997), algoritmos meméticos (MOSCATO, 2003), colônia de formigas (DORIGO, 1999a;1999b), sistemas imunológicos artificiais (DE CASTRO & TIMMIS, 2002), otimização por enxame de partículas (KENNEDY & EBERHART, 1995; KENNEDY & EBERHART, 2001; EBERHART & KENNEDY, 1995), além dos métodos clássicos branch-and-bound e branch-and-cut (LAWLER & WOOD, 1966).

O Capítulo 3 aborda os conceitos básicos relacionados aos mapas auto-organizáveis de Kohonen, uma das metodologias apresentadas nesta tese para solução de problemas combinatórios via auto-organização de estruturas conexionistas na forma de redes neurais artificiais.

O Capítulo 4 trata da aplicação de mapas auto-organizáveis a problemas combinatórios. Será também apresentada uma abordagem híbrida envolvendo mapas auto-organizáveis, busca tabu e inferência nebulosa.

O Capítulo 5 apresenta as principais técnicas de otimização baseadas em sistemas imunológicos. Os Sistemas imunológicos constituem um dos sistemas biológicos mais importantes e de elevada complexidade, comparável à do cérebro. Apresentam uma grande variedade de características interessantes sob o ponto de vista computacional, tais como manutenção da diversidade da população, controle automático do número de indivíduos, memória, capacidade de efetuar reconhecimento de padrões, entre muitas outras.

O Capítulo 6 apresenta uma nova abordagem, denominada *copt-aiNet*, destinada a problemas combinatórios e baseada em sistemas imunológicos. Recentemente, os sistemas imunológicos artificiais têm se posicionado como uma meta-heurística poderosa para a solução de problemas combinatórios, proporcionando maior diversidade que outras abordagens populacionais, tais como algoritmos genéticos, e provendo controle automático do número de indivíduos da população. Serão apresentados resultados envolvendo o problema do caixeiro viajante simétrico.

O Capítulo 7 apresenta resultados obtidos a partir da aplicação de uma das meta-heurísticas propostas nesta tese a uma importante classe de problemas de bioinformática, o problema de ordenação de dados de expressão gênica gerados de experimentos de microarrays.

O Capítulo 8 apresenta as conclusões e principais perspectivas, apontando as principais vantagens das meta-heurísticas propostas nesta tese para contextos específicos, além das possíveis extensões das técnicas propostas a outros problemas, tais como otimização multiobjetivo (KALYANMOY, 2002) e problemas dinâmicos (DE FRANÇA *et al.*, 2005).

No Apêndice A é apresentada a teoria da complexidade computacional e aproximabilidade de problemas com enfoque na análise de problemas combinatórios. Este tópico é de fundamental importância para a compreensão da dificuldade inerente aos problemas combinatórios, justificando assim o desenvolvimento de meta-heurísticas eficientes.

O Apêndice B oferece os conceitos básicos relacionados ao sistema imunológico biológico, enfocando as principais características que o posicionam como uma fonte de inspiração na área da inteligência computacional.

O Apêndice C fornece uma introdução a um dos problemas mais abordados na área da Bioinformática, o problema de ordenação de dados de expressão gênica provenientes de experimentos de microarrays. O problema de ordenação de dados de expressão gênica foi tratado neste trabalho como um problema combinatório e solucionado através das diversas técnicas de otimização aqui propostas.

O Apêndice D introduz os principais conceitos relacionados à lógica nebulosa, empregados na meta-heurística proposta no Capítulo 4.

O Apêndice E contém uma lista com as publicações realizadas pela autora desta tese durante a execução do projeto de pesquisa.

# CAPÍTULO 2

## META-HEURÍSTICAS E OTIMIZAÇÃO COMBINATÓRIA

---

---

Neste capítulo serão introduzidos alguns problemas de otimização combinatória de grande relevância teórica e prática. Em particular, serão apresentados dois problemas de roteamento: o problema do caixeiro viajante (TSP, do inglês *Traveling Salesman Problem*) e o problema de roteamento de veículos capacitados (CVRP, do inglês *Capacitated Vehicle Routing Problem*). Foram escolhidos o TSP e o CVRP por duas razões principais:

- tais problemas são considerados clássicos no campo da pesquisa operacional, sendo uma referência para validação de novas heurísticas;
- existem diversos problemas de ordem prática que podem ser mapeados para problemas de roteamento.

Em seguida, serão apresentadas algumas das principais metodologias empregadas na solução de problemas combinatórios: métodos exatos (LAPORTE & NOVERT, 1987; LAPORTE & DEMET, 2002), busca tabu (GLOVER & LAGUNA, 1998), recozimento simulado (Rees & Ball, 1987), algoritmos genéticos (BÄCK *et al.*, 1997), algoritmos meméticos (MOSCATO & COTTA, 2003), sistemas imunológicos artificiais (DE CASTRO & TIMMIS, 2002) e redes neurais artificiais (HAYKIN, 1999). As redes neurais artificiais e os sistemas imunológicos artificiais serão tratados em mais detalhes nos capítulos 4 e 6, respectivamente.

## 2.1 Introdução

Em geral, problemas combinatórios requerem a consideração de um grande número de cenários. Para a maioria das aplicações, a otimalidade constitui uma meta impossível em face das limitações computacionais, tornando a aplicação de métodos exatos impraticável. Essa dificuldade ocorre mesmo quando consideramos problemas relativamente simples como o de permutações. Suponha uma seqüência de  $n$  elementos. Os  $n$  elementos podem estar arranjados de diversas maneiras, sendo que temos  $n!$  opções distintas. Assim, para  $n=10$  elementos, têm-se  $10! = 3.628.800$  possíveis permutações; para  $n = 20$ , têm-se  $20! = 2,43 \times 10^{18}$  possíveis permutações; para  $n = 100$  elementos, têm-se  $100! = 9,3 \times 10^{157}$  possíveis permutações, valor muito superior ao número estimado de átomos no universo ( $6 \times 10^{78}$ ). A Figura 2.1 ilustra, em escala logarítmica, a variação do número de permutações em função do número de elementos. Observe que o número de soluções possíveis cresce de forma fatorial com relação à dimensão do problema, levando a uma explosão combinatória. Em vista disso, nos últimos 30 anos, com o desenvolvimento de processadores mais poderosos, os métodos heurísticos vêm ganhando relevância e têm sido uma abordagem comumente adotada na busca de uma solução aproximada para problemas combinatórios (BLUM & ROLI, 2003). Uma das características dessa classe de problemas é a elevada dimensão do espaço de busca, levando a uma explosão de candidatos à solução. Dada a inviabilidade de tratamento desses problemas em sua otimalidade via métodos exatos, uma linha de pesquisa que tem se desenvolvido rapidamente refere-se ao estudo de meta-heurísticas. Uma meta-heurística é um processo iterativo que guia heurísticas subordinadas de modo a explorar e explorar o espaço de busca. A literatura é repleta de referências a algoritmos genéticos, algoritmos meméticos, colônia de formigas, busca tabu, recozimento simulado (do inglês *simulated annealing*), enxame de partículas, busca evolutiva, redes neurais artificiais, dentre muitas outras propostas de solução. Referências específicas à literatura serão fornecidas ao longo das próximas seções deste capítulo. Todas essas heurísticas objetivam encontrar soluções aproximadas de alta qualidade para problemas NP-difíceis, empregando para tanto recursos computacionais consideráveis.

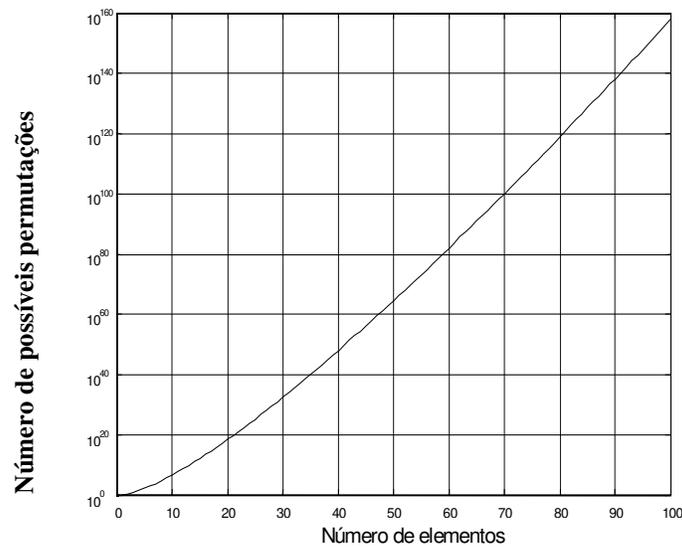
Atualmente, existem conferências, livros e periódicos inteiramente dedicados a meta-heurísticas (GLOVER & KOCHENBERGER, 2002). Problemas cada vez mais complexos vêm sendo tratados, tornando o desenvolvimento de heurísticas mais rápidas e eficientes uma necessidade. A Figura 2.2 ilustra o aumento do porte de problemas do caixeiro viajante que vêm sendo tratados ao longo dos anos. Conforme pode ser observado, a dimensão dos problemas tem crescido significativamente. Esse fato vem sendo impulsionado tanto pelo aumento da capacidade computacional, viabilizando assim o desenvolvimento de novas técnicas de otimização poderosas e o tratamento de grandes quantidades de dados, como pela demanda proveniente de aplicações práticas na indústria.

As heurísticas não têm por objetivo encontrar o ótimo global para um problema, mas sim encontrar soluções de alta qualidade (que eventualmente podem corresponder ao ótimo global) considerando as limitações computacionais envolvidas. As heurísticas modernas, comumente chamadas de meta-heurísticas, incluem diversas abordagens, dentre as quais destacam-se: algoritmos genéticos, algoritmos meméticos, colônia de formigas, enxame de partículas, redes neurais artificiais, busca tabu e recozimento simulado.

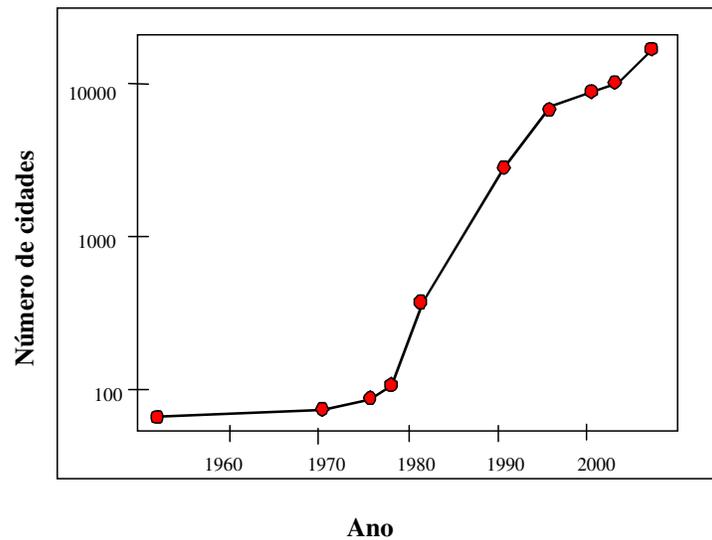
Os algoritmos genéticos (FOGEL, 1995; HOLLAND, 1975) têm se mostrado eficientes na solução de problemas de otimização de caráter combinatório e serão abordados na Seção 2.3.5.1.

Atualmente as abordagens baseadas em redes neurais artificiais exercem um papel ativo na resolução de problemas combinatórios (POTVIN, 1995). Um dos segmentos de redes neurais que vem adquirindo crescente importância e considerável sucesso é procedente do paradigma de aprendizado não-supervisionado, e constitui a classe dos mapas auto-organizáveis (redes competitivas). Estudos efetuados nos últimos anos demonstram que os mapas auto-organizáveis são extremamente eficientes na solução do TSP em larga escala (SOYLU & ÖZDEMIREL, 2000). Em MODARES *et al.* (1999) são apresentados resultados provenientes de uma análise comparativa que envolve diversas metodologias destinadas à solução do MTSP (do inglês *Multiple Traveling Salesman Problem*) e, segundo os autores, as técnicas baseadas

em redes neurais competitivas apresentaram desempenho promissor, tanto em termos de qualidade da solução final como em tempo de execução.



**Figura 2.1.** Número de elementos versus possíveis permutações (escala logarítmica no eixo y).



**Figura 2.2.** Variação da escala das instâncias do TSP no tempo (escala logarítmica no eixo y).

## 2.2 Problemas de Otimização Combinatória

Problemas de otimização combinatória estão relacionados à alocação eficiente de recursos em situações em que as variáveis do problema (associadas a uma função-objetivo e a um conjunto de restrições) pertencem a um espaço discreto.

Nas próximas seções serão apresentadas duas categorias de problemas combinatórios: o caixeiro viajante e o problema de roteamento de veículos. Tais problemas servirão como referência para validação das meta-heurísticas propostas nesta tese. São problemas de elevada complexidade, já provados serem NP-completos (ver Apêndice A), e para os quais uma ampla gama de problemas combinatórios pode ser diretamente mapeada. Conforme colocado nas seções 2.2.1 e 2.2.2, os problemas do caixeiro viajante e de roteamento de veículos são comumente utilizados como campos de prova para validação de métodos heurísticos, estando disponíveis na literatura diversas instâncias de teste.

### 2.2.1 O Problema do Caixeiro Viajante (TSP)

O Problema do Caixeiro Viajante (comumente abreviado para TSP, do inglês *Traveling Salesman Problem*) é um dos problemas mais intensamente estudados na área da matemática computacional. O TSP é definido da seguinte forma: dado um conjunto de cidades e os custos de se viajar de uma cidade a outra, quer-se a seqüência de cidades a serem visitadas de modo que o caixeiro viajante, partindo de uma cidade inicial, passe por todas as cidades uma única vez e retorne ao ponto de partida com custo total mínimo. Matematicamente, o TSP pode ser descrito da seguinte maneira: considere o grafo totalmente conectado  $G=\{V,A\}$ , sendo  $V=\{1,2,\dots,N\}$  o conjunto de vértices, onde cada vértice corresponde a uma cidade, e  $A$  o conjunto de arestas. O valor  $d_{i,j}$  associado à aresta  $(i,j)$  do conjunto  $A$  representa o custo de se trafegar entre as cidades  $i$  e  $j$ . Para o caso do problema simétrico, o custo  $d_{i,j}$  de se trafegar entre as cidades  $i$  e  $j$  é idêntico ao custo  $d_{j,i}$  de se trafegar entre as cidades  $j$  e  $i$ . No caso do problema assimétrico,  $d_{i,j} \neq d_{j,i}$ .

O objetivo do TSP é encontrar o menor caminho fechado que visita cada uma das cidades exatamente uma vez. Dessa forma, deve-se encontrar a permutação  $\pi$  de índices das cidades de modo que  $f(\pi)$  seja mínimo, onde  $f(\pi)$  é definido de acordo com a equação 2.1.

$$f(\pi) = \sum_{k=1}^{n-1} d_{\pi(k)\pi(k+1)} + d_{\pi(n)\pi(1)}. \quad (2.1)$$

Atualmente, ainda não há uma metodologia de solução genérica que seja efetiva para todas as instâncias. O problema é de grande importância prática, podendo ser aplicado a uma variedade de contextos, como na área logística, manufatura de circuitos impressos, problemas de alocação, problemas de bioinformática e muitos outros.

### 2.2.1.1 Melhores Soluções Conhecidas

Por ser um problema NP-difícil clássico e devido ao fato de uma grande variedade de problemas práticos serem mapeados como um problema de caixeiro viajante, o TSP é comumente empregado para validar novas heurísticas. Dessa forma, encontram-se disponíveis diversas bases de dados contendo instâncias do TSP e as melhores soluções conhecidas. A seguir estão relacionadas algumas referências on-line para instâncias do TSP:

- <http://www.tspintl.com/products/tsp/benchmarks/>
- <http://branchandcut.org/>
- <http://www.tsp.gatech.edu/>
- <http://www.zaik.uni-koeln.de/AFS/Projects/Optimization/VehicleRouting/Links/dataset-links.html>
- [http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB\\_home.html](http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html)

### **2.2.2 O Problema de Roteamento de Veículos Capacitados (CVRP – Capacitated Vehicle Routing Problem)**

O problema de roteamento de veículos constitui um problema de otimização discreta de grande relevância industrial e alta complexidade computacional. O problema tem recebido grande atenção desde sua introdução, por DANTZIG & RAMSER (1959). O problema de roteamento de veículos capacitados é a variação mais simples e também a mais estudada do problema de roteamento de veículos, e pode ser descrito da seguinte forma: existem  $N$  consumidores, geograficamente distribuídos em uma região bidimensional. Cada consumidor requer uma quantidade  $D_i$  de produtos ( $i = 1, \dots, N$ ).  $M$  veículos, cada um com capacidade fixa  $C$ , encontram-se disponíveis para efetuar a entrega de produtos armazenados em um depósito. Cada consumidor deve ser visitado somente uma vez e por somente um veículo. O objetivo do problema é definir uma rota para cada um dos  $M$  veículos de forma que a soma das distâncias percorridas pelos  $M$  veículos seja minimizada, sem que as capacidades individuais dos veículos sejam violadas e tal que todas as demandas sejam devidamente atendidas. Na prática, o problema de roteamento de veículos capacitados é uma etapa necessária na implementação de diversos sistemas de distribuição. Além disso, muitos problemas combinatórios podem ser modelados como esse tipo de problema de roteamento.

O planejamento das rotas é, em geral, uma tarefa difícil devido a fatores como: restrições relacionadas à capacidade dos veículos, dependência intrínseca da distribuição geográfica dos consumidores e suas respectivas demandas, e necessidade de alocação de cada consumidor a um veículo. Considerando que o CVRP é um problema NP-difícil, somente instâncias de pequeno porte são passíveis de serem resolvidas na otimalidade. As maiores instâncias que podem ser consideradas para aplicação dos algoritmos exatos desenvolvidos até o momento apresentam cerca de 50 consumidores. Dessa forma, muitos pesquisadores têm voltado sua atenção para métodos heurísticos.

### *2.2.2.1 Variações Importantes do CVRP*

#### ***CVRP com Janelas de Tempo***

O CVRP com janelas de tempo é uma extensão do CVRP em que cada consumidor encontra-se associado a um intervalo de tempo dentro do qual, obrigatoriamente, o consumidor deve ser atendido por um veículo. Existem ainda variações em cima deste modelo. Por exemplo, CHIANG & RUSSEL (2004) propuseram uma heurística para o problema de roteamento de veículos em que há certa flexibilidade nas janelas de tempo.

#### ***CVRP com Categorias de Consumidores***

Neste caso, os consumidores são divididos em duas ou mais categorias. Por exemplo, podem-se atribuir níveis de prioridade de atendimento a consumidores. As rotas devem ser estabelecidas de modo que os consumidores prioritários sejam atendidos anteriormente a qualquer consumidor comum. Outro exemplo de formulação envolve a associação de grupos de consumidores a veículos, ou seja, certos consumidores somente podem ser atendidos por veículos específicos.

#### ***CVRP com Carga e Entrega***

Neste modelo, os consumidores possuem dois valores associados: uma demanda por produtos a serem entregues e uma quantidade de produtos a serem retirados. Os produtos a serem retirados por um veículo devem respeitar as restrições de capacidade do mesmo.

#### ***CVRP com Divisão de Demanda***

Este modelo permite que a demanda de um cliente seja fracionada em dois ou mais conjuntos, sendo permitido que cada conjunto seja suprido por veículos distintos. Com isso, cada consumidor passa a admitir a visita de mais de um veículo.

### *2.2.2.2 Melhores Soluções Conhecidas*

Assim como para o problema do caixeiro viajante, existem bases de dados contendo diversas instâncias e as melhores soluções conhecidas para problemas de roteamento de veículos. A seguir encontram-se relacionadas algumas bases on-line:

- <http://branchandcut.org/>
- <http://www.zaik.uni-koeln.de/AFS/Projects/Optimization/VehicleRouting/Links/dataset-links.html>

Em VIGO & TOTH (2002) podem ser encontradas comparações entre as soluções produzidas por diversas heurísticas.

## 2.3 Técnicas de Solução para Problemas Combinatórios

### 2.3.1 Classes de Problemas de Otimização

Existem diversas classificações possíveis para um problema de otimização, dependendo das características do espaço de busca (contínuo ou discreto) e dos valores que as variáveis do problema podem assumir (reais, inteiros ou binários). Dependendo de sua classificação, um problema de otimização pode ser resolvido algebricamente, na forma fechada, ou iterativamente, empregando métodos exatos ou aproximados. De forma genérica, em um problema de otimização temos uma função-objetivo e um conjunto de restrições, ambos relacionados às variáveis de decisão. O problema pode ser de minimização ou de maximização de uma função-objetivo.

Soluções na forma fechada são expressas por equações algébricas com coeficientes derivados das propriedades específicas do problema. Entretanto, a obtenção de uma solução fechada raramente é possível, pois a maioria dos problemas de otimização apresenta características (por exemplo, não-linearidades na função-objetivo e manipulação de variáveis que assumem apenas valores inteiros) que inviabilizam a obtenção de uma solução analítica, requerendo a aplicação de processos iterativos de busca em um espaço  $X$  de dimensão elevada e/ou sujeito a uma explosão combinatória de candidatos à solução.

### 2.3.2 Meta-Heurísticas para Problemas Combinatórios

Nos últimos anos, uma grande variedade de meta-heurísticas tem sido proposta para a solução de problemas combinatórios. Recorre-se a meta-heurísticas quando não se conhecem métodos exatos de solução ou quando a aplicação dos métodos exatos conhecidos torna-se inviável

devido ao tamanho do problema. Nas seções que se seguem será oferecida uma breve revisão das principais abordagens atualmente empregadas na solução de problemas combinatórios. De um modo geral, as meta-heurísticas podem ser classificadas da seguinte forma:

- **Abordagens bioinspiradas  $\times$  abordagens não-bioinspiradas:** as abordagens bioinspiradas englobam os algoritmos inspirados na biologia, tais como algoritmos genéticos, algoritmos meméticos, colônia de formigas, enxame de partículas, redes neurais artificiais e sistemas imunológicos artificiais. As abordagens não-bioinspiradas incluem a busca tabu e recozimento simulado, entre outras buscas iterativas.
- **Abordagens híbridas:** existem diversas propostas na literatura envolvendo algoritmos híbridos, como, por exemplo, sistemas imunológicos artificiais em conjunto com busca tabu (Gomes *et al.*, 2003).
- **Abordagens populacionais  $\times$  não-populacionais:** outra característica comumente empregada para classificar uma meta-heurística é o número de soluções tratadas em uma iteração. As abordagens populacionais, como colônia de formigas, algoritmos genéticos, algoritmos meméticos, enxame de partículas e sistemas imunológicos artificiais operam com diversas soluções simultaneamente. Os métodos baseados em uma única solução, como recozimento simulado e busca tabu, procuram melhorar a solução atual. As abordagens populacionais podem ser combinadas com as não-populacionais. Um exemplo é a aplicação de um procedimento de busca local (como a busca tabu) aos indivíduos de uma população de um algoritmo genético.
- **Métodos com memória  $\times$  métodos sem memória:** algumas meta-heurísticas, como a busca tabu, fazem uso de uma ou mais memórias que armazenam informações referentes ao histórico da busca. O objetivo é permitir que o algoritmo aprenda com a experiência passada, de modo a evitar ótimos locais prematuros, identificar movimentos que levam a melhoras mais significativas na função-objetivo, evitar que soluções já analisadas em etapas anteriores da busca sejam revisitadas e que movimentos realizados recentemente sejam desfeitos.

### 2.3.3 Métodos Exatos para Espaços Discretos

O método exato mais intuitivo para a solução de problemas combinatórios é certamente a busca exaustiva, que consiste na avaliação de todas as possibilidades de solução para um determinado problema e na seleção daquela que conduz a melhores valores para a função-objetivo a ser otimizada. Conforme explicitado na Seção 2.1, a abordagem exaustiva é computacionalmente intratável para a grande maioria dos problemas combinatórios.

Algoritmos tipo branch-and-bound (LAWLER & WOOD *et al.*, 1966) e branch-and-cut (CAPRARA & FISCHETTE, 1997; JÜNGER & THIENEL, 1998) pertencem à classe dos métodos exatos. Os métodos exatos apresentam garantia de convergência e fornecem a solução ótima global para um determinado problema. Entretanto, dependendo do tamanho do problema a ser resolvido, os requisitos computacionais desses métodos tendem a ser proibitivos. Vale destacar que métodos exatos, como por exemplo branch-and-bound, exigem propriedades específicas, como monotocidade da solução em relação à profundidade na árvore de busca.

Conforme mencionado na seção 2.2, problemas de caráter combinatório levam a uma explosão de candidatos à solução, inviabilizando a exploração completa do espaço de busca pela enumeração de todas as possíveis soluções. O método branch-and-bound constitui uma abordagem de busca por uma solução ótima factível realizando apenas uma enumeração parcial do espaço de soluções, recursivamente particionando o espaço e eliminando regiões que não são promissoras. O método envolve o cálculo de um limitante inferior (no caso da minimização de uma função-objetivo), ou de um limitante superior (caso a função-objetivo deva ser maximizada). Esse limitante é geralmente obtido através de técnicas de relaxação e é eficiente quando é respeitada a propriedade de monotocidade da solução. O algoritmo branch-and-cut combina o algoritmo branch-and-bound e o método de corte de plano. Ao contrário da busca exaustiva, a aplicação desses métodos não é imediata devido à necessidade de se trabalhar com limitantes.

### **2.3.4 Meta-Heurísticas via Abordagens Não-Populacionais**

#### **2.3.4.1 Busca Tabu**

A busca tabu (GLOVER & LAGUNA, 1997; GLOVER & LAGUNA, 1998; REEVES, 1993; ZACHARIASEN & DAM, 1996) é uma meta-heurística de escopo local provida da habilidade de contornar pontos de ótimo local prematuros. Isso é realizado via uma memória seletiva que reflete o histórico da busca, considerando soluções obtidas no decorrer do processo de busca. Assim, restrições que delimitam o espaço de busca são sistematicamente impostas de modo que certas soluções se tornem proibidas ou tabu, no intuito de varrer regiões ainda inexploradas do espaço de soluções.

As origens da busca tabu datam de meados da década de 1960, mas sua versão atual foi formulada por Fred Glover em 1986 (GLOVER & LAGUNA, 1996; 1998). Desde então, o método vem sendo aplicado com sucesso a uma ampla variedade de problemas combinatórios de alta complexidade (ZACHARIASEN & DAN, 1996; GLOVER & LAGUNA, 1996).

A busca tabu trabalha com base em uma única solução. O método de busca parte de uma solução inicial, e a cada iteração a vizinhança da solução corrente é analisada. Na seqüência, uma nova solução é selecionada do conjunto de vizinhos. Esse processo é repetido até que um critério de parada seja alcançado. A principal característica desse método é que ele opera com o objetivo de transcender a otimalidade local, permitindo a degradação da solução durante o processo de busca e até a ocorrência temporária de soluções ineficazes.

A busca tabu conserva um histórico do processo de busca com o objetivo de contornar pontos de ótimo local. Durante a busca, certas soluções (ou movimentos) são considerados proibidos ou tabu. Em geral, soluções ou partes de soluções recentemente visitadas ou frequentemente obtidas são classificadas como tabu. Existem variações da busca tabu propondo melhorias, tais como estratégias de diversificação e memórias adaptativas.

## Elementos da Busca Tabu

Uma das principais características da busca tabu é a *memória*. Os elementos básicos da memória tabu são:

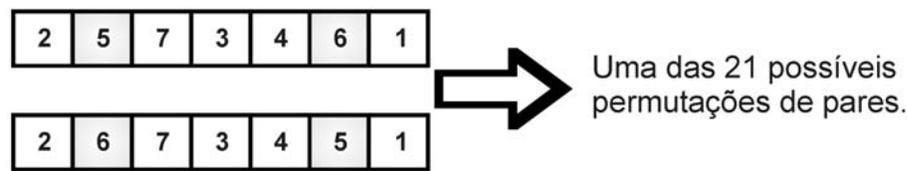
- *Memória de curto prazo*: constitui uma memória de recentidade, na qual as últimas soluções obtidas durante a busca são armazenadas. Essa estratégia objetiva evitar estagnações em ótimos locais prematuros.
- *Memória de médio prazo*: baseia-se em informações relacionadas a pontos de ótimo local de boa qualidade obtidos no decorrer do processo de busca. Visa explorar regiões promissoras do espaço, nas quais existe grande probabilidade de se encontrarem ótimos globais ou ótimos locais de boa qualidade. A estratégia baseada em memória de médio prazo é denominada *intensificação*.
- *Memória de longo prazo*: informações referentes a todo o processo de busca são utilizadas para guiar a pesquisa para regiões ainda inexploradas do espaço. Constitui um processo de *diversificação*.

A busca tabu explora o espaço considerando o conceito de *vizinhança* entre soluções. Assim, dada uma solução atual válida  $T$ , um conjunto de soluções vizinhas  $N(T)$  pode ser obtido através da aplicação de transformações sobre  $T$ . Com base no histórico de busca, a busca tabu impõe certas restrições ao espaço, reduzindo a vizinhança  $N(T)$  para  $N(T,H)$ , onde  $H$  é a memória tabu.

### A Memória de Recentidade

A busca tabu opera sob a hipótese de que uma vizinhança pode ser construída com base na solução atual. Trocas de elementos aos pares (REEVES, 1993) é uma técnica comumente utilizada para definição de vizinhança em problemas de otimização combinatória. A título de exemplificação, considere o problema simples de permutação ilustrado na Figura 2.3. Conforme pode ser observado, uma solução adjacente foi obtida trocando-se os elementos 5 e 6. Considerando que as trocas sejam sempre efetuadas envolvendo pares de elementos, têm-se

21 possibilidades de soluções adjacentes que caracterizam a vizinhança completa, dada pelo seguinte conjunto de permutações de pares de elementos:  $\{(2,5), (2,7), (2,3), (2,4), (2,6), (2,1), (5,7), (5,3), (5,4), (5,6), (5,1), (7,3), (7,4), (7,6), (7,1), (3,4), (3,6), (3,1), (4,6), (4,1), (6,1)\}$ . A cada movimento encontra-se associado um “valor de movimento”, o qual reflete o efeito da troca no valor da função-objetivo. O valor de movimento é fundamental para a avaliação da qualidade do movimento, embora outros critérios também possam ser importantes. Dependendo do problema tratado e das restrições associadas, podem ser empregados variados tipos de movimentos para geração de soluções vizinhas, como movimentos de inserção ou troca de dois ou mais elementos. Por questões de simplicidade e para facilitar a compreensão da operação da memória, nesta seção serão considerados, sem perda de generalidade, apenas movimentos de troca de elementos aos pares.



**Figura 2.3.** Definição de vizinhança e soluções adjacentes.

O principal mecanismo utilizado para guiar o processo de procura pela melhor solução na busca tabu é classificar um conjunto de movimentos para geração de vizinhança como proibidos ou tabu, de modo a produzir uma função-objetivo modificada que não considere somente o valor do movimento. Essa classificação depende do histórico de busca, especialmente de informações armazenadas na tabela de recentidade. A tabela de recentidade armazena os movimentos de troca realizados em iterações recentes, evitando que movimentos realizados recentemente sejam revertidos devido à presença de pontos de ótimo local. A Figura 2.4 ilustra a tabela de recentidade utilizada na busca tabu, encontrando-se destacados movimentos efetuados nas três últimas iterações. Para evitar que movimentos realizados recentemente sejam revertidos, eles são proibidos durante um determinado número de iterações. No exemplo, o movimento de troca 1-7 está proibido por 4 iterações, o movimento 4-5 está proibido por 3 iterações, e o movimento 2-3 por 2 iterações. O número de iterações

consideradas tabu é decrementado a cada iteração. No caso particular do problema simples de permutação colocado na Figura 2.3, a interpretação da tabela de recentidade expressa na Figura 2.4 é a seguinte: o movimento de troca 1-7 está proibido durante 4 iterações, ou seja, o elemento com rótulo 1 não poderá trocar de posição com o elemento 7 durante 4 iterações consecutivas, a não ser que uma condição denominada *critério de aspiração*, a ser descrita na próxima subseção, seja satisfeita.

A duração tabu de cada movimento afeta significativamente a qualidade da solução final, bem como o tempo computacional da busca. Um estudo acerca de durações tabu ótimas para o problema do caixeiro viajante pode ser encontrado em TSUBAKITANI & EVANS (1998).

	1	2	3	4	5	6	7
1							4
	2		2				
		3					
			4		3		
				5			
					6		
						7	

**Figura 2.4.** Tabela de recentidade.

### Critério de Aspiração

Restrições classificadas como tabu não são sempre invioláveis. Uma condição tabu pode ser revogada caso satisfaça o *critério de aspiração*, realizando um “esquecimento estratégico”. Um tipo de critério de aspiração comumente utilizado baseia-se no valor da função-objetivo. Caso a realização de um movimento classificado como tabu produza na função-objetivo um valor melhor do que todos os valores obtidos anteriormente, o critério de aspiração é satisfeito e o movimento pode ser realizado.

## Memória de Médio Prazo

A memória de curto prazo (recentidade) visa evitar a ocorrência de ciclagens e repetições de soluções, contornando ótimos locais prematuros. Entretanto, não fornece garantia de que o espaço de busca seja satisfatoriamente coberto. A memória de médio prazo armazena soluções de boa qualidade encontradas durante todo o processo de busca. Dessa forma, a memória de médio prazo almeja identificar regiões promissoras do espaço de busca, em cuja vizinhança ótimos globais ou locais satisfatórios apresentam grande probabilidade de serem encontrados.

## Memória de Longo Prazo

A memória de longo prazo baseia-se na frequência de realização dos movimentos. Movimentos realizados muito frequentemente são penalizados, de modo a guiar o processo de busca para regiões ainda inexploradas do espaço de soluções.

## O Status Tabu

Com o objetivo de evitar ciclagens, repetições e controlar o processo de busca, a cada solução vizinha da solução corrente encontra-se associado um *estado tabu*. Dessa forma, uma determinada solução pode ser classificada como tabu ou não-tabu. O estado tabu modifica a vizinhança, gerando uma nova vizinhança reduzida composta somente pelas soluções não-tabu.

## Seleção da Melhor Solução Vizinha

Existem diversos critérios para seleção da melhor solução vizinha. Dentre eles, destacam-se:

- Melhor de todas: avaliam-se todos os candidatos pertencentes à vizinhança e seleciona-se aquele que representar melhor resultado na função-objetivo. Os problemas relacionados a este critério estão associados a restrições de custo computacional, devido à necessidade de avaliação de todos os candidatos na vizinhança.
- Primeiro movimento de melhora: seleciona-se o primeiro movimento que acarrete melhora na função-objetivo.

- Primeiro movimento que acarrete melhora na função-objetivo maior que um limiar  $\tau$ .
- Melhor movimento dentre os  $C$  ( $C$  aleatório) primeiros movimentos.
- Melhor movimento dentre os  $C$  ( $C$  constante) primeiros movimentos.
- Melhor movimento dentre os  $C$  ( $C$  constante) primeiros movimentos ou primeiro movimento de melhora.

### 2.3.4.2 Recozimento Simulado (*Simulated Annealing*)

O recozimento simulado é a meta-heurística mais antiga e um dos primeiros algoritmos a adotar uma estratégia explícita para escapar de pontos de ótimo local. O método é inspirado no processo de aquecimento de metais e vidros, que assumem um estado de baixa energia quando resfriados de maneira lenta e gradual.

A idéia básica é contornar pontos de ótimo local permitindo a realização de movimentos que conduzam temporariamente a soluções de pior qualidade que a atual. A probabilidade de se aceitarem movimentos que levem a soluções de pior qualidade decresce ao longo do processo de busca. O algoritmo inicia-se com a geração de uma solução inicial  $s$ , que pode ser obtida aleatoriamente ou de acordo com alguma outra heurística. Em seguida, é inicializado um parâmetro de temperatura  $T$ , o principal parâmetro de controle do recozimento simulado. A cada iteração, uma solução vizinha  $s'$  é tomada aleatoriamente do conjunto de soluções vizinhas  $N(s)$  da solução atual. A aceitação ou não da nova solução  $s'$  leva em consideração três elementos: o custo da solução corrente  $s$ , o custo da solução  $s'$  em análise e o valor do parâmetro  $T$ . Considerando um problema de minimização, caso o valor da função-objetivo da solução em análise seja menor que o valor da função-objetivo da solução atual, a solução em análise é aceita. Caso contrário, a solução será aceita com uma probabilidade  $P$ , geralmente calculada de acordo com a distribuição de Boltzmann:

$$P(s, s', T) = \exp\left(-\frac{f(s') - f(s)}{T}\right). \quad (2.2)$$

A seguir é apresentado um pseudocódigo para o recozimento simulado:

```
Procedimento RecozimentoSimulado  
  Gerar solução inicial  $s$   
  Inicializar temperatura  $T$   
  Enquanto não atingir condição de parada faça  
    Selecionar solução aleatoriamente  $s'$  de  $N(s)$   
    Se  $f(s') < f(s)$  então  $s \leftarrow s'$   
    senão aceitar  $s'$  com probabilidade  $P(s, s', T)$   
    Atualizar temperatura  $T$   
Fim Enquanto  
Fim Procedimento
```

**Algoritmo 2.1.** Recozimento simulado: versão para minimização.

### 2.3.5 Meta-Heurísticas via Abordagens Populacionais

Algoritmos populacionais trabalham com o objetivo de explorar eficientemente o espaço de busca com base em informações obtidas a partir do conjunto de indivíduos que compõem a população a cada iteração do processo de busca.

Existem diversas abordagens populacionais destinadas à solução de problemas combinatórios: algoritmos genéticos (FOGEL, 1995; HOLLAND, 1975), colônias de formigas (DORIGO & DI CARO, 1999a; DORIGO & DI CARO, 1999b), enxame de partículas (KENNEDY *et al.*, 2001) e sistemas imunológicos artificiais (DE CASTRO & TIMMIS, 2002). Dentre as abordagens populacionais, aquela que tem recebido maior destaque na literatura é representada pelos algoritmos genéticos e suas variantes. As subseções 2.3.5.1 e 2.3.5.2 oferecem uma descrição acerca dos algoritmos genéticos. Uma nova abordagem baseada em população que vem ganhando destaque são os sistemas imunológicos artificiais. Os sistemas imunológicos artificiais operam de forma similar aos algoritmos genéticos, porém apresentam mecanismos sofisticados voltados para manutenção da diversidade e dimensionamento automático da

população. Esses mecanismos são inspirados no sistema imunológico biológico e não se encontram explicitamente presentes em outras abordagens populacionais. Uma das contribuições desta tese envolve a proposição de uma nova heurística para tratamento de problemas combinatórios baseada em sistemas imunológicos artificiais. O Capítulo 6 fornece uma descrição detalhada dessa nova metodologia.

### *2.3.5.1 Algoritmos Genéticos*

Os algoritmos genéticos (HOLLAND, 1975) são os mais conhecidos representantes dos chamados algoritmos evolutivos. Algoritmos evolutivos são métodos de busca baseados na seleção natural e na sobrevivência dos indivíduos mais adaptados. Diferem da maioria dos métodos de otimização por se basearem em uma população de soluções, e não em uma única solução. A cada iteração, os indivíduos da população (possíveis soluções candidatas) evoluem através de processos inspirados na natureza, tais como reprodução, mistura de carga genética, mutação de genes, seleção de melhores indivíduos e eliminação de indivíduos mais fracos ou menos adaptados. Os algoritmos evolutivos constituem métodos de busca estocásticos destinados à solução aproximada de problemas de elevada complexidade, cuja formulação matemática não seja conhecida ou cujos requisitos sejam difíceis de serem atendidos via abordagens de otimização convencionais.

#### ***Terminologia***

- População ou genoma: uma característica fundamental dos algoritmos evolutivos é o emprego da abordagem populacional. Algoritmos evolutivos operam de forma paralela com base em uma população de candidatos à solução do problema.
- Cromossomo: nos algoritmos genéticos convencionais, cada cromossomo é composto por vários genes e define um indivíduo da população, ou seja, uma solução candidata para o problema. Geralmente a representação computacional de uma solução candidata é um cromossomo formado por uma lista de atributos que caracterizam unicamente aquela solução candidata.

- Gene: os cromossomos são formados por genes; dessa forma, os genes são componentes de uma solução. A título de exemplificação, considere o problema do caixeiro viajante, descrito na Seção 2.2.1. O objetivo é definir a seqüência em que as cidades serão visitadas de forma que a distância total percorrida seja minimizada. Neste caso, cada gene será representado por um valor inteiro correspondendo ao índice de uma cidade, e um cromossomo fornecerá uma seqüência completa de cidades. Dependendo da aplicação, os genes também podem ser expressos como valores binários ou valores reais.
- Função de fitness: a função de fitness mede o nível de adaptabilidade do indivíduo ao ambiente. Esta adaptabilidade ao ambiente é um conceito abstrato diretamente vinculado à capacidade que a solução candidata tem de resolver o problema. Indivíduos mais adaptados apresentam maior fitness. O fitness é dado por uma função-objetivo a ser maximizada. Tomando novamente como exemplo o problema do caixeiro viajante, uma possível função de fitness seria o inverso da soma das distâncias entre pares de cidades adjacentes, de acordo com a seqüência definida pelo cromossomo. Deste modo, menores somas das distâncias levam a valores maiores para a função de fitness.
- Recombinação ou crossover: combinação de dois ou mais indivíduos através de operadores específicos, realizando troca de informações entre soluções candidatas. Certos procedimentos de recombinação podem levar à produção de soluções infactíveis. Dessa forma, nesta etapa podem ser necessários operadores específicos que garantam a factibilidade, ou o uso de mecanismos que restaurem a factibilidade da solução após a recombinação. Existem diversos operadores de recombinação propostos na literatura, tanto operadores genéricos como operadores direcionados a certas aplicações (MICHALEWICZ, 1997).
- Mutação: modifica um ou mais genes aleatoriamente, de acordo com uma *taxa de mutação*. Assim como ocorre com os operadores de crossover, existem diversas propostas para operadores de mutação (MICHALEWICZ, 1996).

### **Algoritmo**

A estrutura de um algoritmo genético clássico é brevemente descrita no Algoritmo 2.2.

**Procedimento Algoritmo Genético**

Gerar população inicial

Avaliar fitness da população

**Enquanto** não atingir condição de parada **faça**

Selecionar cromossomos-pai

Aplicar operadores de recombinação

Aplicar operadores de mutação

Atualizar população

Avaliar fitness da população

**Fim Enquanto****Fim Procedimento****Algoritmo 2.2.** Algoritmo genético.

O processo seletivo comumente utilizado pelos algoritmos genéticos clássicos é o método da roleta (do inglês *roulette wheel*) (MICHALEWICZ, 1997). Neste método, a cada indivíduo da população é atribuída uma probabilidade de ser selecionado para a próxima geração. Essa probabilidade é calculada com base no fitness do indivíduo em relação ao fitness dos demais indivíduos da população. Quanto maior o fitness relativo, maior a probabilidade de o indivíduo permanecer na população.

Com o objetivo de garantir a manutenção da diversidade e da qualidade da população, uma grande variedade de estratégias seletivas alternativas foram propostas: seleção por torneio, seleção por diversidade, biclassista, salvacionista, método da seleção universal estocástica, como seleção baseada em nichos (MILLER & SHAW, 1996), dentre muitas outras (SIMÕES, 1999).

Enfim, atualmente muitas propostas de modificações para os algoritmos genéticos clássicos foram concebidas em vários níveis: operadores de recombinação e mutação específicos (SIMÕES, 1999), algoritmos híbridos, tais como algoritmos meméticos, ou diferentes mecanismos de seleção.

Entretanto, cabe aqui destacar que grande parte das proposições para melhoria de desempenho se distanciam da motivação biológica original dos algoritmos genéticos convencionais, incluindo características como comportamento social e evolução cultural.

### 2.3.5.2 Algoritmos Meméticos

Os algoritmos meméticos (MOSCATO, 2003; MENDES 2003) são também denominados *algoritmos genéticos híbridos*, pois basicamente combinam um algoritmo genético clássico com procedimentos de busca local. Assim como os algoritmos genéticos, os algoritmos meméticos constituem uma abordagem populacional destinada à solução de problemas de otimização complexos.

Em termos de resultados práticos, os algoritmos meméticos têm alcançado um desempenho superior ao dos algoritmos genéticos convencionais. Em geral, os resultados obtidos via algoritmo híbrido apresentam melhor qualidade e requerem menor tempo de execução do que os genéticos puros e procedimentos de busca local isolados, conduzindo a indivíduos com cargas genéticas mais adaptadas. Isso possibilita a exploração das bases de atração da superfície de busca sem despendar grande esforço computacional. Diversas abordagens vêm sendo propostas com o objetivo de reduzir o tempo computacional e acelerar a convergência dos algoritmos, tais como populações hierarquicamente estruturadas (MOSCATO 2003), migração de indivíduos e processamento paralelo (GARCIA *et al.*, 2001).

O Algoritmo 2.3 apresenta uma possível versão de algoritmo memético. Certas características constituem responsabilidade do planejador do algoritmo, tais como em que ponto aplicar o procedimento de busca local (antes ou depois da aplicação de operadores de recombinação), ou então quantos e quais indivíduos serão otimizados localmente. A busca local tende a ser específica para a aplicação em questão.

**Procedimento Algoritmo Memético**

Gerar população inicial

Avaliar fitness da população

**Enquanto** não atingir condição de parada **faça**

Selecionar cromossomos-pai

Aplicar operadores de recombinação e mutação

Aplicar busca local aos indivíduos da população

Atualizar população

Avaliar fitness da população

**Fim Enquanto****Fim Procedimento****Algoritmo 2.3.** Algoritmo memético.

### 2.3.5.3 Sistemas Imunológicos Artificiais

Conforme mencionado anteriormente, os sistemas imunológicos artificiais fundamentam uma das meta-heurísticas propostas neste trabalho e são tratados em um capítulo à parte. O Apêndice B fornece os fundamentos teóricos do sistema imunológico biológico e o Capítulo 5 apresenta os principais conceitos acerca dos sistemas imunológicos artificiais. No Capítulo 6 é proposta a *copt-aiNet* (acrônimo para a expressão em inglês *Artificial Immune Network for Combinatorial Optimization*), uma nova meta-heurística baseada em sistemas imunológicos artificiais destinada à solução de problemas combinatórios.

### 2.3.5.4 Otimização por Enxame de Partículas (PSO - Particle Swarm Optimization)

Otimização por enxame de partículas (Kennedy & Eberhart, 1995; Eberhart & Kennedy, 1995; Eberhart & Shi, 1988; Eberhart & Shi, 2001) é um método de otimização estocástico populacional inspirado no comportamento social de peixes, aves e insetos.

O PSO é inicializado com uma população de partículas posicionadas aleatoriamente no espaço de busca, sendo que cada partícula corresponde a uma solução candidata para o problema. A

cada iteração, a posição das partículas é atualizada de acordo com duas soluções de referência: a primeira diz respeito à solução **IMax** com melhor valor da função-objetivo encontrado até o momento pela partícula a ser atualizada. A segunda solução **gMax** é aquela que apresentou melhor valor de função-objetivo encontrado até o momento, considerando todas as partículas (ou uma vizinhança topológica, dependendo da implementação) e seu histórico, ou seja, o melhor valor alcançado por qualquer elemento da população de partículas até o presente momento. Após encontrar os dois valores de referência, a velocidade e a posição da partícula são atualizadas de acordo com as fórmulas:

$$\mathbf{v}(i+1) = \mathbf{v}(i) + c_1 \cdot \text{rand\_num} \cdot (\mathbf{IMax}(i) - \mathbf{IAtual}(i)) + c_2 \cdot \text{rand\_num} \cdot (\mathbf{gMax}(i) - \mathbf{IAtual}(i)), \quad (2.3)$$

$$\mathbf{IAtual}(i+1) = \mathbf{IAtual}(i) + \mathbf{v}(i), \quad (2.4)$$

onde  $\mathbf{v}(i)$  é denominada velocidade da partícula na geração  $i$ ,  $\mathbf{IMax}(i)$  é a melhor solução obtida pela partícula até a geração  $i$ ,  $\mathbf{gMax}(i)$  é a melhor solução global encontrada pelo otimizador até a geração  $i$ ,  $\mathbf{IAtual}(i)$  é a solução atual expressa pela partícula  $i$ ,  $\text{rand\_num}$  é um valor aleatório entre 0 e 1 e  $c_1$  e  $c_2$  são constantes que denotam a importância relativa dos termos local e global. Usualmente  $c_1 = c_2 = 2$ .

O pseudo-código para o procedimento é apresentado no Algoritmo 2.4.

**Procedimento EnxameDePartículas**

Inicializar aleatoriamente cada partícula

**Repita**

**Repita** para cada partícula

Avaliar função-objetivo  $f_{Atual}$

**Se**  $f_{Atual} < f_{Best}$  **então**  $f_{Best} = f_{Atual}$

**Fim Repita**

**Para** cada partícula

Calcular a velocidade da partícula de acordo com a equação 2.3

Atualizar a posição da partícula de acordo com a equação 2.4

**Fim Para**

**Até** que critério de parada seja atingido

**Fim Procedimento**

**Algoritmo 2.4.** Otimização por enxame de partículas (versão para minimização).

### 2.3.5.5 Otimização por Colônia de Formigas (ACO – Ant Colony Optimization)

Colônia de formigas (DORIGO & DI CARO, 1999a; DORIGO & DI CARO, 1999b) constitui uma abordagem meta-heurística recente para tratar problemas de caráter combinatório. O algoritmo é inspirado no comportamento de formigas reais, que utilizam essências aromáticas denominadas feromônio como meio de comunicação. As formigas são capazes de encontrar o caminho de menor distância entre o formigueiro e uma fonte de alimento. Cada formiga marca o seu percurso com feromônio e o caminho marcado é empregado como referência por outras formigas, sendo que a concentração de feromônio decai com o tempo. O ACO é baseado na comunicação indireta (através de trilhas de feromônio artificial) de uma coleção de agentes simples denominados *formigas artificiais*. As trilhas de feromônio artificial constituem informações numéricas empregadas pelas formigas artificiais na construção de soluções para o problema. As soluções são construídas heurísticamente com base em um componente

estocástico e em trilhas de feromônio artificial. O componente estocástico permite que as formigas construam uma grande variedade de diferentes soluções. Um aspecto importante do algoritmo é que a experiência das formigas, adquirida ao longo do processo de busca, pode ser empregada na construção das novas soluções.

Para ilustrar a operação do algoritmo, considere um problema do caixeiro viajante simétrico com  $n$  cidades. Modelando o problema utilizando teoria de grafos, teremos um grafo totalmente conectado, as cidades  $x_i$ ,  $i = 1, \dots, n$  como vértices, e  $c_{ij}$  o custo de se viajar da cidade  $i$  à cidade  $j$ . Cada formiga é capaz de gerar uma solução explorando este grafo em busca do trajeto de custo mínimo.

Inicialmente, cada formiga é colocada em uma cidade escolhida aleatoriamente e possui uma memória que armazena a solução parcial construída até o momento. Na partida, a memória contém apenas a cidade inicial.

É importante ressaltar que as formigas se movem concorrentemente e de forma independente, de modo que uma formiga isolada é capaz de gerar uma solução completa para o problema. Boas soluções, entretanto, emergem da interação das formigas, resultante do comportamento coletivo. Partindo de sua cidade de origem, uma formiga iterativamente move de cidade a cidade. Quando posicionada em uma cidade  $i$ , uma formiga  $k$  escolhe ir para uma cidade  $j$  ainda não visitada com probabilidade dada por:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad \text{se } j \in N_i^k, \quad (2.5)$$

onde  $\tau_{ij}(t)$  é a concentração de feromônio na aresta  $(i,j)$  no instante  $t$ , enquanto  $\eta_{ij} = 1/c_{ij}$  é um termo heurístico que indica que custos mais elevados tendem a reduzir as chances daquela aresta ser escolhida. Os parâmetros  $\alpha$  e  $\beta$  determinam a influência relativa da trilha de feromônio e do termo heurístico sobre o processo heurístico e  $N_i^k$  é a vizinhança factível para a formiga  $k$ . A trilha de feromônio  $\tau_{ij}(t)$  indica a conveniência de se passar pela aresta que conecta a cidade  $i$  à cidade  $j$ . Ao concluir o percurso e dispondo da qualidade relativa do

mesmo, cada formiga deposita uma quantidade de feromônio nas arestas que ela percorreu, sendo que esta quantidade é proporcional à qualidade relativa do percurso. Assim, arestas que compõem percursos de melhor qualidade receberão mais feromônio. A trilha de feromônio sofre evaporação a cada iteração e a uma taxa fixa.

A seguir, é apresentado um algoritmo para o método de otimização através de colônia de formigas, desenvolvido para resolver o problema do caixeiro viajante.

### Procedimento ColôniaDeFormigas

#### Repita

Criar as formigas e as distribuir aleatoriamente, uma em cada nó

$t=0$

#### Repita

$t=t+1$

#### Para cada formiga $k$ faça

Selecionar próximo nó (cidade) de acordo com  $p_{ij}^k(t)$ , dado pela Equação 2.5.

#### Fim para

**Até** que todas as formigas tenham construído uma solução completa

Avaliar as soluções encontradas

Atualizar trilha de feromônio

Aplicar regra de evaporação de feromônio

Armazenar melhor solução até o momento

**Até** que critério de parada seja satisfeito

#### Fim Procedimento

**Algoritmo 2.5.** Otimização por colônia de formigas.

### **2.3.6 Redes Neurais Artificiais**

Uma ferramenta de inteligência computacional que tem sido empregada na solução de problemas de otimização combinatória são as redes neurais artificiais. Elas são compostas por várias unidades de processamento (neurônios), interconectadas via conexões sinápticas. As duas principais classes de redes neurais artificiais usadas na solução de problemas combinatórios são as redes de Hopfield (HOPFIELD & TANK, 1985; HAYKIN, 1999) e os mapas auto-organizáveis (KOHONEN, 1982a; KOHONEN, 1982b; KOHONEN, 1984; KOHONEN, 1997; KOHONEN *et al.*, 2001; HAYKIN, 1999).

Conforme descrito nas seções anteriores, as meta-heurísticas para problemas combinatórios trabalham, em sua maioria, com base em permutações. A exploração completa do espaço de busca leva a uma explosão de candidatos à solução, inviabilizando a aplicação de algoritmos exatos. Um aspecto importante relacionado à aplicação de redes neurais a problemas combinatórios é que elas não são baseadas em permutações. Assim, as redes neurais podem constituir uma alternativa interessante para problemas de grande porte quando se dispõe de recursos computacionais limitados. Além disso, conforme será abordado no Capítulo 4, as redes neurais tendem a fornecer soluções estruturalmente diferentes daquelas fornecidas por meta-heurísticas baseadas em permutação. Este é um aspecto interessante quando se utilizam abordagens híbridas, tais como algoritmos genéticos em conjunto com redes neurais. Experimentos práticos mostram que a inclusão de soluções geradas por mapas auto-organizáveis na população de um algoritmo memético pode levar a uma maior diversidade da população (GOMES & VON ZUBEN, 2002a-2002e).

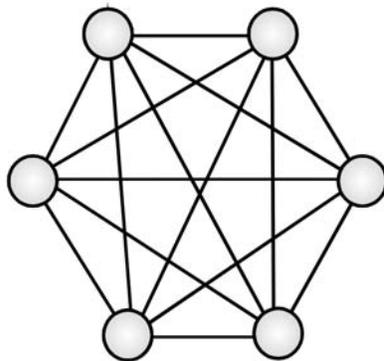
Embora grande parte dos esforços na aplicação de redes neurais a problemas combinatórios tenha se concentrado na solução do problema do caixeiro viajante simétrico, nos últimos anos diversas outras classes de problemas têm recebido crescente atenção, tais como roteamento de veículos e múltiplos caixeiros viajantes. Uma revisão das principais aplicações de redes neurais artificiais a problemas combinatórios pode ser encontrada em SMITH (1999).

### 2.3.6.1 Redes de Hopfield

Esta seção oferece uma breve introdução acerca da aplicação de redes de Hopfield a problemas de caráter combinatório e pressupõe que o leitor esteja familiarizado com os conceitos básicos. Mais detalhes podem ser encontrados em HOPFIELD & TANK (1985).

Os primeiros esforços direcionados para a solução do TSP via redes neurais artificiais concentraram-se na rede de Hopfield. HOPFIELD & TANK (1985) mostraram que, embora soluções ótimas pudessem ser encontradas, elas nem sempre produziriam rotas válidas e estariam sujeitas a efeitos de mínimos locais presentes na função de energia. Além disso, representar problemas tipo TSP através de redes neurais desse tipo requer grande quantidade de memória e acarreta alto custo computacional, além de permitir o tratamento de apenas um pequeno número de cidades (menor que 50).

A rede de Hopfield é uma rede neural recorrente totalmente conexa e que opera em um espaço de estados discreto. Sua arquitetura é composta por unidades de processamento não-lineares totalmente conectadas entre si, conforme ilustrado na Figura 2.5. Nesta seção será abordado o modelo discreto, em que cada unidade de processamento constitui um neurônio de MCCULLOCH & PITTS (1943).



**Figura 2.5.** Rede de Hopfield: conjunto de neurônios totalmente conectados via ligações sinápticas.

A rede de Hopfield surgiu como uma *memória endereçável por conteúdo* (CAM – *Content-addressable memory*) com o propósito de efetuar tarefas de recuperação de informação com base em somente informações parciais ou ruidosas (HOPFIELD, 1982). Possui matriz de pesos sinápticos simétrica e a cada conexão sináptica entre os neurônios  $i$  e  $j$  é associado um peso  $w_{i,j}$ . Os pesos podem ser excitatórios ou inibitórios, sendo representados por valores positivos e negativos, respectivamente.

Quando aplicada à otimização do problema do caixeiro viajante, o objetivo da rede de Hopfield é encontrar o estado que minimiza uma função-custo, no caso uma superfície de energia, de modo a encontrar o estado de menor energia. Assim, a instância do TSP deve ser codificada de modo a ser representada por uma função de energia em que o estado de menor energia coincide com o trajeto de menor custo/distância total. A matriz de pesos sinápticos é definida com base nas distâncias entre as cidades e nas restrições do problema, como, por exemplo, visitar somente uma vez cada cidade. Cada solução possível para o problema corresponde a um estado de equilíbrio do sistema.

A ativação de cada neurônio  $i$  é dada pela soma dos estados  $v_j$  dos demais neurônios conectados a ele, ponderada pelas respectivas conexões sinápticas, além de considerar a polarização  $b_i$  dos neurônios:

$$u_i = \sum (w_{i,j} \cdot v_j - b_i) \quad (2.6)$$

O estado do neurônio  $i$  é modificado de acordo com:

$$v_i = \begin{cases} +1 & \text{se } u_i > 0 \\ -1 & \text{se } u_i < 0 \\ s_{i-1} & \text{se } u_i = 0 \end{cases} \quad (2.7)$$

Supondo que haja 5 cidades, uma possível representação de uma solução é apresentada na Tabela 2.1. As linhas da tabela representam as cidades, enquanto as colunas representam a posição em que a cidade se encontra na seqüência dada pela solução. Portanto, a seqüência de cidades representada é: {5, 4, 2, 3, 1}.

No caso do TSP, o número de neurônios da rede de Hopfield será igual ao número de entradas na tabela, ou seja,  $n^2$ , onde  $n$  é o número de cidades. No caso do TSP, cada neurônio pode assumir somente o estado 0 ou 1. Existem conexões inibitórias entre todos os neurônios de uma mesma linha. Existem também conexões inibitórias entre todos os neurônios de uma mesma coluna. Assim, quando um neurônio de uma linha está ativo (1), todos os demais presentes em sua linha e em sua coluna devem estar inativos (0). As conexões inibitórias tendem a promover a satisfação de certas restrições do problema do caixeiro viajante: uma única cidade apontada em cada posição da seqüência e uma única aparição de cada cidade na seqüência.

**Tabela 2.1.** Representação de uma solução para uma instância do TSP.

	1	2	3	4	5
1	0	0	0	0	1
2	0	0	1	0	0
3	0	0	0	1	0
4	0	1	0	0	0
5	1	0	0	0	0

Os custos entre as cidades devem gerar os pesos sinápticos da rede de Hopfield. Sejam  $x$  e  $y$  cidades (correspondentes a linhas da Tabela 2.1) e  $i$  e  $j$  posições na solução (correspondentes a colunas na Tabela 2.1). Empregando a expressão da superfície de energia definida para a rede de Hopfield:

$$E = \frac{A}{2} \cdot \sum_{x=1}^n \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n v_{x,i} \cdot v_{x,j} + \frac{B}{2} \cdot \sum_{i=1}^n \sum_{x=1}^n \sum_{\substack{y=1 \\ y \neq x}}^n v_{x,i} \cdot v_{y,i} + \frac{C}{2} \cdot \left( \sum_{x=1}^n \sum_{i=1}^n v_{x,i} - n \right) + \frac{D}{2} \cdot \sum_{x=1}^n \sum_{y=1}^n \sum_{\substack{i=1 \\ y \neq x}}^n d_{x,y} \cdot v_{x,i} \cdot (v_{y,i+1} + v_{y,i-1}) \quad (2.8)$$

As constantes  $A$ ,  $B$ ,  $C$  e  $D$  representam:

- $A$ : responsável pela existência de uma única cidade por linha na matriz-solução. Observe que, de acordo com o trecho da equação ponderado por  $A$ , existe um acréscimo na função de energia caso uma cidade apareça mais de uma vez no trajeto.
- $B$ : responsável pela existência de somente uma cidade por coluna na matriz-solução. Neste caso, a ocorrência de uma mesma posição mais que uma vez no trajeto acarreta um acréscimo no valor da função de energia.
- $C$ : fator de inibição global que possui por objetivo acelerar a convergência. O termo ponderado pela constante  $C$  visa contribuir para que todas as cidades apareçam no trajeto.
- $D$ : fator associado às distâncias entre cidades que tem por objetivo a convergência para a solução de menor distância total.

O peso de  $x_i$  para  $y_j$  é dado por:

$$w_{x_i y_j} = -A \cdot \delta_{x,y} (1 - \delta_{i,j}) - B \cdot \delta_{i,j} (1 - \delta_{x,y}) - C - D \cdot d_{x,y} (\delta_{j,i+1} + \delta_{j,i-1}), \quad (2.9)$$

onde  $\delta_{i,j}=1$  se  $i=j$  e  $\delta_{i,j}=0$  caso contrário.

O conjunto de equações expresso acima conduz o sistema a um estado de energia mínima local, após algumas iterações. O número de iterações necessárias para a convergência do sistema para o estado de energia mínima local depende de uma série de fatores, como os valores das constantes e a dimensão do problema (número de cidades).

É importante ressaltar que, devido à existência de atratores espúrios, pontos estáveis não necessariamente correspondem a soluções factíveis. Na prática, a factibilidade foi alcançada somente para instâncias de pequeno porte, envolvendo em média 10 cidades. Além disso, o método tende a ser computacionalmente custoso com o aumento do número de cidades. Por exemplo, para um problema com 10 cidades, são necessários 100 neurônios para se representar uma solução, e portanto  $100 \times 100 = 10000$  conexões sinápticas.

No entanto, a concepção de um hardware dedicado pode permitir o tratamento de problemas de caixeiro viajante com milhares e até milhões de cidades (KENNEDY & CHUA, 1987).

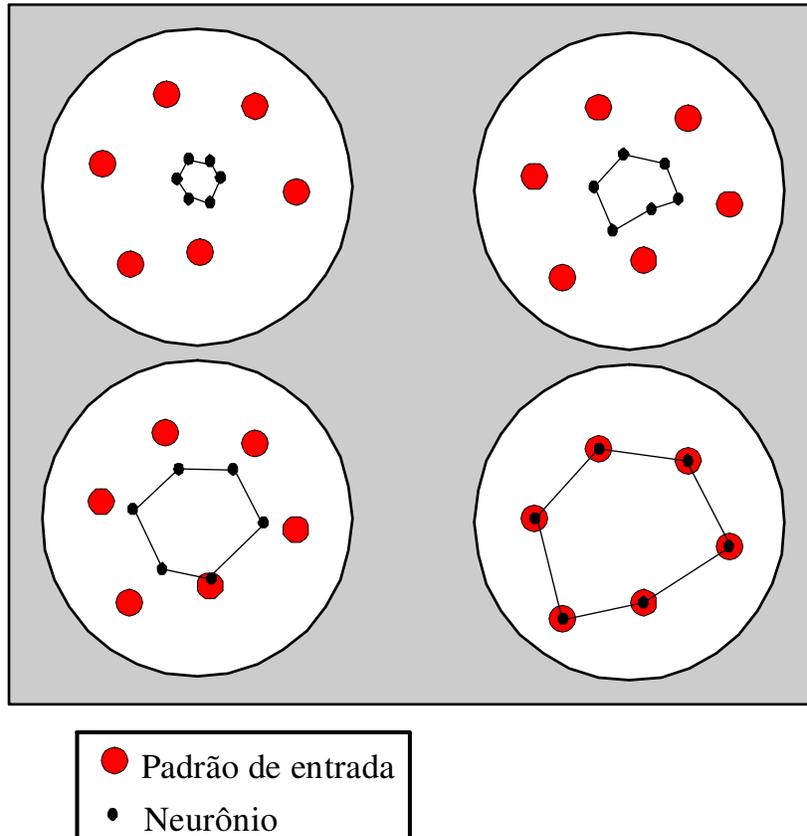
### 2.3.6.2 Mapas Auto-Organizáveis

Esta seção objetiva fornecer uma breve introdução acerca da aplicação de mapas auto-organizáveis a problemas de caráter combinatório e pressupõe que o leitor esteja familiarizado com os conceitos básicos. O Capítulo 3 oferece uma revisão teórica detalhada dos mapas auto-organizáveis.

Os mapas auto-organizáveis têm sido aplicados com relativo sucesso a problemas combinatórios, em particular de caminho mínimo (ANGÉNIOL *et al.*, 1988; GHAZIRI, 1991; GOMES & VON ZUBEN, 2001; GOMES & VON ZUBEN, 2002a-2002e; MATSUYAMA, 1996; MODARES *et al.*, 1999; VAKHUTINSKY & GOLDEN, 1994).

Os mapas auto-organizáveis podem ser diretamente aplicados a problemas combinatórios simples que envolvam permutações, tais como o caixeiro viajante simétrico. A Figura 2.6 ilustra a evolução de um mapa auto-organizável com seis neurônios aplicado a um conjunto de seis cidades.

Quando tratamos de problemas mais complexos, como o de roteamento de veículos, certos aspectos como inicialização sináptica, ajuste do tamanho do passo de atualização dos vetores de pesos e definição do neurônio vencedor são críticos. Uma das contribuições desta tese, detalhada no Capítulo 4, envolve a proposição de uma nova meta-heurística baseada em auto-organização para o problema de roteamento de veículos capacitados. A heurística proposta enfoca justamente os procedimentos de inicialização, atualização de pesos e definição do neurônio vencedor, lembrando que existirão múltiplos mapas auto-organizáveis competindo entre si.



**Figura 2.6.** Exemplo de mapas auto-organizáveis aplicados ao problema do caixeiro viajante simétrico.

### 2.3.7 Abordagens Híbridas

É consenso entre pesquisadores que o desempenho das meta-heurísticas está fortemente ligado às características do espaço de busca. A existência de tantos tipos de métodos de otimização e variações é consequência da observação prática e teórica de que a eficiência de um determinado método de otimização é dependente do problema a ser tratado. Em geral, diferentes métodos ou mesmo diferentes parâmetros relacionados a um método são freqüentemente necessários para abordar diferentes classes de problemas de otimização ou até instâncias distintas pertencentes a uma mesma classe. Por exemplo, para o caso de algoritmos genéticos, encontra-se uma grande variedade de operadores genéticos específicos para problemas e instâncias, além de variadas estratégias que objetivam escapar de ótimos locais:

busca em nichos, migração de indivíduos, populações hierárquicas, co-evolução etc. Uma determinada heurística que apresente o melhor desempenho para um problema de otimização combinatória em particular não necessariamente produzirá os melhores resultados para outros problemas ou outras instâncias. Este fato leva ao projeto de meta-heurísticas híbridas. Para certos problemas, devido a fatores como escala ou outras particularidades de certas instâncias, a eficiência das meta-heurísticas pode ser comprometida. Neste contexto, pode ser necessário integrar diversas meta-heurísticas, produzindo métodos mais robustos e eficientes. Tais métodos são conhecidos como *híbridos*. Em meta-heurísticas híbridas, diversos métodos trabalham em cooperação com a finalidade de produzir soluções melhores e/ou mais diversificadas.

Nos Capítulos 4 e 6 desta tese são propostas duas meta-heurísticas híbridas para problemas combinatórios: a primeira, descrita no Capítulo 4, envolve mapas auto-organizáveis de Kohonen, inferência nebulosa e busca tabu; a segunda abordagem, proposta no Capítulo 6, emprega sistemas imunológicos artificiais e busca tabu.

# CAPÍTULO 3

## MAPAS AUTO-ORGANIZÁVEIS DE KOHONEN

---

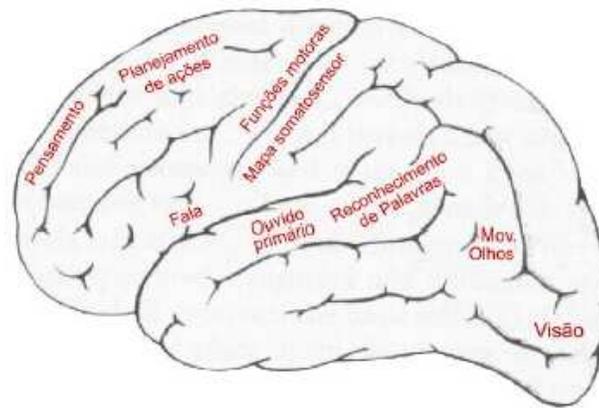
---

Sistemas auto-organizados apresentam comportamento emergente, de forma que os componentes interagem e produzem padrões de comportamento coletivo sem que haja necessidade de interferência direta de fatores externos. O conceito de auto-organização em redes neurais artificiais está ligado ao conceito de aprendizado não-supervisionado. Os mapas auto-organizáveis (SOM – do inglês *self-organizing maps*) constituem uma importante classe de redes neurais artificiais e são um dos principais representantes do paradigma de aprendizado não-supervisionado. Neste capítulo são apresentados os fundamentos de redes neurais auto-organizáveis de Kohonen, abordando aspectos referentes ao processo de treinamento, algoritmos e principais aplicações da metodologia. São brevemente apresentados alguns modelos de redes neurais derivados do paradigma de aprendizado não-supervisionado. Uma importante categoria de modelos derivados do SOM inclui as arquiteturas construtivas, em que é permitida a variação do número de neurônios durante o processo de treinamento.

### 3.1 Introdução

O córtex cerebral pode ser dividido em diversas regiões, nas quais diferentes informações sensoriais são representadas por mapas topologicamente ordenados e regiões distintas respondem a diferentes estímulos (BOHEME *et al.*, 1994). Dessa forma, é possível afirmar que existe uma região específica do córtex cerebral dedicada a estímulos auditivos, outra a estímulos visuais, por exemplo. Visando uma maior rapidez na resposta a um estímulo,

neurônios que participam de uma mesma função situam-se topologicamente próximos. Portanto, estímulos similares tendem a produzir padrões de atividade semelhantes e em regiões vizinhas junto ao mapa neural. Por exemplo, sinais visuais semelhantes, tais como segmentos de reta com inclinações similares, tendem a excitar neurônios próximos. A Figura 3.1 mostra uma representação das várias regiões corticais especializadas do cérebro humano.

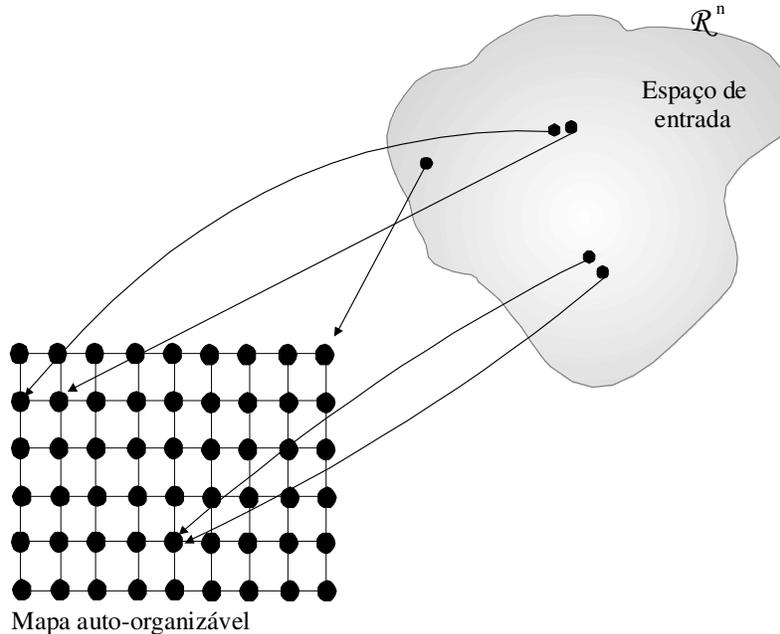


**Figura 3.1.** Regiões corticais especializadas do cérebro humano (ZUCCHINI, 2003).

Os mapas auto-organizáveis foram introduzidos por Teuvo Kohonen (KOHONEN, 1982a, KOHONEN, 1982b) e constituem um dos principais representantes do paradigma de aprendizado não-supervisionado (BARLOW, 1989). Podem ser definidos como modelos matemáticos na forma de redes neurais artificiais que procuram reproduzir computacionalmente e de forma simplificada este mecanismo biológico de especialização a estímulos. A Figura 3.2 ilustra o conceito acima. Observe que padrões de entrada similares, de acordo com algum critério, são mapeados para um mesmo neurônio ou para neurônios vizinhos no mapa auto-organizável.

Uma importante característica dos mapas auto-organizáveis é a capacidade de aprendizado a partir de estímulos do ambiente. O processo de aprendizado baseia-se em um conjunto de padrões de entrada, que correspondem aos estímulos do ambiente. Uma ordem global é alcançada através de interações locais, que ocorrem em dois níveis (HAYKIN, 1999): (i) a partir

da interação direta com o ambiente; e (ii) via relações de vizinhança entre neurônios situados topologicamente próximos.



**Figura 3.2.** Mapeamento de dados em um espaço de entrada para uma rede auto-organizável.

Em um mapa auto-organizável, cada neurônio possui um vetor de pesos sinápticos  $\mathbf{w} = [w_1, w_2, \dots, w_m]^T$ , onde  $m$  representa a dimensão do espaço de entrada. Os neurônios encontram-se distribuídos em um arranjo, geralmente de dimensão 1,2 ou 3, e competem entre si para serem ativados na presença de um padrão de entrada. Ao final do processo competitivo, que se dá após a apresentação de um dos padrões de entrada, haverá somente um neurônio vencedor. O aprendizado não-supervisionado conduz a rede neural a uma representação interna dos padrões de entrada, geralmente com redução dimensional, de modo a produzir um mapa neural que ainda seja capaz de refletir as características estatísticas mais relevantes do conjunto de padrões de entrada. Diferentemente dos paradigmas de aprendizado supervisionado (REED & MARKS, 1999) e por reforço (SUTTON & BARTON, 1998), no aprendizado não-supervisionado não existe um conjunto de rótulos ou saídas desejadas

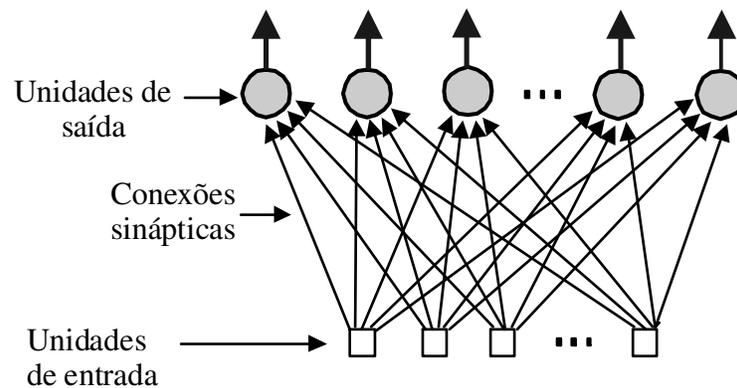
associadas aos padrões de entrada. Como os neurônios que compõem um mapa auto-organizável são estimulados pelos dados de entrada, os vetores de pesos sinápticos dos neurônios apresentam a mesma dimensão do espaço de entrada. Já o mapa de neurônios, normalmente restrito a uma dimensão inteira entre 1 e 3, pode apresentar dimensão igual ou inferior à dimensão do espaço de entrada, de modo que a preservação de certas propriedades estatísticas dos dados somente é possível perante a minimização da violação topológica, a qual pode ser estimada pela diferença entre a posição relativa dos dados e a posição relativa das excitações promovidas pelos dados junto ao mapa neural.

Atualmente, os mapas auto-organizáveis vêm sendo aplicados a diversos contextos, incluindo otimização combinatória (GOMES & VON ZUBEN, 2001; 2002; 2002a; 2002b; 2002c; GOMES *et al.*, 2002; 2002a), análise de dados (COSTA, 1999), agrupamento (clusterização), classificação (DUDA *et al.*, 2001), (FLEXER, 2001), robótica (RITTER *et al.*, 1992; BEKEY & GOLDBERG, 1993), processamento de imagem (SEIFFERT & MICHAELIS, 1995; BUREL & POTTIER, 1991), reconhecimento de padrões (CARPENTER & GROSSBERG, 1991), reconhecimento de voz (KOHONEN, 1984), bioinformática (FERRÁN & FERRARA, 1992; GOMES *et al.*, 2002a; GOMES *et al.*, 2002b), análise de dados econômicos (DEBOECK & KOHONEN, 1998), entre outros. Em grande parte das aplicações de mapas auto-organizáveis, o objetivo principal consiste em descobrir representações sucintas e concisas para conjuntos de dados de elevada cardinalidade e de elevada dimensão. Em geral, para esses conjuntos de dados, pouca informação relevante acerca das características estatísticas encontra-se disponível, razão pela qual os mapas auto-organizáveis se mostram extremamente úteis.

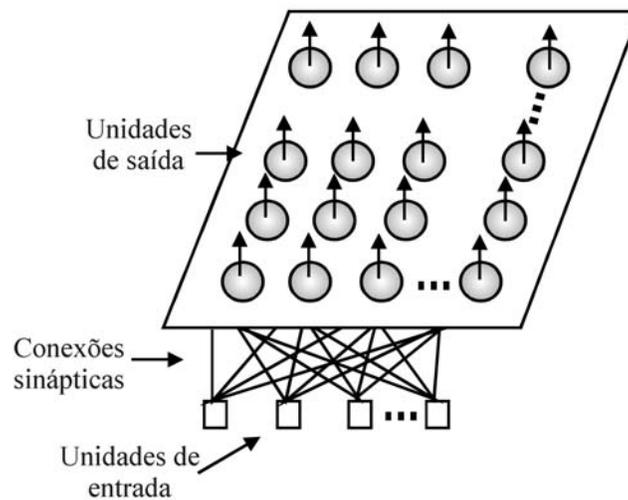
### **3.2 Arquitetura do Mapa Auto-Organizável de Kohonen**

Uma rede neural auto-organizável é composta por um conjunto de neurônios (unidades de saída), conexões sinápticas e um conjunto de unidades de entrada. Os pesos sinápticos dos neurônios são arranjados na forma de vetores  $m$ -dimensionais, onde  $m$  é o número de atributos dos padrões de entrada. Os neurônios encontram-se distribuídos em um arranjo e encontram-se totalmente conectados às unidades de entrada, conforme ilustrado nas Figuras 3.3 e 3.4 para os casos de arranjo de dimensão 1 e 2, respectivamente.

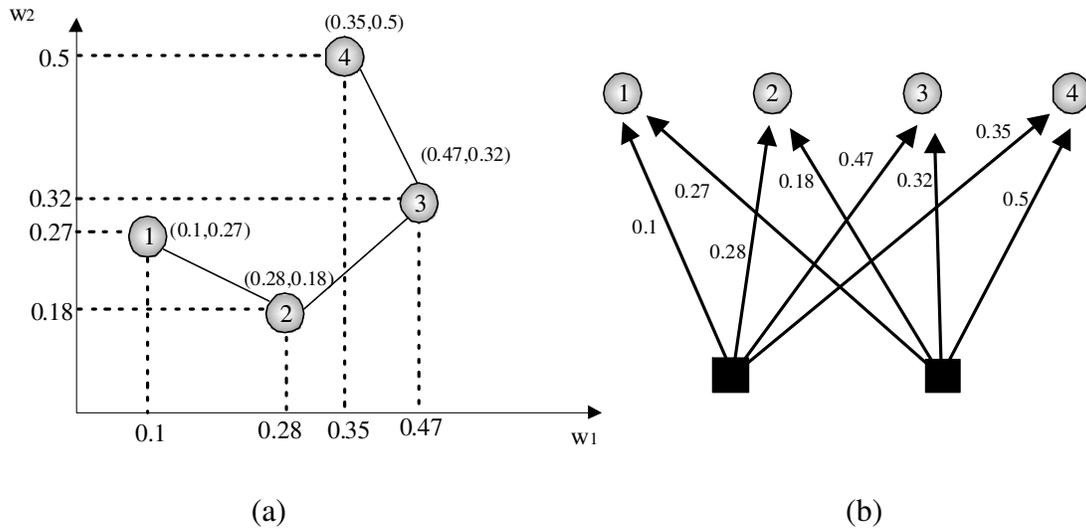
A Figura 3.5(a) exibe uma possível distribuição de pesos sinápticos no espaço de entrada para um mapa auto-organizável com  $m=2$  e arranjo de dimensão 1. Note que, embora o arranjo se modele de forma a permitir que os neurônios se posicionem livremente no espaço original dos dados, sua dimensão continua sendo unitária. Sendo assim, a interpretação do arranjo, após a conclusão do processo de auto-organização dos neurônios, se dará em um espaço unidimensional, independentemente da dimensão original dos dados. A Figura 3.5(b) ilustra o mapa auto-organizável correspondente.



**Figura 3.3.** Arquitetura típica de um mapa auto-organizável unidimensional.



**Figura 3.4.** Arquitetura típica de um mapa auto-organizável bidimensional.



**Figura 3.5.** (a) Possível distribuição dos pesos sinápticos  $w = [w_1 \ w_2]^t$  para o caso de duas entradas ( $m=2$ ) e mapa de dimensão 1, com linhas cheias indicando as relações de vizinhança unidimensional entre os neurônios; (b) Mapa auto-organizável unidimensional.

### 3.3 Treinamento de Mapas Auto-Organizáveis

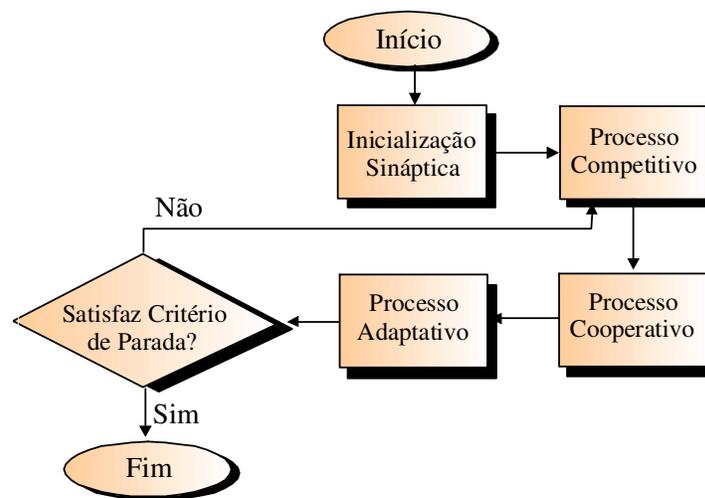
Durante o processo de treinamento de um mapa auto-organizável, os padrões de entrada são sucessivamente apresentados à rede neural. Existem duas principais formas de treinamento: (i) *seqüencial* e (ii) *em lote* ou *batelada*. No treinamento seqüencial, os padrões de entrada são seqüencialmente apresentados ao mapa auto-organizável e os pesos sinápticos dos neurônios são atualizados a cada apresentação de um padrão de entrada. Já no modo de treinamento em lote ou batelada, os pesos sinápticos são atualizados somente após um ciclo completo de apresentação de padrões. Isso é feito tomando-se a resultante das contribuições de todos os padrões para cada neurônio.

Todo algoritmo de treinamento de mapas auto-organizáveis apresenta 4 etapas fundamentais:

- Inicialização sináptica, em que os vetores de pesos sinápticos iniciais dos neurônios presentes na rede são definidos de acordo com algum critério pré-estabelecido.
- Processo competitivo, em que é selecionado o neurônio vencedor. O neurônio vencedor será aquele que apresentar maior nível de ativação para o padrão de entrada apresentado à rede neural.
- Processo cooperativo, em que é definida uma vizinhança topológica de neurônios associada ao vencedor, sendo que os vetores de pesos de todos os neurônios desta vizinhança sofrerão uma adaptação de mesma polarização.
- Processo adaptativo, em que os pesos sinápticos do neurônio vencedor e de sua vizinhança são modificados.

Na versão convencional do algoritmo, o número de neurônios é mantido fixo durante todo o processo de treinamento. Em geral, o treinamento é finalizado quando o mapa atinge um estado de estabilidade, em que não ocorrem variações significativas nos pesos sinápticos após a apresentação de todos os padrões de treinamento.

A Figura 3.6 mostra o algoritmo básico de auto-organização. Os tópicos que se seguem apresentam em detalhes as principais etapas do treinamento dos mapas auto-organizáveis.



**Figura 3.6.** Algoritmo de treinamento básico de um mapa auto-organizável.

### 3.4 Inicialização Sináptica

Consiste na definição dos vetores de pesos sinápticos iniciais. Os pesos sinápticos podem ser inicializados de modo a produzir arranjos planares e uniformes, ou assumir uma configuração dependente da aplicação e de informações previamente conhecidas acerca do problema. A inicialização aleatória dos vetores de pesos não é recomendada, pois promove distorções arbitrárias no arranjo de neurônios, uma vez que não existe uma preocupação voltada para a manutenção da vizinhança topológica, ou seja, neurônios vizinhos no arranjo podem ser inicializados com vetores de pesos arbitrariamente distantes no espaço original dos dados. Uma vez definidos os pesos sinápticos iniciais, é iniciada a fase de treinamento da rede neural.

### 3.5 Processo Competitivo

Durante o processo competitivo, os vetores correspondentes aos padrões de entrada são sucessivamente apresentados ao mapa auto-organizável. Todos os neurônios recebem o mesmo padrão de entrada e, em seguida, competem entre si para serem ativados, sendo que apenas um neurônio será o vencedor para cada padrão apresentado. Essa regra de competição é denominada “o vencedor leva tudo” (do inglês *winner-takes-all*).

Existem quatro conceitos fundamentais envolvidos no processo competitivo:

1. Um conjunto de neurônios com mesma função de ativação.
2. Um conjunto de vetores de pesos sinápticos associados aos neurônios. Cada vetor de pesos sinápticos fornece a posição espacial do neurônio no espaço  $\mathfrak{R}^m$  (espaço dos dados originais).
3. Critério de similaridade adotado.
4. Um mecanismo de competição que permita determinar o neurônio vencedor para os padrões de entrada.

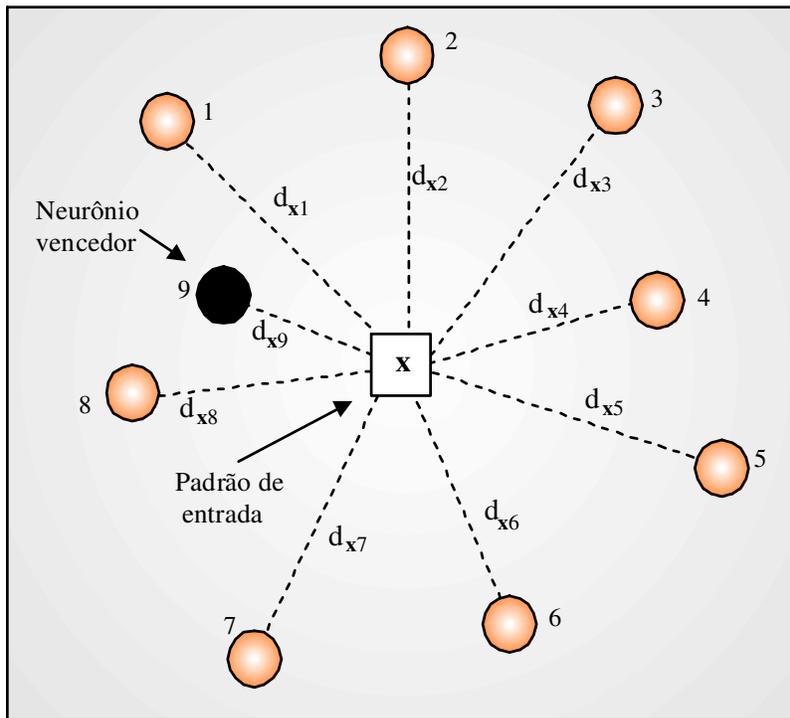
O nível de ativação de cada neurônio é dado por um critério de similaridade entre o vetor correspondente ao padrão de entrada e o vetor de pesos de cada neurônio. Quanto maior for o

grau de similaridade entre um neurônio e um determinado padrão de entrada, maior será o nível de ativação do neurônio. Existem diversos critérios de similaridade que podem ser empregados no processo competitivo (EVERITT *et al.*, 2001; GORDON, 1999). Dentre os critérios comumente adotados na literatura, encontram-se variantes do inverso da distância euclidiana entre o vetor correspondente ao padrão de entrada e o vetor de pesos sinápticos de cada neurônio. O vencedor será o neurônio que apresentar maior similaridade com o padrão de entrada, por exemplo, de acordo com a seguinte fórmula:

$$i(\mathbf{x}) = \arg \max_j \frac{1}{1 + \|\mathbf{x} - \mathbf{w}_j\|}, \quad (3.1)$$

onde  $\mathbf{w}_j$  é o vetor de pesos do neurônio  $j$ ,  $\mathbf{x}$  é o vetor correspondente ao padrão de entrada e  $\|\cdot\|$  é a norma euclidiana.

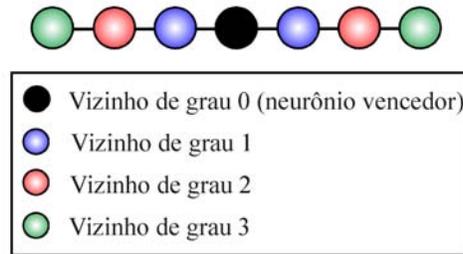
A Figura 3.7 ilustra o mecanismo de competição. Considere um mapa auto-organizável composto por 9 neurônios, e o padrão de entrada  $\mathbf{x}$ . Tomando-se como medida de similaridade o inverso da distância euclidiana, temos que o neurônio vencedor para o padrão  $\mathbf{x}$  é o neurônio  $i(j) = 9$ , representado na figura por um círculo preto, pois a distância euclidiana  $d_{\mathbf{x}j} = \|\mathbf{x} - \mathbf{w}_j\|$  ao padrão de entrada  $\mathbf{x}$  é mínima quando  $j = 9$ . Portanto, o neurônio 9 apresenta maior ativação para o padrão em consideração, produzindo  $i(\mathbf{x}) = 9$ .



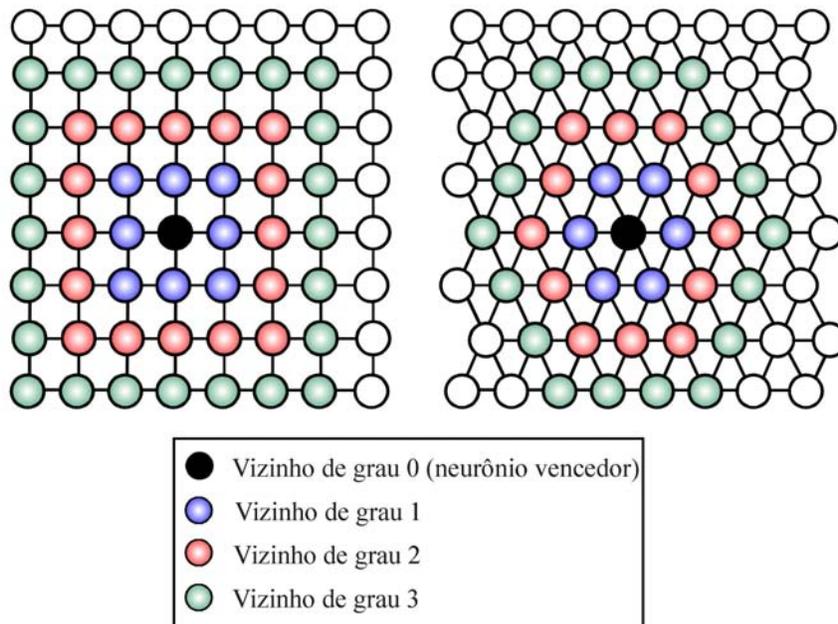
**Figura 3.7.** Processo competitivo: seleção do neurônio vencedor considerando o espaço de entrada como sendo o  $\mathcal{R}^2$  e o critério de similaridade dado por  $i(x)$ .

### 3.6 Processo Cooperativo

O processo cooperativo é inspirado em um mecanismo neurobiológico no qual um neurônio que se encontra no estado “ativo” tende a influenciar o estado dos neurônios vizinhos, de modo que vizinhos situados topologicamente mais próximos sofrem maior influência. Os tipos de vizinhança a serem considerados aqui envolvem uma ou duas dimensões. A Figura 3.8 ilustra quatro graus de relação de vizinhança unidimensional: grau zero, que envolve o próprio neurônio vencedor; grau 1, que envolve o vencedor e os 2 neurônios mais próximos, e graus 2 e 3. A Figura 3.9 mostra quatro níveis de vizinhança considerando dois tipos de relação de vizinhança bidimensional: retangular e hexagonal.



**Figura 3.8.** Processo cooperativo: relação de vizinhança unidimensional.



**Figura 3.9.** Processo cooperativo: duas possíveis relações de vizinhança bidimensional.

A analogia biológica de vizinhança é comumente representada por uma função gaussiana dada por:

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{D_{j,i(\mathbf{x})}^2}{2\sigma^2}\right), \tag{3.2}$$

onde  $i(\mathbf{x})$  denota o neurônio vencedor para o vetor de entradas  $\mathbf{x}$ ;  $D_{j,i(\mathbf{x})}$  é a distância no mapa entre o neurônio vencedor  $i(\mathbf{x})$  e cada neurônio  $j$  de sua vizinhança, ou seja, o grau de vizinhança; e  $\sigma$  é a largura da gaussiana, empregada no controle da amplitude da vizinhança. Vale ressaltar que  $D_{j,i(\mathbf{x})}$  representa o grau de vizinhança, ou seja, a distância dos neurônios no mapa, e não a distância euclidiana entre seus vetores de pesos sinápticos. Em geral, a relação de vizinhança mostra-se particularmente importante durante os estágios iniciais do processo de treinamento. Dessa forma, a largura é controlada de modo que a amplitude da vizinhança seja reduzida com o tempo, de acordo com a seguinte equação:

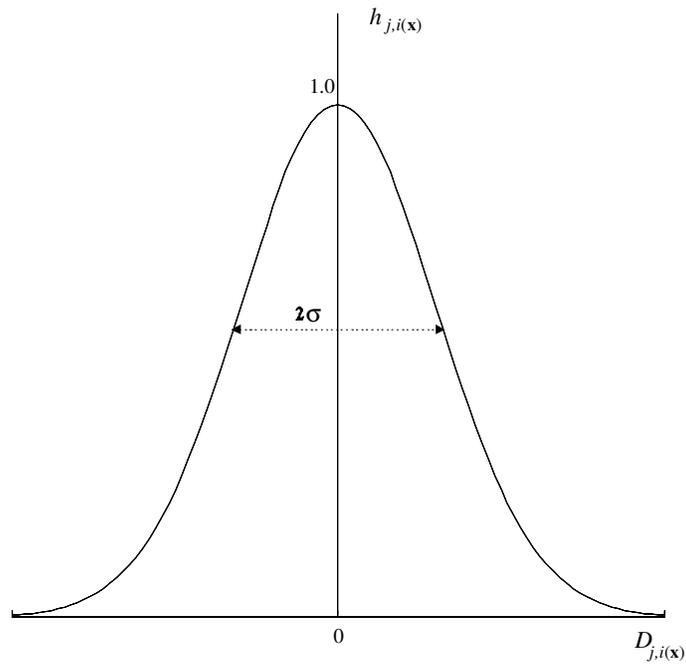
$$\sigma(n) = \sigma_0 \exp\left(\frac{-n}{\tau_1}\right), \quad (3.3)$$

onde  $\tau_1$  é uma constante de tempo e  $n$  é a época, em que cada época corresponde à apresentação de todos os padrões de treinamento. O valor  $\sigma_0$  é definido pelo usuário e é dependente da abrangência da vizinhança requerida para a aplicação.

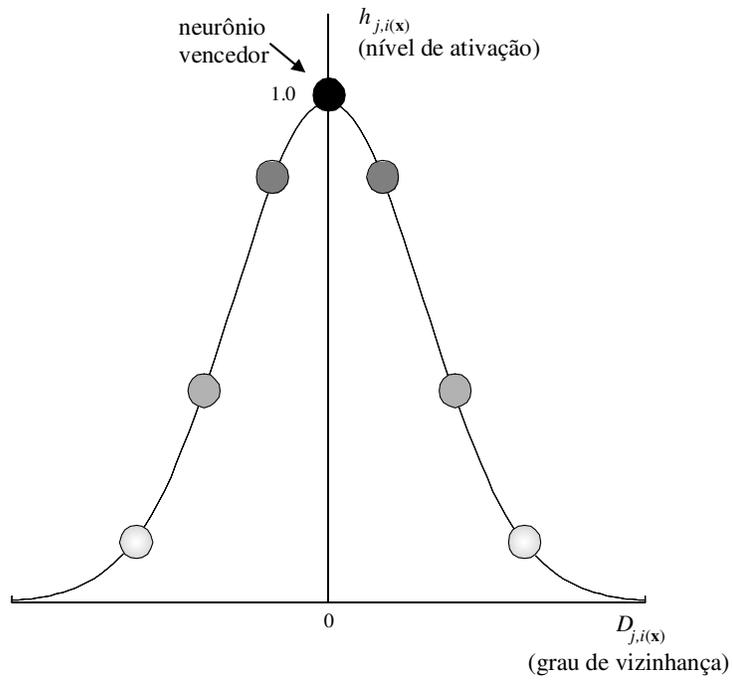
Um valor comumente utilizado para  $\tau_1$  é dado pela fórmula (HAYKIN, 1999):

$$\tau_1 = \frac{1000}{\log \sigma_0}. \quad (3.4)$$

A Figura 3.10 mostra a função gaussiana utilizada na representação da vizinhança. A Figura 3.11 mostra como cada neurônio vizinho é afetado pelo neurônio vencedor, de acordo com a função de vizinhança gaussiana. Note que o neurônio vencedor apresenta o maior valor para  $h_{j,i(\mathbf{x})}$ , seguido pelos neurônios vizinhos de grau 1, grau 2 e grau 3, respectivamente.



**Figura 3.10.** Processo cooperativo: função de vizinhança gaussiana.



**Figura 3.11.** Nível de ativação de cada neurônio de acordo com o grau de vizinhança.

### 3.7 Processo Adaptativo

Durante o processo adaptativo, os vetores de pesos sinápticos do neurônio vencedor e de sua vizinhança são atualizados de forma a se aproximarem do padrão de entrada apresentado, respeitando a relação de vizinhança definida durante o processo cooperativo. A atualização do vetor de pesos sinápticos  $\mathbf{w}_j$  é dada por:

$$\mathbf{w}_{j+1} = \mathbf{w}_j + \Delta\mathbf{w}_j. \quad (3.5)$$

A variação  $\Delta\mathbf{w}_j$  é calculada de acordo com a fórmula:

$$\Delta\mathbf{w}_j = \eta(n) \cdot h_{j,i(\mathbf{x})} \cdot (\mathbf{x} - \mathbf{w}_j), \quad (3.6)$$

onde  $h_{j,i(\mathbf{x})}$  é a função de ativação de vizinhança definida na Seção 3.6;  $\mathbf{x}$  é o vetor associado ao padrão de entrada;  $\eta(n)$  é a taxa de aprendizagem na época  $n$ ; e  $\mathbf{w}_j$  é o vetor de pesos sinápticos associado ao neurônio  $j$ . A atualização da taxa de aprendizagem é dada pela seguinte equação:

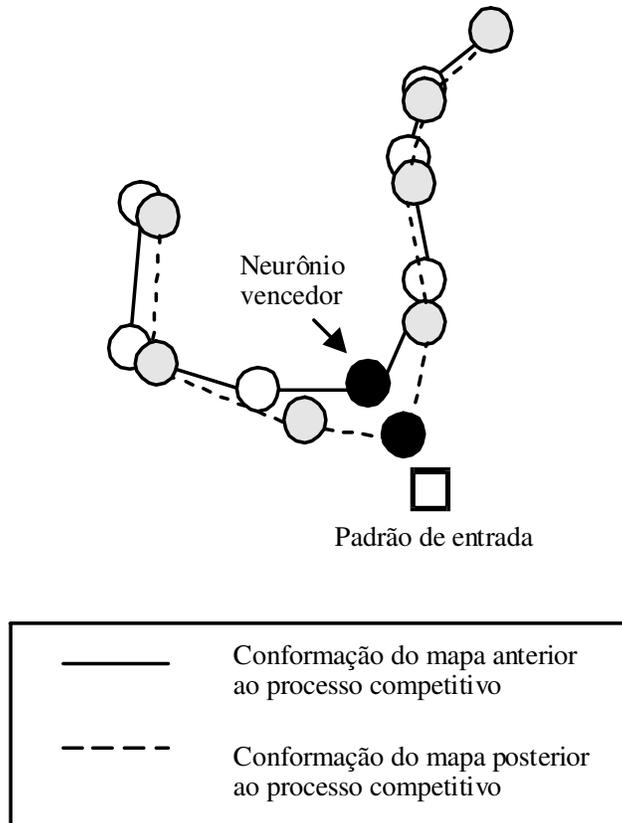
$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \quad (3.7)$$

onde  $\tau_2$  é uma constante de tempo e  $\eta_0$  é a taxa de aprendizagem inicial.

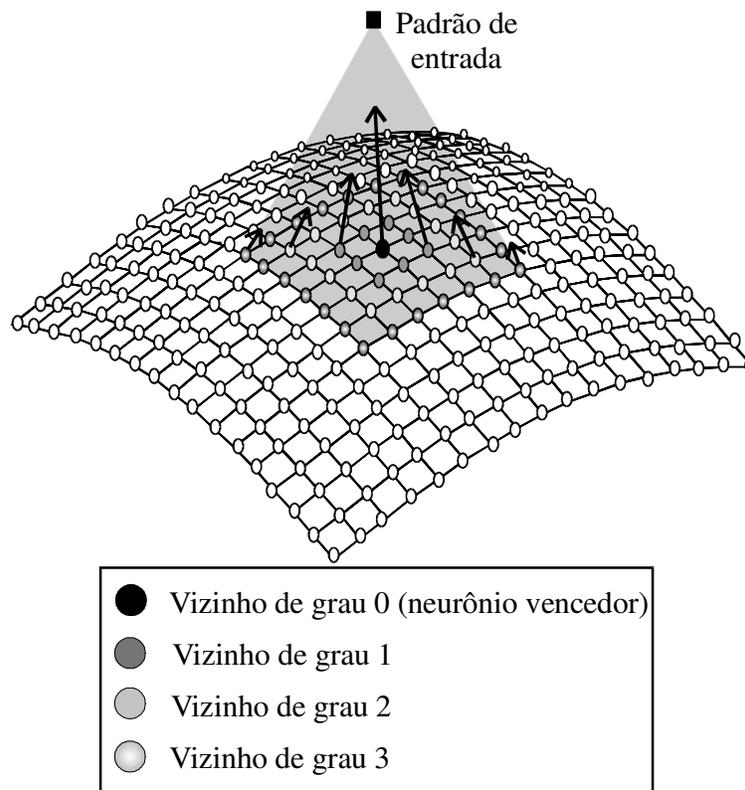
Em geral, a taxa de aprendizado inicial  $\eta_0$  assume um valor próximo de 0,1 e deve decrescer gradualmente até um valor próximo de 0,01. Para que a taxa de aprendizagem assumia valores dentro desse intervalo, a constante de tempo  $\tau_2$  deve assumir um valor próximo de 1000 (HAYKIN, 1999).

A Figura 3.12 ilustra o processo de adaptação sináptica para uma rede neural com relação de vizinhança unidimensional e vizinhança de grau 4. A Figura 3.13 mostra o processo adaptativo para um mapa auto-organizável com relação de vizinhança bidimensional também de grau 4. Observe que, para ambos os casos, o neurônio vencedor sofre um maior deslocamento,

seguido pelos vizinhos de grau 1, grau 2 e grau 3, respectivamente, de acordo com a função gaussiana expressa na Figura 3.10. Na Figura 3.12 são mostrados os mapas antes e após o processo adaptativo, enquanto na Figura 3.13 são apresentados o mapa antes do processo adaptativo e os vetores de variação de peso sináptico correspondentes ao neurônio vencedor e alguns neurônios que compõem a vizinhança de grau 4.



**Figura 3.12.** Processo adaptativo para mapa auto-organizável unidimensional com vizinhança de grau 4.



**Figura 3.13.** Processo adaptativo dos vetores de pesos após a definição do neurônio vencedor para mapa auto-organizável bidimensional com vizinhança de grau 4.

### 3.8 Algoritmo de Treinamento

Existem duas formas de treinamento de um mapa auto-organizável: treinamento seqüencial e treinamento em lote ou batelada. Os principais passos de cada uma das formas de treinamento são mostrados nos pseudo-códigos a seguir.

### 3.8.1 Algoritmo de Treinamento em Modo Seqüencial

#### Procedimento TreinamentoSeqüencial

1. Definir dimensão e topologia do arranjo de neurônios e o número de neurônios do mapa
2. Efetuar inicialização sináptica
3. Definir largura  $\sigma_0$  da gaussiana
4. Inicializar as variáveis definidas pelo usuário:  $\eta_0=0,1; \tau_2=1000; n=0; \tau_1=\frac{1000}{\log \sigma_0}$
5. Enquanto não atingir condição de parada faça:
  - 5.1.  $n=n+1$
  - 5.2. Reordenar aleatoriamente os padrões de entrada;
  - 5.3. Para cada padrão de entrada do conjunto de padrões reordenado faça:

$$\eta(n) = \eta_0 \cdot \exp\left(-\frac{n}{\tau_2}\right); \quad \sigma(n) = \sigma_0 \cdot \exp\left(-\frac{n}{\tau_1}\right)$$

5.3.1. Processo competitivo: calcular o neurônio vencedor para o padrão de entrada

selecionado de acordo com:  $i(\mathbf{x}) = \arg \max_j \frac{1}{1 + \|\mathbf{x} - \mathbf{w}_j\|}$

5.3.2. Processo cooperativo: definir vizinhança do neurônio vencedor  $i(\mathbf{x})$  (processo cooperativo), onde  $i(\mathbf{x}) \in \{1, 2, \dots, N\}$ , sendo  $N$  o número de neurônios do mapa.

5.3.3. Processo adaptativo: atualizar pesos sinápticos do neurônio vencedor e de sua vizinhança de acordo com:

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{D_{j,i(\mathbf{x})}^2}{2\sigma(n)^2}\right); \quad \Delta \mathbf{w}_j = \eta(n) \cdot h_{j,i(\mathbf{x})} (\mathbf{x} - \mathbf{w}_j); \quad \mathbf{w}_{j+1} = \mathbf{w}_j + \Delta \mathbf{w}_j$$

5.4. Fim Para

6. Fim Enquanto

**Fim Procedimento**

**Algoritmo 3.1.** Treinamento de rede auto-organizável em modo seqüencial.

### 3.8.2 Algoritmo de Treinamento em Lote ou Batelada

#### Procedimento TreinamentoEmLote

1. Definir dimensão e topologia do arranjo de neurônios e número de neurônios do mapa
2. Inicializar os vetores de pesos sinápticos dos neurônios do mapa auto-organizável (inicialização sináptica)
3. Definir largura  $\sigma_0$  da gaussiana

4. Inicializar as variáveis:  $\eta_0 = 0,1$ ;  $\tau_2 = 1000$ ;  $n = 0$ ;  $\tau_1 = \frac{1000}{\log \sigma_0}$ ;

5. Enquanto não atingir condição de parada faça:

5.1.  $n = n + 1$

5.2. Para cada padrão de entrada  $x$  do conjunto de padrões faça:

$$5.2.1. \eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right); \sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right)$$

5.2.2. Processo competitivo: calcular o neurônio vencedor para o padrão de entrada selecionado de acordo com:  $i(\mathbf{x}) = \arg \max_j \frac{1}{1 + \|\mathbf{x} - \mathbf{w}_j\|}$

5.2.3. Processo cooperativo: definir vizinhança do neurônio vencedor  $i(\mathbf{x})$ , onde  $i(\mathbf{x}) \in \{1, 2, \dots, N\}$ , sendo  $N$  o número de neurônios do mapa

5.2.4. Calcular e armazenar deslocamento de cada neurônio de acordo com:

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{D_{j,i(\mathbf{x})}^2}{2\sigma(n)^2}\right); \Delta \mathbf{w}_j = \eta(n) \cdot h_{j,i(\mathbf{x})} (\mathbf{x} - \mathbf{w}_j)$$

5.3. Fim Para

5.4. Para cada neurônio faça:

5.4.1. Calcular deslocamento resultante de cada neurônio considerando todo o conjunto de padrões, ou seja, todos os deslocamentos armazenados

5.4.2. Processo adaptativo: atualizar pesos sinápticos dos neurônios vencedores e suas vizinhanças com base no deslocamento resultante do passo anterior

5.5. Fim Para

6. Fim Enquanto

**Fim Procedimento**

**Algoritmo 3.2.** Treinamento de rede auto-organizável em lote ou batelada.

### 3.9 Outros Modelos de Redes Neurais Auto-Organizáveis

A literatura apresenta uma ampla variedade de redes neurais baseadas em aprendizado competitivo/não-supervisionado. As diferenças entre as variadas abordagens residem em muitos aspectos, tais como critério para cálculo do neurônio vencedor, otimizações que aceleram o aprendizado e o processo de convergência do mapa (controle do número de neurônios presentes no mapa, procedimentos de poda e inserção de neurônios na rede, inibição de neurônios etc.), e algumas variações conceituais mais profundas. Existem diversos modelos de redes neurais auto-organizáveis, tais como mapas derivados da aplicação da teoria de ressonância adaptativa (*Adaptive Resonance Theory – ART maps*) (CARPENTER & GROSSBERG, 1991), *Neocognitron* (FUKUSHIMA, 1980), *Redes Neurais Competitivas Hierárquicas (RNCH)* (LI *et al.*, 1993), *Incremental Grid Growing* (BLACKMORE & MIIKKULAINEN, 1993), entre outras. Uma revisão detalhada acerca de modelos de redes neurais competitivas pode ser encontrada em COSTA (1999).

#### 3.9.1 *Adaptive Resonance Theory (ART)*

Os modelos da família de redes tipo ART (*Adaptive Resonance Theory*) foram propostos por Grossberg e Carpenter (CARPENTER & GROSSBERG, 1991). Um dos principais objetivos das redes ART é permitir a adaptabilidade da rede frente a ambientes dinâmicos, isto é, em que os vetores de entrada sofrem variações ao longo do tempo, sendo capaz de aprender novos padrões sem afetar o aprendizado já efetuado. Considere, por exemplo, um problema em que um conjunto de vetores deve ser classificado em um espaço de entrada. O problema pode ser resolvido treinando-se uma rede neural tipo perceptron multicamadas com  $N$  vetores. Ao final de um treinamento bem sucedido, os  $N$  vetores serão corretamente classificados e espera-se que outros vetores pertencentes ao mesmo domínio também o sejam devido à capacidade de generalização da rede. Entretanto, caso o ambiente de entrada se modifique com o tempo, a acuidade da classificação tenderá a reduzir devido ao fato de os pesos sinápticos serem fixos. Este problema poderia ser contornado realizando um novo treinamento da rede, envolvendo todos os padrões de entrada, a cada vez que um novo padrão de entrada é apresentado à rede. Entretanto, essa abordagem poderia ser extremamente ineficiente e conduzir a dados de

treinamento conflitantes. As redes ART foram propostas com o intuito de contornar dificuldades inerentes a ambientes dinâmicos.

A arquitetura de uma rede ART é composta por duas camadas com diferentes funções de ativação: F1 e F2. A camada F1 é responsável pela recepção dos padrões de entrada. Um padrão de entrada é recebido pela camada F1 e, em seguida, é propagado para a camada F2 através de conexões sinápticas. Na camada F2 é definido o neurônio vencedor para o padrão de entrada. Caso o neurônio vencedor seja suficientemente similar ao padrão de entrada, o neurônio vencedor será representante da classe do padrão. Caso não seja suficientemente similar, um novo neurônio será criado em F2 (portanto uma nova classe) para representar esse padrão. A aceitação ou não do neurônio vencedor como representante da classe do padrão de entrada é definido por um “parâmetro de vigilância”. Se esse parâmetro assumir valores elevados, a tolerância a dissimilaridades é alta. Caso o valor seja baixo, a tolerância é baixa. Nesse caso, quanto menor o valor da vigilância, maior será a quantidade de classes.

Uma vez definido o neurônio vencedor, ocorre o procedimento de adaptação sináptica, em que o neurônio vencedor se aproxima do padrão de entrada apresentado. Diferentemente do SOM, apenas o neurônio de maior ativação é associado ao padrão apresentado. Assim, uma das características fundamentais da teoria da ressonância adaptativa em que se baseiam as redes ART é que, se nenhuma classe ressoa com o padrão apresentado, uma nova classe é criada. Isso confere às redes ART um alto grau de plasticidade, no sentido de não afetar padrões previamente aprendidos.

No modelo ART os pesos das conexões sinápticas entre a primeira e a segunda camada podem ser dinamicamente modificados de modo a se adaptarem a novos vetores de entrada. O termo ressonância refere-se ao estado de aprendizado da rede, que emprega ciclos de realimentação entre as camadas de saída e entrada. O aprendizado é realizado somente durante esta fase de ressonância.

Vale destacar que um aspecto importante é a definição de um valor apropriado para o parâmetro de vigilância, o qual pode ser inclusive variável durante o treinamento da rede neural. Assim, existem diversas propostas nesse sentido, as quais incluem técnicas de

aprendizado supervisionado, aprendizado por reforço e inferência nebulosa na determinação do valor do parâmetro de vigilância ao longo do processo de auto-organização do mapa (ZIKIDIS & TZAFESTAS, 2003).

### **3.9.2 Modelos Construtivos**

Um dos maiores problemas presentes em muitas metodologias de classificação automática é a necessidade de definição prévia do número de agrupamentos presentes na estrutura dos dados. Independentemente de a definição do número de agrupamentos ser adequada ou não, o algoritmo forçará os dados a assumirem aquela estrutura. Assim, o número de classes constitui uma importante informação de entrada para o algoritmo, sendo que uma definição apropriada é crucial para que sejam produzidos resultados satisfatórios. Para contornar este problema, foram propostos os modelos construtivos ou dinâmicos, os quais permitem a variação do número de neurônios ao longo do processo de treinamento. Existem várias propostas de modelos construtivos na literatura (ARAS *et al.*, 1999; BRUSKE & SOMMER, 2001; FRITZKE, 1999; MARSLAND *et al.*, 2002; DECKER, 2006).

Um problema inerente aos mapas auto-organizáveis que tende a prejudicar o processo de organização é a presença de neurônios considerados “inativos”, correspondendo àqueles que vencem nenhuma ou poucas vezes o processo competitivo. Ao final do processo de organização do mapa, restam neurônios que apresentam uma pequena região de influência, ou seja, encontram-se associados a poucos padrões ou mesmo a nenhum padrão. À medida que aumenta o número de neurônios no mapa, tende a aumentar a quantidade de neurônios inativos. Um artifício comumente utilizado é a eliminação de neurônios inativos através da definição de um limiar correspondendo ao grau mínimo de ativação que deve ser atingido por cada neurônio. Todos os neurônios que não atingirem esse limiar são considerados inativos e eliminados do mapa.

ARAS *et al.* (1999) propõem uma estratégia de inibição de neurônios para detectar a necessidade de inclusão de novos nós no mapa. Quando um neurônio vence o processo competitivo por um determinado número de vezes, ele é desativado e impedido de vencer a

competição por um certo número de épocas. Caso um neurônio inibido seja o vencedor, um novo neurônio é incluído como vizinho imediato do vencedor.

### **3.9.3 Redes Neurais Competitivas Hierárquicas**

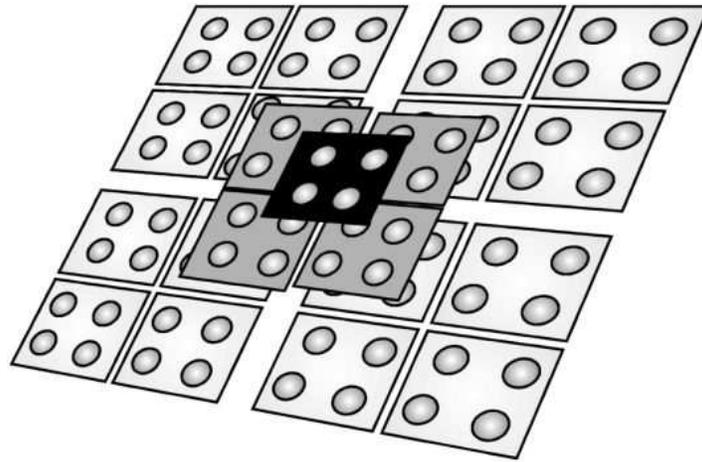
A idéia básica dos mapas auto-organizáveis hierárquicos é o emprego de uma estrutura hierárquica organizada em múltiplas camadas, sendo cada camada constituída por um número de mapas auto-organizáveis independentes. Um mapa é definido como sendo o elemento raiz ou primeira camada da hierarquia. Para cada neurônio deste mapa, é definido um novo mapa na camada imediatamente inferior da hierarquia. O treinamento geralmente é feito de maneira top-down, ou seja, os pesos de uma camada são inicializados considerando os pesos da camada superior. Estas redes neurais podem estar organizadas de acordo com variadas arquiteturas, como piramidal, em árvore etc. Para uma revisão acerca dos principais modelos hierárquicos, consultar COSTA (1999) e VICENTE & VELLIDO (2004).

A Figura 3.14 ilustra uma possível estrutura hierárquica composta por três camadas. A primeira camada possui quatro unidades. Cada neurônio do mapa do primeiro nível corresponde a um mapa distinto no segundo nível, e assim sucessivamente, até se chegar ao nível mais baixo desejado.

O treinamento tem início com o mapa auto-organizável da primeira camada. Quando a configuração do mapa se torna estável, tem início o treinamento da próxima camada.

As redes hierárquicas podem apresentar arquitetura fixa ou flexível e podem ser estáticas ou construtivas, isto é, envolver número fixo de neurônios ou permitir inserção ou eliminação de neurônios ao longo do processo de organização do mapa.

Como principais vantagens das redes neurais hierárquicas, pode-se apontar o fato de serem capazes de prover maior velocidade de treinamento e tornar a escolha do neurônio vencedor mais eficiente. Além disso, existem aplicações, tais como análise de imagens através de técnicas de multirresolução e clusterização hierárquica, às quais a configuração hierárquica se mostra mais apropriada que as configurações em um mesmo nível.



**Figura 3.14.** Mapa auto-organizável hierárquico com três camadas.

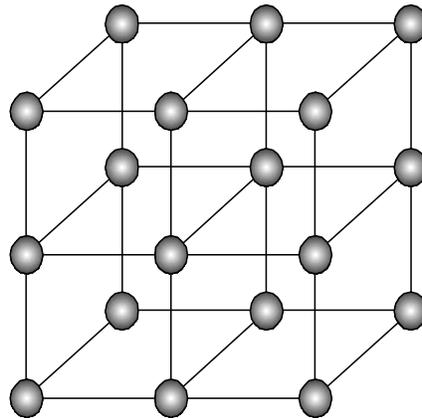
### 3.9.4 Neocognitron

O neocognitron (FUKUSHIMA, 1980) é uma rede neural artificial hierárquica organizada em múltiplas camadas. Cada estágio da hierarquia possui a função de reconhecer características em níveis de complexidade distintos. É bastante utilizada na área de visão computacional, como, por exemplo, para reconhecimento de caracteres e dígitos manuscritos. O mecanismo básico de aprendizado não-supervisionado utilizado neste tipo de rede é a regra de Hebb (HAYKIN, 1999). Existem diversos modelos de neocognitron. O modelo mais simples define três tipos de neurônio, denominados *S*, *C* e *V*. A rede é formada por estágios, sendo que cada estágio é composto por uma camada de células *S* seguida por uma camada de células *C*. As camadas finais efetuam um reconhecimento de ordem local, restringindo-se a peculiaridades do padrão, como orientação de um segmento de reta. As camadas iniciais efetuam um reconhecimento mais generalizado, extraíndo atributos de natureza global. Entre as camadas existem pesos sinápticos excitatórios e inibitórios. As células do tipo *S* são responsáveis pelo processo de extração de características, enquanto células do tipo *C* destinam-se a compensar erros posicionais dos atributos, de modo a permitir a invariância com relação à translação dos padrões. A última camada é do tipo *C*, e é denominada *camada de reconhecimento* em virtude

de representar o resultado de uma determinada classe de padrões. Cada camada é composta por subgrupos de neurônios, denominados planos, sendo que os neurônios são agrupados em planos de acordo com os atributos que eles representam. No caso do aprendizado não-supervisionado, o algoritmo deve definir quais atributos correspondem a cada plano. Cada neurônio de um plano recebe conexões de neurônios situados em uma mesma localização espacial da camada imediatamente anterior.

### **3.9.5 Mapas com Dimensão Elevada**

O mapa auto-organizável efetua mapeamento de um espaço  $p$ -dimensional para um espaço uni ou bidimensional, denotado pela relação de vizinhança entre neurônios. Os neurônios tendem a se agregar nas regiões de alta concentração de dados. É desejável que dados topologicamente próximos no espaço de entrada sejam representados por neurônios próximos no espaço de saída. Entretanto, devido à pequena dimensão dada pela relação de vizinhança entre neurônios, muitas vezes a informação topológica presente nos dados de entrada é perdida. Assim, dados próximos em um espaço  $p$ -dimensional não necessariamente estarão próximos em um espaço bidimensional, dependendo da forma como a operação de projeção foi realizada. Índices para mensurar o grau de violação topológica podem ser encontrados em BAUER & PAWELZIK (1992). Com o objetivo de atenuar o efeito da violação topológica, COSTA (1999) propõe um modelo de mapa auto-organizável com relação de vizinhança maior que 2. Arranjos de neurônios em mapas auto-organizáveis com relação de vizinhança maior que 2 raramente são utilizados na prática, pois geralmente o objetivo principal do mapa auto-organizável é a visualização de aspectos relevantes da distribuição dos dados. A Figura 3.15 mostra uma configuração de neurônios com vizinhança tridimensional.



**Figura 3.15.** Exemplo de relação de vizinhança tridimensional.

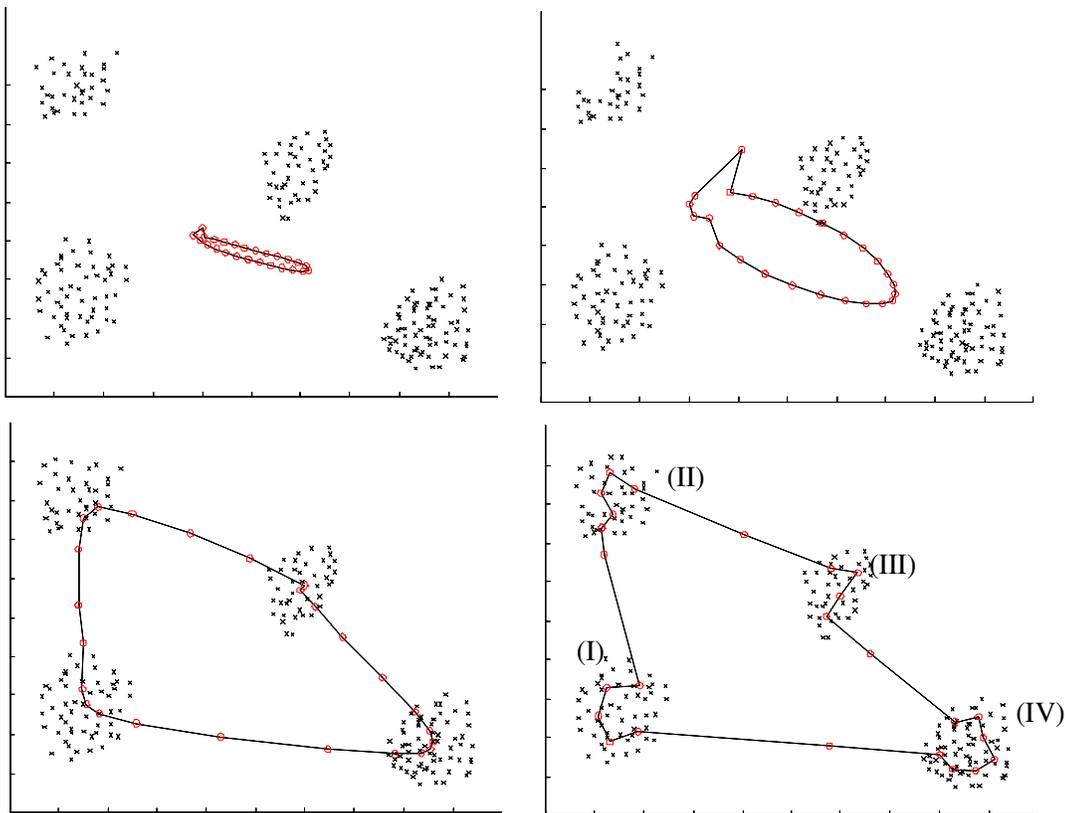
### 3.10 Algumas Áreas de Aplicação

Os mapas auto-organizáveis vêm sendo amplamente aplicados a análise multivariada e problemas de classificação de dados de elevada dimensão, e procedem convertendo relações estatísticas complexas e não-lineares entre os dados para um mapeamento de dimensão reduzida. Embora a maior parte das aplicações de mapas auto-organizáveis tenha se concentrado em análise de dados, agrupamento e classificação, nos últimos anos esforços consideráveis vêm sendo despendidos na área de otimização combinatória (ANGÉNIOL *et al.*, 1988; ARAS *et al.*, 1999; GOMES & VON ZUBEN, 2001; 2002a; 2002b; 2002c; 2002d; MATSUYAMA, 1991; MATSUYAMA, 1996; MODARES *et al.*, 1999).

O Capítulo 4 aborda a solução de problemas combinatórios via mapas auto-organizáveis, apresentando enfoque para os problemas do caixeiro viajante e roteamento de veículos. Para o caso do problema de roteamento de veículos, é necessário considerar simultaneamente múltiplos mapas auto-organizáveis, cada um representando uma rota distinta.

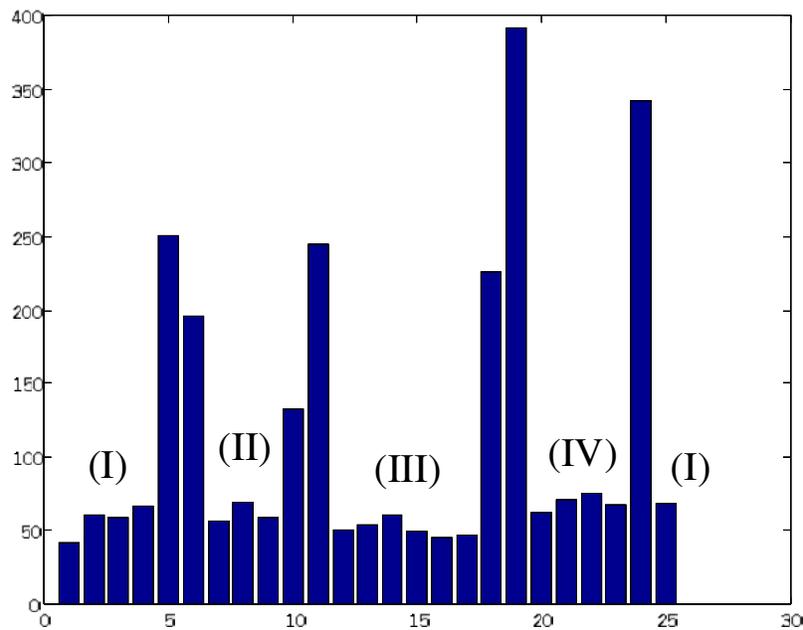
### 3.10.1 Agrupamento (Clusterização) e Classificação

A Figura 3.16 apresenta diversos estágios intermediários do processo de agrupamento de 200 padrões via um mapa auto-organizável unidimensional contendo um número fixo de 25 neurônios. Neste caso, a inicialização sináptica é efetuada de modo que os neurônios se disponham na forma de um anel de raio  $r$  cujo centro coincide com o centro de massa dos padrões a serem agrupados. A vizinhança entre neurônios é circular, ou seja, o primeiro neurônio do mapa é vizinho imediato do último. Os processos competitivo e de adaptação sináptica são idênticos àqueles descritos na Seção 3.3. Os padrões a serem agrupados são representados pelo símbolo 'x' e os vetores de pesos dos neurônios pelo símbolo 'o'. As conexões entre os vetores de pesos dos neurônios indicam a relação de vizinhança.



**Figura 3.16.** Quatro estágios do processo de auto-organização do mapa para um problema de agrupamento.

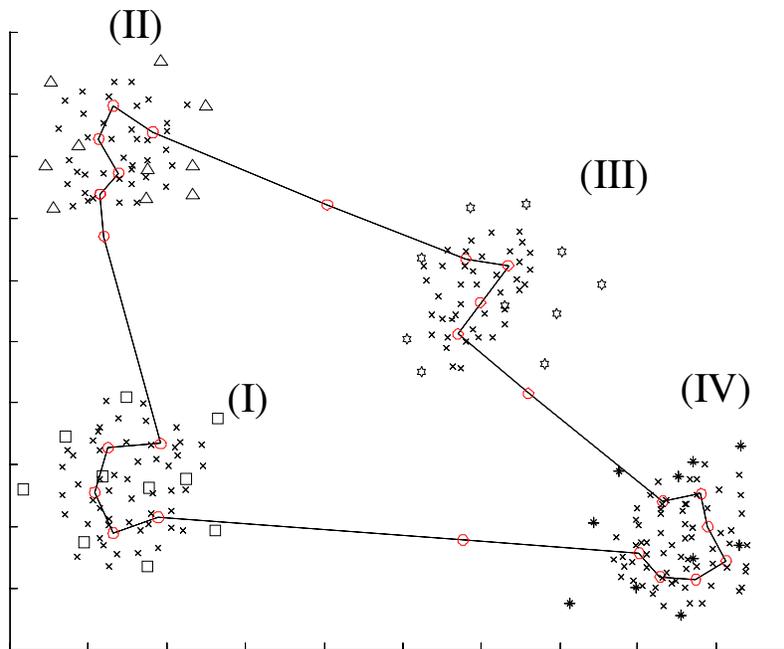
A Figura 3.17 exibe um gráfico de barras indicando as distâncias entre neurônios vizinhos, permitindo a visualização dos grupos (clusters) formados pelo conjunto de padrões. Nesse caso, como a vizinhança é circular, podemos observar claramente a existência de quatro clusters. Note que a distância entre neurônios vizinhos representantes de um mesmo cluster é relativamente pequena quando comparada à distância entre neurônios vizinhos representantes de clusters distintos. Dessa forma, a separação dos clusters através do gráfico de barras é efetuada estabelecendo-se um limiar de distância entre neurônios vizinhos, de modo que uma distância acima do limiar indique um novo cluster. Vale observar que, na Figura 3.16, existem neurônios que não se encontram associados a um padrão específico, apresentando uma distância relativamente grande entre os clusters formados. Esse fator pode também ser notado na Figura 3.17 (neurônios 6, 10 e 18). Os neurônios que não se encontram associados a clusters específicos podem ser eliminados através do uso de um limiar de distância, em que neurônios apresentando distância maior que este limiar para todos os clusters são eliminados. Um outro método seria o uso de um mapa construtivo, em que neurônios que não vencem o processo competitivo são eliminados do mapa durante o treinamento.



**Figura 3.17.** Histograma referente ao vetor de distâncias entre neurônios vizinhos.

Uma outra importante aplicação dos mapas auto-organizáveis são os problemas de *classificação*. Uma vez que os padrões de entrada foram agrupados, novas amostras de mesma natureza podem ser automaticamente classificadas com base no mapa gerado.

Na Figura 3.18 novos padrões, representados por ‘\*’, ‘□’, ‘△’ e ‘☆’, foram introduzidos ao conjunto da Figura 3.16. Cada novo padrão é classificado de acordo com o rótulo definido pelo neurônio mais próximo. Dessa forma, fica evidente que os padrões representados por ‘□’ pertencem à classe (I), os padrões representados por ‘△’ pertencem à classe (II), aqueles representados por ‘☆’ pertencem à classe (III), e aqueles representados por ‘\*’ à classe (IV).



**Figura 3.18.** Processo de classificação via mapas auto-organizáveis.

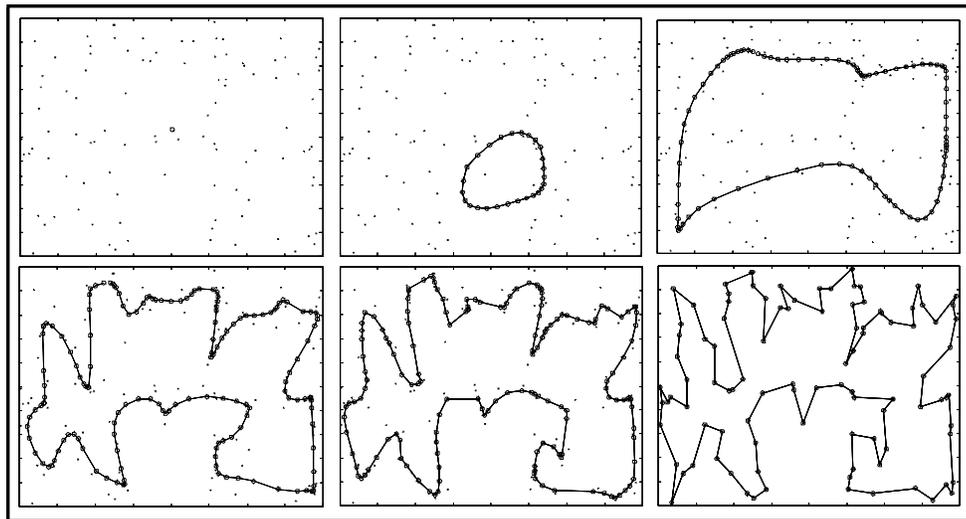
### **3.10.2 O Problema do Caixeiro Viajante e o Problema de Roteamento de Veículos**

Uma outra área de aplicação de mapas auto-organizáveis é a otimização combinatória. Existem na literatura algumas propostas baseadas em mapas auto-organizáveis para solução do problema do caixeiro viajante (ARAS *et al.*, 1999; ANGÉNIOL *et al.*, 1988) e do problema de roteamento de veículos (MODARES *et al.*, 1999).

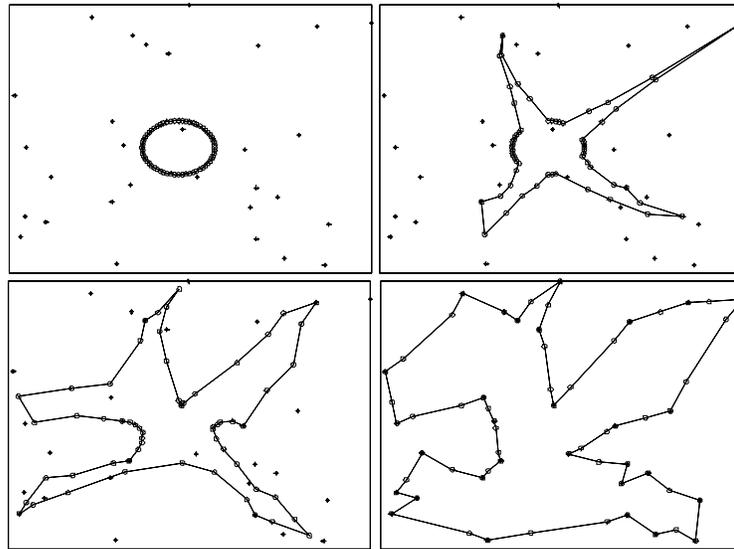
No caso do problema do caixeiro viajante, o percurso é diretamente associado à seqüência de neurônios definida pela relação de vizinhança. Os vetores de pesos sinápticos correspondentes aos neurônios apresentam dimensão 2, ou seja, podem ser vistos como coordenadas em um plano e são associados às coordenadas dos nós (cidades ou clientes) a serem visitados pelo caixeiro viajante. Considerando que o caixeiro viajante deve retornar à cidade de origem, a vizinhança do mapa deve ser circular. A Figura 3.19 ilustra seis estágios intermediários do processo de auto-organização para o problema do caixeiro viajante simétrico, a ser detalhado no Capítulo 4. O problema envolve 100 cidades, representadas por asteriscos, um conjunto de neurônios representados por circunferências e as relações de vizinhança entre neurônios, denotadas por linhas cheias. Neste exemplo, o número de neurônios é variável ao longo do treinamento e ao final do processo de organização do mapa teremos cada um dos neurônios associado a uma única cidade. A relação de vizinhança representa o percurso obtido através do mapa auto-organizável. Este tipo de aplicação emprega um modelo de rede construtiva, descrito na Seção 3.9.2 e não requer modificações consideráveis no algoritmo de treinamento. O ajuste automático de neurônios envolve uma estratégia de poda e inclusão de acordo com o número de vezes que cada neurônio vence o processo competitivo no transcorrer das épocas. Uma possível estratégia, proposta em (ARAS *et al.*, 1999), consiste em inserir novos neurônios próximos a neurônios que vencem o processo competitivo com muita frequência e eliminar neurônios que não venceram o processo competitivo ao longo de  $n$  épocas consecutivas, por exemplo. As Figuras 3.20 e 3.21 ilustram a evolução do processo de auto-organização do mapa para uma instância com 30 cidades para um mapa contendo número fixo de neurônios e para um mapa contendo número variável de neurônios, respectivamente. O mapa da Figura 3.20 foi inicializado com um número de neurônios igual ao dobro do número de cidades,

conforme sugerido em ANGÉNIOL *et al.* (1988) para o caso de arquiteturas fixas. Os neurônios foram dispostos ao redor do centro de área formado pelas coordenadas das cidades. O mapa da Figura 3.21 foi inicializado com um único neurônio posicionado sobre o centro de área formado pelas coordenadas das cidades. Observe que a distância total obtida pela arquitetura fixa foi significativamente maior (55509) que a distância obtida para o caso da arquitetura construtiva (50131). Observe também que em todos os casos a abordagem foi capaz de produzir soluções sem a presença de cruzamentos na rota.

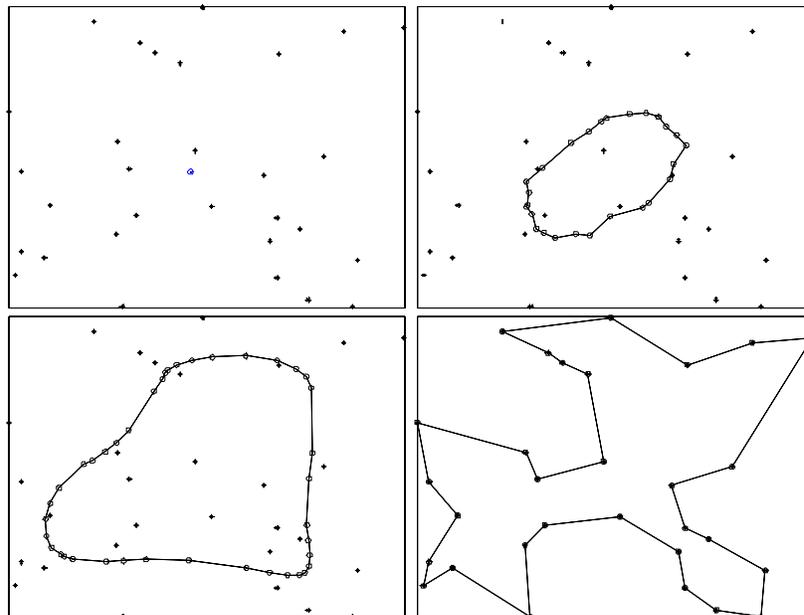
A Figura 3.22 mostra seis estágios do processo de auto-organização para uma instância do problema de roteamento de veículos capacitados. Neste caso, é empregado um modelo que envolve múltiplos mapas auto-organizáveis, sendo necessárias modificações significativas no algoritmo de treinamento convencional apresentado na Seção 3.3. O gerenciamento de múltiplos mapas constitui uma das principais contribuições desta tese. Cada veículo estará associado a um mapa. O número de veículos deve ser previamente definido. Neste caso a definição do neurônio vencedor não é trivial, uma vez que existem restrições relacionadas a carga de veículos a serem satisfeitas. Mais detalhes acerca da técnica de solução para este tipo de aplicação são fornecidos no Capítulo 4.



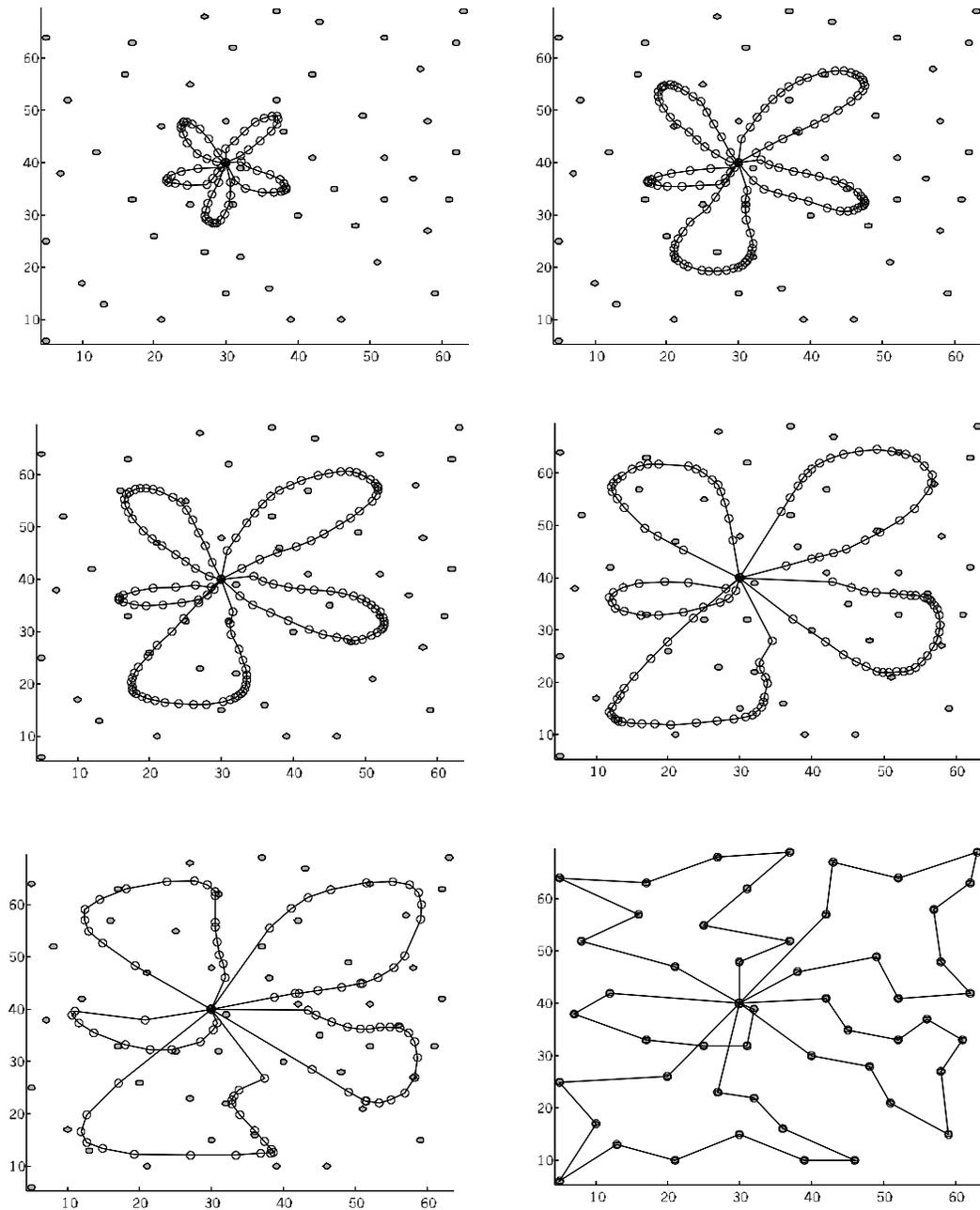
**Figura 3.19.** Processo de evolução do mapa auto-organizável para o problema do caixeiro viajante simétrico.



**Figura 3.20.** Processo de evolução do mapa auto-organizável para o problema do caixeiro viajante simétrico com 30 cidades e número fixo de neurônios. Distância entre cidades na configuração final: 55509.



**Figura 3.21.** Processo de evolução do mapa auto-organizável para o problema do caixeiro viajante simétrico com 30 cidades e estratégia de inserção e poda de neurônios. Distância entre cidades na configuração final: 50131.



**Figura 3.22.** Processo de evolução do mapa auto-organizável para o problema de roteamento de veículos.

### 3.10.3 Regressão para Identificação de Sistemas Dinâmicos

A identificação de sistemas dinâmicos trata da construção de modelos matemáticos para sistemas não-lineares com base em dados resultantes de observações ou medidas no tempo. Em geral, escolhe-se um modelo cujos parâmetros são desconhecidos. Os parâmetros devem ser calculados de forma que o erro entre os dados estimados pelo modelo seja o menor possível em relação aos dados reais, capturando adequadamente a dinâmica do sistema. Este tipo de problema corresponde a um problema específico de aproximação de funções em que uma função relacionada à superfície de erro deve ser minimizada. Redes neurais artificiais, em particular redes do tipo perceptron multicamadas e redes com função de base radial, vêm sendo aplicadas com sucesso a problemas dessa classe. BARRETO & ARAÚJO (2004), por sua vez, abordaram o problema de forma inédita através de mapas auto-organizáveis. Mapas auto-organizáveis já haviam sido aplicados a problemas envolvendo mapeamentos estáticos, em que a saída do sistema depende somente de sua entrada corrente. Com o objetivo de aplicar mapas auto-organizáveis a problemas de regressão, o vetor de entrada é aumentado de modo a compreender a entrada para a função e a saída desejada para esta entrada.

No caso do tratamento de sistemas dinâmicos, a saída atual depende de um conjunto de saídas e entradas anteriores, conforme descrito pela Equação 3.8.

Considere o problema de predição da próxima saída  $y(t + 1)$  a partir do conhecimento de  $n_y$  valores passados da saída e  $n_u$  valores passados da entrada de um sistema dinâmico. A questão é encontrar  $f(\cdot)$  tal que:

$$y(t+1) = f[y(t), \dots, y(t - n_y + 1); u(t), \dots, u(t - n_u + 1)] \quad (3.8)$$

onde  $n_y$  e  $n_u$  são as ordens de memória do modelo dinâmico.

De forma similar ao tratamento de problemas de regressão, o vetor de entrada  $\mathbf{x}(t)$  para o mapa auto-organizável é composto por duas partes: a primeira comporta dados referentes à entrada do mapeamento dinâmico a ser aprendido; a segunda contém dados referentes à saída

desejada para o mapeamento. O neurônio vencedor é determinado com base somente na primeira parte componente do vetor de entrada.

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^e(t) \\ \mathbf{x}^s(t) \end{pmatrix} \text{ e } \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^e(t) \\ \mathbf{w}_i^s(t) \end{pmatrix}, \quad (3.9)$$

onde  $\mathbf{x}^e$  é a entrada do mapeamento,  $\mathbf{x}^s$  constitui a saída desejada e  $\mathbf{w}^e$  e  $\mathbf{w}^s$  correspondem às porções do vetor de pesos que armazenam informações sobre a entrada e a saída, respectivamente, do mapeamento em estudo.

O mapa auto-organizável treinado pode ser empregado para obter estimativas para a saída do mapeamento diretamente a partir da porção do vetor de pesos sinápticos correspondente ao dado de saída. Pode-se levar em conta apenas o neurônio vencedor ou uma combinação linear das propostas de saída de uma vizinhança do neurônio vencedor.

Assim, os vetores de entrada e de saída são definidos da seguinte forma:

$$\mathbf{x}^e(t) = [y(t), \dots, y(t - n_y + 1); u(t), \dots, u(t - n_u + 1)] \quad (3.10)$$

$$\mathbf{x}^s(t) = y(t + 1) \quad (3.11)$$

O neurônio vencedor é computado de maneira similar àquela definida na seção 3.5:

$$i^*(t) = \arg \min_j \|\mathbf{x}^e(t) - \mathbf{w}_j^e(t)\| \quad (3.12)$$

Note que o cálculo do neurônio vencedor envolve somente a parte correspondente às entradas do mapeamento.

O processo de adaptação sináptica também ocorre de forma similar àquela definida na Seção 3.7, sendo contudo efetuado isoladamente para as porções de pesos correspondentes à entrada e saída do mapeamento:

$$\mathbf{w}_{j+1}^e = \mathbf{w}_j^e + \Delta \mathbf{w}_j^e \quad (3.13)$$

$$\mathbf{w}_{j+1}^s = \mathbf{w}_j^s + \Delta \mathbf{w}_j^s, \quad (3.14)$$

onde

$$\Delta \mathbf{w}_j^k = \eta(n) \cdot h_{j,i(\mathbf{x})} \left( \mathbf{x}^k - \mathbf{w}_j^k \right). \quad (3.15)$$

Assim, ao final do processo de treinamento, a saída esperada para o mapeamento é dada por:

$$\hat{y}(t+1) = \mathbf{w}_{i^*}^s(t), \quad (3.16)$$

onde  $i^*$  é calculado de acordo com a Equação 3.12.

Como resultado, os autores demonstraram que o erro de aproximação decresce com o aumento do número de épocas de treinamento e que o paradigma não-supervisionado, embora pouco explorado neste caso, constitui uma opção interessante para problemas de regressão.

### **3.10.4 Organização e Seleção de Documentos**

Com a difusão da Internet, uma das tarefas computacionais mais comuns tornou-se a recuperação e extração de dados a partir de repositórios. Dessa forma, ferramentas que conduzam a uma busca rápida e eficiente por documentos de acordo com suas características ou similaridades com outros documentos têm se tornado cada vez mais necessárias. Os mapas auto-organizáveis vêm sendo aplicados com sucesso na classificação de documentos.

O WEBSOM (HONKELA *et al.*, 1997) constitui uma ferramenta para organizar documentos na forma de texto para fins de classificação e análise de documentos textuais. O algoritmo WEBSOM é baseado em um mapa auto-organizável com vizinhança bidimensional de forma que documentos relacionados sejam vizinhos neste mapa. A arquitetura básica utilizada pelo WEBSOM envolve dois níveis. Em uma primeira etapa ocorre um processo de aprendizagem em que palavras individuais são relacionadas com base em um breve contexto. As palavras são

ordenadas em um mapa bidimensional, formando um mapa capaz de retornar a categoria da palavra. Palavras similares tendem a ser mapeadas para um mesmo neurônio ou para neurônios vizinhos. As palavras são codificadas em um vetor  $n$ -dimensional de valores reais. Mais detalhes acerca do método podem ser encontrados em LAGUS *et al.* (1996). Esse mapa é empregado para formar o histograma de categorias para palavras de documentos completos, compondo uma espécie de “impressão digital” do documento. O histograma de categorias serve como entrada para um segundo mapa auto-organizável, o qual estabelecerá similaridades entre os documentos.

### **3.10.5 Segmentação de Imagens**

Uma das aplicações mais importantes na área de processamento de imagens é o processo de *segmentação*. Segmentação de imagens consiste na divisão da imagem em regiões com atributos similares. Os mapas auto-organizáveis podem ser empregados sem alterações substanciais para detectar as principais características presentes em uma imagem (IIVARINEN, & VISA, 1998).

As características são representadas pelo cromatismo da imagem, ou seja, nos componentes RGB (vermelho, verde e azul). Cada pixel será representado por um vetor de características. A título de exemplificação, considere um vetor de dimensão cinco em que os atributos são linha, coluna, componente vermelho, componente verde e componente azul. Cada vetor é uma entrada para o mapa auto-organizável. Após o treinamento, vetores topologicamente próximos são mapeados para a mesma classe (grupo de neurônios).

# CAPÍTULO 4

## SOLUÇÃO DE PROBLEMAS COMBINATÓRIOS UTILIZANDO REDES NEURAS AUTO-ORGANIZÁVEIS

---

---

Este capítulo propõe um novo método de otimização multicritério, baseado em mapas auto-organizáveis de Kohonen, destinado à solução de problemas combinatórios, em particular o problema de roteamento de veículos capacitados, apresentado no Capítulo 2. A aplicação de redes neurais com treinamento não-supervisionado a problemas combinatórios constitui uma abordagem pouco difundida na literatura e, em geral, é restrita a problemas relativamente simples. Este capítulo apresenta um estudo acerca da aplicação de mapas auto-organizáveis a problemas combinatórios de elevada complexidade, bem como a proposição de novas heurísticas a serem utilizadas em associação com mapas auto-organizáveis. Nas primeiras seções será detalhada a aplicação de mapas auto-organizáveis ao problema do caixeiro viajante simétrico. Nas seções seguintes é apresentada a extensão para o problema de roteamento de veículos capacitados.

### 4.1 Introdução

Embora existam algumas abordagens baseadas em mapas auto-organizáveis aplicadas ao problema do caixeiro viajante simétrico (ARAS *et al.*, 1999; ANGÉNIOL *et al.*, 1988; BURKE & DAMANY, 1992), poucos trabalhos enfocam problemas mais complexos, tais como roteamento de veículos (MODARES *et al.*, 1999), os quais envolvem o tratamento conjunto de múltiplos mapas auto-organizáveis. Neste capítulo é apresentado um método de otimização multicritério baseado em mapas auto-organizáveis (KOHONEN, 1982) e inferência nebulosa (PEDRYCZ &

GOMIDE, 1998; ZADEH, 1965). O algoritmo proposto implementa uma política de penalidades e recompensas, uma estratégia para inibição, inserção e poda de neurônios, levando em conta características estatísticas do espaço de entrada e a existência de múltiplos mapas. Conceitos de sistemas nebulosos são empregados visando minimizar certas limitações relacionadas a incertezas e disponibilidade de somente informações parciais, conduzindo a um processo adaptativo de relaxação de restrições. Para uma introdução acerca dos principais conceitos de lógica nebulosa utilizados neste capítulo, consultar o Apêndice D.

Embora o método a ser proposto possa ser aplicado a uma variedade de contextos, como agrupamento e reconhecimento de padrões, este capítulo enfoca problemas combinatórios. Em particular, serão considerados como casos de estudo os problemas de roteamento de veículos capacitados (CVRP) e de múltiplos caixeiros viajantes (MTSP) (MODARES *et al*, 1999). Nestes casos, o processo de otimização requer o treinamento de múltiplos mapas auto-organizáveis simultaneamente, de forma que cada rota esteja associada a um único mapa.

Os neurônios presentes no mapa auto-organizável apresentam relação de vizinhança unidimensional e circular, ou seja, o último neurônio é vizinho do primeiro. Essas características permitem que se interprete o mapa como um percurso fechado, de modo que cada mapa estará associado ao trajeto de um veículo.

Visando tratar de forma eficiente os múltiplos objetivos envolvidos no CVRP, o treinamento da rede neural é baseado em uma série de regras de inferência nebulosa envolvendo um conjunto de variáveis lingüísticas. Sendo assim, nesta tese é proposto um sistema híbrido combinando mapas auto-organizáveis de Kohonen e inferência nebulosa. Os principais aspectos do método apresentado envolvem: a) treinamento simultâneo de múltiplos mapas auto-organizáveis; b) extensão da política de inserção e poda de neurônios, introduzida por Angéniol e colaboradores (ANGÉNIOL *et al*, 1988; ARAS *et al*, 1999), do caixeiro viajante simétrico para o problema de roteamento de veículos capacitados; c) aprendizado competitivo guiado por regras nebulosas que consideram múltiplos mapas e os múltiplos critérios a serem tratados durante a otimização; e d) relaxação adaptativa de restrições.

O desempenho da abordagem proposta é discutido na Seção 4.3.3, através de uma série de simulações realizadas junto a problemas de teste adotados na literatura.

## **4.2 Aplicação de Mapas Auto-Organizáveis ao Problema do Caixeiro Viajante Simétrico**

Uma aplicação interessante dos mapas auto-organizáveis com vizinhança unidimensional é o problema do caixeiro viajante simétrico (ANGÉNIOL *et al.*, 1988; ARAS *et al.*, 1999). A arquitetura da rede é definida por duas camadas, sendo a primeira (camada de entrada) constituída pelo conjunto de padrões de entrada, que para o caso do TSP correspondem às coordenadas das cidades em um plano, e uma segunda camada formada pelos neurônios de saída. Assim como definido para a rede de Kohonen convencional, o treinamento da rede neural envolve as seguintes etapas: inicialização sináptica; processo competitivo; processo cooperativo; adaptação sináptica. Nas próximas seções será apresentado um algoritmo para o problema do caixeiro viajante baseado nas abordagens de ANGÉNIOL *et al.* (1988) e ARAS *et al.* (1999). O algoritmo envolve estratégias de inserção, poda e inibição de neurônios.

### **4.2.1 Inicialização Sináptica**

Para o caso do TSP, uma inicialização sináptica adequada se faz necessária. O mapa tende a alcançar diferentes organizações para diferentes tipos de inicialização. Uma inicialização e/ou uma definição de vizinhança inadequadas podem levar a soluções de custo elevado.

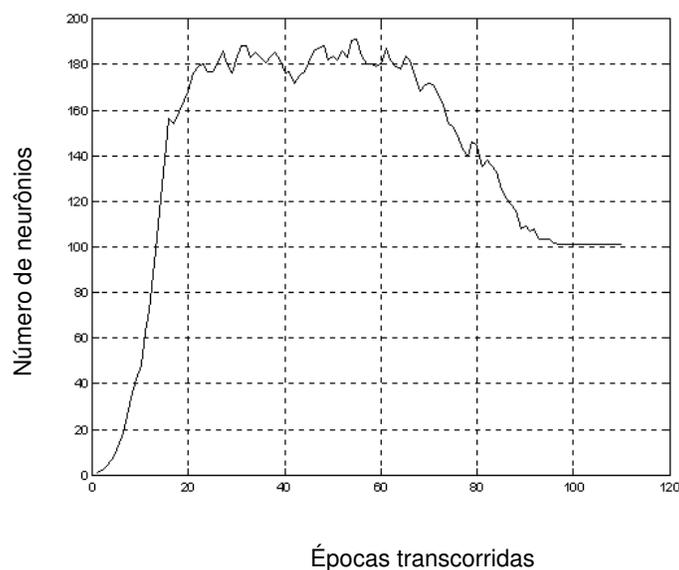
Nesta tese, uma inicialização simples que se mostrou eficiente é efetuada de modo que um único neurônio seja posicionado no centro de massa definido pelas coordenadas das cidades. Esta estratégia requer que o número de neurônios seja variável ao longo do treinamento, tal como no modelo proposto por ANGÉNIOL *et al.* (1988). Outra abordagem bastante empregada e que leva a um processo eficiente de organização do mapa é a inicialização em forma de anel, de modo que o centro de área definido pelos neurônios que compõem o anel coincida com o centro de área definido para as coordenadas das cidades. A relação de vizinhança entre os neurônios é linear e circular, ou seja, o último neurônio é vizinho do primeiro. A relação de vizinhança adotada será detalhada na Seção 4.2.3.

### **4.2.2 O Processo Competitivo**

Uma vez inicializado o mapa, os neurônios são submetidos a um processo competitivo. O processo de definição do neurônio vencedor é idêntico ao processo padrão descrito no Capítulo 3: para cada padrão de entrada, mais especificamente para cada vetor de coordenadas das cidades, é computado um neurônio vencedor de acordo com um critério de similaridade. O vencedor será o neurônio cujo vetor de pesos esteja mais próximo do padrão apresentado em termos de distância euclidiana.

Uma importante distinção entre o algoritmo implementado para o TSP e o algoritmo convencional apresentado no Capítulo 3 reside no controle automático do número de neurônios através de procedimentos de inserção e poda. Além disso, o algoritmo apresenta uma estratégia de inibição de neurônios com o objetivo de promover uma ordenação mais eficiente do mapa: neurônios que vencem com muita frequência são inibidos, ou seja, não participam do processo competitivo durante um certo número de épocas. Caso um determinado neurônio vença o processo competitivo mais de uma vez em um mesmo ciclo, ele é inibido e um novo neurônio é inserido como vizinho imediato e sobre as mesmas coordenadas do neurônio vencedor. O neurônio recém-inserido passa a ser o novo vencedor. Caso um neurônio perca o processo competitivo ao longo de três épocas consecutivas, ele é eliminado da rede e a relação de vizinhança é restabelecida entre vizinhos adjacentes ao neurônio removido. Assim, o número de neurônios varia com o transcorrer das épocas, de forma a contornar problemas relacionados a neurônios que vencem o processo competitivo com muita frequência e neurônios que nunca ou raramente vencem.

A Figura 4.1 ilustra o número de neurônios presentes no mapa com o transcorrer das épocas para uma instância contendo 101 cidades, exibindo a variação resultante dos procedimentos de inserção e poda de neurônios. Observe que, no caso da implementação realizada nesta tese, o processo se inicia com um único neurônio, o número de neurônios se mantém aproximadamente constante e próximo ao dobro do número de cidades durante o processo de organização do mapa, e decai gradativamente, até atingir um valor igual ao número de padrões de entrada (número de cidades).



**Figura 4.1.** Variação do número de neurônios com o transcorrer das épocas para um problema do caixeiro viajante simétrico com 101 cidades, ilustrado na Figura 4.3.

### 4.2.3 O Processo Cooperativo

Conforme descrito no Capítulo 3, um padrão de entrada ativa o neurônio vencedor e uma vizinhança topológica. Essa vizinhança é definida durante o processo cooperativo. Uma vizinhança de grau  $n$  define que o neurônio vencedor influenciará um conjunto de  $2 \cdot n$  vizinhos. No caso do algoritmo para o caixeiro viajante proposto por ARAS *et al.* (1999), a relação de vizinhança é unidimensional, conforme mostrado na Figura 4.2. A Figura 4.2 apresenta uma rede com dez neurônios e um caso ilustrativo com vizinhança de grau 2. Dessa forma, supondo que o neurônio identificado por '3' seja o vencedor, ele possuirá quatro neurônios vizinhos. Um aspecto importante no caso do TSP é que a vizinhança é circular, ou seja, o neurônio '1' é vizinho dos neurônios '2' e '10'. Para os algoritmos implementados nesta tese, o grau de vizinhança é definido como sendo função do número de neurônios. Considerando que o número de neurônios é variável durante o treinamento, dado que o algoritmo conta com mecanismos de inserção e poda, o grau de vizinhança também varia.

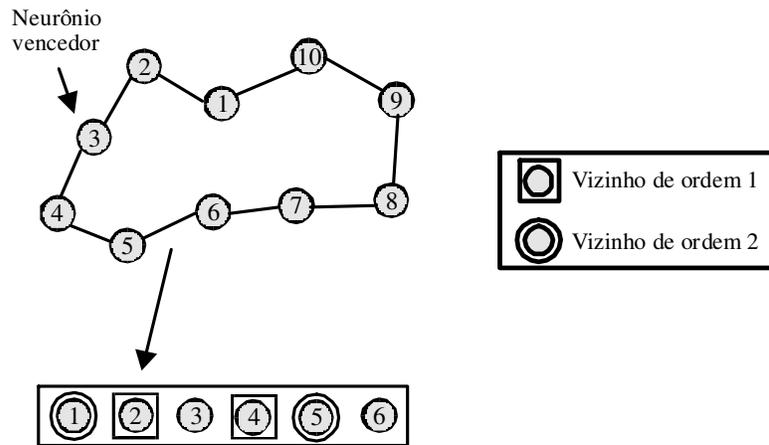


Figura 4.2. Relação de vizinhança adotada para o problema do caixeiro viajante simétrico.

#### 4.2.4 Atualização Sináptica

A rede neural é treinada de forma seqüencial, por meio de sucessivas apresentações dos padrões de entrada. Ao início de cada época, os padrões de entrada são ordenados aleatoriamente a fim de que o processo de organização do mapa se mantenha o mais dissociado possível da seqüência de apresentação dos padrões. Para cada entrada apresentada à rede é determinado, via uma estratégia competitiva, um neurônio vencedor. Em seguida, os pesos sinápticos do neurônio vencedor e sua vizinhança são atualizados de forma a se aproximarem das coordenadas do padrão apresentado. Com o transcorrer das épocas, o mapa inicial se conforma de modo a se adaptar às coordenadas das cidades.

#### 4.2.5 Convergência do Mapa

O processo de organização do mapa é finalizado quando cada neurônio se encontrar associado a uma cidade, isto é, todas as cidades apresentarem distância menor que um limiar  $\lambda$  para um dado neurônio, conforme especificado no algoritmo 4.1.

**Procedimento [CondPar]=VERIFICA\_CONDICÃO\_DE\_PARADA( $P,N$ )**

1. Calcular menor distância  $d$  entre coordenadas das cidades em  $P$
2. CondPar=Verdadeiro
3. Para cada cidade  $i$  em  $P$ 
  - 3.1. Calcular neurônio  $n_i$  mais próximo em  $N$
  - 3.2. Se  $\{n_i\} \in NA$  ou  $\|n_i, P_i\| > \lambda$  então
    - 3.2.1. CondPar=Falso
    - 3.2.2. Vá para Fim Procedimento
  - 3.3. Senão
    - 3.3.1.  $NA = NA \cup \{n_i\}$
  - 3.4. Fim Se
4. Fim Para

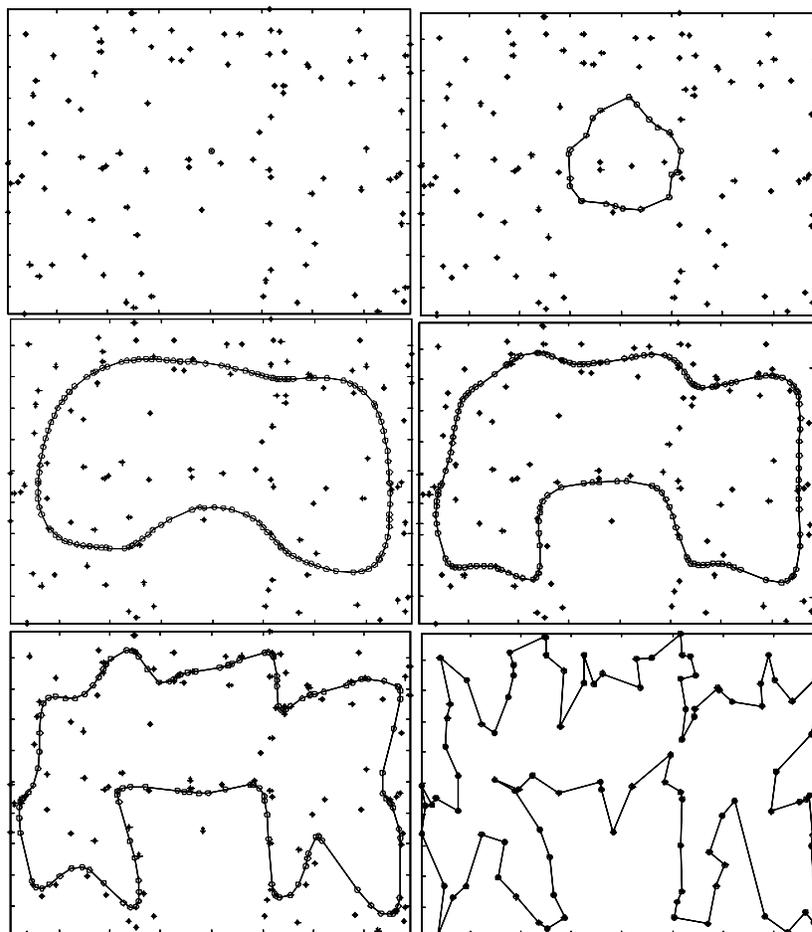
**Fim Procedimento**

**Algoritmo 4.1.** Verificação da condição de parada para o problema do caixeiro viajante simétrico.

No Algoritmo 4.1,  $P$  é o conjunto de coordenadas das cidades,  $N$  é o conjunto de todos os neurônios presentes no mapa e  $NA$  é o conjunto de neurônios que se encontram associados a cidades.

A Figura 4.3 exibe diversos estágios do processo de organização do mapa produzido pelo algoritmo de ANGÉNIOL *et al.* (1988) aplicado a um problema do caixeiro viajante simétrico envolvendo 101 cidades. Ao longo do processo de auto-organização, o número de neurônios varia de acordo com a distribuição de vitórias associadas aos neurônios que compõem o mapa naquele momento. Caso um neurônio vença mais de uma vez o processo competitivo em uma mesma época, um novo neurônio é adicionado sobre suas coordenadas e como vizinho imediato. Essa estratégia garante que regiões apresentando alta densidade de padrões sejam supridas por um número maior de neurônios. O mecanismo de poda e inserção de neurônios é ilustrado no Algoritmo 4.2. Ao longo do treinamento, os neurônios gradativamente se aproximam das cidades. A cada época a condição de parada é verificada de acordo com o

limiar  $\lambda$ . Caso a condição de parada seja atingida, os neurônios têm seus pesos atualizados de forma que coincidam com as coordenadas da cidade mais próxima. Os neurônios que não se encontrarem associados a uma cidade são eliminados da rede. Assim, ao final, têm-se somente os neurônios associados às cidades. Dessa forma, ao atingir o critério de parada, a convergência do mapa para uma solução é garantida. A qualidade da solução final depende do mecanismo de aprendizagem e de parâmetros da rede, tais como taxa de aprendizagem e dinâmica dos neurônios.



**Figura 4.3.** Evolução do processo de organização do mapa auto-organizável para um problema do caixeiro viajante simétrico com 101 cidades.

#### 4.2.6 O Algoritmo

A seguir são apresentados os principais passos do algoritmo baseado em mapas auto-organizáveis para solução do TSP (ANGÉNIOL *et al.*, 1988; ARAS *et al.*, 1999).

##### Procedimento SOM\_TSP( $P$ )

1. Normalizar vetores de coordenadas das cidades contidos em  $P$
2. Inicializar mapa auto-organizável
3. CondPar=Falso
4. Enquanto CondPar=Falso faça
  - 4.1. Reordenar aleatoriamente os padrões de entrada
  - 4.2. Para cada padrão de entrada  $p \in P$  faça
    - 4.2.1. Selecionar o neurônio vencedor do conjunto de neurônios  $N$  (neurônio não inibido mais próximo de  $p$ )
    - 4.2.2. Se neurônio vencedor já venceu alguma vez na época então
      - 4.2.2.1. Inibir neurônio vencedor
      - 4.2.2.2. Inserir novo neurônio com vetor de pesos idêntico ao do vencedor e como vizinho imediato do vencedor
      - 4.2.2.3. Neurônio vencedor  $\leftarrow$  novo neurônio inserido
    - 4.2.3. Fim Se
    - 4.2.4. Definir vizinhança para neurônio vencedor
    - 4.2.5. Executar processo adaptativo
  - 4.3. Fim Para
  - 4.4. Remover neurônios inativos do mapa auto-organizável e redefinir vizinhança
  - 4.5. Desinibir todos os neurônios inibidos
  - 4.6. CondPar=VERIFICA\_CONDIÇÃO\_DE\_PARADA( $P,N$ )
5. Fim Enquanto
6. Associar cidades aos neurônios mais próximos e ainda disponíveis
7. Eliminar neurônios que não se encontram associados a cidades do mapa auto-organizável

##### Fim Procedimento

**Algoritmo 4.2.** Mapa auto-organizável aplicado ao problema do caixeiro viajante simétrico.

### 4.3 Mapas Auto-Organizáveis Aplicados ao Problema de Roteamento de Veículos

Considerando o sucesso da aplicação de mapas auto-organizáveis a problemas do tipo TSP (ANGÉNIOL *et al.*, 1988; ARAS *et al.*, 1999; FAVATA & WALKER, 1991; FORT, 1998), nesta tese é proposta a extensão do método ao problema de roteamento de veículos considerando um conjunto de veículos idênticos com limite de capacidade e um único depósito central.

Entretanto, a extensão não é trivial. O critério de similaridade envolvendo apenas a distância euclidiana entre o neurônio em processo de competição e o padrão de entrada é bastante adequado a problemas do tipo caixeiro viajante, que podem ser vistos como casos particulares de um problema de roteamento de veículos. Entretanto, quando se consideram classes mais complexas de problemas de roteamento, mostra-se necessário o uso de critérios de similaridade mais elaborados. No caso do problema de roteamento de veículos, o procedimento de obtenção de uma função de similaridade é bem mais complexo, pois muitas vezes envolve o tratamento de múltiplos objetivos, geralmente conflitantes, não correlacionados e expressos em escalas e/ou unidades distintas.

#### 4.3.1 Abordagens Existentes Empregando Mapas Auto-Organizáveis

Existem disponíveis na literatura alguns trabalhos tratando o problema de roteamento de veículos através de mapas auto-organizáveis (GHAZIRI, 1991; MICHALEWICZ & FOGEL, 2002; MODARES *et al.*, 1999). Devido à existência de restrições a serem respeitadas no problema de roteamento de veículos, o treinamento da rede neural se dá de forma bastante distinta daquelas propostas em ARAS *et al.* (1999) e ANGÉNIOL *et al.* (1988) para o problema do caixeiro viajante simétrico. O aprendizado competitivo é, em geral, baseado na inicialização dos pesos sinápticos em  $n$  anéis de neurônios posicionados em torno do depósito central, onde  $n$  é o número de veículos em consideração. Esses anéis são modelados ao longo do processo de adaptação sináptica, até que neurônios possam ser associados a consumidores. Neste ponto, o critério de parada é alcançado e teremos  $n$  rotas atendendo aos consumidores. Nas abordagens atualmente tratadas na literatura, o número de neurônios é fixo durante o treinamento. Em

MODARES *et al.* (1999), a rede opera com um número de neurônios igual ao dobro do número de consumidores.

GHAZIRI (1991) propôs o uso de múltiplas redes neurais auto-organizáveis para resolver o problema de roteamento de veículos capacitados. Com o objetivo de assegurar a factibilidade da solução, o autor adotou uma abordagem em que neurônios pertencentes a uma rota sobrecarregada, ou seja, uma rota em que a carga do veículo seja muito próxima ou maior que sua capacidade, são impedidos de vencer o processo competitivo. O processo competitivo envolve a seleção de um mapa vencedor, que é definido com base em uma função de distribuição de probabilidade. Uma vez que o mapa vencedor é definido, o padrão de entrada atual (consumidor) é associado ao neurônio mais próximo situado no mapa vencedor. Como já mencionado, cada mapa auto-organizável vai corresponder a uma rota a ser atribuída a um dos veículos. Esse neurônio e sua vizinhança têm seus pesos sinápticos modificados de forma a se aproximarem do padrão de entrada em consideração.

A função de similaridade entre neurônios e consumidores utilizada na abordagem de MODARES *et al.* (1999) explora a forma quadrática da violação da capacidade, de acordo com a seguinte equação:

$$J = d(i, j) + v \cdot \frac{q^2}{Q^2}, \quad (4.1)$$

onde  $v$  é um parâmetro de ponderação,  $d(i, j)$  é a distância euclidiana entre o vetor de pesos do neurônio  $i$  e o consumidor  $j$ ,  $q$  é a carga correntemente suportada pela rota (mapa) em que o neurônio  $i$  se encontra posicionado e  $Q$  é a capacidade do veículo associado àquela rota. No caso das instâncias tratadas aqui, todos os veículos apresentam capacidades idênticas.

Com o objetivo de superar as limitações da abordagem proposta por Ghaziri, no trabalho de MODARES *et al.* (1999), a seleção do neurônio vencedor é realizada de forma que as restrições sejam respeitadas. Evita-se assim a aplicação de medidas severas, como a inibição completa de neurônios pertencentes a rotas sobrecarregadas. Esse controle é efetuado pelo termo  $q^2/Q^2$ . Esse termo reflete a utilização da carga de um veículo com relação à sua capacidade,

sendo idêntico para todos os neurônios pertencentes a uma determinada rota. Observe que busca-se alcançar um compromisso entre factibilidade e qualidade da solução, não havendo garantia de que todas as restrições sejam completamente respeitadas. O parâmetro  $\nu$  é responsável por estabelecer o compromisso entre factibilidade e qualidade da solução. Considerando que a distância entre neurônios e consumidores é elevada em estágios iniciais do processo competitivo, o parâmetro  $\nu$  é reduzido com o transcorrer das épocas visando evitar dominância do termo  $q^2/Q^2$  em estágios avançados do processo adaptativo da rede neural. Na seção de resultados computacionais, os autores compararam o desempenho de seu algoritmo com os resultados de uma implementação da *rede elástica* proposta por Vakhutinsky e Golden (YAGER *et al.*, 1987), sendo que a abordagem auto-organizável apresentou melhor desempenho em todos os casos testados. O desvio da melhor solução conhecida, no pior caso, foi de 6,5% e, na média, em torno de 4%.

MATSUYAMA (1991;1996) propôs um método de otimização multicritério baseado em mapas auto-organizáveis denominado “competição harmônica”. Competição harmônica consiste no processo de se selecionar um neurônio vencedor considerando um custo principal e diversos subcustos, efetuando um controle dinâmico de parâmetros sobre os termos de ponderação dos subcustos. A competição harmônica foi aplicada a diversos contextos, como problemas de quantização, o problema do caixeiro viajante e problemas de roteamento de veículos. Diferentemente das abordagens propostas em GLOVER & LAGUNA (1998) e MODARES *et al.* (1999), a inicialização do mapa auto-organizável não é feita através da distribuição de neurônios em  $n$  anéis em torno do depósito. O número de neurônios é fixo e os pesos sinápticos iniciais são definidos em uma elipse em torno do depósito. Durante a fase competitiva, um neurônio vencedor é selecionado para cada consumidor e  $n$  neurônios vencedores são selecionados para o depósito, assegurando que todas as rotas incluam o depósito. As simulações apresentadas pelo autor envolvem variações do problema de roteamento de veículos, incluindo restrições adicionais tais como consumidores apresentando preferências por certos veículos. Entretanto, não foi realizado um estudo envolvendo problemas de teste disponíveis na literatura e, portanto, o trabalho não permite comparações com outras abordagens ou com as melhores soluções disponíveis na literatura.

### **4.3.2 Uma Abordagem Neuro-Nebulosa para o Problema de Roteamento de Veículos Capacitados**

Nesta seção são apresentados alguns aspectos gerais da proposta desta tese para a solução do problema de roteamento de veículos, apontando as principais características que distinguem a abordagem proposta de outras disponíveis na literatura.

Assim como para o TSP, a arquitetura da rede é composta de duas camadas. A primeira recebe os padrões de entrada, os quais correspondem às coordenadas dos consumidores em um plano, e a segunda camada é composta pelos neurônios de saída. Tal como definido para o algoritmo do caixeiro viajante na Seção 4.2, a rede neural é treinada através de sucessivas apresentações dos padrões de entrada.

O principal aspecto distintivo da abordagem proposta reside no treinamento da rede neural, o qual é guiado por um conjunto de regras nebulosas (PEDRYCZ & GOMIDE, 1998) que consideram características estatísticas do problema a ser resolvido. A estratégia de poda, inserção e inibição de neurônios proposta por ARAS *et al.* (1999) para o problema do caixeiro viajante foi estendida para o CVRP, visando lidar com neurônios que vencem o processo competitivo com elevada frequência e neurônios que raramente vencem. Dessa forma, neurônios que vencem a competição mais de uma vez na mesma época são excluídos do processo competitivo durante uma época completa e um novo neurônio é inserido sobre as coordenadas do vencedor. Se um neurônio não vence a competição por três épocas consecutivas, ele é considerado inativo e é eliminado do mapa auto-organizável. A convergência do mapa será alcançada quando o número de neurônios for igual ao número de consumidores e cada consumidor se encontrar associado a exatamente um neurônio, de acordo com um limiar de distância previamente estabelecido.

A Figura 4.4 exibe o diagrama de blocos do sistema neuro-nebuloso proposto neste trabalho. Podem ser identificados sete módulos principais:

- Pré-processador: responsável pela geração da configuração inicial dos mapas auto-organizáveis. A proposta deste módulo é identificar regiões no plano caracterizadas por elevadas concentrações de demanda.
- Pesos sinápticos atuais: é um módulo de armazenamento que contém os pesos sinápticos dos neurônios na configuração atual dos mapas.
- Estimador das demandas dos veículos: não existem neurônios definitivamente associados a consumidores até que se atinja a convergência do processo de ordenação dos mapas auto-organizáveis. Dessa forma, as demandas servidas por cada veículo devem ser estimadas de acordo com a configuração atual dos mapas.
- Módulo Nebuloso I: relaxação de restrições: técnicas baseadas em relaxação de restrições visam estabelecer um compromisso entre qualidade e factibilidade da solução, evitando a aplicação de penalidades severas.
- Módulo Nebuloso II: estimador dos custos das rotas: o objetivo deste módulo é estimar o custo de cada rota com base na carga correntemente suportada e na distância entre o centróide formado pelo mapa correspondente à rota e as coordenadas dos consumidores.
- Definição do neurônio vencedor e sua vizinhança: este módulo determina o neurônio vencedor de acordo com uma função-objetivo e identifica uma vizinhança de neurônios estimulados.
- Adaptação sináptica: efetua a adaptação sináptica do neurônio vencedor e de sua vizinhança.

As seções que se seguem irão especificar em detalhes os módulos do diagrama de blocos e apresentar os principais passos do algoritmo.

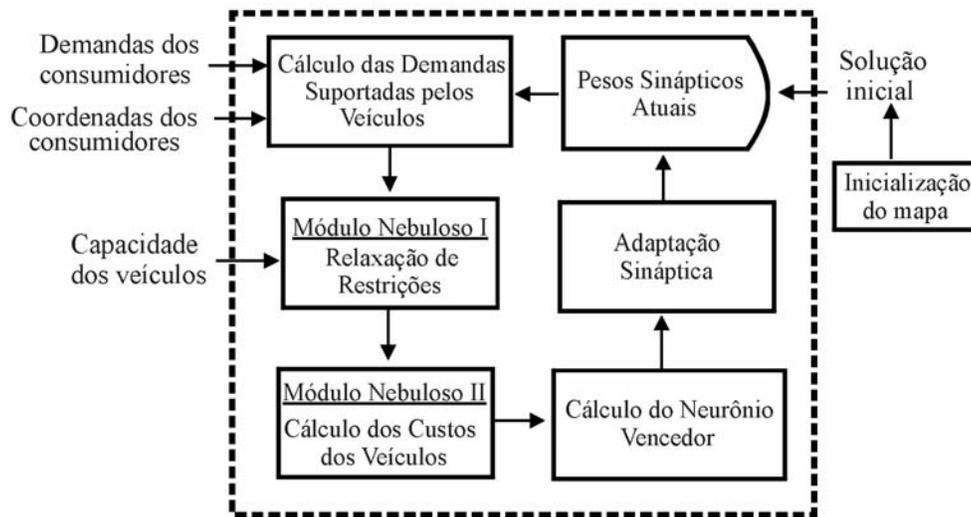
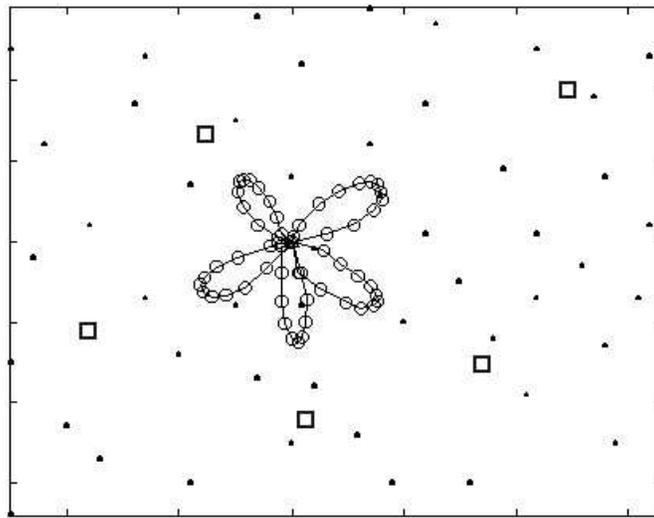


Figura 4.4. Diagrama de blocos do sistema neuro-nebuloso.

#### 4.3.2.1 O Módulo de Pré-Processamento

Durante a fase de inicialização sináptica, as coordenadas dos neurônios são organizadas em  $M$  pequenos anéis em torno das coordenadas do depósito, onde  $M$  é igual ao número de rotas (veículos). Em um primeiro momento, cada anel possui 10 neurônios. Esse valor é arbitrário e não exerce influência significativa sobre a solução final, uma vez que neurônios podem ser inseridos ou removidos do mapa durante o processo de adaptação da rede. A distribuição inicial dos pesos sinápticos dita alguns aspectos da configuração final do mapa, como a ausência de pontos de cruzamento. Dessa forma, uma inicialização adequada é de grande importância para o desempenho do algoritmo. No algoritmo proposto neste trabalho, uma inicialização adequada é alcançada através de uma etapa de pré-processamento efetuada por um algoritmo de agrupamento. Esse pré-processamento é responsável por gerar uma representação compacta dos dados através de protótipos (padrões de entrada), identificando eventuais grupos de consumidores e regiões de elevada concentração de demanda. Os  $M$  anéis são gerados de forma que cada mapa auto-organizável tenda a ser responsável por um dos grupos detectados. A Figura 4.5 ilustra a condição inicial para a instância  $E-n51-k5$ , a ser apresentada posteriormente. Nesse caso, os pesos sinápticos iniciais encontram-se distribuídos

em 5 anéis, cujas direções são definidas pela rede auto-organizável para pré-processamento. Na figura, os quadrados denotam os protótipos dos grupos representando regiões de concentração de demandas e consumidores. Cada mapa caracteriza uma rota ou veículo. Vale ressaltar que o mapa auto-organizável responsável pelo posicionamento dos quadrados opera na forma convencional proposta por KOHONEN (1997), conforme descrito no Capítulo 3, e pode possuir uma vizinhança uni ou bidimensional. O número de veículos deve ser previamente definido, constituindo um parâmetro de entrada para o mapa auto-organizável.



**Figura 4.5.** Inicialização do mapa auto-organizável para a instância E-n51-k5.

#### 4.3.2.2 Processo Competitivo

Existem duas variáveis importantes no cálculo do neurônio vencedor: distância euclidiana entre consumidores e neurônios e custos dos veículos. A distância entre o consumidor e o neurônio em competição é mapeada para um valor real, na faixa entre 0 e 1, considerando valores relativos. Para o consumidor atual, obtêm-se os neurônios mais próximo e mais distante em termos de distância euclidiana. Sejam *Min\_Distância\_Neurônio* e *Max\_Distância\_Neurônio* as distâncias mínima e máxima, respectivamente. A distância entre o neurônio em competição e o consumidor atual é mapeada para um valor *ND* entre 0 e 1, de acordo com uma função linear. Dessa forma, *Min\_Distância\_Neurônio* corresponderá ao valor 0, enquanto *Max\_Distância\_Neurônio* corresponderá ao valor 1. O neurônio vencedor será

aquele que minimiza a seguinte função, cujos termos serão detalhados nas seções que se seguem:

$$J = ND + \text{Custo\_Veículo} \times F \times \text{Saturação\_Veículo}, \quad (4.2)$$

onde:

- *ND*: distância relativa entre o neurônio em competição e o consumidor apresentado à rede.
- *Custo\_Veículo*: custo do veículo correspondente ao mapa auto-organizável do qual o neurônio em competição é membro.
- *Saturação\_Veículo*: nível de saturação do veículo com relação à sua capacidade.
- *F*: parâmetro de ponderação cujo valor decresce (até um valor mínimo configurável pelo usuário, de acordo com o termo que se deseja privilegiar na solução) com o transcorrer das épocas, conforme as cargas efetivas tornam-se mais próximas das cargas aproximadas, evitando que as restrições de carga se sobreponham às distâncias em estágios avançados do processo de auto-organização do mapa.

O processo competitivo baseia-se na definição de um conjunto de variáveis lingüísticas nebulosas e em uma base de regras nebulosas que guia a competição.

#### 4.3.2.3 Módulo Nebuloso I: Relaxação de Restrições

A formulação do problema de roteamento de veículos considerada nesta tese interpreta restrições em termos de capacidades dos veículos. No caso de sempre rejeitar qualquer grau de violação da capacidade, neurônios posicionados em mapas auto-organizáveis correspondentes a veículos sobrecarregados não poderiam vencer o processo competitivo em nenhuma circunstância, o que tende a impactar negativamente sobre a distância total das rotas obtidas. Por outro lado, se uma certa violação da capacidade é aceita, um importante fator que deve ser levado em conta é o nível de violação. O tratamento da violação de capacidade adotado neste trabalho é baseado em uma metodologia nebulosa aplicada à programação matemática (SAKAWA, 1993). As restrições são relaxadas, considerando uma faixa de tolerância

relativamente ampla em estágios iniciais do processo de organização do mapa e tornando-se mais severas com o decorrer do processo adaptativo. A relaxação de restrições é implementada com base em uma função de saturação do veículo, de acordo com a seguinte fórmula:

$$VS = \begin{cases} 0 & \text{se } CL \leq LB, \\ \frac{CL - LB}{UB - LB} & \text{se } LB < CL < UB, \\ 1 & \text{se } CL \geq UB \end{cases} \quad (4.3)$$

onde  $CL$  é a carga correntemente suportada pelo veículo,  $LB$  e  $UB$  são os limitantes inferior (igual ou menor que a capacidade do veículo) e o limitante superior (igual ou maior que a capacidade do veículo), respectivamente, para a carga do veículo em análise. O grau de violação da capacidade pode assumir qualquer valor entre 0 e 1, onde 0 denota o grau de violação mínimo e 1 denota o grau de violação máximo. A Figura 4.6 ilustra uma função que implementa o processo de relaxação de restrições. Dessa forma, a saturação do veículo  $VS$  reflete a adequabilidade de cada veículo em termos da carga total correntemente suportada. O limitante inferior é calculado da seguinte forma:

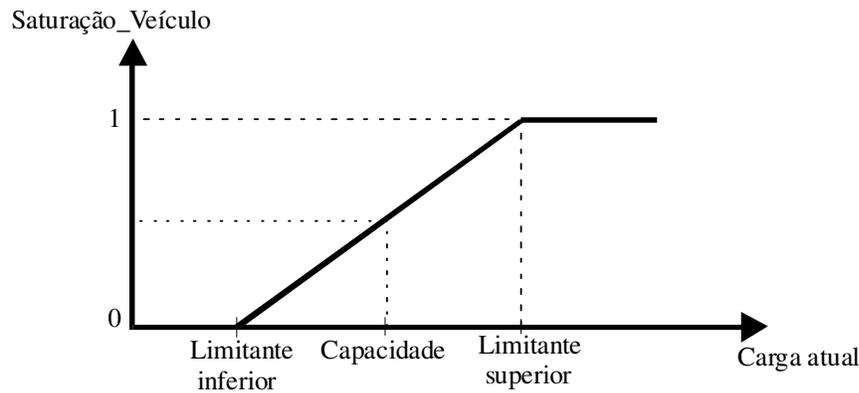
$$LB = \text{Min}\left(\frac{Q}{2}, \frac{1}{N} \cdot \bar{D}\right), \quad (4.4)$$

onde  $Q$  é a capacidade do veículo,  $\bar{D}$  é a média das demandas de todos os consumidores,  $N$  é o número de veículos e  $\text{Min}(A,B)$  retorna o menor valor entre  $A$  e  $B$ .

O limitante superior é calculado como segue:

$$UB = \text{Max}\left(Q, \frac{1}{N} \cdot \bar{D}\right), \quad (4.5)$$

onde  $\text{Max}(A,B)$  retorna o maior valor entre  $A$  e  $B$ .



**Figura 4.6.** Relaxação de restrições: saturação do veículo.

#### 4.3.2.4 Módulo Nebuloso II: Estimativas dos Custos dos Veículos

Este módulo implementa a base de regras que gerencia o processo de treinamento do mapa auto-organizável. Neste ponto, são consideradas certas características globais que refletem o estado geral dos vários mapas auto-organizáveis correspondentes aos veículos:

- Centróide de cada mapa auto-organizável, calculado de acordo com a Equação 4.6.
- Variância das demandas correntemente atribuídas a cada mapa auto-organizável, dada pela Equação 4.9.

O cálculo dos centróides de cada mapa auto-organizável é baseado nos pesos sinápticos dos neurônios pertencentes ao mapa. O centróide da rota  $v$  é dado pela seguinte equação:

$$C_v = \frac{1}{K_v} \sum_{j=1}^{K_v} w_{vj} , \quad (4.6)$$

onde  $K_v$  é o número de neurônios posicionados no mapa auto-organizável que representa o veículo  $v$  e  $w_{vj}$  é o vetor de pesos sinápticos do  $j$ -ésimo neurônio da rota  $v$ .

Diferentemente de outras metodologias comumente aplicadas a problemas de roteamento de veículos, tais como algoritmos genéticos e busca tabu, durante o treinamento do mapa auto-

organizável não existe uma carga efetivamente associada aos veículos. Os anéis são gradualmente expandidos e somente ao final do processo de auto-organização os consumidores estarão definitivamente associados a um veículo. Com o objetivo de estimar as cargas dos veículos, são utilizados dois conceitos:

- carga efetiva;
- carga aproximada.

Para cada padrão de entrada (consumidor) é calculado o neurônio mais próximo. A demanda correspondente ao padrão de entrada é então adicionada à “carga aproximada” do veículo representativo do mapa auto-organizável ao qual pertence o neurônio mais próximo. Se a distância entre o neurônio e o padrão de entrada for menor que um limiar previamente estabelecido, a demanda é então adicionada também à “carga efetiva”. A carga considerada durante o treinamento para um determinado veículo é a soma ponderada das cargas aproximada e efetiva:

$$CL_v = \frac{2 \times AF_v + EF_v}{3}, \quad (4.7)$$

onde  $CL_v$  é a carga associada ao veículo  $v$ ,  $AF_v$  é a carga aproximada do veículo  $v$  e  $EF_v$  é a carga efetiva associada ao veículo  $v$ .

A demanda média para o veículo  $v$  (de cada mapa) é dada por:

$$\bar{D}_v = \frac{1}{N_v} \sum_{n=1}^{N_v} D_{vn}, \quad (4.8)$$

onde  $N_v$  é o número de consumidores servidos pelo  $v$ -ésimo veículo e  $D_{vn}$  é a demanda do  $n$ -ésimo consumidor servido pelo  $v$ -ésimo veículo. Para o cálculo das demandas  $D_{vn}$ , considere-se que um consumidor é atendido pelo mapa que apresenta o neurônio com pesos sinápticos mais próximos das coordenadas do padrão de entrada correspondente ao consumidor, tal como definido para o cálculo da carga aproximada de um veículo.

O cálculo da variância das demandas é dado pela seguinte equação:

$$\text{Variância}(v) = \frac{1}{N_v} \cdot \sum_{n=1}^{N_v} (D_{vn} - \bar{D}_v)^2. \quad (4.9)$$

A variância deve ser controlada de modo a evitar uma distribuição inapropriada das demandas, o que poderia conduzir a veículos responsáveis por um conjunto de demandas predominantemente pequenas e veículos responsáveis por demandas predominantemente grandes. É um fato bem conhecido que baixas variâncias tendem a diminuir a distância total percorrida sempre que as capacidades dos veículos são similares. O cálculo do custo dos veículos requer um mapeamento do estado atual do veículo para um número real, refletindo a afinidade entre o veículo e o consumidor apresentado à rede neural. Esta é uma estimativa genérica e bastante útil, desde que neurônios pertencentes a mapas correspondentes a veículos sobrecarregados, ou cujo centróide encontra-se posicionado muito distante das coordenadas do consumidor, devem apresentar baixa ou mesmo nenhuma probabilidade de vencer o processo competitivo. Essa estratégia permite que veículos mais sobrecarregados sofram maiores penalidades que veículos menos sobrecarregados. Isto é implementado adotando-se uma abordagem nebulosa para guiar o processo de adaptação da rede neural. O custo de cada veículo será expresso por um valor na faixa entre 0 e 1.

As variáveis nebulosas de entrada e saída são expressas pelas seguintes variáveis lingüísticas:

- *Variância\_Relativa* = {Pequena, Média, Alta}.
- *Carga\_Suportada\_Relativa* = {Pequena, Média, Alta}.
- *Distância\_Centróide\_Relativa* = {Pequena, Média, Alta}.
- *Custo\_Veículo* = {Baixo, Médio, Alto, Muito Alto}.

As variáveis nebulosas de entrada e saída são combinadas de modo a compor regras nebulosas. As regras são representadas pela seguinte estrutura básica:

SE  $Variância\_Relativa$  é  $A_i$   $\wedge$   $Carga\_Suportada\_Relativa$  é  $B_j$   $\wedge$   
 $Distância\_Centróide\_Relativa$  é  $C_k$  ENTÃO  $Custo\_Veículo$  é  $S_{ijk}$ .

É utilizada a norma-t “*min*”, denotada por “ $\wedge$ ”, para modelar o conectivo “e”. Todas as funções de pertinência empregadas foram tomadas como sendo triangulares.

O universo de discurso de uma variável nebulosa de entrada varia de seu valor mínimo atual (limitante inferior) até seu valor máximo atual (limitante superior). Por exemplo, os limitantes inferior e superior para a variável *Variância\_Relativa* serão, respectivamente, a menor e a maior variância dentre as variâncias de todos os veículos. O mesmo procedimento é aplicado às variáveis nebulosas *Carga\_Suportada\_Relativa* e *Distância\_Centróide\_Relativa*.

Dependendo das restrições existentes no problema, as variáveis lingüísticas podem acarretar maior ou menor impacto sobre a variável nebulosa de saída. Por exemplo, para instâncias em que existe uma distância máxima que pode ser percorrida por cada veículo, a *Distância\_Centróide\_Relativa* deve produzir maior impacto nos custos do veículo do que se o problema apresentasse somente restrições de capacidade. Dessa forma, o algoritmo pode ser facilmente adaptado para lidar com diferentes tipos de problema.

O estado atual do mapa pode ativar diversas regras simultaneamente. Dessa forma, uma vez que a variável nebulosa de saída é obtida, um processo de *defuzzificação* deve ser realizado com o objetivo de produzir um único valor numérico. O método de *defuzzificação* adotado neste trabalho é o centro de área (Pedrycz & Gomide, 1998).

A Figura 4.7 mostra o universo de discurso para as variáveis nebulosas de entrada e de saída, respectivamente. A Tabela 4.1 apresenta as regras nebulosas utilizadas.

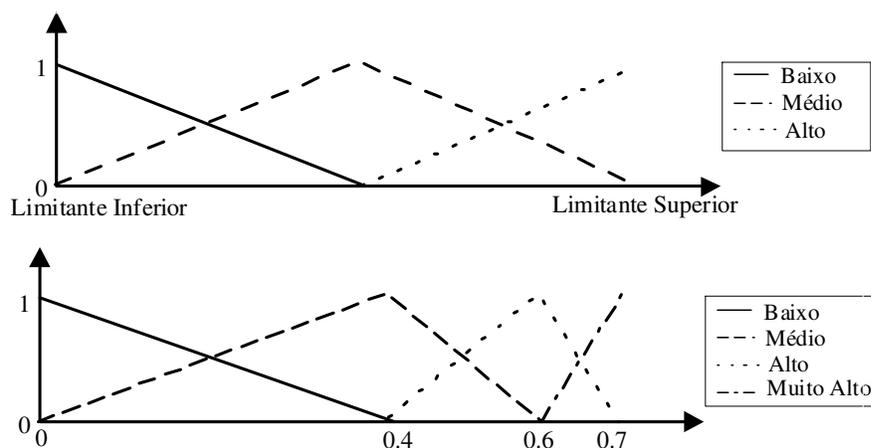


Figura 4.7. Variáveis nebulosas de entrada e saída.

Tabela 4.1. Base de regras nebulosas para o problema de roteamento de veículos capacitados.

Carga_Suportada_Relativa	Distância_Centróide_Relativa	Variância_Relativa	Custo_Veículo
Baixa	Baixa	Baixa	Baixo
Baixa	Baixa	Média	Baixo
Baixa	Baixa	Alta	Baixo
Baixa	Média	Baixa	Médio
Baixa	Média	Média	Baixo
Baixa	Média	Alta	Baixo
Baixa	Alta	Baixa	Alto
Baixa	Alta	Média	Alto
Baixa	Alta	Alta	Médio
Média	Baixa	Baixa	Alto
Média	Baixa	Média	Médio
Média	Baixa	Alta	Médio
Média	Média	Baixa	Alto
Média	Média	Média	Médio
Média	Média	Alta	Médio
Média	Alta	Baixa	Muito Alto
Média	Alta	Média	Alto
Média	Alta	Alta	Alto
Alta	Baixa	Baixa	Muito Alto
Alta	Baixa	Média	Alto
Alta	Baixa	Alta	Alto
Alta	Média	Baixa	Muito Alto
Alta	Média	Média	Muito Alto
Alta	Média	Alta	Alto
Alta	Alta	Baixa	Muito Alto
Alta	Alta	Média	Muito Alto
Alta	Alta	Alta	Muito Alto

#### *4.3.2.5 Política de Poda, Inserção e Inibição de Neurônios*

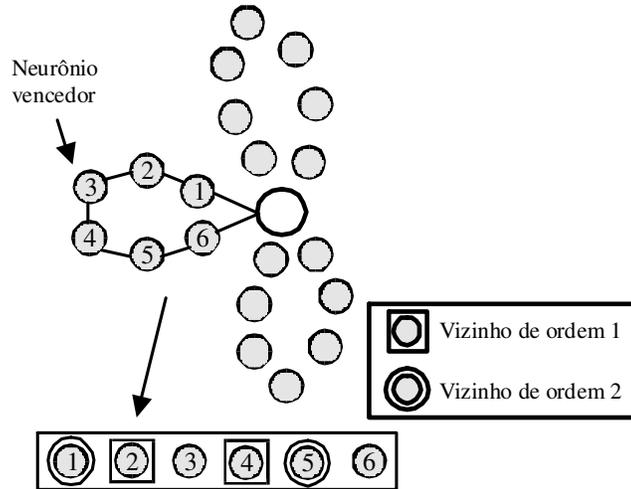
Conforme mencionado anteriormente, o algoritmo provê um controle automático do número de neurônios do mapa. O procedimento para inserção e poda de neurônios é similar àquele proposto pelo algoritmo KNIES (Kohonen Network Incorporating Explicit Statistics), introduzido por ARAS *et al.* (1999). Caso um determinado neurônio vença duas vezes o processo competitivo em uma mesma época, ele é inibido e um novo neurônio é inserido na rede sobre as mesmas coordenadas do vencedor e passa a ser o novo vencedor. Caso um neurônio não vença o processo competitivo por um determinado número de épocas consecutivas, ele será considerado inativo e eliminado da rede.

#### *4.3.2.6 O Processo Cooperativo*

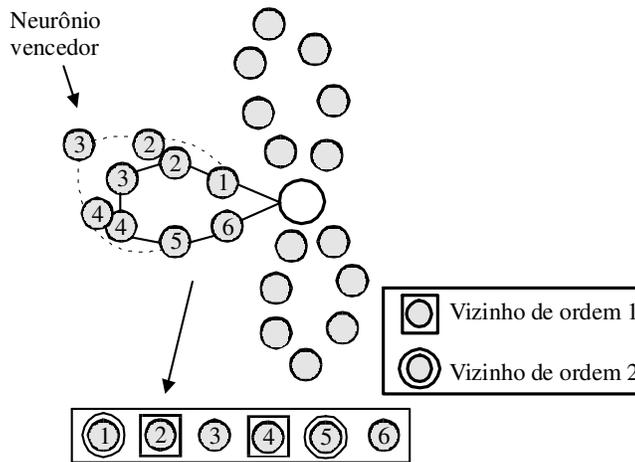
O processo cooperativo é inspirado em um mecanismo neurobiológico em que um neurônio que se encontra no estado ativo tende a afetar o estado de sua vizinhança. A vizinhança considerada neste trabalho é unidimensional, restrita ao anel em que o neurônio vencedor encontra-se posicionado. O número de vizinhos é variável de acordo com o estágio de auto-organização do mapa e com o número de neurônios presentes no mapa auto-organizável em questão.

A função de ativação de vizinhança é gaussiana e, ao início do processo iterativo, envolve cerca de 40% do número total de neurônios presentes no anel. A relação de vizinhança é não-circular, ou seja, o primeiro e o último neurônios de uma determinada rota (neurônios posicionados imediatamente ao lado do depósito) apresentam vizinhos somente ao lado oposto ao do depósito. A Figura 4.8 ilustra a relação de vizinhança considerada neste trabalho. O raio de vizinhança associado ao neurônio vencedor é dado por um percentual do número total de neurônios presentes no mapa associado ao veículo. No caso da implementação realizada neste trabalho, o tamanho da vizinhança utilizado é igual a 20% do número total de neurônios existentes no mapa correspondente. Este percentual é multiplicado por um fator que decresce conforme a carga efetiva se aproxima da carga aproximada, indicativo de que o processo de auto-organização do mapa encontra-se em seus estágios finais. Dado que a vizinhança é definida como um percentual do número total de neurônios presentes no mapa, ela

automaticamente tende a aumentar ou reduzir de acordo com a variação do número de neurônios, decorrente dos mecanismos de inserção e poda.



**Figura 4.8.** Vizinhança para o problema de roteamento de veículos capacitados.



**Figura 4.9.** Adaptação sináptica envolvendo vizinhança de ordem 1.

#### 4.3.2.7 O Processo de Adaptação Sináptica

A adaptação sináptica na época  $n+1$  é realizada de acordo com a seguinte fórmula:

$$w_j(n+1) = w_j(n) + \eta(n) \cdot h_{j,i(x)}(n) \cdot (x - w_j(n)) + \alpha(n) \cdot (w_{j+1}(n) - 2w_j(n) + w_{j-1}(n)), \quad (4.10)$$

onde  $\eta(n)$  é a taxa de aprendizagem,  $x$  é a coordenada do consumidor atual,  $\alpha(n)$  é um parâmetro de controle,  $w_j$  é o vetor de pesos sinápticos do neurônio  $j$  e  $w_{j+1}$  e  $w_{j-1}$  são os vetores de pesos dos neurônios vizinhos imediatos de  $j$ . O termo  $\alpha(n) \cdot (w_{j+1}(n) - 2w_j(n) + w_{j-1}(n))$  visa prover uma separação eficiente entre neurônios vizinhos, de modo que a relação de vizinhança mantenha uma correspondência topológica.

A taxa de aprendizagem  $\eta$  possui um valor inicial igual a 0,1 e decresce com  $n$ , na forma:

$$\eta(n) = \eta_0 \cdot \exp(-n / \tau), \quad (4.11)$$

onde  $n$  é a época atual e  $\tau$  é uma constante de tempo.

O segundo termo na equação, ponderado por  $\alpha(n)$ , é utilizado com o objetivo de criar uma força lateral entre nós vizinhos, permitindo dessa forma uma separação apropriada entre neurônios (MICHALEWICZ & FOGEL, 2002; MODARES *et al.*, 1999).

O parâmetro  $\alpha(n)$  assume um valor inicial igual a 0,35 e é atualizado de acordo com a equação:

$$\alpha(n) = 0,95 \cdot \alpha(n-1), \quad (4.12)$$

onde  $n$  é a época atual.

A Figura 4.9 ilustra o efeito de um passo do processo de adaptação sináptica no contexto do problema de roteamento de veículos e considerando vizinhança de ordem 1.

### 4.3.3 Resultados Computacionais

Esta seção apresenta os resultados experimentais para sete instâncias-padrão do problema de roteamento de veículos capacitados, para algumas variações do problema de roteamento de veículos capacitados e para o problema de múltiplos caixeiros viajantes. São também fornecidos resultados comparativos considerando as abordagens nebulosas (GOMES & VON ZUBEN 2001; 2002a; 2002b; 2002c; 2002d; 2002e) e não-nebulosa (MODARES *et al.*, 1999), avaliando o impacto da introdução das regras nebulosas no treinamento da rede neural. Os

resultados produzidos pelo sistema neuro-nebuloso foram utilizados como condição inicial para uma implementação de busca tabu, conduzindo a melhores resultados. Para mais detalhes acerca de algoritmos de busca tabu, ver Capítulo 2.

Na Seção 4.3.3.2 são consideradas três variações do problema de roteamento de veículos capacitados, com restrições adicionais envolvendo preferências dos consumidores por determinados veículos. Na Seção 4.3.3.3 são consideradas duas instâncias de grande porte do problema de múltiplos caixeiros viajantes.

#### 4.3.3.1 O Problema de Roteamento de Veículos Capacitados (CVRP)

Nesta seção são apresentados os resultados computacionais para um conjunto de sete instâncias do problema de roteamento de veículos capacitados, extraídas do conjunto de testes de Christofides e Eilon (CHRISTOFIDES & EILON, 1969).

Os dados de teste utilizados na simulação, bem como as melhores soluções conhecidas, encontram-se disponíveis em: <http://www.dca.fee.unicamp.br/projects/infobiosys/vrp>.

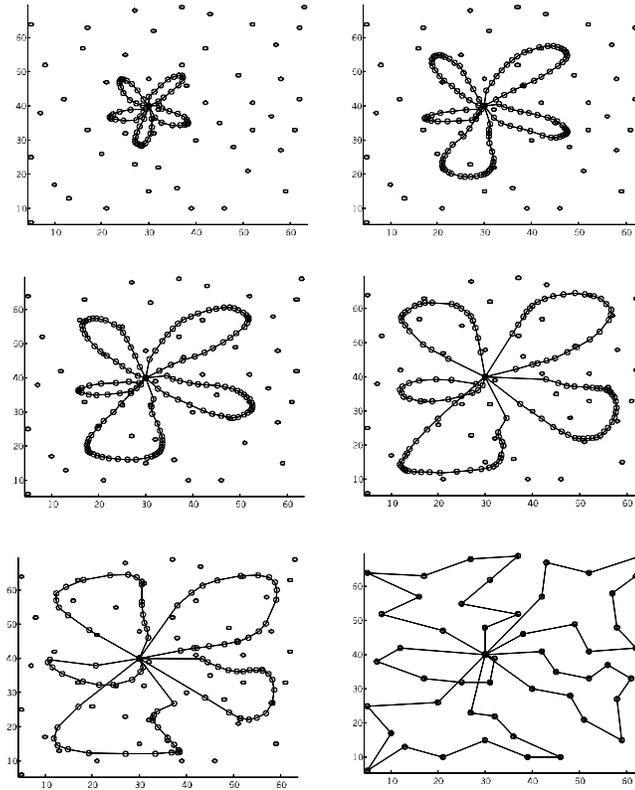
A base de regras nebulosas utilizada nesta tese é mostrada na Tabela 4.1 da Seção 4.3.2.2.

A Tabela 4.2 apresenta as características das instâncias tratadas neste capítulo. São apresentados o nome da instância, o número de consumidores associados, a menor quantidade de veículos (ou número de rotas) conhecida para a solução do problema, a capacidade de carga dos veículos e custo da melhor solução conhecida para cada instância.

**Tabela 4.2.** Identificação, número de consumidores, número ótimo de rotas, capacidade dos veículos e menor distância total conhecida para instâncias-padrão do problema de roteamento de veículos capacitados tratadas nesta tese.

Instância	# de consumidores	# de rotas	Capacidade do veículo	Melhor solução conhecida
E-n22-k4	21	4	6000	375
E-n33-k4	32	4	8000	835
E-n51-k5	50	5	160	521
E-n76-k8	75	8	180	735
E-n76-k10	75	10	140	832
E-n101-k8	100	8	200	817
E-n101-k14	100	14	112	1077

A Figura 4.10 apresenta seis estágios da execução do algoritmo neuro-nebuloso para a instância  $E-n51-k5$ : configuração inicial, quatro momentos distintos do processo de ordenação do mapa e solução final.



**Figura 4.10.** Evolução da solução para a instância E-n51-k5.

A Tabela 4.3 mostra os resultados obtidos para a abordagem neuro-nebulosa proposta nesta tese e para a abordagem baseada em mapas auto-organizáveis convencional ao longo de 50 execuções. Como esperado, os resultados obtidos a partir da abordagem neuro-nebulosa são superiores às soluções produzidas pela implementação da abordagem auto-organizável proposta em MODARES *et al.* (1999) e são um pouco inferiores às melhores soluções conhecidas. Observe que a abordagem neuro-nebulosa, quando comparada à abordagem baseada em mapas auto-organizáveis convencional, conduziu a soluções melhores para todas as instâncias, além de reduzir a violação da restrição de carga. Conforme mencionado anteriormente, devido à natureza da metodologia baseada em mapas auto-organizáveis, ao

final do processo auto-organizável podem ser geradas soluções ineficazes, apresentando pequenas violações das capacidades dos veículos. As violações de capacidade máxima são também mostradas na Tabela 4.3.

Um aspecto relevante reside no fato de que as melhores soluções conhecidas apresentadas na literatura não foram obtidas por um mesmo e único algoritmo de busca, enquanto aqui um mesmo algoritmo foi empregado junto às instâncias apresentadas.

**Tabela 4.3.** Comparação entre os resultados médios obtidos para o problema de roteamento de veículos capacitados empregando as abordagens neuro-nebulosa e convencional ao longo de 50 execuções.

Instância	Distância média total para a abordagem neuro-nebulosa	Violação média da capacidade para a abordagem neuro-nebulosa (%)	Distância média total para a abordagem neural convencional	Violação média da capacidade para a abordagem neural convencional (%)
E-n22-k4	383,51	0,00	384,30	0,00
E-n33-k4	873,74	0,00	968,70	0,09
E-n51-k5	525,74	0,00	542,76	0,40
E-n76-k8	793,00	2,15	876,57	3,00
E-n76-k10	873,68	3,05	887,00	5,20
E-n101-k8	876,57	0,35	943,00	0,58
E-n101-k14	1180,00	1,20	1341,00	3,70

As soluções produzidas pelo sistema neuro-nebuloso podem ser utilizadas como condição inicial de alta qualidade para estratégias mais dedicadas, potencialmente melhorando a qualidade das soluções finais. Neste trabalho, as soluções obtidas pelo sistema neuro-nebuloso foram utilizadas como condição inicial para uma implementação da *busca tabu* (GLOVER & LAGUNA, 1998). Os resultados experimentais encontram-se descritos na Tabela 4.4, que apresenta as melhores soluções encontradas pela busca tabu (TS) com condição inicial gerada pela heurística do vizinho mais próximo (NNH) e busca tabu com condição inicial gerada pelo sistema neuro-nebuloso proposto (FSOM). A Tabela 4.4 também mostra os resultados médios ao longo de 50 execuções de ambas as estratégias.

Com base nos resultados obtidos, pode-se concluir que o uso do sistema neuro-nebuloso para gerar soluções iniciais para a busca tabu levou a uma melhoria na qualidade das soluções

finais e conferiu maior estabilidade ao algoritmo da busca tabu, principalmente quando várias execuções são consideradas.

**Tabela 4.4.** Resultados para 50 execuções da busca tabu (TS) utilizando como condição inicial a solução provida pelo sistema neuro-nebuloso (FSOM) e pela heurística do vizinho mais próximo (NNH), incluindo média das 50 soluções, melhor solução obtida dentre as 50 execuções e tempo de execução.

Instância	TS e NNH: Solução média de 50 execuções	TS e NNH: melhor solução de 50 execuções	TS e NNH: tempo médio de execução (s)	TS e FSOM: solução média de 50 execuções	TS e FSOM: melhor solução de 50 execuções	TS e FSOM: tempo médio de execução (s)
E-n22-k4	375,28	375	2	375,28	375	3
E-n33-k4	839,25	835	4	837,67	835	3
E-n51-k5	533,63	521	12	530,45	521	12
E-n76-k10	873,83	861	9	855,47	835	30
E-n76-k8	774,96	737	15	756,57	738	50
E-n101-k8	842,59	824	68	839,20	821	150
E-n101-k14	1133,45	1094	39	1124,12	1093	70

#### 4.3.3.2 Problemas de Roteamento Estendidos (ECVRP- Extended Vehicle Routing Problems)

Nesta seção são apresentados os resultados computacionais para algumas variações do problema de roteamento de veículos capacitados, envolvendo restrições adicionais que expressam preferências dos consumidores por determinados veículos. As preferências foram aleatoriamente definidas. Esta seção visa ilustrar a flexibilidade da ferramenta e sua capacidade de adaptação a variantes do problema.

As instâncias 1 e 2 constituem modificações da instância *E-n51-k5* e a instância 3 é uma modificação da instância *E-n101-k8*.

A forma geral das restrições adicionais é: consumidores do tipo *i* não podem ser servidos por veículos do tipo *i*. Para resolver o ECVRP, um novo fator de penalização foi incluído no cálculo do neurônio vencedor. Se um neurônio específico pertencer ao mapa correspondente a um veículo rejeitado, este neurônio é penalizado adicionando-se à sua função de ativação um valor elevado.

Os resultados numéricos encontram-se detalhados na Tabela 4.5. A coluna rotulada *Restrições do Veículo* contém os veículos rejeitados pelos respectivos consumidores. Para a instância 1,

existem restrições associadas aos veículos 1, 2 e 3. Dessa forma, o veículo 1 não pode servir consumidores do tipo 1, o veículo 2 não pode servir consumidores do tipo 2 e o veículo 3 não pode servir consumidores do tipo 3. Os veículos 4 e 5 estão aptos a servir quaisquer tipos de consumidores. Para o tratamento dos casos apresentados a seguir, foi introduzida a seguinte modificação no treinamento: durante o processo de auto-organização do mapa os consumidores atraem somente neurônios correspondentes a veículos de tipos específicos requeridos pelo consumidor.

Os tempos computacionais foram obtidos em um computador com 64MB de memória RAM e clock de 500MHz.

A instância 2 envolve 5 restrições adicionais apresentando, portanto, restrições relacionadas a todos os veículos. A instância 3 apresenta restrições relacionadas a 4 veículos. As Figuras 4.11, 4.12 e 4.13 mostram a configuração final para as instâncias 1, 2 e 3, respectivamente. Note que, embora as instâncias 1, 2 e  $E-n51-k5$  envolvam restrições distintas, as três soluções apresentam várias arestas em comum, indicando que o algoritmo opera no sentido de preservar a minimização da distância total e a distribuição topológica dos padrões de entrada. As restrições relacionadas a veículos foram satisfeitas para todos os casos testados.

**Tabela 4.5.** Resultados para o problema de roteamento de veículos capacitados estendidos.

Instância	# de consumidores	# de rotas	Capacidade do veículo	Restrições dos veículos	Solução obtida (distância total)	Violação máxima da capacidade (%)	Tempo de execução (s)
1	50	5	160	1,2,3	668	1,8	12
2	50	5	160	1,2,3,4,5	647	2,5	10
3	100	8	200	1,2,3,4	904	1,0	53

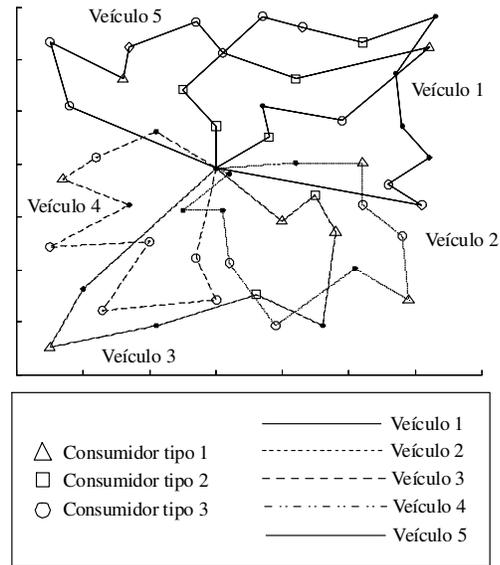


Figura 4.11. Solução para a instância 1 do ECVRP.

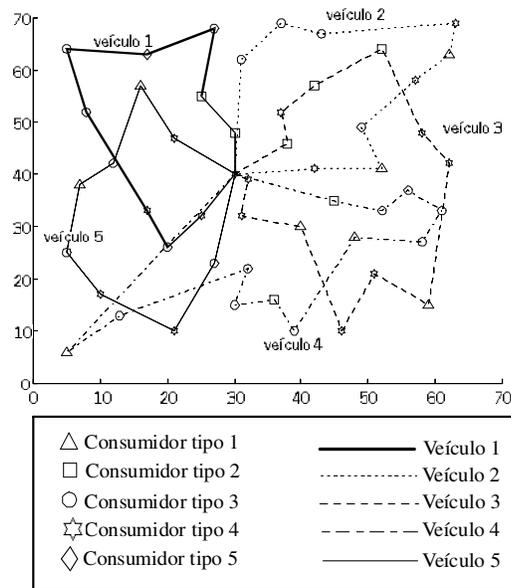
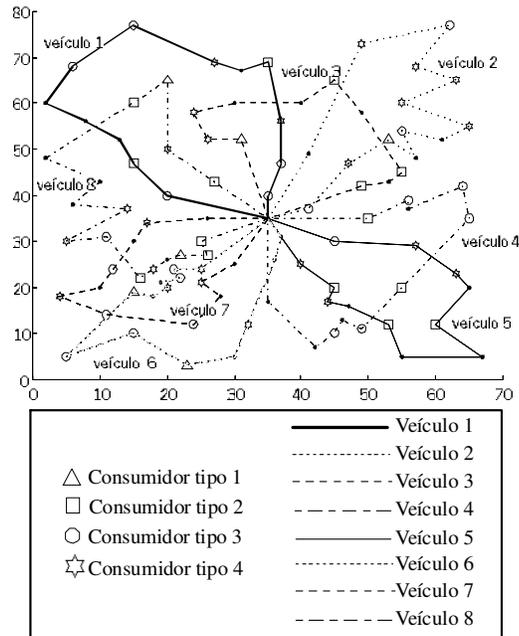


Figura 4.12. Solução para a instância 2 do ECVRP.



**Figura 4.13.** Solução para a instância 3 do ECVRP.

#### 4.3.3.3 O Problema de Múltiplos Caixeiros Viajantes (MTSP - Multiple Traveling Salesman Problem)

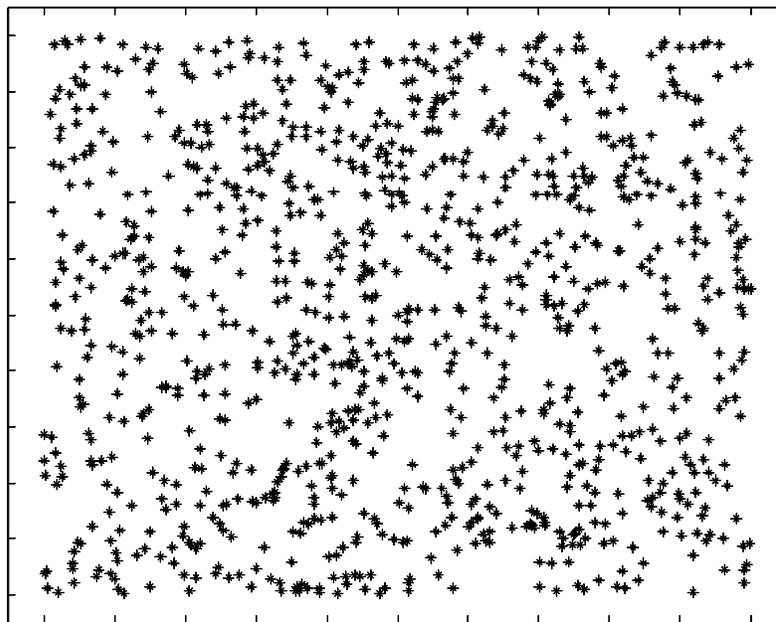
Esta seção apresenta os resultados computacionais obtidos via sistema neuro-nebuloso para o problema de múltiplos caixeiros viajantes. Neste caso, as regras nebulosas foram modificadas considerando os seguintes elementos:

- Tamanho total das rotas, medido tomando-se as distâncias entre neurônios adjacentes.
- Distância euclidiana entre o padrão de entrada atual e os neurônios.
- Distância euclidiana entre o padrão de entrada atual e os centróides dos mapas auto-organizáveis correspondentes aos veículos.

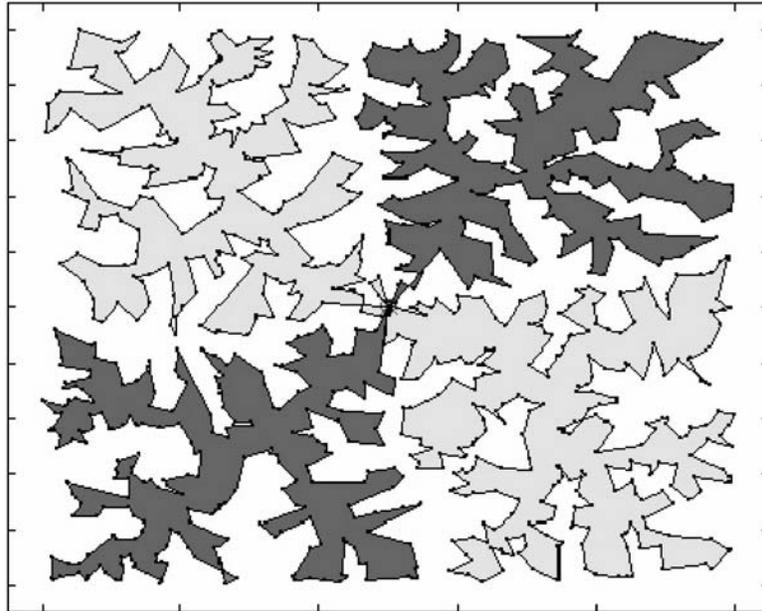
A Figura 4.14 mostra a distribuição de um conjunto de 1000 consumidores cujas coordenadas foram geradas aleatoriamente. A Figura 4.15 ilustra a solução para o problema. Neste caso, os veículos não apresentam restrições de capacidade e o objetivo é gerar rotas com tamanhos

equivalentes. Embora as soluções ótimas para estes casos não sejam conhecidas, uma simples inspeção visual permite atestar a validade da abordagem proposta para lidar com problemas de larga escala e a flexibilidade da técnica a modificações na função-objetivo. Para este caso, o algoritmo rodou em 7 minutos em um computador com 64MB de memória RAM e clock de 500MHz.

Note que a abordagem auto-organizável permite a incorporação de classes de restrições que seriam difíceis de serem modeladas matematicamente e tratadas de forma eficaz via outras abordagens. Por exemplo, pode-se definir que a rota 1 apresente algo próximo ao dobro do comprimento total da rota 2.



**Figura 4.14.** Coordenadas de 1000 consumidores gerados aleatoriamente com distribuição uniforme.



**Figura 4.15.** Resultados obtidos para um problema com 1000 consumidores, 4 veículos e sem restrições de capacidade.

#### **4.4 Síntese do Capítulo**

Neste capítulo foi apresentada uma abordagem neuro-nebulosa baseada em mapas auto-organizáveis de Kohonen projetada para lidar com problemas combinatórios envolvendo múltiplos critérios. Para demonstrar a eficiência do método proposto, o algoritmo foi aplicado a uma série de instâncias conhecidas na literatura.

Mapas auto-organizáveis podem ser diretamente aplicados ao problema do caixeiro viajante simétrico devido à natureza geométrica do problema. Entretanto, quando essa geometria é quebrada através da incorporação de restrições adicionais, a formulação de uma nova medida de similaridade entre padrões de entrada e neurônios mostra-se necessária. Nessa etapa, ajustes de parâmetros visando encontrar um compromisso entre qualidade da solução e satisfação das restrições torna-se uma tarefa difícil. Muitas vezes, os parâmetros de controle são altamente dependentes das características da instância a ser resolvida. Na abordagem proposta neste trabalho, a aplicação de mapas auto-organizáveis ao problema de roteamento de veículos não foi imediata e a concepção de uma estratégia competitiva empregando uma medida de

similaridade multicritério baseada em penalidades e recompensas foi necessária. Além disso, procedimentos para inicialização do mapa e técnicas adaptativas utilizando conceitos nebulosos para lidar com relaxação de restrições mostraram-se importantes para o bom desempenho e simplicidade do algoritmo.

Na Seção 4.3.3 foi apresentada uma comparação entre as abordagens auto-organizáveis nebulosa e não-nebulosa e, a partir dos resultados obtidos, conclui-se que a incorporação do treinamento empregando conceitos de inferência nebulosa reduziu drasticamente a necessidade de ajustes de parâmetros e melhorou significativamente o desempenho do algoritmo. Para cada classe de problemas abordada nas simulações, todos os parâmetros do algoritmo foram mantidos, não importando o número de consumidores e o perfil de demandas.

Considerando que neurônios que raramente vencem o processo competitivo ou que vencem com muita frequência podem levar a um processo ineficiente de ordenação do mapa, uma vantagem do método proposto neste trabalho sobre as abordagens convencionais baseadas em auto-organização é o controle automático do número de neurônios.

Uma outra característica importante do método proposto é sua flexibilidade quanto à adaptação a outras classes de problemas de roteamento através da modificação das regras nebulosas ou de mecanismos do treinamento. Com o intuito de explorar a flexibilidade da função-objetivo, o sistema neuro-nebuloso foi aplicado a algumas variantes do problema de roteamento de veículos.

As principais vantagens da abordagem neuro-nebuloso são:

- ajuste automático dos parâmetros do algoritmo;
- viabilidade de aplicação a problemas de larga escala;
- flexibilidade da função-objetivo.

Dentre as principais contribuições deste capítulo, destacam-se:

1. extensão do método proposto por ANGÉNIOL *et al.* (1988) para o problema de roteamento de veículos (múltiplos mapas auto-organizáveis sendo treinados simultaneamente);
2. definição de medidas de similaridade multicritério;
3. aplicação de conceitos nebulosos ao treinamento de mapas auto-organizáveis;
4. implementação de um processo adaptativo de relaxação de restrições sintetizado através de um conjunto de regras nebulosas;
5. extração de características estatísticas do problema;
6. produção de condições iniciais de alta qualidade para estratégias de busca mais dedicadas, como busca tabu.

Todas as implementações envolveram algoritmo de treinamento em modo seqüencial.

Embora todas as instâncias testadas tenham envolvido veículos com capacidades iguais, o algoritmo pode ser diretamente adaptado a frotas de veículos com capacidades distintas.

Quanto ao tempo de execução, as abordagens auto-organizadas não sofrem um acréscimo preocupante de tempo de execução com um aumento do tamanho do problema, ou seja, com um aumento no número de cidades ou consumidores.

# CAPÍTULO 5

## SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

---

Os sistemas imunológicos artificiais têm desempenhado um importante papel no campo da inteligência computacional. Diversos modelos imuno-inspirados vêm sendo propostos e aplicados aos mais variados contextos, tais como agrupamento de dados, otimização contínua e discreta, robótica e detecção de anomalias. O sucesso dessa abordagem provém justamente de peculiaridades de sua fonte de inspiração: mecanismos sofisticados de manutenção de diversidade e auto-regulação do número de células e outros parâmetros adaptativos. Neste capítulo são apresentados os fundamentos dos sistemas imunológicos artificiais (SIA) e são descritas algumas das principais ferramentas computacionais da engenharia imunológica: CLONALG, ABNET, aiNet e opt-aiNet.

### 5.1 Introdução

Os sistemas imunológicos artificiais constituem uma nova abordagem na área da inteligência computacional, e baseiam-se em modelos e princípios oriundos da imunologia. Assim como outras meta-heurísticas, seu principal objetivo é o tratamento de problemas de elevada complexidade, cujas soluções não podem ser obtidas através de métodos convencionais, tais como otimização linear, não-linear ou programação dinâmica.

Recentemente, sistemas imunológicos artificiais têm se firmado como uma nova abordagem no campo da computação inspirada na biologia (DE CASTRO & TIMMIS, 2002; DE CASTRO & VON ZUBEN, 1999; 2000; 2004).

Os principais conceitos biológicos que fundamentam os sistemas imunológicos artificiais são:

- maturação de afinidade;

- seleção clonal;
- teoria da rede imunológica.

Existem alguns modelos de rede imunológica artificial propostos na literatura, tais como ABNET, aiNet, opt-aiNet (DE CASTRO & TIMMIS, 2002) e copt-aiNet (GOMES *et al.*, 2003), destinadas a variadas aplicações. Embora haja diferenças conceituais entre estes modelos, em todos eles basicamente existe uma população de antígenos a ser reconhecida por um repertório de anticorpos sujeitos a alterações genéticas e a algum grau de inter-relacionamento. As seções que se seguem oferecem uma descrição mais detalhada acerca dos principais modelos.

Boa parte dos sistemas imunológicos artificiais opera sobre uma população de soluções candidatas, considerando particularidades do problema a ser resolvido através de procedimentos de busca local. Essa busca local pode ser diretamente associada ao mecanismo de maturação dos anticorpos e ao sistema imunológico adaptativo.

Uma distinção fundamental entre sistemas imunológicos artificiais e algoritmos evolutivos reside no fato de que algoritmos evolutivos visam explorar o espaço de busca de modo a obter uma solução de alta qualidade dentre uma população de soluções. Com esse objetivo, comumente são adotados processos seletivos elitistas, conduzindo a um problema inerente à maioria dos algoritmos evolutivos: a perda de diversidade da população. Várias estratégias, oriundas de fontes de inspiração diversas, têm sido propostas para suprir esta deficiência, tais como *modelos de ilhas* (WHITLEY *et al.*, 1999) e *nichos* (MILLER & SHAW, 1996). Além disso, há modelos de sistemas imunológicos artificiais, como aqueles inspirados na teoria da rede, que se diferenciam substancialmente dos algoritmos evolutivos, tanto no que se refere à arquitetura do sistema quanto a sua dinâmica. Os sistemas imunológicos artificiais constituem sistemas adaptativos de processamento de informação que operam de forma paralela, com capacidade de auto-organização, permitindo, em muitos casos, definição automática do número de indivíduos na população, critério de parada, taxas de mutação, reprodução, e seleção de quais indivíduos deverão permanecer na população. Apresentam mecanismos sofisticados de identidade e proteção, além de promoverem a manutenção da diversidade da população. Isso conduz a uma habilidade intrínseca de localizar e preservar múltiplas soluções

que são ótimos locais e que podem diferir significativamente em sua qualidade relativa ao ótimo global. A teoria da rede imunológica, que envolve um conjunto de agentes autônomos capazes de interagir uns com os outros e com o ambiente, desempenha um papel fundamental na manutenção da diversidade. A teoria da rede imunológica está de acordo com a *teoria da autopoiese* (auto-criação), proposta na obra de MATURANA & VARELA (1980), atualmente aceita em diversos segmentos da biologia e epistemologia. Assim como outros sistemas vivos, o sistema imunológico atua com base em suas representações internas e nas relações internas dos elementos que o compõem, sejam estas resultantes de perturbações externas ou da própria dinâmica interna, caracterizando o sistema imunológico como um sistema auto-organizável.

O domínio de aplicação de sistemas imunológicos artificiais é vasto, envolvendo diversas áreas (DASGUPTA, 1999; DE CASTRO & TIMMIS, 2002), tais como:

- Reconhecimento de padrões (HUNT & COOKE, 1996).
- Segurança computacional (BALTHROP *et al.*, 2004).
- Robótica (ISHIGURO *et al.*, 1995; MICHELAN, 2003).
- Otimização (HAJELA & LEE, 1996; HAJELA *et al.*, 1997; ENDOH *et al.*, 1998; FUKUDA *et al.*, 1999; DE CASTRO & VON ZUBEN, 1999; DE CASTRO & TIMMIS, 2002; GOMES *et al.*, 2003).
- Controle (BERSINI, 1991).
- Mineração de dados (HUNT *et al.*, 1996).

## **5.2 Conceitos Básicos**

### **5.2.1 Terminologia**

A correspondência entre elementos do sistema imunológico biológico e artificial depende da aplicação e do tipo de algoritmo imunológico implementado. Em um modelo de rede imunológica, denominado aiNet, descrito em detalhes na Seção 5.5, o conjunto de padrões a

serem agrupados (clusterizados) representam antígenos **Ag**, enquanto os protótipos que efetuam a clusterização representam anticorpos **Ab**. A rede é gerenciada com base na similaridade entre seus elementos (**Ab-Ab**), e entre eles e os antígenos (**Ag-Ab**).

A opt-aiNet, descrita na Seção 5.6, constitui uma variação da aiNet destinada à otimização de funções multimodais. Cada anticorpo presente na rede representa uma solução candidata para o problema. Os antígenos não são expressos na rede de forma explícita, tal como ocorre com a aiNet, mas são denotados por uma função-objetivo a ser otimizada. Dessa forma, a afinidade de cada anticorpo da rede é medida pela função-objetivo. A detecção de anticorpos auto-reativos, ou seja, anticorpos que interagem, se dá através de um critério de distância entre soluções, como a distância euclidiana.

### **5.2.2 O Espaço de Formas**

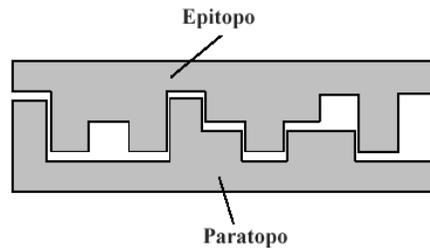
O espaço de formas refere-se a um formalismo proposto para permitir uma representação matemática de antígenos e anticorpos, permitindo assim quantificar seu grau de interação. Conforme descrito no Apêndice B, os anticorpos reconhecem antígenos através de regiões específicas denominadas *paratopos* e *epitopos*. A ligação entre epitopos e paratopos se dá através de complementaridade química e física. A Figura 5.1 ilustra o processo de reconhecimento via complementaridade. Quanto maior o grau de complementaridade, maior o nível de reconhecimento, também denominado afinidade.

Se o paratopo e o epitopo não são totalmente complementares, as duas moléculas ainda podem se ligar, mas com uma afinidade menor. Isso sugere que cada paratopo interage com todos os epitopos cujo complemento esteja em uma região específica de sua vizinhança, constituindo uma *região de reconhecimento* definida por um determinado limiar de afinidade.

Devido aos anticorpos serem capazes de reconhecer todos os epitopos situados em uma região e de um antígeno poder possuir diferentes epitopos, um número finito de anticorpos pode reconhecer um número praticamente infinito de antígenos.

Sob o ponto de vista computacional, visando alcançar maior simplicidade na implementação e interpretação de um sistema imunológico artificial, a ligação entre antígenos e anticorpos se dá

por similaridade, e não por complementaridade. A codificação adotada para representar anticorpos e antígenos exerce uma grande influência sobre o critério de distância a ser utilizado para calcular a afinidade. A afinidade entre antígenos e anticorpos e entre anticorpos e anticorpos é definida por um critério de distância apropriado, como a distância euclidiana, a distância de Hamming, ou então a distância de Manhattan.



**Figura 5.1.** O espaço de formas e reconhecimento parcial via complementaridade.

### **5.3 CLONALG (Clonal Selection Algorithm): Implementação Computacional do Princípio da Seleção Clonal**

#### **5.3.1 Aspectos Gerais**

Nesta seção será apresentado o algoritmo CLONALG (*CLONal Selection ALGORITHM*), proposto por DE CASTRO & VON ZUBEN (2000).

O sistema imunológico desempenha um importante papel na proteção do organismo contra agentes causadores de doenças. Para prover proteção, o sistema imunológico conta com diversos órgãos, células e moléculas. A maior parte das células imunológicas apresenta receptores em sua superfície, denominados anticorpos, permitindo o reconhecimento e a ligação aos agentes patogênicos. O processo de reconhecimento depende do nível de afinidade entre o antígeno e o anticorpo, de modo que, quanto maior a afinidade, maior o grau de reconhecimento. A ligação entre o antígeno e o anticorpo dispara uma resposta imunológica, sinalizando a outras células que o antígeno deve ser neutralizado.

A idéia básica por trás do processo de seleção clonal é que somente células imunológicas capazes de reconhecer o antígeno apresentado irão se proliferar, sendo que a taxa de proliferação de cada célula é dependente de sua afinidade com o antígeno. A teoria da seleção clonal define que, quando um antígeno invade o organismo, anticorpos que reconhecem o antígeno se proliferam, gerando novas células denominadas clones. Quanto maior a afinidade entre um anticorpo e o antígeno, maior o número de clones gerados. Durante a reprodução, os clones são submetidos a um processo de mutação a taxas inversamente proporcionais à afinidade entre o clone e o antígeno: quanto maior a afinidade, menor a taxa de mutação e vice-versa. Os clones mutados que apresentam elevada afinidade em relação ao antígeno são selecionados via processo de seleção natural para se tornarem células de memória, promovendo respostas futuras mais rápidas e eficientes a antígenos similares.

A seleção clonal desempenha um papel fundamental no processo de aprendizagem e memória, promovendo, a cada encontro com um mesmo antígeno ou com outros antígenos similares, a geração de indivíduos melhor adaptados.

O CLONALG foi inicialmente proposto para aplicação a problemas de reconhecimento de padrões, nos quais uma população de antígenos deve ser reconhecida por uma população de anticorpos. Posteriormente, o algoritmo foi estendido a problemas de otimização.

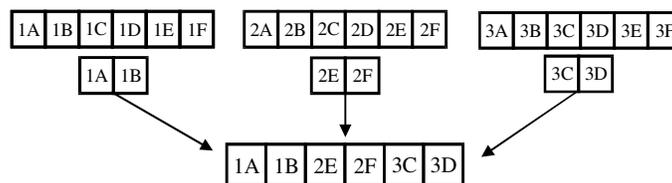
### 5.3.2 Notação

A seguir é apresentada a notação empregada pelo algoritmo CLONALG:

- **Ab**: repertório de anticorpos disponíveis;
- $\mathbf{Ab}_{\{r\}}$ : conjunto de anticorpos da população que não são de memória ( $\mathbf{Ab} - \mathbf{Ab}_{\{m\}}$ );
- $\mathbf{Ab}_{\{i\}}^j$ : conjunto contendo os  $i$  anticorpos de maior afinidade ao antígeno  $j$ ;
- **Ag**: população antigênica;
- $\mathbf{C}^j$ : conjunto de clones associados ao antígeno  $j$ ;
- $f_{i,j}$ : afinidade do anticorpo  $\mathbf{Ab}_i$  ao antígeno  $\mathbf{Ag}_j$ .

### 5.3.3 Descrição do CLONALG

O princípio da seleção clonal estabelece a idéia de que somente os anticorpos capazes de reconhecer antígenos se proliferam, passando por um processo de forte pressão seletiva. Durante a reprodução, as células-clone são submetidas a um processo de hipermutação somática aplicada a elevadas taxas, denominado maturação de afinidade. A maturação de afinidade, juntamente com o processo de pressão seletiva, resulta no aumento da afinidade entre o anticorpo e o antígeno que o selecionou. Ao final do processo de seleção clonal e pressão seletiva, os clones mutados tendem a apresentar maior afinidade ao antígeno do que as células-pai, sendo portanto mais bem adaptados ao antígeno. Esses indivíduos melhor adaptados darão origem a células de memória mais eficientes. Durante a reprodução celular, as células-filhas (clones) sofrem um processo de mutação somática seguido de uma forte pressão seletiva. A taxa de mutação é inversamente proporcional à afinidade da célula. Dessa forma, células com elevada afinidade sofrerão mutação a taxas menores do que células de baixa afinidade. Durante o processo de reprodução e maturação das células, o CLONALG emprega o *modelo da medula óssea*. Todas as células sangüíneas, incluindo as células imunológicas, são geradas na medula óssea (para mais detalhes, consultar Apêndice B). Na medula óssea, genes provenientes de diversas bibliotecas gênicas são concatenados de forma a produzir uma única célula. A Figura 5.2 oferece uma visão pictórica do processo de geração de células via bibliotecas gênicas.



**Figura 5.2.** Conceito de bibliotecas gênicas, com um novo indivíduo gerado através da recombinação de três elementos da biblioteca gênica.

### 5.3.4 O Algoritmo

O CLONALG apresenta duas versões: a primeira destina-se a problemas de reconhecimento de padrões, enquanto a segunda foi projetada para problemas de otimização. O pseudocódigo

para a versão do CLONALG destinada à solução de problemas de reconhecimento de padrões é apresentado no Algoritmo 5.1.

Como principais entradas para o CLONALG, têm-se:

- População **Ag** composta por  $M$  antígenos de comprimento  $L$ .
- Quantidade  $gen$  de gerações a serem executadas (critério de parada).
- Número  $n$  de anticorpos a serem selecionados para clonagem.
- Fator multiplicativo  $\beta$  usado na definição da quantidade de clones.
- Quantidade  $d$  de anticorpos com baixa afinidade  $f$ , que serão substituídos.

**Saída:**

- Matriz **Ab** de anticorpos de memória.
- Afinidade total **F** entre os anticorpos de memória **Ab** e a população de antígenos **Ag**. A matriz **F** é composta pelos valores de afinidade  $f_{i,j}$  entre todos os anticorpos  $i$  obtidos ao final do algoritmo e todos os antígenos  $j$ .

O mesmo Algoritmo 5.1 pode ser empregado para a versão de otimização de funções mediante pequenas modificações:

- Diferentemente da versão para reconhecimento de padrões, na versão para otimização não existe uma população de antígenos a serem reconhecidos, mas sim uma função-objetivo a ser otimizada. Assim, a afinidade de um anticorpo passa a corresponder ao valor da função-objetivo avaliada para o anticorpo. Caso o problema de otimização seja multimodal, com o objetivo de encontrar múltiplos ótimos, a taxa de reprodução de anticorpos proporcional à afinidade não necessariamente se aplica, pois quer-se evitar que os anticorpos concentrem-se somente ao redor do melhor ótimo local encontrado.
- Cada anticorpo representa o valor codificado de um argumento da função a ser otimizada.
- Codificação dos anticorpos: seqüências binárias codificam os valores reais para as variáveis da função a ser otimizada.

**Procedimento [Ab, F]=CLONALG (Ag,  $\beta$ ,  $d$ ,  $n$ ,  $gen$ )**

1. Gerar aleatoriamente população inicial de anticorpos **Ab**
2.  $f_{\{Ag\}}^* = -1$
3. Repita
  - 3.2. Para cada antígeno **Ag<sub>j</sub>**,  $j = 1, \dots, M$ , faça:
    - 3.2.1. Para cada anticorpo **Ab<sub>i</sub>**,  $i = 1, \dots, N$ , faça
      - 3.2.1.1. Calcular a afinidade  $f_{i,j}$  do anticorpo **Ab<sub>i</sub>** ao antígeno **Ag<sub>j</sub>**
    - 3.2.2. Fim Para
    - 3.2.3. Selecionar o conjunto **Ab<sub>{n}</sub><sup>j</sup>** de  $n$  anticorpos de maior afinidade ao antígeno **Ag<sub>j</sub>**
    - 3.2.4. Produzir  $nc$  clones a partir de elementos de **Ab<sub>{n}</sub><sup>j</sup>** a taxas proporcionais à afinidade, de acordo com o fator multiplicativo  $\beta$ , gerando o conjunto de clones **C<sup>j</sup>**
    - 3.2.5. Submeter os anticorpos de **C<sup>j</sup>** a mutação a taxas inversamente proporcionais à afinidade, gerando o conjunto **C<sup>j\*</sup>**
    - 3.2.6. Determinar o vetor de afinidades  $f_{\{C^{j*}\},j}$  entre **Ag<sub>j</sub>** e os anticorpos de **C<sup>j\*</sup>**
    - 3.2.7. Selecione de **C<sup>j\*</sup>** o anticorpo **Ab<sub>k</sub><sup>j</sup>** de maior afinidade  $f_{k,j}$
    - 3.2.8. Se  $f_{k,j} > f_j^*$  então
      - 3.2.8.1.  $f_j^* = f_{k,j}$
    - 3.2.9. Fim Se
  - 3.3. Substituir  $d$  elementos de **Ab<sub>{r}</sub>** por elementos de **C<sup>j\*</sup>** ou por elementos gerados a partir do modelo de medula óssea
  - 3.4. Fim Para
4. Até atingir  $gen$  gerações

**Fim Procedimento****Algoritmo 5.1. CLONALG.**

### 5.3.4.1 Codificação dos anticorpos na versão para otimização

Na versão para otimização proposta em DE CASTRO & VONZUBEN (1999), os valores reais são codificados como seqüências binárias. Primeiramente escolhe-se o número de bits  $L$  a serem utilizados na representação. Considere, por exemplo, uma função a ser otimizada cujas variáveis estejam definidas no intervalo  $[i_1, i_2]$ . Dada uma representação binária qualquer de comprimento  $L$ , a conversão da representação binária que codifica o anticorpo para decimal se dá da seguinte forma:

- converter a seqüência binária  $\langle b_{L-1}, b_{L-2}, \dots, b_0 \rangle$  para decimal, obtendo o valor  $x'$ , de acordo com a Equação 5.1:

$$x' = \sum_{n=0}^{L-1} 2^n \cdot b_n. \quad (5.1)$$

- em seguida, obter o valor de acordo com a seguinte fórmula:

$$x = i_1 + x' \cdot \frac{i_2 - i_1}{2^L - 1}. \quad (5.2)$$

Assim,  $x$  constitui o valor a ser avaliado na função-objetivo, supondo uma única variável como argumento. Para múltiplas variáveis, associam-se  $L$  bits a cada uma. Como  $2^L - 1$  é o máximo valor em base decimal obtido de uma representação binária utilizando-se  $L$  bits, o termo  $\frac{x'}{2^L - 1}$  resulta em um valor entre 0 e 1. Multiplicando-se o termo  $\frac{x'}{2^L - 1}$  por  $i_2 - i_1$ , que é o tamanho do intervalo, e somando-se  $i_1$  obtém-se um valor real entre  $i_1$  e  $i_2$ . Modificando-se os bits da representação binária, modifica-se o valor real obtido, permitindo assim a exploração do espaço de busca através da aplicação de procedimentos de mutação (inversão de  $n$  bits na seqüência binária). O número de casas decimais depende do número de bits utilizados na representação.

O procedimento de mutação envolve a complementação de bits na seqüência que codifica o anticorpo, o que gera diferentes valores dentro do intervalo no qual estão definidas as variáveis da função.

Para mais detalhes acerca da aplicação do CLONALG a problemas de otimização, consultar DE CASTRO & VON ZUBEN (1999; 2002).

### **5.3.5 Resultados do CLONALG**

A seguir serão apresentados alguns resultados computacionais obtidos a partir da aplicação do CLONALG a três funções multimodais.

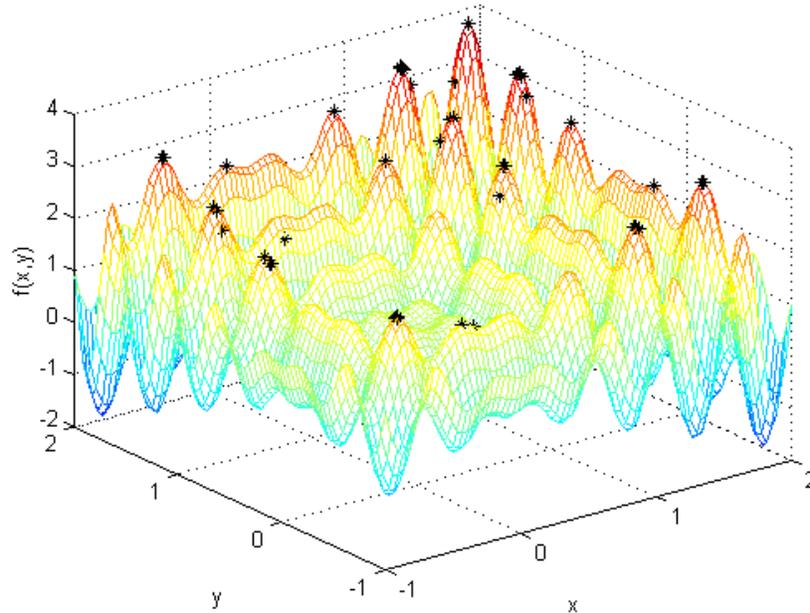
#### *5.3.5.1 Multifunção*

A multifunção é dada por:

$$g(x,y) = x.\sin(4\pi x) - y.\sin(4\pi y + \pi) + 1, \quad (5.3)$$

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

A Figura 5.3 apresenta os múltiplos pontos de ótimo local obtidos via CLONALG. Ao todo, o método localizou e preservou na memória dezoito picos, incluindo o pico correspondente ao ótimo global.



**Figura 5.3.** Resultados obtidos pelo CLONALG para a multifunção (figura extraída de DE CASTRO & TIMMIS, 2002).

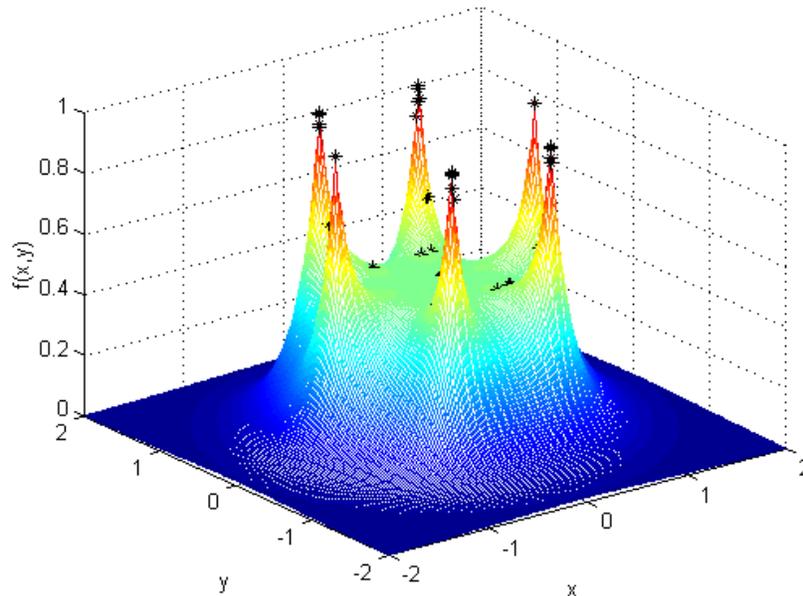
### 5.3.5.2 Função Raízes

A função raízes é dada pela seguinte equação:

$$g(z) = \frac{1}{1 + |z^6 - 1|}, \quad (5.4)$$

onde  $z \in \mathbf{C}$ ,  $z = x + iy$ ,  $x, y \in [-2, 2]$ .

A Figura 5.4 apresenta os ótimos locais encontrados pelo CLONALG. Observe que o CLONALG foi capaz de detectar todos os picos da função, mantendo vários indivíduos em cada pico.



**Figura 5.4.** Resultados obtidos pelo CLONALG para a função raízes (figura extraída de DE CASTRO & TIMMIS, 2002).

### 5.3.5.3 Função de Schaffer

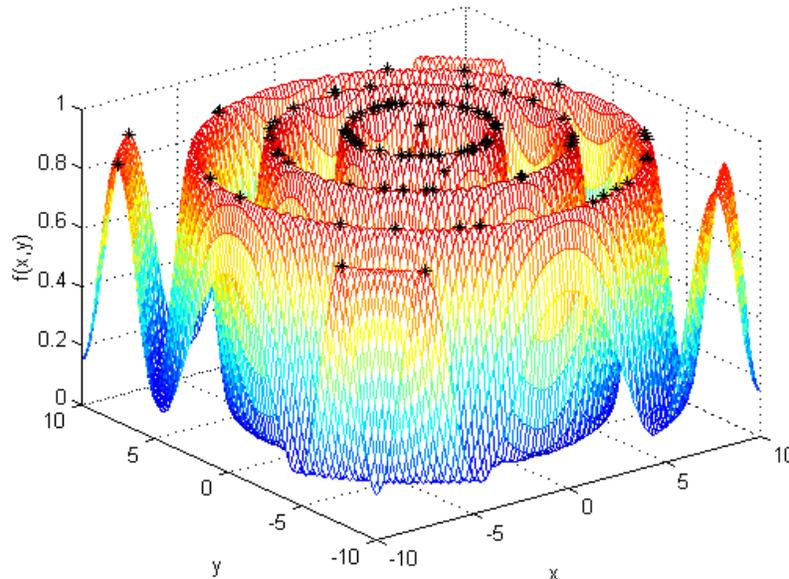
A função de Schaffer é dada por:

$$g(z) = 0,5 + \frac{\sin^2\left(\sqrt{x^2 + y^2}\right) - 0,5}{\left(1 + 0,001 \cdot (x^2 + y^2)\right)}, \quad (5.5)$$

onde  $x, y \in [-10, 10]$ .

A função de Schaffer possui um único ponto de ótimo global na origem e infinitos ótimos locais. O ótimo difere em apenas  $10^{-3}$  do ótimo local mais próximo.

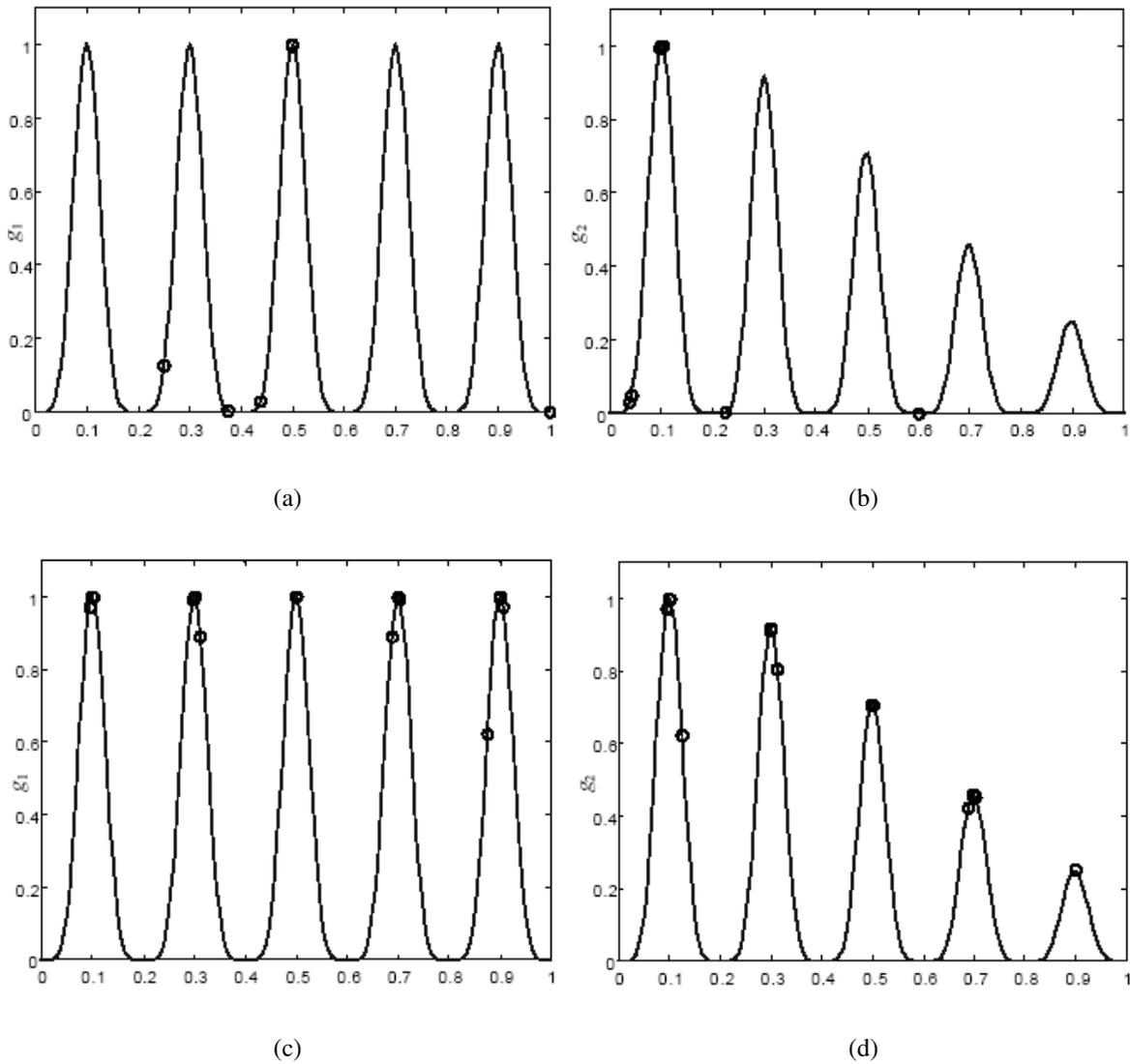
A Figura 5.5 apresenta alguns resultados obtidos pelo CLONALG. Observe que o CLONALG foi capaz de detectar o ponto de ótimo global e de explorar eficientemente o espaço de busca, apresentando células distribuídas sobre toda a superfície em que existem ótimos locais, embora não corresponda a uma distribuição uniforme.



**Figura 5.5.** Resultados obtidos pelo CLONALG para a função de Schaffer (figura extraída de DE CASTRO & TIMMIS, 2002).

#### 5.3.5.4 Comparação com outras abordagens

Em DE CASTRO (2001), são apresentados resultados comparativos envolvendo o algoritmo de seleção clonal e uma implementação de um algoritmo genético convencional. Conforme pode ser observado na Figura 5.6, para as funções  $f(x) = (\text{sen}(5 \cdot \pi \cdot x))^6$  (Figura 5.6 (a) e (c)) e  $g(x) = 2^{(-2 \cdot ((x-0,1)/0,9))^2} \cdot \text{sen}(5 \cdot \pi \cdot x)$  (Figura 5.6 (b) e (d)), o algoritmo genético foi capaz de encontrar o ótimo global e sua população conservou apenas um dos pontos de ótimo local em uma situação em que há múltiplos ótimos locais cujo valor coincide com o ótimo global (Figura 5.6 (a) e (b)), enquanto o CLONALG manteve uma população muito mais diversificada, localizando todos os pontos de ótimo local (Figura 5.6 (c) e (d)). Esta propriedade torna o CLONALG particularmente interessante para aplicações multimodais, em que se desejam obter múltiplas soluções.



**Figura 5.6.** (a) e (b): soluções obtidas por algoritmo genético convencional; (c) e (d) soluções obtidas pelo CLONALG (figura extraída de DE CASTRO, 2001).

## 5.4 ABNET (Antibody Network)

### 5.4.1 Aspectos Gerais

Nesta seção é apresentada a ABNET (*AntiBody NETwork*), baseada em diversos princípios imunológicos. A ABNET (DE CASTRO, 1999) emprega conceitos fundamentais da imunologia, tais como auto-organização, memória imunológica, aprendizagem, maturação de afinidade e reconhecimento de padrões. A proposta da rede de anticorpos é definir automaticamente um repertório de anticorpos explorando o princípio da seleção clonal. A ABNET é uma rede binária que segue o paradigma de aprendizado competitivo e baseia-se no processo de seleção clonal do sistema imunológico biológico. Como o objetivo é definir um conjunto adequado de anticorpos, os quais correspondem aos nós da rede, a ABNET envolve um modelo construtivo, que incorpora mecanismos de inserção e poda de células.

Assim, as principais características da rede de anticorpos ABNET são:

- arquitetura construtiva, baseada no princípio da seleção clonal;
- poda de nós, representando a morte de anticorpos que não são estimulados;
- conexões entre nós denotadas por seqüências binárias;
- aprendizado competitivo.

As seções seguintes fornecerão detalhes acerca da operação da ABNET, apresentando um pseudocódigo com as principais etapas.

### 5.4.2 Notação

A seguir é apresentada a notação empregada pelo algoritmo da ABNET.

- **Ab**: repertório de anticorpos disponíveis;

- $\mathbf{Ab}_i$ :  $i$ -ésimo anticorpo do repertório  $\mathbf{Ab}$ . O anticorpo  $i$  está associado a um vetor de pesos  $\mathbf{w}_i$ , que liga o nó  $i$  a todas as entradas da rede. Um anticorpo é também referido como célula, nó, unidade ou elemento da rede;
- $\mathbf{Ag}$ : população antigênica;
- $\mathbf{Ag}_j$ :  $j$ -ésimo antígeno da população antigênica  $\mathbf{Ag}$ ;
- $\tau_i$ : nível de concentração do anticorpo  $\mathbf{Ab}_i$ , ou seja, o número de antígenos reconhecidos por ele;
- $f_{i,j}$ : afinidade do anticorpo  $\mathbf{Ab}_i$  ao antígeno  $\mathbf{Ag}_j$ .

#### 5.4.3 Descrição da ABNET e Algoritmo

A ABNET foi desenvolvida para aplicações de reconhecimento de padrões binários, ou seja, que possam ser representados por vetores de atributos  $n$ -dimensionais binários, e clusterização. Na rede de anticorpos, os padrões de entrada (ou padrões a serem reconhecidos ou clusterizados) correspondem a antígenos, enquanto os nós da rede correspondem a anticorpos. Inicialmente, assume-se a existência de uma população de antígenos  $\mathbf{Ag}$  a ser reconhecida pelo conjunto de anticorpos  $\mathbf{Ab}$ . Um anticorpo  $\mathbf{Ab}_k$  será representado pelo vetor de pesos  $\mathbf{w}_k$ , que conecta as entradas ao anticorpo. Todas as unidades da ABNET constituem unidades de processamento lineares, pois a saída é uma combinação linear das entradas, ou seja, as entradas são ponderadas pelos pesos. Os pesos são binários, podendo assumir somente os valores 0 ou 1.

A rede pode iniciar-se com um ou mais indivíduos compondo a população de anticorpos. O número de indivíduos aumenta até que a rede apresente habilidade para cobrir satisfatoriamente todo o espaço, ou seja, para se ligar a todos os antígenos existentes na população de antígenos de acordo com um limiar de afinidade  $\varepsilon$ , evitando a presença de células redundantes na rede.

Conforme mencionado previamente, a ABNET baseia-se em aprendizado competitivo. Como os padrões de entrada e os pesos associados aos anticorpos são binários, a afinidade entre o anticorpo e o padrão é medida através da distância de Hamming. O anticorpo  $k$  que apresentar maior afinidade com o antígeno será o vencedor do processo competitivo. O receptor do antígeno e o receptor de cada anticorpo interagem de acordo com o formalismo do espaço de formas (ver Seção 5.2.2), ou seja, ligação por complementaridade. Isso é modelado considerando que as células apresentam similaridade máxima quando a distância de Hamming entre o anticorpo e o antígeno é máxima. Assim, a célula  $\mathbf{Ab}_k$  do repertório de anticorpos com maior afinidade ao antígeno  $\mathbf{Ag}$  será aquela que satisfaz a seguinte equação:

$$k = \arg \max_k (\|\mathbf{Ag} - \mathbf{Ab}_k\|). \quad (5.6)$$

A dinâmica que guia o tamanho da população baseia-se no processo de seleção clonal, em que as células mais estimuladas se multiplicam e maturam e as menos estimuladas são eliminadas. Sem perda de generalidade, pois a rede é construtiva, a arquitetura inicial envolve apenas um anticorpo.

Uma estratégia de hipermutação é empregada com o objetivo de aumentar a afinidade dos anticorpos durante o processo de auto-organização da rede.

O pseudo-código para a ABNET é exibido no Algoritmo 5.2. Conforme pode ser observado, o nível de concentração  $\tau_k$  mede o número de vezes que um anticorpo específico apresentou maior afinidade ao conjunto de antígenos.

A dinâmica de aumento de células na rede mimetiza o processo de seleção e reprodução das células mais estimuladas do sistema imunológico biológico. Na ABNET, o nível de estimulação de uma célula é calculado com base em dois valores: o nível de concentração  $\tau_j$  e o limiar de afinidade  $\varepsilon$ . A célula  $s$  com maior valor  $\tau_j$  é selecionada como candidata a clonagem. Caso a maior afinidade da célula  $s$  seja maior que o limiar  $\varepsilon$ , ela é clonada. Caso contrário, a rede mantém a mesma densidade de anticorpos. O vetor de pesos do novo clone é definido como sendo o complemento do vetor de pesos correspondente ao antígeno com pior

taxa de afinidade do anticorpo  $s$ . Tal estratégia visa prover uma rede que cubra todo o repertório de antígenos apresentados até o momento.

A ABNET envolve também um procedimento de eliminação de anticorpos. Caso um anticorpo mantenha seu nível de concentração  $\tau_j$  igual a zero durante um intervalo de tempo maior que um limiar previamente estabelecido, a célula é eliminada da rede.

O processo de ajuste de pesos dos anticorpos baseia-se em um mecanismo de hipermutação. A taxa de mutação  $\alpha$  determina quantos bits do vetor de pesos serão mutados, de modo que as posições não complementares formam o conjunto de bits candidatos a serem mutados. A taxa de mutação é inversamente proporcional à afinidade do anticorpo em relação aos antígenos.

Como principais entradas para a abNet, têm-se:

- População  $\mathbf{Ag}$  composta por  $M$  antígenos de comprimento  $L$ .
- Quantidade  $gen$  de gerações a serem executadas (critério de parada).
- Limiar de afinidade  $\epsilon$ .
- Taxa inicial  $\alpha_0$  de hipermutação;
- A cada  $\beta$  iterações a taxa  $\alpha$  de hipermutação é decrescida de uma unidade e um anticorpo é escolhido para ser um candidato à clonagem. Se um anticorpo  $p$  não é estimulado por nenhum antígeno durante  $\beta$  iterações, então ele é eliminado da rede.

Como saídas, têm-se:

- Matriz  $\mathbf{Ab}$  de anticorpos de memória.
- Matriz  $\mathbf{F}$  contendo a afinidade de cada anticorpo  $\mathbf{Ab}_i$ ,  $i = 1, \dots, N$  (o valor de  $N$  é automaticamente definido ao longo da execução do algoritmo), em relação a cada antígeno  $\mathbf{Ag}_j$ ,  $j = 1, \dots, M$ .

**Procedimento**  $[\mathbf{Ab}, \mathbf{F}] = \text{ABNET}(\mathbf{Ag}, \varepsilon, gen, \beta, \alpha_0)$

1. Gerar anticorpo inicial  $\mathbf{Ab}_0$ ;  $\alpha = \alpha_0$

2. Repita

2.1. Para cada antígeno  $\mathbf{Ag}_j, j = 1, \dots, M$ , faça:

2.1.1. Para cada anticorpo  $\mathbf{Ab}_i, i = 1, \dots, N$ , faça

2.1.1.1. Calcular a afinidade  $f_{i,j}$  do anticorpo  $\mathbf{Ab}_i$  ao antígeno  $\mathbf{Ag}_j$

2.1.2. Fim Para

2.1.3. Selecionar o anticorpo  $\mathbf{Ab}_k$  de maior afinidade de acordo com a Equação 5.6

2.1.4. Incrementar o nível de concentração do anticorpo  $\mathbf{Ab}_k$ :  $\tau_k \leftarrow \tau_k + 1$

2.1.5. Calcular a taxa de mutação  $\alpha$  com base na afinidade

2.1.6. Atualizar os pesos associados a  $\mathbf{Ab}_k$  de acordo com a taxa  $\alpha$

2.1.7. Atualizar  $f_{k,j}$

2.2. Fim Para

2.3. Para cada anticorpo  $\mathbf{Ab}_i, i = 1, \dots, N$ , faça

2.3.1. Se  $\tau_i = 0$  por mais de  $\beta$  iterações então eliminar  $\mathbf{Ab}_i$  da população

2.3.2. Se  $\tau_i > 1$  então

2.3.2.1. Selecionar antígeno  $n$  ao qual  $\mathbf{Ab}_i$  possui maior afinidade  $f_{i,n}$

2.3.2.2. Se  $f_{i,n} > \varepsilon$  então

2.3.2.2.1. Selecionar antígeno  $m$  ao qual  $\mathbf{Ab}_i$  possui menor afinidade  $f_{i,m}$

2.3.2.2.2. Gerar clone cujos pesos são o complemento de  $\mathbf{Ag}_m$

2.3.2.3. Fim Se

2.3.3. Fim Se

2.4. Fim Para

3. Até atingir  $gen$  gerações

**Fim Procedimento**

**Algoritmo 5.2.** ABNET.

## 5.5 aiNet (Artificial Immune Network)

### 5.5.1 Aspectos Gerais

A aiNet (*Artificial Immune NETWORK*) constitui um modelo de rede imunológica artificial inspirado na teoria da rede imunológica introduzida por JERNE (1974). Foi originalmente proposta por DE CASTRO & VON ZUBEN (1999; 2000). A aiNet incorpora a versão destinada a reconhecimento de padrões do CLONALG, descrita na Seção 5.3, como parte do processo de treinamento da rede. O objetivo da aiNet é criar imagens internas dos antígenos de modo que possa reconhecê-los, identificar e quantificar subgrupos (clusters) de antígenos, e gerar regras de decisão que permitam classificar novos padrões de entrada. As seções seguintes fornecerão detalhes acerca da operação da aiNet, apresentando um pseudocódigo com as principais etapas.

### 5.5.2 Notação

A terminologia adotada para a aiNet, de acordo com DE CASTRO & TIMMIS (2002a), encontra-se especificada a seguir.

- **Ab**: repertório de anticorpos disponíveis ( $\mathbf{Ab} = \mathbf{Ab}_{\{d\}} \cup \mathbf{Ab}_{\{m\}}$ );
- $\mathbf{Ab}_{\{m\}}$ : repertório total de anticorpos de memória ;
- **Ag**: população de  $M$  antígenos a serem reconhecidos;
- $\mathbf{Ab}_{\{d\}}$ :  $d$  novos anticorpos a serem inseridos na população **Ab**;
- $f_j$ : vetor contendo a afinidade de todos os anticorpos em relação ao antígeno **Ag<sub>j</sub>**. A afinidade é inversamente proporcional à distância **Ag-Ab**;
- **S**: matriz de similaridade entre os anticorpos de memória;
- **C**: população de  $N_c$  clones gerada a partir de **Ab**;
- **C\***: população **C** após o processo de maturação de afinidade;

- $f_{\{C^*\},j}$ : vetor contendo a afinidade de todos os clones de  $C^*$  em relação ao antígeno  $Ag$ ;
- $\xi$ : percentual de anticorpos maduros a serem selecionados;
- $M$ : clones de memória para o antígeno  $Ag$  restantes do processo de supressão clonal e seleção dos  $\xi$  anticorpos maduros;
- $\sigma_d$ : limiar de morte natural;
- $\sigma_s$ : limiar de supressão.

### 5.5.3 Descrição da aiNet

A rede imunológica é composta basicamente por um conjunto  $Ab$  de anticorpos, em que cada anticorpo interage com os demais anticorpos da rede e com um conjunto  $Ag$  de antígenos externos à rede. Cada antígeno e cada anticorpo são descritos por um vetor de  $L$  atributos no espaço euclidiano.

A aiNet constitui um grafo composto por um conjunto de nós e um conjunto de arestas ponderadas interligando os nós. Cada nó representa um anticorpo e cada aresta representa o grau de afinidade entre um par de anticorpos da rede. A inexistência de conexão entre um par de anticorpos indica que eles não interagem, ou seja, não há afinidade entre eles. A menos que se indique o contrário, a aiNet forma um grafo totalmente conectado.

O processo adaptativo da aiNet envolve duas etapas: interação de antígenos e anticorpos ( $Ag-Ab$ ) e interação somente de anticorpos ( $Ab-Ab$ ). As interações  $Ag-Ab$  e  $Ab-Ab$  são quantificadas pelo grau de proximidade ou similaridade entre antígenos e anticorpos e entre os anticorpos da rede. A afinidade é calculada com base em uma métrica de distância, de forma que a afinidade entre elementos é inversamente proporcional à distância entre eles. Uma afinidade  $Ab-Ag$  elevada promove uma resposta imunológica, enquanto uma afinidade  $Ab-Ab$  elevada corresponde, em analogia com o sistema imunológico biológico, a células que respondem a antígenos próprios, conhecidas como células auto-reativas. A presença de

anticorpos auto-reativos resulta em supressão da rede. No caso da aiNet, a supressão é realizada eliminando-se um dentre dois anticorpos auto-reativos (o de menor afinidade com os antígenos, por exemplo). Os anticorpos são eliminados de acordo com um limiar de supressão. Assim, anticorpos que apresentarem alta afinidade entre si, maior que um limiar de supressão  $\sigma_s$ , serão candidatos à eliminação.

Conforme a teoria da rede imunológica, os anticorpos existentes na rede irão competir pelo reconhecimento antigênico, sendo que aqueles que apresentarem maior afinidade com os antígenos serão os vencedores e sofrerão processo de clonagem e mutação de acordo com o princípio da seleção clonal. O número de clones gerados é função da afinidade do anticorpo aos antígenos. Quanto maior a afinidade, maior o número de clones, e vice-versa.

A aiNet tem por objetivo construir um conjunto de anticorpos de memória que melhor represente o conjunto de antígenos.

Em termos de espaço de formas, os anticorpos correspondem a protótipos que serão posicionados em regiões que contêm antígenos. O algoritmo 5.3 apresenta o pseudo-código para a aiNet.

As entradas para o algoritmo da aiNet apresentado são:

- População  $\mathbf{Ag}$  composta por  $M$  antígenos.
- Quantidade  $gen$  de gerações a serem executadas (critério de parada).
- Número  $n$  de anticorpos a serem selecionados para clonagem.
- Percentual  $\zeta\%$  de anticorpos maduros a serem selecionados.
- Limiar  $\sigma_d$  de morte natural.
- Limiar  $\sigma_s$  de supressão.
- Quantidade  $d$  de novos anticorpos a serem inseridos na rede.

As saídas são:

- Matriz  $\mathbf{Ab}_{\{m\}}$  de anticorpos de memória.
- Matrix  $\mathbf{S}$  de similaridade entre os anticorpos de memória da rede, composta por elementos  $s_{i,j}$ .

**Procedimento [Ab, S]=aiNet(Ag,  $\sigma_d$ ,  $\sigma_s$ ,  $\xi$ , gen, n, d)**

1. Gerar aleatoriamente população inicial de anticorpos **Ab**

2. Repita

2.1. Para cada antígeno **Ag<sub>j</sub>**,  $j = 1, \dots, M$ , faça

2.1.1. Calcular a afinidade  $f_{i,j}$  de cada anticorpo **Ab<sub>i</sub>**,  $i = 1 \dots N$ , ao antígeno **Ag<sub>j</sub>**

2.1.2. Selecionar o conjunto **Ab<sub>{n}</sub><sup>j</sup>** de  $n$  anticorpos de maior afinidade ao antígeno **Ag<sub>j</sub>**

2.1.3. Clonar elementos do conjunto **Ab<sub>{n}</sub><sup>j</sup>** a taxas proporcionais à afinidade, gerando o conjunto de clones **C<sup>j</sup>**

2.1.4. Submeter **C<sup>j</sup>** a mutação a taxas inversamente proporcionais à afinidade, gerando o conjunto **C<sup>j\*</sup>**

2.1.5. Determinar o vetor de afinidades  $f_{\{C^{j*}\},j}$  entre **Ag<sub>j</sub>** e os anticorpos de **C<sup>j\*</sup>**

2.1.6. Da população **C<sup>\*</sup>**, selecionar  $\xi\%$  dos anticorpos com maior afinidade e colocá-los em uma matriz **M** de clones de memória (memória clonal)

2.1.7. Morte programada: eliminar anticorpos  $k \in M$  para os quais  $f_{k,j} < \sigma_d$

2.1.8. Calcular afinidade  $s_{k,l}$  entre **Ab<sub>k</sub>** e **Ab<sub>l</sub>** para todo  $k \in M$  e  $l \in M$

2.1.9. Supressão clonal: Para todo  $k \in M$  e  $l \in M$ , se  $s_{k,l} > \sigma_s$  então

2.1.9.1. Se  $f_{k,\{Ag\}} > f_{l,\{Ag\}}$  então eliminar **Ab<sub>l</sub>** de **M**, senão eliminar **Ab<sub>k</sub>** de **M**

2.1.10. **Ab<sub>{m}</sub>**  $\leftarrow$  **Ab<sub>{m}</sub>** + **M**

2.2. Fim Para

2.3. Calcular afinidade  $s_{k,l}$  entre **Ab<sub>k</sub>** e **Ab<sub>l</sub>** para todo  $k \in M$  e  $l \in M$

2.4. Supressão da rede: para todo  $k \in M$  e  $l \in M$ , se  $s_{k,l} > \sigma_s$  então

2.4.1. Se  $f_{k,\{Ag\}} > f_{l,\{Ag\}}$  então eliminar **Ab<sub>l</sub>** de **M**, senão eliminar **Ab<sub>k</sub>** de **M**

2.5. Gerar novos anticorpos **Ab<sub>{d}</sub>** aleatoriamente ou através do modelo de medula óssea e introduzir na população: **Ab**  $\leftarrow$  **Ab** + **Ab<sub>{d}</sub>** + **Ab<sub>{m}</sub>**

3. Até atingir *gen* gerações

4. Eliminar anticorpos cujas afinidades aos antígenos nunca é a máxima para ao menos um antígeno

**Fim Procedimento**

**Algoritmo 5.3. aiNet.**

Conforme pode ser observado no Algoritmo 5.3, a resposta imunológica adaptativa ocorre após a apresentação de cada padrão de entrada (antígeno) à rede. Os antígenos são reconhecidos com diferentes níveis de afinidade pelos anticorpos da rede. Os anticorpos mais bem adaptados serão proliferados e sofrerão hipermutações somáticas e supressão, de acordo com os princípios de seleção clonal e maturação de afinidade.

#### **5.5.4 Extração de Conhecimento da aiNet Treinada**

Após o processo de adaptação da rede, têm-se disponíveis as imagens internas dos antígenos, tal como ilustrado na Figura 5.7. A rede treinada, resultante da distribuição de antígenos exibida na Figura 5.7 (a), é mostrada na Figura 5.7 (b).

As linhas tracejadas da Figura 5.7 (b) indicam arestas a serem removidas de forma a gerar subgrafos desconexos, cada subgrafo correspondendo a um grupo específico de antígenos. Geralmente, as arestas a serem removidas são definidas estabelecendo-se um limiar de corte para as conexões, de acordo com seu peso.

O exemplo mostrado na Figura 5.7 ilustra um caso bastante simples, em que há poucos antígenos a serem reconhecidos e distribuídos em um espaço de apenas duas dimensões. Conforme mencionado anteriormente, os anticorpos de memória representam imagens internas dos antígenos, sendo portanto representados por vetores de atributos de mesma dimensão dos antígenos. Esse fato inviabiliza a visualização dos anticorpos em um espaço de dimensão maior que três. Nessas situações, é necessário efetuar um procedimento automático de extração de conhecimento da aiNet treinada. Há diversas técnicas que podem ser empregadas para se alcançar esse objetivo. Uma estratégia simples é suprimir conexões cujos pesos extrapolem um limiar previamente definido, tal como apresentado na Figura 5.7 (b). Entretanto, tal estratégia pode omitir certas informações representadas pela aiNet, tais como formato dos agrupamentos e número de anticorpos representantes de cada agrupamento. Técnicas de clusterização, tais como a clusterização hierárquica, e métodos para se encontrar a árvore geradora mínima a partir da matriz de afinidade entre anticorpos constituem mecanismos bastante efetivos para se extrair a estrutura da aiNet. Mais detalhes acerca da extração de conhecimento da aiNet são apresentadas em DE CASTRO (2002).

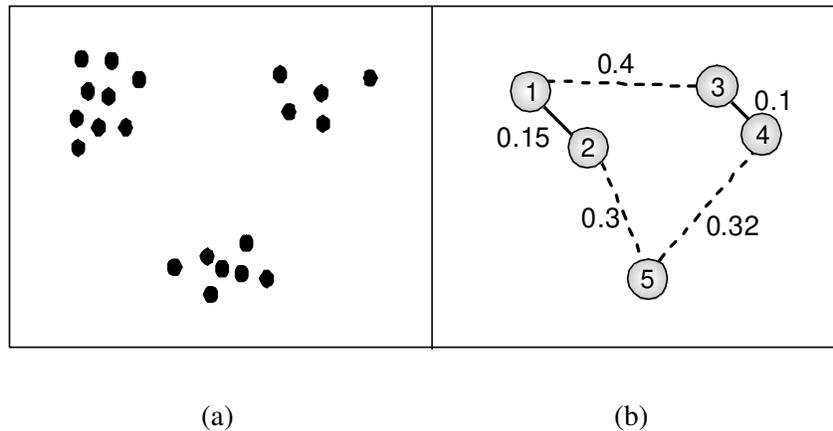


Figura 5.7. (a) Padrões de entrada; (b) Resultado da clusterização via aiNet.

## 5.6 opt-aiNet (Artificial Immune Network for Optimization)

### 5.6.1 Aspectos Gerais

Nesta seção é apresentada a opt-aiNet (*Artificial Immune NETWORK for OPTimization*) proposta por DE CASTRO & TIMMIS (2002). A opt-aiNet representa uma extensão da aiNet de forma que a rede seja capaz de operar eficientemente na otimização de funções multimodais.

### 5.6.2 Notação e Terminologia

A seguir, são apresentados a terminologia e o algoritmo proposto por DE CASTRO & TIMMIS (2002) para a opt-aiNet.

- **Ab**: repertório de anticorpos disponíveis;
- **Ab<sub>{d}</sub>**:  $d$  novos anticorpos a serem inseridos na população **Ab**;
- **f<sub>j</sub>**: vetor contendo a afinidade de todos os anticorpos em relação ao antígeno **Ag<sub>j</sub>**. A afinidade é inversamente proporcional à distância **Ag-Ab**;
- **S**: matriz de similaridade entre os anticorpos de memória com elementos  $s_{i,j}$ ;
- **C**: população de  $N_c$  clones gerada a partir de **Ab**;

- $C^*$ : população  $C$  após o processo de maturação de afinidade;
- $f_{\{C^*\},j}$ : vetor contendo a afinidade de todos os clones de  $C^*$  em relação ao antígeno  $Ag$ ;
- $\xi$ : percentual de anticorpos maduros a serem selecionados;
- $\sigma_s$ : limiar de supressão.

O Algoritmo 5.4 apresenta o pseudo-código da optAiNet. As entradas para o algoritmo são:

- Quantidade  $gen$  de gerações a serem executadas (critério de parada).
- Percentual  $\zeta\%$  de anticorpos maduros a serem selecionados.
- Limiar  $\sigma_s$  de supressão.
- Quantidade  $d$  de novos anticorpos a serem inseridos na rede.
- Quantidade  $Nc$  de clones a serem gerados.

As saídas são:

- Repertório final  $\mathbf{Ab}$  de anticorpos.
- Matrix  $\mathbf{S}$  de similaridade entre os anticorpos de memória da rede, composta por elementos  $s_{i,j}$ .

### 5.6.3 Algoritmo para a opt-aiNet

**Procedimento**  $[\mathbf{Ab}, \mathbf{S}] = \text{opt-aiNet}(\mathbf{Ag}, \sigma_s, Nc, \zeta, gen, d)$

1. Gerar aleatoriamente população inicial de anticorpos  $\mathbf{Ab}$
  2. Repita
    - 2.1. Para cada anticorpo  $\mathbf{Ab}_i, i = 1, \dots, N$ , calcular fitness  $f_i$
    - 2.2. Gerar  $Nc$  clones para cada elemento da rede, produzindo o conjunto de clones  $\mathbf{C}$
    - 2.3. Submeter os anticorpos de  $\mathbf{C}$  a mutação a taxa inversamente proporcional à afinidade, gerando o conjunto  $\mathbf{C}^*$
    - 2.4. Determinar o vetor de fitness  $f_{\{\mathbf{C}^*\}}$  para os anticorpos de  $\mathbf{C}^*$
    - 2.5. Da população  $\mathbf{C}^*$ , selecionar  $\xi\%$  dos anticorpos que apresentam maior afinidade e incorporá-los à população
    - 2.6. Para cada anticorpo  $\mathbf{Ab}_i \in \mathbf{Ab}$  calcular fitness  $f_i$ , selecionar e preservar o clone de maior fitness
    - 2.7. Se a variação de fitness médio da população não for significativamente diferente da iteração anterior então ir para o passo 2.8, senão voltar ao passo 2.1
    - 2.8. Para cada anticorpo  $l, k \in \mathbf{Ab}, l \neq k$ , calcular afinidade  $s_{k,l}$  entre  $\mathbf{Ab}_k$  e  $\mathbf{Ab}_l$
    - 2.9. Supressão: Para cada anticorpo  $l, k \in \mathbf{Ab}$ , se  $s_{k,l} > \sigma_s$  então
      - 2.9.1. Se  $f_k > f_l$ , eliminar  $\mathbf{Ab}_l$  de  $\mathbf{Ab}$ ; Senão, eliminar  $\mathbf{Ab}_k$  de  $\mathbf{Ab}$
    - 2.10. Gerar novos anticorpos  $\mathbf{Ab}_{\{d\}}$  e introduzir na população:  $\mathbf{Ab} \leftarrow \mathbf{Ab} + \mathbf{Ab}_{\{d\}}$
    - 2.11. Fim Para
  3. Até atingir  $gen$  gerações
  4. Supressão: Para cada anticorpo  $l, k \in \mathbf{Ab}$ , se  $s_{k,l} > \sigma_s$  então
    - 4.1. Se  $f_k > f_l$ , eliminar  $\mathbf{Ab}_l$  de  $\mathbf{Ab}$
    - 4.2. Senão, eliminar  $\mathbf{Ab}_k$  de  $\mathbf{Ab}$
- Fim Procedimento**

**Algoritmo 5.4.** opt-aiNet.

O comportamento do algoritmo pode ser explicado da seguinte forma:

Passos 2.1 a 2.6: A cada iteração, a população de células é otimizada localmente, via mutação inversamente proporcional à afinidade.

Passos 2.7 a 2.9: Quando a população alcança um estado de estabilidade, as células da rede interagem, de acordo com a teoria da rede imunológica, e células similares são eliminadas. No passo 2.10, novos anticorpos são inseridos para introduzir diversidade na rede.

Passo 4: Ao final do procedimento, as células da rede interagem, de acordo com a teoria da rede imunológica, e células similares são eliminadas.

A mutação proporcional à afinidade (Passo 2.3) é efetuada de acordo com a seguinte expressão:

$$c' = c + \alpha N(0,1), \quad (5.7)$$

sendo  $\alpha$  dado por:

$$\alpha = (1/\beta) \exp(-f^*), \quad (5.8)$$

onde  $c'$  é uma célula  $c$  mutada,  $N(0,1)$  é uma variável aleatória gaussiana de média zero e desvio padrão  $\sigma = 1$ ,  $\beta$  é um parâmetro que controla o decaimento da exponencial, e  $f^*$  é o fitness do indivíduo normalizado entre 0 e 1.

O critério de parada adotado é baseado no tamanho da população. Após a fase de supressão, um determinado número de células diversificadas é mantido na população. Caso esse número não varie de uma supressão à outra, a rede alcançou um estado de estabilidade e o critério de parada foi atingido.

#### **5.6.4 Exemplos de Comportamento do Algoritmo**

A seguir são apresentados os resultados obtidos por DE CASTRO & TIMMIS (2002) para a opt-aiNet quando aplicada a alguns problemas multimodais bidimensionais. Observe que, para todos os casos exemplificados, múltiplos pontos de ótimo local encontram-se presentes no

escopo da busca. Conforme será apresentado nas próximas subseções, a opt-aiNet foi capaz de determinar todos os pontos de ótimo (locais e global). É evidente que isso foi possível graças à reduzida dimensão do espaço de busca, sendo que em aplicações práticas não é possível garantir que todos os ótimos locais, e mesmo o ótimo global, sejam encontrados. Para o caso em que infinitos pontos de ótimo são definidos, a opt-aiNet foi capaz de explorar de forma efetiva o espaço de busca, encontrando pontos de ótimo uniformemente distribuídos ao longo do lugar geométrico de ótimos locais. Note que a opt-aiNet foi capaz de explorar o espaço de busca de maneira mais eficiente que o CLONALG, apresentado na Seção 5.3. Tal fato mostra que a aplicação conjunta do algoritmo da seleção clonal com a teoria da rede imunológica leva a um controle mais eficiente da população, e a uma distribuição mais eficaz dos indivíduos ao longo do espaço de busca. A habilidade de explorar de forma eficiente o espaço de busca, através de uma população diversificada e otimizada, faz da opt-aiNet uma ferramenta poderosa, principalmente quando consideramos aplicações em que seja importante a obtenção de múltiplas soluções para um mesmo problema e também no caso de problemas com múltiplos objetivos, em que se busca amostrar a superfície de Pareto (COELLO & CORTÉS, 2005). Tal desempenho dificilmente seria alcançado via outras abordagens populacionais, como algoritmos genéticos, voltados para a busca apenas do melhor ponto de ótimo local. Os algoritmos genéticos clássicos, bem como suas variantes, privilegiam os melhores indivíduos da população, tanto no momento do *crossover* como no processo de seleção dos indivíduos para a próxima iteração (seleção elitista, método da roleta (*roulette wheel*) etc.), o que tende a produzir uma população centrada no ponto de ótimo ou mesmo, em particular para casos em que a função a ser otimizada não é suficientemente suave, levar a um ponto de ótimo local de baixa qualidade.

Nesta seção serão apresentados resultados para três funções:

- Multifunção: é uma função com múltiplos pontos de ótimo local e um único ponto de ótimo global.
- Raízes: Possui seis pontos de ótimo local idênticos e dois grandes platôs;
- Função de Schaffer: apresenta infinitos ótimos locais e um único ótimo global.

#### 5.6.4.1 Parâmetros Utilizados

Existem seis parâmetros a serem definidos para a opt-aiNet: limiar de supressão, tamanho inicial da população, número de clones a serem gerados para cada célula, percentual de novos indivíduos a serem introduzidos na população, escala da função de seleção proporcional à afinidade e número máximo de passos do algoritmo. O limiar de supressão indica as células que devem ser suprimidas, ou seja, aquelas cuja afinidade com outra(s) célula(s) da rede seja maior que esse limiar serão eliminadas. O tamanho inicial da população indica o número de indivíduos (células imunológicas) ao início da execução do algoritmo; normalmente este parâmetro pode ter um valor padrão, como, por exemplo,  $N=20$  indivíduos. O número de clones determina quantos clones serão gerados quando um indivíduo for selecionado para clonagem. A porcentagem de novos indivíduos indica quantas novas células serão inseridas na população quando for detectada a necessidade de aumento da população devido à baixa diversidade; este parâmetro também pode ter um valor padrão, por exemplo 5% do tamanho atual da população. Novos indivíduos são geralmente inseridos após um processo de supressão. O valor padrão  $\beta = 100$  é utilizado em todos os experimentos aqui reportados. O valor  $Ngen$  representa o número máximo de iterações que o algoritmo deve executar.

Os parâmetros de treinamento da rede foram os mesmos para os três casos:

- Limiar de supressão:  $\sigma_s = 0,2$ .
- Tamanho inicial da população:  $N = 20$ .
- Número de clones gerados para cada célula da rede:  $Nc = 10$ .
- Porcentagem de novos indivíduos:  $d = 40\%$ .
- Escala da função de seleção proporcional à afinidade:  $\beta = 100$ .
- Número máximo de passos do algoritmo:  $Ngen = 500$ .

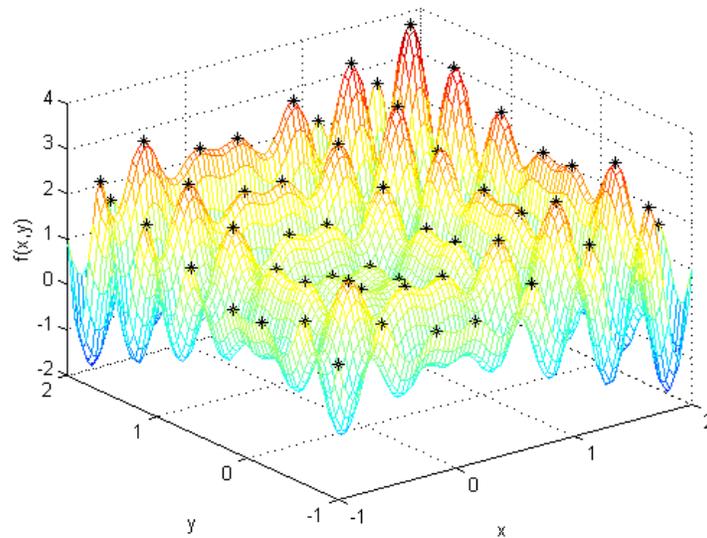
### 5.6.4.2 Multifunção

É reproduzida a seguir a equação já apresentada na Seção 5.2.5.1.

$$g(x,y) = x.\sin(4\pi x) - y.\sin(4\pi y + \pi) + 1, \quad (5.9)$$

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

A Figura 5.8 apresenta os múltiplos pontos de ótimo local obtidos via opt-aiNet. Conforme pode ser observado, a opt-aiNet encontrou 61 picos.



**Figura 5.8.** Resultados gerados pela opt-aiNet para a multifunção (figura extraída de DE CASTRO & TIMMIS, 2002).

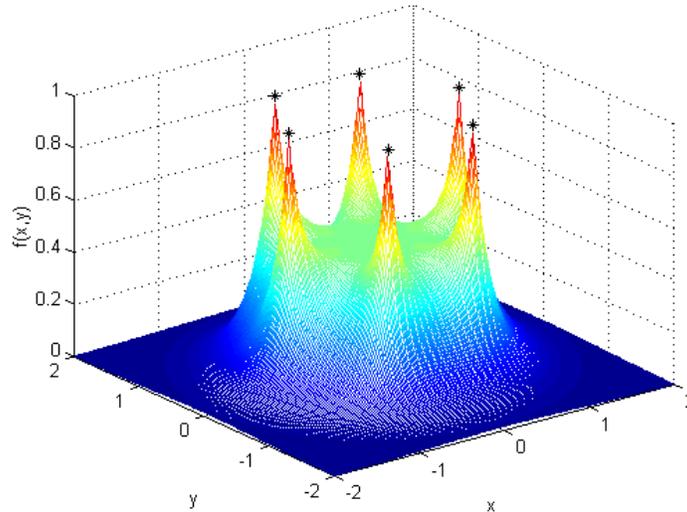
### 5.6.4.3 Função Raízes

A função raízes é dada pela seguinte equação:

$$g(z) = \frac{1}{1 + |z^6 - 1|}, \quad (5.10)$$

onde  $z \in \mathbf{C}$ ,  $z = x + iy$ ,  $x,y \in [-2,2]$ .

A Figura 5.9 apresenta os ótimos locais encontrados pela opt-aiNet. Observe que a opt-aiNet foi capaz de detectar todos os picos da função, com um único indivíduo em cada pico.



**Figura 5.9.** Múltiplos ótimos locais detectados pela opt-aiNet para a função raízes (figura extraída de DE CASTRO & TIMMIS, 2002).

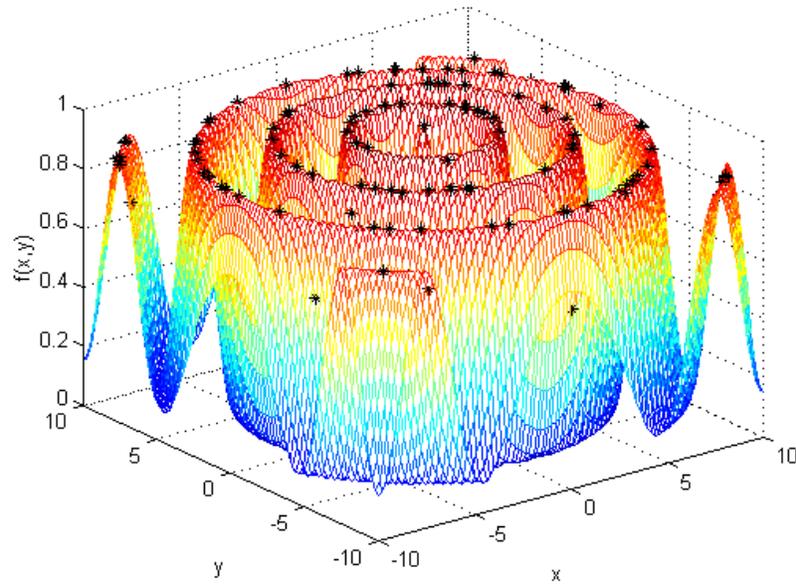
#### 5.6.4.4 Função de Schaffer

A função de Schaffer é dada por:

$$g(z) = 0,5 + \frac{\sin^2\left(\sqrt{x^2 + y^2}\right) - 0,5}{\left(1 + 0,001(x^2 + y^2)\right)}, \quad (5.11)$$

onde  $x, y \in [-10, 10]$ .

A Figura 5.10 apresenta um resultado obtido pela opt-aiNet para a função de Schaffer. Observe que a opt-aiNet foi capaz de detectar o ponto de ótimo global e de explorar eficientemente o espaço de busca, apresentando células distribuídas de forma equilibrada ao longo de toda a superfície em que existem ótimos locais.



**Figura 5.10.** Resultados da função de Schaffer obtidos pela opt-aiNet (figura extraída de DE CASTRO & TIMMIS, 2002).

# CAPÍTULO 6

## **COPT-AINET: UMA ABORDAGEM BASEADA EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS PARA PROBLEMAS COMBINATÓRIOS**

---

---

Neste capítulo é proposta a *copt-aiNet*, uma nova abordagem baseada em sistemas imunológicos artificiais destinada à solução de problemas de caráter combinatório. A *copt-aiNet* é uma extensão da *opt-aiNet*, detalhada no Capítulo 5, e concebida para otimização de funções multimodais definidas em um espaço contínuo e limitado. A *copt-aiNet* constitui uma meta-heurística populacional capaz de encontrar e manter diversas soluções de alta qualidade. Serão apresentados o algoritmo da *copt-aiNet* e alguns resultados computacionais para o problema do caixeiro viajante simétrico.

### **6.1 Introdução**

Esta seção fornece uma breve apresentação dos principais conceitos básicos referentes a sistemas imunológicos necessários para a compreensão da *copt-aiNet* e introduz a terminologia empregada. Para mais detalhes acerca do sistema imunológico biológico, consultar o Apêndice B. O Capítulo 5 provê uma descrição acerca dos tópicos relacionados a ferramentas provenientes de sistemas imunológicos artificiais, as quais serviram como fonte de inspiração para a *copt-aiNet*.

A *copt-aiNet* emprega três importantes conceitos provenientes da área de imunologia:

1. Seleção clonal e maturação de afinidade (DE CASTRO & VON ZUBEN, 2000; DE CASTRO & TIMMIS, 2002a).

2. Teoria da rede imunológica (JERNE, 1974; DE CASTRO & TIMMIS, 2002a).
3. Recombinação gênica (DE CASTRO & TIMMIS, 2002a).

O princípio da seleção clonal, detalhado no Capítulo 5, define que a taxa de reprodução de anticorpos está diretamente associada à afinidade entre o anticorpo e o antígeno, enquanto a taxa de mutação é inversamente proporcional à afinidade. Assim, quanto maior a afinidade maior o número de clones gerados e menor a taxa de mutação aplicada.

Uma outra importante teoria da imunologia utilizada pela *copt-aiNet* é a teoria da rede imunológica. A teoria da rede imunológica coloca que as células imunológicas, além de reconhecerem antígenos, são capazes de reconhecer outras células imunológicas. Assim, o sistema imunológico pode gerar uma resposta devido à interação de células imunológicas. Basicamente, quando uma célula imunológica apresenta alta afinidade em relação a outra célula imunológica, ocorre um processo de supressão, com o objetivo de eliminar células redundantes.

O último princípio de imunologia considerado na proposta da *copt-aiNet* é a geração de moléculas de anticorpos através de bibliotecas gênicas, em que células de alta afinidade com anticorpos tendem a formar bibliotecas de genes com o intuito de gerar anticorpos mais eficientes baseados no cruzamento de grupos de genes das melhores soluções encontradas pela rede imunológica até o momento (modelo da medula óssea, descrito no Capítulo 5).

## **6.2 A *copt-aiNet* (Artificial Immune Network for Combinatorial Optimization)**

### **6.2.1 Aspectos Gerais**

Nesta seção é apresentado o algoritmo para a *copt-aiNet* (*Artificial Immune NETWORK for Combinatorial OPTimization*). A *copt-aiNet* é uma extensão da *opt-aiNet* (*Artificial Immune Network for Optimization*), proposta em DE CASTRO & TIMMIS (2002) e descrita no Capítulo 5. Enquanto a *opt-aiNet* opera em espaços contínuos, a *copt-aiNet* foi concebida para realizar otimização de funções multimodais definidas em espaços discretos. Para mais detalhes acerca

da opt-aiNet e outras metodologias baseadas em sistemas imunológicos artificiais que fundamentam a copt-aiNet, tais como o algoritmo da seleção clonal e a rede imunológica, consultar o Capítulo 5.

A copt-aiNet foi especialmente projetada para otimização de problemas combinatórios, apresentando a habilidade de encontrar e manter múltiplas soluções diversificadas.

A título de exemplificação, a explanação do algoritmo e exemplos de comportamento serão baseados no problema do caixeiro viajante simétrico. Entretanto, o algoritmo pode ser diretamente adaptado a outras classes de problemas.

### **6.2.2 Notação e Terminologia**

A seguir, são apresentados a terminologia e o algoritmo proposto para a copt-aiNet.

- **Ab**: repertório de anticorpos disponíveis;
- **Ab<sub>(d)</sub>**:  $d$  novos anticorpos a serem inseridos na população **Ab**;
- **f<sub>j</sub>**: vetor contendo a afinidade de todos os anticorpos em relação ao antígeno **Ag<sub>j</sub>**. A afinidade é inversamente proporcional à distância **Ag-Ab** e é dada pela avaliação de uma função-objetivo;
- **S**: matriz de similaridade entre os anticorpos de memória, com elementos  $s_{ij}$  (afinidade **Ab<sub>i</sub>, Ab<sub>j</sub>**);
- **C**: população de  $N_c$  clones gerada a partir de **Ab**;
- **C\***: população **C** após o processo de maturação de afinidade;
- **f<sub>{C\*},j</sub>**: vetor contendo a afinidade de todos os clones de **C\*** em relação ao antígeno **Ag<sub>j</sub>**;
- $\xi$ : percentual de anticorpos maduros a serem selecionados;
- $\sigma_s$ : limiar de supressão.

### 6.2.3 Algoritmo para a copt-aiNet

No algoritmo para a copt-aiNet, cada indivíduo da população corresponde a um anticorpo ou célula da rede e é representado por uma cadeia de atributos. Para o caso do caixeiro viajante, cada célula corresponde a uma solução completa para o problema, ou seja, a uma seqüência de cidades. Portanto, cada atributo que compõe a célula corresponde ao índice de uma cidade.

Dois índices são utilizados para avaliar a qualidade (adaptabilidade) de cada célula da rede: o valor de *fitness* e o valor de afinidade entre anticorpos. O valor de *fitness* de uma célula indica seu grau de adaptabilidade ao ambiente externo, ou seja, a avaliação de uma função-objetivo.

A afinidade de uma célula imunológica em relação a outras células indica o grau de reconhecimento entre anticorpos. O grau de reconhecimento é medido através de uma métrica de similaridade. Essa característica é explorada na copt-aiNet para regular o número de indivíduos na população e evitar redundância. Assim, células similares são identificadas e aquelas que apresentarem menor *fitness* serão eliminadas da rede. Essa etapa corresponde ao processo de supressão. O pseudocódigo apresentado no Algoritmo 6.3 ilustra os principais passos da copt-aiNet. Uma descrição mais detalhada de cada etapa será provida na seqüência.

As entradas para o algoritmo da copt-aiNet apresentado são:

- Quantidade *gen* de gerações a serem executadas (critério de parada).
- Percentual  $\zeta\%$  de anticorpos maduros a serem selecionados.
- Limiar  $\sigma_s$  de supressão.
- Quantidade *d* de novos anticorpos a serem inseridos na rede.
- Quantidade *Nc* de clones a serem gerados.
- Número *M* de iterações (gerações) que se admite a não melhoria no fitness dos indivíduos – após isso, será aplicado procedimento de otimização local.
- Número mínimo *m* de indivíduos na população.
- Número *nk* de indivíduos a serem submetidos a processo de maturação (otimização local) caso não haja melhoras no fitness dos indivíduos durante *M* gerações.

As saídas são:

- Repertório final **Ab** de anticorpos.

- Matrix **S** de similaridade entre os anticorpos de memória da rede, composta por elementos  $s_{i,j}$ .

**Procedimento [Ab, S]=copt-aiNet( $\sigma_s, m, M, Max\_It, z, Nc, nk$ )**

1. Gerar aleatoriamente população inicial de anticorpos **Ab**

2. Repita

2.1. Para cada anticorpo  $\mathbf{Ab}_i, i = 1, \dots, N$ , calcular *fitness*  $f_i$

2.3. Gerar  $Nc$  clones para cada elemento da rede, produzindo o conjunto de clones **C**

2.4. Submeter os anticorpos de **C** a mutação a taxa inversamente proporcional à afinidade, gerando o conjunto **C\***

2.5. Determinar o vetor de *fitness*  $f_{\{\mathbf{C}^*\}}$  para os anticorpos de **C\***

2.6. Da população **C\***, selecionar novamente  $\xi\%$  dos anticorpos que apresentam maior afinidade e incorporá-los à população

2.7. Calcular o *fitness* médio da população e preservar o clone de maior *fitness*

2.8. Se a variação de *fitness* médio da população não for significativamente diferente da iteração anterior então ir para o passo 2.9, senão voltar ao passo 2.3

2.9. Para cada anticorpo  $l, k \in \mathbf{Ab}, l \neq k$ , calcular afinidade  $s_{k,l}$  entre  $\mathbf{Ab}_k$  e  $\mathbf{Ab}_l$

2.10. Supressão: Para cada anticorpo  $l, k \in \mathbf{Ab}$ , se  $s_{k,l} > \sigma_s$  então

2.10.1. Se  $f_k > f_l$ , eliminar  $\mathbf{Ab}_l$  de **Ab**; Senão, eliminar  $\mathbf{Ab}_k$  de **Ab**

2.11. Se número de indivíduos na população  $< m$  então

2.11.1. Inserir novos elementos  $\mathbf{Ab}_{\{d\}}$  formados por recombinação gênica e introduzir na população:

$$\mathbf{Ab} \leftarrow \mathbf{Ab} + \mathbf{Ab}_{\{d\}}$$

2.12. Fim Se

2.13. Se nenhum dos  $nk$  melhores indivíduos tiver seu *fitness* aumentado por  $M$  iterações então

2.13.1. Submeter todos os elementos da população a um processo de maturação de afinidade fraco

2.14. Fim Se

3. Até atingir  $Mat\_It$  gerações

4. Submeter os  $nk$  melhores indivíduos da população a um processo de maturação de afinidade intensivo

**Fim Procedimento**

**Algoritmo 6.1.** copt-aiNet.

Os passos 2 a 3 englobam o laço principal do algoritmo. O comportamento do algoritmo pode ser explicado da seguinte forma:

*Passo 1 – Geração da população inicial:* no caso específico do caixeiro viajante, a população inicial é construída seguindo a heurística do vizinho mais próximo, iniciando com diferentes sementes (cidades iniciais). Ao início, a população contém um número predefinido de indivíduos e, ao longo do processo de adaptação, pode aumentar ou reduzir de tamanho, de acordo com as características do espaço de busca e com o estado corrente da rede quando se executam os processos de inserção e supressão de indivíduos.

*Passos 2.3 a 2.7 – Seleção clonal e maturação de afinidade:* a cada iteração, a população de células é modificada localmente via seleção clonal e mutações aplicadas a uma taxa proporcional ao custo da solução. Nesta fase, cada célula dá origem a um certo número de clones. Mesmo as células de maior custo produzirão clones, contribuindo para a capacidade de busca multimodal do algoritmo. Os clones são submetidos a um processo de mutação onde  $n$  genes de cada clone são aleatoriamente selecionados para mutação. O número  $n$  de genes a serem mutados é diretamente proporcional ao custo da solução (ou inversamente proporcional ao *fitness*): células apresentando custo elevado serão submetidas a altas taxas de mutação, enquanto células de baixo custo serão submetidas a baixas taxas de mutação. Três tipos de operadores de mutação foram implementados na *copt-aiNet*: 1) troca de posição de dois genes; 2) troca de posição de três genes; e 3) troca de posição de quatro genes.

*Passos 2.8 a 2.10 – Interações de rede e diversificação:* quando a rede de células alcança a estabilidade em termos de *fitness* médio  $avF$ , ou seja, o *fitness* médio da população não é melhorado durante um número predefinido de iterações, a rede é submetida a um processo de supressão. Durante a supressão, células similares (aquelas que apresentam alta afinidade entre si) são eliminadas da rede no intuito de evitar redundância e assim manter a diversidade da população. A similaridade (afinidade) entre duas soluções é definida como sendo a menor seqüência de operações necessárias para se converter uma solução na outra, e baseia-se no máximo subgrafo comum (BUNKE & SHEARER, 1998). O cálculo do critério de similaridade é detalhado na Seção 6.2.3.1.

*Passos 2.11 a 2.12 – Recombinação gênica:* novos anticorpos são construídos através da recombinação de indivíduos de alta afinidade antigênica mantidos em uma biblioteca gênica. As bibliotecas gênicas são formadas a partir de células de memória. As células de memória são divididas em  $p$  partições de comprimento  $l$ , onde  $p$  é um número inteiro aleatório entre 2 e 4. As partições são aleatoriamente selecionadas e recombinadas de modo a construir uma nova solução.

*Passos 2.13 a 2.14 – Busca local:* caso não incorram reduções de custo em nenhuma das  $k$  melhores soluções durante um número predefinido de iterações  $M$ , todas as células da população são submetidas a um processo de maturação. Durante o processo de maturação, as células imunológicas sofrem uma série de mutações controladas com o objetivo de aumentar sua afinidade em relação aos antígenos. Este processo é implementado através de uma estratégia de busca local. Qualquer procedimento de busca local pode ser empregado. Na implementação proposta nesta tese, é utilizada a busca tabu (GLOVER & LAGUNA, 1998). Para mais detalhes acerca da busca tabu, consultar o Capítulo 2. O processo de maturação é realizado repetidamente até que todos os indivíduos da população tenham seu custo reduzido  $t$  vezes ou até que o custo não seja reduzido ao longo de uma iteração tabu completa.

*Passo 4 – Busca local intensiva:* caso nenhuma das  $k$  melhores soluções tenham seu custo reduzido ao longo de  $Max\_it$  iterações, o critério de parada é atingido e as  $k$  soluções melhor adaptadas são submetidas a um procedimento de maturação intensivo, implementado via a meta-heurística busca tabu. Este passo visa otimizar localmente as soluções obtidas.

Para o caso particular do problema do caixeiro viajante, o *fitness* é calculado de acordo com a seguinte equação:

$$Fitness = \frac{1}{\sqrt{\sum_{i=1}^{Dim\_Solução-1} (\mathbf{x}_i - \mathbf{x}_{i+1})^2 + (\mathbf{x}_1 - \mathbf{x}_{Dim\_Solução})^2 + 1}}, \quad (6.1)$$



**Procedimento CalculaSimilaridade (A, B)**

1.  $k \leftarrow 1$ ;  $Num\_Operações \leftarrow 0$ ;  $Cur\_Elemento \leftarrow A(k)$
2.  $Fim \leftarrow falso$
3. Encontrar índice  $j$  tal que  $B(j) = Cur\_Elemento$
4. Enquanto  $Fim = falso$  faça
  - 4.1.  $k \leftarrow k + 1$
  - 4.2.  $Cur\_Elemento \leftarrow A(k)$
  - 4.3. Procurar por índice  $n$  tal que  $B(n) = Cur\_Elemento$
  - 4.4. Se  $j + 1 > Dim\_Solução$  então  $Índice1 \leftarrow 1$ , senão  $Índice1 \leftarrow j + 1$
  - 4.5. Se  $j - 1 < 1$  então  $Índice2 \leftarrow Dim\_Solução$ , senão  $Índice2 \leftarrow j - 1$
  - 4.6. Se  $n \neq Índice1$  e  $n \neq Índice2$  então
    - 4.6.1.  $Num\_Operações \leftarrow Num\_Operações + 1$
    - 4.6.2.  $j \leftarrow n$
  - 4.7. Fim Se
  - 4.8. Se  $k = Dim\_Solução$  então  $Fim = verdadeiro$

**Fim Procedimento**

**Algoritmo 6.2.** Algoritmo para cálculo do critério de similaridade.

No algoritmo 6.2,  $Dim\_Solução$  é o número de elementos que compõem a solução. Por exemplo, o número de cidades envolvidas em um problema do caixeiro viajante.  $Num\_Operações$  representa o menor número de operações necessárias para se converter uma solução em outra.

### **6.2.4 Exemplos de Comportamento do Algoritmo**

A Tabela 6.1 apresenta os resultados correspondentes à melhor solução obtida pela copt-aiNet para 16 instâncias-padrão do problema do caixeiro viajante simétrico, correspondendo ao melhor indivíduo da população obtido em uma execução do algoritmo. Observe que, para a

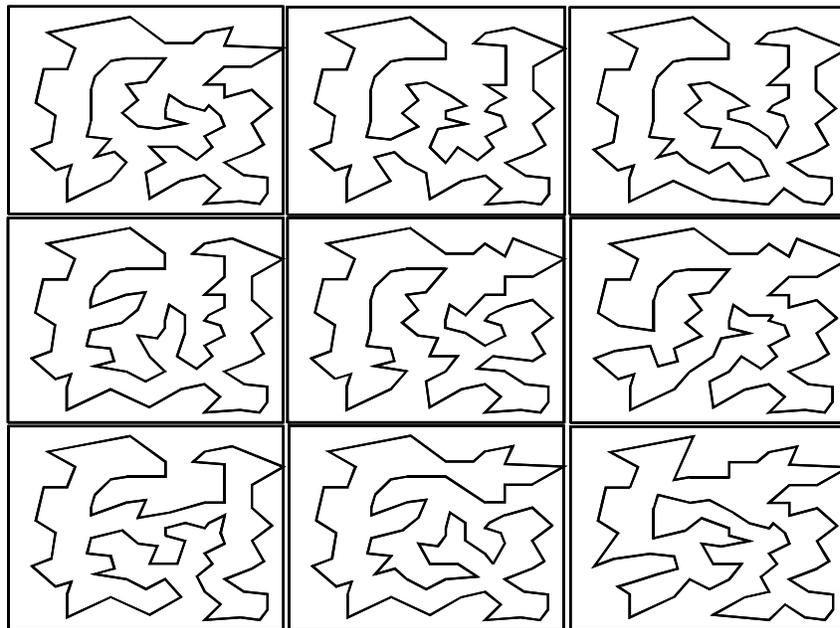
maioria dos casos, as soluções obtidas foram iguais às soluções ótimas conhecidas, evidenciando o potencial do método proposto.

**Tabela 6.1.** Melhores soluções obtidas pela *copt-aiNet* para 16 instâncias-padrão do problema do caixeiro viajante simétrico.

Instância	Número de cidades	Melhor solução conhecida	Melhor solução obtida
A280	279	2579	2583
BERLIN52	52	7542	7542
CH130	130	6110	6151
CH150	150	6528	6543
EIL101	101	629	629
EIL51	51	426	426
EIL76	76	538	538
KROA100	100	21282	21282
KROC100	100	20749	20749
KROD100	100	21294	21294
LIN105	105	14379	14379
PCB442	442	50778	51441
PR76	76	108159	108159
RD100	100	7910	7910
ST70	70	675	675
TSP225	225	3916	3934

Para ilustrar a habilidade da *copt-aiNet* em prover múltiplas soluções diversificadas para problemas de otimização combinatória, o algoritmo foi aplicado à instância EIL76 do caixeiro viajante simétrico. A instância foi extraída da TSPLIB, disponível no endereço <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.

A Figura 6.2 apresenta as nove soluções de menor custo obtidas pela *copt-aiNet* para a instância EIL76 do problema do caixeiro viajante. A Tabela 6.2 ilustra os custos, calculados utilizando-se operações com números inteiros, e as diversidades entre as nove soluções. Por exemplo, o número mínimo de operações para se chegar à solução 2 a partir da solução 1 é 17, e vice versa. Observe que a melhor solução obtida pela *copt-aiNet* corresponde ao ótimo conhecido para a instância EIL76.



**Figura 6.2.** Diversas soluções de alta qualidade obtidas pela copt-aiNet para a instância EIL76; os custos das soluções são: 538, 541, 542, 543, 544, 546, 548, 552, 553.

**Tabela 6.2.** Diversidade e custo de 9 soluções obtidas pela copt-aiNet para a instância EIL76.

Solução	1	2	3	4	5	6	7	8	9	Custo
<b>1</b>	0	17	19	19	14	13	23	15	18	538
<b>2</b>	17	0	14	16	18	17	17	21	27	541
<b>3</b>	19	14	0	17	22	24	17	19	24	542
<b>4</b>	19	16	17	0	21	22	12	13	25	543
<b>5</b>	14	18	22	21	0	13	21	18	24	544
<b>6</b>	13	17	24	22	13	0	27	23	22	546
<b>7</b>	23	17	17	12	21	27	0	19	27	548
<b>8</b>	15	21	19	13	18	23	19	0	24	552
<b>9</b>	18	27	24	25	24	22	27	24	0	553
Custo da melhor solução conhecida: 538										

A habilidade de explorar de forma eficiente o espaço de busca através de uma população diversificada e otimizada faz da copt-aiNet uma ferramenta poderosa, principalmente quando consideramos aplicações em que seja importante a obtenção de múltiplas soluções para um mesmo problema e também no caso de problemas com múltiplos objetivos, em que se busca amostrar a superfície de Pareto (COELLO & CORTÉS, 2005). Tal desempenho dificilmente pode ser alcançado via outras abordagens populacionais convencionais, tais como algoritmos

genéticos. Os algoritmos genéticos clássicos, bem como suas variantes, privilegiam os melhores indivíduos da população, tanto no momento do *crossover* como no processo de seleção dos indivíduos para a próxima geração (seleção elitista, *roulette wheel* etc.), o que tende a conduzir a uma população centrada no ponto de ótimo ou mesmo, em particular para casos em que a função a ser otimizada não é suficientemente suave, levar a um ponto de ótimo local de baixa qualidade.

### **6.3 Síntese do Capítulo**

Neste capítulo foi proposta uma nova ferramenta, denominada *copt-aiNet*, destinada à otimização combinatória multimodal. A *copt-aiNet* visa amenizar um problema inerente à maioria dos algoritmos evolutivos: a perda de diversidade da população.

Diversas estratégias vêm sendo propostas para manutenção da diversidade, tais como modelos de ilha (WHITLEY & HECKENDORN, 1999), nichos (MILLER & SHAW, 1996), entre outras. A principal vantagem da *copt-aiNet* em relação a outras abordagens evolutivas é sua habilidade de localizar e manter múltiplas soluções diversificadas para problemas de caráter combinatório de elevada complexidade, e isso ao longo de uma mesma execução do algoritmo. Esta propriedade faz da *copt-aiNet* uma ferramenta particularmente interessante em casos de otimização multimodal em que é requerida a obtenção de múltiplas soluções de alta qualidade, e para o caso de otimização multiobjetivo, em que múltiplas soluções correspondendo a diferentes objetivos são mantidas na população. Empregando apenas o algoritmo da seleção clonal, proposto por DE CASTRO & VON ZUBEN (2002), NARELI & COELLO (2005) já obtiveram resultados relevantes junto a uma variedade de instâncias de problemas multiobjetivo.

No Capítulo 7 desta tese é apresentada a aplicação da *copt-aiNet* ao problema de ordenação de dados de expressão gênica, que é modelado como um problema combinatório. Neste caso, freqüentemente há múltiplas relações entre genes que não podem ser expressas através de uma única solução, sendo portanto um problema tipicamente multimodal.

# CAPÍTULO 7

## ANÁLISE DE DADOS DE EXPRESSÃO GÊNICA

---

Neste capítulo são apresentados os resultados obtidos através da aplicação da *copt-aiNet*, proposta no Capítulo 6, ao problema de organização de dados de expressão gênica extraídos de experimentos de microarrays. Os resultados produzidos são comparados com aqueles obtidos através da clusterização hierárquica, um dos métodos mais comuns de clusterização aplicados à organização de dados de expressão gênica. Este capítulo assume o conhecimento acerca do procedimento de realização de experimentos de microarrays. Para mais informações acerca de experimentos de microarrays, bem como seu mapeamento para um conjunto de valores numéricos, consultar Apêndice C.

### 7.1 Introdução

Embora dados biológicos venham sendo produzidos a elevadas taxas, existe relativamente pouco conhecimento acerca da funcionalidade do genoma. Passadas cinco décadas da descoberta da estrutura em dupla hélice do DNA, o genoma humano e de diversos outros organismos, como *Escherichia coli*, *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*, já foram completamente seqüenciados. A análise dos níveis de expressão dos genes sob variadas condições pode prover importantes informações acerca da funcionalidade do genoma, uma vez que os níveis de expressão gênica estão diretamente associados à produção de proteínas, sendo portanto um fator determinante no controle de processos celulares. A medição dos níveis de expressão gênica pode auxiliar cientistas na identificação da função de genes, e de quais processos eles participam, além de contribuir para desvendar como os níveis de expressão são influenciados por doenças e medicações, entre muitas outras aplicações.

Nos últimos anos, a tecnologia de microarrays (SCHENA, 1996) tem atraído grande atenção de pesquisadores. Tais experimentos permitem o monitoramento de milhares de genes simultaneamente. Assim, a extração de conhecimento e análise da informação de níveis de expressão requer o processamento de grandes quantidades de dados, tornando impraticável a simples análise visual dos dados brutos. Dessa forma, ferramentas computacionais que auxiliem cientistas nesta tarefa vêm se tornando cada vez mais necessárias. Um método comumente empregado envolve a reordenação de genes de acordo com sua funcionalidade. Existem basicamente duas abordagens na literatura para efetuar esta tarefa: métodos de agrupamento (BAR-JOSEPH *et al.*, 2001; EISEN *et al.*, 1998) e métodos de ordenação (GOMES *et al.*, 2002; GOMES *et al.*, 2002f; GOMES *et al.*, 2002g; TSAI *et al.*, 2002). Nesta tese, a *copt-aiNet*, descrita no Capítulo 6, é empregada como um método de ordenação. O principal fator que levou à aplicação da *copt-aiNet* ao problema reside no fato de que, no contexto de análise de dados de expressão gênica, a manutenção da diversidade exerce um papel fundamental. Considerando que diversas proteínas podem apresentar simultaneamente diferentes funções no organismo, diferentes genes podem estar simultaneamente expressos em variados grupos. Os métodos comumente empregados não são capazes de detectar apropriadamente esses casos, de modo que relações entre genes condicionalmente co-regulados são frequentemente omitidas (GASH & EISEN, 2002). A habilidade da *copt-aiNet* de manter soluções diversificadas de alta qualidade permite a identificação de relações importantes entre genes.

## 7.2 Microarrays e Expressão Gênica

O objetivo dos experimentos de microarrays é comparar os níveis de expressão gênica de duas amostras de material genético submetidas a variadas condições ou experimentos. Para mais detalhes acerca de experimentos de microarrays, bem como acerca da terminologia empregada neste capítulo, consultar o Apêndice C. A visualização de dados de expressão gênica geralmente é realizada através de uma representação gráfica, como a representação proposta em EISEN *et al.* (1998). A representação gráfica requer uma matriz de expressão  $M$ , onde as linhas denotam genes e as colunas denotam experimentos. As cores e intensidades obtidas no array são mapeadas para a matriz de expressão  $M$ , onde  $M_{ij}$  é um valor real calculado para o

gene  $i$  e o experimento  $j$  (para detalhes consultar o Apêndice C). Portanto, a matriz de expressão  $M$  contém linhas correspondendo à expressão relativa de um gene específico e colunas representando os experimentos. Genes subexpressos corresponderão a valores negativos, enquanto genes sobreexpressos são representados por valores positivos. Genes igualmente expressos corresponderão ao valor zero.

### 7.3 Organizando Dados de Expressão Gênica

O resultado de um experimento de microarray em geral envolve um grande número de genes e experimentos. Assim, métodos que auxiliem cientistas a organizar os dados resultantes, permitindo a visualização de relações entre genes co-regulados, exercem um papel fundamental na análise de dados de expressão gênica. Uma técnica amplamente utilizada envolve o rearranjo dos dados de expressão gênica, reordenando os genes de acordo com sua funcionalidade. Existem basicamente duas abordagens na literatura para se realizar esta tarefa: métodos de clusterização (agrupamento) e ordenação linear. Diversos trabalhos podem ser encontrados na literatura para as duas abordagens, desde métodos baseados em algoritmos clássicos até novas meta-heurísticas. Essencialmente, tanto a clusterização como a ordenação linear resultam em um problema de otimização combinatória, cujo objetivo é reorganizar os genes de modo que genes apresentando perfis de expressão similares sejam posicionados próximos. Até o momento, estudos de expressão gênica têm focado métodos de agrupamento para classificar genes de acordo com os perfis de similaridade de expressão. Vários algoritmos de agrupamento foram aplicados ao problema. O mais comumente utilizado é a clusterização hierárquica (EISEN *et al.*, 1998).

Há outros métodos de agrupamento bastante empregados, como a clusterização por particionamento, geralmente implementada utilizando mapas auto-organizáveis (TAMAYO *et al.*, 1999) e  $k$ -means (GASH & EISEN, 2002).

Existem alguns trabalhos na literatura que tratam o problema de organizar dados de expressão gênica como um problema de ordenação linear. Em TSAI *et al.* (2002), os autores propuseram um algoritmo evolutivo para encontrar soluções de alta qualidade para o problema. Com o

objetivo de manter a diversidade da população, foi empregado um mecanismo em que cada indivíduo da população é considerado o “pai”, e é combinado com outros indivíduos escolhidos aleatoriamente, gerando  $\lambda$  soluções-filhas. Ao final do ciclo, os  $\mu$  indivíduos apresentando maior *fitness* são selecionados dentre os  $(\mu + \lambda)$  filhos gerados. Os indivíduos melhor adaptados formam a nova população e o processo se repete até que o critério de parada seja atingido. O método apresentou resultados satisfatórios, provendo, em geral, resultados de qualidade superior àqueles obtidos via clusterização hierárquica e mapas auto-organizáveis.

Em MERZ & ZELL (2002), os autores propuseram um algoritmo memético utilizando o procedimento de agrupamento *k*-means para efetuar uma busca local. Neste trabalho, o procedimento de agrupamento é visto como um problema combinatório no qual o vetor de genes deve ser associado a grupos de forma que a soma quadrática das distâncias dos vetores ao centróide de seu grupo seja mínima. Os resultados obtidos indicaram a superioridade do algoritmo memético com relação aos métodos clássicos de agrupamento.

Em COTTA *et al.* (2003), os autores aplicaram um algoritmo memético ao problema de ordenação gênica. As principais características da abordagem são o uso de uma função de *fitness* capaz de capturar propriedades globais do conjunto de genes, o uso de uma árvore para representação da solução, o uso de procedimentos de busca local e o uso de múltiplas populações em um modelo de ilhas com migração de indivíduos. Os resultados apresentados sugerem que o algoritmo memético apresenta melhor desempenho que os métodos de agrupamento clássicos.

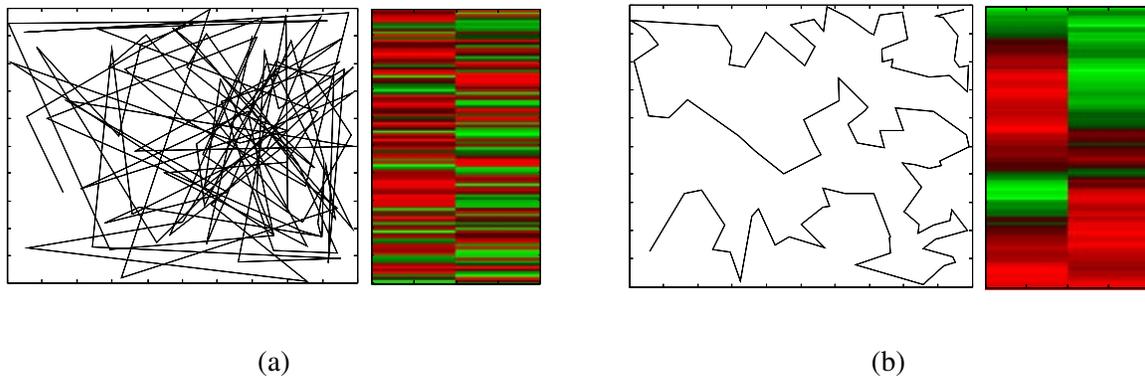
XIAO *et al.* (2003) propuseram uma abordagem evolucionária híbrida envolvendo mapas auto-organizáveis e otimização de partículas, de forma que os pesos sinápticos do mapa auto-organizável são determinados pelo algoritmo de otimização de partículas. Em um primeiro estágio, os pesos são obtidos através do treinamento do mapa auto-organizável. Em seguida, são otimizados pelo algoritmo de otimização de partículas com o objetivo de refinar o processo de agrupamento. Os autores proveram uma comparação entre o método proposto e a clusterização hierárquica clássica. Na maioria das simulações executadas, a hibridização entre mapa auto-organizável e otimização de partículas conduziu a melhores resultados.

Embora amplamente utilizada, a clusterização hierárquica é um método muito susceptível a mínimos locais. Algumas estratégias vêm sendo propostas na literatura para contornar esse aspecto, tais como reordenar as folhas da árvore gerada pelo procedimento de clusterização hierárquica (BAR-JOSEPH *et al.*, 2001, BIEDL *et al.*, 2001). Em geral, técnicas de agrupamento são bastante efetivas quando o conjunto de dados apresenta dimensão reduzida. Entretanto, quando se consideram dados de elevada dimensão, as relações entre genes tendem a ser múltiplas.

## 7.4 Técnicas de Agrupamento e de Ordenação

Técnicas de agrupamento organizam genes de modo que genes pertencentes a um mesmo grupo sejam similares de acordo com algum critério. Em geral, a escolha da técnica de agrupamento mais apropriada não é uma tarefa fácil. A maioria das técnicas requer que o número de grupos seja definido a priori, além de requerer o ajuste de diversos parâmetros. Um outro aspecto relevante é que os dados podem indicar grupos com diferentes formatos e tamanhos. Geralmente, métodos de agrupamento pressupõem a existência de *clusters* (grupos) bem definidos, além de suporem uma configuração espacial para os grupos. Diferentes técnicas de agrupamento tendem a produzir soluções distintas, e a melhor escolha é altamente dependente das características do conjunto de dados em análise. A técnica de agrupamento mais popular utilizada para análise de dados de expressão gênica é a *clusterização hierárquica* (EISEN *et al.*, 1998; JAIN *et al.*, 1999). A saída de um procedimento de clusterização hierárquica é uma árvore, tendo como folhas os perfis de expressão dos genes, de modo que cada folha corresponde a uma linha da matriz de expressão gênica  $M$ . A seqüência de folhas constitui um rearranjo das linhas da matriz de expressão gênica. Entretanto, um grande número de rearranjos é possível sem que sejam violadas as relações hierárquicas definidas por um procedimento de clusterização hierárquica. Assim, técnicas de pós-processamento se fazem necessárias para execução da permutação dos ramos da árvore, tal como proposto em BIEDL *et al.* (2001), onde um algoritmo de tempo polinomial para reorganizar as folhas da árvore é utilizado.

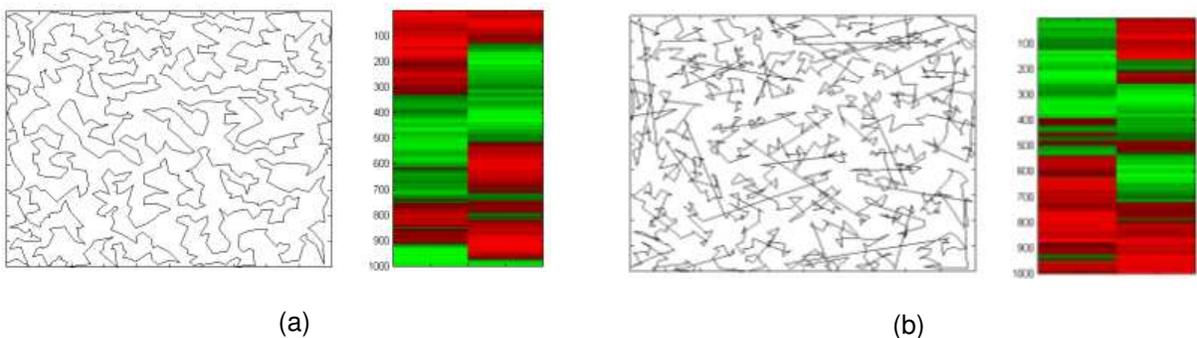
Métodos para ordenação buscam pela melhor ordenação linear para dados de expressão gênica. Nesse sentido, o problema de ordenação se torna equivalente ao problema do caixeiro viajante simétrico, apresentado no Capítulo 2. O objetivo é encontrar uma ordenação linear de genes que minimize a soma das distâncias (geralmente medida empregando-se a distância euclidiana ou o coeficiente de correlação (EISEN *et al.*, 1998)) entre genes adjacentes. Uma vez que o problema de reorganizar dados de expressão gênica é modelado como um problema de otimização combinatória, a meta é definir uma seqüência de genes, dentre uma enorme quantidade de candidatos, de modo que genes apresentando perfis de expressão similares no espaço  $N$ -dimensional (onde  $N$  é o número de experimentos) sejam posicionados o mais próximo possível (GOMES *et al.*, 2002). A Figura 7.3(a) ilustra a condição inicial (seqüência arbitrária de 102 genes e seus respectivos padrões de expressão para dois experimentos) e a Figura 7.3(b) mostra a ordenação obtida. São plotados os níveis de expressão dos dois experimentos no plano e é apresentada a sua correspondente representação gráfica antes e após a aplicação do procedimento de ordenação linear. Note que a representação gráfica na Figura 7.3(b) enfatiza a existência de genes com perfis de expressão similares, posicionando padrões similares topologicamente próximos. Tanto na Figura 7.3(a) como na Figura 7.3(b), a equação de cada gene corresponde a um ponto no gráfico da esquerda, e genes vizinhos na matriz de expressão da direita têm seus respectivos pontos ligados no gráfico da esquerda.



**Figura 7.3.** Exemplo de aplicação de um algoritmo de ordenação para um conjunto de expressões gênicas artificialmente geradas. (a) Ordenação inicial para 102 genes e 2 experimentos no plano e sua respectiva representação gráfica; (b) seqüência de expressões gênicas no plano e representação gráfica correspondente após a aplicação de um algoritmo de ordenação.

Devido à dificuldade de se selecionar o método de agrupamento mais apropriado, especialmente quando há pouca informação relacionada ao conjunto de dados em análise, métodos baseados em ordenação linear são potencialmente capazes de prover informações valiosas para a análise de especialistas, constituindo assim uma alternativa interessante para ser empregada isoladamente ou em conjunto com técnicas de agrupamento.

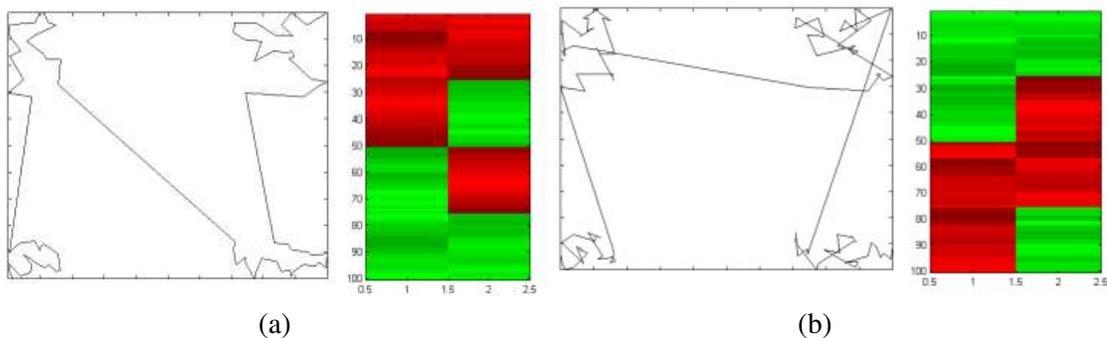
A Figura 7.4(a) mostra a ordenação para os níveis de expressão de 1000 genes gerados aleatoriamente com distribuição uniforme no plano bidimensional e a representação gráfica correspondente após o procedimento de ordenação linear. O mesmo conjunto após a aplicação do algoritmo de clusterização hierárquica aglomerativo é exibido na Figura 7.4(b). Observe que a configuração de genes, tendo sido aleatoriamente gerada, não caracteriza grupos bem definidos. Assim, o emprego de qualquer técnica de agrupamento tenderia a produzir clusters artificiais. Note também que, neste caso, a representação gráfica gerada a partir do algoritmo de ordenação linear apresentou um aspecto mais uniforme que aquele gerado via procedimento de clusterização hierárquica, justamente pelo fato de o conjunto de dados não apresentar clusters bem definidos, mas sim uma distribuição aleatória. Ligando-se os pontos correspondentes aos níveis de expressão de genes vizinhos (gráficos da esquerda, tanto na Figura 7.(a) como na Figura 7.4(b)), percebe-se que a ordenação apresentou melhor desempenho que a clusterização hierárquica.



**Figura 7.4.** Ordenação versus agrupamento para os níveis de expressão de um conjunto com 1000 genes gerados aleatoriamente e usando distribuição uniforme. (a) solução (custo 24) obtida por um algoritmo de ordenação e representação gráfica correspondente; (b) seqüência de genes (custo 40) no plano e respectiva representação gráfica após a aplicação do procedimento de clusterização hierárquica.

A Figura 7.5(a) ilustra o método de ordenação aplicado a uma configuração com 100 genes e 2 experimentos, cujos níveis de expressão foram artificialmente gerados e em que 4 grupos bem definidos no espaço bidimensional podem ser observados. A Figura 7.5(b) ilustra o resultado da aplicação da clusterização hierárquica ao mesmo conjunto de dados. Note que, em casos caracterizados por clusters bem definidos, o resultado da ordenação linear pode ser facilmente mapeado para um conjunto de clusters, uma vez que genes próximos na seqüência fazem parte de um mesmo cluster, enquanto genes distantes pertencem a clusters distintos. No caso apresentado na Figura 7.5(a), os clusters obtidos seriam exatamente aqueles definidos pela clusterização hierárquica, exibidos na Figure 7.5(b).

Note também que, conforme esperado, a representação gráfica obtida a partir da ordenação linear apresenta-se mais uniforme que aquela produzida pela clusterização hierárquica.

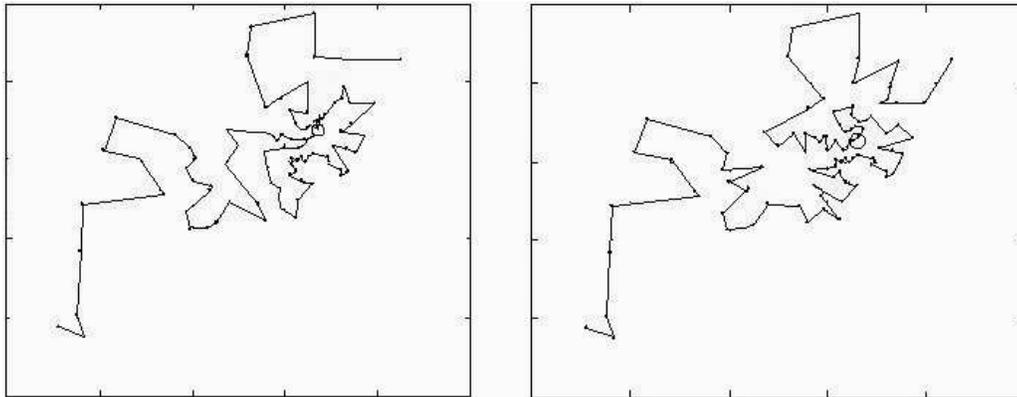


**Figura 7.5.** Ordenação linear e clusterização para quatro clusters bem definidos. (a) solução (custo 5,37) produzida por um algoritmo de ordenação para os níveis de expressão de 100 genes e 2 experimentos distribuídos em 4 clusters e representação gráfica correspondente; (b) seqüência de genes (custo 8,25) no plano e representação gráfica correspondente após aplicação da clusterização hierárquica.

## 7.5 Aplicação da copt-aiNet ao Problema de Ordenação de Dados de Expressão Gênica

### 7.5.1 Motivação

Atualmente é fato conhecido que muitos processos celulares são afetados pelas combinações genotípicas, havendo uma crescente demanda por métodos capazes de identificar relações múltiplas entre genes (PARK & HASTIE, 2005). A determinação de múltiplas soluções é particularmente interessante para o problema de análise de dados de expressão gênica. Considere as duas soluções exibidas na Figura 7.6, para uma instância com 100 genes e 2 experimentos. Os eixos de coordenadas correspondem aos níveis de ativação dos genes em cada experimento e cada ponto no plano representa o padrão de expressão de um gene. Como pode ser observado pelo pequeno círculo desenhado nas duas figuras (pouco à direita e acima do centro da figura), há dois genes possivelmente co-regulados. Entretanto, existe um longo caminho entre os dois genes na primeira solução, enquanto a segunda solução traz os genes muito próximos. Assim, pode-se concluir que algumas relações de genes correlacionados podem ser omitidas para representar outras relações, de modo que uma única solução não é capaz de prover uma visão de todas as relações possíveis entre genes.



**Figura 7.6.** Duas soluções para um problema no espaço bidimensional: cada solução evidencia diferentes configurações potencialmente capazes de representar diferentes relações entre genes, como exibido para os dois genes circulados.

### 7.5.2 Resultados Computacionais

Considerando que a clusterização hierárquica é o método de clusterização mais empregado na aplicação de organização de dados de expressão gênica, nesta seção são apresentados resultados envolvendo a *copt-aiNet* e um procedimento de clusterização hierárquica para três conjuntos de dados reais de expressão gênica:

- *Diffuse Large B-cell Lymphoma*: 380 genes e 63 experimentos da base de dados descrita em ALIZADEH *et al.* (2000). Os dados utilizados, bem como informações adicionais, podem ser encontrados em <http://lmpp.nih.gov/lymphoma/data/figure4>.
- *Yeast Saccharomyces cerevisiae*: 300 genes e 79 experimentos, constituindo um subconjunto da base de dados em EISEN *et al.* (1998). O artigo original de Eisen, bem como o conjunto de dados completo e informações adicionais, encontram-se disponíveis em <http://rana.jbl.gov/EisenData.htm>.
- *Yeast Saccharomyces cerevisiae*: 500 genes e 79 experimentos, constituindo outro subconjunto da base de dados provida em EISEN *et al.* (1998).

Todos os experimentos contendo valores ausentes foram preenchidos com o valor '0'. O conjunto de dados empregado nesta seção encontra-se disponível na página do *Infobiosys Research Group* <http://www.dca.fee.unicamp.br/projects/infobiosys/geneexp>.

Para ilustrar a habilidade da *copt-aiNet* na solução do problema de organização de dados de expressão gênica, foi realizado um estudo comparativo envolvendo a *copt-aiNet* e um procedimento de clusterização hierárquica seguido pela reordenação das folhas da árvore gerada. A ferramenta *TreeArrange* (BIEDL *et al.*, 2001) foi empregada para reordenar as folhas da árvore.

Os resultados aqui apresentados correspondem aos custos das melhores soluções alcançadas com o software *TreeArrange*,  $Custo_{TA}$ , e o custo de várias soluções obtidas pela *copt-aiNet*. O custo de uma solução é calculado como sendo a soma das distâncias euclidianas entre cada par

de genes adjacentes. Além disso, é importante comparar a similaridade entre as soluções obtidas pela *copt-aiNet*, calculada de acordo com o critério apresentado no Capítulo 6, Seção 6.2.3.1. Um valor inteiro correspondendo ao número de operações necessárias para se converter uma solução em outra representa a diversidade entre pares de soluções: valores maiores indicam soluções menos similares, enquanto valores menores indicam soluções mais similares. Caso duas soluções correspondam à mesma ordenação, a diversidade para este par é igual a zero.

As Tabelas 7.1, 7.2 e 7.3 apresentam os resultados para oito indivíduos da população evoluída via *copt-aiNet* em termos de diversidade. O custo da solução final é comparado com o custo da solução obtida via o software *Tree-Arrange* ( $Custo_{TA}$ ). Cada tabela contém os resultados dos algoritmos para um dos três conjuntos de dados escolhidos para validação da proposta desta tese. Conforme pode ser observado nas Tabelas 7.1, 7.2 e 7.3, para todos os casos, as soluções obtidas via *copt-aiNet* não somente apresentaram maior qualidade que a solução gerada via o software *Tree-Arrange* como também apresentaram alto grau de diversidade. Conforme mencionado no Capítulo 6, uma das principais vantagens da *copt-aiNet* em relação a outras abordagens é que ela é capaz de prover múltiplas soluções de alta qualidade para o problema em estudo, o que conduz à possibilidade de identificação de genes co-regulados. Assim como ilustrado na Figura 7.4 para o caso bidimensional, as soluções obtidas não são caracterizadas pela alocação distinta dos mesmos grupos, mas sim pela proposição de novos grupos.

**Tabela 7.1.** Diversidade e custo das melhores soluções obtidas para um problema com 380 genes e 63 experimentos da base de dados *Diffuse Large B-cell Lymphoma*. Os custos correspondem à soma das distâncias euclidianas entre cada par de genes adjacentes.

Solução	1	2	3	4	5	6	7	8	Custo
1	0	102	87	102	99	91	118	107	1827,25
2	102	0	107	108	102	105	93	115	1827,69
3	87	107	0	93	104	108	108	123	1829,07
4	102	108	93	0	99	110	117	117	1829,18
5	99	102	104	99	0	111	117	100	1829,58
6	91	108	105	110	111	0	115	117	1830,44
7	118	93	108	117	117	115	0	130	1830,83
8	107	115	123	117	100	117	130	0	1831,40
$Custo_{TA}$ :									1909,50

**Tabela 7.2.** Diversidade e custo das melhores soluções obtidas para um problema com 300 genes e 79 experimentos da base de dados *Yeast Saccharomyces cerevisiae*. Os custos correspondem à soma das distâncias euclidianas entre cada par de genes adjacentes.

Solução	1	2	3	4	5	6	7	8	Custo
1	0	78	72	71	92	82	96	93	1070,54
2	78	0	85	88	101	102	94	108	1072,80
3	72	85	0	84	89	90	95	110	1072,81
4	71	88	84	0	88	94	95	101	1074,52
5	92	101	89	88	0	93	106	110	1074,66
6	82	102	90	94	93	0	107	105	1075,59
7	96	94	95	95	106	107	0	110	1075,73
8	93	108	110	101	110	105	110	0	1077,09
<i>Custo<sub>TA</sub></i> :									1120,10

**Tabela 7.3.** Diversidade e custo das melhores soluções obtidas para um problema com 500 genes e 79 experimentos da base de dados *Yeast Saccharomyces cerevisiae*. Os custos correspondem à soma das distâncias euclidianas entre cada par de genes adjacentes.

Solução	1	2	3	4	5	6	7	8	Custo
1	0	177	182	174	179	174	182	172	1913,84
2	177	0	167	170	171	177	187	183	1916,80
3	182	167	0	186	177	194	200	187	1917,31
4	174	170	186	0	189	187	183	190	1917,70
5	179	171	177	189	0	176	164	192	1918,35
6	174	177	194	187	176	0	121	200	1919,00
7	182	187	200	183	164	121	0	215	1919,79
8	172	183	187	190	192	200	215	0	1920,77
<i>Custo<sub>TA</sub></i> :									2016,10

# CAPÍTULO 8

## CONCLUSÕES E PERSPECTIVAS

---

Neste capítulo são apresentadas as principais contribuições desta tese e são apontadas algumas linhas de pesquisa relacionadas a possíveis extensões deste trabalho.

### 8.1 Contribuições

Nos últimos anos, as meta-heurísticas têm desempenhado um papel fundamental em otimização combinatória. O uso de métodos exatos, ou mesmo métodos aproximados que oferecem garantia de proximidade da solução ótima, é limitado pela elevada demanda por recursos computacionais. Para grande parte dos problemas reais, que em geral apresentam alta complexidade, os métodos exatos se mostraram intratáveis, o que tem motivado a pesquisa na área de métodos heurísticos.

As principais contribuições desta tese residem na proposta de duas meta-heurísticas para problemas combinatórios baseadas em diferentes paradigmas:

- Redes neurais artificiais e sistemas nebulosos: foi proposto um novo método empregando múltiplos mapas auto-organizáveis para a solução de problemas de roteamento de veículos e de múltiplos caixeiros viajantes. O método emprega conceitos de inferência nebulosa para guiar o treinamento dos mapas auto-organizáveis.
- Engenharia imunológica: nesta linha, foi proposto um novo método, denominado *copt-aiNet* (*Artificial Immune Network for Combinatorial Optimization*) e baseado na *opt-aiNet* (*Artificial Immune Network for Optimization*), proposta por DE CASTRO &

TIMMIS (2002). A *copt-aiNet* está voltada para a solução de problemas multimodais em espaços de busca discretos.

Foram dois os principais motivos que levaram ao estudo da aplicação de mapas auto-organizáveis a problemas combinatórios:

1. São escassos os trabalhos envolvendo auto-organização na solução de problemas combinatórios. Alguns trabalhos apresentam resultados promissores para instâncias simples do problema do caixeiro viajante simétrico. Entretanto, a carência de um estudo mais aprofundado, considerando problemas combinatórios de maior complexidade, é evidente.
2. O custo computacional tende a ser substancialmente menor que para outros algoritmos baseados em permutação, principalmente quando tratamos instâncias de larga escala. Embora a qualidade da solução obtida tenda a ser menor que das soluções obtidas por meta-heurísticas baseadas em permutação, tais como algoritmos genéticos ou sistemas imunológicos artificiais, os mapas auto-organizáveis constituem uma alternativa atraente para aplicações de larga escala e em tempo real, ou para aquelas em que existe um elevado grau de incerteza na definição do problema. Uma outra vantagem da abordagem auto-organizável é que ela é extremamente flexível no tratamento de certos tipos de restrições.

Nesta tese é proposta uma extensão dos algoritmos apresentados na literatura para o problema de roteamento de veículos capacitados (CVRP – *Capacitated Vehicle Routing Problem*) e para o problema de múltiplos caixeiros viajantes (MTSP – *Multiple Traveling Salesman Problem*). Entretanto, devido à elevada complexidade do CVRP e do MTSP, o mapa auto-organizável convencional, proposto por KOHONEN (1997), não pôde ser diretamente aplicado ao problema, tal como ocorre para o problema do caixeiro viajante simétrico, tendo sido necessária a proposição de uma nova estratégia de treinamento do mapa guiada por inferência nebulosa. O desempenho das estratégias nebulosa e convencional foi comparado através de

uma série de simulações, mostrando que a abordagem nebulosa conduz a resultados substancialmente melhores.

A segunda principal linha de contribuições desta tese reside na proposição de uma nova meta-heurística baseada em sistemas imunológicos artificiais e destinada à otimização de problemas combinatórios. O método desenvolvido representou uma extensão da opt-aiNet (Artificial Immune Network for Optimization), proposta por DE CASTRO & TIMMIS (2002) para otimização multimodal de problemas não lineares em espaços contínuos. O método foi adaptado de modo a operar em espaços discretos. Uma das características mais relevantes do método proposto é a habilidade de encontrar e manter múltiplas soluções de alta qualidade e adequadamente distribuídas no espaço de soluções candidatas. O critério para definir boas soluções inclui, além da qualidade relativa da solução, sua diversidade junto à população como um todo, indo além da mera análise do valor da função-objetivo. É estabelecido um critério de similaridade entre soluções, permitindo um gerenciamento eficiente da população de indivíduos. Esta habilidade confere maior autonomia aos algoritmos, com a maioria dos parâmetros automaticamente ajustável pelo próprio algoritmo. Tal autonomia é derivada justamente da inspiração no sistema imunológico biológico, em que se busca manter um repertório de anticorpos diversificado, com eliminação de células redundantes e apto a responder prontamente a patógenos aos quais o organismo foi previamente exposto, bem como a patógenos similares a estes.

No Capítulo 7 foram apresentados resultados promissores provenientes da aplicação da opt-aiNet a um dos problemas mais relevantes do campo da bioinformática: análise de dados de expressão gênica provenientes de experimentos de microarrays. Dado um conjunto de genes submetidos a uma série de experimentos ou em diferentes estágios de um processo (diferentes etapas do ciclo de vida celular, por exemplo), o experimento de microarray permite identificar genes que se encontram ativados ou expressos em cada experimento. O principal objetivo é estabelecer relações entre os níveis de expressão dos genes. Um aspecto importante é que vários genes podem estar co-regulados, ou seja, se encontram expressos em um mesmo processo ou experimento simultaneamente. Neste caso, diferentes soluções (ordenações) podem ser associadas a genes co-regulados. Considerando que o estudo de processos co-

regulatórios é de fundamental importância para a compreensão das funcionalidades do genoma, a obtenção de múltiplas soluções diversificadas é particularmente útil para análise de expressão gênica.

## **8.2 Perspectivas e Pesquisas Futuras**

O trabalho desenvolvido ao longo da elaboração desta tese admite várias vertentes, cada uma delas podendo conduzir ao desenvolvimento de uma linha de pesquisa.

### ***8.2.1 Extensão dos métodos apresentados para o problema do caixeiro viajante assimétrico***

Uma extensão imediata da *copt-aiNet* e dos mapas auto-organizáveis consiste em sua adaptação de modo a suportar instâncias assimétricas do caixeiro viajante.

### ***8.2.2 Emprego do mapa auto-organizável proposto em outras aplicações***

No Capítulo 4 foi proposta uma nova abordagem baseada em inferência nebulosa para treinamento de mapas auto-organizáveis, podendo inclusive envolver múltiplos mapas simultaneamente. Um tópico para pesquisa interessante seria avaliar o desempenho da rede proposta para outras aplicações, como problemas de agrupamento (clusterização), mineração de dados, otimização multiobjetivo, já que o mapa auto-organizável proposto permite que se considerem vários objetivos ponderados embutidos na função de similaridade, entre outros.

### ***8.2.3 Otimização multiobjetivo e de problemas dinâmicos empregando a *copt-aiNet****

Atualmente, é fato bem conhecido que abordagens populacionais têm se mostrado eficazes no tratamento de problemas com múltiplos objetivos, pois, além de levarem a uma exploração eficiente do espaço de busca, permitem que se obtenham vários elementos do conjunto de Pareto simultaneamente, em uma única execução do algoritmo. Abordagens evolutivas, tais como algoritmos genéticos e sistemas imunológicos artificiais, vêm sendo amplamente utilizadas nesse contexto (COELLO & NARELI, 2005). Conforme mencionado na seção anterior,

a *copt-aiNet* dispõe de um mecanismo elaborado de manutenção da diversidade da população. Tal característica torna a *copt-aiNet* particularmente atraente para otimização multiobjetivo e de problemas dinâmicos mono e multiobjetivo. Alguns trabalhos já vêm sendo desenvolvidos no contexto de otimização dinâmica dentro do grupo de pesquisa do LBiC (Laboratório de Bioinformática e Computação Bio-Inspirada), vinculado ao DCA/FEEC/Unicamp. Em DE FRANÇA *et al.* (2006), a *copt-aiNet* é aplicada à otimização de instâncias do problema do caixeiro viajante simétrico dinâmico, em que as ponderações das arestas (custos de se viajar entre cidades) é variável. Nesta publicação, o desempenho da *copt-aiNet* foi comparado àquele obtido por uma implementação do método de colônia de formigas. Conforme esperado, foi constatado que a *copt-aiNet* apresentou melhores resultados no ambiente dinâmico e, além disso, consumindo menos recursos computacionais.

#### **8.2.4 Aplicação da *copt-aiNet* a problemas de bioinformática**

Conforme apresentado no Capítulo 7, a *copt-aiNet* foi aplicada com sucesso a problemas de ordenação de dados de expressão gênica. Foram empregadas algumas bases de dados públicas disponíveis via Web. Os resultados obtidos indicaram que a *copt-aiNet* é uma ferramenta potencialmente útil na organização de dados de expressão gênica, levando à produção de múltiplas soluções diversificadas e, observando-se o diagrama, visualmente distintas. Por envolver pré-processamento de dados e recursos gráficos para visualização, muitas outras ferramentas computacionais de apoio ao desenvolvimento de interfaces com o usuário poderiam ser incorporadas, assim como o estabelecimento de uma interação efetiva com potenciais usuários. Além disso, existem muitos outros problemas combinatórios em bioinformática (ALURU, 2005) junto aos quais a *copt-aiNet* pode ser aplicada.

#### **8.2.5 Ensembles**

Uma linha de pesquisa futura interessante envolve o conceito de *ensembles*, que são compostos por uma combinação de um conjunto de componentes sintetizados para se realizar uma tarefa. Uma das principais motivações para o emprego de *ensembles* é o fato de que diferentes propostas de solução podem explorar diferentes aspectos relevantes de um

problema, enquanto muitas vezes uma única proposta de solução não é capaz de explorar todos os aspectos relevantes simultaneamente (LIMA, 2004). Assim, a combinação de diferentes métodos ou mesmo o emprego de diferentes conjuntos de parâmetros em ensembles leva à construção de poderosas ferramentas, potencialmente capazes de prover uma exploração mais eficiente do espaço de busca. Tal abordagem geralmente leva ao aumento da qualidade da solução final, à obtenção de um conjunto de soluções finais mais diversificado, e à identificação de aspectos relevantes existentes em diversas propostas de solução.

Considerando a habilidade da *copt-aiNet* em produzir uma população de soluções altamente diversificadas, o método tende a constituir um poderoso componente na composição de ensembles, abrindo uma vertente a ser explorada.

### **8.2.6 Meta-otimização**

A última e possivelmente mais desafiadora linha de pesquisa futura apontada nesta tese envolve o conceito de meta-otimização. Em geral, diferentes métodos são freqüentemente necessários para abordar diferentes classes de problemas de otimização ou mesmo instâncias distintas pertencentes a uma mesma classe. Por exemplo, nesta tese foram apresentadas situações em que os mapas auto-organizáveis tendem a ser mais efetivos que as abordagens baseadas em permutação (quando a restrição envolve tamanho relativo das rotas), e situações em que abordagens baseadas em permutação são mais indicadas.

A existência de tantos tipos de métodos de otimização e variações é consequência da observação prática e teórica de que a eficiência de um determinado método de otimização é dependente do problema a ser tratado. Tal fato leva a duas questões: como determinar qual técnica de otimização é mais adequada a uma instância específica de um problema? Como caracterizar um problema através de atributos que permitam identificar o método de otimização apropriado? A resposta a estas questões pode ser vista como um complexo problema de otimização, comumente chamado de *meta-otimização*.

# APÊNDICE A

## COMPLEXIDADE COMPUTACIONAL

---

Neste apêndice serão brevemente apresentados alguns conceitos relacionados a complexidade computacional. A importância das meta-heurísticas tornou-se evidente no início da década de 1971-1980, quando descobriu-se que muitos problemas importantes de otimização não podiam ser resolvidos na otimalidade em tempo polinomial (PETRANK, 1995). A importância prática dos problemas combinatórios levou pesquisadores a relaxar os requisitos de garantia de convergência e garantia de obtenção da solução ótima e investigar métodos de otimização aproximados baseados em meta-heurísticas.

### A.1. Dificuldade de Problemas: Problemas P, NP, NP-Completos e Reduções

A teoria da complexidade computacional estuda a complexidade de problemas computáveis através de um algoritmo. A complexidade de um problema está diretamente ligada ao consumo de tempo e memória de um algoritmo ótimo ou do melhor algoritmo conhecido para o problema. O consumo de tempo e memória é medido em função do tamanho da instância. De acordo com sua complexidade, um problema pode ser *polinomial* ou *não-polinomial*. Um problema polinomial possui complexidade dada por  $O(n^k)$  para algum  $k$ , onde  $n$  é o tamanho da instância, ou seja, um algoritmo é dito executar em tempo polinomial ou consumir uma quantidade polinomial de memória para este problema se existe um polinômio  $p(n)$  tal que o algoritmo aplicado a uma entrada de dimensão  $n$  sempre encontra uma solução para o problema em tempo  $p(n)$ , isto é, após efetuar  $p(n)$  instruções simples, ou então consumindo

uma quantidade  $p(n)$  de espaço em memória. Um problema é dito não-polinomial caso não exista  $k$  tal que a complexidade do problema seja dada por  $O(n^k)$ , seja no tempo, no espaço, ou em ambos.

A classe de problemas P engloba todos os problemas que podem ser resolvidos em tempo polinomial através de um algoritmo. A classe de problemas NP (do inglês *non-deterministic polynomial*) é o conjunto de todos os problemas de decisão que admitem um algoritmo verificador. Um algoritmo verificador constitui um algoritmo capaz de executar em tempo polinomial um processo de verificação da veracidade de uma condição dada uma solução qualquer para o problema. Por exemplo, dados os números naturais  $a$ ,  $b$  e  $c$ , verificar se existe um valor  $d$  tal que  $a \cdot d^3 + b \cdot d^2 + c \cdot d = 0$ . Neste caso, com apenas sete operações aritméticas pode-se verificar a veracidade da condição expressa pela equação acima. Tomando agora como exemplo o problema do caixeiro viajante, para verificar se o problema é NP deve-se, primeiramente, formulá-lo como um problema de decisão. Por exemplo: dado um conjunto de cidades  $C = \{1, \dots, N\}$  e as distâncias  $d_{i,j}$  entre as cidades  $i$  e  $j$ , deseja-se verificar se o trajeto dado pela seqüência  $s$  de cidades apresenta distância total menor que um valor denotado por  $Dist$ . O algoritmo A.1 ilustra o procedimento de verificação, sendo capaz de verificar a veracidade da condição em tempo polinomial, provando que o problema é NP.

Observe que o processo de verificação não exige que se encontre uma solução polinomial para o problema NP, mas sim que se determine em tempo polinomial se uma condição específica é verdadeira ou não.

Um problema de decisão  $E$  é classificado como NP-completo se todo problema da classe NP é redutível a  $E$  em tempo polinomial. Exemplos de problemas NP-completos são a versão de decisão do problema do caixeiro viajante e o problema do ciclo hamiltoniano, entre muitos outros. Problemas de otimização cujas versões de decisão são NP-completas são classificados como NP-difíceis. A categorização NP-completo refere-se a problemas de decisão, enquanto a categorização NP-difícil diz respeito à versão de otimização.

Embora não haja uma prova formal até a presente data, é consenso entre especialistas da área que  $P \neq NP$ , ou seja, conjectura-se que problemas da classe NP-completo não podem ser resolvidos em tempo polinomial. Um problema  $P1$  é classificado como NP-completo se  $P1$  pertence à classe NP-completo e qualquer outro problema em NP pode ser reduzido a  $P1$  em tempo polinomial. Um problema  $P2$  é redutível a um problema  $P1$  se um algoritmo para  $P1$  pode ser transformado em tempo polinomial em um algoritmo para  $P2$ . Existem diversos problemas NP-completos: o problema da mochila e o problema do caixeiro viajante, entre outros. Em GAREY & JOHNSON (1979), pode ser encontrada uma relação de problemas já classificados como NP-completos. O problema do caixeiro viajante foi um dos primeiros a ser provado como pertencente à classe NP-completo, sendo que nenhum dos algoritmos já propostos é capaz de encontrar a solução ótima em tempo polinomial.

**Procedimento Checa\_Solução( $d, s, Dist$ )**

DistânciaTotal := 0

**Para**  $i:=1$  até  $n-1$  **faça**    DistânciaTotal = DistânciaTotal +  $d_{i,i+1}$ **Fim**DistânciaTotal = DistânciaTotal +  $d_{n-1,n}$ **Se** DistânciaTotal <  $Dist$  **então**

Verificação := Verdadeiro

**Senão**

Verificação := Falso

**Fim do Procedimento**

**Algoritmo A.1.** Verificação da veracidade de uma versão de decisão para o problema do caixeiro viajante: dadas as distâncias entre todas as cidades, a seqüência de cidades  $s$  corresponde a um percurso cuja distância total é menor do que  $Dist$ ?

# APÊNDICE B

## O SISTEMA IMUNOLÓGICO

---

Neste apêndice será apresentada uma breve revisão acerca do sistema imunológico biológico dos vertebrados, enfatizando alguns conceitos que serviram como base para o desenvolvimento de uma nova abordagem de inteligência computacional: os sistemas imunológicos artificiais (SIA).

### B.1. Conceitos Básicos

O sistema imunológico é composto por um conjunto de moléculas, células e órgãos que trabalham de forma agregada com o objetivo principal de proteger um indivíduo de invasores externos causadores de doenças (patógenos), como vírus, bactérias, fungos e parasitas. O corpo é provido de barreiras naturais que constituem mecanismos de defesa inespecíficos, agindo contra uma ampla variedade de agentes agressores. Esses elementos que atuam de forma genérica ou não-específica compõem o *sistema imunológico inato*. O sistema inato está presente desde o nascimento, não é aperfeiçoado com o histórico de exposições a agentes patológicos e baseia-se, principalmente, na ação de células fagocitárias (macrófagos e neutrófilos) capazes de ingerir e digerir microorganismos sem a necessidade de exposição prévia ao agente patogênico. Dessa forma, indivíduos nascem com uma capacidade de resposta imunológica relativamente fraca, fundamentada em uma imunidade natural gerada pelos anticorpos recebidos da mãe.

Por sua vez, o *sistema imunológico adaptativo* age em resposta específica a um determinado agente infeccioso e possui a habilidade de se adaptar no sentido de aperfeiçoar sua resposta ao longo do tempo, de acordo com o histórico de exposições a agentes capazes de promover uma resposta imunológica. O sistema adaptativo possui a habilidade de aprender a reconhecer tipos

específicos de patógenos e de manter uma memória com o intuito de prover respostas futuras mais rápidas e eficazes ao mesmo patógeno e a algumas variantes deste patógeno.

O processo de aprendizagem ou adaptação imunológica tem início na chamada *resposta primária*. A resposta primária é em geral lenta, tornando-se aparente apenas alguns dias após a infecção. Já a memória é constituída por um conjunto de células e moléculas (anticorpos) com ciclos prolongados de vida. Caso o corpo seja reinfestado pelo mesmo agente ou agentes similares, o sistema imunológico o reconhece com base na memória imunológica, sendo capaz de prover uma resposta rápida e eficaz, denominada *resposta secundária*. Dessa forma, a principal diferença entre as respostas adaptativa e inata reside no fato de que a resposta inata permanece aproximadamente constante durante toda a vida do indivíduo, enquanto a resposta adaptativa é incrementada e aperfeiçoada com a exposição repetida a antígenos. Um antígeno pode ser definido como qualquer substância reconhecida pelo sistema imunológico e capaz de desencadear uma resposta imunológica, a qual pode acarretar ou não a produção de anticorpos.

Os sistemas imune inato e adaptativo, em conjunto, compõem um notável e complexo mecanismo de defesa. Quando o sistema inato não é capaz de combater um determinado agente infeccioso, o sistema adaptativo gera uma resposta. Neste ponto vale ressaltar que, embora haja a distinção entre sistema imune inato e adaptativo, os mecanismos atuadores ativados por uma resposta específica fazem uso de células próprias do sistema inato, ou seja, ambos os sistemas atuam em conjunto para manter o equilíbrio interno do organismo.

As respostas imunológicas, sejam elas provenientes do sistema inato ou adaptativo, dependem da ação das células brancas do sangue, os leucócitos.

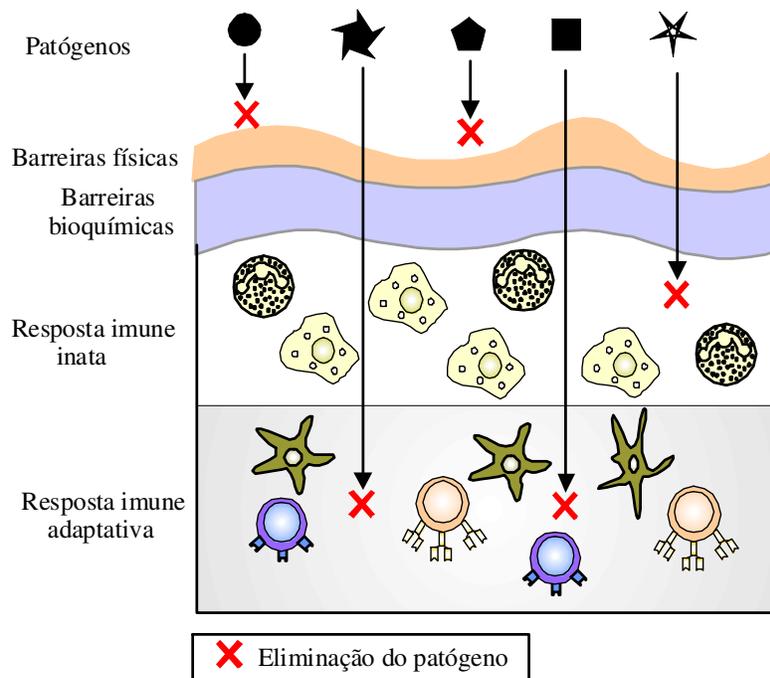
## **B.2. Estrutura do Sistema Imunológico, Elementos Constituintes e Mecanismos de Defesa**

### ***B.2.1. Estrutura do Sistema Imunológico***

O sistema imunológico é organizado em camadas, estabelecendo uma estrutura com defesas em múltiplos níveis. A primeira camada é composta por barreiras físicas, tais como pele,

membranas e pêlos. A segunda camada envolve barreiras bioquímicas, como saliva, lágrima, enzimas e ácidos, propiciando condições de vida inadequadas a certos tipos de invasores. As demais camadas são formadas pelos sistemas imune inato e adaptativo. A Figura B.1 apresenta uma visão geral da estrutura do sistema imunológico, com as várias camadas de defesa do organismo.

Uma vez que o agente patogênico entrou no organismo, o sistema imunológico inato constitui a primeira linha de defesa, sendo capaz de eliminar muitos microorganismos comuns. Porém, nem sempre o sistema inato é capaz de reconhecer e/ou combater eficazmente os organismos infectantes, uma vez que existem muitos organismos que ele não é capaz de reconhecer. Nestes casos entra em ação o sistema imunológico adaptativo.

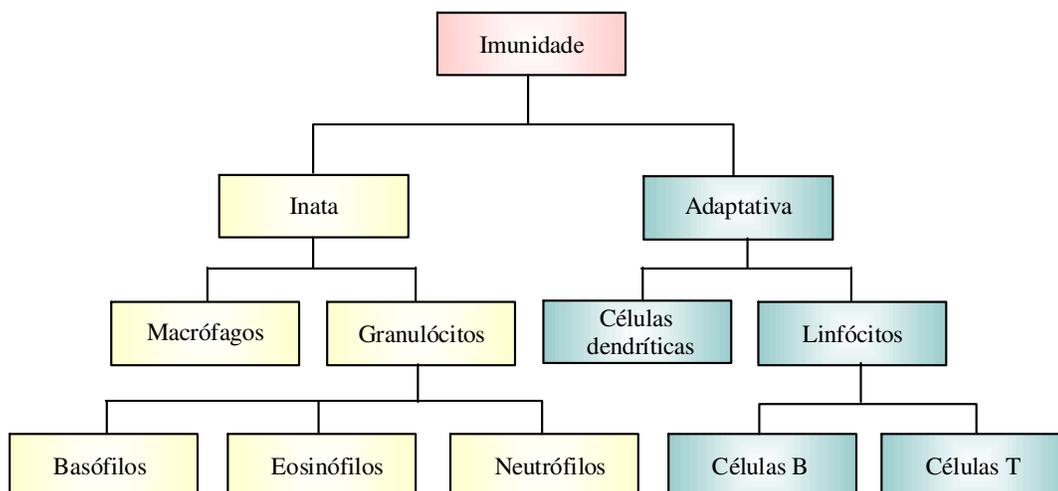


**Figura B.1.** Estrutura do sistema imunológico.

### B.2.2. Principais Células do Sistema Imunológico

Os processos de detecção e eliminação de patógenos são guiados por afinidades físico-químicas. As células do sistema imunológico possuem superfícies providas de receptores, alguns dos quais aptos a se ligarem a agentes patogênicos, outros aptos a se ligarem a células do sistema imunológico e outros aptos a reconhecer estímulos moleculares. Todos estes receptores possuem o objetivo comum de prover reconhecimento, comunicação e sinalização entre células e moléculas, promovendo assim um gerenciamento eficiente da resposta imunológica.

Tanto o sistema imunológico inato como o adaptativo dependem da ação dos *leucócitos* ou *glóbulos brancos*. Os leucócitos são responsáveis pelo reconhecimento e eliminação de patógenos. Os *linfócitos* constituem leucócitos mononucleares, e podem ser encontrados principalmente no sangue e em alguns órgãos linfóides (ver Seção B.2.3). Os linfócitos compõem o principal grupo de células imunológicas investigadas em sistemas imunológicos artificiais. Outras variações de leucócitos que desempenham um papel fundamental no mecanismo de defesa são os *macrófagos*, *granulócitos* e *células dendríticas*. A Figura B.2 ilustra as principais células atuantes no sistema imunológico.

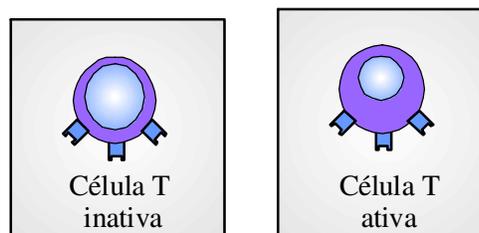


**Figura B.2.** Principais células do sistema imunológico participantes das defesas inata e adaptativa.

As principais funções dos linfócitos, macrófagos, granulócitos e células dendríticas encontram-se descritas a seguir:

- **Linfócitos T ou células T** – Desempenham um importante papel na produção de uma resposta imunológica. Ao serem ativadas, se diferenciam em células T efetoras, com uma variedade de funções. Participam da resposta adaptativa e são assim chamadas devido ao fato de sua maturação se dar no timo (ver Seção B.2.3). Podem ser de dois tipos:
  - célula T auxiliar (*helper T-Cell*), que tem por função promover respostas imunológicas através da secreção de substâncias químicas que ativam outras células imunológicas, como os macrófagos (os quais são responsáveis por digerir microorganismos e outros elementos antigênicos) e alertando as células B quanto à necessidade de iniciar a produção de anticorpos;
  - célula T citotóxica ou supressora (*killer T-Cell*), que efetua a eliminação de células infectadas por agentes patológicos. Não possuem especificidade para antígenos.

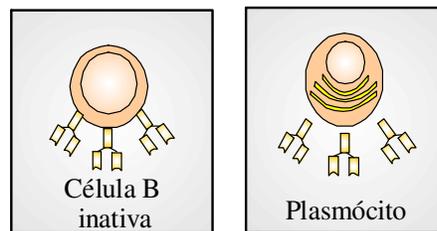
As células T são providas de receptores antigênicos para detectar antígenos derivados de proteínas estranhas ao organismo ou patógenos. A Figura B.3 mostra a representação adotada nesta tese para uma célula T inativa, responsável pela detecção de antígenos, e uma célula T ativa, responsável por alertar outras células do sistema imunológico acerca da presença do antígeno reconhecido.



**Figura B.3.** Representação para célula T inativa e célula T ativa.

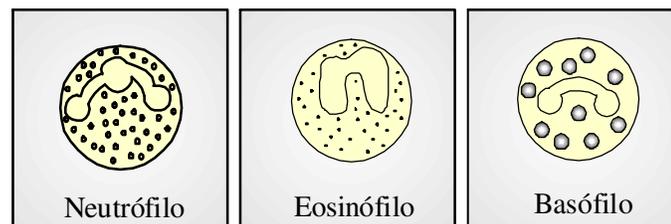
- **Linfócitos B ou células B** – Quando ativadas, essas células diferenciam-se em células plasmáticas secretoras de anticorpos (plasmócitos). Sua principal função é a produção

de anticorpos em resposta a agentes patogênicos. Os anticorpos constituem proteínas que reconhecem e se ligam a uma outra proteína específica, sendo portanto capazes de reconhecer e se ligar a agentes patogênicos. Cada célula B é programada para produzir um anticorpo específico, exercendo assim um papel fundamental na imunidade adaptativa. A Figura B.4 ilustra a representação de uma célula B inativa e uma célula B ativa (plasmócito).



**Figura B.4.** Representação para célula B inativa e célula B ativa (plasmócito).

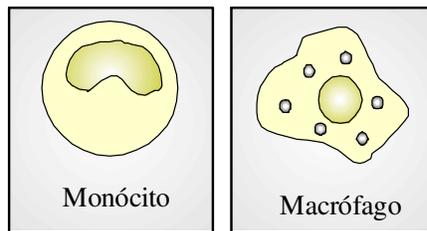
- **Granulócitos ou leucócitos polimorfonucleares** – São importantes na remoção de parasitas e agentes invasores do organismo. Os corpos estranhos são fagocitados e degradados por enzimas produzidas pelos granulócitos. Há três tipos de granulócitos: os neutrófilos, os eosinófilos e os basófilos. Os granulócitos circulam pelo sangue e entram nos tecidos somente quando alertados pela presença de pontos de infecção ou inflamação. A Figura B.5 ilustra os três tipos de granulócitos: neutrófilos, eosinófilos e basófilos.



**Figura B.5.** Representações para células granulócitas: neutrófilo, eosinófilo e basófilo.

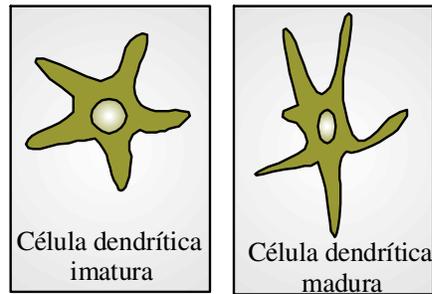
- **Macrófagos** – Leucócito agranulócito correspondendo à forma madura dos monócitos. Os monócitos circulam pelo sangue e, ao alcançarem os tecidos, se diferenciam em macrófagos. Os macrófagos desempenham ação fagocitária, apresentando capacidade de

circular pelo corpo e digerir microorganismos e outros elementos antigênicos. Possuem um importante papel tanto no sistema imunológico inato quanto no adaptativo. Os macrófagos podem também apresentar antígenos às células T, ativando-as. Essa característica faz com que sejam denominados células apresentadoras de antígenos (APC – antigen presenting cell). A Figura B.6 exibe a representação de um monócito (macrófago imaturo) e de um macrófago.



**Figura B.6.** Representação para monócito e monócito maduro (macrófago).

• **Células dendríticas** – são também denominadas “células apresentadoras de antígenos” (APCs), uma vez que são responsáveis pela captura do antígeno e por sua apresentação aos linfócitos para o procedimento de reconhecimento. São células fagocitárias especializadas, de vida duradoura e baixa taxa de proliferação. Encontram-se presentes na maioria dos tecidos. As células dendríticas imaturas migram do sangue para os tecidos, onde realizam fagocitose e pinocitose. Possuem receptores em sua superfície capazes de reconhecer estruturas comuns a vários patógenos. Ao encontrarem um patógeno, maturam rapidamente e migram para os linfonodos, onde ativam os linfócitos T antígeno-específicos. Assim, as células dendríticas são fagocitárias enquanto ainda estiverem imaturas. Quando amadurecem, passam a exercer o papel de apresentadoras de antígenos aos linfócitos. A principal função de uma célula dendrítica não é destruir o patógeno, mas sim levar os antígenos patogênicos aos órgãos linfóides periféricos para apresentação aos linfócitos T. A Figura B.7 mostra uma representação pictórica de uma célula dendrítica imatura e madura, respectivamente.



**Figura B.7.** Representação para célula dendrítica imatura e madura.

### **B.2.3. Os Órgãos do Sistema Imunológico**

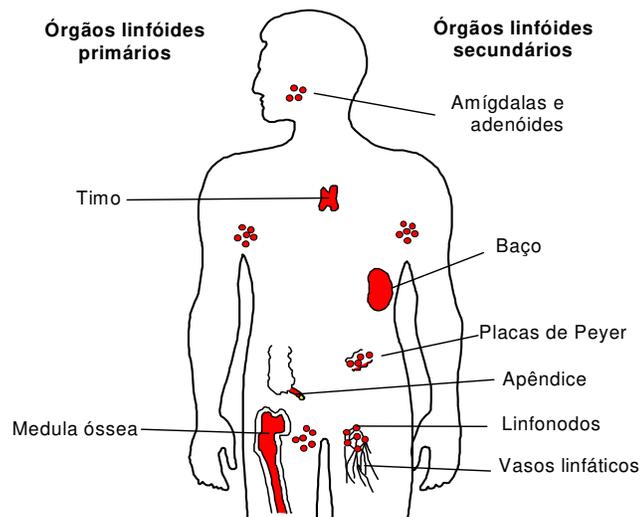
Os órgãos do sistema imunológico, denominados *órgãos linfóides*, classificam-se em primários ou secundários. Os órgãos linfóides primários são responsáveis pela produção e maturação dos linfócitos, enquanto nos órgãos linfóides secundários ocorrem os estímulos antigênicos. Nos órgãos linfóides, os linfócitos interagem com outras células e com elementos antigênicos, o que pode ocorrer tanto durante o processo de maturação como durante uma resposta imunológica. A Figura B.8 ilustra os órgãos linfóides. A medula óssea e o timo constituem os órgãos linfóides primários. Os órgãos linfóides secundários são as amígdalas e adenóides, linfonodos, apêndice e placas de Peyer, vasos linfáticos e baço.

As principais funcionalidades de cada órgão encontram-se detalhadas a seguir:

- **Medula óssea** - todas as células do sistema imunológico derivam da medula óssea. Algumas células amadurecem na própria medula óssea, enquanto outras migram e sofrem maturação em outros órgãos linfóides. A medula óssea produz células B, que constituem células produtoras de anticorpos, e células T. Entretanto, somente as células B se diferenciam na própria medula óssea. Uma vez que as células B têm seu processo de maturação finalizado, elas entram na corrente sanguínea, migrando para os órgãos linfóides secundários.
- **Timo** - a principal função do timo é a produção de células T maduras. Células T imaturas migram da medula óssea e amadurecem no timo. Durante o processo de maturação, células

T que são benéficas para o sistema imunológico proliferam-se, enquanto células que possam acarretar uma resposta auto-imune (ou seja, uma resposta contra o próprio organismo) são eliminadas. As células T maduras são liberadas para o sangue e para a linfa, através dos quais percorrem o organismo em busca de agentes patogênicos.

- **Baço** - é o maior órgão linfóide secundário. Possui a função de filtração do sangue, combatendo os microorganismos patogênicos que invadem a corrente sanguínea.
- **Linfonodos** - constituem órgãos linfáticos secundários. Desempenham o papel de filtros para um fluido corporal conhecido como *linfa*. São localizados em pontos de convergência dos vasos do sistema linfático. Os linfonodos abrigam as células fagocitárias que promovem a filtração da linfa, eliminando antígenos antes que estes retornem à circulação sanguínea.
- **Amígdalas e adenóides** – fazem parte do sistema imunológico associado a mucosas e ao intestino.
- **Apêndice e placas de Peyer** – contêm células imunológicas destinadas à proteção do trato gastrointestinal.
- **Vasos linfáticos** – constituem um conjunto de canais que transportam a linfa dos tecidos para o sangue e para os órgãos linfóides, carregando as células portadoras de antígenos dos locais de infecção para o reconhecimento nos órgãos linfóides.



**Figura B.8.** Órgãos linfóides.

## **B.2.4. Mecanismos Básicos de Defesa**

### **B.2.4.1. Imunidade Inata**

Os microorganismos que superam as barreiras físico-químicas e penetram no organismo entram em contato com células e moléculas que podem desencadear uma resposta imune inata. O sistema imune inato reconhece uma ampla gama de substâncias (antígenos).

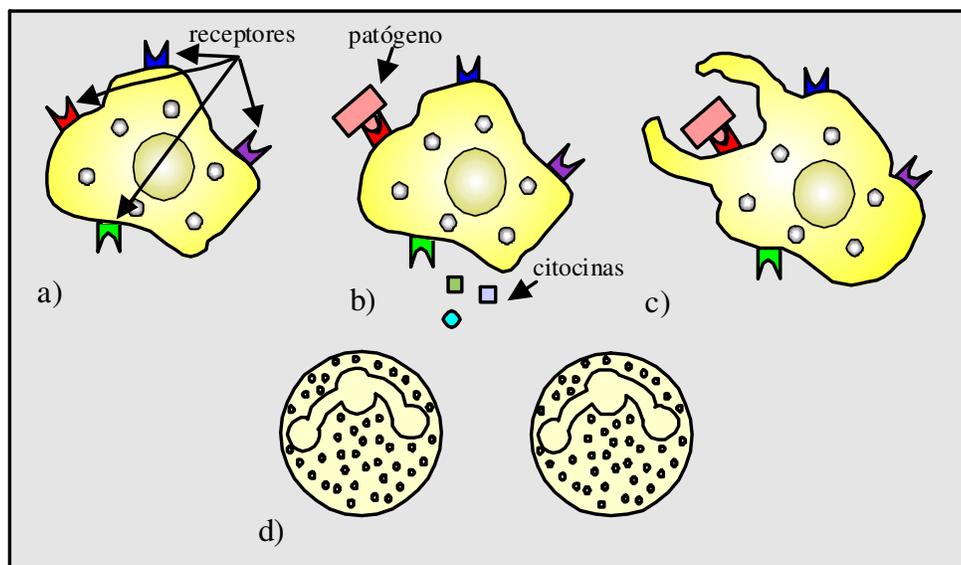
A ação de uma célula fagocitária encontra-se ilustrada na Figura B.9. Os macrófagos e neutrófilos são os principais elementos atuantes na defesa inata. Eles identificam os patógenos através de seus receptores de superfície, os quais se ligam ao patógeno (Figuras B.9 a) e b)). As moléculas bacterianas que se ligam a esses receptores levam a célula a englobar a bactéria (Figura B.9 c)). Células apresentadoras de antígenos, tais como os macrófagos, circulam pelo corpo em busca de agentes patogênicos. Quando um patógeno é encontrado, ele é ingerido e fragmentado em *peptídeos antigênicos* (NOSSAL, 1993).

Os macrófagos são também encarregados de alertar outras células do sistema inato sobre a presença de um patógeno. Isso é feito através da secreção de certas quimiocinas denominadas *citocinas* (Figura B.9 b)), as quais são identificadas por outras células providas de receptores

para quimiocinas (Figura B.9 d)), tais como neutrófilos e monócitos (que rapidamente se diferenciam em macrófagos).

A resposta inata é fundamental para a ocorrência de uma resposta adaptativa: as células dendríticas e os macrófagos realizam a apresentação de antígenos aos linfócitos T.

Um outro aspecto importante do sistema inato é sua notável capacidade de discriminar de forma efetiva células próprias e superfícies de patógenos, distinguindo entre o *próprio* e o *não próprio*, passando também essa informação para o sistema adaptativo.



**Figura B.9.** Visão de uma resposta do sistema imunológico inato.

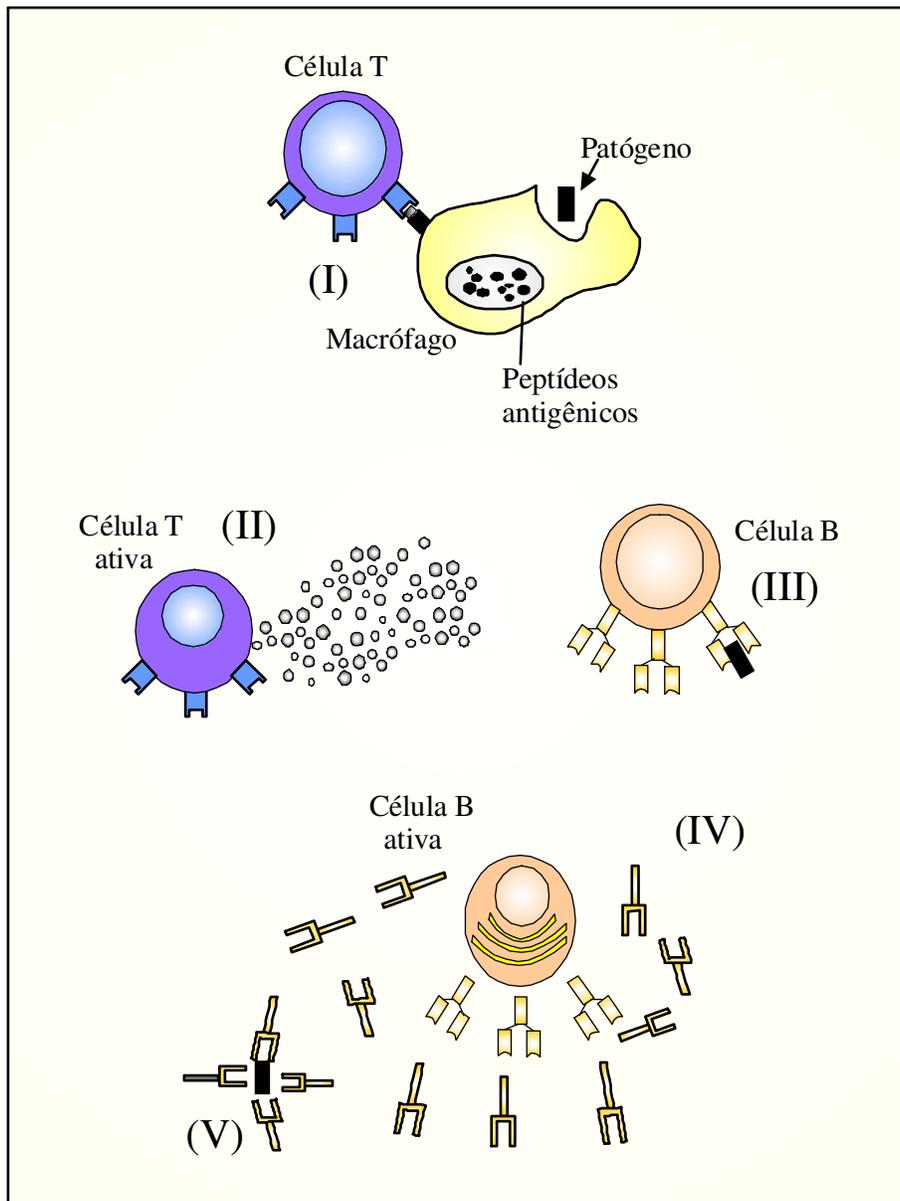
#### B.2.4.2. Imunidade Adaptativa

A defesa adaptativa é promovida pelos linfócitos B e linfócitos T, apresentando uma resposta específica para cada patógeno. Na resposta adaptativa, as células reconhecem um microorganismo como invasor e respondem a ele através da produção de anticorpos específicos, promovendo a eliminação do antígeno. Ela também conduz à geração de células de memória, propiciando uma proteção imunológica permanente ou de longo prazo para novas exposições ao mesmo antígeno ou a antígenos similares.

A resposta adaptativa tem início quando um patógeno é ingerido por uma célula dendrítica imatura. Eventualmente, todas as células dendríticas migram através da linfa em direção aos linfonodos, onde interagem com os linfócitos. Se a célula dendrítica não estiver ativada, ela induz tolerância aos antígenos próprios que elas carregam.

A Figura B.10 ilustra as principais etapas de uma resposta adaptativa. As células T auxiliares possuem em sua superfície receptores específicos capazes de reconhecer o antígeno através dos peptídeos antigênicos ligados ao macrófago (I). Nessa etapa é importante ressaltar que cada linfócito apresenta receptores com uma única especificidade. Embora tenha uma única especificidade, os receptores são desenvolvidos de modo que possam reconhecer também variações dessa especificidade, assegurando o reconhecimento de um antígeno mesmo mediante pequenas mutações. Isso garante que, com os linfócitos trabalhando em conjunto, possa-se dar origem a inúmeras especificidades, formando o *repertório de receptores de linfócitos* do indivíduo. A Seção B.3 detalha um dos processos de reconhecimento de padrões do sistema imunológico.

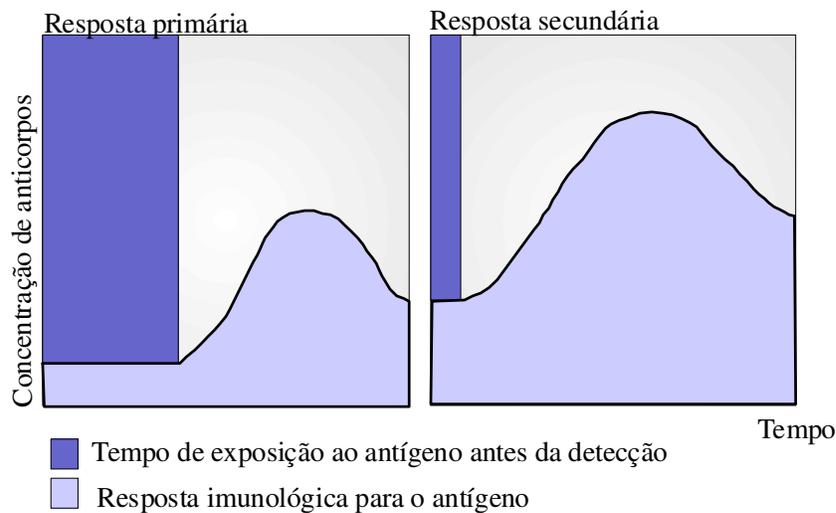
Uma vez que o antígeno foi identificado, as células T se proliferam e alertam outras células de defesa sobre a presença do antígeno através da liberação de substâncias denominadas *linfocinas* (II). As linfocinas desencadeiam a ativação de células B. Quando uma célula B é ativada pelas linfocinas (III), ela dá origem a diversas *células plasmáticas*, que fabricam os anticorpos específicos para o antígeno (IV). Os anticorpos se desprendem da célula B e se ligam aos antígenos, inativando o agente patogênico e sinalizando o início de um processo de fagocitose (V).



**Figura B.10.** Mecanismos básicos de defesa do sistema imunológico.

Quando o sistema imunológico detecta a presença de um determinado antígeno pela primeira vez, existe um período de latência para que anticorpos específicos sejam produzidos. Se, posteriormente, o organismo entrar em contato novamente com o mesmo antígeno, anticorpos específicos estarão presentes e atuando muito mais rapidamente e em maior quantidade. A

Figura B.11 ilustra a concentração de anticorpos nas respostas primária e secundária ao longo do tempo.



**Figura B.11.** Concentração de anticorpos nas respostas primária e secundária ao longo do tempo.

### B.3. Reconhecimento de Padrões

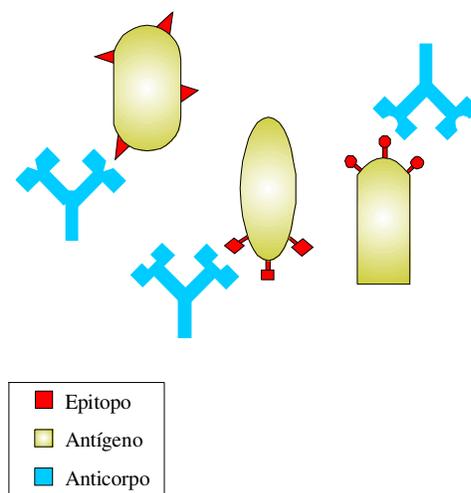
Um dos aspectos mais importantes do sistema imunológico com relação ao reconhecimento de padrões é que diversas células, como os linfócitos, são providas de moléculas receptoras capazes de efetuar o reconhecimento de antígenos. O receptor do linfócito B é capaz de interagir com moléculas antigênicas livres, enquanto o receptor do linfócito T é capaz de reconhecer antígenos fragmentados em peptídeos antigênicos, os quais são apresentados pelas células apresentadoras de antígenos.

O receptor de antígeno da célula B é o anticorpo ligado à sua membrana de superfície. Caso a célula seja ativada por um agente patogênico, os anticorpos serão secretados. Os anticorpos apresentam uma região denominada *paratopo* destinada ao reconhecimento de antígenos específicos. A região do antígeno que se liga ao paratopo é chamada *epítopo*. A intensidade da ligação entre paratopos e epítopos é denominada *afinidade*. Os receptores são ditos específicos devido ao fato de se ligarem somente a estruturas de epítopos complementares. Os receptores

de linfócitos distintos podem diferir entre si, mas os receptores de um mesmo linfócito são idênticos, o que indica que um determinado linfócito é específico para epítomos similares. Um linfócito somente será ativado se sua afinidade com os patógenos for suficientemente alta.

Enquanto o anticorpo possui um único tipo de receptor, os antígenos podem apresentar múltiplos epítomos, podendo ser simultaneamente reconhecido por diversos anticorpos.

A Figura B.12 ilustra o processo de reconhecimento dos antígenos por parte dos anticorpos através dos epítomos.



**Figura B.12.** Processo de reconhecimento de antígenos através dos epítomos.

#### B.4. A Teoria da Seleção Clonal

A seleção clonal dos linfócitos é um dos princípios mais importantes da imunidade adaptativa (JANEWAY, 2001). Com o objetivo de produzir células efetoras em quantidade suficiente para o combate de uma infecção, um linfócito ativado deve se proliferar. A teoria da seleção clonal (BURNET, 1959) propõe que, quando um antígeno invade o organismo, células imunológicas que reconhecem esse antígeno se proliferam, gerando novas células denominadas *clones*. Quanto maior for a afinidade entre o anticorpo e o antígeno, mais clones serão gerados. Durante a proliferação, os clones estão sujeitos a mutações com taxas inversamente proporcionais à afinidade com o antígeno (NOSSAL, 1993). Os clones mutados com maior

afinidade são selecionados via processo de seleção natural para se tornarem células de memória ativadoras de respostas imunológicas secundárias futuras. As células de memória circulam pela linfa, pelo sangue e pelos tecidos e, quando são novamente expostas ao mesmo estímulo, rapidamente se diferenciam em plasmócitos capazes de produzir anticorpos.

Uma das principais características da seleção clonal é a habilidade de eliminação de linfócitos capazes de reagir a antígenos próprios, correspondentes a padrões antigênicos expressos pelo próprio organismo. Linfócitos auto-reativos ou apresentando baixa afinidade são neutralizados antes mesmo de seu amadurecimento.

Uma outra importante característica é a geração de mutações genéticas via mecanismo de hipermutação somática (KEPLER & PERELSON, 1993). Dessa forma, os clones gerados são submetidos a processos de mutação, produzindo novos tipos de anticorpos.

A seleção clonal ocorre tanto para as células B como para as células T. As células T não produzem anticorpos, mas são fundamentais para a adequada ativação das células B.

Em síntese, as principais propriedades da seleção clonal são:

- proliferação de células que possuem receptores específicos de alta afinidade a um antígeno correspondente a um determinado agente patogênico;
- inativação de linfócitos que reagem a antígenos próprios, expressos pelo próprio organismo;
- geração de variações genéticas aleatórias (hipermutações) durante o processo de clonagem;
- promoção de um processo de maturação de afinidade.

A Figura B.13 ilustra a teoria da seleção clonal.

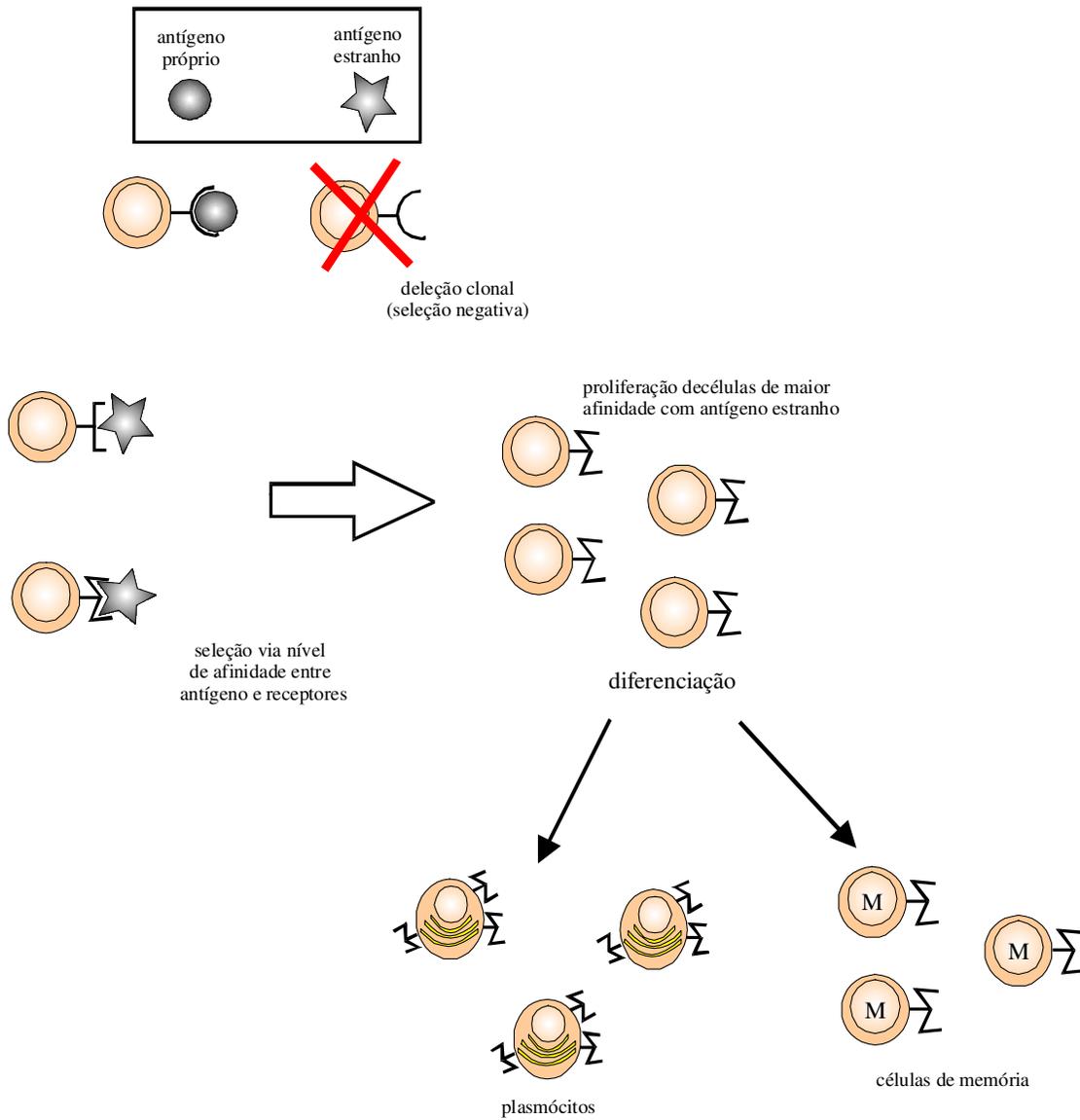


Figura B.13. Mecanismos envolvidos na teoria da seleção clonal.

## B.5. Diversidade do Receptor

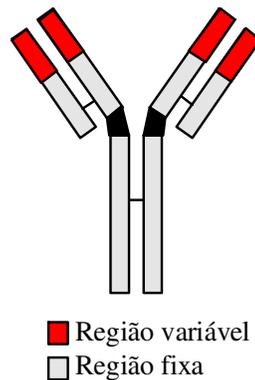
Os anticorpos apresentam uma região fixa e uma região variável, conforme ilustrado na Figura B.14. A região fixa pode assumir apenas um número limitado de estados bem definidos, enquanto a região variável pode assumir uma variedade praticamente infinita de formas distintas, permitindo que se liguem a um número também praticamente infinito de antígenos.

As regiões variáveis são idênticas entre si para qualquer molécula de anticorpo e determinam a especificidade de ligação ao antígeno. A parte constante determina as ações que devem ser tomadas para inocular o patógeno. A diversidade dos anticorpos é, portanto, garantida pelas regiões variáveis da molécula.

Durante o desenvolvimento das células B, os mecanismos de recombinação e mutação se dão de forma orquestrada. Primeiramente, ocorre a recombinação dos segmentos gênicos da molécula de anticorpo, gerando uma população celular de ampla diversidade em termos de especificidade a antígenos. O mecanismo de hipermutação somática é então ativado para operar durante a proliferação (clonagem), podendo levar ao aumento da afinidade entre o anticorpo e o antígeno.

No genoma de cada indivíduo existem múltiplos segmentos gênicos que codificam parte do receptor. Esses segmentos devem ser agrupados de modo a formar uma molécula de anticorpo completa no linfócito B. Assim, existem “bibliotecas gênicas” que, através dos procedimentos de recombinação e mutação, formam os receptores dos linfócitos. Em resumo, o repertório de células B é diversificado através de três mecanismos: hipermutação somática, deleção de genes e recombinação de genes. Mutações aleatórias são introduzidas nos genes da região variável durante o processo de clonagem das células B. Eventualmente, algum clone modificado pode apresentar uma maior afinidade ao antígeno do que a célula que o gerou. Essas células com alta afinidade tornam-se *células de memória*. Devido à aleatoriedade do processo de mutação, uma grande parcela dos elementos mutados torna-se não funcional ou auto-reativa e deve ser eliminada em estágios preliminares do processo de maturação.

O sistema imunológico deve apresentar um repertório de receptores suficientemente variado de modo a assegurar que os linfócitos respondam a variados patógenos. Considerando que o número de patógenos a serem reconhecidos é muito maior que o número de receptores existentes, o repertório de receptores seria insuficiente para reconhecer cada um dos possíveis patógenos se o repertório fosse fixo. Sendo assim, a diversidade e a constante renovação de partes do repertório de células imunológicas são elementos fundamentais para garantir uma defesa efetiva contra um universo praticamente infinito de patógenos.



**Figura B.14.** Regiões fixa e variável de um anticorpo.

## B.6. Teoria da Rede Imunológica

A *teoria da rede imunológica*, originalmente proposta por JERNE (1974), sugere que o sistema imunológico é composto por uma rede de células e moléculas capazes de se reconhecerem independentemente da presença de antígenos. Foi demonstrado experimentalmente que os anticorpos, além de possuírem em sua região variável (região-V) receptores, conhecidos como paratopos, para reconhecimento de antígenos, possuem também seu próprio determinante antigênico (epítopo). Os epítopos dos anticorpos permitem que eles sejam reconhecidos por outros linfócitos do próprio sistema imunológico. As células imunológicas podem responder positivamente ou negativamente a um determinado estímulo. Uma resposta positiva conduzirá à proliferação e ativação de células e produção de anticorpos, ao passo que uma resposta negativa resultará em um processo de *supressão*. Conforme descrito na Seção B.7, durante o processo de maturação certos anticorpos podem se tornar auto-reativos. Basicamente, quando uma célula imunológica reconhece um antígeno ou então uma outra célula imunológica, ela é estimulada. Por outro lado, quando uma célula imunológica é reconhecida por outra célula imunológica, ela é suprimida. Quando um organismo promove uma reação imunológica aos antígenos próprios, ele pode sofrer de uma doença *auto-imune*. Portanto, a eliminação ou neutralização de linfócitos diferenciados capazes de reagir com padrões antigênicos expressos pelo próprio organismo exerce um papel fundamental. Assim, a característica central da teoria da rede imunológica é a definição da identidade molecular do indivíduo.

A teoria da rede imunológica é particularmente interessante para manutenção da diversidade e controle do tamanho da população de células imunológicas através de um processo auto-organizado, sendo assim extremamente útil na construção de ferramentas computacionais para solução de problemas de engenharia. Recorrendo ao conceito de espaço de formas, a teoria da rede imunológica pode ser interpretada como um processo auto-organizado que define regiões de tolerância e reatividade no espaço de formas.

## **B.7. Maturação de Afinidade**

A maturação de afinidade está relacionada ao aumento de afinidade entre anticorpos e antígenos durante uma resposta imunológica, e está diretamente ligada ao processo de mutação dos genes que codificam o anticorpo (ver Seção B.5), seguido pelo processo de seleção clonal (ver Seção B.4). Tanto o processo de mutação como o de seleção ocorrem nos órgãos linfóides secundários.

A maturação de afinidade pode ser claramente observada em uma resposta imunológica secundária: os anticorpos envolvidos em uma resposta secundária apresentam em média maior afinidade aos antígenos do que os anticorpos presentes na resposta primária. Tal fato é consequência dos processos de hipermutação somática seguida pelos mecanismos de seleção natural de células contendo anticorpos de mais alta afinidade aos antígenos. Portanto, os mecanismos de seleção clonal, manutenção de diversidade e memória imunológica são fundamentais para a maturação da resposta imunológica.

# APÊNDICE C

## EXPRESSÃO GÊNICA

---

Neste apêndice serão apresentados alguns conceitos básicos de biologia molecular e uma breve introdução acerca da técnica de microarrays. Os microarrays representam o estado da arte em termos de tecnologia molecular para geração de dados de expressão gênica, permitindo que se extraiam os níveis de expressão de milhares de genes em um único experimento. Tal possibilidade levou a uma revolução no escopo do projeto genoma. Juntamente com essa revolução, emergiu a necessidade de organizar e interpretar grandes quantidades de dados. A bioinformática vem desempenhando um papel fundamental nessa tarefa. Atualmente, diversas fontes de bases de dados e softwares para análise estão disponíveis on-line. Este apêndice traz também algumas referências importantes e apresenta a interpretação das matrizes de expressão gênica para o formato comumente encontrado nos repositórios on-line.

### C.1. Introdução

Embora a produção de dados biológicos esteja ocorrendo a elevadas taxas, existe relativamente pouco conhecimento acerca da funcionalidade do genoma. Cinco décadas após a descoberta da estrutura em dupla hélice do DNA, o genoma humano e o genoma de diversos organismos, como *Escherichia coli*, *Saccharomyces cerevisiae*, *Caenorhabditis elegans* e *Drosophila melanogaster*, foram completamente seqüenciados.

Após o seqüenciamento completo do genoma humano, a descoberta das funções das proteínas constitui uma das mais desafiadoras linhas de pesquisa no campo da biologia molecular. Apesar de a seqüência de aminoácidos de uma proteína ser definida pelas informações contidas no DNA, não é possível obter o proteoma apenas conhecendo-se o genoma. Enquanto

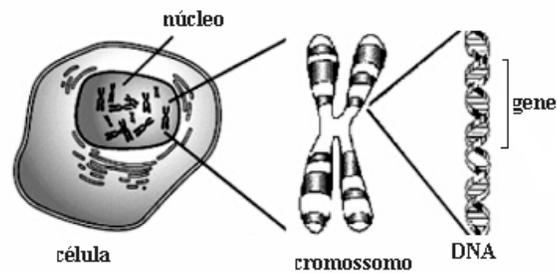
o genoma de um organismo é praticamente invariante para todas as células, cada tipo de célula aciona apenas uma parte dos genes para formar proteínas, ou seja, somente uma parte de seus genes é expressa. A análise dos níveis de expressão de genes sob variadas condições pode fornecer conhecimento acerca da funcionalidade do genoma. Os níveis de expressão dos genes encontram-se estritamente relacionados à produção de proteínas, sendo, portanto, um fator determinante no controle de diversos processos celulares. A medição dos níveis de expressão gênica em larga escala tem auxiliado cientistas em diversas áreas de pesquisa, como identificação do papel de genes, especificação de processos em que os genes atuam e análise de efeitos de medicamentos. Por essas razões, a análise de expressão gênica constitui um dos tópicos mais estudados na área de bioinformática (LUSCOMBE *et al.*, 2001).

## **C.2. Conceitos Básicos de Biologia Molecular**

### **C.2.1. O Genoma e os Genes**

As células são consideradas as unidades vitais dos seres vivos, pois nelas é realizada a maioria dos processos metabólicos e está contido o material genético. Células contêm o genoma, que contém genes, que são compostos de DNA (ácido deoxirribonucléico). Embora cada célula contenha uma cópia do genoma completo, nem todos os genes estarão expressos e diferentes genes poderão estar expressos em diferentes circunstâncias, caracterizando, por exemplo, diferentes tecidos, doenças e fases do ciclo celular.

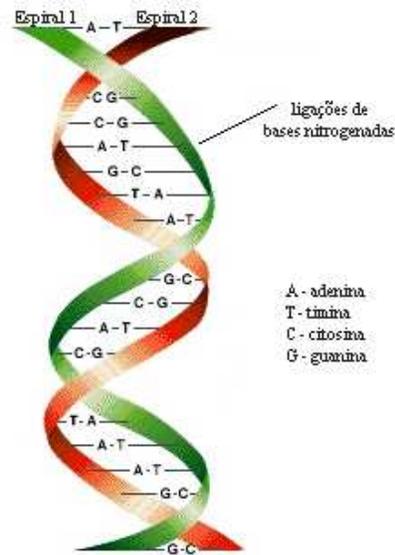
As células são capazes de se reproduzir, gerando novas células que contêm uma cópia de seu material genético ou genoma. O genoma de uma espécie engloba todo o seu conjunto de genes. No caso particular do genoma humano, ele é composto por vinte e três pares de cromossomos. No caso dos eucariontes, os cromossomos residem no núcleo celular e comportam as moléculas de DNA, conforme pode ser observado na Figura C.1. Os genes mantêm o código para produção de substâncias específicas e para controle de características hereditárias, determinando as diferenças estruturais e funcionais entre organismos. O nível de expressão dos genes pode caracterizar processos biológicos, doenças, efeitos de medicações e prover informações acerca da funcionalidade do gene.



**Figura C.1.** Cromossomo em célula eucariote.

### **C.2.2. DNA**

O DNA forma o código genético para características e processos regulatórios e construtivos em um organismo. Consiste em duas cadeias conectadas em uma estrutura de dupla hélice. As cadeias são agregadas através da ligação de pares de bases, de acordo com certas regras de pareamento. O modelo em dupla hélice foi descoberto por James Watson e Francis Crick em 1953. Os elementos componentes dos ácidos nucleicos denominam-se *nucleotídeos*. Cada nucleotídeo é formado por três componentes: um açúcar, um fosfato e uma base, produzindo uma longa cadeia acoplada com uma série de bases. O açúcar pode ser uma ribose ou uma desoxirribose. No caso do DNA, o açúcar é uma desoxirribose. As bases que compõem o DNA podem ser de quatro tipos: adenina (A), guanina (G), timina (T) e citosina (C). A adenina forma duas pontes de hidrogênio com a timina e a citosina forma três pontes de hidrogênio com a guanina. As cadeias de DNA são complementares e, devido à regra de ligação de bases (adenina com timina, e citosina com guanina), conhecer a seqüência de uma cadeia (ou fita) permite que se deduza a seqüência da segunda cadeia. Cada códon (trinca de bases) corresponde a um aminoácido. A Figura C.2 ilustra uma molécula de DNA com as duas cadeias ligadas através de bases nitrogenadas.



**Figura C.2.** Cadeias de DNA ligadas através de bases nitrogenadas.

### **C.2.3. RNA**

A tradução da codificação do DNA em aminoácidos é realizada pelo RNA, o qual apresenta estrutura similar ao DNA, com a base Uracila no lugar da base Timina. Assim como a Timina, a Uracila é complementar à base Adenina. Enquanto o açúcar componente do DNA é a desoxirribose, o RNA é composto por ribose. O RNA é formado por uma cadeia simples, e não por uma cadeia dupla, como o DNA.

Existem três tipos de RNA:

- RNA Mensageiro (mRNA): contém o código transcrito a partir do DNA para ser utilizado na síntese de proteínas. Ele é gerado a partir do DNA através de um processo denominado *transcrição*.
- RNA Transportador (tRNA): responsável pelo transporte dos aminoácidos até os ribossomos para execução da síntese protéica.
- RNA Ribossômico (rRNA): é integrante da estrutura dos ribossomos e é responsável pelo processo de tradução dos códons para a construção das proteínas.

Devido à regra de pareamento de bases, pode-se gerar uma nova molécula de DNA a partir do RNA via processo de *transcrição reversa*. Experimentalmente, o RNA mensageiro pode ser isolado de uma célula e ser convertido para a forma de DNA através de um processo de transcrição reversa. O resultado da transcrição reversa é o DNA complementar (cDNA).

#### **C.2.4. Síntese Protéica**

Uma função fundamental do código genético está ligada à síntese de proteínas. A seqüência de nucleotídeos no DNA de um cromossomo determina a seqüência de aminoácidos em uma proteína e, conseqüentemente, suas propriedades. Genes com codificação para produção de proteínas são transcritos através do mRNA. Os mRNAs constituem os elementos fundamentais da síntese protéica. Os humanos, assim como outros organismos vivos, utilizam vinte tipos de aminoácidos na síntese de proteínas. A Tabela C.1 exhibe os aminoácidos que compõem as proteínas. A Tabela C.2 ilustra as proteínas especificadas por seqüências de três bases (códon). A existência de mais de um códon associado a alguns aminoácidos é denominada degeneração do código genético.

**Tabela C.1.** Aminoácidos que compõem as proteínas.

<b>Aminoácido</b>	<b>Sigla</b>	<b>Aminoácido</b>	<b>Sigla</b>
Glicina	GLY	Alanina	ALA
Valina	VAL	Leucina	LEU
Isoleucina	ILE	Serina	SER
Treonina	THR	Ácido Aspártico	ASP
Ácido Glutâmico	GLU	Lisina	LYS
Arginina	ARG	Asparagina	ASN
Glutamina	GLN	Cisteína	CYS
Metionina	MET	Fenilalanina	PHE
Tirosina	TYR	Triptofano	TRP
Histidina	HIS	Prolina	PRO

Tabela C.2. Proteínas especificadas por seqüências de três bases.

SEGUNDA LETRA						
PRIMEIRA LETRA		U	C	A	G	
	U	UUU } <i>PHE</i> UUC } UUA } <i>LEU</i> UUG }	UCU } UCC } <i>SER</i> UCA } UCG }	UAU } <i>TYR</i> UAC } UAA } <i>STOP</i> UAG }	UGU } <i>CYS</i> UGC } UGA } <i>STOP</i> UGG } <i>TRP</i>	U C A G
	C	CUU } CUC } <i>LEU</i> CUA } CUG }	CCU } CCC } <i>PRO</i> CCA } CCG }	CAU } <i>HIS</i> CAC } CAA } <i>GLN</i> CAG }	CGU } CGC } <i>ARG</i> CGA } CGG }	U C A G
	A	AUU } AUC } <i>ILE</i> AUA } AUG } <i>MET</i>	ACU } ACC } <i>THR</i> ACA } ACG }	AAU } <i>ASN</i> AAC } AAA } <i>LYS</i> AAG }	AGU } <i>SER</i> AGC } AGA } <i>ARG</i> AGG }	U C A G
	G	GUU } GUC } <i>VAL</i> GUA } GUG }	GCU } GCC } <i>ALA</i> GCA } GCG }	GAU } <i>ASP</i> GAC } GAA } <i>GLU</i> GAG }	GGU } GGC } <i>GLY</i> GGA } GGG }	U C A G
					TERCEIRA LETRA	

### C.3. Análise de Expressão Gênica

#### C.3.1. Hibridização Comparativa

O objetivo dos experimentos de microarray baseados em hibridização comparativa é comparar os níveis de expressão gênica de duas amostras de material genético. Todas as células em um organismo possuem o mesmo DNA genômico, sendo que identidades celulares distintas decorrem de variações nos níveis de expressão gênica.

Um experimento de microarray basicamente envolve os seguintes elementos:

- Um *array*, que consiste em uma placa de vidro contendo centenas ou milhares de *spots*. Cada *spot* possui em média um tamanho de 200 micrômetros ou menos.
- Um conjunto de amostras, que são imobilizadas nos *spots* do *array*. A escolha das amostras de DNA a serem imobilizadas no *array* determina quais genes serão detectados.
- Duas populações de amostras rotuladas, consistindo de fitas simples de cDNA (DNA complementar, sintetizado através do RNA mensageiro via processo de transcrição reversa) de uma amostra conhecida e moléculas correspondentes ao cDNA a ser analisado.
- Um detector que permite medir os níveis de expressão gênica das amostras rotuladas.

Inicialmente, seqüências de DNA conhecidas são imobilizadas nos *spots* do *array*. Os genes que compõem essa seqüência determinam os genes cujo nível de expressão será analisado no experimento. O próximo passo na execução de um experimento de microarray é definir uma população de amostras composta de fitas simples de cDNA a serem hibridizadas no *array*. Em geral, um experimento envolve duas amostras: uma amostra de controle contendo cDNA conhecido, e uma amostra a ser estudada. As amostras de controle e de análise poderiam ser compostas, por exemplo, por células saudáveis e por células com doenças genéticas, ou células em diferentes etapas do ciclo celular. Na seqüência, as fitas simples de cDNA de controle e de cDNA a ser analisado são hibridizadas nos *spots* do *array*. A hibridização é feita via bases complementares. Desde que genes são formados com base em um alfabeto específico, composto por adenina (A), timina (T), citosina (C) e guanina (G), com as regras de formação de pares A-T e C-G, as amostras podem ser construídas de forma a se ligarem ao material genético das amostras rotuladas de cDNA imobilizadas nos *spots*. Para preparar um experimento de microarray, o mRNA deve ser purificado de todo o conteúdo celular restante e convertido para a forma mais estável de DNA complementar através de um processo de transcrição reversa. As duas amostras de cDNA são testadas hibridizando-as no microarray. O *array* possui centenas ou milhares de pontos, cada um contendo uma seqüência diferente de

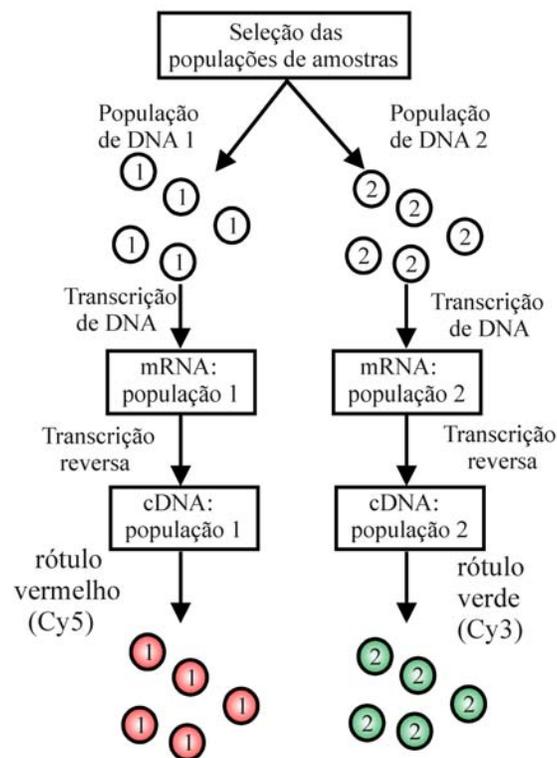
DNA. Se uma amostra de cDNA contém uma seqüência complementar ao DNA posicionado no *spot*, o cDNA irá se hibridizar ao DNA.

Em geral, moléculas de cDNA são rotuladas utilizando colorações fluorescentes (cianinas) verde e vermelha (Cy3 e Cy5) para as amostras de controle e aquela a ser analisada, respectivamente. As moléculas fluorescentes são detectáveis quando estimuladas por laser. Assim, uma vez que as amostras de cDNA se hibridizaram, o *array* deve ser escaneado de modo a determinar o nível de hibridização de cada amostra. Moléculas que não se hibridizaram ao material genético são removidas e o *array* é estimulado por laser, emitindo luz detectável. A intensidade da luz emitida por cada *spot* reflete a quantidade de amostras que se ligaram ao *spot*.

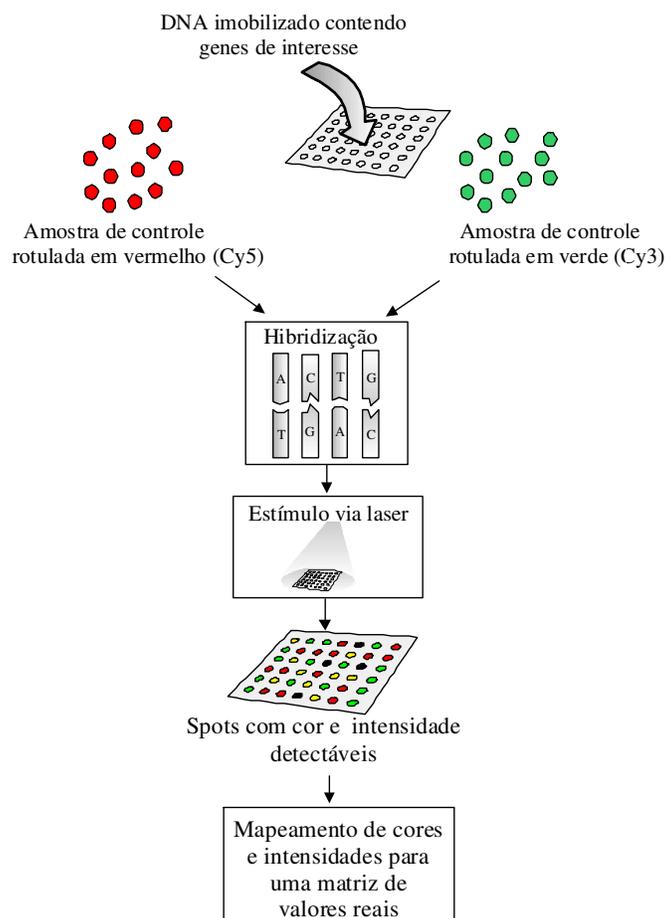
O produto de um experimento de microarray é uma imagem composta por vários *spots* coloridos, cada *spot* correspondendo a uma seqüência distinta de cDNA, a qual define os genes a serem analisados. A cor de cada *spot* é dada pela quantidade de cDNA hibridizada. *Spots* que apresentarem maior quantidade de cDNA de uma ou outra população serão predominantemente verdes ou vermelhos: pontos vermelhos correspondem a genes da população de controle sobreexpressos, enquanto pontos verdes correspondem a genes subexpressos. Pontos amarelos correspondem a genes igualmente expressos em ambas as amostras, e a cor preta indica pontos em que nenhuma das amostras se hibridizou. As cores e intensidades obtidas no *array* são mapeadas para uma matriz de valores reais  $A$ , onde  $A_{i,j}$  é a taxa  $\log_2(Cy5/Cy3)$  para o gene  $i$  e o experimento  $j$ . Portanto, a matriz de expressão gênica  $A$  contém linhas correspondendo à expressão relativa para um gene específico, e colunas correspondendo a experimentos. Genes subexpressos corresponderão a valores negativos, enquanto genes sobreexpressos serão representados por valores positivos. Genes igualmente expressos serão representados pelo valor zero.

A Figura C.3 resume os principais passos para se iniciar um experimento de microarray, mostrando desde a seleção de amostras a serem analisadas até a identificação de cada população de amostras com tinta fluorescente. Inicialmente, selecionam-se as amostras de DNA de controle e de análise. Os DNA são submetidos a um processo de transcrição, sendo

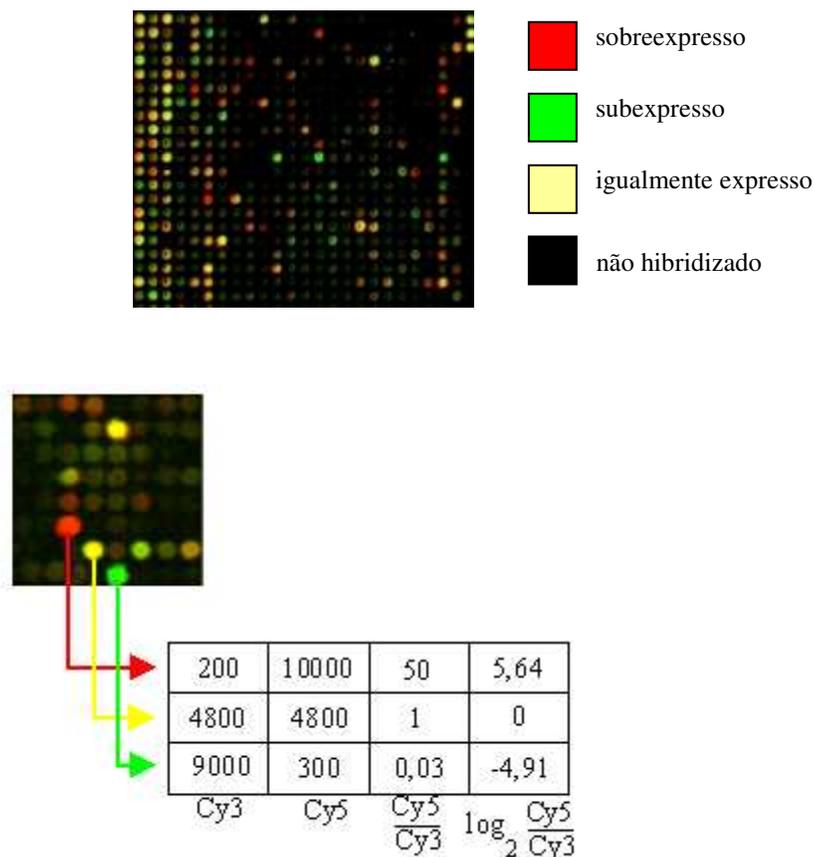
convertidos para RNA mensageiro (mRNA). O mRNA sofre então um processo de transcrição reversa, em que é convertido para DNA complementar. A amostra de cDNA de controle (população 1) é rotulada com coloração vermelha, enquanto a amostra de cDNA a ser analisada (população 2) é rotulada com a coloração verde. A Figura C.4 mostra a execução do experimento de microarray propriamente dito. Uma vez disponíveis as amostras rotuladas, elas são imobilizadas no array e ocorre o processo de hibridização, gerando uma matriz de níveis de expressão. A Figura C.5 ilustra a interpretação dos níveis de expressão com base nas cores dos spots do array e a conversão dos níveis de expressão para valores numéricos.



**Figura C.3.** Principais etapas de um experimento de microarray: seleção das populações de controle (população 1) e análise (população 2), extração do cDNA e coloração das amostras.



**Figura C.4.** Principais etapas de um experimento de microarray: hibridização das amostras rotuladas, obtenção da coloração no array e mapeamento de cores e intensidades para valores reais representativos do nível de expressão gênica.

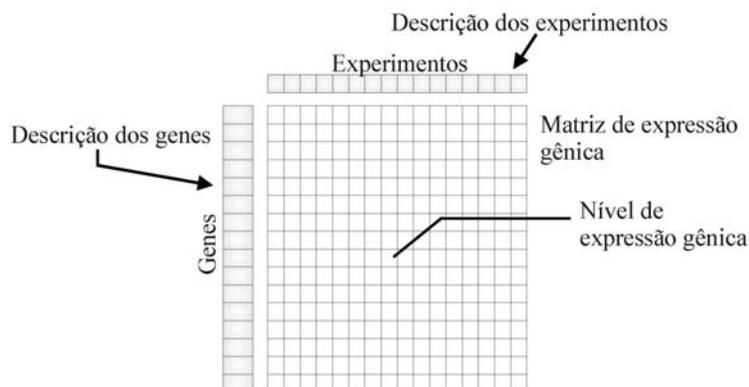


**Figura C.5.** Interpretação dos níveis de expressão gênica através das cores dos spots do array: indicação de genes sobreexpressos, subexpressos e igualmente expressos através da coloração resultante do array.

### C.3.2. Repositórios de Dados de Expressão Gênica

Dentre os vários desafios que a tecnologia de microarrays apresentam aos cientistas, compartilhamento de dados é um dos mais relevantes (SPELLMAN *et al.*, 2002). Encontram-se na literatura propostas para padronização de dados de expressão gênica, tais como a MIAME – Minimal Information for the Annotation of a Microarray Experiment (SPELLMAN *et al.*, 2002). De acordo com esta padronização, um conjunto de dados de expressão gênica pode ser visualizado como uma tabela cujas linhas correspondem a genes a serem avaliados e as colunas representam experimentos aos quais os genes são submetidos, formando uma matriz de expressão gênica. Além da matriz de expressão gênica, um experimento de microarray deve

conter informações acerca dos genes cujos níveis de expressão foram medidos e acerca das condições experimentais em que cada amostra foi tomada. Dessa forma, a informação necessária para descrever um experimento de microarray envolve três partes: informações sobre genes, informações sobre experimentos e matriz de expressão gênica, conforme ilustrado na Figura C.6.



**Figura C.6.** Matriz de dados de expressão gênica.

Existem diversas bases públicas de dados de expressão gênica, dentre elas:

- Stanford Microarray Database: <<http://genome-www.stanford.edu/microarray>>;
- Array Express: <<http://www.ebi.ac.uk/arrayexpress>>;
- Gene Expression Omnibus: <<http://www.ncbi.nlm.nih.gov/geo>>;
- Human Gene Expression Index: <<http://www.hugeindex.org>>;
- Yale Microarray Database: <<http://info.med.yale.edu/microarray>>;
- Eisen Lab: <<http://rana.lbl.gov/EisenData.htm>>.

Embora as bases de dados se apresentem normalmente no formato apresentado na Figura C.6, algumas variações podem ser encontradas. Os níveis de expressão podem estar normalizados ou não. Alguns arquivos, além de conter a descrição dos genes e dos experimentos, podem

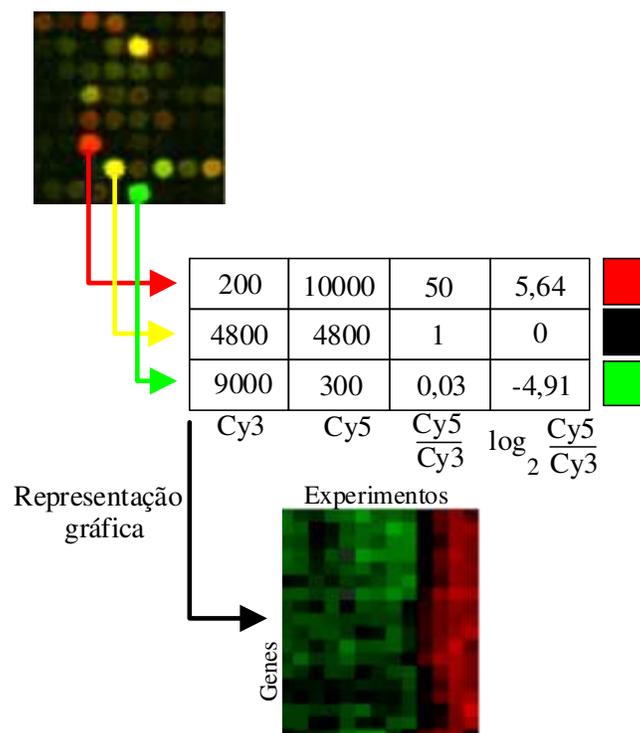
conter um cabeçalho adicional. Outros ainda podem fornecer somente a matriz de expressão gênica, não incluindo cabeçalhos ou rótulos para genes ou experimentos.

### ***C.3.3. Softwares para Visualização e Análise de Dados de Expressão Gênica***

Um software comumente empregado na análise de dados de expressão gênica é o CLUSTER, desenvolvido pela equipe do Eisen Lab, na Universidade de Stanford (BAR-JOSEPH *et al.*, 2001; BIEDL *et al.*, 2001). O CLUSTER apresenta diversas opções de organização dos dados, tais como ordenação linear e diversas modalidades de agrupamento, tais como hierárquico e k-médias. O software TREEVIEW pode ser empregado para visualização dos dados de expressão gênica através de uma representação gráfica similar àquela ilustrada na Figura C.7. Nesta representação, os genes sobreexpressos (spots predominantemente vermelhos) são representados em vermelho, os genes subexpressos (spots predominantemente verdes) são representados em verde e genes igualmente expressos (spots amarelos) são representados em preto. Amostras que não hibridizaram não são consideradas na representação gráfica.

Um aspecto importante a ser considerado antes de se utilizar um software específico é a formatação dos dados de entrada. Conforme mencionado na Seção C.3.2, as bases de dados podem apresentar variações, como cabeçalhos e normalização de dados.

Tanto o CLUSTER como o TREEVIEW requerem dados de entrada tal como especificados em <http://rana.lbl.gov/EisenData.htm>. Ambos os softwares podem ser obtidos em <http://rana.lbl.gov/EisenSoftware.htm>.



**Figura C.7.** Representação gráfica para resultados de experimentos de microarrays.

# APÊNDICE D

## SISTEMAS NEBULOSOS

---

Neste apêndice será apresentada uma introdução aos sistemas nebulosos, abordando alguns aspectos fundamentais relacionados a conjuntos, variáveis e inferência nebulosa. Os conceitos aqui apresentados são necessários para a compreensão do método desenvolvido no Capítulo 4 desta tese, o qual faz uso de inferência nebulosa com o objetivo de guiar o processo de treinamento de mapas auto-organizáveis aplicados ao problema de roteamento de veículos.

### D.1. Introdução

O conceito de lógica nebulosa foi introduzido por Lotfi Zadeh em 1965 através da publicação do trabalho intitulado “*Fuzzy Sets*”, no jornal acadêmico “*Information and Control*” (ZADEH, 1965).

Os sistemas nebulosos são empregados com o propósito de prover o processamento de dados de modo a permitir pertinências parciais dos dados a certos conjuntos, sendo mais efetivo e proporcionando maior facilidade de implementação no tratamento de dados imprecisos ou ruidosos. Assim, o uso de sistemas nebulosos é indicado quando não se conhece ou não existe um modelo matemático para o problema, quando o modelo matemático é muito complexo, quando sua execução em tempo real é inviável ou quando existe um alto nível de ruído ou imprecisão nos dados de entrada do sistema (REGATTIERI, 2002).

O campo de aplicação é vasto e diversificado: medicina, controle e automação industrial, reconhecimento de padrões, processamento de imagens, reconhecimento de voz, otimização matemática, finanças, bioinformática (NADJAH & MOURELLE, 2006) e emprego em conjunto com outras abordagens da área da inteligência computacional, tais como redes neurais artificiais, algoritmos genéticos e técnicas de clusterização (JAIN & JAIN, 1997).

## D.2. Conjuntos Nebulosos x Conjuntos Convencionais

A teoria dos conjuntos nebulosos pode ser vista como uma extensão da teoria dos conjuntos clássica e está associada a conceitos de funções de pertinência, operações com números nebulosos, operações de conjunto nebuloso e regras de inferência.

Seja  $X$  um conjunto de objetos cujos elementos  $x$ . Um conjunto nebuloso  $A$  sobre  $X$  pode ser formalmente definido através da coleção de pares ordenados

$$A = \{(x; \mu_A(x)), x \in X\}, \quad (\text{D.1})$$

onde  $\mu_A(x)$  é a função de pertinência que fornece o grau com que o elemento  $x$  pertence ao conjunto  $A$ . No caso de conjuntos clássicos, a função de pertinência pode assumir apenas os valores 0 ou 1, onde 0 indica que o elemento  $x$  não pertence ao conjunto  $A$ , enquanto o valor 1 indica que o elemento  $x$  pertence ao conjunto  $A$ .

A distinção entre a teoria de conjuntos nebulosos e a teoria clássica pode ser ilustrada com um exemplo simples: considere o problema de classificação de um conjunto de pessoas de acordo com sua estatura, definindo o conjunto  $X$  de “pessoas altas”. Tomando-se a coleção de valores  $X = \{1,40, 1,50, 1,60, 1,70, 1,75, 1,80, 1,90\}$  e de acordo com a teoria clássica de conjuntos, cada um dos valores deve ser classificado como pertencente ou não ao conjunto  $X$ , conforme definido a seguir:

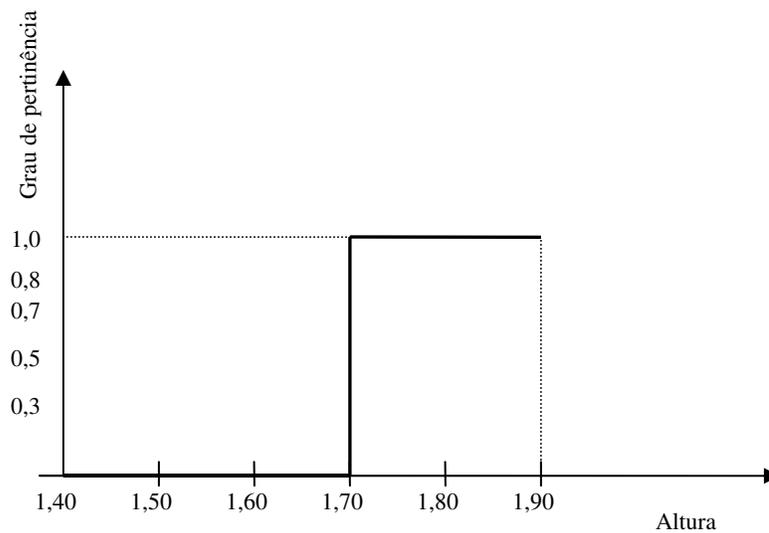
$$A = \{(1,40;0), (1,50;0), (1,60;0), (1,70;1), (1,80;1), (1,90;1)\}$$

A função de pertinência correspondente ao conjunto  $A$  na teoria clássica é exibida na Figura D.1.

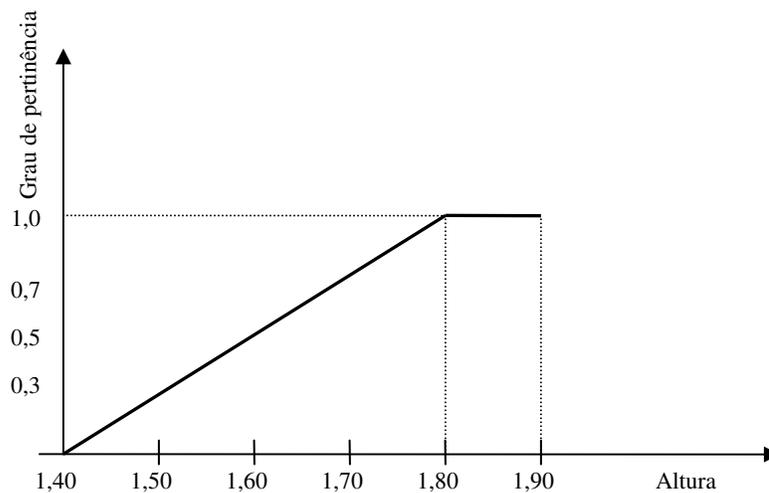
De acordo com a teoria de conjuntos nebulosos, são admissíveis valores de pertinência intermediários a 0 e 1. Uma possível relação de pertinência é apresentada a seguir:

$$A = \{(1,40;0), (1,50;0,3), (1,60;0,5), (1,70;0,7), (1,80;1,0), (1,90;1)\}.$$

A função de pertinência correspondente ao conjunto  $A$  é exibida na Figura D.2.



**Figura D.1.** Função de pertinência para o conjunto  $A$  de acordo com a teoria de conjuntos clássica: os níveis de pertinência podem assumir somente o valor 0 ou o valor 1.

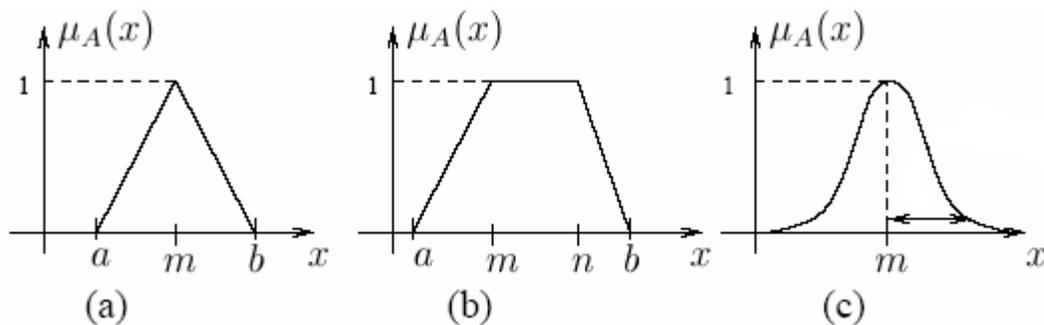


**Figura D.2.** Função de pertinência para o conjunto  $A$  de acordo com a teoria de conjuntos nebulosos: os níveis de pertinência podem assumir valores entre 0 e 1.

Pode-se afirmar que os elementos do conjunto  $A$ , no caso restrito ao intervalo  $[1,4;1,9]$ , formam o universo de discurso da variável lingüística “pessoas altas”.

### D.2.1. Funções de Pertinência

Os conjuntos nebulosos são caracterizados por uma função de pertinência que associa cada um dos elementos no universo de discurso da variável lingüística ao seu grau de pertinência ao conjunto. Os formatos mais comuns para as funções de pertinência são triangular, trapezoidal e gaussiana. A Figura D.3 ilustra os formatos comumente utilizados. Os parâmetros associados às funções são geralmente definidos por um especialista.



**Figura D.3.** Funções de pertinência comumente empregadas: (a) triangular; (b) trapezoidal; (c) gaussiana.

## D.3. Sistemas baseados em Inferência Nebulosa

Conforme apresentado na Seção D.2, a lógica nebulosa pode ser vista com uma extensão das lógicas bivalores e multivalores convencionais. Permite a modelagem de sentenças em uma linguagem, que depende do conceito de *variáveis lingüísticas*, da definição das funções de pertinência associadas às variáveis e de um conjunto de regras nebulosas. Nesta seção será inicialmente introduzido o conceito de variáveis lingüísticas. Em seguida é descrita a metodologia de inferência nebulosa e são brevemente apresentados alguns métodos de *defuzzificação*.

### D.3.1. O Conceito de Variáveis Lingüísticas

Em 1973, Lotfi Zadeh propôs o conceito de variáveis lingüísticas. Em uma dada linguagem  $L$  têm-se as variáveis, que podem ser definidas como objetos associados a um nome, como “temperatura”, “velocidade” ou “pressão”.

Expressões lingüísticas são típicas da natureza humana no momento da tomada de decisões. Por exemplo, “Se a *Temperatura* está *muito baixa* então ligar o aquecedor no *alto*”. As classificações “baixa” e “alto” não se traduzem a um valor exato em particular, mas a uma faixa consideravelmente ampla de valores.

Formalmente, as variáveis lingüísticas são caracterizadas pelos seguintes elementos:

- Nome: rótulo para a variável, por exemplo “Temperatura” ou “Velocidade”.
- Termos lingüísticos: compõem os rótulos que a variável pode assumir. Por exemplo, “alto”, “baixo”, “muito baixo”. A quantidade de rótulos da variável é denominada *granularidade* da variável lingüística.
- Universo de discurso: representa todos os valores que a variável lingüística pode assumir.
- Conjunto de regras: responsável pela associação de cada um dos termos lingüísticos associados à variável nebulosa, considerando seu universo de discurso.

### D.3.2. Computação com Regras Nebulosas

As regras nebulosas apresentam o formato genérico “Se  $X$  é  $A$  então  $Y$  é  $B$ ”, onde  $A$  e  $B$  representam conjuntos nebulosos e  $X$  e  $Y$  constituem variáveis nebulosas. As regras podem ser compostas por *conjunções*, *disjunções* e *implicações* nebulosas. No caso de regras que envolvem múltiplas variáveis, por exemplo, o antecedente e o conseqüente podem envolver conjunções e/ou disjunções:

Se  $X_1$  é  $A_1$  e  $X_2$  é  $A_2$  então  $Y_1$  é  $B_1$  e  $Y_2$  é  $B_2$

A conjunção “e” é comumente denotada pelo símbolo “ $\wedge$ ”, enquanto a disjunção “ou” é denotada por “ $\vee$ ”. O conectivo “ $\wedge$ ” é comumente representado pela *norma-t min*, que constitui a norma-t adotada nos algoritmos propostos nesta tese. Há outros modelos de normas-t empregados no processo de inferência nebulosa (PEDRYCZ & GOMIDE, 1998). Assim, tem-se que:

$$\mu_A(x) \wedge \mu_B(y) = \min(\mu_A(x), \mu_B(y)). \quad (\text{D.2})$$

A disjunção “ $\vee$ ” é comumente representada pela *norma-s max*, havendo também para este caso outros modelos (PEDRYCZ & GOMIDE, 1998). Assim, tem-se que:

$$\mu_A(x) \vee \mu_B(y) = \max(\mu_A(x), \mu_B(y)). \quad (\text{D.3})$$

O raciocínio nebuloso permite a computação com dados imprecisos. Um exemplo prático é o controle automático da temperatura da água de um chuveiro: embora seja improvável que uma pessoa possa definir precisamente a temperatura da água, ela é capaz de identificar com facilidade se a água está muito fria, muito quente ou a uma temperatura confortável.

Uma possível regra para operação do sistema é:

- **Se** *Temperatura* < 18° **então** aquecer água a uma taxa de 2°C por segundo.

Uma versão nebulosa para a regra exposta acima pode ser:

- **Se** *Temperatura é muito fria* **então** aquecer água *rapidamente*.

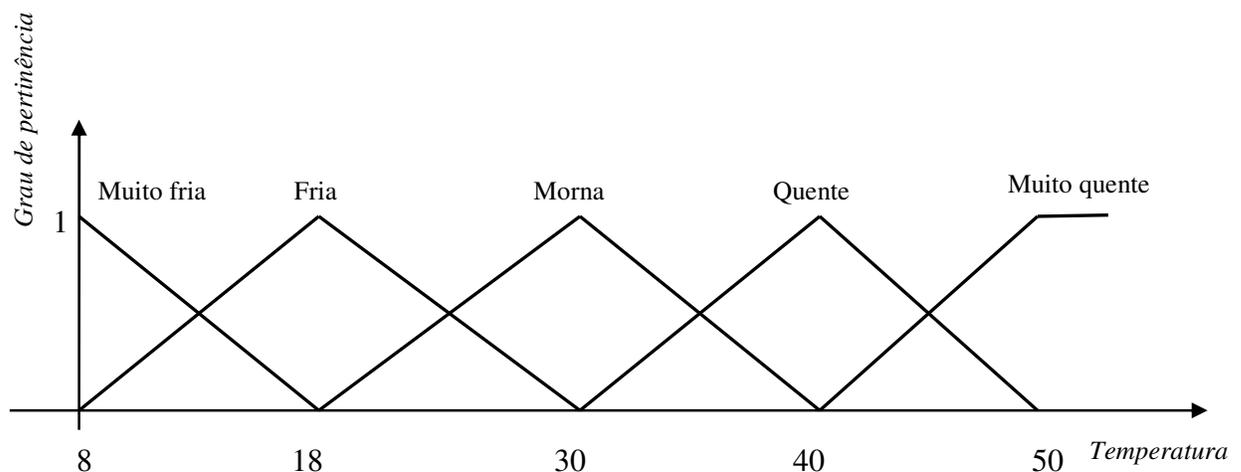
Para a elaboração de um modelo nebuloso, são necessários certos parâmetros numéricos, porém os valores exatos não constituem um requisito crítico. A faixa de valores aceitáveis pode depender da imprecisão dos dados ou da taxa de ruído ou mesmo do grau de liberdade que se tem no sistema. Por exemplo, retomando o exemplo da temperatura da água de um chuveiro, tanto 30,3°C como 35,2°C ou 36°C podem ser consideradas temperaturas confortáveis para o ser humano.

A construção de um modelo nebuloso envolve a elaboração de funções de pertinência para cada termo lingüístico das variáveis, a definição de um conjunto de regras e um processo de *defuzzificação*, que efetua a conversão da saída do processo de inferência para um valor numérico, de modo que possa ser interpretado como saída do sistema.

### D.3.3. Funções de Pertinência

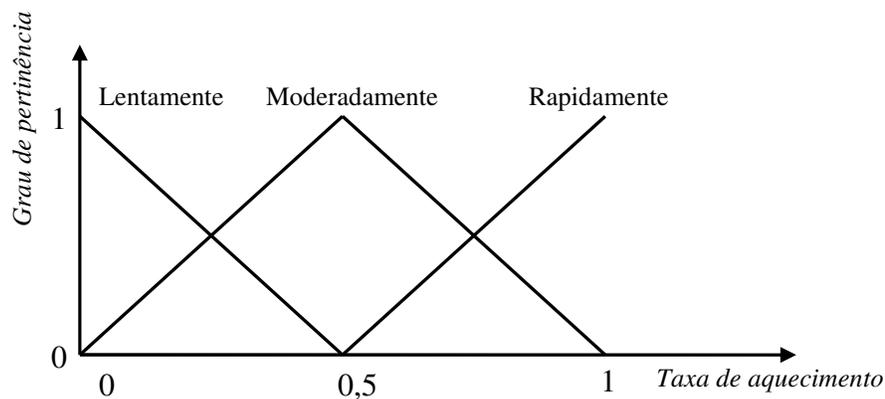
Os conjuntos nebulosos requerem a definição de funções de pertinência correspondentes aos termos lingüísticos definidos para as variáveis, considerando seu universo de discurso. Tomando como exemplo o sistema de aquecimento da água apresentado na Seção D.3.2, assumo que a temperatura da água possa apresentar valores reais entre 10°C e 70°C.

Particionando-se o universo de discurso em conjuntos nebulosos caracterizados por sua função de pertinência, um possível modelo para a variável *Temperatura* é ilustrado na Figura D.4. Note que há valores que fazem parte de mais de um conjunto, como, por exemplo, *Temperatura* igual a 33°C pertence ao conjunto “Morna” e “Quente” simultaneamente, o que pode levar à ativação de múltiplas regras durante o processo de inferência nebulosa.



**Figura D.4.** Partição do universo de discurso para a variável nebulosa *Temperatura*.

A variável de saída fornece a taxa de aquecimento. A taxa de aquecimento pode se enquadrar em três conjuntos nebulosos, conforme mostrado na Figura D.5: *Lentamente*, *Moderadamente* e *Rapidamente*.



**Figura D.5.** Partição do universo de discurso para a variável nebulosa de saída correspondente à taxa de aquecimento água.

Uma vez definidas as funções de pertinência para os termos lingüísticos das variáveis nebulosas de entrada e saída, estas devem ser relacionadas através de um conjunto de regras nebulosas do tipo:

- **Se** *Temperatura é Morna* **então** aquecer água *Lentamente*.
- **Se** *Temperatura é Muito Fria* **então** aquecer água *Rapidamente*.

Considerando-se o valor da variável de entrada, o nível de ativação da regra será dado pelo grau de pertinência do valor de entrada ao conjunto nebuloso associado ao antecedente da regra, no caso em que a regra envolve somente uma variável de entrada. Caso a regra envolva múltiplas variáveis de entrada, o nível de ativação será dado pelo menor nível dentre todos os graus de pertinência das variáveis de entrada. Este critério é empregado no Capítulo 4 desta tese, em que um conjunto de regras de inferência nebulosa é utilizado com o objetivo de guiar o processo de treinamento de mapas auto-organizáveis.

### D.3.4. Métodos de Defuzzificação

Na *defuzzificação* ocorre o processo inverso da *fuzzificação*. As regras são processadas e o resultado é convertido em um único valor numérico, o qual é retornado ao usuário ou ao sistema de controle. Diferentemente dos sistemas convencionais, diversas regras podem ser ativadas simultaneamente, sendo que cada regra é ativada com um grau compatível com as condições de entrada.

Existem diversos métodos de *defuzzificação* (YAGER, 1994): centro de área, média dos máximos, distância de Hamming, barras verticais, método da altura, entre outros. Neste apêndice será detalhado somente o método do centro de área, pois este foi o método utilizado ao longo do Capítulo 4.

#### D.3.4.1. Centro de Área

O método do centro de área é baseado no cálculo do centro de área da figura resultante das regras ativadas. Com base nos valores  $X$  fornecidos como entrada ao sistema nebuloso, verifica-se que  $X$  está contido no conjunto definido pelos termos lingüísticos  $A_i$ , onde  $i \geq 1$ . Assim, o centro de área é dado pela Equação D.4.

$$CA = \frac{\sum_{i=1}^{nr} w_i \cdot A_i}{\sum_{i=1}^{nr} A_i}, \text{ onde} \quad (D.4)$$

$nr$ : número de regras ativadas;

$w_i$ : Nível de ativação da regra  $i$ ;

$A_i$ : área do termo de cada regra  $i$ .

# APÊNDICE E

## PUBLICAÇÕES

---

---

1. Gomes, L.C.T., Von Zuben, F.J. Multiple criteria optimization based on unsupervised learning and fuzzy inference applied to the vehicle routing problem. *Journal of Intelligent & Fuzzy Systems (IOS Press)*, vol. 13, no. 2-4, pp. 143-154, 2003. (ISSN: 1064-1246)
2. de Sousa, J.S., Gomes, L.C.T., Bezerra, G.B., de Castro, L.N., Von Zuben, F.J. An Immune-Evolutionary Algorithm for Multiple Rearrangements of Gene Expression Data. *Special Issue on Biological Applications of Genetic and Evolutionary Computation, Genetic Programming and Evolvable Machines*, vol. 5, no. 2, pp. 157-179, 2004. (ISSN: 1389-2576)
3. Gomes, L.C.T., Von Zuben, F.J., Moscato, P. A Proposal for Direct-Ordering Gene Expression Data by Self-Organising Maps. *International Journal of Applied Soft Computing*, vol. 5, pp. 11-21, 2004. (ISSN: 1568-4946)
4. Gomes, L.C.T., de Sousa, J.S., Bezerra, G.B., de Castro, L.N., Von Zuben, F.J. Copt-aiNet and the Gene Ordering Problem. *Revista Tecnologia da Informação, Universidade Católica de Brasília*, vol. 3, no. 2, 27-33, Dezembro 2003. (ISSN: 1516-9197)
5. Gomes, L.C.T., Von Zuben, F.J. Vehicle Routing Based on Self-Organization With and Without Fuzzy Inference. *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'2002)*, vol. 2, pp. 1310-1315, in the 2002 IEEE World Congress on Computational Intelligence (WCCI'2002), Honolulu, Hawaii, May 12-17, 2002.
6. Gomes, L.C.T., Von Zuben, F.J. A Neuro-Fuzzy Approach to the Capacitated Vehicle Routing Problem. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'2002)*, vol. 2, pp. 1930-1935, in the 2002 IEEE World Congress on Computational Intelligence (WCCI'2002), Honolulu, Hawaii, May 12-17, 2002.

- 
7. Gomes, L.C.T., Von Zuben, F.J., Moscato, P.A. Ordering Microarray Gene Expression Data Using a Self-Organising Neural Network. Proceedings of the 4th International Conference on Recent Advances in Soft Computing (RASC 2002), Nottingham, United Kingdom, pp. 307-312, December 2002.
  8. de França, F. O., Gomes, L. C. T., de Castro, L. N., Von Zuben, F. J., “Handling Time-Varying TSP Instances”, *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE-CEC 2006)*.
  9. Gomes, L.C.T., Von Zuben, F.J. Uma Abordagem Baseada em Mapas Auto-Organizáveis de Kohonen Aplicada ao Problema de Roteamento de Veículos. V Simpósio Brasileiro de Automação Inteligente, artigo no. 1059, 6 páginas, 7 a 9 de novembro de 2001.
  10. Gomes, L.C.T., Von Zuben, F.J. Uma Heurística Baseada em Aprendizado Competitivo e Inferência Fuzzy Aplicada ao Problema de Roteamento de Veículos Capacitados. Anais do XIV Congresso Brasileiro de Automática (CBA’2002), Natal, RN, 02 a 05 de setembro, pp. 2504-2510, 2002.
  11. Gomes, L.C.T., Von Zuben, F.J., Moscato, P. Ordering Gene Expression Data Using One-Dimensional Self-Organizing Maps. Anais do I Workshop Brasileiro de Bioinformática (WOB’2002), Gramado, RS, pp. 91-93, 18 de outubro, 2002. (ISBN: 85-88442-35-3)
  12. Gomes, L.C.T., Von Zuben, F.J. A Heuristic Method Based on Unsupervised Learning And Fuzzy Inference for the Vehicle Routing Problem. Proceedings of the VII Brazilian Symposium on Neural Networks, Porto de Galinhas, Recife, November 11-14, pp. 130-135, 2002. (IS BN: 0-7695-1709-9)

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- [1] Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, L. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, I., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., (2000), “Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling”, *Nature* 403, pp. 503-511.
- [2] Angéniol, B., Vauboiss, C. & Le Texier, J., Y., (1988), “Self-organizing feature maps and the travelling salesman problem”, *Neural Networks*, vol 1, pp. 289-293.
- [3] Aras, N., Oommen, B.J. & Altinel, I.,K., (1999), “The Kohonen Network Incorporating Explicit Statistics and its Application to The Travelling Salesman Problem”, *Neural Networks*, vol 12, pp. 1273-1284.
- [4] Bäck, T., Fogel, D. B., Michalewicz, Z. (eds), (1997), “Handbook of Evolutionary Computation”, Institute of Physics Publishing and Oxford University Press.
- [5] Balthrop, J., Forrest, S., Newman, M. E. J. and Williamson, M. M., (2004), “Technological Networks and the Spread of Computer Viruses”, *Science*, vol 304, Issue 5670, pp. 527-529.
- [6] Bar-Joseph, Z., Gifford, D. and Jaakkola, T., (2001), “Fast Optimal Leaf Ordering for Hierarchical Clustering”, *Bioinformatics (Proceedings of ISMB 2001, Denmark)*, vol 17, pp. 22-39.
- [7] Barlow, H. B. (1989), “Unsupervised Learning”, *Neural Computation*, vol 1, pp. 295-311, Kluwer Academic Pub.
- [8] Barreto, G. A. and Araújo, A. F. R., (2004), “Identification and Control of Dynamic Systems using the Self-Organizing Map”, *IEEE Transactions on Neural Networks*, vol 15, n.5, pp. 1244-1259.

- [9] Bauer, H.-U. and Pawelzik, R. K., (1992), “Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps”, *IEEE Transactions on Neural Networks*, vol 3, n.4, pp. 570-579.
- [10] Bekey, G. A. and Goldberg, K. Y. (eds), (1993), “Neural Networks in Robotics”, Kluwer Academic Pub.
- [11] Bersini, H., (1991), “Immune Network and Adaptive Control”, *Proc. of the First European Conference on Artificial Life*, MIT Press, pp. 217-226.
- [12] Biedl, T., Brejová, B., Demaine, E. D., Hamel, M. A., Vinar, T., (2001), “Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data”, *Technical Report 2001-14*, University of Waterloo, Canada, 2001.
- [13] Blackmore, J. and Miikkulainen, R., (1993), “Incremental Grid Growing: Encoding High-Dimensional Structure into a Two-Dimensional Feature Map”, In: *Proceedings of the IEEE International Conference on Neural Networks (ICNN’93)*, San Francisco CA, USA, vol 1, pp. 450-455.
- [14] Blum, C. and Roli, A., (2003), “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”, *ACM Press*, vol 35, n.3, pp. 268-308.
- [15] Boheme, H. J., Braumann, U. D. and Gross, H. M. (1994), “A Neural Network Architecture for Sensory Controlled Internal Simulation”, In Maria Marinaro and Pietro G. Morasso (Eds), *Proc. ICANN’94, Int. Conf. on Artificial Neural Networks*, vol 2, pp. 1189-1192.
- [16] Bruske, J. and Sommer, G., (2001), “Dynamic Cell Structure Learns Perfectly Topology Preserving Map”, In K. Obermayer and T.J. Sejnowski, editors, *Self-Organizing Map Formation*, MIT Press, Cambridge, Massachusetts, pp. 385-404.
- [17] Bunke, H. and Shearer, K., (1998), “A Graph Distance Metric based on the Maximum Common Subgraph”, *Pattern Recognition Letters*, vol 19, pp. 255-259.

- [18] Burel, G. and Pottier, I. (1991), “Vector Quantization of Images Using Kohonen Algorithm. Theory and Implementation”, *Revue Technique Thomson-CSF*, vol 23, n.1, pp. 137-159.
- [19] Burke L. I. and Damany, P., (1992), “The Guilty Net for the Travelling Salesman Problem”, *Computers and Operations Research* 19, pp. 255-265.
- [20] Burnet, F. M., (1959), “The Clonal Selection Theory of Acquired Immunity”, Cambridge University Press.
- [21] Caprara, A. and Fischetti, M., (1997), “Branch-and-Cut Algorithms”, in “Annotated Bibliographies in Combinatorial Optimization”, Dell’Amico, M., Maffioli, F., Martello, S. (editors), Wiley, pp. 45-64.
- [22] Carpenter, G. A. and Grossberg, S., (Editors), (1991), “Pattern Recognition by Self-Organizing Neural Networks”, MIT Press.
- [23] Chiang, W. C. and Russel, R., (2004), “A Metaheuristic for the Vehicle Routing Problem with Soft Time Windows”, *Journal of the Operational Research Society*, vol 55, pp. 1298-1310.
- [24] Christofides, N. and Eilon, S., (1969), “An algorithm for the Vehicle Dispatching Problem”, *Operational Research Quartely* 20, pp. 309-318.
- [25] Coello, C. A. C. and Cortés, N. C., (2005), “Solving Multiobjective Optimization Problems using an Artificial Immune System”, *Genetic Programming and Evolvable Machines*, vol 6, n. 2, pp. 163-190.
- [26] Costa, J. A. F., (1999), “Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis”, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (UNICAMP).
- [27] Cotta, C., Mendes, A., Garcia, V., França and P., Moscato, P., (2003), “Applying Memetic Algorithms to the Analysis of Microarray Data”, In *Proceedings of EvoBIO2003 – 1st European Workshop on Evolutionary Bioinformatics*, Lecture

- Notes in Computer Science, n. 2611, Springer-Verlag, , Colchester, England, pp. 22-32.
- [28] Moscato, P. and Cotta, C., (2003), “A gentle introduction to memetic algorithms”, In F. Glover and G. Kochenberger (eds), *Handbook of Metaheuristics*, Kluwer Academic Publishers, Boston MA, pp. 105–144.
- [29] Dantzig, G. B. and Ramser, R. H., (1959), “The Truck Dispatching Problem”, *Management Science* 6, pp. 80-91.
- [30] Dasgupta, D. (Ed.) (1999), “Artificial Immune Systems and Their Applications”, Springer-Verlag.
- [31] Deboeck, G. and Kohonen, T. (eds.), (1998), “Visual Explorations in Finance with Self-Organizing Maps”, Springer.
- [32] Decker, R., (2006), “A Growing Self-Organizing Neural Network for Lifestyle Segmentation”, *Journal of Data Science* 4, pp. 147-168.
- [33] de Castro, L. N. and Von Zuben, F. J. (1999), “Artificial Immune Systems: Part I – Basic Theory and Applications”. Technical Report- RT DCA 01/99.
- [34] de Castro, L. N. and Von Zuben, F. J. (1999), “Artificial Immune Systems: Part I – Basic Theory and Applications”, Technical Report – RT DCA 01/99.
- [35] de Castro, L. N. and Von Zuben, F. J. (1999), “Artificial Immune Systems: Part II – A Survey of Applications”, Technical Report – RT DCA 02/00.
- [36] de Castro, L. N. (2001), “Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais”, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (UNICAMP)..
- [37] de Castro, L. N. and Von Zuben, F. J. (2002), “Learning and Optimization using the Clonal Selection Principle”, *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems, vol 6, n.3, pp. 239-251.

- [38] de Castro, L. N. and Timmis, J. (2002), “Artificial Immune Systems: A New Computational Intelligence Approach”, Springer-Verlag.
- [39] de Castro, L. N. and Timmis, J., (2002a), “An Artificial Immune Network for Multimodal Optimization”, In 2002 Congress on Evolutionary Computation. Part of the 2002 IEEE World Congress on Computational Intelligence, pp. 699-704, Honolulu, Hawaii, USA, May 2002. IEEE.
- [40] de Castro, L. N. and Von Zuben, F. J., (2004), “Recent Developments in Biologically Inspired Computing”, Idea Group Publishing.
- [41] de França, F. O., Von Zuben, F. J. and de Castro, L. N., (2005), “An Artificial Immune Network for Multimodal Function Optimization on Dynamic Environments”, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’2005), ACM Press, June 25-29, Washington, D.C., pp. 289-296.
- [42] de França, F. O., Gomes, L. C. T., de Castro, L. N. and Von Zuben, F. J., (2006), “Handling Time-Varying TSP Instances”, Proceedings of the IEEE Congress on Evolutionary Computation (IEEE-CEC 2006), pp. 2830-2837.
- [43] Fritzke, B., (1999), “Growing self-organizing networks - history, status quo, and perspectives”, In Kohonen Maps, Proceedings of WSOM-99, E. Oja et. al. (eds), Elsevier 1999, pp. 131-144.
- [44] Dorigo, M. and Di Caro, G. (1999a), “Ant Algorithms for Discrete Optimization, Artificial Life”, vol 5, n.2, pp.137-172.
- [45] Dorigo, M, Di Caro, G., (1999b), “The Ant Colony Optimization Meta-Heuristic”, in New Ideas in Optimization, McGraw Hill, pp. 11-32.
- [46] Duda, R. O., Hart, P. E., Stork, D. G., (2001), “Pattern Classification”, Wiley-Interscience, 2<sup>nd</sup> Edition.

- [47] Eberhart, R. C. and Kennedy, J., (1995), “A new optimizer using particle swarm theory”, Proceedings of the sixth international symposium on micro machine and human science, IEEE service center, Piscataway, NJ, Nagoya, Japan, pp. 39-43..
- [48] Eberhart, R. C. and Shi, Y., (1988), “Evolving artificial neural networks. Proc”, 1988 Int' l Conf. on neural networks and brain, pp. PL5-PL13, Beijing, P. R. China.
- [49] Eberhart, R. C. and Shi, Y., (2001), “Particle swarm optimization: developments, applications and resources”, Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea, vol 1, pp. 81-86..
- [50] Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998), “Cluster Analysis and Display of Genome-Wide Expression Patterns”, Proc Natl Acad Sci USA, vol 95, n.25, pp. 14863-14868.
- [51] Endoh, S., Toma, N. and Yamada, K., (1998), “Immune Algorithm for n-TSP”, Proc. of the IEEE SMC'98, pp. 3844-3849.
- [52] Everitt, B. S., Landau, S. and Leese, M., (2001), “Cluster Analysis”, Edward Arnold, 4<sup>th</sup> Edition.
- [53] Favata, F. and Walker, R., (1991), “A Study of the Application of Kohonen-Type Neural Networks to the Traveling Salesman Problem”, Biological Cybernetics 64, pp. 463-468.
- [54] Ferrán, E. A. and Ferrara, P. (1992), “Clustering Proteines into Families Using Artificial Neural Networks”, Computer Applications in the Biosciences, vol 8, n.1, pp. 39-44.
- [55] Flexer, A., (2001), “On the Use of Self-Organizing Maps for Clustering and Visualization”, Intelligent Data Analysis, vol 5, pp. 373-384.
- [56] Fogel, D. B., (1995), “Evolutionary Computation: Toward a New Philosophy of Machine Intelligence”, IEEE Press.

- [57] Fort, J. C., (1988), "Solving a Combinatorial Problem via Self-Organizing Maps", *Biological Cybernetics*, 59, pp. 33-40.
- [58] Fukuda, T. and Kubota, N., (1999), "An Intelligent Robotic System Based on a Fuzzy Approach", *Proceedings of the IEEE*, vol 87, n. 9, pp. 1448-1470.
- [59] Fukushima, K. (1980), "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", *Biological Cybernetics*, vol 36, n.4, pp. 193-202.
- [60] Garcia, V., Mendes, A. S., Moscato, P. A. e França, P. M., (2001), "Algoritmo Memético Paralelo Aplicado a Problemas de Sequenciamento em Máquinas Simples", *CIENTÍFICO NACIONAL, XXXIII Simpósio Brasileiro de Pesquisa Operacional*, vol 1, pp. 971-981, Campos do Jordão, SP, BRASIL.
- [61] Garey, M. R. and Johnson, D., (1979), "Computers and Intractability: A Guide to the Theory of NP-completeness", Freeman.
- [62] Gash, A P. and Eisen, M. B., (2002), "Exploring the Conditional Coregulation of Yeast Gene Expression through Fuzzy k-means Clustering", *Genome Biology*, vol 3, n.11, pp. 1-22.
- [63] Ghaziri, H., (1991), "Solving Routing Problems by a Self-Organizing Map", in Kohonen, T., Makisara, K., Simula, O., Kangas, J. (editors), "Artificial Neural Networks", pp. 829-834.
- [64] Glover, F. and Laguna, M., (1997), "Tabu Search", Kluwer Academic Publishers.
- [65] Glover, F. W. and Laguna, M. (1998), "Tabu Search", Kluwer Academic Publishers.
- [66] Glover, F. W. and Kochenberger, G. A., (2002), *Handbook of Metaheuristics*", Kluwer Academic Publishers.
- [67] Golden, B. L. and Assad, A. A., (eds.), (1988), "Vehicle Routing: Methods and Studies, *Studies in Management Science and Systems*", vol 16, Elsevier Science Publ., Amsterdam, 1988, pp. 7-45.

- [68] Gomes, L. C. T. e Von Zuben, F. J., (2001), “Uma Abordagem Baseada em Mapas Auto-Organizáveis de Kohonen Aplicada ao Problema de Roteamento de Veículos”, Anais do V Simpósio Brasileiro de Automação Inteligente (V SBAI), artigo no. 1059, 6 páginas.
- [69] Gomes, L. C. T., Von Zuben, F. J. and Moscato, P., (2002), “Ordering Microarray Gene Expression Data Using a Self-Organising Neural Network”, Proceedings of the Recent Advances in Soft Computing (RASC2002), pp. 307-312.
- [70] Gomes, L. C. T. and Von Zuben, F. J., (2002a), “Vehicle Routing Based on Self-Organization with and without Fuzzy Inference”, Proc. of the IEEE International Conference on Fuzzy Systems, vol 2, pp. 1310-1315, in the IEEE WCCI'2002.
- [71] Gomes, L. C. T., Von Zuben, F. J., (2002b), “A Neuro-Fuzzy Approach to the Capacitated Vehicle Routing Problem”, Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN' 2002 , vol 2, pp. 1930-1935, in the 2002 IEEE World Congress on Computational Intelligence (WCCI' 2002).
- [72] Gomes, L. C. T., Von Zuben, F. J., (2002c), “Vehicle Routing Based on Self-Organization with and without Fuzzy Inference”, Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE' 2002) , vol 2, pp.1310-1315, in the 2002 IEEE World Congress on Computational Intelligence (WCCI' 2002) .
- [73] Gomes, L. C. T. e Von Zuben, F. J., (2002d), “Uma Heurística Baseada em Aprendizado Competitivo e Inferência Fuzzy Aplicada ao Problema de Roteamento de Veículos Capacitados”, XIV Congresso Brasileiro de Automática (CBA 2002), pp. 2504-2510.
- [74] Gomes, L. C. T. and Von Zuben, F. J., (2002e), “A Heuristic Method Based on Unsupervised Learning and Fuzzy Inference for the Vehicle Routing Problem”, Proc. of VII Brazilian Symposium on Neural Networks, pp. 130-135.

- [75] Gomes, L. C. T., Von Zuben, F. J. and Moscato, P., (2002f), “Ordering Microarray Gene Expression Data Using a Self-Organising Neural Network”, Proceedings of the 4<sup>th</sup> International Conference on Recent Advances in Soft Computing, pp. 307-313.
- [76] Gomes, L. C. T., Von Zuben, F. J. and Moscato, P., (2002g), “Ordering Gene Expression Data Using Self-Organizing Maps”, Proceedings of 1<sup>st</sup> Brazilian Workshop on Bioinformatics, pp. 91-93.
- [77] Gomes, L.C.T., de Sousa, J.S., Bezerra, G.B., de Castro, L.N. and Von Zuben, F.J., “opt-aiNet and the Gene Ordering Problem”, Proceedings of the Second Brazilian Workshop on Bioinformatics (WOB’2003), Macaé, RJ, pp. 28-37, 2003.
- [78] Gordon, A. D., (1999), “Classification”, Chapman Hall CRC Press, 2<sup>nd</sup> Edition.
- [79] Hajela, P. and Lee, J., (1996), “Constrained Genetic Search via Schema Adaptation: An Immune Network Solution”, Structural Optimization, vol 12, n.1, pp. 11-15.
- [80] Hajela, P., Yoo, J. and Lee, J., (1997), “GA Based Simulation of Immune Networks – Applications in Structural Optimization”, Journal of Engineering Optimization, vol 29, pp. 131-149.
- [81] Hart, W. E., (1994), “Adaptive global optimization with local search”, Ph.D. Thesis, University of California, San Diego.
- [82] Haykin, S., (1999), “Neural Networks: A Comprehensive Foundation”, 2<sup>nd</sup> Edition, Prentice Hall.
- [83] Hirota, K. and Pedrycz, W., (1999), “Fuzzy Computing for Data Mining”, Proceedings of the IEEE, vol 87, n. 9, pp. 1575-1600.
- [84] Holland, J. H., (1975), “Adaptation in Natural and Artificial Systems”, University of Michigan Press.
- [85] Holland, J. H., (1992), “Adaptation in Natural and Artificial Systems”, MIT Press, Cambridge.

- [86] Honkela, T., (1997), “Data Mining and Document Modeling”. Collected Papers of the Spring 1997 Postgraduate Seminar.
- [87] Hopfield, J. J., (1982), “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”, Proc. National Academics of Sciences, vol 79, pp. 2554-2558.
- [88] Hopfield, J. J. and Tank, D. W., (1985), “Neural Computation of Decisions in Optimization Problems”, Biological Cybernetics, 52, pp. 141-152.
- [89] Hunt, J. E. and Cooke, D. E., (1996), “Learning Using an Artificial Immune System”, Journal of Network and Computer Applications, vol 19, pp. 189-212.
- [90] Iivarinen, J. and Visa, A., (1998), “Unsupervised Image Segmentation with the Self-Organizing Map and Statistical Methods”, In Casasent, D.P. (ed.), Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Proc. SPIE 3522, pp. 516-526.
- [91] Ishiguro, A., Kondo, T., Watanabe Y., and Uchikawa, Y., (1995), “Dynamic Behavior Arbitration of Autonomous Mobile Robots Using Immune Networks”, In Proceedings of 8 IEEE International Conference on Evolutionary Computation, pp. 722-727.
- [92] Jain, A. K., Murty M. N. and Flynn, P. J., (1999), “Data Clustering: A Review”, ACM Computer Surveys, vol 31, n. 3, pp. 264-323.
- [93] Jain, L. C. & Jain, R. K., Hybrid Intelligent Engineering Systems (Advances in Fuzzy Systems – Application and Theory) (1997), World Scientific Publishing.
- [94] Janeway, C., (2001), “Immunobiology: The Immune System in Health and Disease”, Garland Publishing.
- [95] Jerne, N. K., (1974), “Towards a Network Theory of the Immune System”, Ann. Immunol. (Inst. Pasteur) 125C, pp. 373-389.

- [96] Jünger, M. and Thienel, S., (1998), “Introduction to ABACUS – A Branch-And-Cut System”, *Operations Research Letters*, vol 22, pp. 83-95.
- [97] Kalyanmoy, D., (2002), “Multiobjective Optimization using Evolutionary Algorithms”, Wiley.
- [98] Kennedy, J. and Eberhart, R. C., (1995), “Particle swarm optimization”, *Proc. IEEE int' l conf. on neural networks* vol 4, pp. 1942-1948. IEEE service center, Piscataway, NJ.
- [99] Kennedy, J., Eberhart, R. C., Shi, Y., (2001), “Swarm Intelligence”, Morgan Kaufmann Publishers.
- [100] Kennedy, M. and Chua, L., (1987), “Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin”, *Circuits and Systems, IEEE Transactions on*, vol 34, Issue: 2, pp. 210- 214.
- [101] Kepler, T. B. and Perelson, A. S., (1993), “Somatic Hypermutation in B cells: an Optimal Control Treatment”, *J. Theor. Biol.*, vol 164, pp. 37-64.
- [102] Kirby, W. M., (2003), “Operational Research in War and Peace – The British Experience from the 1930s to 1970”, Imperial College Press.
- [103] Kohonen, T., (1982a), “Self-Organizing Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, vol 43, n.1, pp. 56-59.
- [104] Kohonen, T., (1982b), “Analysis of a Simple Self-Organizing Process”, *Biological Cybernetics*, vol 44, n.2, pp. 135-140.
- [105] Kohonen, T., (1984), “Self-Organization and Associative Memory”, Springer, Berlin, Heidelberg.
- [106] Kohonen, T., (1997). “Self-Organizing Maps”, 2nd Edition, Springer Verlag.
- [107] Kohonen, T., Huang, T. S. and Schoreder, M. R., (Editors), (2001), “Self-Organizing Maps”, Spring.

- [108] Kung, S.-Y., Taur, J. and Lin, S.-H., (1999), “Synergistic Modeling and Applications of Hierarchical Fuzzy Neural Networks”, *Proceedings of the IEEE*, vol 87, n. 9, pp. 1550-1574.
- [109] Lagus, K., Honkela, T., Kaski, S. and Kohonen, T., (1996), “Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration”, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 238-243.
- [110] Laporte, G. and Novert, Y., (1987), “Exact Algorithms for the Vehicle Routing Problem”, *Annals of Discrete Mathematics*, vol 31, pp. 147-184.
- [111] Laporte, G. and Demet, F., (2002), “Classical Heuristics for the Capacitated VRP”, in “The Vehicle Routing Problem”, Toth, P, Vigo, D. (editors), *Society for Industrial and Applied Mathematics (SIAM)*.
- [112] Lawler, E. L. and Wood, D. E., (1966), “Branch-And-Bound Methods: A Survey”, *Operations Research*, vol 14, n. 4 (Jul. - Aug., 1966), pp. 699-719.
- [113] Li, T., Fang, L. and Li, Q.-Q., (1993), “Hierarchical Classification and Vector Quantisation with Neural Trees”, *Neuro Computing*, vol 5, pp. 119-139.
- [114] Lima, C. A. M. (2004), “Comitê de Máquinas: Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte”, *Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (UNICAMP)*.
- [115] Luscombe, N., Greenbaum, D. and Gerstein, M., (2001), “What is bioinformatics? A proposed definition and overview of the field.”, *Methods Inf Med*. 2001, vol 40, n.4, pp. 346-58.
- [116] Marsland, S., Shapiro, J., and Nehmzow, U., (2002), “A self-organizing network that grows when required”, *Neural Networks*, 2002, pp. 1041-1058.

- [117] Matsuyama, Y., (1991), “Self-Organization via Competition, Cooperation and Categorization Applied to Extended Vehicle Routing Problems”, Proc. International Joint Conference on Neural Networks, vol 1, pp. 385-390.
- [118] Matsuyama, Y., (1996), “Harmonic Competition: A Self-Organizing Multiple Criteria Optimization”, IEEE Transactions on Neural Networks, vol 7, n.3, pp. 652-668.
- [119] Maturana, H. R. and Varela, F. F., (1980), “Autopoiesis and Cognition: The Realization of the Living”, Knowledge-Based Intelligent Information and Engineering Systems, Springer, pp. 22-29.
- [120] McCulloch, W. S. and Pitts, W. (1943), “A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biophysics, vol 5, pp.115-133.
- [121] Merz, P. and Zell, A., (2002), “Clustering Gene Expression Profiles with Memetic Algorithms”, Proceedings of the 7th International Conference on Parallel Problem Solving from Nature - PPSN VII, Lecture Notes in Computer Science 2439, Springer, Berlin, Heidelberg, pp. 811-820.
- [122] Michalewicz, Z. and Fogel, D. B., (2002), “How to Solve It: Modern Heuristics”, Springer Verlag.
- [123] Michelin, R. (2003), “Evolução de Redes Imunológicas para Coordenação Automática de Comportamentos Elementares em Navegação Autônoma de Robôs”, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação – Unicamp, Julho, 2003.
- [124] Miller, B. L. and Shaw, M. J., (1996), “Genetic algorithms with dynamic niche sharing for multimodal function optimization”, in Proc. 1996 IEEE Int.Conf. Evolutionary Computation. Piscataway (NJ: IEEE Press), pp. 786-791.
- [125] Modares, A., Somhom, S. and Enkawa, T., (1999), “A Self-Organizing Neural Network Approach for Multiple Traveling Salesman and Vehicle Routing Problems”, Int. Transactions in Operational Research, vol 6, pp. 591-606.

- [126] Nadjah, N. and Mourelle, L. M., (2006), "Fuzzy Systemas Engineering: Theory and Practice", Springer.
- [127] Nossal, G.J. V., (1993), "The Molecular and Cellular Basis of Affinity Maturation in the Antibody Response", *Cell*, vol 68, pp. 1-2.
- [128] Park, M-Y. and Hastie, T., (2006), "Regularization Path Algorithms for Detecting Gene Interactions".
- [129] Pedrycz, W. and Gomide, F., (1998), "An Introduction to Fuzzy Sets", MIT Press.
- [130] Petrank, E., (1995), "Knowledge Complexity Versus Computational Complexity", PhD thesis, Israel Institute of Technology.
- [131] Potvin, J.-I., Robillard, C, (1995), "Clustering for Vehicle Routing with a Competitive Neural Network", *Neurocomputing*, vol 8, pp. 125-139.
- [132] Reed, R. D. and Marks II, J. R., (1999), "Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks", MIT Press.
- [133] Rees, S. and Ball, R. C., (1987), "Criteria for an Optimum Simulated Annealing Schedule for Problems of the Travelling Salesman Type", *J. Physics, A: Math. Gen.*, vol 20, pp. 1239-1249.
- [134] Reeves, C., (1993), "Modern Heuristic Techniques for Combinatorial Problems", Wiley & Sons.
- [135] Regattieri, M. (2002), "Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva", Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (UNICAMP).
- [136] Reinelt, G., (1991), "TSPLIB - A Traveling Salesman Problem Library", *ORSA Journal on Computing*, vol 3, pp. 376-384.
- [137] Ritter, H. J., Martinetz, T. and Schulten, K., (1992), "Neural Computation and Self-Organizing Maps: An Introduction", Addison Wesley, Reading, MA.

- [138] Sakawa, M., (1993), “Fuzzy sets and interactive multiobjective optimization”, Plenum Press.
- [139] Seiffert, U. and Michaelis, B., (1995), “Three-Dimensional Self-Organizing Maps for Classification of Image Properties”, Proc. of the International Conference on Artificial Neural Networks and Expert Systems, pp. 310-313.
- [140] Schena, M., (1996), “Genome Analysis with Gene Expression Microarrays”, Bioessays, vol 18, pp. 427-431.
- [141] Simões, A. B., (1999), “Transposição : estudo de um novo operador genético inspirado biologicamente”, Dissertação de mestrado, Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.
- [142] Smith, K. A., (1999), “Neural Networks for Combinatorial Optimization: A Review of More than a Decade of Research”, INFORMS Journal on Computing, vol 11, n. 1, pp. 15-34.
- [143] Soylu, M., Ozdemirel, N. E., and Kayaligil, S. (2000), “Self-organizing neural network approach for the single AGV routing problem, European Journal of Operational Research, vol 121, n.1, pp. 124-137.
- [144] Sutton, R. S. and Barton, A. G. (1998), “Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)”, MIT Press.
- [145] Taillard, E. D., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J. Y., (1997), “A tabu search heuristic for the vehicle routing problem with soft time windows”, Transportation Science, vol 31, pp. 170-186.
- [146] Toth, P. and Vigo, D., (2002), “The Vehicle Routing Problem”, Society for Industrial and Applied Mathematics (SIAM).
- [147] Tsai, H.-K., Yang, J.-M., Kao, C.-Y., (2002), “Applying Genetic Algorithms to Finding the Optimal Gene Order in Displaying the Microarray Data”, in the Proc. of

- The Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 610-617.
- [148] Tsubakitani, S. and Evans, J. R., (1998), "Optimizing Tabu List Size for the Traveling Salesman Problem", *Computers and Operations Research*, vol 25, pp. 91-97.
- [149] Vakhutinsky, A. I. and Golden, B. L., (1994), "Solving Vehicle Routing Problems Using Elastic Nets", *Proc. of IEEE International Conference on Neural Networks*, vol 7, pp. 4535-4540.
- [150] Vicente, L. and Vellido, A., (2004), "Review of Hierarchical Models for Data Clustering and Visualization", In R.Giráldez, J.C. Riquelme, J.S. Aguilar-Ruiz (eds.) *Tendencias de la Minería de Datos en España. Red Española de Minería de Datos*.
- [151] Whitley, D., Rana, S. and Heckendorn, R. B., (1999), "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence", *Journal of Computing and Information Technology* vol 7, pp. 33-47.
- [152] Xiao, X., Dow, E., Eberhart, R., Miled, Z. B. and Oppelt, R. J., (2003), "Gene Clustering using Self-Organizing Maps and Particle Swarm Optimization", *Second IEEE International Workshop on High Performance Computational Biology (HiCOMB 2003)*.
- [153] Yager, R. R. and Filev, D. P. (1994), "Essentials of Fuzzy Modeling and Control", Wiley.
- [154] Yager, R., Ovchinnikov, S., Tong, R. and Ngugen, H., (eds), (1987), "Fuzzy Sets and applications", Wiley, New York.
- [155] Yager, R. R.. and Zadeh, L. A., (eds), (1994), "Fuzzy Sets, Neural Networks and Soft Computing", Van Nostrand Reinhold, New York.
- [156] Zachariasen, M. and Dan, M., (1996), "Tabu Search on the Geometric Traveling Salesman Problem", in *Meta-Heuristics: Theory and Applications (Proceedings from Metaheuristics International Conference, Colorado)*, pp. 571–587.

- [157] Zadeh, L. A., (1965), “Fuzzy Sets”, *Inform.Control*, 8, pp. 338-353.
- [158] Zikidis, K. C. and Tzafestas, S. G., (2003), “ART-Based Neuro-Fuzzy Modelling Applied to Reinforcement Learning”, *Knowledge-Based Intelligent Information and Engineering Systems*, Springer, pp. 22-29.