DANIEL GIBILINI

**APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA
ARTIFICIAL NA ALOCAÇÃO DINÂMICA DE
CANAIS EM REDES SEM FIO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia.

São Paulo
2006

DANIEL GIBILINI

**APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA
ARTIFICIAL NA ALOCAÇÃO DINÂMICA DE
CANAIS EM REDES SEM FIO**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia.

Área de Concentração:

Sistemas Digitais

Orientador:

Prof. Dr. Marco Túlio Carvalho de Andrade

São Paulo
2006

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 28 de abril de 2006.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Gibilini, Daniel

Aplicação de Técnicas de Inteligência Artificial na Alocação Dinâmica de Canais em Redes Sem Fio/ D. Gibilini. – ed. rev. – São Paulo, 2006.

140 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Inteligência artificial 2.Sistemas híbridos 3.Wireless 4.Simulação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. t.

Para minha mãe, que foi o início de tudo.

Para Lilian, que acompanhou a conclusão.

AGRADECIMENTOS

Este texto é o resultado de pesquisas, dúvidas, dificuldades, trabalhos e muita, muita ajuda.

Em primeiro lugar, sou grato aos meus pais, responsáveis pelo meu desenvolvimento inicial. Em especial, agradeço a minha mãe, que sempre considerou a educação essencial e esforçou-se muito para oferecer boas oportunidades aos filhos.

Agradeço a Lilian pela infinita compreensão e pelo auxílio na preparação do texto; Cristina, pelo incentivo; e Francisco Méa, por todo o suporte oferecido.

Mesmo correndo o risco de suprimir algum nome, gostaria de agradecer: meu orientador, Prof. Dr. Marco Túlio Carvalho de Andrade, pelo acompanhamento, estímulo e paciência durante o tumultuado período da minha pós-graduação; Prof. Dr. Paulo Sérgio L. M. Barreto, pelo suporte, principalmente, em simulação de sistemas; Humberto Sandmann, pelas explicações sobre redes neurais artificiais; Prof. Dr. Edson Satoshi Gomi, pelos apontamentos e observações; Prof. Dr. Jaime Simão Sichman, pelos esclarecimentos sobre sistemas multi-agentes; Prof. Dr. Reginaldo Arakaki, pelo auxílio em modelagem de sistemas; Rosângela de Paula Amorim, pela ajuda com dados e camada de transmissão de redes celulares.

Devo, também, mencionar a compreensão de Raphael Balabanian e Marcos Anzai, que tornaram possível a conclusão dos créditos necessários.

Por fim, peço perdão aos demais que não estão aqui mencionados. Espero que se sintam representados e incluídos em meus agradecimentos.

RESUMO

Nos últimos anos, as redes de comunicação móveis se tornaram de fundamental importância para a infraestrutura dos sistemas de comunicação. Uma das áreas de maior crescimento é a computação móvel. Realizada através de sinais de rádio, a quantidade de canais disponíveis raramente é suficiente para atender a crescente demanda.

Este trabalho apresenta uma solução para a questão da alocação de canais, um tópico desafiador dentro da área de redes móveis. A implementação de alocação dinâmica com uso de técnicas computacionais clássicas melhora a utilização dos recursos disponíveis, mas necessita de ajustes periódicos para se adequar a novos cenários. Para a construção de um sistema mais flexível e adaptável, a abordagem escolhida utiliza técnicas de Inteligência Artificial.

O modelo proposto combina Teoria Nebulosa, Redes Neurais Artificiais e Sistemas Multi-Agentes. As características de cada técnica foram analisadas e identificamos as partes do sistema que poderiam ser beneficiadas por cada uma.

O sistema é resultado da combinação coordenada das três técnicas, e constitui um método eficiente e flexível para gerenciamento de recursos de rádio.

Após o detalhamento do modelo, realizamos uma simulação de uma rede celular com o sistema proposto e seu comportamento é comparado com uma rede de referência, para verificação das diferenças e melhorias alcançadas.

Por fim, apresentamos a situação atual da pesquisa e os possíveis caminhos para aprimoramento do sistema.

Palavras-chave Inteligência Artificial, Teoria Nebulosa, Redes Neurais Artificiais, Sistemas Distribuídos, Sistemas Multi-Agentes, Redes sem Fio, Alocação Dinâmica de Canais.

ABSTRACT

In the last years, mobile networks became more important for communication systems' infrastructure. One area of great growth is mobile computation, which is performed through radio signals. The amount of available channels rarely is enough to attend the increasing demand.

This work presents a solution for the channel allocation topic, a challenging topic inside mobile networks area. The implementation of dynamic allocation using classic computational techniques improves the use of available resources, but it needs periodic and frequent adjustments for new scenarios. The construction of a more flexible and adaptable system was achieved using Artificial Intelligence techniques.

Proposed model combines Fuzzy Logic, Artificial Neural Networks and Multi-Agents Systems. Features of each technique had been analyzed and we identified the system modules which could be benefited by them.

The system is the result of coordinated combination of these three techniques, and constitutes an efficient and flexible method for radio resources management.

After model detailing, we executed a cellular network simulation using proposed system, and its behavior is compared with a reference network, presenting reached differences and improvements.

Finally, we present current situation of this research and possible ways for system improvement.

Keywords Artificial Intelligence, Fuzzy Logic, Artificial Neural Networks, Distributed Systems, Multi-Agents Systems, Wireless Networks, Dynamic Channel Allocation.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

1	Introdução	1
1.1	Escopo do Problema	2
1.2	Motivação	3
1.3	Justificativa	4
1.4	Objetivos	5
1.5	Método	5
1.6	Estrutura	6
2	Detalhamento do Problema	7
2.1	Redes sem Fio	8
2.1.1	Distribuição de Recursos	10
2.1.1.1	Realocação Dinâmica	14
2.2	Trabalhos Relacionados	15
2.2.1	Métrica	16
3	Inteligência Artificial	17
3.1	Teoria Nebulosa	19

3.1.1	Teoria de Conjuntos Nebulosos	21
3.1.1.1	Medições utilizando a Teoria Nebulosa	23
3.1.1.2	Conjuntos Nebulosos	23
3.1.1.3	Variáveis Lingüísticas	27
3.1.1.4	Regras de Inferência	28
3.1.1.5	Conversão de Variáveis Lingüísticas	29
3.2	Redes Neurais Artificiais	30
3.2.1	Redes Neurais em Camadas (RNC)	31
3.2.2	Aprendizado	34
3.2.2.1	Lei de Hebb	36
3.2.2.2	Regra Delta	37
3.2.2.3	Retropropagação	37
3.2.3	Implementação	39
3.3	Sistemas Multi-Agentes	39
3.3.1	Modelo BDI	43
3.3.2	Comunicação	45
3.3.2.1	Linguagens	45
4	Sistema Proposto	47
4.1	Visão Geral	48
4.2	Agentes do Sistema	48
4.2.1	Sensor	50

4.2.2	Comunicador	50
4.2.2.1	Sistema de Controle	51
4.2.2.2	Agentes Vizinhos	51
4.2.3	Base de Conhecimento	53
4.2.3.1	Vizinhos	53
4.2.3.2	Canais da Rede	54
4.2.3.3	Canais Alocados	55
4.2.3.4	Canais em Uso	56
4.2.3.5	Canais Oferecidos	56
4.2.3.6	Horário	57
4.2.3.7	Utilização	57
4.2.3.8	Utilização Prevista	58
4.2.3.9	Utilização Passada	59
4.2.3.10	Ocupação	59
4.2.3.11	Situação	60
4.2.3.12	Regras de Inferência	61
4.2.3.13	Passo de Previsão	61
4.2.3.14	Rede Neural de Previsão	62
4.2.3.15	Passo de Treinamento	66
4.2.4	Decisor	67
4.2.4.1	Tomada de Decisão	68
	Neutro	69

Captação	69
Liberação	70
4.2.4.2 Recepção de Eventos	72
Conexão	72
Desconexão	72
4.2.4.3 Recepção de Mensagens	72
Iniciação	73
Consulta	73
Atualização	74
Aviso	74
Alocação	74
Liberação	75
Solicitação	75
Colaboração	76
4.2.5 Executor	77
4.2.5.1 Atualizações	77
4.2.5.2 Comunicações	78
4.2.5.3 Comandos	79
4.3 Sistema de Controle	80
4.4 Funcionamento	81
4.4.1 Chegada de pedido de conexão	82
4.4.2 Aviso de desconexão	84

4.4.3	Sistema de Controle	85
4.4.4	Alocação e Liberação de Canais	87
4.4.5	Solicitação	88
4.4.6	Colaboração	89
5	Simulação de Operação	91
5.1	Cenário da Simulação	91
5.1.1	Região	91
5.1.2	Demanda	91
5.2	Eventos	92
5.2.1	Geração	93
5.2.1.1	Distribuições	96
5.3	Implementação	97
5.3.1	Controle	98
5.3.1.1	Eventos	99
5.3.1.2	Variáveis e Previsões	99
5.3.2	Fluxo de Execução	100
	Controle	100
	Agentes	100
5.3.3	Rede Neural	101
5.3.4	Teoria Nebulosa	102
5.3.5	Registros de Teste	102

5.4	Execução	102
5.4.1	Definição e Distribuição de Canais	103
5.4.2	Simulação das Redes	104
6	Análise dos Resultados	107
6.1	Rede Neural	107
6.1.1	Verificações	108
	Ajuste de Limites	110
6.1.1.1	Validação I	110
6.1.1.2	Validação II	112
6.1.1.3	Validação III	114
6.1.2	Análise das Validações	117
6.1.3	Implementação	117
6.2	Simulação	118
6.2.1	Mapas Comparativos	119
6.2.2	Chamadas e Bloqueios	120
6.2.3	Realocação de Canais	121
7	Considerações Finais	123
7.1	Cumprimento dos Objetivos	123
7.2	Contribuições	124
7.3	Otimizações	125
7.4	Trabalhos Futuros	125

Referências	127
Apêndice A – Resultados: Figuras	133
A.1 Mapas Comparativos	133
A.2 Chamadas	137

LISTA DE FIGURAS

2.1	Exemplo de região coberta por células hexagonais.	10
2.2	Região coberta por 7 células hexagonais.	11
3.1	Quadro de Kanizsa.	18
3.2	Função de pertinência do conjunto $\mathbb{A}$	21
3.3	Representação Gráfico do conjunto Nebuloso $\mathbb{E}$	25
3.4	Representação das operações de disjunção e conjunção.	26
3.5	Exemplo de Variável Lingüística.	28
3.6	Perceptron simples.	31
3.7	Rede Multicamada.	32
3.8	Arquiteturas RNC.	34
3.9	Modelo geral de agente.	40
3.10	Arquitetura BDI.	44
4.1	Estrutura interna do agente utilizado.	49
4.2	Região coberta por células hexagonais.	54
4.3	Conjuntos referentes à variável nebulosa <i>utilização</i>	58
4.4	Conjuntos referentes à variável nebulosa <i>ocupação</i>	60
4.5	Estrutura da Rede Neural Artificial de Previsão.	63
4.6	Curva da tangente hiperbólica (<i>tanh</i>).	63
4.7	Canais utilizados em uma célula no período de 7 dias.	64

4.8	Canais utilizados em uma célula no período de 1 dia.	65
4.9	Conjuntos referentes à variável nebulosa <i>utilização prevista</i>	66
4.10	Conjuntos referentes à variável nebulosa <i>situação</i>	69
4.11	Estabelecimento de uma conexão com utilização de canais ociosos. . .	82
4.12	Estabelecimento de uma conexão com utilização de canais livres. . . .	83
4.13	Estabelecimento de uma conexão com solicitação de canais.	84
4.14	Finalização de uma conexão.	84
4.15	Finalização de uma conexão.	85
4.16	Consulta de parâmetros pelo <i>Sistema de Controle</i>	85
4.17	Atualização de parâmetros pelo <i>Sistema de Controle</i>	86
4.18	Iniciação de um agente.	86
4.19	Alocação de canais.	87
4.20	Liberação de canais.	87
4.21	Alocação de canais.	88
5.1	Células hexagonais definidas para a execução da simulação.	92
5.2	Curvas Características.	93
5.3	Curva característica: utilização média de canais no período de um dia. .	93
5.4	Curva gerada por um conjunto de eventos.	96
5.5	Algoritmo utilizado na distribuição inicial de canais.	105
6.1	Diagrama da rede neural criada no MatLab.	108
6.2	Curva característica da Validação I.	111
6.3	Trecho da curva de utilização gerada para a Validação I.	111

6.4	Validação I: valores esperados e previstos.	112
6.5	Curva característica da Validação II.	112
6.6	Trecho da curva de utilização gerada para a Validação II.	113
6.7	Validação II: valores esperados e previstos.	113
6.8	Trecho da curva de utilização gerada para a Validação III.	114
6.9	Detalhe da Figura 6.8 (quinto e sexto dias).	115
6.10	Validação III: relativo a 5 dias da semana.	116
6.11	Validação III: relativo a 2 dias da semana.	116
6.12	Respostas para <i>Curva 3</i> (Treinamento com MatLab e Joone).	118
6.13	Respostas para <i>Curva 3</i> (Treinamento com MatLab).	118
6.14	Mapa comparativo da Simulação 10.	120
6.15	Comparativo de chamadas da simulação 10.	120
6.16	Comportamento da célula 10 (simulação 10).	121
6.17	Comportamento da célula 17 (simulação 10).	121
6.18	Canais alocados na célula 10 (simulação 10).	122
A.1	Mapa comparativo da Simulação 1.	133
A.2	Mapa comparativo da Simulação 2.	133
A.3	Mapa comparativo da Simulação 3.	134
A.4	Mapa comparativo da Simulação 4.	134
A.5	Mapa comparativo da Simulação 5.	134
A.6	Mapa comparativo da Simulação 6.	135
A.7	Mapa comparativo da Simulação 7.	135

A.8	Mapa comparativo da Simulação 8.	135
A.9	Mapa comparativo da Simulação 9.	136
A.10	Mapa comparativo da Simulação 10.	136
A.11	Comparativo de chamadas da simulação 1.	137
A.12	Comparativo de chamadas da simulação 2.	137
A.13	Comparativo de chamadas da simulação 3.	138
A.14	Comparativo de chamadas da simulação 4.	138
A.15	Comparativo de chamadas da simulação 5.	138
A.16	Comparativo de chamadas da simulação 6.	139
A.17	Comparativo de chamadas da simulação 7.	139
A.18	Comparativo de chamadas da simulação 8.	139
A.19	Comparativo de chamadas da simulação 9.	140
A.20	Comparativo de chamadas da simulação 10.	140

LISTA DE TABELAS

2.1	Alocação inicial de canais.	11
2.2	Canais disponíveis para alocação.	12
2.3	Alocação após expansão da célula A.	12
2.4	Alocação após expansão das demais células.	12
2.5	Configuração após realocação de c_1	13
2.6	Configuração após redução dos recursos nas células A e C.	14
4.1	Distribuição de canais nas células.	53
4.2	Relação de vizinhos do agente presente na célula B.	54
4.3	Definição dos índices de separação.	55
4.4	Índices de separação dos canais da rede relativos à Célula A.	56
4.5	Definição da variável <i>situação</i> a partir das combinações das entradas.	70
6.1	Conjunto de eventos.	109
6.2	Lista (<i>início</i> , <i>fim</i>).	109
6.3	<i>Lista de utilização</i>	110
6.4	Chamadas bloqueadas em cada simulação.	119

1 INTRODUÇÃO

Nos últimos anos observou-se um enorme avanço nas áreas de computação e telecomunicação (LINS, 2001; GRIFFA, 2005; SOKOLOV, 2006). O poder de processamento e a capacidade de armazenamento de sistemas computadorizados evoluíram de forma extraordinária, bem como a capacidade e a velocidade das conexões presentes em redes de computadores.

A tecnologia de rede aprimorou-se e difundiu-se e hoje já não é possível conceber a computação isolada, onde um equipamento opera sozinho sem a necessidade de comunicar-se com outras máquinas para realizar suas tarefas. Um dos fatores importantes para a disseminação das redes de computadores foi o surgimento da Internet, a grande rede mundial que permite que computadores em praticamente qualquer ponto do globo se comuniquem, possibilitando, de forma rápida e simples, interconexões, interações, compartilhamentos de dados e cooperações entre equipamentos.

A complexidade das redes aumenta continuamente, exigindo esforços cada vez maiores em projeto, especificação, implantação e, principalmente, em gerenciamento e manutenção. A quantidade de variáveis envolvidas e a velocidade com que o estado da rede é alterado tornam complexas a sua configuração e o seu ajuste, sendo, algumas vezes, impossível acompanhar a variação da rede de forma satisfatória.

Atualmente, as redes móveis possuem importância cada vez maior na infraestrutura dos sistemas de comunicação. Entretanto, seu futuro depende de uma série de questões que devem ser respondidas por pesquisas na área de redes sem fio. A limi-

tação de banda e a interferência entre canais de células vizinhas tornam necessário um bom método de alocação de canais para possibilitar o fornecimento de serviços confiáveis e satisfatórios aos usuários.

Este trabalho apresenta uma solução para a questão da alocação de canais, um tópico desafiador dentro da área de redes móveis. O modelo proposto usa técnicas de Inteligência Artificial para o desenvolvimento de um eficiente sistema distribuído para realocação dinâmica de canais.

1.1 Escopo do Problema

A demanda por comunicação móvel cresce continuamente enquanto o espectro eletromagnético reservado para as redes sem fio é limitado. Como consequência, o gerenciamento eficiente dos recursos de rádio é uma das questões mais importantes dentro da área, sendo abordado em um grande número de trabalhos ((PATTAVINA; QUADRÌ; TRECORDERI, 1999; SHEN; MARK; YE, 2000; ZHANG; DAS; JIA, 2004; BAIOCCHI; SESTINI, 1996; BOUKERCHE; HONG; JACOB, 2002; BOUKERCHE; EL-KHATIB; HUANG, 2004; LAWLOR; WHITE, 2003)).

A comunicação em redes sem fio é realizada através do uso de frequências de rádio que, normalmente, são divididas em *canais de rádio*. Esses canais podem ser utilizados ao mesmo tempo em diferentes regiões, mas canais adjacentes utilizados em locais próximos podem causar interferências.

Os canais podem ser divididos de diferentes formas e três técnicas predominam atualmente: Divisão por Frequência, Divisão por Tempo e Divisão por Código. Na primeira, os usuários utilizam diferentes faixas de frequências para comunicação. Na segunda, a mesma faixa de frequência transporta toda a comunicação e o tempo de comunicação é dividido em intervalos de tempo (*timeslots*). Cada usuário recebe um *timeslot* para realizar a comunicação. A divisão por código consiste na divisão dos

canais por códigos de modulação. Todas as faixas de frequências são utilizadas pelas chamadas, que compartilham a banda utilizando códigos individuais. Essas técnicas podem ser combinadas, resultando em mais métodos para divisão dos canais de rádio. Entretanto, independente da técnica utilizada, a divisão não pode ser realizada indefinidamente, pois deve-se garantir um mínimo de qualidade para cada canal.

A região geográfica atendida por uma rede sem fio, a *área de cobertura*, é particionada em diversas regiões adjacentes denominadas *células*. Cada célula possui uma *estação base* que atende certo número de *usuários móveis*. Estações bases próximas são agrupadas e controladas por um *controlador de estações base*, que executa algumas funções centralizadas. Diversos controladores e estações base são atendidos por uma *central de comutação móvel*, que gerencia uma grande área geográfica e serve de interface para outras redes, sejam elas móveis ou fixas. Quando um usuário deseja realizar uma comunicação, ele envia um sinal à estação base que, de acordo com o método de alocação implementado, escolhe um canal disponível para atender o usuário ou, caso não existam recursos disponíveis, simplesmente rejeita a chamada.

1.2 Motivação

Embora uma rede bem projetada e dimensionada permita um bom mapeamento da região e uma distribuição satisfatória de canais, a computação móvel possui características intrínsecas que tornam impossível alcançar uma ótima distribuição fixa. Os equipamentos podem se conectar à rede de lugares diferentes, em movimento e, ainda, podem apresentar concentrações diferentes a cada momento.

A mobilidade dos usuários traz grandes dificuldades para a distribuição dos recursos da rede. Uma rede móvel planejada com recursos suficientes para atender os usuários nos horários de maior movimento terá recursos ociosos nos demais horários. No outro extremo, uma rede planejada para atender os usuários apenas nos horários de

baixo tráfego apresentará congestionamentos e negações de pedidos de conexão nos horários de pico. Mesmo uma rede planejada para atender a média da demanda não será uma solução ótima, uma vez que não suportará todos os usuários nos períodos de alto tráfego e possuirá recursos ociosos nos períodos de baixo tráfego.

As redes cabeadas, devido à própria natureza do meio de transmissão, podem ter sua capacidade expandida apenas com investimentos em equipamentos e cabeamento. Por outro lado, as limitações impostas pelo meio nas transmissões de rádio dificultam as ampliações das redes sem fio, restringindo sua abrangência e capacidade.

Assim, para proporcionar crescimento, aumento de velocidade e melhoria de qualidade das redes sem fio, a solução proposta nesta pesquisa visa obter uma gerência eficiente dos recursos disponíveis, maximizando a utilização dos canais existentes.

1.3 Justificativa

Uma das maiores dificuldades no gerenciamento de redes sem fio é o controle e a utilização eficaz do meio de transmissão, que é compartilhado por todos os equipamentos que desejam transmitir ou receber dados em uma determinada região geográfica. Além da quantidade limitada de recursos disponíveis, os sinais ainda enfrentam outros problemas, como interferências ocasionadas pelo clima, por fontes de radiação eletromagnética e por objetos que degradam ou atenuam o sinal.

Dessa forma, a alocação ótima de recursos não é apenas uma forma de melhorar as condições da rede, mas uma necessidade para que os serviços sejam prestados de forma satisfatória e seus usuários sejam atendidos.

Outro ponto importante é a falta de equipamentos e sistemas capacitados para a realização de otimizações desse tipo.

Esses aspectos nos trazem ao sistema proposto, que tem o objetivo de suprir as deficiências existentes nos sistemas em uso no mercado.

1.4 Objetivos

O objetivo principal deste trabalho é obter um sistema eficiente e adaptável para realizar a alocação dinâmica de canais em uma rede sem fio.

Além disso, analisamos as vantagens proporcionadas pela aplicação de técnicas de Inteligência Artificial na solução da questão, avaliando o ganho oferecido na resolução de problemas específicos.

Com o sistema, objetiva-se oferecer maior capacidade à rede, proporcionar melhor aproveitamento dos recursos disponíveis e garantir a qualidade dos serviços prestados. Sua arquitetura deve possibilitar a inclusão de módulos à infra-estrutura das redes em operação atualmente, sem necessidade de grandes investimentos.

1.5 Método

O problema apresentado foi analisado e dividido em itens específicos, com o objetivo de identificar as técnicas que oferecem as melhores soluções para cada caso.

O comportamento dinâmico dos usuários e a constante alteração de demanda na rede sugere que a alocação dos recursos de rádio deva ser dinâmica, de forma a acompanhar as variações de sua utilização. A implementação de alocação dinâmica com base em técnicas clássicas melhora o uso dos recursos disponíveis, mas necessita de ajustes periódicos para se adequar a novos cenários. Uma abordagem que se mostra mais promissora é a aplicação de técnicas de *Inteligência Artificial*.

A solução final proposta é um *Sistema Inteligente* no qual idéias presentes em três técnicas de *Inteligência Artificial* são combinadas: *Teoria Nebulosa*, *Redes Neurais Artificiais* e *Sistemas Multi-Agentes*. Cada uma possui características que serão utilizadas para solucionar partes do problema abordado. Através da combinação coordenada dessas técnicas, apresentamos um *Sistema Híbrido* que será detalhado no Capítulo 4.

1.6 Estrutura

No Capítulo 2 abordamos o problema mais profundamente, destacando o funcionamento global da rede, e de alguns de seus componentes, e esclarecemos como a alocação dinâmica pode ajudar na otimização do uso dos recursos disponíveis.

A seguir, no Capítulo 3, incluímos uma breve introdução à Inteligência Artificial e apresentação das três técnicas utilizadas no sistema proposto.

No Capítulo 4 apresentamos uma visão geral do sistema proposto e sua utilização para solucionar o problema. Cada módulo é detalhado, com a apresentação seu funcionamento interno e interações com outros componentes. Alguns cenários são analisados para esclarecer o funcionamento global do sistema.

O Capítulo 5 detalha a elaboração da simulação realizada para validação da solução proposta, enquanto no Capítulo 6 expomos os resultados obtidos.

O Capítulo 7, por sua vez, reapresenta os objetivos propostos e analisa os resultados alcançados, com destaque para as contribuições do trabalho para as áreas de rede e sistemas inteligentes. Por fim, são relacionados os tópicos para estudos futuros e as possíveis linhas de pesquisa que poderão ser seguidas para continuidade e evolução do assunto abordado.

2 DETALHAMENTO DO PROBLEMA

Em uma rede sem fio, se cada chamada fosse alocada em um canal diferente, o número de usuários simultâneos suportado seria igual ao número de canais disponíveis. Devido à escassez dos recursos de rádio, a capacidade das redes sem fio seria exageradamente pequena, tornando a prestação de serviços inviável.

Na prática, para aumentar a capacidade da rede, os mesmos canais são utilizados em diferentes células, desde que estejam distantes o suficiente para evitar, ou pelo menos minimizar, as interferências.

Diversas estratégias de alocação de canais foram propostas e, de modo geral, é possível classificá-las em duas categorias básicas: *Alocação Fixa de Canais* e *Alocação Dinâmica de Canais*. Na alocação fixa (OH; TCHA, 1992; ZHANG; YUM, 1991; ZHANG; YUM, 1989), durante a fase de projeto ou no início de operação da rede, um conjunto de canais é definido e alocado permanentemente em cada célula. Os canais podem ser reutilizados em outras células desde que respeitem a mínima *distância de reuso*. Esse tipo de alocação apresenta um bom desempenho em redes com demanda uniforme. A desvantagem aparece quando o tráfego não é uniforme e/ou quando pode ultrapassar a capacidade máxima da célula, resultando em bloqueio de chamadas.

Por outro lado, a alocação dinâmica (COX; REUDINK, 1973; OKADA; KUBOTA, 1992; BECK; PANZER, 1989; CHOY; SINGH, 1996a) distribui os canais nas células de acordo com a demanda dos usuários. As células não possuem canais próprios, mas requisitam os recursos quando necessário. Cada célula pode utilizar qualquer canal que

não viole as limitações de reuso. As técnicas de alocação dinâmica apresentam vantagens sob tráfego não uniforme e variável no tempo, mas aumentam a complexidade e custo de implementação. Existe ainda a alocação híbrida (HAC; CHEN, 2000), que nada mais é que a utilização das técnicas de alocação fixa e dinâmica simultaneamente.

Além da forma de alocação, pode-se dividir as técnicas em *centralizadas* e *descentralizadas*. No modelo centralizado ((DAS; SEN; JAYARAM, 1997; DAS; SEN; JAYARAM, 2000; ZHANG; DAS, 2000)), a definição dos canais é realizada por um único elemento, normalmente a central de comutação móvel. Essa abordagem apresenta algumas desvantagens, como sobrecarga do módulo central e ponto único de falha. No descentralizado ((CAO; SINGHAL, 1998; CAO; SINGHAL, 2000a; GARG; PAPATRIANTAFILOU; TSIGAS, 1996; PRAKASH; SHIVARATRI; SINGHAL, 1995)), as estações base definem a alocação e escolhem os canais que serão utilizados. Vantagens dessa abordagem são disponibilidade, confiabilidade e escalabilidade. Entretanto, o custo de implementação e a troca de mensagens entre as estações é maior.

Neste trabalho, o foco será a alocação dinâmica de canais de forma descentralizada, principalmente pelo grande potencial de desempenho que essa abordagem possui.

2.1 Redes sem Fio

Existem diversos tipos de redes sem fio em operação atualmente. Algumas, como as celulares, utilizam faixas de frequência reservadas e necessitam de autorização do governo para serem utilizadas. Já outras, como as redes *Wi-Fi*¹, utilizam frequências sem restrições de uso. Apesar das diferenças, como faixa de frequência e forma de codificação da informação, todas fazem uso de sinais eletromagnéticos para transmitir informação, o que significa que o meio recebe constantemente sinais em diversas

¹**Wireless Fidelity**: termo desenvolvido pela Wi-Fi Alliance para descrever produtos WLAN (Wireless Local Area Network) baseados no padrão IEEE 802.11 (WI-FI.ORG,).

freqüências provenientes de uma infinidade de dispositivos e de redes diferentes.

As freqüências de rádio utilizadas em uma rede sem fio são, normalmente, divididas em *canais de rádio*. Esses canais podem ser utilizados ao mesmo tempo em diferentes regiões, mas canais adjacentes utilizados em locais próximos podem causar interferências.

Os canais podem ser divididos de diferentes formas e três técnicas predominam atualmente: Divisão por Freqüência, Divisão por Tempo e Divisão por Código. Na Divisão por Freqüência, cada usuário utiliza uma freqüência diferente para a comunicação. Na Divisão por Tempo, a mesma freqüência transporta diversas comunicações e o tempo é dividido em intervalos (*timeslots*). Cada usuário recebe um *timeslot* para realizar sua comunicação. A Divisão por Código consiste na criação de diferentes canais através da modulação por códigos. Essas técnicas podem ser combinadas, resultando em mais métodos para divisão dos canais de rádio. Como exemplo, podemos citar o GSM, que utiliza divisão por freqüência e divisão por tempo. Entretanto, independente da técnica utilizada, a divisão não pode ser realizada indefinidamente, pois é necessário garantir um mínimo de qualidade para cada canal.

A região geográfica atendida por uma rede sem fio, a *área de cobertura*, é particionada em diversas regiões adjacentes denominadas *células*. Cada célula possui uma *estação base* que atende certo número de *usuários móveis*. Devido à dispersão do sinal de rádio por perda de potência ou interferências, a área de cobertura de cada estação base é limitada. Estações bases próximas são agrupadas e controladas por um *controlador de estações base*, que executa algumas funções centralizadas. Diversos controladores e estações base são atendidos por uma *central de comutação móvel*, que gerencia uma grande área geográfica e serve de interface para outras redes, sejam elas móveis ou fixas.

Figura 2.1: Exemplo de região coberta por células hexagonais.

2.1.1 Distribuição de Recursos

A cobertura de uma grande área geográfica é obtida através da divisão da região em células, cada uma atendida por uma estação base. Cada estação base possui uma ou mais antenas que cobrem uma área definida. Existem diversos tipos de antena, cada um com características próprias de cobertura. É comum encontrarmos estações com três antenas de 120° que cobrem uma região aproximadamente circular. A Figura 2.1 mostra uma região coberta por células hexagonais.

A quantidade de canais disponíveis é limitada – a faixa de frequências é estreita – e, normalmente, a região a ser coberta é enorme. Para que um número relativamente pequeno de canais atenda a área desejada, diferentes pontos de acesso precisam utilizar os mesmos canais, de modo a prover a rede com capacidade suficiente. Os canais devem ser distribuídos seguindo *regras de reutilização*, com o objetivo de minimizar interferências no sinal como, por exemplo, co-canal e canal adjacente².

As regras de reutilização criam limitações nas opções de configurações disponíveis. Por exemplo, tomemos uma região coberta por células, como indicado na Figura 2.2, sendo $c_1, c_2 \dots c_{10}$ os canais disponíveis, onde definimos uma regra de reutilização segundo a qual as células podem dispor de qualquer canal que não esteja em uso nas células vizinhas.

²Mais informações sobre interferências no sinal de rádio podem ser obtidas em Ericsson (1998)

Figura 2.2: Região coberta por 7 células hexagonais.

Se, no planejamento, identificarmos que a célula A possui necessidade de recursos superior às demais células, podemos, inicialmente, alocar os canais conforme apresentado na Tabela 2.1.

Célula	Canais
A	c_1, c_8, c_9
B	c_2
C	c_3
D	c_4
E	c_5
F	c_6
G	c_7

Tabela 2.1: Alocação inicial de canais.

Na distribuição sugerida, o canal c_{10} não é utilizado em nenhuma célula e, portanto, pode ser alocado em uma célula que necessitar de mais recursos. Ainda analisando a Tabela 2.1, outros canais podem ser alocados, contanto que a regra de reutilização definida seja respeitada. A Tabela 2.2 apresenta os canais que podem ser alocados em cada célula de acordo com a distribuição apresentada na Tabela 2.1. Denominamos *canais livres* os canais disponíveis que podem ser alocados em uma célula de acordo com as regras de uso.

Suponhamos que a demanda na célula A aumente. Para que os usuários presentes nessa célula sejam atendidos, precisaremos alocar mais canais. Como pode ser obser-

Célula	Opções
A	c_{10}
B	c_4, c_5, c_6, c_{10}
C	c_5, c_6, c_7, c_{10}
D	c_2, c_6, c_7, c_{10}
E	c_2, c_3, c_7, c_{10}
F	c_2, c_3, c_4, c_{10}
G	c_3, c_4, c_5, c_{10}

Tabela 2.2: Canais disponíveis para alocação.

vado na Tabela 2.2, a única opção de canal livre para a célula é c_{10} , e a distribuição resultante é apresentada na Tabela 2.3.

Célula	Canais em Uso	Canais Livres
A	c_1, c_8, c_9, c_{10}	—
B	c_2	c_4, c_5, c_6
C	c_3	c_5, c_6, c_7
D	c_4	c_2, c_6, c_7
E	c_5	c_2, c_3, c_7
F	c_6	c_2, c_3, c_4
G	c_7	c_3, c_4, c_5

Tabela 2.3: Alocação após expansão da célula A.

Seguindo o mesmo raciocínio, caso as demais células necessitem de mais recursos, poderemos adicionar até dois novos canais em cada uma. Uma distribuição possível é apresentada na Tabela 2.4 e, neste caso, não existem mais canais livres em nenhuma célula, ou seja, a disponibilidade de recursos nas células já atingiu o limite da rede.

Célula	Canais em Uso	Canais Livres
A	c_1, c_8, c_9, c_{10}	—
B	c_2, c_4, c_6	—
C	c_3, c_5, c_7	—
D	c_4, c_6, c_2	—
E	c_5, c_7, c_3	—
F	c_6, c_2, c_4	—
G	c_7, c_3, c_5	—

Tabela 2.4: Alocação após expansão das demais células.

No cenário representado na Tabela 2.4, para aumentar a oferta de recursos em qualquer uma das células será necessário analisar se existe alguma célula com canais

ociosos e verificar se um remanejamento é possível.

Para exemplificar, consideremos que a demanda nas células A e C diminua enquanto na célula B aumente. Para disponibilizarmos mais canais em B, precisamos analisar as células vizinhas³ (A, C e G) e verificar se existe a possibilidade de redução do número de canais disponibilizados. Se uma das células vizinhas permitir a redução, ainda é necessário verificar se ela possui canais que possam ser disponibilizados sem interferir nas demais células vizinhas.

Analisando a Figura 2.2 e a Tabela 2.4, identificamos que a célula B pode, inicialmente, utilizar os canais c_1 , c_3 , c_5 , c_7 , c_8 , c_9 e c_{10} , pois não estão alocadas em B. O próximo passo é verificar as regras de reutilização. Os c_3 , c_5 e c_7 não podem ser transferidos de C para B pois também estão alocados em G. Assim, restam c_1 , c_8 , c_9 e c_{10} , todos presentes na célula A. Nesse exemplo, podemos escolher qualquer canal dentre as opções existentes que o resultado será o mesmo. Transferimos, então, c_1 da célula A para a célula B, resultando na configuração apresentada na Tabela 2.5. A retirada do canal c_1 da célula A apresenta um outro efeito: c_1 aparece como canal livre para as células D, E e F.

Célula	Canais em Uso	Canais Livres
A	c_8, c_9, c_{10}	—
B	c_2, c_4, c_6, c_1	—
C	c_3, c_5, c_7	—
D	c_4, c_6, c_2	c_1
E	c_5, c_7, c_3	c_1
F	c_6, c_2, c_4	c_1
G	c_7, c_3, c_5	—

Tabela 2.5: Configuração após realocação de c_1 .

Uma outra forma de realizar a realocação de canais é manter cada célula apenas com os recursos necessários. Seguindo essa linha, a identificação de recursos ociosos em A e C implica liberação de canais. Retiramos, por exemplo, c_1 da célula A e c_3

³Uma célula Y é considerada *vizinha* de uma célula X se Y é adjacente a X, ou seja, as áreas de cobertura de ambas as células possuem intersecção ou estão muito próximas.

da célula C. Isso faz com que o canal c_1 fique disponível em todas as células, como se constata na Tabela 2.6. Dessa forma, o aumento de demanda em B pode ser facilmente solucionado pois existe um canal livre para ser alocado na célula.

Célula	Canais em Uso	Canais Livres
A	c_8, c_9, c_{10}	c_1
B	c_2, c_4, c_6	c_1
C	c_5, c_7	c_1
D	c_4, c_6, c_2	c_1
E	c_5, c_7, c_3	c_1
F	c_6, c_2, c_4	c_1
G	c_7, c_3, c_5	c_1

Tabela 2.6: Configuração após redução dos recursos nas células A e C.

2.1.1.1 Realocação Dinâmica

Observamos que, para otimizar o uso de recursos em uma rede celular, devemos ter a capacidade de redistribuir os canais existentes conforme a demanda, acompanhando as variações da rede e respondendo rapidamente às requisições dos usuários. Mesmo bastante simples, o exemplo apresentado na Seção 2.1.1 mostra que a alocação fixa de recursos pode resultar em congestionamento de células e sub-utilização de recursos.

Nas redes comerciais instaladas, o planejamento da distribuição de canais, apesar de ser configurado com parâmetros estáticos, é uma atividade complexa, pois a rede pode conter células com diferentes tamanhos, formatos e quantidades de canais. Além disso, as regras de reutilização podem ser mais restritivas e exigir que diversos aspectos sejam analisados antes que um canal seja configurado em uma estação rádio base.

Em uma rede real, a concentração de usuários e a demanda de tráfego⁴ variam rapidamente. É impossível otimizar a utilização de recursos em células que possuem configurações estáticas e que, conseqüentemente, não se adaptam às mudanças inerentes à computação móvel. Não é raro que os recursos disponíveis em algumas regiões

⁴O tráfego é definido por vários parâmetros, incluindo tipo de conexão, duração, taxa de transferência e limite máximo de atraso.

se esgotem enquanto outras apresentem níveis de utilização muito baixos.

A capacidade de realocação dinâmica de canais, contudo, é apenas o ponto de partida para a otimização. Para que esse dinamismo contribua para um melhor uso dos recursos existentes, as realocações devem acompanhar as variações de tráfego de forma a suportar a demanda atual e suas oscilações no curto prazo.

2.2 Trabalhos Relacionados

Existem diversos estudos que abordam a otimização de recursos em redes celulares através da realocação dinâmica de canais. A grande maioria apresenta algoritmos, centralizados e/ou descentralizados, que utilizam as informações disponíveis para ajustar as capacidades das células da rede, de forma a acompanhar a demanda de seus usuários.

No trabalho de Pattavina, Quadri e Trecordi (1999), um algoritmo de realocação dinâmica é proposto para melhorar a resposta de uma rede celular, tanto sob tráfego uniforme como não uniforme. Uma simulação é realizada em uma região coberta por 100 células hexagonais, e a rede possui 35 frequências com 4 canais (timeslots) cada. Os resultados são apresentados com base, principalmente, na probabilidade de bloqueio chamadas.

Shen, Mark e Ye (2000) apresentam um sistema adaptativo de inferência nebulosa para estimar e prever a probabilidade das informações de mobilidade de uma rede sem fio, com o objetivo de atingir um equilíbrio entre qualidade de serviço e utilização de recursos. O foco do trabalho é a mobilidade do usuário. O sistema proposto estima o trajeto do usuário, prevendo sua posição futura e adaptando as células envolvidas.

O trabalho desenvolvido por Zhang, Das e Jia (2004) propõe um algoritmo descentralizado para alocação dinâmica de canais, o D-CAT. Os resultados de simulações são comparados com outros algoritmos, destacando as diferenças e melhorias atingidas em

bloqueio de chamadas, bem como o número e o custo das mensagens trocadas durante o processo de realocação.

Baiocchi e Sestini (1996) apresentam algumas estratégias de alocação dinâmica. A rede de referência adotada emprega alocação fixa de canais. Os resultados obtidos em simulações são comparados utilizando as probabilidades de bloqueio de chamadas obtidas em cada técnica e na rede de referência. A análise dos resultados evidenciam o grande potencial da alocação dinâmica em redes sob tráfego não uniforme, em especial da estratégia *GDCA* (*Geometric Dynamic Channel Allocation*).

O algoritmo *DDRA* (*Distributed Dynamic Resource/channel Allocation*) é proposto por Boukerche, Hong e Jacob (2002). Semelhante aos demais trabalhos, simulações são realizadas para validação do modelo. Além disso, o tempo de resposta do sistema é analisado sob diferentes situações de carga da rede. Em outro trabalho, Boukerche, El-Khatib e Huang (2004) comparam o desempenho do *DDRA* com os algoritmos Prakash-Shivaratri-Singhal (PRAKASH; SHIVARATRI; SINGHAL, 1999), Choy-Singh (CHOY; SINGH, 1996b) e Cao-Singhal (CAO; SINGHAL, 2000b). Em sua conclusão, destacam o bom desempenho do *DDRA* aliado à baixa complexidade das mensagens.

2.2.1 Métrica

Uma das métricas mais comuns presentes nos trabalhos analisados é a *taxa de bloqueio*. Através desse parâmetro é possível analisar a eficiência da rede do ponto de vista dos usuários, verificando a quantidade de pedidos de conexão que foram rejeitados pela rede móvel.

Devido à facilidade de obtenção e análise e à grande importância desse parâmetro no funcionamento da rede, a simulação realizada no Capítulo 5 utiliza a taxa de bloqueio como parâmetro de análise e, no Capítulo 6, o desempenho do sistema é avaliado com base nessa métrica.

3 INTELIGÊNCIA ARTIFICIAL

Inteligência Artificial é o ramo que estuda o conjunto de paradigmas que procura obter um comportamento inteligente a partir de implementações artificiais, normalmente com a utilização de sistemas computacionais. Ainda é polêmico o que pode ser considerado um sistema inteligente, em grande parte pela grande dificuldade em definir o que é inteligência. Apesar dessa discussão, é possível identificar um sistema inteligente de forma indireta. Por exemplo, um programa de computador é considerado inteligente se realiza tarefas que seriam consideradas inteligentes se fossem feitas por um humano.

Deve-se evitar a confusão entre sistemas complexos e sistemas inteligentes. Uma máquina industrial que realiza tarefas difíceis, movimentos complexos e síncronos e possui sensores e atuadores sensíveis e precisos não deve ser considerada inteligente. Nesse exemplo, o equipamento apenas segue uma sequência de operações fixas e previamente configuradas, sem possuir capacidade para aprender ou se adaptar à novas situações. O conceito de inteligência está fortemente ligado aos conceitos de aprendizado, adaptação e compreensão.

A idéia básica dos sistemas inteligentes é a reprodução da capacidade de auto-organização e de aprendizado do cérebro como, por exemplo, a emulação da fisiologia realizada por *Redes Neurais Artificiais* ou a emulação da psicologia presente na *Teoria Nebulosa*.

Na área de IA, é comum encontrarmos menção a Teoria Clássica e Sistemas Clássi-

cos. Esses termos são utilizados para identificar conceitos, idéias e técnicas anteriores às técnicas de inteligência artificial. Dentro dessa definição, *Teoria de Conjuntos e Funções* e *Álgebra de Boole* pertencem à Teoria Clássica.

Uma aplicação na qual técnicas de inteligência artificial apresentam resultados superiores aos dos sistemas clássicos é o reconhecimento de padrões. Nossa capacidade de percepção pode ser observada através do exemplo do Quadro de Kanizsa, apresentado na Figura 3.1. Nessa imagem, a informação fornecida é completada e é possível visualizar um quadrado branco¹. O quadrado formalmente não existe e provavelmente não seria encontrado por um programa de computador, já que não existe um elemento de quatro lados na figura. Em engenharia, a existência de padrões incompletos é comum e muitas vezes é essencial que a informação original seja recuperada. Redes Neurais Artificiais podem ser treinadas para a identificação de padrões incompletos.

Figura 3.1: Quadro de Kanizsa.

Outro exemplo aparece quando um problema envolve informações desconexas, conflitantes ou até mesmo paradoxais. Na teoria clássica não existe solução para o paradoxo. Na Teoria Nebulosa os paradoxos são reduzidos a "*meias verdades*" ou "*meias mentiras*" através de uma lógica multi-valorada. O verdadeiro (1) e o falso (0)

¹Algumas pessoas têm, ainda, a impressão de que o quadrado branco identificado é mais claro que o restante da superfície.

são substituídos por *graus de pertinência* que podem assumir qualquer valor entre 0 e 1. Dentro desses limites, o valor 0,5 pode descrever um paradoxo.

Além de tentarmos reproduzir os aspectos fisiológicos e psicológicos do cérebro humano, podemos ainda reproduzir o comportamento social, ou seja, o comportamento de indivíduos dentro de um grupo. Essa é a idéia de *Sistemas Multi-Agentes*. Nessa técnica, diversos indivíduos com características e comportamentos simples interagem entre si e com o meio ambiente, fazendo emergir um comportamento inteligente. Um agente nada mais é que uma entidade (software e/ou hardware) que possui uma lógica simples e é, em princípio, incapaz de realizar tarefas complexas. Por outro lado, o Sistema Multi-Agente como um todo, apesar de ser formado por indivíduos limitados, pode realizar tarefas de grande complexidade, utilizando os diversos agentes disponíveis para analisar, reagir e interagir com o ambiente.

3.1 Teoria Nebulosa

A *Teoria Nebulosa*² foi apresentada em 1964 por Lotfi A. Zadeh, professor da Universidade da Califórnia, quando trabalhava com problemas de classificações de conjuntos que não possuíam fronteiras bem definidas, ou seja, a transição entre um conjunto e outro ocorria de forma suave e não de forma abrupta. Outros estudos já haviam surgido, como a lógica de três valores proposta por Lukasiewicz, que definia os valores verdadeiro (1), falso (0) e possível (1/2) (REZNIK, 1997).

Na teoria clássica, normalmente não existe dificuldade na classificação de elementos como pertencentes ou não a um determinado conjunto. Podemos afirmar com convicção que o número 50 pertence ao conjunto dos números inteiros e que 50,5 não pertence. Esse é um exemplo no qual não existe dúvida e a lógica booleana pode ser aplicada.

²Embora seja comum a utilização dos termos *Lógica Nebulosa*, *Lógica Difusa* ou *Lógica Fuzzy* (do inglês *Fuzzy Logic*), aqui utilizaremos o termo *Teoria Nebulosa*.

Há, porém, casos onde não é possível obter uma resposta única e objetiva. Existe dificuldade em afirmar se 55 pertence ou não aos números *aproximadamente iguais a 50*. A afirmação dependerá do contexto, algumas vezes sendo mais adequado considerar que 55 pertence, e outras que não pertence.

O universo de situações onde a relação de pertinência não é bem definida é bem vasto. Zadeh, com a teoria nebulosa, trouxe flexibilidade à pertinência de elementos aos conjuntos, criando os conceitos de *conjunto nebuloso* e *grau de pertinência*. A idéia foi publicada em 1965 (ZADEH, 1965; ZADEH, 1973), que é considerado o ano de nascimento da Teoria de Conjuntos Nebulosos.

O nome nebuloso (do inglês *fuzzy*) é utilizado para transmitir a idéia de imprecisão, de falta de conhecimento completo do sistema ou valor analisado. Seguindo o exemplo utilizado, como devemos criar o conjunto $\mathbb{A}$ dos *números aproximadamente iguais a 50*? Os números 45 e 60 pertenceriam a esse conjunto? E quanto aos números 20 e 100? Como já destacado, a resposta depende do contexto. A proposta é trabalhar com graus de pertinência e não apenas considerar *pertence* ou *não pertence* como resposta.

Dessa forma, para nosso conjunto $\mathbb{A}$ dos *números aproximadamente iguais a 50*, no universo dos números naturais $\mathbb{N}$ poderíamos, por exemplo, definir uma função de pertinência que retornasse 0 para o grau de pertinência do número 100 (equivalente a não pertinência da teoria clássica), onde 45 possuiria grau de pertinência 0,75 e 50 seria apresentado com grau de pertinência 1 (representando a pertinência total).

A extensão dos valores 0 e 1 para o intervalo $[0; 1]$ ³ originou a idéia de conjuntos nebulosos e permitiu a utilização de *variáveis linguísticas* no desenvolvimento de sistemas.

A figura 3.2(a) representa a função de pertinência discreta, utilizando o domínio dos números naturais $\mathbb{N}$ definido no exemplo. Podemos estender o conceito e defi-

³Os valores-limite 0 e 1 podem ser redefinidos e, na prática, quaisquer valores podem ser utilizados como extremos, embora os intervalos mais comuns sejam $[0; 1]$ e $[-1; 1]$.

nir a função de pertinência para outros domínios como, por exemplo, o conjunto dos números reais $\mathbb{R}$, apresentada na figura 3.2(b).

Figura 3.2: Função de pertinência do conjunto $\mathbb{A}$.

A grande contribuição da Teoria Nebulosa é visível em áreas onde é necessário lidar com a imprecisão, como em engenharia (YAGER; FILEV, 1994; KOSKO, 1997; SHAW; SIMÕES, 1999)) e química (FONSECA; KNAPP, 2000), e com subjetividade e desconhecimento, como ecologia (BARROS; BASSANEZI; TONELLI, 2000), economia (OH; KIM; LEE, 1990) e psicologia (AVERKIN; TARASOV, 1987). Nessas situações, a teoria tem mostrado sua capacidade em auxiliar a obtenção de modelos aderentes às necessidades dos profissionais.

3.1.1 Teoria de Conjuntos Nebulosos

Em um primeiro momento é comum confundirmos a *Teoria Nebulosa* com a *Teoria de Probabilidades* e entender o conceito de grau de pertinência como uma função de distribuição estatística. A confusão é justificável, já que existe uma estreita relação entre as duas teorias e, sob certos aspectos, a Teoria Nebulosa se apresenta de forma muito semelhante à de Probabilidades. Apesar da forte relação entre elas, é essencial que suas diferenças sejam compreendidas para que possamos utilizar todo seu potencial.

Tomemos um exemplo simples para evidenciar algumas diferenças entre as duas teorias. Considere uma caixa que contém diversas bolas brancas e pretas. Tendo em

vista que existem n_{br} bolas brancas e n_{pr} bolas pretas na caixa, podemos calcular facilmente qual a probabilidade de retirarmos uma bola branca. É uma situação onde o evento é bem definido: retirar uma bola branca. A incerteza existe até que o evento ocorra. Assim que o evento ocorre (uma bola é sorteada) a incerteza desaparece. Supondo, agora, que a caixa contenha bolas com diversos tons de cinza, variando do branco ao preto. Nessa situação não podemos apenas buscar a probabilidade de sortear uma bola branca, pois existe dúvida na definição do que pode ou deve ser considerado uma bola branca. Nesse caso, o evento não é bem definido e há dúvida após a ocorrência do evento (sorteio de uma bola). É necessário definir qual é a pergunta e como proceder para respondê-la.

As teorias de probabilidade e possibilidade, como também a nebulosa, têm se mostrado muito mais complementares do que concorrentes. Existem também muitos trabalhos abordando probabilidades nebulosas (STEIN, 1985; PIASECKI, 1985; PIASECKI, 1986; BORDLEY, 1989).

A teoria de conjuntos nebulosos vem sendo estendida a diversas abordagens, e o termo *Lógica Fuzzy* possui dois sentidos distintos: um sentido mais restrito, que se refere a um sistema lógico que flexibiliza a lógica clássica, e outro mais amplo, que engloba o sentido restrito, e abrange todas as teorias e tecnologias onde o conceito de conjuntos nebulosos é aplicado (processos de decisão, agrupamento, relações, modelos híbridos etc.).

Neste trabalho, a utilização da Teoria Nebulosa pode ser reduzida a três conceitos básicos:

- conjuntos nebulosos,
- variáveis linguísticas e
- regras de inferência.

3.1.1.1 Medições utilizando a Teoria Nebulosa

Sugeno (1974) apresentou um método para avaliação de incertezas, incorporando à medida a subjetividade de quem a faz (BARROS, 1992). Exemplos de aplicação são o valor de uma jóia, de uma obra de arte e da realização de um trabalho. Apesar da subjetividade inerente ao processo, a comparação de valores de mesma natureza é simples. Podemos facilmente concluir que uma mercadoria m_a com melhor qualidade e em maior quantidade que uma mercadoria m_b deve receber uma avaliação maior. Podemos representar a idéia na forma como $\mu(m_b) \leq \mu(m_a)$ onde μ pode ser ou não uma medida subjetiva.

3.1.1.2 Conjuntos Nebulosos

Conjuntos nebulosos foram criados para superar as limitações existentes nos conjuntos clássicos quando as transições entre conjuntos ocorrem de forma suave. As definições de conjuntos clássicos são generalizadas para a criação dos conjuntos nebulosos, e os conjuntos clássicos passam a figurar como um caso específico de conjuntos nebulosos.

Na teoria clássica, podemos definir a função característica como:

$$\mu_{\mathbb{A}}(x) = \begin{cases} 0 & \text{se e somente se } x \notin \mathbb{A} \\ 1 & \text{se e somente se } x \in \mathbb{A} \end{cases} \quad (3.1)$$

onde $\mathbb{A}$ é um subconjunto do conjunto Universo $\mathbb{U}$ e x é um elemento de $\mathbb{U}$.

A função característica divide o conjunto $\mathbb{U}$ em duas partições bem definidas: uma com elementos que pertencem ao conjunto $\mathbb{A}$ e outra com elementos que não pertencem. A teoria clássica apresenta as operações básicas de *união*, *intersecção* e *complemento*, que podem ser expressos com base na função característica:

$$\mu_{\mathbb{A} \cup \mathbb{B}}(x) = \text{MAX}(\mu_{\mathbb{A}}(x), \mu_{\mathbb{B}}(x)) \quad (3.2a)$$

$$\mu_{\mathbb{A} \cap \mathbb{B}}(x) = \text{MIN}(\mu_{\mathbb{A}}(x), \mu_{\mathbb{B}}(x)) \quad (3.2b)$$

$$\mu_{\bar{\mathbb{A}}}(x) = 1 - \mu_{\mathbb{A}}(x) \quad (3.2c)$$

Para obtermos as operações em conjuntos nebulosos, basta generalizar a função característica da teoria clássica para o intervalo $[0; 1]$, o que implica que a pertinência não estará limitada a seus limites 0 (não pertence) e 1 (pertence), podendo assumir valores entre esses dois estados. Dessa forma, um elemento x qualquer dentro do conjunto universo $\mathbb{U}$ pertencerá ao conjunto $\mathbb{A}$ com grau de pertinência $\mu_{\mathbb{A}}(x)$ que é um valor dentro do intervalo $[0; 1]$.

Como a pertinência não é mais um dado binário, não é possível definirmos um conjunto apenas pela sua lista de elementos. Conjuntos nebulosos são definidos por pares que indicam o elemento e seu grau de pertinência, conforme apresentado na equação 3.3.

$$\mathbb{A} = \{(x, \mu_{\mathbb{A}}(x)) \mid x \in \mathbb{U}\} \quad (3.3)$$

Conjuntos discretos podem ser representados conforme a equação 3.4, onde a somatória representa a operação de união (disjunção) e o termo $\mu_{\mathbb{A}}(x_i)/x_i$ indica que o elemento x_i pertence ao conjunto $\mathbb{A}$ com grau de pertinência $\mu_{\mathbb{A}}(x_i)$.

$$\mathbb{A} = \sum_i \mu_{\mathbb{A}}(x_i)/x_i \quad (3.4)$$

Normalmente, para simplificar a visualização, a lista dos elementos de conjuntos discretos apresenta apenas os elementos que possuem grau de pertinência diferente de zero.

Para exemplificar a notação utilizada na representação de elementos de conjuntos nebulosos, vejamos a figura 3.3, que apresenta graficamente a pertinência dos elementos ao conjunto $\mathbb{E}$. Observamos que o conjunto possui elementos com grau de pertinência diferente de zero entre os valores 1 e 10. O conjunto $\mathbb{E}$ pode, então, ser representado da seguinte forma:

$$\mathbb{E} = 1/0.2, 2/0.4, 3/0.6, 4/0.8, 5/1.0, 6/1.0, 7/0.8, 8/0.6, 9/0.4, 10/0.2$$

Figura 3.3: Representação Gráfico do conjunto Nebuloso $\mathbb{E}$.

Conjuntos contínuos são representados pela própria função de pertinência. As mais utilizadas em sistema nebulosos são as *lineares por partes* (triangular, trapezoidal), *quadrática* e *gaussiana*. Apesar de haver certo consenso na utilização das funções, a escolha apropriada da função de pertinência ainda é um problema em aberto e, muitas vezes, as funções são obtidas empiricamente.

As funções básicas presentes na teoria clássica de conjuntos também existem na teoria nebulosa, considerada a generalização aplicada à função característica. As operações de disjunção (*OU*) e conjunção (*E*) costumam utilizar as operações de máximo (*MAX*) e mínimo (*MIN*) respectivamente. Exemplos gráficos das operações de conjunção e disjunção são apresentados na figura 3.4.

Figura 3.4: Representação das operações de disjunção e conjunção.

Algumas definições são importantes no desenvolvimento de sistemas nebulosos, como os conceitos de *suporte* e α -*nível*. O *suporte* de um conjunto $\mathbb{A}$ é o conjunto clássico de elementos cujo grau de pertinência seja diferente de zero. O conjunto α -*nível* é o conjunto de elementos cujo grau de pertinência é maior que α . A equação 3.5 apresenta os dois conceitos matematicamente.

$$S(\mathbb{A}) = \{x \in \mathbb{U} | \mu_{\mathbb{A}}(x) > 0\} \quad (3.5a)$$

$$\mathbb{A}_{\alpha} = \{x \in \mathbb{U} | \mu_{\mathbb{A}}(x) > \alpha\} \quad (3.5b)$$

Também são definidos os conceitos de *cardinalidade* e *altura* de conjuntos nebulosos. Na teoria clássica, a cardinalidade é o número total de elementos do conjunto. Como, na teoria nebulosa, os elementos podem pertencer parcialmente ao conjunto, a cardinalidade é a soma dos graus de pertinência de cada elemento, ou, matematicamente:

$$Card(\mathbb{A}) = \sum_{x_i} \mu_{\mathbb{A}}(x_i) \quad (3.6)$$

Dessa forma, a cardinalidade do conjunto $\mathbb{E}$ definido anteriormente é dada por:

$$\text{Card}(\mathbb{E}) = 0.2 + 0.4 + 0.6 + 0.8 + 1.0 + 1.0 + 0.8 + 0.6 + 0.4 + 0.2 = 6.0$$

A altura de um conjunto nebuloso é o valor máximo de sua função de pertinência. Os conjuntos cuja altura é 1 são denominados normais e os de altura menor que 1 são os *subnormais*.

3.1.1.3 Variáveis Lingüísticas

As variáveis lingüísticas são utilizadas para expressar a pertinência a conjuntos nebulosos. O valor de uma variável lingüística é expresso qualitativamente através de um termo lingüístico e quantitativamente por uma função de pertinência. É caracterizada por $\{n; T; X; m(n)\}$, onde:

n : nome da variável. Identifica o que a variável representa dentro de um sistema nebuloso. Exemplos: temperatura, peso, altura, distância.

T : Conjunto de termos lingüísticos que a variável pode assumir. Exemplos: {frio, morno, quente}, {leve, pesado}, {baixo, médio, alto}.

X : Domínio de valores que determina o significado termo lingüístico. Exemplos: quente pode estar entre 50 e 70° C, alto pode ser acima de 1,70m.

$m(n)$: Função semântica que mapeia o significado de cada elemento t do conjunto T no espaço de conjuntos nebulosos (X).

A figura 3.5 representa uma variável lingüística de nome *altura*, utilizada para definir a estatura de uma pessoa. Os termos do conjunto T são: *baixa*, *média* e *alta*. O domínio corresponde, teoricamente, ao intervalo $[0; +\infty[$, apesar de, na prática, seu limite superior ser pouco mais de 2 metros. Cada termo t possui um conjunto nebuloso ($m(t)$) que o caracteriza.

Figura 3.5: Exemplo de Variável Linguística.

A variável linguística é expressa através de uma variável básica que possui o valor da medida. No exemplo, o valor da altura é medido em metros. O caráter qualitativo da medida é fornecido pelas variáveis linguísticas, que expressam conceitos presentes no cotidiano. Ainda utilizando o exemplo, é comum dizermos algo como "ela é *muito baixa*" e não expressões quantitativas como "ela tem 1 metro e 38 centímetros". As sentenças linguísticas possuem significado e contêm as informações necessárias para expressar seus fatores qualitativos.

Em sistemas nebulosos, a participação de um especialista no assunto em questão é essencial. É ele quem define o domínio de valores das variáveis linguísticas e o conjunto de termos necessários para representação qualitativa dos valores.

3.1.1.4 Regras de Inferência

As regras de inferência nebulosa são amplamente utilizadas e podem ser estendidas de diversas formas. Normalmente representam situações cuja inferência nos leva a um resultado desejado. Uma regra pode conter um conhecimento específico e um conjunto de regras pode representar todo um sistema. As regras são formadas por *antecedentes* *conseqüentes*:

SE <antecedentes> ENTÃO <conseqüentes>

Os *antecedentes* descrevem as condições necessárias e os *conseqüentes* representam o resultado ou ações que podem ser executadas quando os antecedentes são verificados. Diferente das regras na teoria clássica, uma regra nebulosa pode ser avaliada mesmo se os antecedentes não forem completamente satisfeitos.

A definição das regras de um sistema é feita com conhecimentos específicos e muitas vezes empírico sobre o problema que se deseja resolver. Especialistas traduzem o funcionamento do sistema em regras nebulosas simples que são utilizadas para representar a dinâmica do sistema.

As regras são avaliadas por uma *máquina de inferência*. Um método bastante comum é o de Mandani (MANDANI, 1974; MANDANI; ASSILLAN, 1975; MANDANI, 1977). As regras são processadas em paralelo e todas as saídas são obtidas simultaneamente e combinadas em uma variável lingüística de acordo com um padrão pré-definido.

3.1.1.5 Conversão de Variáveis Lingüísticas

Diversas aplicações necessitam de valores numéricos como resposta de um sistema. Na teoria nebulosa, os parâmetros de entrada do sistema são mapeados em variáveis lingüísticas que são utilizadas na avaliação de regras para geração de variáveis de saída, que também são variáveis lingüísticas.

A *defuzificação* (do inglês, *defuzzification*) consiste na interpretação quantitativa de uma variável para exprimir seu valor na forma de um número para utilização na teoria clássica.

O processo utilizado para essa conversão depende das características do sistema em questão, ou seja, não existe um processo genérico que pode ser aplicado para todo e qualquer sistema nebuloso. Os mais utilizados são (KLIR; YUAN, 1995):

Média dos Máximos Consiste na média dos valores de saída com maior grau de pertinência.

Centro de Área Também conhecido como *Centro de Gravidade*, é a técnica mais utilizada. Considera toda a distribuição da variável no cálculo, semelhante ao cálculo de centro de gravidade em física, e pode ser interpretado como uma média ponderada.

Método das Alturas Os elementos que serão utilizados no cálculo são defuzzificados e o valor clássico é calculado com baixo custo computacional. Pode ser considerado uma aproximação do método do centro de área.

3.2 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA's) é uma linha de pesquisa com grande potencial tecnológico. Sua proposta é a construção de modelos cognitivos artificiais inspirados nos modelos naturais de redes neurais.

Há centenas de anos o homem busca o conhecimento sobre o cérebro humano, considerado umas das estruturas mais complexas conhecidas. Hoje, diversos tipos de neurônios são conhecidos, com formas, tamanhos e funções variadas, mas todos possuem a mesma estrutura básica: uma parte central (*corpo celular*) com milhares de ramificações (*dendritos*) que são receptores de sinais. Além dos dendritos, do corpo celular parte uma fibra tubular (*axônio*) que é responsável pela transmissão de sinais. O cérebro pode ser entendido, de forma simplificada, como uma vasta rede altamente interconectada de células relativamente simples.

A estrutura do cérebro serviu de inspiração para grande parte dos conceitos existentes em RNA, que são implementadas como um conjunto, muitas vezes numeroso, de entidades simples denominadas *nodos* ou *neurônios*. Espera-se que, com essa estrutura, seja possível obter capacidades cerebrais, como executar operações paralelas, adaptar-se a novas circunstâncias e tratar informações com ruído.

O processo de aprendizado do cérebro está relacionado às mudanças nas conexões

entre os neurônios (*sinapses*) (HEBB, 1949). A quantidade estimada de neurônios no cérebro está em torno de 10^{10} e 10^{11} e cada neurônio se liga a aproximadamente $10^3 - 10^4$ outros neurônios, o que implica que o cérebro possui algo entre 10^{13} e 10^{15} conexões sinápticas. As evidências obtidas no estudo do cérebro nos levam a seguinte conclusão: *uma rede neural pode ser ensinada*. Essa é a base fundamental do estudo de RNA's.

3.2.1 Redes Neurais em Camadas (RNC)

As Redes Neurais em Camadas seguem uma das arquiteturas mais conhecidas e utilizadas. Nesse modelo, os neurônios estão organizados em camadas e se comunicam apenas com neurônios de camadas diferentes. Essas redes também são conhecidas como redes não recorrentes ou *feedforward*. A informação segue em apenas um sentido e não existe retroalimentação, ou seja, a saída de neurônios da camada i é utilizada como entrada de neurônios da camada $i + 1$. O representante mais simples desta arquitetura é o *perceptron*, apresentado na figura 3.6. Embora simples, o perceptron apresenta comportamentos interessantes e é amplamente utilizado em pesquisas (SEUNG; SOMPOLINSKY; TISHBY, 1992; WATKIN; RAU; BIEHL, 1993).

Figura 3.6: Perceptron simples.

Ampliando o conceito do perceptron, surgem as *redes multicamadas* (figura 3.7) que apresentam, além das camadas de entrada e saída, uma ou mais camadas escondidas. A quantidade de camadas escondidas e o número de neurônios em cada camada

estão relacionados com as representações internas da rede, que determinam a complexidade do mapeamento e a quais classes de problemas podem ser aplicadas.

Figura 3.7: Rede Multicamada.

Dado um neurônio, cada entrada pode apresentar contribuições para seu estado final. Tomando a representação da figura 3.6, os sinais recebidos pelo neurônio de saída podem ter *pesos* diferentes na formação da saída, dependendo da conexão que transmitiu o sinal. Assim, a saída de um neurônio é obtida através, mas não apenas, dos sinais de entrada, como também da importância de cada sinal para o neurônio em questão. Os *vetores sinápticos* $J_{m,k}$ representam os pesos das conexões, onde m representa a m -ésima camada escondida e k representa o k -ésimo neurônio dessa camada. O estado dos neurônios da camada m são representados por S_m .

Em uma unidade escondida, a relação de entradas e pesos é definida pelo conceito

de campo local:

$$h_{m,k}^{\mu} \equiv \frac{S_m^{\mu} \cdot J_{m,k}^{\mu}}{J_{m,k}^{\mu}} \quad (3.7)$$

Na equação, μ indica o μ -ésimo exemplo de um conjunto qualquer de exemplos.

Para uma entrada S_m^{μ} , a rede neural associa uma saída Σ_m^{μ} , função dos vetores S_m^{μ} e $J_{m,k}^{\mu}$, que pode ser escrita como:

$$\Sigma_J^{\mu} = F(h_{m,k}^{\mu}) \quad (3.8)$$

A função ganho ou função de transferência (F) pode ser uma sigmóide, uma função linear ou uma função sinal.

A máquina de paridade exemplifica os conceitos e equações apresentados. Ela possui uma camada escondida e, portanto, necessita de dois vetores sinápticos: um define as conexões entre a camada de entrada e a camada escondida e o outro define as conexões entre a camada escondida e a camada de saída. Outra característica é que os pesos entre a camada escondida e a de saída são fixos e iguais a 1. A saída da máquina de paridade é dada pelo sinal do produto dos sinais de saída dos neurônios da camada escondida, ou seja:

$$\Sigma_J^{\mu} = \text{signal} \left(\prod_{k=1}^K \sigma_k^{\mu} \right) \quad (3.9)$$

Onde $\sigma_{m,k}^{\mu} = \text{signal} \left(h_{m,k}^{\mu} \right)$, com $h_{m,k}^{\mu}$ definido em 3.7.

Assim, a saída será positiva sempre que $\sigma_{m,k}^{\mu} = -1$ para um número par de neurônios e negativa sempre que $\sigma_{m,k}^{\mu} = -1$ para um número ímpar.

As RNC's podem ser construídas de duas formas distintas. Na arquitetura *sobreposta* (do inglês, *overlapping*), apresentada na figura 3.8(a), cada neurônio da camada i

está conectado a todos os neurônios da camada $i + 1$. Isso implica que todos os neurônios da camada $i + 1$ recebem os mesmos sinais, diferenciando apenas os pesos das sinapses. Por outro lado, a arquitetura *não sobreposta*⁴ (*non-overlapping*), mostrada na figura 3.8(b), apresenta uma estrutura mais simples, onde cada neurônio da camada i se conecta a um subconjunto da camada $i + 1$. Em uma rede não overlapping com K ramos é interessante dividir o vetor de entrada S^μ em k vetores S_k^μ , com $k = 1 \dots K$, assim como o vetor sináptico J .

Figura 3.8: Arquiteturas RNC.

3.2.2 Aprendizado

O aprendizado é a capacidade de produzir um comportamento diferente a um estímulo, devido às informações recebidas no passado, e pode ser entendido como aquisição de conhecimento. Em inteligência artificial é comum encontrarmos menção a aprendizado de máquinas e a capacidade de aprender pode ser considerada uma característica fundamental de um comportamento inteligente.

O aprendizado é alcançado por memorização, exemplificação, analogia, exploração e/ou descoberta. RNA's possuem a capacidade de aprender a partir de exemplos (exemplificação) e de realizar interpolações (analogia) do que foi aprendido e armazenado nas conexões sinápticas (memorização).

⁴A arquitetura *não sobreposta* também é conhecida como *estrutura de árvore*.

Uma das formas de classificar o processo de aprendizagem é com relação à presença ou não de retroalimentação, constituindo o aprendizado *supervisionado* e *não supervisionado*, respectivamente.

Aprendizado Supervisionado O processo apresenta diversos exemplos à rede e verifica as saídas obtidas. Para cada exemplo inserido, as saídas são comparadas com os resultados esperados ou desejados. O *supervisor* ou *professor* indica se a resposta da rede foi boa ou ruim. Caso a saída fornecida não seja satisfatória, os pesos das conexões sinápticas são ajustados e um novo exemplo é apresentado. Esse processo é repetido até que a rede forneça saídas compatíveis com seu objetivo. A correção dos pesos das sinapses é realizado por um algoritmo que utiliza a diferença entre a resposta da rede e o resultado esperado (erro).

Aprendizado Não Supervisionado O processo de aprendizado não tem conhecimento das respostas esperadas e não possui nenhuma informação externa para reavaliar os valores associados às conexões. O objetivo da rede é apresentar saídas semelhantes para entradas semelhantes. Devido a essa característica básica de funcionamento, esse processo também é conhecido como *descobridor de regularidades* ou *redes auto-organizadas*. É utilizado quando não existe necessidade de respostas específicas para as entradas fornecidas. A intenção é obter resultados semelhantes para entradas semelhantes.

Conclui-se, então, que o processo de aprendizado de uma rede neural é constituído basicamente por regular os pesos das conexões sinápticas (vetor J). Em aprendizado supervisionado, utiliza-se um *algoritmo de aprendizagem* que, dado um *vetor de exemplos de entrada* e um *vetor de respostas desejadas*, aplica as entradas de treinamento à rede, compara as respostas obtidas com as respostas esperadas e, a partir do erro detectado, redefine os pesos das conexões de acordo com regras pré-definidas.

3.2.2.1 Lei de Hebb

O mecanismo de facilitação das redes neurais naturais inspirou a *Lei de Hebb* (HEBB, 1949):

"A intensidade de uma conexão sináptica entre dois neurônios aumenta quando os dois neurônios estão excitados simultaneamente."

que pode ser estendida para:

Ajuste a intensidade da conexão entre dois neurônios de uma quantidade proporcional ao valor da ativação simultânea deles. Se, no entanto, um dos neurônios tentar excitar o outro e não conseguir, a conexão enfraquece.

Uma característica importante da Lei de Hebb é a *propriedade da localidade*, ou seja, apenas informações locais são utilizadas na alteração do peso de uma conexão. Isso confere coerência biológica ao algoritmo de aprendizagem.

Assim, a expressão da variação do peso de uma conexão é apresentada na equação 3.10, onde w_{ij} é o peso da conexão entre os neurônios i e j , Δw_{ij} é o acréscimo a ser inserido na conexão, η define a intensidade da conexão e é denominado *taxa de aprendizado*, x_i é um estado de ativação (saída) do neurônio i e o_j é a resposta do neurônio j .

$$\Delta w_{ij} = \eta x_i o_j \quad (3.10)$$

A expressão utilizada na Lei de Hebb é bastante simples e diversos outros métodos foram propostos como, por exemplo, Regra Delta, Retropropagação, Aprendizado Natural, Aprendizado Competitivo, Aprendizado Reforçado, Aprendizado Aleatório e Aprendizado Evolutivo.

A seguir, detalharemos um pouco mais a Regra Delta e a Retropropagação, que pode ser considerada uma generalização da Regra Delta.

3.2.2.2 Regra Delta

Analisando uma sinapse de uma rede neural natural, observa-se que:

- o valor de modificação da intensidade da sinapse (η) pode variar com o tempo, mesmo se considerarmos as mesmas excitações;
- a modificação da intensidade da sinapse (Δw_{ij}) pode depender do valor atual (w_{ij});
- a modificação da intensidade da sinapse pode depender de neurônios vizinhos.

Incorporando as considerações acima à equação 3.10, obtemos uma expressão mais completa, onde a variação do peso da conexão entre os neurônios i e j é uma função da excitação real do neurônio i , da excitação desejada do neurônio i , do peso atual da conexão entre os neurônios i e j , da saída do neurônio j e do tempo:

$$\Delta w_{ij} = \Phi(x_i, d_i, w_{ij}, o_j, t) \quad (3.11)$$

A Regra Delta é obtida a partir da expressão 3.11, fazendo a função Φ depender da diferença entre a excitação real e a desejada:

$$\Delta w_{ij} = \eta (d_i - x_i) o_j \quad (3.12)$$

3.2.2.3 Retropropagação

A retropropagação (do inglês, *backpropagation*) é utilizada em redes neurais com mais de duas camadas, isto é, pelo menos uma camada escondida existe (RUMELHART;

MCCLELLAND; GROUP, 1986). Nesse tipo de arquitetura, o treinamento da rede regula os pesos internos e a rede funciona como se houvesse uma representação interna para a solução do problema apresentado.

Na retropropagação, uma entrada é apresentada à rede e sua resposta é comparada com o resultado desejado, obtendo-se o *erro*. O gradiente do erro é calculado em relação aos pesos entre a camada escondida e a camada de saída, que são atualizados utilizando uma taxa pré-definida. Esse processo é repetido para o próximo nível, ou seja, os pesos calculados para as conexões de entrada da camada de saída são utilizados para obtenção do erro do peso das sinapses entre a camada de entrada e a camada escondida. O gradiente desse erro é calculado e os pesos são corrigidos. Assim, o erro é propagado em direção à camada de entrada, calculando novos pesos para as conexões. Um outro exemplo é apresentado à rede e o processo é repetido até que o erro obtido seja satisfatório para o problema em questão.

Essa técnica é uma das mais difundidas para treinamento e diversas implementações de redes neurais a utilizam em sua forma original, com cálculo de gradiente ou com alguma variação no cálculo do erro para melhorar o desempenho em casos específicos.

É importante destacar que a retropropagação não pode ser aplicada a qualquer rede direta. Devido à derivação, essa técnica somente pode ser implementada em redes cujos neurônios possuam funções de ativação deriváveis.

A retropropagação é bastante utilizada como solução geral na criação de redes neurais para problemas nos quais ainda não foi possível definir uma outra técnica mais específica. É uma regra simples e que fornece resultados satisfatórios para uma grande quantidade de aplicações, incluindo previsão. Por suas características, a retropropagação é utilizada para a realização de treinamento da rede neural presente no sistema proposto.

3.2.3 Implementação

As redes neurais artificiais podem ser implementadas como simulações em sistemas computacionais, em circuitos especializados ou diretamente em circuitos digitais ou analógicos.

A simulação de RNA's necessita de um programa ou ambiente de simulação que permita a representação da rede, com seus neurônios e sinapses, e a aplicação de um algoritmo de aprendizado. Após a criação da rede a ser simulada, o ambiente deve possibilitar a execução de testes, permitindo a entrada de dados, verificação de saídas e a ativação de funcionamento autônomo.

RNA's implementadas por circuitos são interessantes por não apresentarem as limitações de desempenho existentes em simulações computacionais. As pesquisas na área são impulsionadas pelos avanços na tecnologia VLSI⁵.

3.3 Sistemas Multi-Agentes

A *Inteligência Artificial Distribuída* (IAD) se preocupa com todos os aspectos relativos à computação distribuída dentro de sistemas de inteligência artificial (BOND; GASSER, 1988; FERBER, 1999; HUHNS, 1987; GASSER; HUHNS, 1989). A técnica de *Sistemas Multi-Agentes* (SMA ou, do inglês, MAS - Multi-Agent Systems) constitui uma área dentro de IAD.

A proposta de Sistemas Multi-Agentes é a construção de sistemas computacionais a partir de entidades autônomas (agentes) que interagem entre si e com o meio ambiente em que estão inseridos. Para tornar isso possível, mecanismos de interação e coordenação devem ser implementados. Cada agente pode possuir características e capacidades específicas, além de objetivos próprios, o que torna a interação essencial

⁵VLSI (sigla de *Very Large Scale Integration*) e permite a construção de circuitos integrados com elevado número de componentes internos, normalmente acima de 100.000 transistores.

para que um agente satisfaça seus objetivos.

Existem algumas dificuldades na construção de Sistemas Multi-Agentes. Do ponto de vista do sistema⁶, a criação de mecanismos de controle genéricos para a coordenação dos diversos agentes apresenta-se como um grande desafio. Na outra extremidade do modelo, um outro problema é a especificação do próprio agente, que envolve a representação do seu conhecimento sobre o meio e sobre os demais agentes, de seus objetivos e de como será possível atingí-los. A figura 3.9 apresenta um modelo geral apresentado por Wooldridge (1998).

Figura 3.9: Modelo geral de agente.

É importante esclarecer que o termo *agente* possui significados diferentes de acordo com a área e o assunto abordado. Como exemplo, podemos citar o domínio de redes de computadores, onde existe o conceito de agentes e gerentes para realizar o controle e administração de equipamentos e da rede como um todo. Nesse modelo, os agentes estão localizados nos equipamentos e são meros coletores e/ou atuadores controlados pelos gerentes que, normalmente, são software instalados em servidores de gerência.

Dentro do domínio de Sistemas Multi-Agentes ainda não existe um consenso sobre o que constitui um agente, apesar de algumas definições terem se destacado (WO-

⁶O sistema como um todo costuma ser referenciado como uma sociedade de agentes.

OLDRIDGE; JENNINGS, 1995; FRANKLIN; GRAESSER, 1997). Contudo, mesmo com o não estabelecimento de uma definição final, algumas características importantes de um agente podem ser apontadas.

Sendo o modelo SMA uma área da Inteligência Artificial Distribuída, entende-se que técnicas de IA são utilizadas na criação dos agentes e/ou do sistema. Seguindo esse caminho, um agente é uma entidade que deve possuir capacidade de:

percepção: identificação de alterações no ambiente, na sociedade e nos demais agentes;

comunicação: cooperação com outros agentes e coordenação;

ação: transformação do ambiente e interação com outros agentes na busca pela satisfação de seus objetivos;

representação: manutenção de uma base de conhecimento para armazenar dados sobre o ambiente, sobre os outros agentes e sobre ele mesmo;

motivação: existência de desejos ou objetivos que o agente deve satisfazer;

deliberação: capacidade de tomada de decisão para, dado um cenário, atuar sobre o ambiente orientado pela motivação.

raciocínio: utilização de técnicas de inteligência artificial para melhorar o desempenho, sobretudo na deliberação.

Os agentes em SMA podem ser divididos em dois grupos básicos: *agentes reativos* e *agentes cognitivos*. As características apresentadas anteriormente referem-se aos cognitivos.

A proposta dos sistemas que empregam apenas agentes reativos é obter um comportamento inteligente a partir da interação de um grande número de agentes simples.

Esse enfoque possui grande influência da *entomologia*⁷. Nesses sistemas, os agentes não armazenam representações sobre o meio ambiente, sobre outros agentes ou sobre suas próprias ações passadas. O comportamento de cada agente pode ser implementado como um autômato finito simples, com regras que mapeiam dados obtidos do ambiente diretamente em ações (estímulo $\rightarrow$ resposta).

Os sistemas cognitivos, por sua vez, normalmente empregam poucos agentes, que possuem uma complexidade considerável e apresentam as características mencionadas anteriormente. Uma característica importante de um agente cognitivo é sua capacidade de tomada de decisão. Por esse motivo, um sistema cognitivo é utilizado no modelo proposto neste trabalho.

Com relação à tomada de decisão, existem duas linhas distintas de estudo. Alguns pesquisadores utilizam a *Teoria dos Jogos* para a coordenação dos agentes. Essa linha utiliza técnicas para escolha de estratégia em jogos com dois ou mais jogadores e é comumente encontrada em Teoria da Decisão e em Economia. Outra vertente faz uso da *Lógica Matemática* e a coordenação pode ser alcançada através do raciocínio social (SICHTMAN, 1998; SICHTMAN; CONTE; GILBERT, 1998).

Analisando as idéias presentes no modelo de Sistemas Multi-Agentes, é fácil perceber a influência de diversas outras disciplinas que não a Ciência da Computação. O estudo desses sistemas recebe grande inspiração da psicologia, sociologia, entomologia, economia, entre outras. A contribuição da sociologia dentro da área de inteligência artificial trouxe um novo paradigma, mudando a forma monolítica tradicional dos sistemas de Inteligência Artificial. Os sistemas multi-agentes são criados em um ambiente onde os agentes atuam e interagem. Essa interação pode levar à formação de grupos simples, comunidades ou até sociedades de agentes. Analisando os aspectos internos dos agentes, também existem modelos que utilizam conceitos trazidos por outras ciên-

⁷Entomologia é a ciência que estuda os insetos. Os SMA reativos procuram reproduzir o comportamento observado em colônias de insetos onde, apesar de os indivíduos possuírem capacidades limitadas, a sociedade apresenta um comportamento bastante complexo.

cias. Um agente possui *estados mentais*, que lhe fornecem conhecimento e raciocínio na forma de crenças, desejos, intenções, compromissos, etc.

Outro ponto interessante é com relação à computação distribuída. Observa-se claramente a característica descentralizada de um sistema que faz uso do modelo multi-agente. Pela própria natureza do sistema e de seus agentes, verifica-se que não existe um controle central. Os agentes são entidades autônomas⁸ que realizam tarefas e interação na busca da satisfação de seus próprios objetivos.

Tendo em vista as características apresentadas, a implementação de sistemas multi-agentes apresenta grupos distintos de pesquisas: a representação, a comunicação e a organização. A representação diz respeito aos conhecimentos do agente sobre o ambiente, sobre os outros agentes e sobre ele mesmo. A comunicação estabelece regras para possibilitar a interação entre os agentes e sociedades. A organização estabelece a visão do sistema e se os agentes possuem ou não conhecimento da organização (LEMAÎTRE; EXCELENTE, 1998).

3.3.1 Modelo BDI

Uma das arquiteturas mais utilizadas na implementação de agentes é baseada em crenças, desejos e intenções (do inglês, *BDI - Beliefs, Desires and Intentions*). A fundamentação dessa concepção pode ser observada nos trabalhos de Dennett (1987) e Bratman (1987). A primeira implementação de um sistema baseado nessa arquitetura foi o IRMA (**I**ntelligent **R**esource-bounded **M**achine **A**rchitecture) (BRATMAN; ISRAEL; POLLACK, 1988), do qual outros sistemas derivaram. Além de implementações, surgiram também arquiteturas abstratas de agentes baseadas em BDI e a concepção de *Lógica BDI*. A estrutura básica do modelo é apresentada na figura 3.10.

As crenças representam o conhecimento do agente sobre o ambiente e sobre os demais agentes presentes. Uma crença não necessariamente é uma verdade, mas algo

⁸Uma entidade é considerada autônoma se age apenas por decisão própria.

Figura 3.10: Arquitetura BDI.

que o agente acredita ser verdadeiro. Os desejos representam o estado que o agente deseja atingir e pode envolver o ambiente e outros agentes. Teoricamente, um agente pode ter desejos contraditórios, ou seja, desejos que na prática são mutuamente exclusivos. O agrupamento de desejos compatíveis são referenciados como objetivos. Por fim, as intenções representam o comprometimento do agente na execução de ações para satisfação de determinado objetivo.

As crenças são atualizadas de acordo com as alterações de estado do ambiente percebidas pelo agente e pela interação com outros agentes. Como a visão de mundo do agente muda com o tempo, o conjunto de desejos possíveis de serem satisfeitos é dinâmico, ou seja, alterações nas crenças do agente podem tornar impossíveis desejos que estavam sendo buscados e, por outro lado, tornar possíveis desejos que não eram considerados. Essa mudança é refletida nas intenções do agente (respeitando os comprometimentos já firmados). A partir do conhecimento (crenças) e da motivação (desejos ativos) do agente, as seqüências de ações existentes são analisadas e algumas são selecionadas. Essa escolha deve verificar as tarefas já iniciadas ou comprometidas para evitar incoerências. O conjunto de intenções do agente é atualizado e uma ação é escolhida para ser executada.

3.3.2 Comunicação

Em muitos sistemas, a comunicação é essencial para a satisfação dos objetivos em um ambiente compartilhado. Em sociedades, as ações dos agentes são coordenadas, algumas vezes para cooperação, outras para competição, mas sempre possuem um papel importante. A comunicação é, também, uma maneira de um agente compartilhar seu estado com outros agentes.

Como apontado anteriormente, a área de Sistemas Multi-Agentes fundamenta-se em pesquisas provenientes de outras disciplinas, principalmente da área de Humanas. Da mesma forma, as pesquisas relativas à comunicação são inspiradas em trabalhos tradicionais envolvendo linguagem, especialmente a teoria dos *atos de fala*. Austin (1962)) apresentou o entendimento da comunicação como sendo uma ação. Nessa linha, sentenças podem ser entendidas como requisições, sugestões, respostas, ordens, etc. Esse tipo de sentença é denominada *performativa*.

3.3.2.1 Linguagens

Uma tecnologia bastante difundida para comunicação entre agentes é a ACL (**A**gent **C**ommunication **L**anguage) (GENESERETH; KETCHPEL, 1994; GENESERETH; SINGH, 1994; SINGH; GENESERETH; SYED, 1995). Esse padrão define o KIF (**K**nowledge **I**nterchange **F**ormat), formato que determina a codificação do conteúdo da mensagem (GENESERETH; FIKES; OUTROS, 1992), e a KQML (**K**nowledge **Q**uery and **M**anipulation **L**anguage), linguagem para formação de mensagens com contexto (FININ; FRITZSON; MCKAY, 1992; FININ et al., 1994a; FININ et al., 1994b; LABROU, 1996; LABROU; FININ, 1994; MAYFIELD; LABROU; FININ, 1995; MAYFIELD; LABROU; FININ, 1996). Uma mensagem ACL completa é formada por uma expressão KQML com argumentos escritos em KIF.

As mensagens KQML possuem objetivos bem definidos como, por exemplo, *tell*

(afirmação - para troca de crenças entre agentes) e *achieve* (solicitação - para envio de pedidos). Algumas organizações, como a FIPA (**F**oundation for **I**ntelligent **P**hysical **A**gents) e a *Agent Society*, promovem o estudo e desenvolvimento de linguagens de comunicação e impulsionam a utilização de KQML.

A adoção de padrões de comunicação possibilita a interação entre agentes diferentes, permitindo a comunicação entre entidades de diversos fornecedores, que podem ser implementadas com tecnologias distintas.

A comunicação é essencial para o sucesso do sistema proposto. Em uma rede sem fio pode ser composta por diferentes equipamentos, de diversos fornecedores e com capacidades específicas, determinando a criação de agentes distintos, de acordo com as características de cada elemento da rede. Sendo assim, um padrão de comunicação deve ser utilizado, de forma a permitir a troca de informações entre os agentes.

4 SISTEMA PROPOSTO

Conforme apresentado no Capítulo 1, o objetivo deste trabalho é obter um sistema distribuído para a realocação dinâmica de canais de rádio em uma rede sem fio. O gerenciamento completo de uma rede constitui uma tarefa complexa, que envolve diversas atividades e grande variedade de detalhes. O sistema aqui apresentado foi projetado especificamente para o problema em questão, e sua atuação limita-se à gerência dos recursos de rádio. Nada obstante, o modelo definido pode ser estendido para tratar outras classes de problemas existentes em gerência de redes.

O modelo utiliza como base a arquitetura de agentes proveniente de *Sistemas Multi-Agentes*. Essa técnica fornece a infra-estrutura para o armazenamento de informações, tomada de decisão e comunicação do sistema. O modelo original foi simplificado, proporcionando aderência ao problema com baixa necessidade de processamento.

Essa abordagem é bastante atraente, pois as redes de comunicação são formadas por diversos equipamentos interconectados que realizam diversas tarefas, algumas isoladamente e outras em conjunto com os demais equipamentos, fornecendo o ambiente ideal para a inclusão de agentes inteligentes. Dessa forma, a arquitetura presente atualmente nas redes sem fio permite a inserção de novos módulos de software (agentes) para a implantação do sistema proposto em sistemas em produção, sem grandes complicações.

O modelo tem, ainda, a intenção de tornar o sistema altamente adaptável e flexível,

de forma a suportar novos serviços e tarefas a medida que evoluir. A limitação do escopo do problema neste trabalho (realocação dinâmica de canais) possibilita que uma simulação específica e bem controlada seja realizada e analisada de forma clara e objetiva, sem sofrer interferências de outros fatores e/ou problemas. O sistema permite a análise do modelo, detalhando seus principais módulos e destacando a participação das técnicas de inteligência artificial na resolução da questão.

4.1 Visão Geral

O modelo é formado por dois blocos distintos: um *Sistema de Controle* e um *Conjunto de Agentes Inteligentes*. A função do sistema de controle é configurar os agentes da rede, através da alteração de parâmetros e variáveis internas, bem como iniciar e preparar novos agentes que sejam instalados em antenas adicionadas à rede móvel. O *Sistema de Controle* pode ser facilmente incorporado aos sistemas de gerenciamento de rede em operação atualmente. Os agentes, presentes nas estações base, coletam dados da antena onde estão alocados (estação local) e se comunicam com os agentes vizinhos (estações base vizinhas), armazenam as informações relevantes e as utilizam para aprendizado e tomada de decisão.

A seguir, os dois blocos são detalhados e a utilização de cada técnica é abordada, destacando as principais colaborações para a solução do problema.

4.2 Agentes do Sistema

Conforme mencionado, os agentes foram inspirados nas idéias provenientes de Sistemas Multi-Agentes. Conforme apresentado na Seção 3.3, um dos modelos mais difundidos e utilizados em agentes cognitivos é o BDI (*Beliefs, Desires and Intentions*). A arquitetura original definida pelo modelo foi simplificada e adaptada para a obtenção de um agente com baixas necessidades de armazenamento e processamento. A

Figura 4.1 apresenta a arquitetura interna do agente e suas interações com os demais elementos da rede.

Figura 4.1: Estrutura interna do agente utilizado.

Como apresentado pela Figura 4.1, o agente possui cinco módulos: *Sensor*, *Comunicador*, *Base de Conhecimento*, *Decisor* e *Executor*. Cada um desses módulos é responsável por uma parcela limitada e bem definida do ciclo de funcionamento do agente.

Cabe destacar, ainda, que o agente proposto possui alto grau de modularidade e flexibilidade. Sua estrutura reduzida proporciona uma implementação que requer pouco poder de processamento e capacidade de armazenamento, o que permite que os agentes sejam incluídos diretamente nos equipamentos de rede, sem a necessidade de grandes investimentos em hardware. Além disso, a arquitetura interna possui módulos simples e interconectados, possibilitando a criação de blocos altamente especializados que podem ser direcionados ao problema em questão ou a diferentes classes de problemas.

Essas características podem ser observadas a seguir, na apresentação dos subsistemas.

4.2.1 Sensor

Os dados operacionais da estação base local (a antena onde o agente está instalado) são enviadas ao agente através de *mensagens de evento*. Para tanto, o software presente na estação deve possuir um módulo que implemente essa capacidade de comunicação.

A comunicação entre a antena e o agente é realizada pelo módulo **Sensor**. É ele quem recebe as mensagens provenientes da estação local, identifica os eventos ocorridos e envia as informações pertinentes ao módulo **Decisor**.

Para o problema em questão, existem apenas dois eventos possíveis: *conexão* e *desconexão*.

A antena envia uma mensagem de consulta sempre que recebe um pedido de conexão de um usuário da rede móvel. A mensagem apenas informa ao agente que existe um desejo de estabelecer uma conexão e não que uma conexão foi estabelecida pela estação base. Essa mensagem, que contém também a quantidade de canais desejada, é capturada pelo **Sensor** e o módulo **Decisor** é acionado para definir a ação a ser executada, ou seja, se o pedido de conexão será ou não aceito.

A mensagem de desconexão ocorre quando há a interrupção de uma conexão, seja por iniciativa do usuário ou devido a eventos ou problemas ocorridos na rede. A antena envia ao agente uma mensagem informando os canais que foram liberados. Essa mensagem é recebida pelo módulo **Sensor**, que aciona o **Decisor** para realizar as ações correspondentes.

4.2.2 Comunicador

O módulo **Comunicador** é responsável pela troca de mensagens entre o agente e seus vizinhos e com o sistema de controle.

O tratamento das mensagens de controle e de agentes vizinhos realizado por um

agente será apresentado nas Seções 4.2.4 e 4.2.5, onde os processos internos de decisão para geração de mensagens também serão esclarecidos.

4.2.2.1 Sistema de Controle

O **Comunicador** realiza toda a interação entre o agente e o sistema de controle. As mensagens enviadas pelo sistema de controle ao agente são recebidas por esse módulo, que as identifica e aciona o **Decisor** o qual executará a ação correspondente. Existem três tipos de *mensagens de controle*: mensagem de iniciação, mensagem de consulta e mensagem de atualização.

A mensagem de consulta enviada ao agente contém a lista de parâmetros que o sistema de controle deseja consultar. O agente recebe a mensagem, identifica as informações desejadas, consulta a **Base de Conhecimento** e envia uma resposta com os valores requisitados. Para realizar alterações nas informações armazenadas no agente, o controle envia mensagem de atualização, que contém uma lista de parâmetros e seus respectivos valores. O agente recebe a mensagem e realiza as modificações desejadas. A mensagem de iniciação é enviada pelo controle para informar o agente que os parâmetros foram configurados e ele pode iniciar sua operação. Essa mensagem é utilizada apenas uma vez e é responsável pela ativação do agente (e sua estação base correspondente) na rede móvel.

4.2.2.2 Agentes Vizinhos

A interação entre os agentes presentes na rede também é realizada através do módulo **Comunicador**. Os tipos de mensagens que podem ser trocadas são: mensagem de aviso, mensagem de alocação, mensagem de liberação, mensagem de solicitação e mensagem de colaboração.

A mensagem de aviso é gerada por um processo de iniciação. Esse processo é controlado pelo **Sistema de Controle** e consiste na criação do agente e na configuração

de seus parâmetros internos. Um agente que é iniciado envia essa mensagem a todos os seus vizinhos para informar que foi ativado. A mensagem contém, ainda, a lista de canais alocados no novo agente. Os agentes vizinhos que recebem essa mensagem atualizam a **Base de Conhecimento** para incluir o agente recém ativado e atualizar seus controles internos de canais disponíveis.

Quando um agente aloca canais livres para uso local, uma mensagem de alocação é enviada para todos os vizinhos, informando quais foram os canais alocados. De forma análoga, se um agente liberar canais que estavam alocados, ele envia uma mensagem de liberação para os agentes vizinhos. Os agentes que recebem mensagens de alocação ou mensagens de liberação identificam os canais enviados e atualizam suas **Base de Conhecimento**.

Caso um agente necessite de recursos de rádio e não existam canais livres disponíveis, ele pode decidir enviar uma mensagem de solicitação para seus vizinhos, informando sua necessidade. Os agentes vizinhos verificam a possibilidade de liberar canais e enviam mensagens de colaboração ao agente solicitante, contendo a relação de canais que podem ser liberados ou, caso não possam abrir mão de seus recursos, uma lista vazia. O agente solicitante recebe todas as mensagens de colaboração e, após escolher quais canais alocará, envia mensagens de alocação aos vizinhos. Os vizinhos que ofereceram canais verificam se alguma de suas ofertas foram aceitas e, caso positivo, liberam os canais correspondentes e enviam mensagens de liberação para que seus vizinhos possam atualizar suas informações. Os demais agentes apenas atualizam as **Bases de Conhecimento**.

Vale lembrar que, em todos os casos, o **Comunicador** é apenas responsável pela troca de mensagens. A tomada de decisão e a realização de tarefas são de responsabilidade dos módulos **Decisor** e **Executor**.

4.2.3 Base de Conhecimento

É o módulo responsável pelo armazenamento de parâmetros de configuração e dos dados sobre o agente, a estação local (antena), os agentes vizinhos, a rede neural de previsão, os conjuntos nebulosos, as variáveis lingüísticas e demais informações disponíveis para o agente.

A seguir, as principais informações da *Base de Conhecimento* são detalhadas. Além delas, existem outros parâmetros que serão mencionados no decorrer do texto, devido ao seu significado ser muito específico e necessitar de detalhes do modelo que ainda não foram abordados.

4.2.3.1 Vizinhos

A lista de vizinhos contém a relação das estações base cujas células são adjacentes à célula atendida pelo agente. Cada entrada da relação representa um agente vizinho e contém uma lista de canais que estão alocados na antena correspondente.

Por exemplo, considerando a região coberta por células hexagonais apresentada na Figura 4.2 e a distribuição de canais da Tabela 4.1, a relação de vizinhos do agente presente na célula B é apresentada na Tabela 4.2.

Célula	Canais Alocados
A	c_1
B	c_2, c_{14}
C	c_3, c_9
D	c_4, c_{10}
E	c_5, c_{11}
F	c_6, c_{12}
G	c_7, c_{13}

Tabela 4.1: Distribuição de canais nas células.

Figura 4.2: Região coberta por células hexagonais.

Agente Vizinho	Canais Alocados
A	c_1
C	c_3, c_9
G	c_7, c_{13}

Tabela 4.2: Relação de vizinhos do agente presente na célula B.

4.2.3.2 Canais da Rede

É a lista de canais disponíveis para alocação na rede sem fio. Contém todos os canais da rede que podem ser utilizados para comunicação. Além disso, a lista associa para cada canal da rede um *índice de separação*. Esse índice é utilizado pelo módulo **Decisor** para selecionar canais para alocação (Seção 4.2.4).

O índice de separação é um número que indica o grau de interferência de um canal com relação aos canais alocados nas estações local e vizinhas e é calculado de acordo com a Tabela 4.3.

Para exemplificar, consideremos uma rede móvel atendida por 20 canais, $c_0 - c_{19}$, cuja área geográfica é coberta por células hexagonais, conforme ilustrado pela Figura 4.2, e com a distribuição de canais apresentada na Tabela 4.1. O agente responsável pela Célula A possuiria os *índices de separação* para os canais da rede apresen-

Índice	Definição
0	canal alocado na estação local.
1	canal alocado em pelo menos uma das estações vizinhas.
2	canal adjacente a pelo menos um dos canais alocados na estação local. Nessa situação, diz-se que o canal analisado possui distância 1 com um canal alocado local localmente.
3	canal adjacente a pelo menos um dos canais alocados em pelo menos uma das estações vizinhas. Nessa situação, o canal analisado possui distância 1 com um canal de estação vizinha.
4	canal com distância 2 com pelo menos um dos canais alocados na estação local.
5	canal com distância 2 com pelo menos um dos canais alocados em pelo menos uma das estações vizinhas.
6	canal com distância 3 com pelo menos um dos canais alocados na estação local.
7	canal com distância 3 com pelo menos um dos canais alocados em pelo menos uma das estações vizinhas.
8	canal com distância 4 com pelo menos um dos canais alocados na estação local.
9	canal com distância 4 com pelo menos um dos canais alocados em pelo menos uma das estações vizinhas.
10	canal livre na vizinhança.

Tabela 4.3: Definição dos índices de separação.

tados na Tabela 4.4.

Pode-se observar, pelos valores presentes na Tabela 4.4, que o índice de separação indica o potencial de uso de um canal em determinada célula, calculado de acordo com o nível de interferência existente. Dessa forma, quanto maior for o índice de separação, menor será o impacto na vizinhança caso o canal em questão seja alocado para uso na estação base.

4.2.3.3 Canais Alocados

Armazena a relação dos canais configurados para utilização pela estação local, estejam eles em uso no momento ou não. Quando o módulo *Decisor* precisa escolher um canal para responder a um pedido de conexão, a primeira opção é verificar se existem canais ociosos nessa lista.

Canal	Índice	Canal	Índice
0	2	10	1
1	0	11	1
2	1	12	1
3	1	13	1
4	1	14	1
5	1	15	3
6	1	16	5
7	2	17	7
8	3	18	9
9	1	19	10

Tabela 4.4: Índices de separação dos canais da rede relativos à Célula A.

4.2.3.4 Canais em Uso

É a lista de canais ocupados na estação local. É um subconjunto da lista de Canais Alocados e apresenta os canais que estão suportando conexões de clientes na estação base.

4.2.3.5 Canais Oferecidos

Quando um agente envia uma mensagem de solicitação, os vizinhos que decidirem oferecer canais reservam os canais ofertados colocando-os na lista de Canais Oferecidos. Dessa forma, caso cheguem novos pedidos de conexão ou novas solicitações de canais, os canais oferecidos não serão considerados até que o agente que recebeu a oferta informe se irá ou não utilizá-los. Ou seja, os canais oferecidos ficam bloqueados até que o agente solicitante envie a mensagem de alocação informando quais dos canais oferecidos foram aceitos. Após a recepção dessa mensagem, os canais transferidos ao agente solicitante são liberados e a lista de Canais Oferecidos fica novamente disponível para uso.

4.2.3.6 Horário

O horário é armazenado em uma variável no domínio $[-1; 1]$. O horário, originalmente obtido em horas e minutos (hh:mm) é convertido em um número real h no intervalo $[0; 24[$ e armazenado como $h' = (h/12) - 1$.

Isso é necessário pois a variável é utilizada como entrada da rede neural de previsão. Maiores detalhes serão apresentados na Seção 4.2.3.14.

4.2.3.7 Utilização

A *utilização* é definida a partir da quantidade média de canais ocupados na estação local e é representada por conjuntos como os apresentados na Figura 4.3. Cada agente armazena a informação de utilização da estação local (a própria antena), que é utilizada como entrada da rede neural de previsão (Seção 4.2.3.14) e na avaliação de regras de inferência nebulosa realizada pelo *Decisor* (Seção 4.2.4).

O emprego de uma variável nebulosa permite uma transição suave entre os níveis *baixo*, *médio* e *alto* de *utilização* dos canais alocados. Os conjuntos nebulosos aqui definidos são apenas exemplos e, mesmo após a implantação do sistema em uma rede sem fio, podem ser alterados através do Sistema de Controle. Assim, os limites de cada conjunto podem ser modificados (por exemplo, a definição dos horários de manhã, tarde e noite), bem como a forma de transição (as regiões intermediárias *baixa—média* e *média—alta*). Além disso, cada agente pode ser configurado com conjuntos diferentes, de acordo com a região atendida pela antena onde o agente se encontra.

Como pode ser observado na Figura 4.3, foi considerada predominantemente *baixa* a utilização de até 10 canais, predominantemente *média* entre 10 e 22 e predominantemente *alta* acima de 22. Essa classificação pode não ser adequada para todas as antenas da rede e, portanto, durante a fase de planejamento, os níveis baixo, médio e alto devem ser definidos para cada a estação.

Figura 4.3: Conjuntos referentes à variável nebulosa *utilização*.

Além disso, a utilização também é armazenada na variável *util* no domínio $[-1; 1]$ para utilização pela rede neural de previsão. O valor de *util* é obtido a partir do valor medido e do valor máximo definido pelo conjunto *alta* de *utilização*:

$$util = MIN \left(1; 2 \cdot \frac{util_{med}}{util_{max}} - 1 \right) \quad (4.1)$$

Utilizando os conjuntos apresentados na Figura 4.3, temos:

$$util = MIN \left(1; 2 \cdot \frac{util_{med}}{32} - 1 \right) \quad (4.2)$$

Assim, situações com 6, 20 e 35 canais em uso apresentam, respectivamente, $util = -0,625$, $util = 0,25$ e $util = 1$.

4.2.3.8 Utilização Prevista

A *utilização prevista* usa os mesmos conjuntos da variável *utilização* (Figura 4.3) e é obtida a partir do resultado da rede neural de previsão. Ela é analisada no módulo **Decisor** na avaliação de regras de inferência nebulosa para a determinação da *situação* da antena, que será abordada na Seção 4.2.4.

A resposta da rede neural é apresentada no domínio $[-1; 1]$ e é convertida para um valor que indica a quantidade esperada de canais ocupados, seguindo a mesma idéia adotada na conversão da utilização, ou seja, o valor máximo definido pelos conjuntos nebulosos é utilizado na troca de domínio:

$$util_{prev} = util_{max} \cdot \frac{util_{rna} + 1}{2} \quad (4.3)$$

Na equação, $util_{prev}$ é a quantidade prevista de canais em uso, $util_{max}$ é o maior valor definido pelo conjunto nebuloso *alta* e $util_{rna}$ é a resposta da rede neural de previsão.

Após a conversão, a variável $util_{prev}$ é utilizada para a definição da variável nebulosa *utilização prevista*, utilizando os mesmos conjuntos definidos para a determinação da variável *utilização*.

4.2.3.9 Utilização Passada

A variável $util_{pas}$ armazena a $util_{prev}$ para treinamento da rede neural. Ela é empregada para determinar a diferença (erro) entre a previsão de utilização e o valor da utilização efetivamente alcançado. Maiores detalhes sobre esse processo serão abordados na Seção 4.2.3.14.

4.2.3.10 Ocupação

A variável *ocupação* representa, como o próprio nome indica, o grau médio de ocupação dos canais da antena e é definida pelos conjuntos apresentados na Figura 4.4. Ela é obtida a partir da porcentagem de ocupação calculada pela Equação 4.4, onde c_u e c_a são o número de canais em uso e alocados, respectivamente. Para exemplificar, se existirem 32 canais em uso de um total de 40 alocados, teremos $ocup = 80\%$, ou seja,

de acordo com os conjuntos da Figura 4.4, a *ocupação* será 0,8 *média* e 0,2 *alta*.

$$ocup = \frac{c_u}{c_a} \quad (4.4)$$

Podemos observar, pela definição dos conjuntos, que a ocupação média ocorre em 75%. Como será apresentado na Seção 4.2.4, os agentes tentarão manter a estação base na posição *média*, trabalhando próximos da capacidade máxima oferecida pelos canais alocados mas garantindo uma reserva para picos esporádicos.

É importante destacar que, apesar de as variáveis *utilização* e *ocupação* serem relacionadas com os canais em uso, seu significado é diferente e são empregadas em pontos distintos. A *ocupação* permite a identificação da situação da antena e depende diretamente da quantidade de canais alocados no agente. A *utilização*, por sua vez, fornece informações quantitativas sobre a utilização da estação que independem da quantidade de canais alocados.

Figura 4.4: Conjuntos referentes à variável nebulosa *ocupação*.

4.2.3.11 Situação

O módulo **Decisor** avalia um conjunto de regras de inferência nebulosa e obtém a variável *situação*. Essas regras utilizam as variáveis *ocupação*, *utilização* e *utilização*.

prevista para determinar o estado do agente, que pode ser: *liberação*, *captação* ou *neutro*. A definição da *situação* determina qual será o objetivo do agente, ou seja, qual será o conjunto de ações permitidas. Através da análise da vizinhança, o agente definirá qual será a ação a ser tomada. Mais detalhes sobre esse processo encontram-se na Seção 4.2.4.

4.2.3.12 Regras de Inferência

O conjunto de regras de inferência para determinar a *situação* do agente também fica armazenado na **Base de Conhecimento**. Isso torna possível a alteração ou ajuste das regras que determinam o estado da antena pelo **Sistema de Controle**.

As regras serão apresentadas na Seção 4.2.4

4.2.3.13 Passo de Previsão

O valor armazenado nessa variável representa um intervalo de tempo e é utilizado para definir o significado das variáveis *utilização* e *utilização prevista*. O *passo de previsão* pode ser alterado através do sistema de controle de acordo com o comportamento e precisão desejados.

Ele define o intervalo de tempo considerado no cálculo da utilização média de canais (u_m) e, também, o tempo no futuro com o qual a *utilização prevista* está relacionada. O parâmetro u_m indica a quantidade média de canais ocupados na célula durante o último passo de previsão. Para exemplificar, se o passo de previsão for definido como 30 minutos, u_m representa o número médio de canais ocupados na última meia hora. A variável nebulosa *utilização* é obtida diretamente através da fuzificação de u_m .

Da mesma forma, a variável *utilização prevista* indica o número médio esperado de canais em uso na célula para o próximo passo de previsão. Utilizando o mesmo passo do exemplo anterior, a variável representa a expectativa da quantidade média de canais ocupados na próxima meia hora.

4.2.3.14 Rede Neural de Previsão

Além de ser o módulo responsável pelo armazenamento de toda informação disponível para o agente, a **Base de Conhecimento** também abriga uma rede neural artificial para previsão da quantidade de canais em uso na célula no futuro.

Se o sistema for implantado em uma rede em operação, a rede neural será treinada, inicialmente, com a base histórica de utilização da rede. Os registros armazenados pelo próprio sistema de gerenciamento em uso na rede pode fornecer os dados necessários para esse aprendizado. Caso o sistema seja parte de uma nova rede móvel, isto é, não existem dados históricos para realizar o treinamento inicial da rede neural, a previsão não será utilizada pelo sistema até que um treinamento seja realizado. Para tanto, pode-se coletar dados pelo sistema de gerência e utilizá-los para o aprendizado ou pode-se deixar o uso da previsão desativado no módulo **Decisor**, mas ativa-se o treinamento. Na última opção, os agentes realizam previsões, mas elas não são analisadas na tomada de decisão. A resposta da rede apenas fornece $util_{prev}$ para ser comparada com o valor efetivamente atingido ($util_{med}$) e realizar o treinamento da rede.

Conforme apresentado na Seção 4.2.3.13, as previsões utilizam um *passo de previsão*, que define o intervalo de tempo considerado pelas variáveis utilização.

A Figura 4.5 apresenta a estrutura da rede neural. As variáveis horário (h no domínio $[-1; 1]$) e $util$ (domínio $[-1; 1]$) são utilizadas como entrada da rede e a variável $util_{prev}$ (domínio $[-1; 1]$) é obtida como resposta.

Cada variável de entrada é inserida na rede através de um neurônio. Os dois neurônios da camada de entrada possuem sinapses para todos os neurônios da camada escondida e estes, por sua vez, estão conectados ao neurônio da camada de saída.

A arquitetura da rede neural utilizada é conhecida como um aproximador universal de funções (HORNIK; STINCHCOMBE; WHITE, 1989). Todos os neurônios utilizam a tangente hiperbólica ($\tanh$) como função de transferência, cuja curva é apresentada na

Figura 4.5: Estrutura da Rede Neural Artificial de Previsão.

Figura 4.6.

Figura 4.6: Curva da tangente hiperbólica ($\tanh$).

Tomando como base redes móveis em produção, a curva que representa o número de canais em uso em uma célula normalmente varia principalmente de acordo com a hora do dia. A Figura 4.7 apresenta a quantidade de canais ocupados em uma célula durante uma semana (o detalhe de um dia é apresentado na Figura 4.8). Cada ponto representa a quantidade média de canais ocupados nos últimos 30 minutos. Observa-se que a quantidade de recursos em uso segue um padrão que pode ser previsto a partir do período do dia.

Dessa forma, a rede neural poderia receber apenas a variável h para realizar a previsão de utilização. Após o treinamento, ela aprenderia a distribuição da quantidade

Figura 4.7: Canais utilizados em uma célula no período de 7 dias.

de canais em uso na célula e poderia fornecer as previsões desejadas. Entretanto, a rede seria capaz de aprender apenas uma curva de utilização, em função do horário. A adição da variável *util* na entrada da rede permite que ela aprenda mais de um padrão para cada horário. Uma célula presente em uma região altamente comercial poderia ter um comportamento durante os dias úteis e outro diferente nos finais de semana. Teria também comportamentos distintos em datas especiais, como Natal ou Carnaval. A inserção de mais um parâmetro de entrada permite que a rede neural seja treinada com diversos comportamentos para o mesmo período.

A quantidade de curvas de comportamento diferentes que poderá ser aprendida pela rede neural de previsão está diretamente relacionada com a quantidade de neurônios na camada escondida, uma vez que serão as sinapses existentes que proporcionarão essa capacidade. Entretanto, pretende-se que o número de nós seja o menor possível, para evitar que padrões indesejados sejam aprendidos incorretamente, bem como para diminuir a necessidade de processamento da rede. Além disso, a rede não tem o objetivo de aprender e armazenar todos os padrões identificados na célula. Comportamentos antigos que não são reforçados devem ser descartados.

Figura 4.8: Canais utilizados em uma célula no período de 1 dia.

Na Seção 6, alguns cenários de treinamento são apresentados, bem como os resultados obtidos, e o funcionamento da rede neural de previsão é analisado mais detalhadamente.

Para finalizar, a resposta da rede neural precisa ser armazenada na variável nebulosa *utilização prevista*. A saída do neurônio da camada de saída é convertida, conforme apresentado na Seção 4.2.3.8, para obter a quantidade esperada de canais em uso, que é *fuzificada* de acordo com os conjuntos definidos (Figura 4.9).

É importante destacar, ainda, que a definição da variável *util*, apresentada na Seção 4.2.3.7, permite que a rede neural trabalhe com valores relativos de utilização, aprendendo a curva de comportamento sem levar em conta os valores absolutos envolvidos. Assim, caso a demanda em uma célula sofra alterações de amplitude, mas mantenha a mesma curva de comportamento, o agente pode ser reconfigurado apenas alterando-se os conjuntos nebulosos que definem a variável *utilização*, o que redefinirá o cálculo de *util*. Essa alteração torna os valores de *util* válidos para o comportamento já armazenado na rede neural, eliminando a necessidade de treinamentos adicionais

Figura 4.9: Conjuntos referentes à variável nebulosa *utilização prevista*.

para aprender o novo comportamento.

A previsão de utilização fornecida pela rede neural traz duas vantagens. Primeiro, permite que o agente antecipe uma situação de alta carga e prepare a estação para suportar a demanda. Outra contribuição é a eliminação de seqüências de liberação e alocação de canais em casos de oscilação. Sem a previsão, um agente pode identificar que possui recursos ociosos e decidir liberá-los. Instantes depois, devido a um aumento na demanda, esse mesmo agente pode não ter os recursos necessários para atender os usuários e resolve alocar mais canais. Com a previsão, o agente tem acesso a uma estimativa de uso futuro e pode manter canais ociosos alocados para suprir a demanda do próximo ciclo.

4.2.3.15 Passo de Treinamento

Após a implantação do sistema, a rede neural é treinada continuamente. As previsões realizadas são armazenadas e, no próximo passo de previsão, são comparadas com os valores de utilização realmente atingidos. A diferença entre os valores previstos e observados são utilizados para treinar a rede neural, permitindo a adaptação constante dos agentes. A frequência do aprendizado pode ser configurada através do *passo de*

treinamento. O ajuste da rede neural será realizado a cada n previsões, onde n é o valor inteiro definido como *passo de treinamento*.

Se o valor 1 for armazenado na variável, o treinamento será efetuado a cada passo de previsão, ou seja, todas as previsões são verificadas e realimentadas na rede neural.

Para exemplificar, se o passo de previsão for configurado como 30 minutos e o passo de treinamento for igual a 48, o treinamento seria realizado a cada 24 horas. Esse comportamento não é desejado, pois a rede sempre seria treinada no mesmo horário. Para contornar esse problema, poderíamos definir o passo de treinamento com o valor 49, o que garantiria que todos os horários seriam cobertos. O primeiro treinamento ($h = 0$) seria às 00h00min do dia 1, o segundo ($h = 49$) seria às 00h30min do dia 2, o terceiro ($h = 98$) seria à 01h00min do dia 3, o quarto ($h = 147$) seria à 01h30min do dia 4 e assim por diante, passando por todos os horários cobertos pelo passo de previsão. Esse tipo de cobertura também pode ser obtido com passos de treinamento menores (como, por exemplo, 25, 23, 13, 11, entre outros) ou maiores (como, por exemplo, 59, 61, 95, 97, entre outros).

Essa configuração permite controlar o aprendizado e evita que a rede neural seja submetida a um número muito grande de padrões de treinamento, o que pode comprometer a qualidade de suas previsões. Além disso, pode-se controlar a demanda por processamento no elemento de rede onde o agente está instalado.

4.2.4 Decisor

O módulo **Decisor** é responsável pela tomada de decisão do agente. Ele pode ser ativado pelo **Sensor**, que captura os eventos ocorridos na célula através da antena, ou pelo **Comunicador**, que recebe mensagens do sistema de controle e dos agentes presentes na vizinhança.

O **Decisor** controla, ainda, a utilização da rede neural de previsão, ou seja, é res-

ponsável pela realização das previsões e pelo treinamento da rede, utilizando o passo de previsão e o passo de treinamento configurados no agente e armazenados na **Base de Conhecimento**.

Assim, a execução de uma tomada de decisão pode ser ativada por:

- mensagens de conexão provenientes do *Sensor*;
- mensagens de desconexão provenientes do *Sensor*;
- mensagens de solicitação provenientes do *Comunicador*;
- mensagens de colaboração provenientes do *Comunicador*;
- nova previsão de utilização;

Esses eventos ativam a avaliação das regras de inferência nebulosas, que definem o estado do agente.

4.2.4.1 Tomada de Decisão

O primeiro passo para a tomada de decisão é identificar a situação da antena, ou seja, o estado em que o agente se encontra. Esse estado pode ser: *neutro*, *liberação* ou *captação*. A variável nebulosa *situação* é definida de acordo com os conjuntos apresentados na Figura 4.10. A determinação do seu valor é realizada através da avaliação de regras de inferência que utilizam as variáveis nebulosas *utilização*, *utilização prevista* e *ocupação*.

Sempre que as variáveis utilizadas na inferência nebulosa são alteradas, uma nova avaliação é efetuada para verificação do estado do agente e, conseqüentemente, determinação das ações que serão executadas.

Figura 4.10: Conjuntos referentes à variável nebulosa *situação*.

Neutro

Uma das metas do agente é manter a *situação* da antena no estado *neutro*. Assim, caso essa seja a situação, nenhuma ação é necessária.

Captação

Se a avaliação das regras resultar no estado de *captação*, significa que o agente necessita de canais adicionais para retornar ao estado *neutro*, que é sua meta. Primeiro, o **Decisor** verifica os *índices de separação* (Tabela 4.3) dos canais presentes na lista *Canais da Rede* armazenada na **Base de Conhecimento**. A preferência é dada para os canais com maior índice, respeitando um *valor mínimo para alocação*, que pode ser configurado através do **Sistema de Controle**. Abaixo desse valor mínimo, o canal não é considerado elegível para alocação na estação local.

Caso o **Decisor** não encontre canais livres suficientes para retornar o agente ao estado *neutro*, os agentes vizinhos serão consultados para verificarem a possibilidade de liberar recursos. Isso é realizado através do envio de mensagens de solicitação para os agentes da vizinhança e será detalhando posteriormente.

Liberação

No estado de *liberação*, o agente possui canais ociosos que podem ser liberados. O **Decisor** verifica as listas *Canais Alocados* e *Canais em Uso* para a seleção dos canais que serão desalocados. Através do módulo **Executor**, um comando de liberação contendo a lista de canais desalocados é enviado à antena e mensagens de liberação são enviadas aos agentes vizinhos.

A identificação da *situação* do agente é determinada através da avaliação de regras de inferência nebulosa. As Tabelas 4.5a, 4.5b e 4.5c apresentam os valores que a variável *situação* deveria assumir para cada combinação possível das variáveis *ocupação*, *utilização* e *utilização prevista*.

ocupação baixa					ocupação média				
		utilização					utilização		
		baixa	média	alta			baixa	média	alta
utilização prevista	baixa	lib	lib	lib	utilização prevista	baixa	ntr	ntr	lib
	média	ntr	lib	lib		média	ntr	ntr	ntr
	alta	ntr	ntr	lib		alta	cap	ntr	ntr

(a)

ocupação alta				
		utilização		
		baixa	média	alta
utilização prevista	baixa	cap	ntr	ntr
	média	cap	cap	ntr
	alta	cap	cap	cap

(c)

Abreviações: **lib**: liberação — **ntr**: neutro — **cap**: captação

Tabela 4.5: Definição da variável *situação* a partir das combinações das entradas.

A distribuição apresentada evita o aparecimento de seqüências de transições entre os estados de liberação e captação, caso ocorram oscilações nas variáveis de entrada. Se considerarmos que não há transições bruscas nas variáveis de entrada, ou seja, as variáveis não alternam de *baixa* para *alta* ou de *alta* para *baixa* em curtos espaços

de tempo, pode-se verificar, através da análise das tabelas apresentadas, que fixando qualquer uma das variáveis, a variação das demais altera a *situação* de forma suave, permitindo apenas transições **lib**→**ntr**, **ntr**→**lib**, **ntr**→**cap** e **cap**→**ntr**.

A obtenção da *situação* diretamente das Tabelas 4.5a, 4.5b e 4.5c tornaria o sistema rígido e algumas das características desejadas da Teoria Nebulosa seriam perdidas. Assim, as 27 relações definidas nas tabelas foram reduzidas, conforme apresentado a seguir:

Se *ocupação* é baixa e (*utilização* é alta ou *utilização prevista* é baixa) então *situação* é liberação
 Se *ocupação* é baixa e *utilização* é baixa e *utilização prevista* é alta então *situação* é neutro

Se *ocupação* é média e *utilização* é alta e *utilização prevista* é baixa então *situação* é liberação
 Se *ocupação* é média e (*utilização* é média ou *utilização prevista* é média) então *situação* é neutro
 Se *ocupação* é média e *utilização* é baixa e *utilização prevista* é alta então *situação* é captação

Se *ocupação* é alta e *utilização* é alta e *utilização prevista* é baixa então *situação* é neutro
 Se *ocupação* é alta e (*utilização* é baixa ou *utilização prevista* é alta) então *situação* é captação

O conjunto de regras é armazenado na **Base de Conhecimento** junto com as demais informações e configurações. Isso significa que cada agente possui suas próprias regras de inferência. O exemplo a seguir apresenta um outro conjunto de regras utilizado para definir a *situação*.

Se *ocupação* é baixa então *situação* é liberação
 Se *ocupação* é média então *situação* é neutro
 Se *ocupação* é alta então *situação* é captação

Se *utilização* é não alta e *utilização prevista* é alta então *situação* é captação
 Se *utilização* é não baixa e *utilização prevista* é baixa então *situação* é liberação

A utilização de conjuntos diferentes nos diversos agentes da rede permite que comportamentos distintos sejam configurados, de acordo com a estratégia de prestação de

serviço do controlador da rede móvel.

4.2.4.2 Recepção de Eventos

O **Decisor** também é responsável pelo tratamento dos eventos captador pelo módulo **Sensor**. Conforme visto na Seção 4.2.1, o **Sensor** captura os eventos de conexão e desconexão.

A recepção de eventos aciona automaticamente a avaliação das regras de inferência nebulosa para redefinição da *situação* do agente.

Conexão

Quando uma mensagem de conexão chega ao **Decisor**, ele verifica se possui canais alocados ociosos para estabelecer a conexão. Caso possua, aciona o **Executor** para enviar comandos à antena informando os canais que deverão ser usados. Se não possuir, o agente entra no estado de captação, caso ainda não esteja, e opera como apresentado anteriormente (Seção 4.2.4.1).

Desconexão

Quando recebe um evento de desconexão, o **Decisor** precisa apenas atualizar a **Base de Conhecimento**, uma vez que os canais ocupados já foram efetivamente liberados pela antena.

4.2.4.3 Recepção de Mensagens

O **Comunicador** pode enviar ao **Decisor** as mensagens provenientes do **Sistema de Controle** (mensagem de iniciação, mensagem de consulta e mensagem de atualização) e dos agentes vizinhos (mensagem de aviso, mensagem de alocação, mensagem de liberação, mensagem de solicitação e mensagem de colaboração). De acordo com a mensagem

recebida, o módulo **Decisor** realiza as funções correspondentes, conforme apresentado a seguir.

Iniciação

Sempre que um novo agente é inserido na rede móvel, ele precisa ser iniciado. Após a configuração do agente, através de mensagens de atualização, o **Sistema de Controle** envia uma mensagem de iniciação ao agente para que ele inicie suas operações.

Ao receber a mensagem, o **Comunicador** ativa o **Decisor** que realiza as seguintes operações:

1. preparação dos contadores internos para determinação da utilização e ocupação.
2. preparação dos contadores internos para acompanhamento do passo de previsão e do passo de treinamento.
3. definição das variáveis nebulosas.
4. envio de mensagens de aviso aos agentes vizinhos para informar sua presença e canais alocados.

Quando uma mensagem de aviso atinge um agente vizinho, este responde com uma mensagem de alocação informando sua lista de canais alocados. Após o recebimento de todas as mensagens de alocação da vizinhança, o novo agente entra em funcionamento.

Consulta

Ao receber uma mensagem de consulta, o **Decisor** identifica os dados requisitados e aciona o **Executor** para preparar as informações e enviá-las ao **Sistema de Controle**. Qualquer informação armazenada na **Base de Conhecimento** pode ser solicitada, incluindo parâmetros de configuração, como a lista de agentes vizinhos e variáveis de operação (por exemplo, *utilização* e *situação*).

Atualização

A mensagem de atualização contém um conjunto de nomes e valores. O *Decisor* identifica os nomes dos parâmetros e variáveis recebidos e aciona o *Executor*, que atualiza a *Base de Conhecimento* com os valores recebidos.

Além de atualizar diretamente as informações presentes na *Base de Conhecimento*, a mensagem de atualização também pode realizar o treinamento da rede neural de previsão, conforme será abordado na Seção 4.2.5.1.

Aviso

A mensagem de aviso informa que um novo agente foi ativado e está pronto para entrar em operação. Quando essa mensagem é recebida, o agente deve atualizar sua lista de vizinhos e os índices de separação presentes na lista de canais da rede. Além disso, deve enviar uma resposta através de uma mensagem de alocação, informando sua relação de canais alocados.

Alocação

A função de uma mensagem de alocação é indicar que um agente vizinho alocou canais na antena. O *Decisor* identifica os canais e aciona o *Executor* para atualizar a *Base de Conhecimento*, ou seja, para modificar a lista de vizinhos para incluir os canais alocados no agente correspondente e recalcular os índices de separação da lista de canais da rede.

Ela também é utilizada como resposta a uma mensagem de aviso enviada por um agente recém incorporado à rede. Isso ocorre para que o novo agente receba a lista de canais alocados em cada vizinho e atualize sua tabela de índices de separação.

Liberação

De forma análoga à alocação, o recebimento de uma mensagem de liberação resulta em modificação da lista de vizinhos, para remover os canais do agente correspondente, e redefinição dos índices de separação da lista de canais da rede.

Solicitação

O recebimento dessa mensagem indica que um agente vizinho necessita de mais recursos e informa a quantidade de canais que o agente solicitante deseja alocar. A função do **Decisor** nesse momento é verificar se pode oferecer canais ao agente solicitante. O tratamento de uma mensagem de solicitação pelo **Decisor** envolve a análise do estado interno do agente (variável *situação*).

Inicialmente, o agente verifica se existem canais ociosos para liberação. Caso não existam, envia uma mensagem de colaboração contendo uma lista vazia de canais. Se houver canais desocupados, as regras de inferência são avaliadas para atualização do estado do agente. Se a variável *situação* possuir um grau de pertinência no conjunto *liberação* maior que um mínimo pré-determinado, o agente conclui que pode fornecer recursos, caso contrário, responde com uma mensagem de colaboração contendo uma lista vazia. O valor mínimo do grau de pertinência pode ser configurado através do **Sistema de Controle** para cada agente implantado na rede móvel.

Se o **Decisor** decidir fornecer canais, o próximo passo é criar a lista de canais que será enviada como resposta ao agente solicitante. A quantidade de canais oferecidos pode ser configurada através do **Sistema de Controle**. Pode-se configurar um número máximo de forma que, independentemente da quantidade de canais requisitada e da quantidade de canais ociosos, a lista de canais oferecidos nunca ultrapasse esse valor máximo.

Outra abordagem, mais eficiente e flexível, é a utilizada na simulação do sistema

(Seção 5). A quantidade de canais oferecida depende do número de canais desejados pelo solicitante (c_d) e do número de canais ociosos (c_o), sendo determinada da seguinte forma:

$c_d > c_o/2$ Se a quantidade de canais desejados for maior que a metade dos canais disponíveis, o agente envia uma lista com metade dos canais disponíveis. Dessa forma, o agente colaborador manterá alguns canais ociosos para futuras conexões, sem necessidade de alocar mais canais. Para a determinação dos canais que serão oferecidos, o agente colaborador calcula o índice de separação dos canais ociosos para o agente solicitante, e envia os de maior valor.

$c_d \leq c_o/2$ Se a quantidade de canais desejados for maior ou igual que a metade dos canais disponíveis, o agente envia a lista completa de canais disponíveis. Uma vez que a necessidade do solicitante é relativamente pequena e não esgotará os recursos do agente colaborador, o envio da lista completa não trará problemas ao agente colaborador. Além disso, a lista completa oferece mais opções de canais, com diferentes índices de separação, aumentando a probabilidade de uso de canais pelo agente solicitante.

Colaboração

Conforme visto no item anterior, uma mensagem de colaboração é enviada como resposta a uma mensagem de solicitação. Assim, um agente só recebe uma mensagem de colaboração se estiver buscando recursos na vizinhança.

O módulo **Decisor** armazena em uma lista todos os canais (ofertas) que receber dos agentes vizinhos através de mensagens de colaboração. Quando todos os vizinhos tiverem respondido, mesmo que com listas vazias de canais, o **Decisor** calcula o índice de separação de cada canal oferecido, considerando que o canal não estaria mais presente no agente colaborador correspondente, e ordena a lista de acordo com os valores

obtidos. Os canais cujo índice de separação ficarem abaixo do mínimo estabelecido (ver Captação na Seção 4.2.4.1) são automaticamente descartados.

A seguir, o **Decisor** percorre a lista de canais a partir do item com maior índice de separação e aciona o **Executor** para realizar três ações: enviar comandos para a antena alocar o canal, atualizar a **Base de Conhecimento** e ativar o **Comunicador** para enviar mensagens de alocação aos vizinhos informando o canal alocado. A lista é percorrida até que a quantidade de canais desejada seja alocada ou até seu esgotamento.

4.2.5 Executor

O **Decisor** avalia as informações disponíveis, mensagens recebidas e eventos capturados para determinar a ação a ser realizada e os parâmetros envolvidos. O **Executor** é, então, ativado pelo **Decisor** e recebe as instruções necessárias para efetivamente realizar as tarefas.

As tarefas realizadas pelo **Executor** podem ser: *atualizações, comunicações ou comandos*.

4.2.5.1 Atualizações

Uma atualização consiste em modificar as informações armazenadas na **Base de Conhecimento**. Podem ser atualizações de parâmetros de configuração, variáveis de operação, variáveis e conjuntos nebulosos, ou reconfiguração da rede neural.

Em todos os casos, o módulo **Decisor** aciona o **Executor** e informa os parâmetros que devem ser alterados e os dados que deverão ser utilizados nas modificações. O **Executor** identifica o tipo de atualização em questão e realiza as operações necessárias na **Base de Conhecimento**.

No caso de parâmetros de configuração, variáveis de operação e variáveis nebulosas, a intervenção do **Executor** é muito simples. Basta identificar os parâmetros e

atribuir os novos valores recebidos.

Para a alteração de conjuntos nebulosos, além de reconfigurar as estruturas internas responsáveis pelo armazenamento dos conjuntos, após a alteração de um deles, as variáveis nebulosas que utilizam o conjunto modificado devem ser novamente fuzzificadas. Além disso, as regras de inferência que dependem das variáveis recalculadas devem ser reavaliadas para refletir os novos conjuntos e variáveis.

Embora raro, o *Sistema de Controle* também pode enviar mensagens para reconfigurar a rede neural de previsão. Esse procedimento deve ser realizado quando uma região sofreu grandes modificações na infraestrutura da rede móvel ou no comportamento dos usuários. Isso pode ocorrer por diversos fatores, como investimentos na rede, catástrofes climáticas, criação de centros comerciais etc.

A reconfiguração da rede neural é realizada através de treinamentos. O *Sistema de Controle* envia mensagens de atualização contendo o conjunto de entradas e o conjunto de saídas desejadas, bem como a quantidade de iterações que devem ser executadas. O *Executor* executa o procedimento para aprendizado da rede, com base na diferença entre o resultado obtido e o esperado para ajustar os pesos das sinapses da rede neural.

4.2.5.2 Comunicações

Quando o módulo *Decisor* determina que uma comunicação deve ser realizada, ele ativa o *Executor* e informa o que deve ser transmitido e para quais entidades do sistema. O *Executor* recebe essas instruções e ativa o *Comunicador* para enviar as mensagens necessárias.

Nesse processo, as interações entre os módulos do agente ocorrem em níveis diferentes. A interação entre o *Decisor* e o *Executor* ocorre através de instruções amplas, de alto nível. Por exemplo, o *Decisor* pode enviar uma instrução para o *Executor* informar os agentes vizinhos que o canal c_x foi alocado na antena. O *Decisor* não informa

qual é o tipo de mensagem a ser utilizado e nem quais são os agentes que deverão receber a mensagem.

A interação entre o *Executor* e o *Comunicador* utiliza instruções mais detalhadas, de baixo nível, ou seja, o *Executor* deve acionar o *Comunicador* e informar o tipo de mensagem que ele deseja enviar, seu conteúdo e seu destino. Seguindo o exemplo, o *Executor* consulta a lista de vizinhos armazenada na *Base de Conhecimento* e, para cada agente a_i presente na lista, instrui o *Comunicador* para enviar uma mensagem de alocação contendo o canal c_x para o agente a_i .

Essa divisão de tarefas permite a criação de módulos especializados e flexíveis. Podemos, por exemplo, trocar o protocolo de comunicação apenas modificando o módulo *Comunicador*, sem interferir no funcionamento dos demais subsistemas.

4.2.5.3 Comandos

Para simplificar a arquitetura interna do agente, o próprio *Executor* envia os comandos para a antena. Idealmente, essa função deveria ser realizada por um módulo *Atuador*, que traria a mesma flexibilidade apresentada pela divisão de tarefas entre os módulos *Executor* e *Comunicador*. Entretanto, como o tratamento das instruções recebidas do *Decisor* e as operações necessárias para o envio dos comandos são relativamente simples e cada *Executor* será responsável por enviar comandos apenas para uma antena, a estação local, optou-se por implementar o *Atuador* diretamente no *Executor*.

O *Executor* pode enviar os seguintes comandos para a estação: comando de alocação, comando de liberação e comando de uso.

Após deliberação, o *Decisor* pode acionar o *Executor* e requisitar a alocação de um ou mais canais. O *Executor* recebe, então, uma lista de canais que deverão ser alocados na estação local. Essa lista é enviada para a antena através de um comando de

alocação.

De forma análoga, caso o **Decisor** resolva remover alguns canais da estação local, o **Executor** recebe uma instrução de desalocação e uma lista de canais. O **Executor** envia um comando de liberação para a antena contendo os canais que deverão ser liberados.

O comando de uso é enviado como resposta a uma mensagem de consulta. Quando um pedido de conexão chega à estação local, ele é capturado pelo **Sensor**, que repassa o evento para o **Decisor**. O **Decisor** determina se o pedido será atendido e, em caso afirmativo, quais canais serão utilizados. O **Executor** é ativado e envia um comando de uso para a antena contendo a relação de canais definida pelo **Decisor**.

O comando de uso também é empregado quando o **Decisor** resolve alocar outros canais na estação base para atender novos pedidos de conexão. A antena, nesse caso, deve alocar um novo canal e utilizá-lo para estabelecer a conexão solicitada.

4.3 Sistema de Controle

O **Sistema de Controle** possibilita a interação com os agentes distribuídos do sistema. Ele pode ser implementado como programas de linha de comando ou como uma aplicação gráfica. Em ambos os casos, sua função é permitir que operadores criem, consultem e configurem os agentes do sistema.

Este módulo realiza três operações básicas: *consulta*, *atualização*, *iniciação*. Todas as operações são executadas na forma de mensagens, que são enviadas aos agentes. Os agentes, por sua vez, interpretam as informações recebidas e enviam mensagens de resposta ao **Sistema de Controle**.

Através das mensagens de consulta, pode-se consultar qualquer informação presente na **Base de Conhecimento** do agente. Essa mensagem deve conter o identificador do agente consultado e uma lista contendo os parâmetros desejados. O agente destino enviará uma mensagem de consulta com uma lista de pares (*nome*, *valor*), que possui

os dados solicitados.

As atualizações são efetuadas por meio de mensagens de atualização que contém o identificador do agente destino e uma relação de pares (*nome, valor*). O agente verifica a lista recebida e responde com uma mensagem de atualização com pares (*nome, resultado*), que indicam, no *resultado*, se a atualização foi efetuada com sucesso. O caso específico de atualização da rede neural também utiliza esse padrão. O **Sistema de Controle** envia pares no formato (*nome, valor*) informando os conjuntos de treinamento (entrada e saída esperada).

A mensagem de iniciação é executada para ativar o agente e iniciar sua operação. Ela contém apenas o identificador do agente e a resposta possui apenas o *resultado*, que indica se a iniciação foi ou não executada com sucesso.

De acordo com as mensagens apresentadas, o procedimento para a criação de novos agentes na rede começa pela configuração, realizada através de mensagens de atualização. Após a configuração de todos os parâmetros e variáveis de operação, deve-se enviar uma mensagem de iniciação para que o agente seja ativado e comece a funcionar.

Caso o **Sistema de Controle** seja implementado como uma aplicação gráfica, outra funcionalidade possível é a apresentação de agentes, que permite visualizar todos os agentes presentes na rede, seus estados e interconexões com os agentes vizinhos.

4.4 Funcionamento

Para melhor esclarecer o funcionamento do sistema, as próximas seções apresentam diagramas de sequência dos principais cenários, destacando as entidades, mensagens e informações envolvidas em cada etapa.

4.4.1 Chegada de pedido de conexão

Esse cenário inicia com a chegada de um pedido de conexão para a antena. Isso ocorre sempre que um usuário da rede móvel deseja estabelecer uma comunicação. A antena envia o evento ao *Sistema de Controle*, que aciona o *Decisor*.

Neste ponto, o *Decisor* consulta a *Base de Conhecimento* e, caso existam canais ociosos suficientes para estabelecer a conexão, o *Decisor* determina quais serão os canais utilizados e instrui o *Executor* a estabelecer a comunicação através deles. O *Executor* recebe a mensagem do *Decisor* e envia os comandos correspondentes para a antena. A Figura 4.11 apresenta este cenário.

Figura 4.11: Estabelecimento de uma conexão com utilização de canais ociosos.

Caso não existam canais ociosos suficientes para o estabelecimento da comunicação requisitada, o *Decisor* verifica se existem *canais livres*. Caso existam canais livres suficientes para, somados aos ociosos, efetuar o estabelecimento da conexão, o *Decisor* seleciona alguns canais para alocação e cria uma lista de canais combinando os canais ociosos, se houver, com os canais livres escolhidos. Essa relação é enviada ao *Executor* que envia comandos para a antena alocar e utilizar os canais informados para a criação da conexão. O *Executor* também aciona o *Comunicador* para informar aos vizinhos os canais livres que foram alocados. O cenário completo é apresentado na Figura 4.12.

Caso não existam canais ociosos e livres suficientes, o agente pode, ainda, solicitar canais aos agentes vizinhos. Para isso, o *Decisor* consulta a *Base de Conhecimento* e

Figura 4.12: Estabelecimento de uma conexão com utilização de canais livres.

avalia as regras de inferência nebulosa para determinar a *situação*. Caso o grau de pertinência dessa variável no conjunto *captação* seja maior que um valor configurado, o agente opta por verificar se os agentes vizinhos poderão ajudar. Se estiver abaixo desse valor, o processo é finalizado e a antena rejeitará o pedido de conexão por *timeout*, ou seja, como a antena não recebeu instruções do agente dentro de um tempo configurado na antena, o pedido de conexão não será atendido.

Para solicitar canais aos agentes vizinhos, o **Decisor** aciona o **Comunicador** para que mensagens de solicitação sejam enviadas. Cada agente vizinho responderá com uma mensagem de colaboração, informando os canais que podem oferecer. Após o recebimento das respostas de todos os vizinhos, o **Decisor** verifica se os canais oferecidos podem ser alocados, através do cálculo do índice de separação. Por último, tendo obtido a lista completa de canais disponíveis, formada por ociosos, livres e oferecidos, o **Decisor** verifica se eles são suficientes para a conexão solicitada pelo usuário. Caso positivo, o **Executor** é acionado e envia as instruções para a antena. Caso contrário, a antena rejeitará o pedido por *timeout*. O cenário completo é apresentado na Figura 4.13. A linha tracejada na antena marca o tempo de *timeout* configurado, depois do qual a antena rejeitará o pedido se não receber instruções.

Em todas as situações onde a conexão é estabelecida, o **Executor**, além de enviar

Figura 4.13: Estabelecimento de uma conexão com solicitação de canais.

os comandos para a antena, também atualiza a **Base de Conhecimento**.

4.4.2 Aviso de desconexão

Quando uma comunicação é finalizada, seja por desejo do usuário, falha na rede, áreas de sombra, ou qualquer outro motivo, a antena identifica o evento, que é capturado pelo **Sensor**. O **Decisor** é acionado e instrui o **Executor** para atualizar a **Base de Conhecimento**.

Figura 4.14: Finalização de uma conexão.

Nesse momento, as regras de inferência são avaliadas para a determinação da variável *situação*. Caso o grau de pertinência dessa variável no conjunto *liberação* seja superior a um valor configurado, o **Decisor** decide liberar canais, ou seja, alguns canais ociosos são desalocados, ficando disponíveis para utilização de outros agentes da

vizinhança. Além disso, o **Comunicador** é instruído a enviar mensagens de liberação aos agentes vizinhos informando quais foram os canais liberados.

A Figura 4.14 apresenta o cenário mais simples, onde apenas ocorre a atualização da **Base de Conhecimento**. O cenário completo, com a atualização da **Base de Conhecimento** e a desalocação de canais ociosos é apresentado na Figura 4.15.

Figura 4.15: Finalização de uma conexão.

4.4.3 Sistema de Controle

Conforme mencionado, o **Sistema de Controle** pode realizar três operações: consulta, atualização e iniciação dos agentes. Na consulta, o **Sistema de Controle** envia uma mensagem de consulta ao agente, que é recebida pelo **Comunicador**. O **Decisor** é acionado e ativa o **Executor**, que fará a leitura dos parâmetros desejados na **Base de Conhecimento**. Os valores dos parâmetros são obtidos e enviados ao **Comunicador**, que responderá ao **Sistema de Controle**. A Figura 4.16 apresenta a seqüência de operações.

Figura 4.16: Consulta de parâmetros pelo **Sistema de Controle**.

A atualização, apresentada na Figura 4.17, é muito semelhante à consulta. A mensagem de atualização é recebida pelo mensagem de consulta e repassada ao **Decisor**, que ativa o **Executor** para realizar a atualização dos parâmetros fornecidos. Após a atualização, o **Executor** verifica se os valores foram armazenados corretamente e aciona o mensagem de consulta para que uma mensagem de atualização seja enviada ao **Sistema de Controle** com o resultado da operação.

Figura 4.17: Atualização de parâmetros pelo *Sistema de Controle*.

A iniciação começa com a recepção da mensagem pelo **Comunicador**. O **Decisor** é acionado e ativa o **Executor**. A partir da lista de vizinhos armazenada na **Base de Conhecimento**, o **Executor** instrui o **Comunicador** a enviar uma mensagem de aviso aos agentes vizinhos, informando os canais alocados.

Figura 4.18: Iniciação de um agente.

Os agentes vizinhos identificam que um novo agente foi inserido na rede e cada

agente responde com uma mensagem de alocação, com o objetivo de atualizar a **Base de Conhecimento** do novo agente com os canais alocados na vizinhança.

As respostas dos agentes vizinhos são transmitidas até o **Executor**, que atualiza a lista de vizinhos na **Base de Conhecimento** e recalcula os índices de separação dos canais da rede.

4.4.4 Alocação e Liberação de Canais

A alocação e a liberação de canais por agentes vizinhos são tratadas de forma muito semelhante (Figuras 4.19 e 4.20). Quando o **Comunicador** identifica uma mensagem de alocação ou uma mensagem de liberação, ele envia os dados ao **Decisor**, que aciona o **Executor** para atualizar a lista de canais do agente na **Base de Conhecimento** e recalcular os índices de separação dos canais.

Figura 4.19: Alocação de canais.

Figura 4.20: Liberação de canais.

4.4.5 Solicitação

O **Comunicador** recebe uma mensagem de solicitação e a envia ao **Decisor**, que avalia as regras de inferência nebulosa para obtenção da *situação*. Caso o grau de pertinência de *situação* no conjunto *liberação* seja superior a um valor configurado, o **Decisor** seleciona alguns canais para oferecer ao agente solicitante. Esses canais são armazenados na lista de Canais Oferecidos e ficam reservados. O **Executor** é acionado e ativa o **Comunicador** para enviar uma mensagem de colaboração ao agente solicitante. Após deliberação, o agente solicitante envia uma mensagem de alocação informando quais foram os canais alocados por ele. Caso a lista contenha canais oferecidos, o **Decisor** instrui o **Executor** a realizar a liberação desses canais: atualização da **Base de Conhecimento**, envio de mensagens de liberação aos agentes vizinhos e recálculo dos índices de separação. O **Decisor** também é responsável por remover os canais da lista de Canais Oferecidos. A Figura 4.21 apresenta o cenário.

Figura 4.21: Alocação de canais.

4.4.6 Colaboração

Uma mensagem de colaboração somente pode ser recebida se o agente enviou uma mensagem de solicitação. Por outro lado, uma mensagem de solicitação só será enviada se um agente não conseguir atender um usuário com recursos próprios ou utilizando recursos livres na rede (canais livres). Esse cenário já foi apresentado na Seção 4.4.1.

5 SIMULAÇÃO DE OPERAÇÃO

Com o objetivo de verificar o comportamento dos agentes, uma versão simplificada do sistema foi construída. Para a execução dos testes, criou-se uma programa para simular uma rede móvel que atende uma região geográfica delimitada.

Com o teste, avaliamos o atendimento aos usuários da rede, analisamos as decisões tomadas pelos agentes e verificamos os benefícios obtidos pelo uso da rede neural de previsão, presente em cada agente.

5.1 Cenário da Simulação

5.1.1 Região

A Figura 5.1 apresenta a região geográfica definida para a simulação. Foram criadas 30 células hexagonais de mesmo tamanho para atender a área de cobertura. Cada célula é atendida por uma estação base, que possui um agente para gerenciamento dos canais.

5.1.2 Demanda

A simulação cobre o funcionamento da rede durante o período de 24 horas. No início da simulação, para cada célula foi associada uma curva característica, que define a demanda no decorrer do dia. Foram criadas 4 curvas características distintas, identificadas por *Curva 1*, *Curva 2*, *Curva 3* e *Curva 4*, conforme apresentado na Figura 5.2.

Figura 5.1: Células hexagonais definidas para a execução da simulação.

As curvas características foram distribuídas uniformemente entre os agentes da simulação, criando regiões com diferentes demandas. A criação e distribuição de curvas diferentes possibilita a obtenção de um cenário mais próximo da realidade, onde cada célula atende uma região com necessidades próprias.

O objetivo do sistema é realocar os recursos ociosos em regiões de baixo tráfego para regiões congestionadas. A distribuição não uniforme obtida através das curvas características permite observar o ganho proporcionado pelo sistema proposto num cenário compatível com o problema para o qual foi modelado.

5.2 Eventos

A interação dos usuários com a rede móvel foi modelada por um *conjunto de eventos*. Para cada teste executado, uma lista de pedidos de conexão foi gerada para cada célula. Essa lista é formada por pares (*instante*, *duração*). O *instante* indica o minuto do dia no qual o pedido chegou, entre 0 e 1440, cobrindo um dia de utilização da célula. A *duração* informa quanto tempo a conexão permaneceu ativa, caso tenha sido aceita, e varia entre 1 e 20 minutos.

Figura 5.2: Curvas Características.

5.2.1 Geração

A criação dos conjuntos de eventos é realizada automaticamente pelo simulador no início da execução. Para cada célula é associada uma *curva de característica* que define o número médio de canais ocupados em determinado momento. A Figura 5.3 apresenta um exemplo de curva de utilização.

Figura 5.3: Curva característica: utilização média de canais no período de um dia.

Para a obtenção dos instantes de chegada dos pedidos, utilizou-se conceitos de *Simulação de Sistemas* (JAIN, 1991). Se iniciarmos com $t_0 = 0$, podemos obter todos os instantes subsequentes através da obtenção do intervalo entre chegadas. Para isso, precisamos do número médio de usuários ($E[n]$), do tempo médio de serviço ($E[s]$), do tempo médio de resposta ($E[r]$) e da taxa média de chegada (λ).

Foi considerado que todos os pedidos de conexão solicitam apenas um canal de comunicação¹, portanto, a quantidade de canais em uso é igual ao número de usuários sendo atendidos. Assim, $E[n]$ pode ser obtido diretamente da curva de comportamento.

Conforme apresentado, a duração das chamadas segue uma distribuição uniforme entre 1 e 20, assim:

$$E[s] = \left(\frac{1 + 20}{2} \right) = 10,5 \quad (5.1)$$

Se considerarmos o caso ideal, onde a célula sempre possui recursos suficientes para atender a demanda, um pedido de conexão que chega em uma célula é atendido quase que imediatamente, logo o tempo de resposta é praticamente igual ao tempo de serviço:

$$E[r] = E[s] \quad (5.2)$$

Como:

$$E[n] = \lambda \cdot E[r] \quad (5.3)$$

¹Essa suposição não afeta os resultados da simulação, uma vez que o foco são canais em uso e não usuários. O único objetivo e consequência dessa simplificação é facilitar a geração de massas de teste.

Temos:

$$\lambda = \frac{E[n]}{E[r]} = \frac{E[n]}{E[s]} = \frac{E[n]}{10,5} \quad (5.4)$$

Os eventos foram gerados em passos de 30 minutos, ou seja, a quantidade de canais ocupados obtida da *curva característica* em $t = 30$ foi utilizada como sendo o número médio de usuários em atendimento para a geração dos eventos entre $t_{i_1} = 0$ e $t_{f_1} = 30$, a quantidade de canais ocupados em $t = 60$ foi utilizada para geração entre $t_{i_2} = 31$ e $t_{f_2} = 60$ e assim por diante. Assim, o valor de λ é redefinido a cada 30 minutos, permitindo que a chegada de eventos acompanhe a *curva de comportamento*.

Tendo definido λ para cada intervalo de 30 minutos, os eventos são gerados utilizando uma distribuição exponencial para obter o intervalo entre chegadas e uma distribuição uniforme para determinar a duração do pedido de conexão (JAIN, 1991).

Os números aleatórios foram obtidos através do pacote **Colt**², desenvolvido pelo CERN (*European Organization for Nuclear Research*).

Os instantes de chegada iniciam em $t_0 = 0$. Para determinar o próximo instante de chegada (t_{i+1}), o gerador obtém um número aleatório com distribuição exponencial de média $1/\lambda$ (que varia de acordo com o intervalo de 30 minutos no qual t_i está inserido) e o soma ao último instante obtido (t_i). A duração do pedido é determinada pela geração de um número aleatório com distribuição uniforme entre 1 e 20. De posse do instante de chegada e da duração do pedido, o par (*instante, duração*) é inserido na lista de eventos.

A Figura 5.4(a) apresenta o número médio de canais ocupados a cada minuto para

²Copyright (c) 1999 CERN - European Organization for Nuclear Research.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

um conjunto de eventos gerado a partir da curva de comportamento da Figura 5.3, enquanto a Figura 5.4(b) apresenta a média de canais ocupados a cada 30 minutos no mesmo período.

(a) Média a cada minuto.

(b) Média a cada 30 minutos.

Figura 5.4: Curva gerada por um conjunto de eventos.

5.2.1.1 Distribuições

Conforme mencionado, a geração dos eventos utiliza uma distribuição exponencial para obter o intervalo entre chegadas de pedidos de conexão. Para simplificação do modelo, consideramos que os intervalos entre retentativas de conexão, realizadas pelos usuários que tiveram pedidos recusados pela rede, também seguem uma distri-

buição exponencial. Desse modo, podemos as tentativas aparecerem incorporadas nos intervalos entre chegadas, sem necessidade de tratamento diferenciado.

Outra simplificação é a utilização de uma distribuição uniforme para a duração das chamadas. Pode-se verificar, em redes móveis em produção, que a grande maioria das comunicações ficam entre valores mínimo e máximo bem delimitados. Essa abordagem também é comum nos trabalhos existentes, como pode ser observado no trabalho de Hac e Chen (2000).

5.3 Implementação

Para simplificar a codificação sistema, os dados armazenados nas *Bases de Conhecimento* são acessíveis a todos os agentes, ou seja, um agente tem acesso direto às informações armazenadas em qualquer outro agente da rede. Isso diminui a comunicação entre agentes, facilitando a implementação do *Comunicador*.

O *Sistema de Controle* que seria responsável pelo gerenciamento dos agentes do sistema, não foi implementado. Esse módulo foi considerado desnecessário para a simulação e, conseqüentemente, para a validação do modelo. Por essa razão, a simulação não aborda o tratamento das mensagens do *Sistema de Controle* que seriam recebidas.

A linguagem *Java*³ foi utilizada para a implementação dos agentes e do controle da simulação. Essa linguagem possui características que a tornam uma escolha ideal para a criação do simulador. Entre elas, destacam-se:

- forte orientação a objeto, que permite rápido desenvolvimento e associação entre modelo e código;
- simplicidade e facilidade na criação de aplicações multi-tarefas;

³*Java* é uma linguagem de programação orientada a objetos e independente de plataforma desenvolvida pela *Sun Microsystems*. Mais informações podem ser encontradas em <http://java.sun.com>.

- disponibilidade de pacotes para a implementação dos módulos neural e nebuloso.

Pequenas alterações no modelo foram introduzidas para a construção do sistema de simulação. As modificações realizadas nos módulos foram simples e pontuais, e não interferem na arquitetura apresentada no Capítulo 4.

A simulação ocorre em tempo discreto, em passos que acompanham a ocorrência de eventos. Isso permite que o ambiente de simulação seja simples, sem tratamento de eventos em tempo real e sem necessidade de grande poder de processamento para simular, simultaneamente, o grande número de células e usuários envolvidos.

Além disso, um novo componente, **Controle**, foi criado para controlar a execução da simulação.

5.3.1 Controle

Esse módulo, adicionado para a execução do teste, realiza o gerenciamento da simulação e fornece registros dos eventos e ações dos agentes distribuídos.

Como não existem antenas reais na simulação, o **Controle** também faz o papel de antena para todos os agentes, ou seja, é ele quem envia mensagens de conexão e mensagens de desconexão aos **Sensores**. Além disso, ele também recebe os comandos que o **Executor** enviaria à antena, com instruções para alocação, liberação e utilização de canais.

Com isso, os agentes interagem com o **Controle** da mesma forma que o fariam com uma antena real, garantindo uma ótima aproximação da realidade.

O **Controle** também é responsável pela criação dos agentes, papel que seria do **Sistema de Controle**. Com base na região geográfica definida para a simulação (Figura 5.1), os agentes são criados com suas respectivas listas de agentes vizinhos. A seguir, cada agente recebe uma lista de canais que serão alocados na célula. A forma

de distribuição dos canais entre os agentes será abordada posteriormente.

5.3.1.1 Eventos

Quando a simulação é iniciada, o **Controle** gera um conjunto de eventos para cada célula presente na simulação. Cada evento criado recebe uma identificação que informa a qual célula ele pertence. Todos os conjuntos são, então, combinados e sorteados em ordem crescente de ocorrência. Dessa forma, cria-se uma lista cronológica de eventos da rede, que é consultada pelo **Controle** durante a simulação.

A cada passo, o **Controle** retira a próxima entrada (*célula, instante, duração*) presente na lista de eventos e envia uma mensagem ao **Sensor** do respectivo agente, que efetua o tratamento do evento.

Quando o **Controle** envia uma mensagem de conexão ao **Sensor**, o agente verifica se o pedido será ou não aceito. Caso a comunicação seja estabelecida, o **Executor** responde ao **Controle** informando qual canal deverá ser utilizado (na simulação, considerou-se que cada pedido de conexão solicita apenas um canal). O **Controle** recebe essa informação e insere o novo evento na lista de eventos, respeitando a ordenação crescente do *instante*. A nova entrada inserida possui os dados (*célula, instante, canal*), informando a qual célula pertence, o instante de desconexão e o canal que deverá ser liberado.

5.3.1.2 Variáveis e Previsões

Além de transmitir os eventos aos agentes, o **Controle** também aciona a atualização das variáveis lingüísticas e a realização de novas previsões. A cada 30 minutos (tempo simulado), o **Controle** ativa a realização de uma nova previsão e o ajuste das variáveis *ocupação, utilização e utilização prevista*, para cada agente da rede.

5.3.2 Fluxo de Execução

Controle

Em cada passo, o *Controle* realiza as seguintes operações:

1. Consulta a lista de eventos para identificação do próximo evento;
2. Envia mensagem ao *Sensor* do agente correspondente;
3. Ativa o *Sensor* para tratamento da mensagem;
4. Verifica a necessidade de enviar mensagens aos agentes vizinhos;
5. Se mensagens foram enviadas, ativa o *Comunicador* dos vizinhos para tratamento das mensagens;
6. Se espera respostas dos vizinhos, ativa o *Comunicador* do agente para tratamento;

Essas operações são realizadas até que não existam mais eventos dentro do intervalo correspondente ao passo de execução.

A cada leitura da lista de eventos (item 1), o tempo simulado é consultado para ativar, a cada 30 minutos, a atualização das variáveis nebulosas dos agentes.

Agentes

A ativação dos *Sensores* pelo *Controle* inicia o processamento do agente, o qual é o elemento alvo da simulação. O *Sensor* repassa as informações ao *Decisor* que, conforme o estado interno do agente, decide quais serão as ações realizadas, segundo o funcionamento apresentado na Seção 4.4.

Os eventos, consultas, decisões, ações e respostas de cada agente são registrados em um arquivo para análise. Esse registro armazena os valores das variáveis internas

que determinam o estado do agente e as decisões e ações realizadas. Com isso, pode-se verificar o comportamento dos agentes para cada evento individual ocorrido no cenário de simulação.

5.3.3 Rede Neural

A rede neural foi implementada utilizando o pacote *Joone (Java Object Oriented Neural Engine)*, sistema que permite a criação, treinamento e teste de redes neurais artificiais.

Como a simulação cobre a operação da rede durante um dia e a demanda em cada célula segue uma curva característica definida (Seção 5.1.2), o treinamento contínuo suportado pelo modelo proposto não foi implementado, uma vez que é totalmente desnecessário para a execução dos testes.

Para que as redes neurais dos agentes operem de forma correta, o treinamento foi realizado antes da execução dos testes. Um treinamento foi realizado para cada curva característica, resultando em 4 conjuntos de vetores para configuração das redes neurais. No momento em que a curva característica de uma célula é definida pela simulação (Seção 5.1.2), a rede neural presente na **Base de Conhecimento** do agente é configurada com os vetores correspondentes à curva escolhida.

Desse modo, eliminou-se a necessidade de treinamento através do Joone, que foi utilizado apenas para simulação. O agente fornece os valores de horário (h) e de utilização ($util$) para a rede neural, que responde com a utilização prevista ($util_{prev}$), de acordo com os pesos das conexões sinápticas pré-configuradas.

5.3.4 Teoria Nebulosa

Para inserir a teoria nebulosa nos agentes, foi utilizado o pacote *Fuzzy Engine for Java 1.0*, desenvolvido por *Edward S. Sazonov*⁴. Este sistema possui código fonte aberto e foi projetado para permitir a utilização de variáveis lingüísticas e a avaliação de regras de inferência nebulosa em programas Java.

Através desse pacote, as variáveis lingüísticas e seus respectivos conjuntos nebulosos foram criados na **Base de Conhecimento** e a avaliação das regras nebulosas foi implementada no **Decisor**. O conjunto de regras também foi armazenado na **Base de Conhecimento**, seguindo o modelo apresentado na Seção 4.2.3.12.

5.3.5 Registros de Teste

Além de gravar os eventos ocorridos nas células e as decisões e ações dos agentes, o **Controle** também registra se os pedidos de conexão foram aceitos ou recusados. Essa informação indica a capacidade da célula em atender os pedidos de seus usuários.

Em especial, o registro de conexões recusadas permite que a *taxa de bloqueio* (*blocking rate*) seja determinada. Essa taxa é a métrica mais importante da simulação.

5.4 Execução

Para realizar uma análise quantitativa do desempenho do sistema, os testes executados foram comparados com uma *rede de referência*, ou seja, cada teste realizado na *rede de teste* (rede que utiliza o sistema proposto) foi também executado em uma rede sem o sistema.

A comparação entre os dois casos de teste possibilita a verificação do ganho proporcionado pelo sistema proposto em relação a uma rede móvel padrão, com distribui-

⁴Edward S. Sazonov é Ph.D. em Engenharia de Computadores pela West Virginia University, USA.

ção fixa de recursos (sem realocação dinâmica).

Quando o **Controle** inicia a simulação, o primeiro passo é criar a rede móvel que será utilizada no teste. Para isso, as seguintes tarefas são executadas:

1. Criação dos agentes;
2. Atualização da Lista de Vizinhos de cada agente;
3. Definição dos canais existentes na rede;
4. Distribuição dos canais entre os agentes;
5. Definição da curva característica de cada célula;
6. Criação de um conjunto de eventos para cada célula;
7. Atualização da rede neural de cada agente de acordo com a curva característica da célula;
8. Unificação dos conjuntos de eventos em uma lista de eventos em ordem cronológica;

5.4.1 Definição e Distribuição de Canais

Das tarefas listadas, apenas os itens 3 e 4 ainda não foram abordados: a definição e distribuição dos canais. A definição dos canais da rede informa quantos canais estão disponíveis para uso. Na simulação, os canais são identificados por índices, que variam de c_1 até c_n , onde n é a quantidade de canais configurada.

Definidos os canais ($c_1 \cdots c_n$), eles são distribuídos entre as células de forma a evitar dois tipos de interferência: *co-canal* e *canal adjacente*. A interferência de co-canal ocorre quando o mesmo canal é utilizado em células adjacentes. A interferência de canal adjacente ocorre quando dois canais sucessivos são utilizados na mesma célula.

Com isso, de acordo com a Tabela 4.3 (Seção 4.2.3.2), o *índice de separação* mínimo configurado nos agentes da simulação foi 3.

Uma vez que o objetivo principal da simulação é avaliar a capacidade de adaptação do sistema, a distribuição inicial de canais entre os agentes não recebeu atenção especial. A Figura 5.5 apresenta o algoritmo utilizado para realizar a distribuição.

Devido ao algoritmo de distribuição e ao posicionamento das células na região geográfica, a rede foi criada com 81 canais, o que permite a distribuição exata de 24 canais por célula.

As curvas características foram definidas com um valor máximo de 24 canais em uso. Isso significa que, nas regiões de alta demanda de uma curva, os eventos criados geram trechos que oscilam em torno da média definida de 24 canais.

Esses números foram utilizados por serem próximos das quantidades encontradas em redes celulares reais, além de permitir a análise do modelo proposto em situações de trabalho próximas da capacidade máxima suportada pela rede.

5.4.2 Simulação das Redes

Para garantir que as duas redes simuladas (referência e teste) sejam comparadas corretamente, o **Controle** cria duas instâncias idênticas e independentes da rede. A instância correspondente à rede de referência é configurada para desabilitar algumas funções dos agentes. Os agentes dessa rede não são capazes de alocar canais livres, liberar canais ociosos, nem solicitar ou fornecer canais para agentes vizinhos.

A distribuição das curvas características serão as mesmas nas duas redes, ou seja, se a *Curva 3* for associada a *Célula 1* na *rede de teste*, o mesmo ocorre na *rede de*

Figura 5.5: Algoritmo utilizado na distribuição inicial de canais.

referência. Da mesma forma, os conjuntos de eventos gerados para as células são utilizados nas duas redes. Assim, todas as células de uma rede são submetidas exatamente aos mesmos eventos da outra.

Durante a execução, o ***Controle*** gerencia as duas redes simultaneamente. Cada evento é enviado à duas células, uma pertencente à rede de referência e outra pertencente à rede de teste.

Cada rede trabalha isoladamente e não existe comunicação e/ou interferência entre as duas simulações. Os registros de eventos, decisões e ações são armazenados separadamente, permitindo a análise independente da simulação de cada rede.

6 ANÁLISE DOS RESULTADOS

A análise dos resultados obtidos está dividida em duas partes: rede neural e simulação.

A análise da rede neural tem o objetivo de determinar qual a arquitetura mais adequada para utilização nos testes e avaliar se os resultados obtidos são satisfatórios.

Após a definição da rede neural, a simulação foi executada no cenário definido na Seção 5.1 e avaliado o funcionamento do sistema a partir dos registros de eventos armazenados.

6.1 Rede Neural

Seguindo a arquitetura apresentada na Seção 4.2.3.14, diversas redes neurais foram criadas com diferentes quantidades de neurônios na camada escondida. Entre outros valores, foram testadas redes com 6, 8, 12, 16, 24 e 32 neurônios. Pelos resultados obtidos, a escolha recaiu sobre a rede com 16 camadas, por apresentar um comportamento satisfatório no aprendizado de mais de um comportamento, com um número reduzido de neurônios.

Assim, a rede neural de previsão construída para a simulação possui 2 neurônios de entrada, 16 neurônios na camada escondida e 1 neurônio de saída.

A rede neural foi construída utilizando o aplicativo MatLab¹ (Figura 6.1). Esse

¹O MatLab é um software da empresa *MathWorks* para simulação, cálculo, estatística e processamento matemático, entre outros. O aplicativo é bastante conhecido e amplamente utilizado no meio

software possui um módulo (*nntool*) específico para construção, treinamento e simulação de redes neurais.

Figura 6.1: Diagrama da rede neural criada no MatLab.

6.1.1 Verificações

Para validar a rede neural, inicialmente criamos dois *conjuntos de treinamento*. O primeiro verificou o aprendizado da rede neural para uma curva característica determinada. O segundo foi utilizado para verificar a adaptação da rede caso a curva de comportamento sofra alterações.

Após a validação da rede através dos dois conjuntos de treinamento criados, um terceiro conjunto foi usado para avaliar a capacidade da rede em aprender mais de uma curva característica. Isso é necessário para atender células que possuem comportamentos distintos e cíclicos, como, por exemplo, um comportamento durante os dias úteis da semana e outro nos finais de semana.

Cada *conjunto de treinamento* é formado por dois *conjuntos de eventos*. Eles fornecem os vetores para treinamento e simulação da rede neural. O primeiro conjunto de eventos foi utilizado para o aprendizado da rede. O *vetor de entrada* foi apresentado para a rede neural, que também recebe o *vetor alvo*, para ajuste dos pesos das conexões sinápticas. Uma vez treinada a rede, o segundo conjunto de eventos realizou a simulação. O vetor de entrada foi apresentado à rede e as respostas foram armazenadas para comparação com o vetor alvo correspondente. Com isso, pode-se comparar os valores esperados (vetor alvo do segundo conjunto de eventos) com os valores obtidos pela

acadêmico e científico.

rede neural de previsão.

Conforme apresentado na Seção 5.2, os conjuntos de eventos são seqüências formadas por pares (*instante*, *duração*). Partindo de um conjunto de eventos, criamos uma lista na forma (*início*, *fim*), onde *início* é o instante no qual um canal é ocupado e é igual ao *instante* de um elemento do conjunto de eventos, enquanto *fim* é o instante no qual o canal é liberado, e é igual ao valor do *instante* somado com o valor da *duração* de um elemento do conjunto.

Para exemplificar, o conjunto de eventos apresentado na Tabela 6.1 gera a lista (*início*, *fim*) apresentada na Tabela 6.2.

<i>instante</i>	<i>duração</i>
01,162176932	11,924646600
01,230468751	19,392060290
03,714153441	06,822513164
13,611949940	01,170537946
14,068402780	16,288187680
15,887735100	08,989196993
19,867282190	03,855984336
21,797546610	08,578122254
27,661631850	02,164180532
28,815280920	11,969100560
30,702724780	08,734231977
31,688706530	12,880802480
34,671086220	05,591736050
36,231613220	10,037032710
38,922122990	06,276398977
39,787733400	13,581702110
40,590581250	13,963406210
41,277036330	14,702189080
42,718156250	10,565772850
47,261363580	03,183669928

Tabela 6.1: Conjunto de eventos.

<i>início</i>	<i>fim</i>
01,162176932	13,086823530
01,230468751	20,622529040
03,714153441	10,536666610
13,611949940	14,782487890
14,068402780	30,356590460
15,887735100	24,876932100
19,867282190	23,723266520
21,797546610	30,375668860
27,661631850	29,825812380
28,815280920	40,784381480
30,702724780	39,436956760
31,688706530	44,569509020
34,671086220	40,262822270
36,231613220	46,268645940
38,922122990	45,198521970
39,787733400	53,369435510
40,590581250	54,553987450
41,277036330	55,979225410
42,718156250	53,283929100
47,261363580	50,445033510

Tabela 6.2: Lista (*início*, *fim*).

As previsões são realizadas em passos de 30 minutos, portanto o treinamento da rede neural deve ser feito aplicando esse mesmo passo. O vetor da entrada horário (*h*) percorre um dia de utilização, ou seja, assume os valores: 0,0, 0,5, 1,0, ..., 23,5. O vetor de entrada de utilização (*util*) é obtido a partir da lista (*início*, *fim*). Para cada intervalo de 30 minutos, verificam-se quantas conexões estavam ativas, quantas foram iniciadas e quantas terminaram. O valor resultante é a quantidade de canais em uso

naquele intervalo.

Seguindo o exemplo, a lista (*início*, *fim*) apresentada na Tabela 6.2 resulta na *lista de utilização*, apresentada na Tabela 6.3, que é a base para gerar o *vetor de entrada*, conforme apresentado na Seção 6.1.

<i>intervalo</i>	<i>canais em uso</i>
00:00 - 00:30	$0 + 10 - 4 = 6$
00:30 - 01:00	$6 + 10 - 11 = 5$

Tabela 6.3: *Lista de utilização.*

$$\begin{bmatrix} h \\ util \end{bmatrix} = \begin{bmatrix} 0,5 & 1,0 \\ 6 & 5 \end{bmatrix} \quad (6.1)$$

Cada conjunto de eventos criado cobre o período de 60 dias, ou seja, os vetores utilizados nas verificações da rede neural possuem instantes de chegada que variam de $t = 0$ min até $t = 86400$ min, que constituem 2880 intervalos de 30 minutos. Como a *utilização prevista* ($util_{prev}$) para $h = h_i$ é igual a *utilização* ($util$) para $h = h_{i+1}$, a criação de 2881 intervalos é necessária, de forma a fornecer a *utilização prevista* para o último intervalo de tempo do vetor.

Ajuste de Limites

Antes de inserir os vetores para treinamento e simulação da rede neural, os valores devem ser ajustados para o intervalo $[-1; 1]$. Isso é necessário pois a função de transferência utilizada é a tangente hiperbólica (Seção 4.2.3.14).

6.1.1.1 Validação I

Os conjuntos de eventos criados para a primeira verificação foram obtidos a partir da curva de comportamento apresentada na Figura 6.2. A Figura 6.3 apresenta os dados da *lista de utilização* para a primeira semana.

Figura 6.2: Curva característica da Validação I.

Definido o vetor de entrada, que contém os valores $util$ e h , a variável $util_{prev}$ foi definida como sendo o valor da utilização no próximo intervalo, ou seja, para $h = h_i$, $util = util_i$ e $util_{prev_i} = util_{i+1}$.

Figura 6.3: Trecho da curva de utilização gerada para a Validação I.

Os vetores de entrada e alvo foram apresentados à rede para treinamento, e os pesos das sinapses foram ajustados. Concluído o treinamento, o segundo conjunto de eventos, criado a partir da mesma curva característica (Figura 6.2), foi utilizado para simulação da rede neural.

A Figura 6.4 apresenta os valores da utilização esperada (vetor alvo em verde) e os valores previstos (vetor resposta em azul) para o intervalo $[-1; 1]$.

Figura 6.4: Validação I: valores esperados e previstos.

Podemos constatar que os valores referentes ao vetor alvo (verde) apresentam uma variação muito maior que os presentes no vetor resposta (azul). A chegada de pedidos de conexão segue uma distribuição exponencial, o que implica uma curva de utilização com grande variação em torno da média, como apresentado na Figura 5.4, apresentada na Seção 5.2.1. A rede neural, treinada com esses dados, aprende o comportamento médio e, como consequência, a curva resultante possui um desvio muito menor.

6.1.1.2 Validação II

Após a validação do comportamento da rede neural com o primeiro conjunto de treinamento, o segundo conjunto foi inserido para verificar a adaptação da rede caso a curva de comportamento seja modificada.

Figura 6.5: Curva característica da Validação II.

Para isso, procedeu-se de forma análoga ao primeiro treinamento. Os dois conjuntos de eventos (um para treinamento e outro para validação) foram obtidos a partir da curva de comportamento apresentada na Figura 6.5. A Figura 6.6 apresenta os dados da *lista de utilização* para a primeira semana.

Figura 6.6: Trecho da curva de utilização gerada para a Validação II.

Figura 6.7: Validação II: valores esperados e previstos.

A rede neural foi, então, submetida a um treinamento com os novos vetores de entrada e alvo do primeiro conjunto de eventos, enquanto o segundo conjunto foi utilizado para simulação.

A Figura 6.7 apresenta os valores da utilização esperada (vetor alvo em verde) e os valores previstos (vetor resposta em azul) para o intervalo $[-1; 1]$.

Observa-se (Figura 6.7) que a resposta de rede se adaptou completamente ao novo comportamento, visto que aprendeu o novo padrão e realizou previsões corretamente. Assim, conclui-se que a rede neural se adapta facilmente a novos padrões caso a curva característica seja substituída.

6.1.1.3 Validação III

Como mencionado na Seção 4.2.3.14, a rede neural seria capaz de aprender comportamentos como os utilizados nas Validações I e II apenas com um neurônio de entrada, que receberia o valor da variável horário (h). Entretanto, células de redes móveis em produção não apresentam comportamentos tão simples. Comprovado o funcionamento da rede para os conjuntos de treinamento apresentados, passamos para a verificação de casos mais complexos.

Figura 6.8: Trecho da curva de utilização gerada para a Validação III.

Para essa nova validação, um terceiro conjunto de treinamento foi criado, com

base nas curvas características apresentadas nas Figuras 6.2 e 6.5. Os 60 dias foram divididos em semanas nas quais os 5 primeiros dias seguem a curva característica do primeiro conjunto de treinamento (Figura 6.2) e os outros 2 dias seguem a curva característica do segundo conjunto (Figura 6.5). Com isso, desejamos que a rede aprenda as duas curvas características e estime a utilização de acordo com o comportamento identificado.

A Figura 6.8 apresenta os dados da primeira semana do conjunto de eventos. A Figura 6.9 mostra um detalhe, destacando as diferenças entre os dias 5 e 6 da semana.

Figura 6.9: Detalhe da Figura 6.8 (quinto e sexto dias).

Os vetores de entrada e alvo obtidos foram, então, inseridos para o treinamento da rede neural. Para simulação, utilizamos os mesmos vetores de entrada inseridos para simulação nas Validações I e II. As respostas podem ser verificadas nas Figuras 6.10 e 6.11.

Os dados mostram que a rede foi capaz de aprender as duas curvas características. Entretanto, o comportamento que caracteriza 5 dos 7 dias da semana apresenta uma resposta mais definida que a curva que caracteriza os outros 2 dias. Esse desbalanceamento faz com que a rede tente, em princípio, acompanhar o comportamento predominante e, conseqüentemente, crie um ruído maior na previsão do outro comportamento. Isso ocorre porque as informações fornecidas na entrada não são suficientes para que

Figura 6.10: Validação III: relativo a 5 dias da semana.

o comportamento correto seja identificado (regiões em torno das 07:00 e em torno das 13:00 na Figura 6.11). Todavia, a resposta é satisfatória e atende às necessidades do sistema proposto.

Figura 6.11: Validação III: relativo a 2 dias da semana.

Cabe destacar que a resposta da rede pode ser melhorada se ampliarmos as informações fornecidas na camada de entrada. Uma possibilidade é inserir um histórico de utilização, fornecendo, por exemplo, os valores de $util_i$, $util_{i-1}$ e $util_{i-2}$ no vetor de entrada.

6.1.2 Análise das Validações

Pelas respostas obtidas nas três validações realizadas, constata-se que a rede neural tem a capacidade necessária para aprender as curvas características propostas. As Validações I e II mostram que a rede consegue se adaptar e alterar seu comportamento caso a curva característica da célula seja alterada, quer dizer, se um padrão anteriormente aprendido não for reforçado, o novo padrão se sobrepõe ao antigo treinamento da rede neural.

Por outro lado, se dois padrões são continuamente apresentados à rede e frequentemente reforçados, ela é capaz de aprender ambos e fornecer respostas satisfatórias.

6.1.3 Implementação

O pacote Joone, utilizado na construção da rede neural presente nos agentes, não apresentou resultados satisfatórios nos treinamentos realizados. Apesar de suportar diversas configurações e executarmos diversos testes de treinamento e validação pelo aplicativo gráfico que o acompanha, o aprendizado da rede neural de previsão construída com o Joone não convergiu. A partir da mesma arquitetura de rede neural, o treinamento através do MatLab apresentou resultados satisfatórios (dados em azul na Figura 6.12). Infelizmente, o mesmo não ocorreu através do treinamento pelo Joone (dados em vermelho na Figura 6.12).

Para o mesmo caso, se configurarmos as sinapses da rede neural construída através do Joone com os valores obtidos pelo treinamento no MatLab, a resposta da rede (dados em vermelho na Figura 6.13) não é tão boa quanto a obtida pelo MatLab (dados em azul na Figura 6.13), mas é satisfatória para a utilização na solução proposta.

Como a simulação não necessita de treinamento (Seção 5.3.3), os pesos das redes neurais foram obtidos por treinamentos com o MatLab. Os vetores obtidos configuraram as redes neurais construídas com o Joone.

Figura 6.12: Respostas para *Curva 3* (Treinamento com MatLab e Joone).

Figura 6.13: Respostas para *Curva 3* (Treinamento com MatLab).

6.2 Simulação

Um total de dez testes foram executados, utilizando o cenário apresentado no Capítulo 5. A cada simulação, o sistema escolheu, de forma aleatória, a curva característica de cada uma das 30 células presentes no cenário de teste e gerou os eventos de acordo com o comportamento definido.

Para cada uma das redes simuladas, referência e teste, o *Controle* registrou todos os pedidos de conexão enviados às células e as operações que os agentes executaram internamente para atender cada requisição, além de identificar se a chamada foi ou não aceita. A Tabela 6.4 apresenta a quantidade de chamadas bloqueadas nas células em cada simulação.

Pelos resultados obtidos, verifica-se que o sistema proporcionou um aumento glo-

Teste	Rede de Referência	Rede de Teste	Diferença	Ganho (%)
1	12.748	10.884	1.864	14,62
2	14.986	12.346	2.640	17,62
3	13.772	11.818	1.954	14,19
4	14.748	12.170	2.578	17,48
5	22.486	17.670	4.816	21,42
6	20.186	16.844	3.342	16,56
7	14.706	11.962	2.744	18,66
8	18.920	16.070	2.850	15,06
9	19.312	14.880	4.432	22,95
10	13.982	11.180	2.802	20,04
Total	165.846	135.824	30.022	18,10

Tabela 6.4: Chamadas bloqueadas em cada simulação.

bal na capacidade da rede. A rede de teste apresenta uma taxa de bloqueio de chamadas, em média, 18% menor que a apresentada pela rede de referência.

6.2.1 Mapas Comparativos

Conforme abordado na Seção 5.4, as redes simuladas trabalham próximas da capacidade limite. Nesse cenário, para que o sistema proporcione melhor desempenho em uma célula, é esperado que algumas das células vizinhas apresentem menor desempenho que a rede de referência, ou seja, maiores taxas de bloqueio. Em outras palavras, a redução da quantidade de chamadas recusadas em uma célula pode ocasionar um aumento dessa quantidade em células vizinhas.

Esse comportamento pode ser observado no *mapa comparativo* da Figura 6.14. O mapa da área de cobertura apresenta os resultados gerais da simulação 10 para cada célula da rede. As células esverdeadas indicam redução da quantidade de chamadas bloqueadas, enquanto as avermelhadas indicam aumento de pedidos bloqueados, ambos com relação à rede de referência. As demais células, acinzentadas, apresentaram as mesmas quantidades de chamadas bloqueadas nas redes de referência e teste.

Os mapas de todas as simulações efetuadas são apresentados no Apêndice A, Fi-

Figura 6.14: Mapa comparativo da Simulação 10.

guras A.1 a A.10. Os resultados contidos nos mapas englobam todo o período da simulação, a saber, todas as chamadas recebidas nas 24 horas simuladas.

6.2.2 Chamadas e Bloqueios

Figura 6.15: Comparativo de chamadas da simulação 10.

Para realizar uma análise mais profunda, comparamos os comportamentos das redes de referência e testes no decorrer das 24 horas simuladas. A Figura 6.15 apresenta as quantidades de chamadas recebidas, aceitas e recusadas, para as redes de referência e teste. Como ambas foram submetidas aos mesmos eventos, o número de chamadas recebidas é idêntico nas duas redes. Constata-se que a quantidade de chamadas bloqueadas pela rede de teste é sempre inferior à quantidade de bloqueios na rede de referência. Conseqüentemente, o desempenho da rede de teste é superior ao da rede de referência, por apresentar um maior número de conexões efetuadas.

Figura 6.16: Comportamento da célula 10 (simulação 10).

Do mesmo modo, o comportamento individual das células também pode ser verificado. Ainda considerando a simulação 10 (Figura 6.14), a Figura 6.16 contém o comportamento da célula 10, a qual apresentou redução na quantidade de chamadas bloqueadas, enquanto a Figura 6.17 apresenta os resultados para a célula 17, que aumentou o número de pedidos recusados.

Figura 6.17: Comportamento da célula 17 (simulação 10).

Os gráficos comparativos de todas as simulações efetuadas são apresentados no Apêndice A, Figuras A.11 a A.20. Os resultados contidos nos gráficos englobam todo o período da simulação, de forma a apresentar as quantidades de chamadas recebidas, aceitas e recusadas para todas as células nas 24 horas simuladas.

6.2.3 Realocação de Canais

Na rede de referência, os canais alocados em cada célula são fixos. Durante toda a simulação, os canais permanecem na célula mesmo se forem desnecessários. No

cenário simulado, cada célula permanece com os mesmos 24 canais alocados durante todo o período considerado.

Figura 6.18: Canais alocados na célula 10 (simulação 10).

Por outro lado, as células da rede de testes não possuem canais fixos pré-determinados. Cada célula aloca e desaloca canais conforme a necessidade e situação identificadas pelo agente. Ademais, as condições das células vizinhas também podem interferir. A quantidade de canais alocados pode ser alterada através de mensagem de solicitação e mensagem de colaboração trocadas entre os agentes vizinhos.

A Figura 6.18 apresenta esse comportamento para a célula 10 da simulação 10. O gráfico mostra a quantidade média de canais alocados a cada intervalo de 30 minutos.

Comparando os valores apresentados pela Figura 6.18 com a Figura 6.16, observamos que a quantidade de canais alocados na célula acompanhou a demanda, ou seja, o agente controlou o número de canais alocados na célula de forma a possuir recursos suficientes para suportar a quantidade de chamadas recebidas em cada período.

7 CONSIDERAÇÕES FINAIS

As redes de comunicação são formadas por diversos elementos distribuídos. Em redes móveis, as diversas antenas existentes são controladas e gerenciadas por equipamentos com crescente capacidade de processamento. Essa infra-estrutura foi explorada pelo sistema proposto através da técnica de Sistemas Multi-Agentes. Um agente é designado para cada antena, o qual decide quais recursos serão utilizados e quais pedidos serão aceitos.

Os agentes cognitivos utilizados empregam a Teoria Nebulosa para realizar a tomada de decisão. Essa técnica possibilita que os estados do agente e do ambiente sejam capturados qualitativamente, permitindo transições suaves entre as possíveis respostas a eventos externos.

Redes Neurais Artificiais foram inseridas para melhorar a percepção dos agentes, oferecendo uma previsão do estado do mundo em um instante futuro. Essa previsão é utilizada pelas regras de inferência nebulosa na identificação da situação interna do agente.

7.1 Cumprimento dos Objetivos

A simulação forneceu resultados satisfatórios e indicou que o sistema atingiu os objetivos definidos. Em todos os testes executados, a rede de teste obteve desempenho superior à de referência, ambas submetidas aos mesmos conjuntos de eventos em cada simulação.

As técnicas de Inteligência Artificial foram incorporadas ao sistema em pontos específicos, auxiliando na solução completa do problema. A aplicação de Teoria Nebulosa e Redes Neurais Artificiais forneceu flexibilidade ao sistema, permitindo a adaptação do comportamento caso novos cenários se apresentem.

A escolha da arquitetura descentralizada, implementada na forma de agentes, apresenta maior disponibilidade, uma vez que não existe um ponto único de falha. Todavia, necessita de maior troca de mensagens e de maior poder de processamento nos elementos de borda da rede. Ainda que exija mais dos elementos de rede, os agentes podem ser facilmente incorporados aos equipamentos em uso atualmente, com pouco ou nenhum investimento em hardware.

7.2 Contribuições

Uma das contribuições do trabalho é a criação de um sistema híbrido com aplicação de três técnicas de Inteligência Artificial. O problema em questão foi analisado e foram identificados os blocos que poderiam se beneficiar das vantagens oferecidas por cada técnica. A aplicação de cada técnica foi detalhada destacando como foi implementada e incorporada ao sistema final.

Ressaltamos, também, a solução proposta para resolver a escassez de recursos de rádio em uma rede sem fio. O trabalho apresenta um sistema altamente flexível e configurável, que permite a otimização do uso de canais. Além disso, o sistema possui grande disponibilidade e pode ser facilmente adicionado às redes existentes.

Outra contribuição é a modelagem da rede para a simulação do sistema. Para verificar o funcionamento e o desempenho do sistema, um cenário de teste foi modelado e implementado. A geração de eventos é apresentada de forma detalhada, destacando os principais fatores envolvidos. O ambiente de simulação apresentado pode ser utilizado para verificar o comportamento de redes móveis em outras situações, permitindo sua

utilização em diversas classes de problemas.

7.3 Otimizações

O simulador criado para validação do sistema utilizou uma versão simplificada do modelo proposto e as células foram submetidas a eventos gerados aleatoriamente. Embora a criação dos eventos seguisse distribuições que acompanhavam situações reais, a utilização de dados provenientes de uma rede em produção produziria resultados muito mais precisos.

Algumas modificações no simulador poderiam melhorar a resposta do sistema. O ideal seria criar um cenário real de uma rede celular, a partir de dados de uma rede comercial, e utilizá-lo para treinamento das redes neurais de previsão, além de auxiliar a definição das regras nebulosas.

Nesse contexto, diferentes conjuntos nebulosos e regras de inferência poderiam ser empregados, diferenciando o comportamento dos agentes do sistema. Assim, os critérios para alocação, desalocação, solicitação e oferecimento de canais seriam diferentes, possibilitando um melhor aproveitamento da capacidade de deliberação dos agentes.

Além disso, de acordo com a estratégia da rede, algumas células poderiam ter prioridade sobre outras, orientando os agentes em casos de impasse, quando diversos agentes de uma vizinhança necessitassem de recursos.

7.4 Trabalhos Futuros

Alguns pontos importantes não foram abordados neste trabalho. Entre outros, ressaltamos os aspectos de segurança, processamento e método de previsão.

A realocação de canais nas células tem o objetivo de acompanhar a demanda exis-

tente. Entretanto, pedidos falsos podem levar a rede a um desbalanceamento, reduzindo a capacidade total de prestação de serviços. Pedidos de conexão podem ser enviados a uma célula com a única intenção de congestioná-la e reduzir a quantidade de recursos nas células vizinhas (ataque de negação de serviço). Dependendo da configuração da rede, diversas realocações podem ocorrer, acarretando a transferência de um grande número de canais para a célula atacada. Com isso, além de a célula atacada não conseguir atender os usuários legítimos, outras células na rede podem ter seu desempenho degradado.

Outro aspecto não resolvido completamente é a comunicação entre os agentes. Ela pode ocorrer tanto por redes cabeadas como por canais de administração configurados nas frequências de rádio. Em ambos os casos, a segurança deve ser analisada, se não para criptografar as mensagens, pelo menos para garantir a autenticidade delas. Esse aspecto se torna crítico e essencial se as comunicações forem realizadas por canais de rádio. O impacto de um protocolo de comunicação criptografado, que exige maior capacidade de processamento, também deve ser avaliado.

É possível, ainda, utilizar a estrutura do sistema para resolver outras classes de problemas. Alguns exemplos são roteamento de pacotes, reconhecimento de padrões de utilização, identificação de fraudes e detecção de intrusão.

Por fim, existe a técnica de previsão de utilização implementada. Pode-se verificar o impacto de alterações na rede neural, como mudança de arquitetura, redimensionamento das camadas e/ou variáveis de entrada. Outra linha é comparar os resultados obtidos através de redes neurais com outras técnicas para previsão ou aproximação de funções, verificando a aderência ao problema e a necessidade de processamento.

REFERÊNCIAS

- AUSTIN, J. L. *How to Do Things with Words*. London: Oxford University Press, 1962.
- AVERKIN, A. N.; TARASOV, V. B. The fuzzy modelling relation and its application in psychology and artificial intelligence. *Fuzzy Sets and Systems*, v. 22, p. 3–24, 1987.
- BAIOCCHI, A.; SESTINI, F. Near-optimality of distributed load-adaptive dynamic channel allocation strategies for cellular mobile networks. *Wireless Networks*, v. 2, p. 129–142, 1996.
- BARRETO, J. M. *Introdução às Redes Neurais Artificiais*. Monografia (Monografia) — Laboratório de Conexionismo e Ciências Cognitivas, Departamento de Informática e de Estatística da Universidade Federal de Santa Catarina, Florianópolis, SC, 2002.
- BARROS, L. C. *Modelos determinísticos com parâmetros subjetivos*. Dissertação (Mestrado) — Instituto de Matemática, Estatística e Ciência da Computação da Universidade de Campinas, Campinas, Brasil, 1992.
- BARROS, L. C.; BASSANEZI, R. C.; TONELLI, P. A. Fuzzy modeling in population dynamics. *Ecological Modelling*, v. 128, p. 27–33, 2000.
- BECK, R.; PANZER, H. Strategies for handover and dynamic channel allocation in micro-cellular mobile radio telephone systems. *IEEE Vehicular Technology Conference*, v. 1, p. 178–185, 1989.
- BOND, A. H.; GASSER, L. *Readings in Distributed Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann, 1988.
- BORDINI, R. H.; VIEIRA, R.; MOREIRA, Álvaro F. *Fundamentos de Sistemas Multiagentes*. Monografia (Monografia) — Instituto de Informática da Universidade Federal do Rio Grande do Sul; Centro de Ciências Exatas e Tecnológicas, Centro de Ciências da Comunicação da Universidade do Vale do Rio dos Sinos; Departamento de Informática da Universidade de Caxias do Sul, Porto Alegre, RS; São Leopoldo, RS; Caxias do Sul, RS, 2001.
- BORDLEY, R. Fuzzy set theory, observer bias and probability theory. *Fuzzy Sets and Systems*, v. 33, p. 347–354, 1989.
- BOUKERCHE, A.; EL-KHATIB, K.; HUANG, T. A performance comparison of dynamic channel and resource allocation protocols for mobile cellular networks. *MobiWac*, p. 27–34, 2004.

- BOUKERCHE, A.; HONG, S.; JACOB, T. A distributed algorithm for dynamic channel allocation. *Mobile Networks and Applications*, v. 7, p. 115–126, 2002.
- BRATMAN, M. E. *Intentions, Plans and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.
- BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. Plans and resource-bounded practical reasoning. *Computational Intelligence*, n. 4, p. 349–355, 1988.
- CAO, G.; SINGHAL, M. Efficient distributed channel allocation for mobile cellular networks. *Proceedings of IEEE 7th International Conference on Computer Communications and Networks*, p. 50–57, 1998.
- _____. An adaptive distributed channel allocation strategy for mobile cellular networks. *Journal of Parallel and Distributed Computing*, v. 60, p. 451–473, 2000.
- _____. An adaptive distributed channel allocation strategy for mobile cellular networks. *Journal of Parallel and Distributed Computing*, p. 451–473, 2000.
- CHOY, M.; SINGH, A. Efficient distributed algorithm for dynamic channel assignment. *Proceedings of IEEE 7th International Symposium on Personal, Indoor and Mobile Radio Communication*, 1996.
- _____. Efficient distributed algorithm for dynamic channel assignment. *Proceedings of 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, 1996.
- COX, D. C.; REUDINK, D. O. Increasing channel occupancy in large scale mobile radio systems: Dynamic channel reassignment. *IEEE Transactions on Communications*, v. 21, n. 11, p. 1302–1306, 1973.
- DAS, S. K.; SEN, S. K.; JAYARAM, R. A dynamic load balancing strategy for channel assignment using selective borrowing in cellular mobile environment. *Wireless Networks*, v. 3, p. 333–347, 1997.
- _____. D-lbsb: A distributed load balancing algorithm for channel assignment in cellular mobile networks. *Journal of Interconnection Networks*, v. 1, n. 3, p. 195–220, 2000.
- DENNETT, D. C. *The Intentional Stance*. Cambridge, MA: The MIT Press, 1987.
- ERICSSON. *GSM System Survey: Student Text*. 1998. EN/LZT 123 3321 R3A.
- FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. [S.l.]: Addison-Wesley, 1999.
- FININ, T.; FRITZSON, R.; MCKAY, D. A language and protocol to support intelligent agent interoperability. In *Proceedings of the CE & CALS Washington '92 Conference*, 1992.
- FININ, T. et al. Kqml as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press, 1994.

- FININ, T. et al. *Specification of the KQML agent-communication language (plus example agent policies and architectures)*. Monografia (Monografia) — Draft, The DARPA Knowledge Sharing Initiative-External Interfaces Working Group, 1994.
- FONSECA, D. J.; KNAPP, G. M. An expert system for reliability centered maintenance in the chemical industry. *Expert Systems with Applications*, v. 19, p. 45–57, 2000.
- FRANKLIN, S.; GRAESSER, A. Is it an agent, or just a program?: A taxonomy for autonomous agents. *Intelligent Agents III - Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), ECAI'96 Workshop*, n. 1193, p. 21–35, 1997.
- GARG, N.; PAPATRIANTAFILOU, M.; TSIGAS, P. Distributed list coloring: How to dynamically allocate frequencies to mobile base stations. *8th IEEE Symposium on Parallel and Distributed Processing*, p. 18–25, 1996.
- GASSER, L.; HUHNS, M. N. *Distributed Artificial Intelligence, volume II - Research Notes in Artificial Intelligence*. London; San Mateo, CA: Pitman; Morgan Kaufmann, 1989.
- GENESERETH, M. R.; FIKES, R. E.; OUTROS. *Knowledge interchange format version 3.0 reference manual*. Monografia (Monografia) — Stanford University, Computer Science Department, Logic Group, Stanford University, Stanford, CA, 1992.
- GENESERETH, M. R.; KETCHPEL, S. P. Software agents. *Communications of the ACM*, n. 37, p. 48–53, 1994.
- GENESERETH, M. R.; SINGH, N. P. *A knowledge sharing approach to software interoperation*. Monografia (Monografia) — Logic Group, Department of Computer Science, Stanford University, Stanford, CA, 1994.
- GRIFFA, E. *Tendencias en Tecnología Celular*. 2005. Apresentação. [http://www.fain.uade.edu.ar/simcel2005/presentaciones/mobile systems market update abril 2005.pdf](http://www.fain.uade.edu.ar/simcel2005/presentaciones/mobile%20systems%20market%20update%20abril%202005.pdf), acessado em 06 de fevereiro de 2006.
- HAC, A.; CHEN, Z. A hybrid channel allocation method for wireless communication networks. *International Journal of Network Management*, p. 59–74, 2000.
- HEBB, D. O. *The organization of behavior*. New York: Wiley, 1949.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359–366, 1989.
- HUHNS, M. N. *Distributed Artificial Intelligence - Research Notes in Artificial Intelligence*. London; San Mateo, CA: Pitman; Morgan Kaufmann, 1987.
- JAIN, R. *The Art of Computer Systems Performance Analysis*. [S.l.]: John Wiley & Sons Inc., 1991. 685 p. ISBN 0-471-50336-3.

- KLIR, G.; YUAN, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. USA: Prentice-Hall, 1995.
- KOSKO, B. *Fuzzy Engineering*. USA: Prentice-Hall, 1997.
- LABROU, Y. *Semantics for an Agent Communication Language*. Monografia (Monografia) — Computer Science and Electrical Engineering Department, University of Maryland Graduate School, Baltimore, MD, 1996.
- LABROU, Y.; FININ, T. A semantics approach for kqml-a general purpose communication language for software agents. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press, 1994.
- LAWLOR, M.; WHITE, T. A self organizing social insect model for dynamic frequency allocation in cellular telephone networks. *AAMAS*, p. 1048–1049, 7 2003.
- LEE, C. C. Fuzzy logic in control systems: Fuzzy logic controller - part i. *IEEE*, 1990.
- _____. Fuzzy logic in control systems: Fuzzy logic controller - part ii. *IEEE*, 1990.
- LEMAÎTRE, C.; EXCELENTE, C. B. Multi-agent organization approach. In: . [S.l.: s.n.], 1998.
- LINS, B. F. E. *Avanços Recentes no Setor de Telecomunicações e suas Implicações nos Campos Educacional e Social*. 2001. Estudo. Consultor Legislativo da Área IV - Comunicações, Informática, Ciência e Tecnologia.
- MANDANI, E. H. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the IEEE (Control and Science)*, v. 121, p. 298–316, 1974.
- _____. Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *Proceedings of the IEEE (Trans. Computers)*, v. 126, p. 1182–1191, 1977.
- MANDANI, E. H.; ASSILLAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, v. 7, p. 1–13, 1975.
- MAYFIELD, J.; LABROU, Y.; FININ, T. Desiderata for agent communication languages. In *Proceedings of the AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI-95 Spring Symposium, Stanford University*, 1995.
- _____. Evaluation of kqml as an agent communication language. *Intelligent Agents II- Proceedings of the Second International Workshop on Agent Theories, Architectures, and Languages (ATAL'95)*, n. 1037, p. 347–360, 1996.
- OH, S.; KIM, W.; LEE, J. K. An approach to causal modeling in fuzzy environment and its application. *Fuzzy Sets and Systems*, v. 35, p. 43–55, 1990.
- OH, S.-H.; TCHA, D.-W. Prioritized channel assignment in a cellular radio network. *IEEE Transactions on Communications*, v. 40, n. 7, p. 1259–1269, 1992.

- OKADA, K.; KUBOTA, F. On dynamic channel assignment strategies in cellular mobile radio systems. *IEICE Transactions on Fundamentals*, v. 75, p. 1634–1641, 1992.
- ORTEGA, N. R. S. *Aplicação da Teoria de Conjuntos Fuzzy a Problemas da Biomedicina*. Tese (Doutorado) — Instituto de Física da Universidade de São Paulo, São Paulo, Brasil, 2001.
- PATTAVINA, A.; QUADRI, S.; TRECORDI, V. Reuse partitioning in cellular networks with dynamic channel allocation. *Wireless Networks*, v. 5, p. 299–309, 1999.
- PIASECKI, K. Probability of fuzzy events defined as denumerable additivity measure. *Fuzzy Sets and Systems*, v. 17, p. 271–284, 1985.
- _____. Fuzzy probability spaces defined by means of the fuzzy relation 'less than'. *Fuzzy Sets and Systems*, v. 19, p. 273–289, 1986.
- PRAKASH, R.; SHIVARATRI, N. G.; SINGHAL, M. Distributed dynamic channel allocation for mobile computing. *Proceedings of 14th Annual ACM Symposium on Principles of Distributed Computing*, p. 47–56, 1995.
- _____. Distributed dynamic fault-tolerant channel allocation for mobile computing. *IEEE Transactions on Vehicular Technology*, p. 1874–1888, 11 1999.
- REZNIK, L. *Fuzzy Controllers*. Reino Unido: Newnes, 1997.
- RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, P. *Parallel Distributed Processing - vol 1: Foundations*. [S.l.]: The MIT Press, 1986.
- SEUNG, H. S.; SOMPOLINSKY, H.; TISHBY, N. Statistical mechanics of learning from examples. *Physical Review*, n. A45, p. 6056–6091, 1992.
- SHAW, I. S.; SIMÕES, M. G. *Controle e Modelagem Fuzzy*. Brasil: Edgard Blücher, 1999.
- SHEN, X.; MARK, J. W.; YE, J. User mobility profile prediction: An adaptive fuzzy inference approach. *Wireless Networks*, v. 6, p. 363–374, 2000.
- SICHMAN, J. S. Depint: Dependence-based coalition formation in an open multi-agent scenario. *Journal of Artificial Societies and Social Simulation*, v. 1, n. 2, p. 71–97, 1998.
- SICHMAN, J. S.; CONTE, R.; GILBERT, N. Multi-agent systems and agent-based simulation. *Proceedings of the International Workshop held as part of the Agents' World, Paris*, n. 1534, p. 4–7, 1998.
- SIMONETTI, R. *Generalização e Robustez: Aprendizagem em Redes Neurais na Presença de Ruído*. Tese (Doutorado) — Instituto de Física da Universidade de São Paulo, São Paulo, SP, 1997.
- SINGH, N. P.; GENESERETH, M. R.; SYED, M. A. A distributed and anonymous knowledge sharing approach to software interoperation. *International Journal of Cooperative Information Systems*, n. 4, p. 339–367, 1995.

- SOKOLOV, N. *Some Aspects of Russian Telecommunications*. 2006. Hipertext. [Http://www.comsoc.org/pubs/gcn/gcn0106.html](http://www.comsoc.org/pubs/gcn/gcn0106.html), acessado em 06 de fevereiro de 2006.
- STEIN, W. E. Fuzzy probability vectors. *Fuzzy Sets and Systems*, v. 15, p. 263–267, 1985.
- SUGENO, M. *Theory of fuzzy integrals and its applications*. Tese (Doutorado) — Instituto de Tecnologia de Tokio, Tokio, Japão, 1974.
- WATKIN, T. L. H.; RAU, A.; BIEHL, M. The statistical mechanics of a learning rule. *Reviews of Modern Physics* 65, p. 499–556, 1993.
- WI-FI.ORG. *Glossary*. Hipertext. <http://www.wi-fi.org/OpenSection/glossary.asp>, acessado em 06 de fevereiro de 2006.
- WOOLDRIDGE, M. Agent-based computing. *Interoperable Communication Networks*, v. 1, p. 71–97, 1998.
- WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, v. 10, n. 2, p. 115–152, 1995.
- YAGER, R. R.; FILEV, D. P. *Essential of Fuzzy Modelling and Control*. USA: John Wiley, 1994.
- ZADEH, L. A. Fuzzy sets. *Information and Control*, v. 8, p. 338–353, 1965.
- _____. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE*, 1973.
- ZHANG, M.; YUM, T.-S. P. Comparisons of channel-assignment strategies in cellular mobile telephone systems. *IEEE Transactions on Vehicular Technology*, v. 38, n. 4, p. 211–215, 1989.
- _____. The nonuniform compact pattern allocation algorithm for cellular mobile system. *IEEE Transactions on Vehicular Technology*, v. 40, n. 2, p. 387–391, 1991.
- ZHANG, Y.; DAS, S. K. An efficient load-balancing algorithm based on a two-threshold cell selection scheme in mobile cellular networks. *Computer Communications*, v. 23, n. 5–6, p. 452–461, 2000.
- ZHANG, Y.; DAS, S. K.; JIA, X. D-cat: An efficient algorithm for distributed channel allocation in cellular mobile networks. *Mobile Networks and Applications*, v. 9, p. 279–288, 2004.

Apêndice A – RESULTADOS: FIGURAS

A.1 Mapas Comparativos

Figura A.1: Mapa comparativo da Simulação 1.

Figura A.2: Mapa comparativo da Simulação 2.

Figura A.3: Mapa comparativo da Simulação 3.

Figura A.4: Mapa comparativo da Simulação 4.

Figura A.5: Mapa comparativo da Simulação 5.

Figura A.6: Mapa comparativo da Simulação 6.

Figura A.7: Mapa comparativo da Simulação 7.

Figura A.8: Mapa comparativo da Simulação 8.

Figura A.9: Mapa comparativo da Simulação 9.

Figura A.10: Mapa comparativo da Simulação 10.

A.2 Chamadas

Figura A.11: Comparativo de chamadas da simulação 1.

Figura A.12: Comparativo de chamadas da simulação 2.

Figura A.13: Comparativo de chamadas da simulação 3.

Figura A.14: Comparativo de chamadas da simulação 4.

Figura A.15: Comparativo de chamadas da simulação 5.

Figura A.16: Comparativo de chamadas da simulação 6.

Figura A.17: Comparativo de chamadas da simulação 7.

Figura A.18: Comparativo de chamadas da simulação 8.

Figura A.19: Comparativo de chamadas da simulação 9.

Figura A.20: Comparativo de chamadas da simulação 10.